Proceedings of the 2021 Winter Simulation Conference
S. Kim, B. Feng, K. Smith, S. Masoud, Z. Zheng, C. Szabo, and M. Loper, eds.

INITIAL ASSESSMENT OF THE INFLUENCE OF ROBUSTNESS ON THE WEIGHTED TARDINESS FOR A SCHEDULING PROBLEM WITH HIGH DEMAND VOLATILITY BASED ON A SIMULATION MODEL

Žiga Letonja
Nikolaus Furian
Johannes Pan
Siegfried Vössner

Institute of Engineering
and Business Informatics
Graz University of Technology
Kopernikusgasse 24
Graz, 8010, AUSTRIA

Melanie Reuter-Oppermann

Information Systems – Software & Digital
Business Group
Technical University of Darmstadt
Hochschulstraße 1
Darmstadt, 64289, GERMANY

ABSTRACT

Nowadays, customer behavior changes rapidly, resulting in high demand volatility throughout immensely globalized supply chains, especially from key-account customers. Hence, requiring not only excellent products produced in the shortest time but also abiding to defined due dates. In this paper, we propose a multi-objective large neighborhood search (MO-LNS) algorithm with a frozen period focusing solely on minimizing the total weighted tardiness of a two-stage production with random rush orders. The total weighted tardiness is set as the primary objective function and the robustness as the secondary one. The algorithm is implemented in a discrete event simulation (DES) model to observe the effects of varying objective function weights at different levels of order dynamism. The results suggest that there is not a single dominant weight pair, but rather a range, which dominates others. Thus, using a pair from the dominant range increases the likelihood of reducing the tardiness over a period.

1 INTRODUCTION

Coping with highly volatile demand, especially from key-account customers, has become an increasing challenge in many industries, in particular for make-to-order (MTO) production. At the same time, these challenges, which are influenced greatly by the globalization of supply chains, offer many opportunities for those, who are able to take advantage of simulations and analytics. This case study deals with a rescheduling problem at a label printing company, with the aim of minimizing tardiness, while accommodating both rush and regular orders as well as considering their relative importance, i.e., job weights.

The industrial partner targeted in this work produces labels with a technology called flexography, which is a continuous rotary printing method, using rubber or photo-polymer relief image plates and UV lights, which promote a faster ink curing process (FFTA 1999). This technology's main advantages are its speed and consistency of quality, making it suitable for medium to high-volume productions. In contrast, one of its major drawbacks is its extensive setup process, which makes it less favorable for small production lots because the setup can potentially take longer than the manufacturing. At the same time, the setup time of consecutive jobs can be reduced notably if they utilize the same inks, varnishes, or cutting die. Also, flexography requires a post processing production process, in which the printed labels are cut into ribbons and re-rolled into smaller rolls. The setup process of consecutive jobs on the second stage can be reduced also, if smaller rolls of the same diameter are used. This results in a two-stage manufacturing process.

978-1-6654-3311-2/21/$31.00 ©2021 IEEE

A plethora of environmental factors impact consumer behavior, which consequently results in high demand volatility of labels, which are necessary to market and sell products in stores, leading to a range of order lead times. The majority of customers place their orders in advance, enabling the company to procure materials and tools as well as plan the production accordingly - to meet the required due dates. At the same time, some customers regularly place rush orders, i.e., require products with a lead time of only a few days or less. This requires frequent rescheduling upon the arrival of such orders.

For a rescheduling strategy to be successful it must consider a range of constraints to yield not only feasible production plans but efficient ones. First of all, the updated plans must aim to prevent tardiness, i.e., if a job would be completed after its due date. In case this is not possible the relative importance of all current orders and customers needs to be considered and the "least violating" solution is chosen. Moreover, the reduced setup effort on a machine for consecutive jobs, which are similar, must be considered. Therefore, scheduling exclusively by due dates might not be the most efficient method. Another important constraint is maintaining a level of schedule stability while rescheduling to ensure smooth operations throughout the company - shop floor, sales office, and warehouse. Lastly, the planning personnel must expect that the next rush order may occur any moment, requiring a re-optimization of the current schedule.

Summing up, this raises the question of how to create schedules, which consistently keep the total weighted tardiness at a minimum. In particular, determining the right balance between optimizing this objective for the present as well as planning for the future, i.e., exploiting the flexibility of schedules to deal with random rush orders at later periods, while keeping the schedules as stable as necessary. From the practitioner's point of view this translates to the question of which objective weight pairing, i.e., which solution on the Pareto-Frontier, should be chosen whenever (re-)scheduling, given a specific degree of order dynamism. Hence, this paper attempts to answer the question of whether dominant objective weight pairings for order situations with distinct degrees of dynamism can be identified.

A recent review of the approaches used for scheduling in dynamic job shops was presented by Mohan et al. (2019), whereas Xie et al. (2019) examined flexible job shop problems. Also, González-Neira et al. (2017) reviewed comprehensively the literature on flow shop scheduling problems under uncertainties. What is more, the research dealing with lateness or tardiness as objectives in job shops is vastly outnumbered by the works focusing on makespan optimization, according to Chaudhry and Khan (2016). At the same time, the majority of the analyzed literature which is associated with industry include some form of tardiness, lateness, or earliness in the objective function, while only a part of the observed purely academic research considers these objectives (Chaudhry and Khan 2016). These findings further emphasize the notion that, from a practitioner's point of view, meeting the customer's expected due dates precisely is of the utmost importance, often more than theoretical efficiency.

