Proceedings of the 2014 Winter Simulation Conference
A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, eds.

SCHEDULING PREVENTIVE MAINTENANCE TASKS WITH SYNCHRONIZATION CONSTRAINTS FOR HUMAN RESOURCES BY A CP MODELING APPROACH

Jan Lange
Dirk Doleschal
Gerald Weigert

Electronics Packaging Laboratory
Technische Universität Dresden
D-01062 Dresden, GERMANY

Andreas Klemmt

Infineon Technologies
Königsbrücker Str. 180
D-01099 Dresden, GERMANY

ABSTRACT

This paper presents an approach for scheduling different types of preventive maintenances (PMs) for a work center of a semiconductor manufacturing facility. The PM scheduling problem includes time-dependent synchronization constraints and is implemented in a constraint programming model. A mix of periodic and workload-specific maintenances is scheduled considering the synchronization to available engineers which have individual shift schedules and skills that define the range of feasible maintenances. This also comprises maintenances having process durations covering multiple shifts, which requires a continuous availability of sufficiently skilled engineers. Additionally to the PMs, also handling and maintaining of unscheduled downs is considered in the model. Multiple objectives are investigated and used for optimization and tested on realistic data. To compare the results an additional simulation model is built up.

1 INTRODUCTION

In this paper the scheduling of preventive maintenance (PM) tasks is investigated. For this scheduling problem, a constraint programming (CP) based optimization approach is presented. This CP model deals with different requirements, which make scheduling complex. These requirements are for example serial crew constraints or different parallel maintenances. Due to the complexity of the investigated problem and its structure, an integer programming formulation was not considered as practicable option. So, a CP formulation was preferred. Thereby, the CP model was implemented in the Optimization Programming Language (OPL) within IBM ILOG CPLEX Optimization Studio 12.6. However for describing the constraints and objectives, a more general mathematical description is used here.

1.1 Literature Overview

The investigated problem is generally a personnel scheduling problem. A comprising literature review for personnel scheduling is presented in Van den Bergh et al. 2013. Other reviews can also be found in Ernst 2004 and Alfares 2004, where the literature published since 1990 is classified for the tour scheduling problem. Abdennadher and Schlenker 1999 proposed a constraint logic programming approach for a nurse scheduling problem that involves three shifts per day. He, F. and Qu, R. 2012 present a hybrid constraint programming based column generation approach to nurse rostering problems. In Laporte and Pesant 2004 a CP-based approach is presented for a manpower planning scheduling problem with multi-shift rotating schedules. A crew composition problem is considered in Li and Womer 2009. Another publication about maintenance service and staff planning is presented by Lilly et al. 2007, which addresses a long term capacity planning problem for evaluating the cost-effectiveness of a 4 day work schedule for power-generating stations. Lange et al. 2013 developed a constraint programming based model to schedule maintenance tasks with time dependent synchronization constraints where crews are needed to perform a maintenance task. In comparison with this in this paper no crew constraint exist. Otherwise here different maintenance can be scheduled in parallel, the granularity is much finer and the preventive maintenance tasks could be shift overlapping.

1.2 Environment Description

The general objective is the creation of an optimized schedule for preventive maintenances and human resources (engineers) which are assigned to maintenances. For this the created schedule shall comprise a time horizon of about one week.

The number of considered engineers is about several dozen. Each engineer has individual qualification skill levels, which are defined specifically for all considered machine groups. Those different qualification levels can be 1, 2 or 3. Thereby, an engineer with high qualification level can also perform actions which require a lower qualification level, but not vice versa. The general shift schedule has 3 fixed shifts per day, each with a duration of 8 hours, thus maintenances can be performed 24 hours a day. There is an individual shift schedule for every engineer separately.

The model comprises three different machine groups, each with multiple cluster tools. A machine group again consists of multiple identical machines whose maintenances can be performed with equal qualification skills. A cluster tool again comprises multiple process chamber and load ports, which are in following also named as entities. There are three types of maintenance activities. Firstly, short PMs (SPMs) are maintenances, which usually last about 3 to 4 hours and are performed by one engineer within a shift. The qualification level of the engineer must be 1 or higher. The common planning horizon for those shifts is about a few days. This depends on the machine load. Secondly, long PMs (LPMs) are usually performed by a group of engineers over several shifts without any gap, where only one engineer is required within each shift. For this, all engineers need to have a qualification level of 2 or higher. The duration of long PMs is usually about 1 to 2 days and the time horizon is about 1 month. Thirdly, unscheduled activities (DPMs) are to be performed, which comprise the repairs and maintenances for all machines which are down. Those activities always start immediately and can last about a few hours up to a few days. Thus engineers might be planned over several shifts. Furthermore the engineer qualification levels should be as high as possible.

