An Algorithm for Multidimensional Optimization and Robustness Evaluation within a Rescheduling Procedure

Leonardo Brignolo, Christian Fabig, and Roberto Revetria

***Abstract*— Project schedules supporting the production and maintenance of heavy special machinery, ships and aircrafts have to be updated, e.g. due to changed order contents and dates, resource breakdowns, further projects etc. In addition to the production target, an update should also aim to have as few deviations as possible to a current schedule (i.e. solution robustness), aiming to prevent revisions of further decisions (e.g. material procurement, subcontracting, worker attendance). This raises the need for a multi-objective evaluation of deviations and a proper rescheduling procedure.**

The aim of the paper is to design two algorithms for optimization and robustness evaluation, perceiving robustness as deviations between one current schedule and one possible reschedule created by the simulation-based optimization. In contrast to existing approaches, we propose to measure those deviations with respect to various dimensions. Two algorithms are designed, measuring timely deviations (e.g. deviation of milestones) and those of resources (e.g. short term change of shifts). To facilitate practical implications, a case study from aircraft maintenance is used. Results reveal that the multidimensional evaluation of deviations serves as a valuable extension of optimization tools in order to gather practical preferences and limitations within a heuristic rescheduling procedure.

***Index Terms*— Rescheduling, Solution Robustness, Aircraft Maintenance.**

I. Introduction

Multi-project manufacturing typically involves a few resource groups with some hundreds of individual resources that are shared by all projects. In those complex environments, there are various reasons that alter the state of the system during project realization. Thus deviations from a predetermined schedule (called "baseline") will occur, making it infeasible. Rescheduling updates an existing schedule in response to deviations. Especially in multi-project manufacturing, changes of the baseline will incur further planning adjustments. As a consequence, deviations will be very costly due to, e.g., penalty costs for missed milestones, changed material orders and modified subcontracts [4]. One method aiming to reduce deviations is (partial) rescheduling [10, 14] and may be done using one of the reactive scheduling procedures [8, 12]. Existing reactive procedures optimize one production target, e.g. production cost minimization. Additionally, the "instability costs" (IC) is frequently considered for measuring "solution robustness", i.e. stability or deviations [5, 7]. Rescheduling procedures may also include IC as a part of the cost function applied [4].

IC is defined as the weighted absolute deviation of activity starting times. Each activity is associated with a weight, i.e. marginal costs of deviating it´s starting time by one time unit. However, from a practical point of view, marginal costs of deviating activities will hardly, if ever, be available. Mostly those costs are not related to expenses but to administrative efforts that are inaccurately covered as overhead expenses. Moreover, when rescheduling, schedules can at least have deviations in two dimensions: time and resource availability. IC only account for timely deviations.

Hence, there is a need for a multi-objective rescheduling procedure, taking into account the production target as well as an evaluation of multidimensional deviations. In this paper, we will design algorithms and measurements with the characteristics mentioned in order to support practical applications where no accurate cost of incurred efforts from deviations are available. Moreover, we aim at combining those measurements into one single robustness measure that shall be used within a heuristic search method of a simulation-based optimization tool.

The paper refers to a possible application for aircraft maintenance in Section II with the goal of giving a clearer perspective of the issues related to this topic. Next, an overview of the robustness algorithm and the related research on which they are based is given in Section III and IV. A comparison of possible updated schedules is carried out (Section V), resulting in the extraction of the most robust one.

Manuscript received March 13, 2018, revised April 04, 2018.

Leonardo Brignolo is with the department of Mechanical, Energy, Management and Logistics Engineering (DIME), 16145, ITALY (e-mail: brignolo.leonardo@gmail.com).

Christian Fabig is with the Institute of Material Handling and Industrial Engineering at the Technische Universität Dresden, 01062, GERMANY (e-mail: christian.fabig@tu-dresden.de).

Roberto Revetria is with the department of Mechanical, Energy, Management and Logistics Engineering (DIME), 16145, ITALY, (e-mail: roberto.revetria@unige.it).

II. Planning and Optimization of Aircraft Maintenance

Proper planning is of key importance in aircraft maintenance. Multiple examples and research papers show that cost savings can be gained through a fitted, robust capacity planning and scheduling [13]. Low and predictable costs as well as guaranteed turnaround times are the main production targets in aircraft maintenance [9]. Therefore, planning tools have to be constantly improved.

