An Algorithm for Efficient Generation of Customized Priority Rules for Production Control in Project Manufacturing with Stochastic Job Processing Times

Mathias Kühn *, Michael Völker and Thorsten Schmidt

Institute of Material Handling and Industrial Engineering, Technische Universität Dresden, 01062 Dresden, Germany; michael.voelker@tu-dresden.de (M.V.); thorsten.schmidt@tu-dresden.de (T.S.)

* Correspondence: mathias.kuehn@tu-dresden.de

Abstract: Project Planning and Control (PPC) problems with stochastic job processing times belong to the problem class of Stochastic Resource-Constrained Multi-Project Scheduling Problems (SRCMPSP). A practical example of this problem class is the industrial domain of customer-specific assembly of complex products. PPC approaches have to compensate stochastic influences and achieve high objective fulfillment. This paper presents an efficient simulation-based optimization approach to generate Combined Priority Rules (CPRs) for determining the next job in short-term production control. The objective is to minimize project-specific objectives such as average and standard deviation of project delay or makespan. For this, we generate project-specific CPRs and evaluate the results with the Pareto dominance concept. However, generating CPRs considering stochastic influences is computationally intensive. To tackle this problem, we developed a 2-phase algorithm by first learning the algorithm with deterministic data and by generating promising starting solutions for the more computationally intensive stochastic phase. Since a good deterministic solution does not always lead to a good stochastic solution, we introduced the parameter Initial Copy Rate (ICR) to generate an initial population of copied and randomized individuals. Evaluating this approach, we conducted various computer-based experiments. Compared to Standard Priority Rules (SPRs) used in practice, the approach shows a higher objective fulfilment. The 2-phase algorithm can reduce the computation effort and increases the efficiency of generating CPRs.

Keywords: simulation-based optimization; stochastic project scheduling; genetic algorithm; discrete event simulation; composite priority rules

---

1. Introduction

One of the challenges in industrial environments is the handling of a large number of product variants [1] (p. 46). In production, this trend is accompanied by shorter product life cycles and smaller batch sizes with simultaneously increased demands on the objective such as shorter delivery times [2] (p. 5). Short delivery times play a decisive role for the customer [3] (pp. 25–26). The resulting shorter time for product development as well as the prototype character of the products lead to an increase in fuzzy and missing process parameters. These circumstances result in additional requirements for Project Planning and Control (PPC), especially in complex assembly processes that are characterized by human work. Stochastic influenced job processing times are inherent in the process due to human error [4] (p. 31) and will therefore continue to exist in the future. Representative examples of the characteristics mentioned above include the final assembly of customer-specific machine tools and printing machines. An example for job processing time deviations in the considered industry is shown in Figure 1. In addition, each project has its own objectives. At present, neither centrally acting

Manufacturing Execution Systems (MES) with integrated Advanced Planning and Scheduling Systems (APS-Systems) [5] (pp. 63–64) nor decentrally used Simple Priority Rules (SPRs, e.g., FIFO (First In–First Out)) can be used to control these requirements [6] (p. 8).
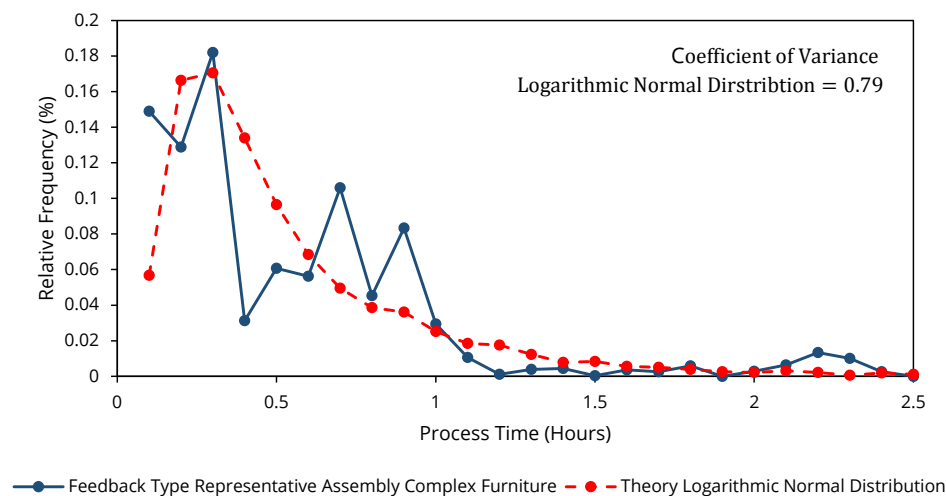
Figure 1. Example of job processing time deviations for a type representative in individual furniture assembly.