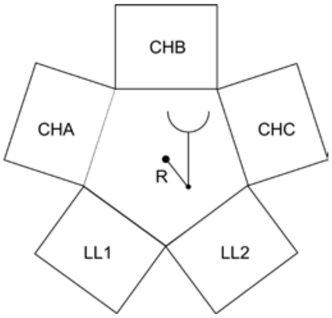


Figure 1: Example structure of a cluster tool

Figure 2: Cluster Tool

Short PMs are always assigned to single entities (chambers or load ports) of a cluster tool. Other entities of that machine are not affected from these short PMs. On the other hand, long PMs are always assigned to whole machines, which affects all connected entities of those machines. Unscheduled repairs and maintenances can be both, assigned to whole machines or single entities.

Short PMs are basically load-dependent maintenances, this means by influencing the load on the machine, also the domain of the next expected short PM can be varied and controlled. However in the provided model the machine load is not altered (mostly machines have high work load).

2 MODEL

The model covers a time horizon of one week with a detail level of 0.5 hours. This means there is a time vector $T_o\left(o=1..t\right)$ with t is 336.

There are mg machine groups $MG_u\left(u=1..mg\right)$ with m_u machines $M_{u,k}\left(k=1..m_u\right)$. Those machines are cluster tools containing of $ch_{u,k}$ chambers or load ports (entity) $CH_{u,k,c}\left(u=1..mg;k=1..m_u;c=1..ch_{u,k}\right)$. The structure is shown in Figure 3.

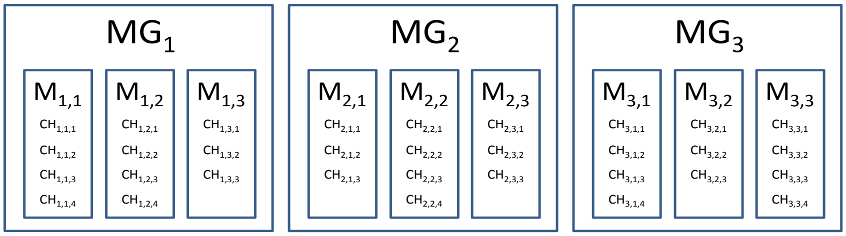


Figure 3: General structure of machine groups, machines and entities

To perform the maintenances, there are e engineers $E_n\left(n=1..e\right)$. These engineers have individual shift schedules $S_{n,o}\left(n=1..e;o=1..t\right)\in\{0,1\}$, where 1 means that the according engineer n is available (for 0 not available) at time slot o. Additionally, engineers have individual qualification skill levels $Q_{n,u}=\left(n=1..e;u=1..mg\right)$ for each machine group mg where $Q_{n,u}\in\{1,2,3\}$. Three different maintenance types are to be planned. There are spm short PMs $SP_{sl}\left(sl=1..spm\right)$, lpm long PMs $LP_{ll}\left(ll=1..lpm\right)$ and dpm downtime or blocking tasks $DP_{dl}\left(dl=1..dpm\right)$. Each SPM task is assigned to one entity $CH_{u,k,c}$. Each LPM task is assigned to a cluster tool *of* $M_{u,k}$ and all connected entities $CH_{u,k,c}$. Each DPM task can be whether assigned to a single entity or to a whole machine. Altogether, there are pm PMs $PM_l\left(l=1..pm\right)$ where $pm=spm+lpm+dpm$ and $PM_l=\bigcup\left(SP_{sl},LP_{ll},DP_{dl}\right)$.

We define auxiliary functions $f_1\left(sl,n,o\right)\in\{0,1\}$ which describes that a SPM task SP_{sl} is processed by an engineer E_n at time slot o; function $f_2\left(ll,n,o\right)\in\{0,1\}$ which describes that a LPM task LP_{ll} is processed by an engineer E_n at time slot o; function $f_3\left(dl,n,o\right)\in\{0,1\}$ which describes that a DPM task DP_{dl} is processed by an engineer E_n at time slot o. The function $f_{all}\left(l,n,o\right)\in\{0,1\}$ with $f_{all}=f_1+f_2+f_3$ is the combination of the separated functions.

2.1 Basic Structure/Constraints

Defining the time of the scheduled start of a task as $Task^{Start}$ and the time of the scheduled end of a task as $Task^{End}$, it is state, that all time slots of a task are concatenated which means that $Task^{Start}+Task^{Duration}=Task^{End}$. For creating the CP model the following constraints are defined:

PM task domain (C1)

PM tasks are to be scheduled inside the bounds of their domains. In the following, D_l^{start} and D_l^{end} are the start and the end of a PM task domain D_l.

$$\forall x \in \{1,2,3\}: \quad \forall l,n,o: \quad \left(\left(o < D_l^{start}\right) \vee \left(o > D_l^{end}\right)\right) \rightarrow \left(f_x\left(l,n,o\right) = 0\right) \tag{1}$$

In the OPL model this is ensured by initializing interval sets for short PMs (SPMs), long PMs (LPMs) and DPMs, which comprising all maintenance tasks whose domains start with D_l^{start} and end with D_l^{end}.