So called “heavy maintenance checks” (HMC) represent checks that are complex due to high intensity of workload and scope. HMC projects are labour-intensive and often subcontracted to a third-party service provider. One of them is Elbe Flugzeugwerke GmbH (EFW), located in Dresden, Germany. As common practice among those service providers, EFW handles each customer order for aircraft maintenance as an individual project while multiple projects are realized contemporaneously (i.e. multi-project manufacturing). Planning and scheduling are subject to considerable uncertainties, originating from unknown conditions of the aircraft components. Damages and areas subject to wear and tear will appear during the inspection work. Therefore, the projects´ workload always consist of work orders that are known prior to the project (i.e. “scheduled” or “routine” work) as well as unforeseen work orders needed to repair or replace components (i.e. “non-routine” work; see [13]).

A. Heavy Maintenance Processes

Various manuals (e.g. Aircraft Maintenance Manual, AMM) are available from aircraft manufacturers, defining work content and workflow, required personnel qualifications and equipment for routine HMC operations. AMM and other manuals are structured in a component-oriented fashion, i.e. manuals do not provide an overall project network but procedures for maintaining one specific component. Maintenance organizations are using those manuals to create their own work orders (called “job cards”), including further information (e.g. estimated working hours per activity, resource amount, equipment). Routine works are already known and planned prior to project start. As shown in Figure 1, during the inspection phase further unplanned works will occur. For medium-term production planning, non-routine efforts must be estimated and categorized based on experience using, e.g., aircraft age, type and operator:

• *DR non-routine*: When components to be repaired or replaced are detected, a Discrepancy Reports (DR) is created. The complaint treatment lasts up to one week, since work often has to be coordinated with the aircraft manufacturer.

• *ASR non-routine*: Since there is a particularly high degree of customer integration, additional service requests (ASR) can still be ordered during the project realization.

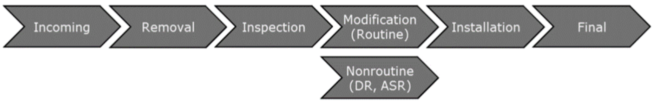


Figure 1: Phases of aircraft heavy maintenance

A special case of HMC is the Passenger-to-Freighter conversion (PtoF) where a former passenger aircraft is modified into a freighter aircraft. Typical HMC checks are performed in parallel to the standardized PtoF conversion process – which is a real benefit for aircraft operators to minimize overall maintenance periods. However, from a service provider’s point of view, the production complexity increases. Due to an extended modification phase, the project duration of a PtoF is approximately 16 weeks whereas a stand-alone HMC project is carried out in 4 to 8 weeks. The unknown non-routine workload accounts for more than a half of the man-hours necessary for a stand-alone HMC. Due to a higher proportion of standardized processes, non-routine is only one third in a PtoF. The proportions of routine and non-routine are visualized in Figure 2.

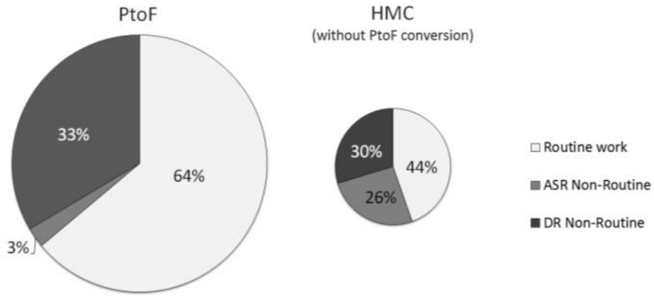


Figure 2: Ratio between routine and non-routine work

HMC phases are determined for technical and/or safety reasons. E.g., incoming and final tests require on-board electricity whereas component removal, modification and installation prohibit electrical power. However, planning is supported by detailed project networks consisting of approximately 300 work packages for stand-alone HMC and up to 1.400 work packages for a PtoF project. Work packages are created concerning aircraft conditions (e.g. electrical power on/off, stress-free aircraft fuselage etc.) that have to be prepared before executing a certain group of activities. Activities within a work order are therefore usually not directly successively executable, but are each additionally dependent on certain assembly states.

Workers are highly specialized. Skills (also called “qualifications”) needed for executing specific activities are defined by maintenance manuals and approved by aviation authorities. E.g., for airframe repairs a structure repair certificate is mandatory. In case structure repair has to be executed within the wing structure, workers additionally have to hold a fuel tank permission. Another restriction is working space within and on the aircraft. For activities that are mandatory to be executed in a defined space, a maximum number of workers must be considered. For example, within the cockpit, a maximum of three fitters can work in parallel.