In sequence control performed by MES and APS, production follows a deterministic schedule that defines the next job [7] (p. 392). For the industrial domain under consideration, this approach is only suitable to a limited extent, since permanent rescheduling is necessary due to the fuzzy process parameters. Due to the lack of feedback on the processing status of individual work packages [8] (pp. 80–81), it is not clear whether a schedule is still valid or not. In short, the schedule can be obsolete after release. Furthermore, the necessary input of master and transaction data is often insufficient [9]. Sequencing with Priority Rules (PRs) means that a priority (e.g., shortest processing time) is assigned to each job. The priority determines the next job [10] (p. 111). However, the effect of SPRs on the objective values cannot be predicted in a dynamic production environment [11]. Moreover, project-specific objectives are not explicitly taken into account. Promising approaches consist in the combination of priority rules, the so-called Combined Priority Rules (CPRs). Here, different job attributes are combined, and thus, a priority index is calculated for determining the next job. In project manufacturing, CPRs are rarely used for decentralized control. On the one hand, the time for generation is very high; on the other hand, approaches for project-specific objectives are rare. In this paper, we take up the challenge of computational time for the generation of CPRs and develop a concept for project-specific objective fulfillment under stochastic influences.

In the presented paper, we generate project-specific CPRs by combining different individual job attributes with a weighted sum approach. These CPRs are assigned to the project jobs and used for short-term project scheduling. The proposed Genetic Algorithm (GA) used as a hyper-heuristic optimizes weight allocation of these individual attributes. For this purpose, we generate the CPRs by iterative simulation and evaluate the results with the Pareto dominance concept. Since the resulting optimized CPRs in deterministic condition do not necessarily lead to good results under stochastic conditions, we propose a 2-phase algorithm to determine the CPRs. The first phase serves to explore the solution space and to select the best individuals for the second stochastic phase. We validate the concept with computer-aided experiments on a software platform developed especially for this problem. As a simulation method, we use discrete event simulation combined with the theory of simulation-based optimization. Within the scope of the paper, we perform experiments on computation time and performance of the algorithm. In the context of the paper, we address the following research questions:

- Is it possible to improve mean value and standard deviation of project individual objectives by applying the generated CPRs?
- Is it possible to save computation time with the 2-phase algorithm?

The structure of the paper is as follows. Section 2 gives an overview of the state-of-the-art literature related to the possibilities of the automated generation of CPRs. Section 3 contains the model description. In Section 4, we describe the developed 2-phase algorithm. Furthermore, we present the developed software framework for experiment execution. In Section 5, we run experiments to investigate the computation time and performance of the approach. Section 6 provides a summary and outlook.

2. State-of-the-Art

2.1. Problem Definition: RCMPSP

The basic model of resource-constrained multi-project scheduling problem (RCMPSP) was developed by Pritsker et al. [12], Lova et al. [13], and Confessore et al. [14], among others. Based on the work by Confessore et al. [14], Homberger [15] (pp. 556–567) formulated some extensions with regard to decentralized control and developed a benchmark library [16] based on this model. The so-called Multi-Project Scheduling Problem Library (MPSPLIB) is a library with 140 different models differing in number of resources, projects, jobs, and network character. Solutions can be uploaded and checked for feasibility. The MPSPLIB is based on models of the Project Scheduling Problem Library (PSPLIB, [17]) which considers the RCPSP. The models differ mainly in terms of the number of projects ($|I| = 2, \dots, 20$) and the number of jobs ($|J_i| = 30, \dots, 120$). Furthermore, there are differences regarding the number of different instances of the individual projects ($iz = 1$: all projects have the same instances; $iz = 2$: each project has an individual instance). For mathematical modelling of the basic problem of the decentral resource constrained multi-project scheduling problem (DRCMPSP) [15] (pp. 556–557), the problem is briefly described as follows: A set of projects with individual objective functions has to be controlled in parallel in a production system. The production system consists of local (only used by project *i*) and global renewable resources (used by all projects $|I|$). Each project consists of a set of jobs with precedence constraints. The processing times of the jobs on the resources are stochastic. At the time of sequencing, only the expected value of the distribution and the coefficient of variation are known. The projects use both local and global renewable resources. Other project properties concern the earliest start time and due date. As Blasewicz et al. [18] proved, the resource-constrained project scheduling problem as a generalization of the job shop problem is one of the NP-hard (non-deterministic polynomial-time hardness) optimization problems. As a generalization of the resource-constrained project scheduling problem, the resource-constrained multi-project scheduling problem with specific extensions of the process time variations considered in this paper thus belongs to the NP-hard optimization problems [19] (p. 153). Exact procedures that lead to optimal results are largely suitable for generating benchmark solutions [20] (p. 149). The number of calculation steps and thus the calculation time increase with the problem size [21] (p. 38). Already from 60 jobs on, most problems cannot be solved with exact methods [22]. For practical problems, which usually involve considerably more jobs, heuristics are therefore indispensable in order to achieve good results in a shorter computation time. The application of the theory of simulation-based optimization, coupled with heuristics, is also more appropriate for complex models if complex relationships cannot be described simply with the help of mathematical formulas. Furthermore, the advantage is the clarity of simulation models (step-by-step event handling by discrete event simulation), which leads to a higher acceptance than with mathematical optimization models [21] (p. 30). For this reason, we use discrete event simulation as a simulation method. There are many software frameworks. With many of these software frameworks, the job-shop problem can be mapped. For modelling project scheduling problems, however, a high effort of customization is necessary.