Moreover, a large part of the scheduling literature, which targets the minimization of the total weighted tardiness, used a heuristic optimization approach since Du and Leung (1990) have proven that optimizing this objective on even a single machine is NP-hard in the ordinary sense. In general, most researchers have used the total weighted tardiness as the sole objective function in their heuristic approaches. E.g., Komaki et al. (2014) and Nishi et al. (2010) have done so for a flow shop environment. Whereas, Kirlik and Oguz (2012) and Lee et al. (1997), amongst others, have done so in the context of single machine experiments. At the same time, a few works propose using methods in addition to an objective function in an attempt to improve the total weighted tardiness. One method is the insertion of idle time blocks, where their duration and position in the schedule are based on a distribution of historic events, i.e., disturbances (Mehta and Uzsoy 1998). Another experiment was conducted by Jensen (2001), who attempted to improve the robustness and flexibility of a tardiness scheduling problem with breakdowns in two ways: 1) through reformulation of the neighborhood-based robustness measure technique from Jensen (2003) and 2) by minimizing the lateness, i.e., increasing slack, instead of focusing on the tardiness.

In this paper, we propose a multi-objective meta-heuristic scheduling algorithm to minimize the overall total weighted tardiness. It is based on two objective functions, where the primary objective is the total weighted tardiness of the current job set and the secondary one is accounting for robustness, as defined by

Pinedo (2009), which implicitly attempts to prevent tardiness in the future. Several weighted combinations of these two objectives are evaluated in this study. Additionally, the required level of schedule stability is achieved by implementing a variation of the frozen period principle.

To evaluate the effectiveness of the scheduling algorithm in a real-world setting, a discrete event simulation (DES) model of the industrial partner's shop floor was created, which replicates production processes. It allows to analyze how the chosen solution approach deals with different order situations and varying objective function weights. Notably, it enables to determine the right balance between flexibility, i.e., robustness, to accommodate high-weighted rush orders and schedule quality with respect to overall tardiness over a period, which stretches over multiple planning horizons.

The rest of this paper is structured as follows. Section 2 describes in detail the shop floor, the main scheduling challenges and their evaluation, and constraints. Section 3 outlines the basic principles of the selected algorithm, elaborates on how the DES model is built, and describes in detail the experiments and the sampling of the data. In Section 4 the results of the experiments are presented and interpreted. Lastly, Section 5 summarizes the paper and outlines promising research directions.

2 PROBLEM DESCRIPTION

2.1 Shop Floor Layout and Use-Case Specific Constraints

Currently, the industrial partner's shop floor consists of 11 machines, three parallel ones on the first stage, i.e., printing and cutting, and eight parallel ones on the second, i.e., ribboning and rerolling. The shop floor layout, including the flow of materials, consumables, and tools for an illustrative product are depicted in Figure 1. Furthermore, such a two-stage production layout with multiple machines on each stage is considered to be an industry standard for high-volume label printing.

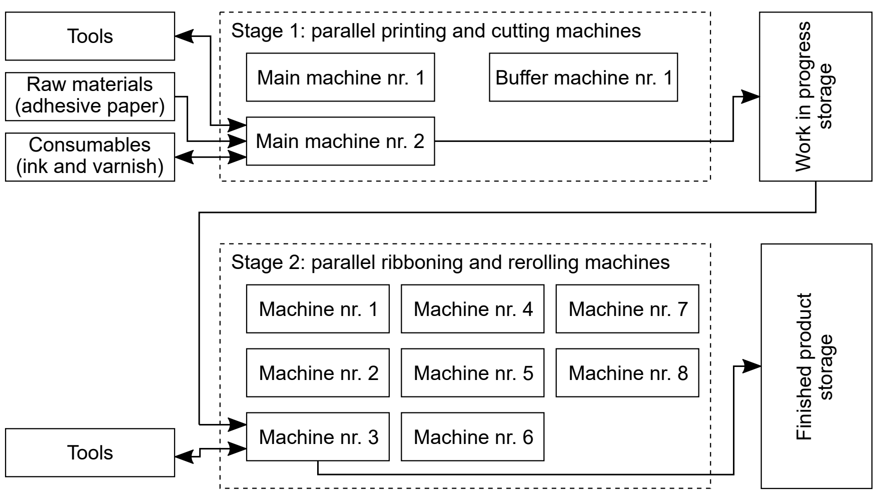


Figure 1: Layout of the industrial partner's shop floor and the flow of materials, consumables, and tools for an illustrative product.

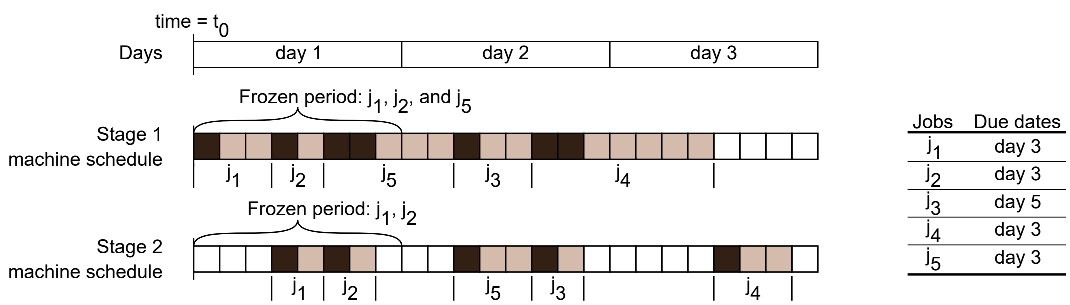
This specific production process is subject to the following constraints. At any time, a machine can produce only one product, which applies to all machines on both stages. All products must be processed on both stages; at the same time, a product must be processed on the first stage entirely before it can start processing on the second. Also, the setup times are sequence dependent.