B. Theoretical classification and problem definition

The scheduling problem of multi-project manufacturing is classified as RCMPSP, a resource-constrained multi-project scheduling problem [1]. Concerning the optimization of aircraft maintenance, especially the following real-world extensions have to be supported:

- *Skilled resources:* Personnel resources are shared between all projects. Whether they are authorized to execute a specific activity is restricted by the skills preserved and the skills essential for an activity. Also, the processing time is dependent on the skills as higher skilled resources are generally speaking more efficient. Resources are therefore linked to a “qualification matrix” including a level of efficiency [3]. As a result, more realistic models can be simulated. Further resources such as working spaces, hangars, machines and tools can be modelled in the same kind.

- *Varying resource availabilities*: Available resource amounts are varying per time period due to, e.g., predefined shift models and illness.
- *Subcontracted resources:* Subcontracting is also a proper option in practice. E.g. specific testing procedures like eddy currents and X-ray could be done by external resources. Subcontracting will either be negotiated for a complete work package (i.e. all associated activities) or on a daily basis. Therefore, the simulation model also specifies work packages that can be subcontracted and groups of external resources that can be used on a daily basis. An optimizer has to support the creation of multiple production scenarios where temporal subcontracting is a planning option.

Concerning the optimization of the described RCMPSP problem within a rescheduling procedure, a simulation-based optimization tool has to support the following:

- *Model preparation*: Completed activities have to be excluded from the models and the simulation has to start from a specific time point. For practical needs, this time point should be set to the near future, e.g. the upcoming shift, leaving a "frozen zone" for the creation of a reschedule and propagation onto shop floor.
- *Production scenarios*: Several valid planning scenarios have to be created and evaluated based on key performance indicators (KPI).
- *Optimization procedures*: Practical scheduling problems can´t be solved to optimality in reasonable time. In order to create schedules for practical problem sizes, simulation-based optimization is a promising approach. It combines heuristic search methods with a Discrete-Event-Simulation (DES) to determine near-optimal schedules with respect to a production target [11]. Concerning rescheduling, there is a trade-off between minimized deviations (i.e. stability, solution robustness) and a scheduling objective (e.g. minimized production costs or makespans, see [4]). One algorithm that has been shown to perform a rapid search for pareto-optimal solutions in two conflictive targets is the genetic algorithm NSGA-II [2].

The mentioned real-world extensions and optimization features have been combined in a simulation-based optimization tool called "OptimSim" [11]. Its internal plugin-based Java model rests on the SysML modelling paradigms, a standardized description language in the field of systems engineering. Currently, OptimSim is being extended to fully support robust capacity planning and scheduling of aircraft maintenance. One extension is the deviation evaluation algorithm that we will show in detail.

III. Robustness Evaluation

A. Definition of Robustness

Before proceeding into the description of the evaluation process, a few concepts must be defined; in fact the need of this procedure comes from the impossibility to eliminate completely disruption risks while keeping projects cost effective. This underlines the trade-off problem between cost and time, which often deviates from the baseline plan. However no comprehensive guidance about robustness measures has been defined so far, with the consequence of having managers that take intuitive decisions. To overcome this issue the evaluation must take into account not only several different parameters directly related to the updated schedule itself but also to the previous one, through measuring deviation. Robustness can be in fact defined as *"the persistence of certain specified system features despite the presence of perturbations in the system's environment"* [6]. The modern approaches that can be applied to increase project's robustness are *Reactive Scheduling, Proactive Scheduling and Proactive/Reactive Scheduling* [10].

Reactive scheduling is based on a rescheduling process whenever a disruption occurs, while *Proactive Scheduling* is based on the generation of a robust scheduling that will not be changed during projects execution time; in this paper the combination of these two approaches is used, increasing robustness through a periodical comparison of the deviation between the actual plan and the new ones generated by the algorithm.

Whenever there is a deviation, several approaches can be applied, such as *right-shifting, partial rescheduling* and *full rescheduling* [14].