Engineers shift schedule (C2)

Engineers can perform tasks only in their individual shift schedule. This means, if a function f_1, f_2 or f_3 is true for a given engineer *n* and a given timeslot *o*, also the according shift availability $S_{n,o}$ is required to be true.

$$\forall x \in \{1,2,3\}: \quad \forall l,n,o: \quad \left(f_x\left(l,n,o\right) = 1\right) \rightarrow \left(S_{n,o} = 1\right) \tag{2}$$

In OPL an array with optional intervals was defined, which describes the set of engineers for each SPM. The dimension of the array is $|SP| \times |E|$. For each SPM the interval domains were limited to the specific SPM domains. Additionally, by a *ForbidExtend* constraint, the domains of the array intervals were restricted according to the specific engineer availability were $S_{n,o}\left(n = 1..e; o = 1..t\right) \in \{0,1\}$ is not 0.

Engineers are single resources (C3)

Engineers can perform not more than a single task simultaneously.

$$\forall n,o: \sum_{l=1}^{p} f_{all}\left(l,n,o\right) \leq 1 \tag{3}$$

This constraint has also the effect of adjusting the heterogeneous LPMs, SPMs and DPMs by limiting the sum of engineer tasks.

In OPL this was realized by a *noOverlap* constraint over all intervals of each engineer.

Chambers and Machines are single resources (C4)

Only one PM-task can be performed on a machine simultaneously. The function $g\left(l,k,o\right) \in \{0,1\}$ describes that a PM task P_l on machine M_k is active at time (slot) T_o.

$$\forall k,o: \sum_{l=1}^{p} g\left(l,k,o\right) \leq 1 \tag{4}$$

In OPL each PM has also information about the machine in which the PM is performed. Thus, a *noOverlap* constraint was defined for all machines over all PM intervals (SPM, LPM and DPM).

Engineer Assignment (C5)

The engineers are assigned in dependency of the PM type. However equation (5) is used for all types:

$$\forall l \in 1..pm: \quad \min_{\substack{n=1..e \\ f_{all}(l,n,o)=1}} \left(o\right) + PM_l^{Duration} = \max_{\substack{n=1..e \\ f_{all}(l,n,o)=1}} \left(o\right) \tag{5}$$

For SPMs:

$$\forall sl \in 1..spm: \quad \sum_{\substack{n=1..e \\ o=1..t}} f_1\left(sl,n,o\right) = SP_{sl}^{Duration} \tag{6}$$

$$\forall n \in E, sl \in 1..spm: \quad \sum_{\substack{n=1..e \\ o=1..t}} f_1\left(sl,n,o\right) = \left\{0, SP_{sl}^{Duration}\right\} \tag{7}$$

In OPL this is done by the *Alternative* constraint, which assigns/selects for each short PM task one engineer task to that short PM task.

For LPMs:

LPMs usually cover multiple shifts where are group of sufficiently skilled engineers are continuously available. This is described by formula (5) and (8).

$$\forall ll \in 1..lpm: \sum_{\substack{n=1..e \\ o\in 1..t}} f_2(ll,n,o) = LP_{ll}^{Duration} \tag{8}$$

In OPL an interval set for all long PMs was created. Furthermore, a three dimensional array of optional engineer tasks was created. A *span* constraint was used to span the long PM interval over all active engineer tasks of the long PM. Engineer tasks were additionally constrained by *noOverlap*. Furthermore, engineer tasks are forced to concatenate without a gap by a *startAtEnd* constraint.

Only engineer tasks of engineers having a skill level of 2 or 3 for this machine can be active.

For DPMs:

$$\forall dl \in 1..dpm: \sum_{\substack{n=1..e \\ o=1..t}} f_3(dl,n,o) = DP_{dl}^{Duration} \tag{9}$$

$$\forall n \in E, dl \in 1..dpm: \sum_{\substack{n=1..e \\ o=1..t}} f_3(dl,n,o) = \left\{0, DP_{dl}^{Duration}\right\} \tag{10}$$

By definition DPM tasks run from the beginning. However, it may be that initially more DPMs incur as engineers are available. This would lead to a model with no solution. To prevent this, the possible start of a DPM is flexible and thus can also start later. However, by optimizing it is always tried to ensure the earliest possible start of the DPMs.