2.2. Approaches for Decentralized Control of RCMPSP

In contrast to deterministic optimization, at least one parameter in stochastic optimization is subject to random events [23] (p. 45). Solutions, e.g., schedules or PRs, generated under deterministic conditions may be invalid in the stochastic environment [24] (p. 2); [10] (p. 111); [25] (pp. 21–23). The suitable methods for solving the stochastic optimization problem depend mainly on the dimension of the stochastic influences [19] (p. 249). If stochastic influencing variables can only be described by random variables, methods of stochastic optimization are suitable for scheduling [26] (p. 160). With a static approach, the PRs do not change during the observation period. In contrast, in a dynamic approach, the PRs are adapted during the production process. Thus, we focus on stochastic optimization with a static approach of PR since in the case study of the customized assembly of complex large products the job processing times are characterized by random variables of a logarithmic normal distribution and there are no possibilities for adaption of the PR during production.

PRs such as CPRs are suitable for the considered problem domain with similar functionality and application as SPRs. Simply stated, CPRs are attributes calculated according to a rule from the information on the product, process, resource, and system. The result of the calculation determines the job priority in the queue. A comprehensive review about CPRs and hyper-heuristics is given by Branke et al. [10]. Their paper proposes a taxonomy and guidelines for designing CPRs. As a hyper-heuristic, GAs or neuronal networks are mostly used. Multi-objective algorithms, such as the NSGA-III [27], are not used in the literature.
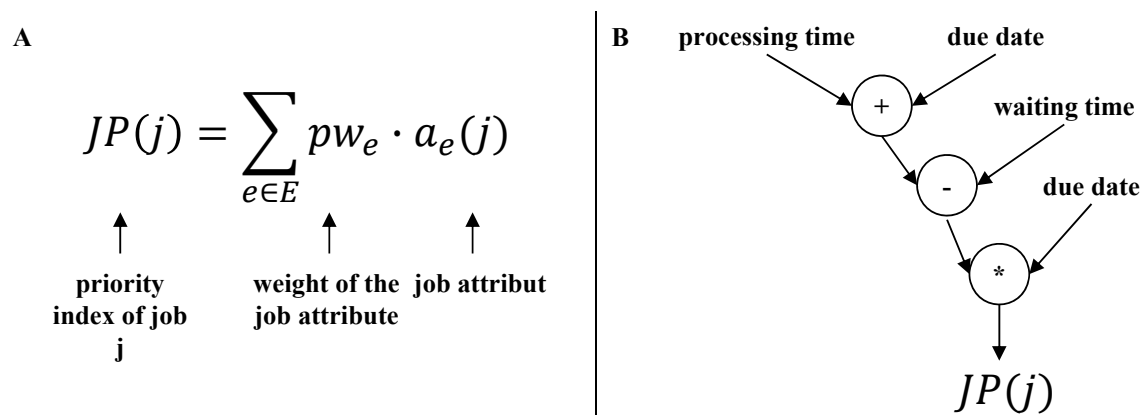
There are a lot of PPC approaches using CPRs for production control. The majority of existing approaches using CPRs focus on job-shop problems [10] (pp. 110–111), which are in general less complex (e.g., less complex precedence constraints and disjoint resources) compared to resource-constrained project scheduling problems (RCPSP). Kück et al. [28] developed a method based on real-time data to adapt the control model based on CPRs. For this purpose, a simulation is permanently carried out at the same time as actual production. If changes or disturbances occur, the simulation model and consequently the control model are adapted. Permanent data availability is necessary for concept implementation. Grundstein et al. [29] presented an approach to maintain planning reliability while simultaneously taking advantage of the benefits of autonomous control. Central planning provides information such as start and end times. Sequential decision trees determine job release, sequencing, and resource allocation. In addition to the information provided by central planning, these decision trees depend on extensive information on system status in production. Moreover, this concept requires an extensive database which is not available yet.

Decentralized approaches for solving the project scheduling problem focus mainly on agent-based auction and negotiation approaches [30] (pp. 695–704). Only Chand et al. [31] solves the control problem of project manufacturing with CPRs in a stochastic environment. Chand et al. [31] considers the stochastic RCPSP. Project-specific objectives are not considered. Hildebrandt et al. [32] also points out that a Pareto optimization (multi-criteria) is hardly considered.