Additionally, a variation of the frozen period principle is considered when scheduling. It is used to bring stability to the schedule and prevents confusion of the personnel. The frozen period is set for the duration of an entire workday from the time a new schedule is being created or an existing one is being updated. This way only the products, which are not affected by the frozen period can be rescheduled together with new arriving non-rush orders. However, the frozen period could be considered semi-frozen because rush orders are inserted into the frozen part of the schedule via basic greedy insertion to utilize the current schedule's flexibility. The impact of this principle is shown in detail in Figure 2.

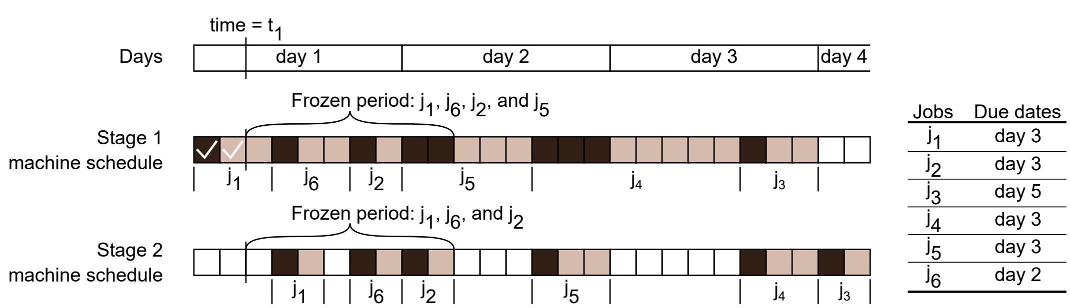
Lastly, the industrial partner uses a special planning technique for the stage one machines. In essence, it splits the three printing machines into two groups: 1) two main machines and 2) one buffer machine. The main task of the worker on the buffer machine is to help with the setup processes on the two main machines, which reduces the required setup time by one third, according to our industrial partner. The buffer machine can produce if and only if the entire production process, setup and processing, can be performed fully in a time window where both main machines are producing and the worker on the buffer machine is not helping with the setup on either of the main ones.

2.2 Schedules and their Evaluation

A schedule is a sequence of jobs, which are planned to be processed on a machine. Figure 2a shows an example of two machines and their schedules at the time t_0 for two stages, where five jobs need to be processed (j_1, j_2, j_3, j_4, and j_5.) The darker squares represent the setup time and the lighter ones the processing, whereas the white squares show unplanned time. The frozen period on the first stage machine encompasses j_1, j_2, and j_5; and on the second stage machine j_1 and j_2. Additionally, the due dates for each job are noted on the second stage schedules. Moreover, Figure 2b depicts how the initial schedule from the time t_0 is adapted when a rush order (j_6, with a due date on day two) arrives at time t_1. The new job is inserted into the frozen period on both stages, i.e., between j_1 and j_2 on the first and second stage. Furthermore, to prevent j_4 from being late it is switched with j_3. However, this causes the loss of the setup benefit from processing j_3 before j_4.



(a) Example schedule at time t_0, with due dates.



(b) Adapted schedule at time t_1 - after the rush order j_6 is inserted into the frozen period, with due dates.

Figure 2: Example schedules of a two-stage two-machine production before and after a rush order arrival.

To evaluate the schedules, two objective functions are used: 1) the total weighted tardiness $wT = \sum_{i=1}^{n} w_i T_i$, and 2) the robustness $R = {\sum_{i=1}^{n} w_i(d_i - C_i)} / {\sum_{i=1}^{n} w_i d_i}$ (Pinedo 2009).

Where the w_i is the individual importance weight of an order, T_i stands for its tardiness (or positive lateness $T_i = max(C_i - d_i, 0)$), C_i is the completion time, and d_i represents the due date. It is important to note that both the completion times and due dates are measured in whole workdays.

wT is a weighted measure of the tardiness of products; and R is a weighted measure of how much in advance these are scheduled. A high robustness means that the products are produced sufficiently in advance and a rush order can be squeezed in without deteriorating the schedule's performance. These two formulations are used because, as stated in the introduction, the main challenge, which the industrial partner faces when scheduling, is utilizing the highest amount of flexibility to accommodate highly important rush orders, while preventing tardiness and thereby not punishing customers, who ordered ahead of time.

2.3 General Assumptions

The following simplifications were assumed. All materials and tools are always on stock and available in the right quantities. There are no work-in-progress and finished product area limitations, i.e., there is always enough space for waiting products. There are no machine breakdowns or employees on sick leave, i.e., the machines are available each workday for both shifts. Moreover, in the observed period there are no vacation days. The manufactured products are of perfect quality, i.e., there are no rejects or scrap. Additionally, there are no preemptions, which means once a production starts it must be fully processed.

3 SOLUTION APPROACH

3.1 Multi-Objective Large Neighbourhood Search

To generate feasible and potentially better solutions than with manual planning, a Multi-Objective Large Neighborhood Search (MO-LNS) algorithm, which is based on an LNS algorithm, was chosen, as it can be easily adapted to a complex environment due to its flexible meta-heuristic nature. Also, the utilized MO framework enables the usage of unconnected objectives with different value ranges and/or units.