Qualification Constraints(C6)

To perform maintenances, engineers need specific skills. For short PMs all skills are allowed, for long PMs and DPM are restricted to engineers with skill level 2 and 3. Where long PMs are primarily are to be done with level 2 and DPMs are primarily to be done with level 3.

2.2 Soft Constraints

Reserve available soft constraint (SC1)

A reserve team of two engineers is to be established for every time slot of the whole planning horizon. However, this constraint may not be satisfied due to the variation of engineers' presences and the number of unscheduled maintenances.

Two optional tasks per time slot *o* and a size of 1 were created. So, $SC1_{o,k} \in \{0,1\}$ is defined for each time slot *o* and two optional reserve tasks ($k = 1..2$) As well as in *C2*, reserve tasks are restricted by engineer availabilities, which is shown in formula (11). Further these tasks are scheduled together with all other tasks which is shown in formula (12).

$$\forall n,o: \left(f_r(n,o) = 1\right) \rightarrow \left(S_{n,o} = 1\right) \tag{11}$$

$$\forall n,o: f_r(n,o) + \sum_{l=1}^{p} f_{all}(l,n,o) \leq 1 \tag{12}$$

2.3 Objectives

Multiple objectives were defined and optimized separately but also combined.

2.3.1 Reserve Fit

This objective uses the soft constraint *SC1* with the aim of maximizing the number of time slots where the reserve condition is fulfilled over the planning horizon (13).

$$ReserveFit = \sum_{o=1}^{t} \sum_{k=1}^{2} SC1_{o,k} \rightarrow \max \tag{13}$$

For the whole week, it is counted each feasible reserve task. Consequently, with 336 time slots and 2 reserve tasks per time slot, there is a maximum of 672 for this constraint.

2.3.2 Ideal Delay

Every PM task has a predefined domain in which this task is to be performed. Delaying those maintenances inside their domains means also delaying future maintenances of concerning machines. However, to avoid domain violations through uncertain influences in reality, maintenances should not be performed at the very end of their domains. Consequently, maintenance tasks are to be performed preferably between the third and fourth quarter of their domains. In the model this is implemented by triangular functions defined over each domain of maintenance task. This is shown in Figure 4. The function $V_l(o)$ is the triangular function for a PM task *l* and is described in equation (14). For every PM task, its start date is used as input for its triangular function multiplied. Thus the sum of these functions is to be maximized (15).

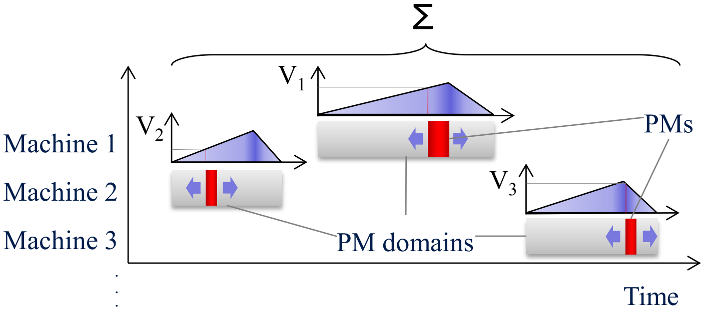


Figure 4: IdealDelay function over each PM task domain

$$V_l(o) = \begin{cases} \frac{4}{3} \cdot \frac{o - D_l^{start}}{D_l^{end} - D_l^{start}} & D_l^{start} \le o \le \frac{3}{4} D_l^{end} \\ 1 - \left(4 \cdot \frac{o - D_l^{start}}{D_l^{end} - D_l^{start}} - 3\right) & \frac{3}{4} D_l^{start} < o \le D_l^{end} \\ 0 & o < D_l^{start}, o > D_l^{end} \end{cases} \tag{14}$$

$$IdealDelay = \sum_{l=1}^{p} V_l(P_l^{Start}) \rightarrow \max \tag{15}$$

2.3.3 Simultaneous PM (SimPM)

One of the basic objectives is the minimization of simultaneously performed PM tasks to assure low workforce utilization. The principle of the objective is illustrated in Figure 5.