Strategies that are mainly used in shop-floor production can also be implemented in project-oriented production and vice versa. In contrast with shop floor manufacturing with machines as the main resource, project controlled environments with human resources have less sophisticated data. Thus, it should not be assumed in such environments. While automatically generated CPRs are widely used in job-shop scheduling [10], applications of automatically generated CPRs for the solution of the RCPSP [31] and especially for the solution with a decentralized control of the stochastic RCPSP are rather rare in the literature. We can only guess the reasons for this: while the job-shop environment is traditionally stochastic, the RCPSP environment is also increasingly stochastic (Section 1). Classical solutions for RCPSP by generating a baseline schedule are no longer effective, and the interest in scheduling without a baseline schedule increases. One possibility for efficient scheduling without a baseline schedule are those CPRs. The challenge consists in the project-specific fulfilment of different individual objectives under stochastic influences.

2.3. Reducing Computation Time for Generating CPRs

Branke et al. [10] describe the challenge of computation time for CPRs in their work. We draw the following conclusions for our presented work: while sequencing with CPRs is comparatively fast, generating CPRs with hyper-heuristics (e.g., with evolutionary algorithm) is often very computationally intensive. If stochastic values are to be used as recommended during the generation of the CPRs (so-called training phase), the computational effort increases further. A first possibility to reduce computational effort is the representation of CPRs (Figure 2). The grammar-based tree representation is a composition of individual components to a configurable function based on mathematical operators. This representation has in theory no fixed length and is suitable for larger computational budgets [33]. The advantage of this representation is a large search space which fits the large computational budget [34]. Another variant is parameter-based representation, which is a configurable function with a defined format. The defined format means a fixed length and thus a calculable computational effort but, at the same time, a smaller search space.

A

$$JP(j) = \sum_{e \in E} pw_e \cdot a_e(j)$$

↑   ↑   ↑

priority   **weight of the**   **job attribut**
index of job   **job attribute**
j

B   **processing time**   **due date**

(+) **waiting time**

(-) **due date**

(*)

$$JP(j)$$

Figure 2. Combined Priority Rules (CPRs): (**A**) parameter-based representation and (**B**) grammar-based representation.

A further method to reduce computational effort is the generation of an optimized initial population, which is currently of minor interest [35]. Most researchers generate the initial population randomly (e.g., Werner [36]; Hildebrandt et al. [32]; Nguyen et al. [37]). Compared to a randomly generated population, the algorithm converges later than with an optimized diverse population. Thus, a decrease of simulation effort will be evolved with an optimized initial population. An approach for improvement consists for example in the use of existing PRs to form the initial population (e.g., Nguyen et al. [38]; Omar et al. [39]). However, research potential for the generation of good initial solutions is an ongoing issue which we address in our paper.

2.4. Contribution and Motivation

Based on the above study on the use of CPRs in project control, some of the differences in this paper compared to existing research are the following points:

- We consider the use of CPRs for short-term production control in an SRCMPSP environment. The CPRs are assigned to each project.
- We perform a Pareto optimization [40] (pp. 197–199) with the NSGA-III on project level, where the mean and standard deviation of delay and makespan of a single project are considered to evaluate the generated CPRs.
- A deterministic and stochastic optimization phase takes place for reducing computational effort. For the selection of deterministic solutions, we introduce the parameter Initial Copy Rate (ICR), which indicates how many solutions are copied and how many are randomized.
- We developed a software framework for generating CPRs and compared results with several different PRs.

3. Model Extensions of the Stochastic RCMPSP

In order to achieve the requirements of the project-specific objectives, a Pareto optimization is aimed as the multi-objective optimization method ($minimize(f_1(x), f_2(x), \ldots, f_{zf}(x))$ with $zf > 2$ objective functions f_y).

Due to the stochastic influences, it is necessary to evaluate the individual objective values with statistical parameters. A stochastic scenario n is a realized random number of a specific distribution for job processing times (Section 2.1). The mean value and the standard deviation are considered. The mean value is an indicator for optimality of the solution. The standard deviation is an indicator for the robustness of the solution. A low standard deviation means that many solutions are close to the mean value and therefore stochastic influences are better compensated. The following objective functions are set up for the stochastic scenario.

For measuring the tardiness of project i of scenario n, the project delay PD_{in} is calculated by Equation (1):

$$f_y = PD_{in} = \max\left(0, \omega_{|J|in} - fz_i\right) \tag{1}$$

where $\omega_{|J|in}$ is the scheduled finish time of the last job j of project i of the stochastic scenario n and fz_i is the determined finish time of project i.