Implementing an LNS algorithm into an MO framework was recently done by Oddi et al. (2015) and Schaus and Hartert (2013). Both started by first changing the definition from one best solution to a set of best solutions, i.e., a set of Pareto-efficient solutions.

For the characterization of the best solution sets the principles outlined by Oddi et al. (2015) were followed. They used two separate sets of best solutions: 1) an elite-set ($PS*$) and 2) a next-best set (PS); both abiding to the non-inferiority principle. The former contains overall dominant solutions, while the latter contains solutions dominated only by the solutions of the elite-set to a certain degree, which is set by a threshold parameter T; this parameter controls the level of diversification. In contrast, not just a single threshold but two were implemented, i.e., a minimum threshold T_{min} and a maximum threshold T_{max}, respectively. Such an approach fits the use case because the diversification of marginally improving solutions can be controlled effectively by setting appropriate threshold parameters.

Moreover, the selection of the starting solution for each run is not just random but follows the approach of Schaus and Hartert (2013), i.e., the nearest neighbor, which is much more sophisticated. With it the voids between Pareto-efficient solutions are filled more efficiently. Here, a random point, with a uniform distribution, on the hyper-plane spanning between the extreme solutions is selected. Then the solution with the smallest Euclidean distance to the random point is picked as the initial solution. It is important to add that the search for a starting solution considers both solution sets.

For the destroy-heuristic an adapted version of the Shaw Removal Heuristic (SHR), which was first introduced by Shaw (1997), was implemented. It is based on the idea that it is easier to rearrange similar products than different ones, as changing the positions of products with comparable production times is less disturbing for the schedule. Furthermore, this heuristic uses two control parameters: 1) the removal limiter γ and 2) the relatedness weight δ. The former denotes the percentage of the solution to be destroyed, while the latter how strongly similar jobs should be favored.

For rebuilding a destroyed solution, the Basic Greedy Insertion (BGI) heuristic, which attempts to find the best place to reinsert a product into the schedule following the sequence in which they were removed, was implemented. This heuristic offers very good results with acceptable calculation times.

The pseudo-code of the used LNS algorithm is shown in Algorithm 1 and the adapted MO-framework is shown in Algorithm 2.

Algorithm 1 Used LNS Algorithm.

input: $k_{fail}, objective, x_{start}$;
$x_b \leftarrow x_{start}$;
$k \leftarrow 0$;
while $(k < k_{fail})$ **do**
    $x_{temporary} \leftarrow r(d(x, objective))$;
    if $(c(x_{temporary}, objective) < c(x_b, objective))$ **then**
        $x_b \leftarrow x_{temporary}$;
        $k \leftarrow 0$;
    else
        $k++$;
return (x_b);

Algorithm 2 Used MO-LNS Framework.

Let PS^* and PS be two sets of solutions;
Generate initial solution using some heuristic and add it to PS^*;
Set k_{max}, k_{fail} and T;
while $(k < k_{max})$ **do**
    $x_{start} \leftarrow$ nearestNeighbour(PS, PS^*);
    $strategy \leftarrow$ selectRandomOptimizationObjective(obj_1, obj_2);
    $x_{temporary} \leftarrow$ LNS$(k_{fail}, strategy, x_{start})$;
    updateEliteSet$(x_{temporary}, T, PS^*, PS)$;
    $k++$;
return (PS^*);

3.2 Simulation Model

The industrial partner's production facility was replicated using a DES model. It was used because events and states, which are cornerstones of DES, according to Cassandras and Lafortune (2008), are detailed enough to describe the relevant processes in such a manufacturing environment. These events include new orders entering the systems, shift starts and finishes, setup starts and finishes, production starts and finishes, waiting period starts and finishes, and products being shipped out. Relevant states for machines are: idle (or waiting for scheduled products), shut-off, and processing; and for the products consist of: unprocessed, processed on the first stage, and processed on both stages, i.e., finished.

The first step of designing the shop floor model was the definition of a software independent conceptual model. For this the Hierarchical Control Conceptual Modelling (HCCM) framework, which was first introduced by Furian et al. (2015), was used. This framework replaces conventional queues and conditional activities with a hierarchical control tree with embedded general rules, thus offering the practitioner a higher degree of flexibility even before programming the actual model. In addition to these control trees, which consist of multiple control units, the framework uses entities, events, requests, and activities. We refer the interested reader to Furian et al. (2015) for a more in-depth elaboration of the framework.

In this use-case the model's hierarchical control tree consists of two levels, the shop floor control unit controlling two underlying stage control units - a stage 1 control unit and a stage 2 control unit. In addition to the two stage control units, the main control unit oversees the shift starts and finishes. Furthermore, each of the two stage control units control the processing and waiting activities of all products and machines.

In the next step the DES model was developed. Here the HCDESLib, which is an open-source C# library for DES from the same developers as the used framework, was used. Also, this step included the definition of appropriate interfaces in the main control unit, through which various scheduling algorithms can be implemented and changed easily.