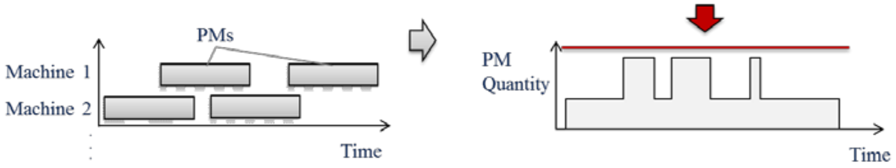


Figure 5: Minimizing simultaneously performed PM tasks

For this, a function is defined over the whole time (global) of the planning horizon cumulating the number of active maintenance tasks at time o (16). Consequently, the upper bound of this function is minimized (17) in order that the number of active maintenance tasks does not exceed the maximum $simPM_{\max}$ for the whole planning horizon.

$$simPM_o = \sum_{l=1}^{p} \left(\sum_{n=1}^{e} f\left(l,n,o\right) \ge 1 \right) \tag{16}$$

$$\max_{o=1..t}\left(simPM_o\right) = simPM_{\max} \rightarrow \min \tag{17}$$

2.3.4 ShiftPM

The optimization with respect to simultaneous maintenance can sometimes lead to results, where single shifts still have to process a high load. In order to obtain a very uniform load over all shifts, the maximal number of SPMs and LPMs per shift is minimized. This is represented by the formulas (18) and (19), where s defines a shift with a duration of 16 time slots (8 hours). Thus, 21 shifts cover the whole week of the planning horizon.

$$\forall s = 1..21: \quad shiftPM_s = \sum_{\substack{sl \in SP \\ n \in E \\ 16(s-1) \le o < 16s}} f_1\left(sl,n,o\right) + \sum_{\substack{ll \in LP \\ n \in E \\ 16(s-1) \le o < 16s}} f_2\left(ll,n,o\right) \tag{18}$$

$$\max_{s=1..21}\left(simPM_s\right) = simPM_{\max} \rightarrow \min \tag{19}$$

2.3.5 SkillSum

As introduced before, different kinds of PMs need different skill levels. SPM require at least level 1, LPM and UPM require at least level 2. This objective aims the usage of low level engineers for SPMs, this results in a maximized number high skilled engineers for upcoming DPMs covered by prevented reserve team. The following formula defines the minimization of schedule engineering skills for SPMs.

$$SkillSum = \sum_{\substack{sl=1..spm \\ n=1..e \\ o=1..t}} \left(f_1\left(sl,n,o\right) \cdot Q_{n,u} \right) \quad \text{where } machine(SP_{sl}) \in M_{u,k} \tag{20}$$

$$SkillSum \rightarrow \min \tag{21}$$

2.3.6 Combined Objective

The model contains a combined objective function (22) for optimization which contains all introduced single objectives each connected with an individual weight w.

$$\begin{aligned} CombineObjective = w_{ReserveFit} \cdot ReserveFit + w_{IdealDelay} \cdot IdealDelay + w_{SimPM} \cdot SimPM \\ + w_{ShiftPM} \cdot ShiftPM + w_{SkillSum} \cdot SkillSum \end{aligned} \tag{22}$$

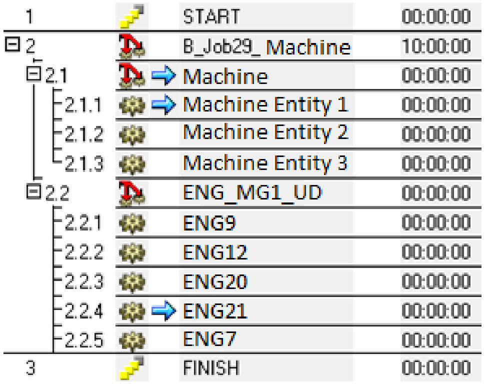
3 SIMULATION MODEL

To compare the results of the constraint programming approach a simulation model is generated. With this model, a PM plan is generated. Also a simple optimization is build up with the simulation model. The used simulation environment is the simcron MODELLER – an easy to use simulation system, which can be extended with additional user specific event handler. With this event handler extension, additional functionality could be included, which is not implemented within the main simulation system. The main objects in the simcron MODELLER are stations (machines and queues), routes and branches, jobs and shift schedules.

With these main objects, a first simulation model could be implemented, which depict the main functionality of the PM plan. For this, the following objects are used:

- The engineers are modelled as machines, where each engineer/machine has his own shift schedule linked. So the shift schedule functionality could be implemented directly within the simulation system
- To manage the qualifications of the engineers, branches for each machine group and PM type are generated. Within these branches the allowed engineers/machines are added.
- The entities of the cluster tools are also modelled by machines. Here no additional constraints like shift schedules are needed.
- The cluster tools themselves are build up as branches, which means, a cluster tool has a set of entities, which are implemented in the simulation model. If a cluster tool is maintained, all machines/entities within the branch have to be free and while the maintenance is done, all machines within this branch are used. This can be also done with the functionality of the simulation system by setting the parallelism in a branch to all. This means, all objects in the branch are needed, if a job is performed.
- All maintenances (PMs) are covered as jobs which have to be planned on the corresponding machines. So a PM needs either an entity of a cluster tool or the whole cluster tool. In the first case, only the corresponding machine of the entity is used and in the second case the created branch of the cluster tool is implemented within the route. Additionally, to perform a maintenance an engineer is needed. Here we also add the corresponding branch of engineers to the route of the job. Now to perform the job, the needed machine or entity of the machine has to be free and also an allowed engineer must be free and available. The job also gets a release date and a due date. The release date matches with the first starting point of the PM domain and the due date is the end of the PM domain.