Another objective value is the makespan MS_{in}, which is calculated by Equation (2):

$$f_y = MS_{in} = \left(\omega_{|J|in} - \alpha_{1in}\right) \tag{2}$$

where α_{1in} is the starting time of the first job j of project i of the stochastic scenario n. The following objective functions are set up for statistic evaluation of the stochastic scenarios. Equation (3) calculates the mean of project delay M_PD_i of $|N|$ stochastic scenarios:

$$f_y = M\_PD_i = \frac{1}{|N|} \sum_{n \in N} PD_{in} \tag{3}$$

Equation (4) calculates the deviation of project delay M_PD_i of $|N|$ stochastic scenarios:

$$f_y = STD\_PD_i = \sqrt{\frac{1}{|N|} \sum_{n \in N} (PD_{in} - M\_PD_i)^2} \tag{4}$$

The corresponding statistical objective values for makespan MS_i are calculated in the same way as Equations (3) and (4) (STD_MS_i; M_MS_i). Statistical parameters for each objective value are assigned individually ($STD\_$ or $M\_$) or in combination ($STD\_$ and $M\_$), so that, in total, six objective functions are considered. For the objective makespan, the objective functions are given as an example. Minimize the average makespan of individual projects M_MS as in Equation (5):

$$f = M\_MS = (M\_MS_i, \ldots, M\_MS_{|I|}) \rightarrow \min \tag{5}$$

Minimize the standard deviation makespan of individual projects STD_MS using Equation (6):

$$f = STD\_MS = (STD\_MS_i, \ldots, STD\_MS_{|I|}) \rightarrow \min \tag{6}$$

Minimize the average and standard deviation of the makespan of individual projects M_STD_MS using Equation (7):

$$f = M\_\_STD\_MS = (STD\_MS_i, \ldots, STD\_MS_{|I|}, M\_MS_i, \ldots, M\_MS_{|I|}) \rightarrow \min \tag{7}$$

4. Proposed Algorithm for Generating CPR

4.1. Representation of CPR

As a sequence heuristic (SH), we use the basic approach of CPRs with weighting factors, the so-called linear representation of CPRs (Figure 2). Job attributes a_e are, for example, the expected processing time or the required completion date. The weightings can be determined either by an expert, randomly, or by an algorithm. As an extension to previous research work and in order to meet the requirement of project-specific objective fulfillment, weighting factors pw_{ie} are assigned individually to each project i. Thus, according to the individual objective function f_y and project characteristics, individual job attributes a_e can be preferred or disadvantaged.

The considered job attributes a_e are summarized in the tuple AA:

$$AA = (a_1, \dots, a_e), \tag{8}$$

with $e \in E$, where $e \in \mathbb{N}\,[1 \dots E]$. A project-specific weight set PSW_i is defined as follows and has the same length like AA:

$$PSW_i = (pw_{i1}, \dots, pw_{ie}), \tag{9}$$

with

$$\sum_{e \in E} pw_{ie} = 1 \text{ where } pw_{ie} \in \mathbb{R}\,[0, 1] \tag{10}$$

For each job in a queue, the first step is to calculate the individual job priority according to the project-specific weight set and job attribute. Formally, the queue is a list of jobs with corresponding job attributes. Normalizing the values is necessary due to the different dimensions of the job attributes. After calculating the priority, the next step is sorting the jobs in descending order of priority. The job with the highest priority is executed next. Calculation of the job priority JP_{ji} of job j of project i is done with the following equation:

$$JP_{ji}\left(ji_e, max\ J_{I_e}\right) = PSW_i \cdot AA\left(ji_e, max\ J_{I_e}\right) = \sum_{e \in E} pw_{ie} \cdot \frac{a_e\left(ji_e\right)}{a_e\left(max\ J_{I_e}\right)} \tag{11}$$

where with ji_e is the considered value of attribute a_e of the considered job j of project i and $max\ J_{I_e}$ is the maximum value of attribute a_e of all jobs J_I in the considered queue. The job attributes considered were based on locally determined attributes (here, among other things, information is obtained from the examples in MPSPLIB) and are oriented on the used information of the SPR (e.g., processing time, due date, starting time, work remain, and cumulative waiting time).

4.2. Two-Phase Genetic Algorithm for Generating CPRs

The parameterization of the weighting set poses a combinatorial problem with a large solution space. It is not possible to check all individual parameter combinations due to the high computational effort involved. To handle such large search spaces, evolutionary algorithms such as GAs are suitable. With regard to the problem of parameterizing the equation for determining job attributes, the GA corresponds to a hyper-heuristic. Therefore, they are suitable for the problem under consideration. Parallel computing can shorten computation time, especially in stochastic optimization. Computation time increases if the stochastic influences are already taken into account during the training phase. Since there is a correlation between the deterministic and stochastic solutions, we see potential in the combination of deterministic and stochastic training data for saving computation time. If the objective is to use this correlation, i.e., to generate a good quality initial population of solutions for a shorter computation time in the next phase, it is necessary to consider a multitude of deterministic solutions in the stochastic environment. We investigate the quantity and the best solutions for this purpose. Within the scope of the present research, a two-phase algorithm is proposed.