Lastly, the model was verified and validated; the former by checking and comparing the process flow of the model with the real production; and the latter by using a real data-set of a period and comparing the achieved utilization rates of the machines. What is more, in the validation runs some of the solutions on the Pareto-Frontier, which was generated by our algorithm, were slightly better than the real production, with respect to total weighted tardiness. However, the real production plan follows negotiated due dates as well as a handful of make-to-stock products, which we did not have. Everything considered, the developed algorithm is able to generate schedules, which perform nearly on the same level as the industrial partner's production planner with decades of experience.

3.3 Design of Experiments

The goal of the experiments is to assess whether locally optimizing (or rescheduling) with less focus on the tardiness at the current time, i.e., the objective function weight of total weighted tardiness, and more focus on preventing future tardiness, i.e., the objective function weight of robustness (or flexibility), over a period yields an improved global optimum with regard to total weighted tardiness. The findings serve as support for setting the best objective function weights when rescheduling an order situation with a particular level of dynamism. Also, the design of experiments (DoE) nomenclature is based on Giunta et al. (2003).

3.3.1 Design Constants

The following parameters remain constant for all samples and experiments. The number of machines on both stages, production shift start and end times, the planners' shift start and end times, the periodic scheduling times, the definition of rush orders, normal orders, and omitted orders, frozen period duration, and the simulation and evaluation periods.

As previously mentioned, the first stage consists of two main machines and one buffer machine, each with different capabilities, while the second stage consists of eight machines, also with different capabilities. The production starts at 6.00 AM and finishes at 8.30 PM each weekday (except national holidays), where breaks and shift exchanges do not interfere with production; also, the production pauses when the shift ends and resumes uninterrupted when the next one starts. The planners start their shift at 8.00 AM and end it at 4.30 PM, during which schedules can be adapted. Further, if a rush order arrives outside of that time-window, it must wait until the planners' next shift starts. What is more, the schedules are updated twice per day, no matter how many and which new orders arrived; this happens at 8.15 AM and 4.15 PM. A rush order has a due date of less than or equal to two workdays from the time of arrival, normal orders a due date of more than two workdays, and omitted orders a due date of more than 15 workdays; the latter ones are scheduled only after their due date gets down to 15 workdays. The frozen period lasts for two shifts, i.e., 14.5 hours, starting from each scheduling. The production is simulated from Monday, 3.6.2019, to Thursday, 1.8.2019. To avoid transient machine behavior at the start the evaluation period starts one week later, i.e., 10.6.2019.

3.3.2 Design Variables

The design variables are divided into two groups: 1) the objective function weights and 2) the job sets, where the former control the focus of the optimization, whereas the latter sets the level of dynamism.

The objective function weights are part of a design space with a lower bound of 0 and an upper bound of +1 and their sum always equals one. Furthermore, the weight pairs are observed in increments of .1, which results in 11 distinct combinations. This number of steps provides a good trade-off between the level of gained insight and computational requirements.

Each sample of jobs consists of base jobs and additional rush orders. The analysis of the list of the provided annual orders showed that the industrial partner received an average of 52 orders each week in 2019. This is taken as the constant number of base jobs per week. Contrarily, the number of rush orders per week is part of a design space with a lower bound of 0 and an upper bound of +15 and is observed in increments of five, which results in four distinct levels of dynamism.

Moreover, the used data-sets were generated from the industrial partner's annual orders list for the year 2019. First, the orders of sequential months were combined into two-month sets; then, the sets were further subdivided into weeks. Lastly, the orders were re-grouped by weeks one through ten across the two-month sets. This way seasonal effects, unusually dynamic periods, as well as exceptionally stale ones are smoothed out. The resulting weekly order sets serve as the sampling pool. This was done to create generic job sets with a constant degree of dynamism throughout the observed period.

3.3.3 Experiments and Samples

For this use-case we designed four experiments, one at each level of dynamism, i.e., job sets with zero, five, ten, and 15 rush orders per week. Further, the base orders of a job set are used in each of the experiments, with the rush orders being added according to the experiment's observed dynamism, thereby improving the comparability despite a smaller sample size. Lastly, each of the samples in an experiment is subdivided into 11 sub-samples, one for each distinct objective weight pair. The use of a sample across the four experiments (or dynamism levels) is depicted in Figure 3.

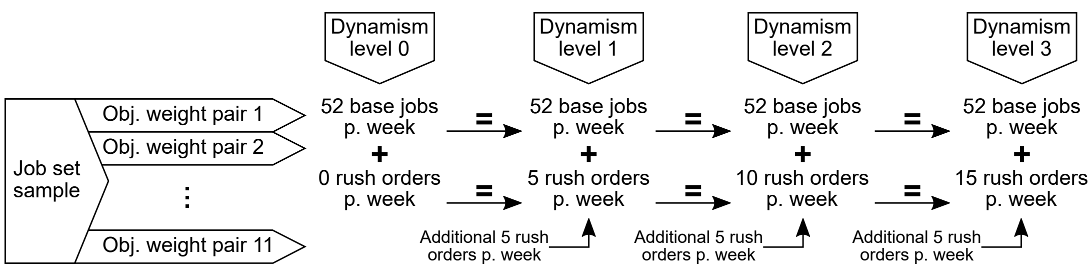


Figure 3: Overview of the use of a job set sample across the four observed levels of dynamism.