Right-shifting activities imply that no rescheduling process is performed and all the successors of the delayed activity are postponed. This approach is successful only for small disruptions and if the slack times have been dimensioned to contain that delay. *Full rescheduling*'s success depends on the degree of uncertainties which characterize the projects and it also has a high computational cost. *Partial rescheduling* is the approach chosen by the authors because it was judged to be more applicable to reality, therefore a time window is defined, where the new activities of the projects will be scheduled.

B. Robustness Measurements

In this paragraph the authors propose several measurements that will be applied to each schedule in order to have a new schedule judged to be the most robust one. The evaluation is divided in two main groups: *Resource* and *Deadline Evaluation*.

In the first group, there have been designed the following measurements:

- *Number of Active Resources (NAR):* resources that are currently assigned to an activity are counted for the whole duration of the time window. The deviation between the baseline and the updated schedule is computed as the absolute value; the schedule with the lower deviation is judged by the algorithm as the most robust since it minimizes the change in shifts. This measurement can be modified in time by applying a weighting function chosen by the user; through this function it is possible to highlight more deviations in certain moments, depending on what the user prefers.
- *Available Resources:* the distribution of assigned resources might vary in time as well as the number of resources that are not currently applied but can be assigned whenever a disruption occurs. A schedule with a higher availability of resources is more stable and more reactive to disturbances;

The second group analyses and quantifies delays in terms of time and cost:

- *Tardiness Cost:* each project is divided in three milestones, each one has penalties for every hour of delay. The costs may be set by the user at the beginning of the simulation; the resulting costs are compared between the two updated schedules, evaluating as more robust the one with lower penalties.
- *Start deviation:* the algorithm sums all the delays of the activities at the start, underlining the schedule with lower delays than the other.
- *End deviation:* this measurement counts all the delays in ending the activities.
- *Shrinking factor:* even if a schedule starts with a delay, it might recover from it or, vice versa, it might increase it. This measurement tracks the status of the delay, extracting the schedule with lower outcomes.

All the results can be prioritized depending on the manager preferences (see Section IV, D); the outcomes of each measurement are merged into one method; its function is also to normalize these outcomes so that they can be comparable. Then, user preferences are applied and measurements that are meant to be minimized are converted in sign; the last step is the extraction of the evaluation score through the weighted mean formula. The schedule with the highest score is extracted as the most robust one (see Appendix).

C. User Preferences

At the beginning of the simulation the user is asked whether default values are acceptable or they have to be modified; in this last case, the following values will be asked:

- *Priority Preferences:* for each measurement a number between 0 and 100 can be attributed in order to give it the preferred priority;
- *Available Resources*: it consists of 4 values corresponding to the maximum number of available resources in each week of the month analyzed;
- *Weighting function:* this will be applied to the NAR measurement to highlight it more in certain moments, such as using an increasing function to give more importance to deviations at the end of the time window;
- *Tardiness cost:* three values must be inserted, corresponding to the penalties in € for every hour of delay of the milestones;
- *Type of Analysis:* the parallel projects of the same environment can be analyzed separately of globally;
- *Exportation process:* the data analyzed in the algorithm can be exported to *.xls* files. This choice is based on the size of the time window of the projects, since it slightly increase the duration of each computation;
- *Printer:* values can also be printed in the log screen of the program, making this process faster compared to exportation, but without storing data externally.

D. Results Normalization

The values coming from each measurement differ widely, due to the different nature of each one. As a consequence several simulations with different schedules were performed in order to establish the values between which a measurement can range, so that the final evaluation can be balanced.

IV. Example Of Schedule Comparison

In this chapter, three examples will be carried out, each one resulting in different outcomes depending on user preferences. In Table 1 are shown the preference values that will be applied:

Table 1: user priorities applied

User Measurement's Priority [%]	Examples		
	1	2	3
NAR	75%	7%	10%
Available Resources	10%	75%	7%
Tardiness Cost	7%	10%	75%
Start delay	2%	3%	3%
End delay	3%	3%	3%
Shrinking Factor	3%	2%	2%

As we can observe from Table 1, on each example, one measurement is prioritized more over the others. In the following paragraphs each example is discussed.

A. Example 1: NAR prioritization

This example will be carried out by discussing three different sub-cases which differ one from the other depending on the weighting function that will be applied (see Table 2 for the output results of this example).