Phase one (Figure 3) starts with an initialization. This means that initial parameters of GA are selected along with the production model (planning task), and it is possible to define the attributes used for the CPRs. Next, initial solutions are generated randomly. Random numbers determine the weightings pw_{ie} of the attributes a_e. Based on the defined weightings for the respective attributes, the production is simulated (shop-floor simulation) with deterministic process times. The result is evaluated, and after a negative check of the stop criterion (unchanged objective values after a defined number of iterations), the weightings are changed by the GA. This process continues until the stopping criteria is reached. The result is a set of different weighting variations evaluated with respect to the objective value. In phase two, the corresponding weighting variants from phase one are firstly selected with the parameter initial copy rate ICR. An $ICR = 0.2$ means that the initial solution for stochastic optimization consists of 20% copied solutions from the first phase. The other 80% are random solutions. Then, in accordance with the theory of simulation-based optimization, GA adjusts weights until the stopping criteria is reached. Although the algorithm generated many schedules with different CPRs, the result is only the CPRs with which the best schedules were generated under stochastic conditions. Under stochastic conditions, the objective value will be identical when using this CPRs. The result of the training phase is therefore the result of the test phase.
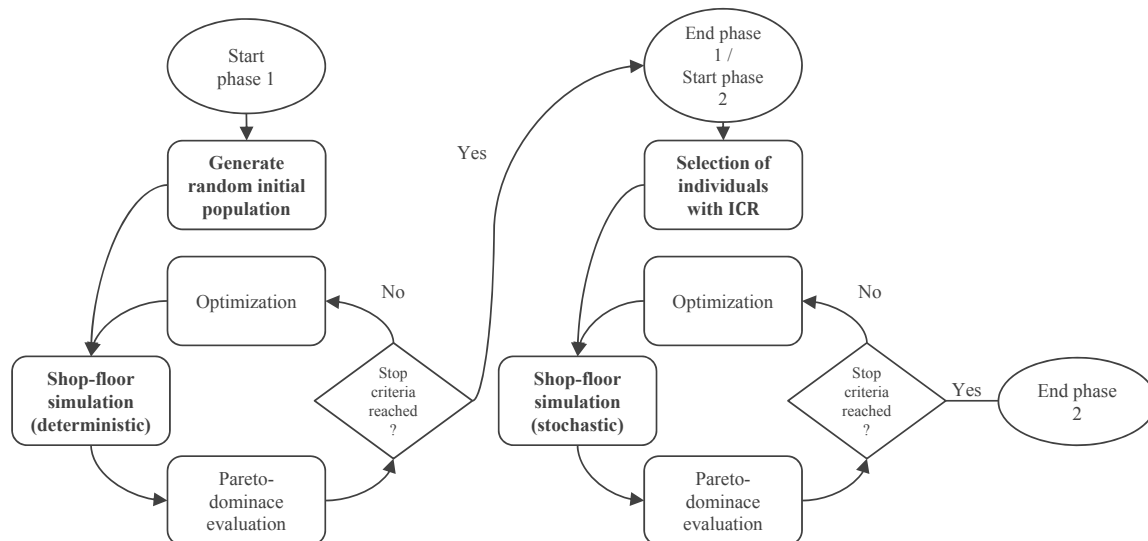

**Figure 3.** Two-phase genetic algorithm.

In accordance with the research aim of project-specific multi-objective optimization, the differentiated evaluation of the quality of solutions (solutions are the project-specific weight sets PSW_i, in the context of GA, an individual Ind) is carried out according to the Pareto dominance concept. Assuming two solution tuples A and B with zf objective functions f_y and the goal of minimizing the objective values $(f_y(a); f_y(b))$, solution A dominates solution B if $f_y(a) \leq f_y(b)$ applies for $y = 1, 2, \dots, zf$ and for at least one y $f_y(a) < f_y(b)$.

Based on this rule, all individuals Ind of a population P_{gen} of generation gen can be compared in the current scenario. Non-dominated solutions are elements of a non-dominated solution front ($front = 1$). Since there are more than one solution in the Pareto front, we aggregate and normalize the individual objective to an overall objective value as a tiebreaker. The best solution gets the index *Best*. The formation of a new population P_{gen+1} is carried out with the NSGA-III. The algorithm was developed for multi-objective problems with $zf > 3$ objective functions f_y. The entire algorithm for generating the CPRs is called Genetic Algorithm Weighted Sum (GAWS).