In Figure 6, an exemplary route for a PM job is shown. Here, the job starts at a START queue and finish in the FINISH queue. The second step of the route is a branch with a parallelism of all, this means, all elements of the route have to be available. In this branch the cluster tool (also a branch with machines/entities) and the engineer branch is implemented.



1	START	00:00:00
⊟ 2	B_Job29_ Machine	10:00:00
⊟ 2.1	⇨ Machine	00:00:00
2.1.1	⇨ Machine Entity 1	00:00:00
2.1.2	Machine Entity 2	00:00:00
2.1.3	Machine Entity 3	00:00:00
⊟ 2.2	ENG_MG1_UD	00:00:00
2.2.1	ENG9	00:00:00
2.2.2	ENG12	00:00:00
2.2.3	ENG20	00:00:00
2.2.4	⇨ ENG21	00:00:00
2.2.5	ENG7	00:00:00
3	FINISH	00:00:00

Figure 6: Exemplary route within the simulation model for a DPM

This first simulation model covers the main parts of the maintenance planning and also uses only the standard functionality provided by the simulation system. But because of the shift schedules for the engineers it can happen, that a maintenance could not be fully performed within one shift. So, at this point the additional functionality of creating user-specific event handler within the simulation model is needed. This event handler first has to ensure that short PMs are only done within one shift and could not be performed in two shifts with two different engineers. But the more complex part of the event handler is to ensure that if a LPM or a DPM has to be scheduled, allowed engineers in the consecutive shifts are available. This means, if a maintenance is started, the maintenance could not be aborted or paused because of missing engineers. So this event handler checks if a maintenance should be started, if this maintenance could be completed in the same shift and if it could not be completed and is not a SPM, than this event handler checks the engineers in the consecutive shifts on availability. If the maintenance could be started, the event handler blocks the corresponding engineers.

This additional event handler also shows the problem of standard simulation systems: A simulation system could not cover all possible functionality which is needed in practice. In the practice there mostly exist special constraints, which have to be implemented user specific within such a simulation environment.

An additional problem of such an simulation system is the optimization. A simulation system does not optimize the result. It only calculates a feasible solution (if the simulation model is build up in the right way). If an optimization should be implemented, dispatching rules could be used and/or different simulation runs with different input parameters have to be performed.

In this approach, the jobs have priorities which depend on their type. In this case, DPMs have the highest priority. The SPMs have the lowest priority in the model and therefore the long PMs have a priority between SPMs and DPMs. The jobs are now sorted in the START queue depending on their priorities and are also tried to place on the machines in this way. Now the simulation model tries to process the jobs in the order of their priority. If a job could not be placed, the next job in the queue is tried.

To optimize the results gained by the simulation model, a simple optimization approach is implemented (c.f. Figure 7).

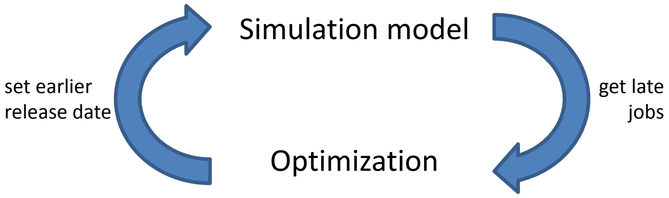


Figure 7: Simulation-based optimization

This optimization approach tries to optimize the result by optimizing the release dates of the jobs/PMs. For this in the first step all SPMs and LPMs get the ideal release date calculated in 2.3.2. This means the ideal start date is at ¾ of the domain. Now a simulation run is performed and the resulting PM schedule is used to find jobs/PMs which are too late. For these late jobs, the release date now is set earlier and a new simulation run is performed. If all jobs are finished in time or all late jobs have a release date which is equal with the original release date the optimization run is stopped.

With this optimization an ideal delayed PM plan (from the viewpoint of simulation system) is generated. The results of this plan are compared with the constraint programming approach in the next section.

4 RESULTS

For testing the CP approach realistic data are used. Thereby for this publication an artificially generated test instance is used which is roughly orientated on reality. This data consists the planned maintenance for about one week. Also a long PM exists and several DPMs have to be planned at the beginning. In summary, the test data set contains 27 SPMs, one LPM and 12 DPMs which have to be scheduled. Also about 30 engineers with individual shift schedules exists.