In summary, each job set is simulated 44 times, 11 distinct objective weight pairs at each of the four levels of dynamism. To achieve a level of statistical relevance 198 job sets were generated from the sampling pool, resulting in 8.712 simulation runs. Furthermore, combining the number of simulations with the model's complexity leads to expensive computations; the number of re-optimizations, i.e., the number of rush orders and their inter-arrival time, being the detrimental factor. Thus, the experiments were carried out on two simulation servers with two Intel(R) Xeon(R) processors with eight cores running at 2.00 GHz using 262.144 GB RAM, each; also, both running on Microsoft Windows 10.

4 RESULTS

As the first step, the average total weighted tardiness of each experiment across all samples for each sub-sample is observed. The data for this analysis is presented in Table 1. From these results no clear conclusion can be made as to which objective weight scenario produces schedules of the highest quality, regarding weighted tardiness at a certain level of dynamism. Therefore, a more detailed approach is necessary.

For the initial step of the comprehensive analysis the results were aggregated into four 11x11 matrices, each representing an experiment by combining the results across all samples. The resulting heat maps are

Table 1: Average total weighted tardiness across all samples for each experiment.

Objective weight pair	wT objective function weight	R objective function weight	Average total weighted tardiness of each experiment			
			Experiment 1 (dynamism level 0)	Experiment 2 (dynamism level 1)	Experiment 3 (dynamism level 2)	Experiment 4 (dynamism level 3)
0	1.0	0.0	0.449	7.672	19.929	39.727
1	0.9	0.1	0.434	7.606	20.581	39.879
2	0.8	0.2	0.434	7.692	20.434	39.449
3	0.7	0.3	0.419	7.737	20.131	39.732
4	0.6	0.4	0.409	7.429	19.455	40.667
5	0.5	0.5	0.449	7.439	20.439	40.040
6	0.4	0.6	0.444	7.515	20.409	41.439
7	0.3	0.7	0.434	7.975	20.177	40.525
8	0.2	0.8	0.455	7.843	20.177	41.318
9	0.1	0.9	0.460	7.636	20.066	41.308
10	0.0	1.0	0.465	7.732	20.591	42.646

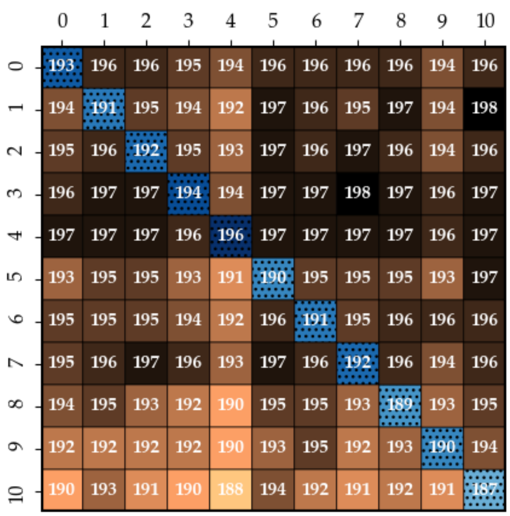
shown in Figure 4. The values in the diagonals count the number of times an objective weight pair was able to create a schedule with the lowest total weighted tardiness among all 11 objective weight pairs of an experiment. The rest is filled after the following rule. Assume that the rows are represented by an index i and the columns by j. Thus, the non-diagonal values count the number of times where the objective weight pair i was able to create a schedule with a lower than or equal total weighted tardiness than pair j. Moreover, the higher the value of a cell the darker the color, for both the diagonal and non-diagonal.

The heat map of the first experiment (Figure 4a) shows no clear patterns, which is expected as there are no rush orders. Nevertheless, a slightly darker path forms in the rows of the objective pair numbers three and four. In particular, pair number four's row being the darkest, i.e., beating the other pairs the most, and its column being clearly the lightest, i.e., getting beaten by the others the least, which means that it is marginally dominant. Further, comparing the segment under pair number four, row-wise, and left of it, column-wise, with its counter part, which is mirrored across the diagonal, it is evident that the objective pairs up to and including number four dominate the remaining ones somewhat.

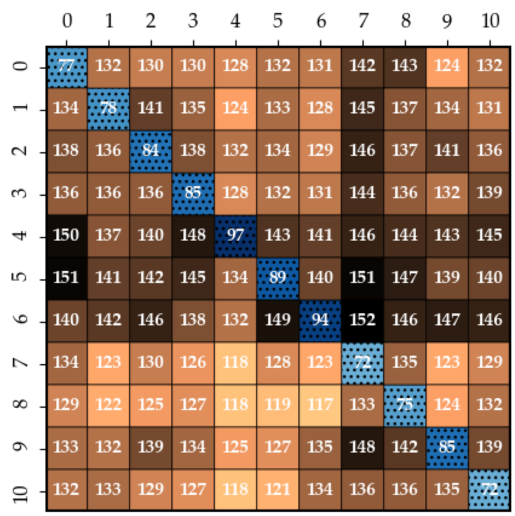
The next heat map, i.e., experiment two (Figure 4b), shows a slight peak on the diagonal for the objective weight pair numbers four, five, and six. Notably, the rows of the weight pair numbers four and six form the darkest path, on average beating the other scenarios in over 70% of the experiments. Again, objective weight pair number four's column is the lightest, i.e., getting beaten by the other pairs the least, indicating a slight dominance. What is more, analyzing the segment under objective weight pair number six and its mirrored counterpart, analogue to the previous experiment, it appears that the pairs up to and including number six dominate the rest slightly.