4.3. Overall Concept for Using CPRs and Proposed Software-Framework

The presented concept (Figure 4) for decentralized sequencing is suitable for short-term sequencing at the operation level. For a short-term horizon, SHs are defined by how individual jobs are operatively sequenced on the shop floor. The result is therefore not an implemented schedule by the control system but evolved CPRs that are applied by the control system. The input to generate the CPRs is therefore a defined project pool with jobs or a predefined time horizon. The definition of the input is not part of the analysis but is taken as given in medium-term planning (e.g., MRP-II). Based on a project pool, the proposed algorithm for generating the CPRs is applied. The CPRs are transferred to production resources and applied for the defined validity period. This process is comparable to transfer of the production schedule to actual production. Various options are conceivable for technical implementation: In addition to the transfer to stationary computers at the workstations, the transfer to mobile devices (smartphones and handhelds) is also possible. A possible technical implementation of the CPRs with regard to job identification is for example barcode technology. The barcode contains all information about the projects. The stationary PC or handheld device contains CPRs for determining job priority. New incoming jobs can thus be added to the queue, and if renewable resources become available, the next job to be processed can be determined according to the CPRs. Thus, a rough status regarding finished jobs is known. Moreover, it is possible to share reconfiguration-relevant events (resource failure, etc.) through feedback and to initiate recalculation of CPRs based on the adjusted production parameters.
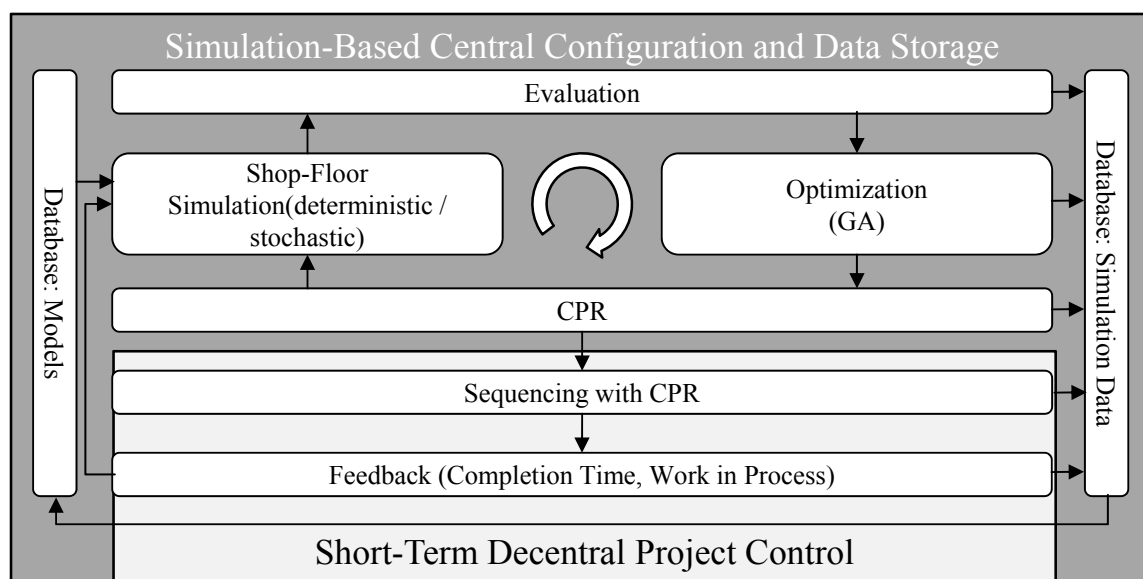
Figure 4. Concept hybrid Project Planning and Control (PPC).

We developed a software framework based on the Python programming language [41] called PyScOp (Python-Based Scheduling and Optimization Framework) to test the algorithm when generating the project-specific CPRs [42,43]. The developed software framework covers all functionalities from model generation to discrete-event simulation-based optimization and evaluation. To calculate the proposed CPRs in the form of weighting sets, the model is first imported. The standard format of MPSPLIB is used. The next step is parameterization of the model (definition of project-specific objectives) and the algorithm (stop criterion, etc.). With subsequent simulation and optimization, the project-specific CPRs are generated. In addition to the standard function for calculating CPRs, various options for evaluating the performance of the algorithm itself and the CPRs are integrated, which largely correspond to the experiments discussed in Section 5. For technical details, please refer to the method documentation within the software code. The software framework has a graphical user

interface for simplified use by third parties. Intended tool tips make it easier for the first user to carry out experiments.