1) No Weighting Function

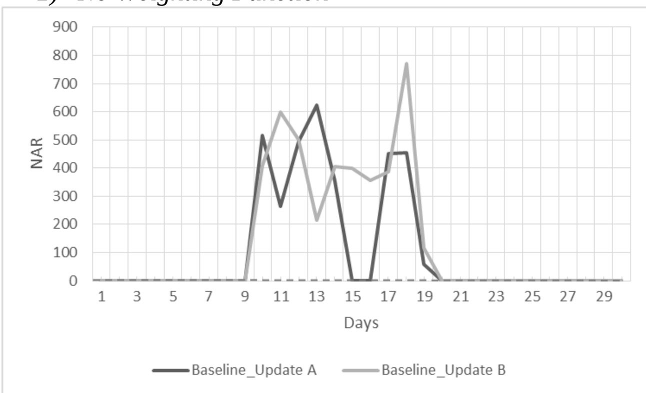


Figure 3: NAR deviation comparison between two Updated schedules with no weighting function applied

As we can observe from Figure 3 above, there is no difference between the two curves before day 9, which means that the updated schedule were generated after that day, called, from now on, *scheduling point.* After day 20 the curves assume the constant null value, meaning that the project's time window has concluded. From Figure 3 it can be observed that the Update B has higher values of *NAR* for most of the time, which means that Update A is more robust.

2) Linearly increasing function applied

It might be reasonable to weight more the deviation since the project at its time window's conclusion should have the smallest deviation possible; in this case the deviation in the right side of the graph have been prioritized, still resulting with Update A as the most robust (see Table 2).

3) Linearly decreasing function applied

Since uncertainties are more probable to increase as the time from the scheduling point passes, it might be best to

ISBN: 978-988-14048-9-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

prioritize more the first deviations that will occur in time. Also in this case Update A is judged to be the most robust (Table 2).

Table 2: Output evaluation values

	No Weight		Increasing		Decreasing	
	A	**B**	**A**	**B**	**A**	**B**
N.A.R.	-48,2	-62,2	-6,0	-8,1	-8,2	-10,2
Av. Res.	8,0	8,4	8,0	8,4	8,0	8,4
Td. Cost	-6,5	-6,0	-6,5	-6,0	-6,5	-6,0
Start Del.	-0,6	-0,6	-0,6	-0,6	-0,6	-0,6
End Del.	-0,6	-0,6	-0,6	-0,6	-0,6	-0,6
Shr. Fact.	-0,4	-0,6	0,4	-0,6	-0,4	-0,6
Score	-48,5	-61,9	-6,3	-7,8	-8,6	-9,8

B. *Available Resources Prioritization*

Resources are the base thanks to which a schedule can be executed; whenever the number of resources applied to the schedule is close to the maximum number available, the risk of being subjected to disruptions arises. Since the number of available resources in a project might vary from a week to another, the behaviour of NAR plots strongly influences the stability of the schedule. Before proceeding further, it has to be noted that from now on the simulations will be carried out without weighting the NAR function. In this case the interest goes to maximising the gap between the available resources and the ones that are active in one activity at that time, which is an index of stability. The results can be observed in Figure 4 where it can be noticed that Update A has its highest value of active resources and has also a smaller gap with the available resources curve, meaning that almost all the available resources are applied to activities in that moment.

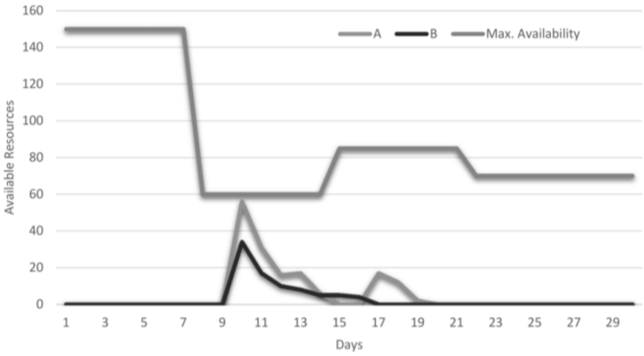


Figure 4: Resource Availability during projects execution

This implies that Update A is more likely to be subjected to disruptions, since in case of a delay, there might not be enough extra resources to react to it, therefore Update B is judged to be the more robust schedule.

A. *Tardiness Cost Prioritization*

Whenever there is a delay in a milestone, penalties are applied, affecting the final profit of the company which is executing the projects. In this example the minimization of these additional costs is pursued; as we can observe from Figure 5, which shows the hours of delay accumulated by each schedule, Update B has lower delays on each of the three milestones.