With this test instance results are calculated by the CP model with different objective functions and the simulation model using the described optimization approach.

In Figure 8 the usage of the engineers over the time is shown. Thereby the gray colored areas show engineers which are not used. The yellow areas show engineers which are used for SPMs, the green areas show the engineers used for LPMs and the orange areas show the engineers used for DPMs.

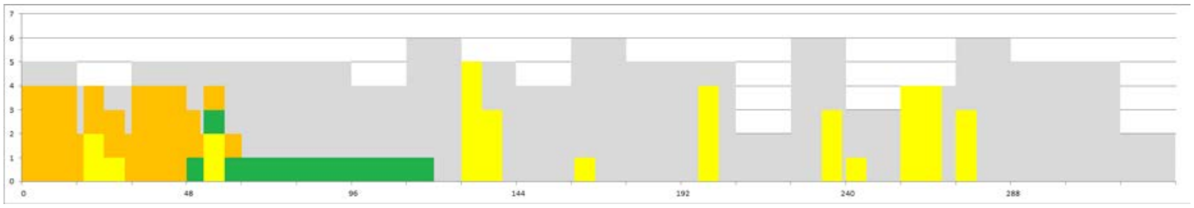


Figure 8: Capacitive engineer utilization for simulation approach

As you can see, the balancing of the engineers is not done very well by the simulation. This is also a disadvantage within simulation. The DPMs are scheduled at the beginning as desired. To balance the workload over all engineers, a very complex dispatching rule has to be implemented. In comparison to Figure 8 the engineer utilization shown in Figure 9 is an exemplary result of the constraint programming approach, which show a far better balancing for SPMs and LPMs.

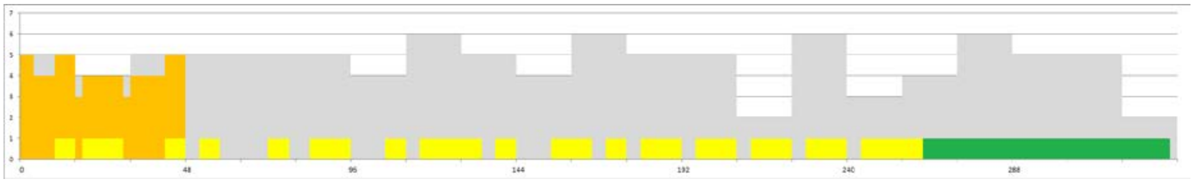


Figure 9: Capacitive engineer utilization for CP approach with combined objective

Several objectives were investigated for the CP approach. According to (19) the weights which are shown in Table 1 were used for the objective function. Here the weights for CP6 are chosen in the way that all objectives have the same impact on the objective function. So they are equally weighted.

Table 1: Weighting of optimization objectives for different scenarios

	SimPM	IdealDelay	ShiftPM	ReserveFit	SkillSum
CP1	0	0	0	0	1
CP2	0	0	0	1	0
CP3	0	0	1	0	0
CP4	0	1	0	0	0
CP5	1	0	0	0	0
CP6	100	-10	100	-1	1

In Figure 10 the results are shown in a normalized form for different objectives. The chart shows results for the CP approach with five single optimization objectives and one combined optimization objective. For comparing also results for simulation approrach are shown as reference values.

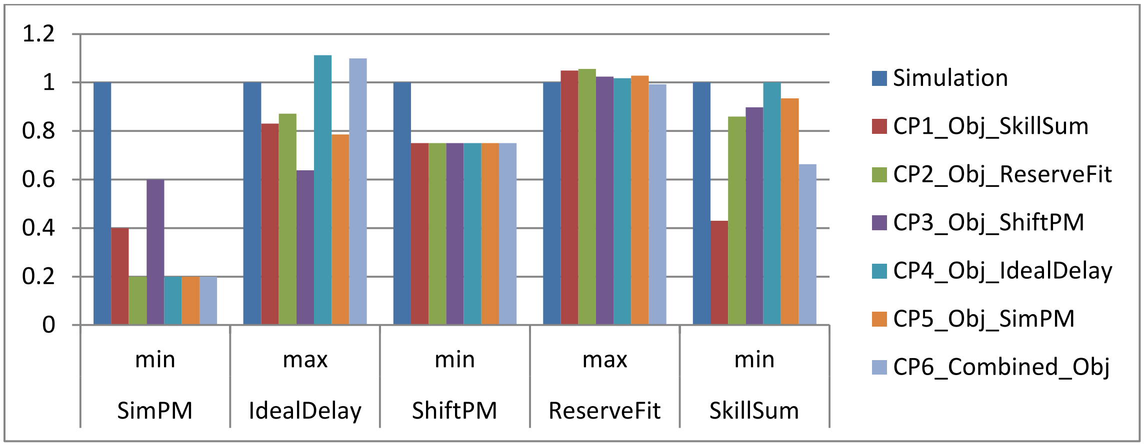


Figure 10: Results for simulation and CP approach

The results show, that the constraint programming based approach generates better results for almost all objectives. A comparison between the simulation result and the result gained by the combined CP objective (CP6) show, that simulation performs worse except for the ReserveFit objective. But overall the results of the different CP runs achieve mostly better results. However, the simulation model was not implemented to optimize all shown objectives.