In the penultimate heat map, i.e., experiment three (Figure 4c), the diagonal appears distinctly low, with a minor peak at the objective weight pair numbers three and four. Especially, pair number four producing the darkest path, i.e., beating the others the most, specifically in slightly more than 60% of the samples. At the same time, the pair is the lightest column-wise. Unexpectedly, the next most dominant pair appears to be number nine, which might be caused by a handful of outstandingly dynamic samples. Another point supporting this claim is the rise of diagonal values from objective weight pair number six on, peaking at the last one. Moreover, inspecting the segment under the objective weight pair number four and its mirrored segment, same as in the first experiment, suggests that the pair numbers three and four dominate the pair numbers five and six marginally, after that the dominance fades again.

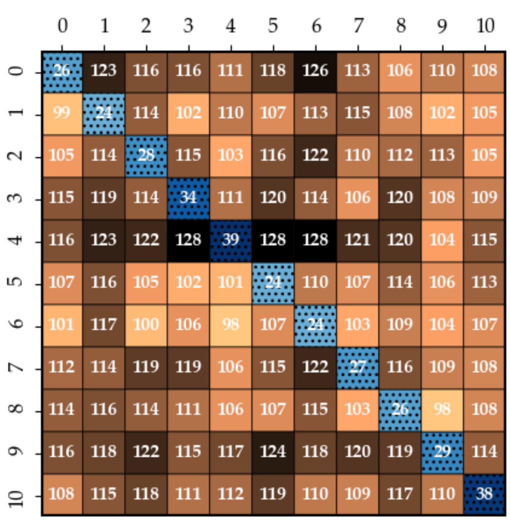
In the last heat map, i.e., experiment four (Figure 4d), the diagonal values are clearly lower than the rest, as with the previous experiment, with a slim peak forming at objective weight pair number two. Also, the row of this pair is a shade darker, i.e., is beating the others somewhat more, with the pair number



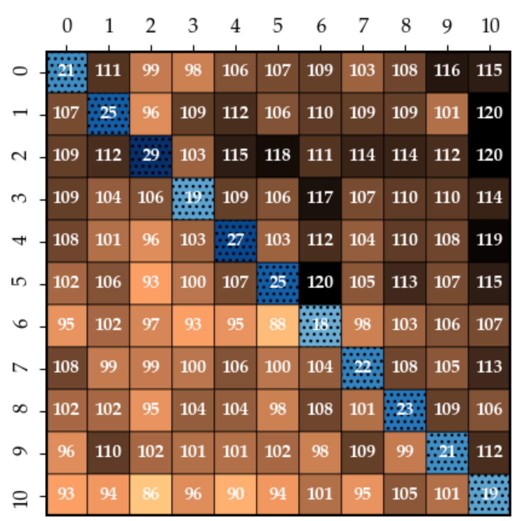
(a) Heat map of experiment 1.



(b) Heat map of experiment 2.



(c) Heat map of experiment 3.



(d) Heat map of experiment 4.

Figure 4: Heat maps of the experiments, with the darker colors representing better values.

three achieving similar numbers. Further, considering the matrix segment under pair number three and its mirrored opposite, as in experiment one, it is evident that the initial four pairs dominate the rest slightly.

Everything considered, the results do not highlight an objective weight pair, which dominates the rest by a significant margin, in any experiment. At the same time, the outcome suggests that by using an objective weight pair from a certain range of pairs is highly likely going to result in schedules, which prevent more tardiness than the others by utilizing an appropriate amount of focus on the objective of robustness.

5 CONCLUSION

In this paper we present the initial results of our attempt at assessing the dominance objective function weight pairs at various degrees of order dynamism in an attempt to optimize the total weighted tardiness of a scheduling problem over multiple planning horizons using two objective functions with numerous objective weight pairs. The primary objective being the total weighted tardiness - improving the target objective for the current planning horizon. The secondary objective being the robustness - preparing the schedule for random rush orders in a way to potentially prevent tardiness in future planning horizons. These two objectives were utilized in an MO-LNS algorithm, which was implemented in a DES, following the HCCM framework. With it we replicated a two-stage printing production facility and thereby observed the effect of varying objective function weights at different levels of order dynamism.

Overall, the obtained results do not point at an outstandingly dominant objective function weight pair at any observed level of dynamism. Nonetheless, the outcome of the simulations implies that specific ranges of objective weight pairs slightly dominate others. Thus, when rescheduling and choosing a solution on the Pareto-frontier, which corresponds to a weight pair from the dominant range for the current degree of order dynamism, the total weighted tardiness is highly likely going to be reduced over a period. In the first experiment the initial five pairs dominate the rest, in the second one the first seven weight pairs dominate the remaining ones, in the third experiment only the pair numbers three and four dominate the pairs five and six, and in the last experiment the initial four weight pairs dominate the rest.

The validity of this study would benefit through additional simulation runs. What is more, to further improve the understanding of the influence of the robustness as a secondary objective function the levels of dynamism should be controlled with more parameters than just the number of rush orders. Another interesting research topic would be to formalize and study the effectiveness of different definitions of robustness as a means of hedging schedules against future tardiness caused by random rush orders.

REFERENCES

Cassandras, C. G., and S. Lafortune. 2008. *Introduction to Discrete Event Systems*. 2 ed. Boston, MA: Springer.

Chaudhry, I. A., and A. A. Khan. 2016. "A research survey: review of flexible job shop scheduling techniques". *International Transactions in Operational Research* 23:551–591.

Du, J., and J. Y.-T. Leung. 1990. "Minimizing Total Tardiness on One Machine is NP-Hard". *Mathematics of Operations Research* 15:381–572.