Table 3: Output evaluation values prioritizing Available Resources

Measurement	Update A	Update B
N.A.R.	-4,4	-5,8
Av. Res.	60,0	63,2
Td. Cost	-9,3	-8,6
Start Del.	-0,6	-0,6
End Del.	-0,6	-0,6
Shr. Fact.	-0,4	-0,6
Score	-43,8	-44,0

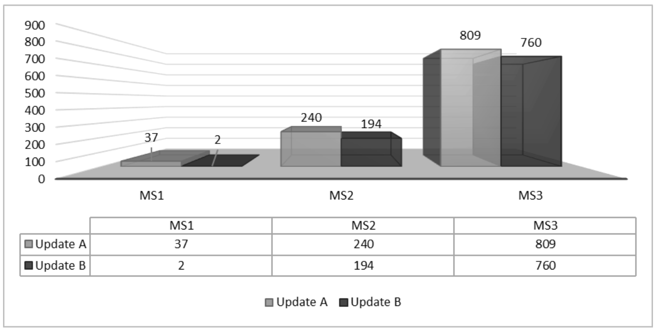


Figure 5: Tardiness Cost between Milestones

This behaviour might vary from case to case, since milestone delays are not related one to the other; also the cost per hour of each milestone (MS) is different, which is relevant in terms of final cost. In fact a schedule might have higher delays on a milestone with lower penalties or vice versa, leading to the need of computing the total cost for each milestone. However, since in our case, one schedule has always a lower delay than the other (see Figure 5), the result of extracting Update B as the most robust one is predictable (see Table 4).

Table 4: Output evaluation values prioritizing Tardiness Cost

Measurement	Update A	Update B
N.A.R.	-6,4	-8.3
Av. Res.	5,6	5,9
Td. Cost	-70,0	-64,9
Start Del.	-0,6	-0,6
End Del.	-0,6	-0,6
Shr. Fact.	-0,4	-0,6
Score	-72,6	-69,3

V. RESULTS

To sum up the outcomes of the previous examples, the following table is created, which provides an overall analysis of the two Updates:

Table 5: Final results of the previous examples

Example	1	2	3
Priority	NAR	Av. Resources	Tardiness Cost
Update with higher robustness	A	B	B

The parameters that have been chosen generated several different evaluations which, as it can be seen in Table 5,

ISBN: 978-988-14048-9-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

shows that Update A is the more robust schedule in terms of reductions in shift changes (*NAR*) while Update B is judged to be more robust in terms of availability and tardiness cost. As a consequence, the extracted Update will be set as baseline from the scheduling point on.

VI. Conclusion

The algorithm proposed in this paper has been used by setting parameters judged by the authors to be key factors to the projects execution and to increase their robustness; nevertheless only a few of the possible configurations have been shown that the user might apply, aiming to explain the principles upon which this project is based. Since no comprehensive guide on this topic has been defined yet, in this paper, some possible aspects have been proposed, based on literature, which are useful to evaluate possible updates that would not be feasible to analyse without automated systems, due to the huge amount of data of each project. Further upgrades of these algorithms will be carried out in the future with the goal to design tools that can be adapted to any industrial Multi-Project Environment, providing an agile and efficient tool to accomplish more complex goals in less time and with less expenses.

References

[1] BROWNING T. R. & YASSINE A. A. (2010): Resource-constrained multi-project scheduling: Priority rule performance revisited. International Journal of Production Economics 126(2):212–228.

[2] DEB K. & TIWARI S. (2008): Omni-optimizer: A generic evolutionary algorithm for single and multi-objective optimization. European Journal of Operational Research 185(3):1062–1087.

[3] FABIG C., KERN A. & VÖLKER M. (2016): Herausforderungen einer robusten Ressourcenplanung der Flugzeugumrüstung und -wartung. In: Schmidt T, Schmauder M, Rank S (eds) Tagungsband zum Forum Technische Logistik und Arbeitssysteme 2016. Selbstverlag der Technischen Universität Dresden, pp 55–68.

[4] HERROELEN W. (2014): A Risk Integrated Methodology for Project Planning Under Uncertainty. In: Pulat PS, Sarin SC, Uzsoy R (eds) Essays in Production, Project Planning and Scheduling, vol 200. Springer US, Boston, MA, pp 203–217.