5 CONCLUSION

In this paper a constraint programing based optimization approach for a preventive maintenance scheduling is presented. These maintenances have to be synchronized with engineers, which have to be available. Also different objective functions are implemented within the CP-approach. All these objectives could be optimized separately or in combination. The maintenances differs between short maintenances, long maintenances and unscheduled downs. Here, long maintenances and unscheduled downs could be or have to be scheduled through multiple shifts. To compare the results a simulation model was build up and an easy optimization was used with this optimization model. Thereby the simulation model has disadvantages for finding optimal strategies for the different objectives. So only the SkillSum and the IdealDelay was used to optimize.

The results show that the constraint programming approach reach better results for all defined objectives. This is also a result from the different viewpoints of the simulation and CP-approach. The simulation normally only knows what happens at a special time point and the scheduling CP have a look over the whole problem. On the other hand, the simulation is very fast where the constraint programming approach needs much more time to calculate results. Also the complexity of the CP-model increases rapidly if more shift overlapping maintenances have to be planned.

In conclusion, the constraint programming approach reached much better results than the simulation based model. Vice versa the constraint programming model needs much more time to calculate good results in comparison to the simulation model. To find optimal strategies for optimizing the simulation model much more manpower than usually in the constraint programming model is needed.

REFERENCES

Abdennadher, S., Schlenker, H. 1999. “Nurse scheduling using constraint logic programming.” In *Proceedings of the National Conference on Artificial Intelligence*, 838-843.

Alfares, H. K. 2004. “Survey, categorization, and comparison of recent tour scheduling literature.” *Annals of Operations Research*, 127(1-4), 145-175.

Ernst, A. T., Jiang, H., Krishnamoorthy, M., and D. Sier 2004. Staff scheduling and rostering: “A review of applications, methods and models.” *European journal of operational research*, 153(1), 3-27.

Lange, J., Weigert, G., Klemmt, A., and Doherr, P. 2013. “Scheduling Maintenance Tasks with Time-Dependent Synchronization Constraints by a CP Modeling Approach.” In *Proceedings of the 2013 Winter Simulation Conference*, edited by R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl, 3642–3653. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Laporte, G., Pesant, G. 2004. “A general multi-shift scheduling system.” *Journal of the Operational Research Society*, 55(11), 1208-1217.

Li, H., Womer, K. 2009. “A decomposition approach for shipboard manpower scheduling.” *Military Operations Research,* 14(3), 67-90.

Lilly, M. T., Emovon, I., Ogaji, S. O. T., and Probert, S. D. 2007. “Four-day service-staff work-week in order to complete maintenance operations more effectively in a Nigerian power-generating station.” *Applied energy*, 84(10), 1044-1055.

He, F., Qu, R. 2012. “A constraint programming based column generation approach to nurse rostering problems” *Computers & Operations Research*, 39(12), 3331-3343

Van den Bergh, J., Beliën, J., De Bruecker, P. ‚Demeulemeester, E., and L. De Boeck 2013. "Personnel scheduling: A literature review." *European Journal of Operational Research*, Elsevier, vol. 226(3), 367-385.

AUTHOR BIOGRAPHIES

JAN LANGE received his master's degree in Information Systems Technology at the Dresden University of Technology in 2008. He is now a Research Assistant at the Electronics Packaging Laboratory of the Dresden University of Technology. His email is Jan.Lange@tu-dresden.de.

DIRK DOLESCHAL studied mathematics at the Dresden University of Technology. He obtained his master degree in 2010 in the field of optimization. He has been a Research Assistant at the Electronics Packaging Laboratory since 2010. His email is Dirk.Doleschal@tu-dresden.de.

GERALD WEIGERT is an Assistant Professor at Electronics Packaging Laboratory of the Dresden University of Technology. Dr. Weigert works on the field of production control, simulation & optimization of manufacturing processes. His email is Gerald.Weigert@tu-dresden.de.

ANDREAS KLEMMT received his master’s degree in mathematics in 2005 and Ph.D. in 2011 at the Dresden University of Technology. He is employed as staff engineer in the operations research and engineering group of Infineon. His email is Andreas.Klemmt@infineon.com.