FFTA 1999. "Flexography: Principles & Practices". In *Volume 1* (5 ed.). Ronkonkoma, NY: Foundation of Flexographic Technical Association, Inc.

Furian, N., M. O'Sullivan, C. Walker, S. Vössner, and D. Neubacher. 2015. "A conceptual modeling framework for discrete event simulation using hierarchical control structures". *Simulation Modelling Practice and Theory* 56:82–96.

Giunta, A., S. Wojtkiewicz, and M. Eldred. 2003. *Overview of Modern Design of Experiments Methods for Computational Simulations (Invited)*.

González-Neira, E. M., J. R. Montoya-Torres, and D. Barrera. 2017. "Flow-shop scheduling problem under uncertainties: Review and trends". *International Journal of Industrial Engineering Computations* 8:399–426.

Jensen, M. T. 2001. "Improving robustness and flexibility of tardiness and total flow-time job shops using robustness measures". *Applied Soft Computing* 1:35–52.

Jensen, M. T. 2003. "Generating robust and flexible job shop schedules using genetic algorithms". *IEEE Transactions on Evolutionary Computation* 7:275–288.

Kirlik, G., and C. Oguz. 2012. "A variable neighborhood search for minimizing total weighted tardiness with sequence dependent setup times on a single machine". *Computers & Operations Research* 39:1506–1520.

Komaki, M., S. Sheikh, and E. Teymourian. 2014. "A Hybrid Harmony Search algorithm to minimize total weighted tardiness in the permutation flow shop". In *2014 IEEE Symposium on Computational Intelligence in Production and Logistics Systems (CIPLS)*, 1–8.

Lee, Y. H., K. Bhaskaran, and M. Pinedo. 1997. "A heuristic to minimize the total weighted tardiness with sequence-dependent setups". *IIE Transactions* 29:45–52.

Mehta, S. V., and R. M. Uzsoy. 1998. "Predictable scheduling of a job shop subject to breakdowns". *IEEE Transactions on Robotics and Automation* 14:365–378.

Mohan, J., K. Lanka, and A. N. Rao. 2019. "A Review of Dynamic Job Shop Scheduling Techniques". *Procedia Manufacturing* 30:34–39.

Nishi, T., Y. Hiranaka, and M. Inuiguchi. 2010. "Lagrangian relaxation with cut generation for hybrid flowshop scheduling problems to minimize the total weighted tardiness". *Computers & Operations Research* 37:189–198.

Oddi, A., R. Rasconi, and A. Cesta. 2015. "A Multi-objective Large Neighborhood Search Methodology for Scheduling Problems with Energy Costs". In *2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)*, 453–460.

Pinedo, M. L. 2009. *Planning and Scheduling in Manufacturing and Services*. 2nd ed. New York, NY: Springer.

Schaus, P., and R. Hartert. 2013. "Multi-Objective Large Neighborhood Search". In *Principles and Practice of Constraint Programming. CP 2013. Lecture Notes in Computer Science, vol 8124*, edited by C. Schulte, 611–627. Berlin, Heidelberg: Springer.

Shaw, P. 1997. "Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems". In *Principles and Practice of Constraint Programming — CP98. CP 1998. Lecture Notes in Computer Science, vol 1520*, edited by M. Maher and J. Puget, 417–431. Berlin, Heidelberg: Springer.

Xie, J., L. Gao, K. Peng, X. Li, and H. Li. 2019. "Review on flexible job shop scheduling". *IET Collaborative Intelligent Manufacturing* 1:67–77.

AUTHOR BIOGRAPHIES

ŽIGA LETONJA is a project assistant at the Department of Engineering and Business Informatics at Graz University of Technology. He holds a master's degree in production science and management and is a member of the doctoral school in techno-economics with a focus on planning and optimization of supply chain operations. He can be contacted by email at ziga.letonja@tugraz.at.

NIKOLAUS FURIAN is an Assistant Professor in the Department of Engineering and Business Informatics at Graz University of Technology. He holds a master's degree in technical mathematics and a Phd in industrial engineering with focus on operations research. His main research interests are in the fields of simulation, optimization, and data analytics. He can be contacted by email at nikolaus.furian@tugraz.at

JOHANNES PAN is a project assistant at the Department of Engineering and Business Informatics at Graz University of Technology. He holds a master's degree in production science and management and is a member of the doctoral school in techno-economics with a focus on the maintenance strategy selection problem. His main research interests are in the field of optimization in maintenance operations, data analytics, and risk analysis. He can be contacted by email at johannes.pan@tugraz.at.

SIEGFRIED VÖSSNER is a Professor and the head of the Department of Engineering and Business Informatics at Graz University of Technology. He holds a Ph.D. in Mechanical / Industrial Engineering. His research interests include modeling and simulation of engineering-, business- and social systems as well as systems architecture and systems engineering. He has also been a project manager for McKinsey & Company and a visiting scholar / visiting professor at Stanford University and at the University of Auckland, NZ. He can be contacted by email at voessner@tugraz.at.

MELANIE REUTER-OPPERMANN is a Postdoc at the Information Systems group at the Technical University of Darmstadt. She holds a PhD in operations research from the Karlsruhe Institute of Technology (2017.) Her research interests include (multi-objective) optimization, healthcare, and e-mobility. She can be contacted by email at oppermann@is.tu-darmstadt.de.