[5] HU X., DEMEULEMEESTER E., CUI N., WANG J. & TIAN W. (2017): Improved critical chain buffer management framework considering resource costs and schedule stability. Flex Serv Manuf J 29(2):159–183.

[6] J. M. CARLSON AND J. DOYLE. Highly Optimized Tolerance: Robustness and Design in Complex Systems. Physical Review Etters, 84:2529-2532, 2000.

[7] LAMBRECHTS O., DEMEULEMEESTER E. & HERROELEN W. (2008): Proactive and reactive strategies for resource-constrained project scheduling with uncertain resource availabilities. Journal of Scheduling 11(2):121–136.

[8] LV S. & QIAO L. (2013): Process planning and scheduling integration with optimal rescheduling strategies. International Journal of Computer Integrated Manufacturing 27(7):638–655.

[9] REMÉNYI C. & STAUDACHER S. (2014): Systematic simulation based approach for the identification and implementation of a scheduling rule in the aircraft engine maintenance. International Journal of Production Economics 147:94–107.

[10] SABUNCUOGLU I. & GOREN S. (2009): Hedging production schedules against uncertainty in manufacturing environment with a review of robustness and stability research. International Journal of Computer Integrated Manufacturing 22(2):138–157.

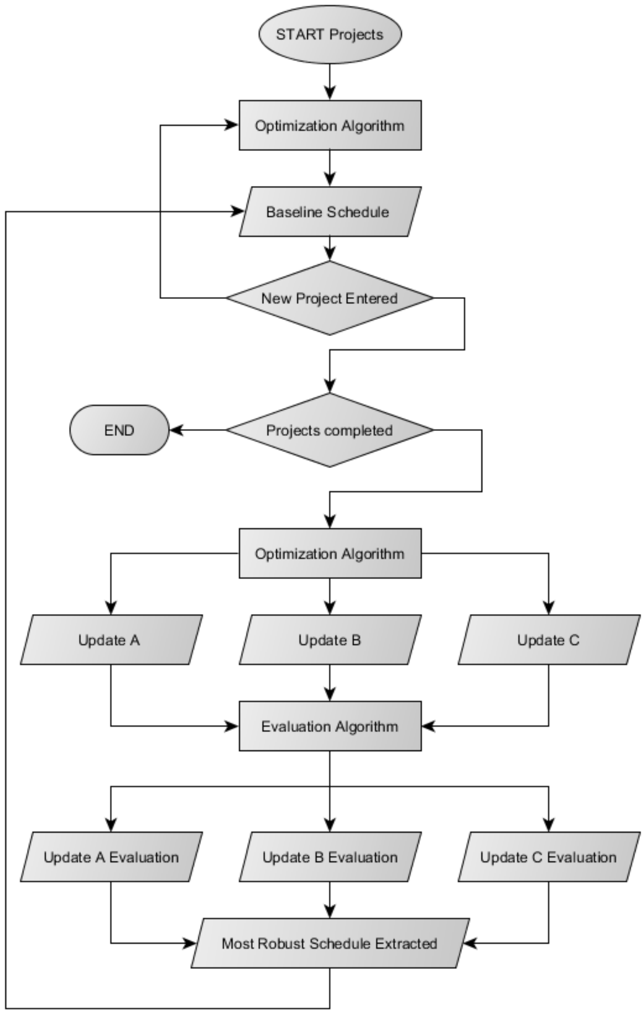
[11] SCHÖNHERR O., WINTER E., FABIG C. & VÖLKER M. (2015): Ein für die Wissenschaft freies Werkzeug zur Modellierung, Simulation und Optimierung. In: Rabe M, Clausen U (eds) Simulation in Production and Logistics 2015. Fraunhofer, Stuttgart, pp 101–110.

[12] VAN DE VONDER S., DEMEULEMEESTER E. & HERROELEN W. (2007): A classification of predictive-reactive project scheduling procedures. J Sched 10(3):195–207.

[13] VAN DEN BERGH J., BELIËN J., DE BRUECKER P. & PEETERS J. (2013): Aircraft maintenance operations: state of the art. Research Paper, Katholieke Universiteit Leuven.

[14] VIEIRA G. E., HERRMANN J. W. & LIN E. (2003): Rescheduling Manufacturing Systems: A Framework of Strategies, Policies, and Methods, Journal of Scheduling.

Appendix



The schedule with the highest score

ISBN: 978-988-14048-9-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)