5. Results

5.1. Experiment Design for Concept Evaluation

As mentioned, we use benchmark problems from the MPSPLIB to evaluate the concept. We carried out preliminary investigations to determine the runtime of an optimization run (Intel® Xenon® Gold 6136 with 3.00 GHz CPU and 64 GB RAM). The average optimization takes about 12 h (average from small model $|J_i| = 30$, $|I| = 2$ to large model $|J_i| = 120$, $|I| = 5$). The number of experiments must be limited for this runtime. We chose a two-factor plan as the experimental plan for the model parameters. Each factor is assigned two levels and is combined. The levels should represent a low and a high level. This type of experimental design is chosen if many factors are to be investigated and experiments are highly computation intensive. To avoid process-inherent coincidences, the number of repetitions per single experiment is set to $|W| = 10$ [44] (p. 630); [45] (p. 593). The selected models of the MPSPLIB differ in the number of jobs $|J_i| = 30$ and 120, number of projects $|I| = 2$ and 5, and different instances (network character) $iz = 1$ and 2, so that a total of 8 models M are considered. The distribution of the process time is considered with $cv_{lognorm} = 0.1$ and 0.9. Other parameters are considered to be multifactorial and combined with the mentioned two-factor plan. These parameters are the objective functions f (6 factor stages) and the initial copy rate ICR (6 factor stages). These factor combinations are combined with the number of stochastic scenarios $|N|$. Standard values are assumed for the genetic operators by Schmidt et al. [43]. The population size is calculated according to Das et al. [46]. The number of stochastic scenarios for the evaluation of one CPR = GAWS is $|N| = 100$ (usually, in Freitag et al. [47], $|N| = 50$, while in Hildebrandt et al. [48], $|N| = 100$) with $n \in \mathbb{N}$.

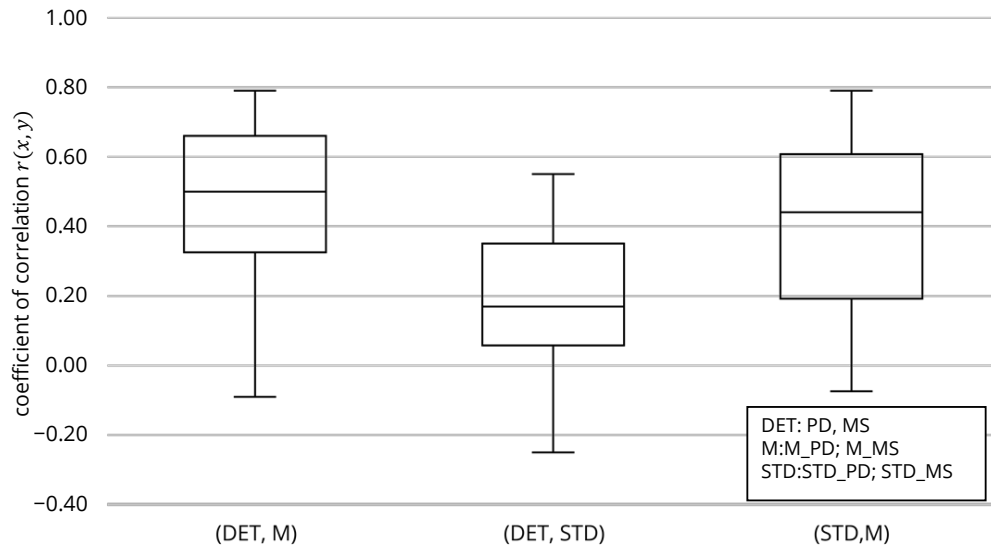### 5.2. Comparing Deterministic and Stochastic Solutions

For the comparison between deterministic and stochastic solutions, a deterministic optimization is performed first. Afterwards, the generated deterministic solutions ($f = PD$ and $f = MS$) are tested with stochastic values of the scenarios $|N|$. $f = M\_PD$, $f = STD\_PD$, $f = M\_MS$, and $f = STD\_MS$ are used as objective functions to evaluate the stochastic influences. The maximum number of generations is $gen = 100$. The deterministic optimization aborts after $gen = 20$ generations with an unchanged objective value. $Exp_{amt} = 2 \cdot cv_{lognorm} \cdot 8M \cdot 6f \cdot 10W = 960$ experiments were performed. To evaluate the correlation, the Pearson correlation coefficient $r(x, y)$ is calculated. For this purpose, the correlation between the deterministic solution (DET) and mean value (M) of the stochastic solution, the correlation between the deterministic solution and standard deviation (STD) of the stochastic solution, and the correlation between the mean value (M) and standard deviation (STD) of the stochastic solution are calculated. The results are shown in Figure 5 as a boxplot diagram and in Table 1 combined with the mean value and confidence interval $ki_{0.95}$.

Table 1. Statistical analysis coefficient of correlation.

$r(x,y)$	$\bar{x}$	$ki_{0.95}$
$DET,\ M$	0.47	± 0.08
$DET,\ STD$	0.18	± 0.07
$STD,\ M$	0.34	± 0.08

A high correlation exists between the deterministic solution (DET) and the mean value of the stochastic solution (M). Furthermore, there is a significant correlation between the mean value (M) and the standard deviation of the stochastic solution (STD). A low correlation exists between the deterministic solution (DET) and the standard deviation of the stochastic solution (STD). This reinforces

the statement that deterministic solutions can be suitable as initial solutions for stochastic optimization. Therefore, setting the parameter initial copy rate *ICR* is of interest.

Figure 5. Boxplot correlation between the deterministic (DET) and stochastic (STD, M) solutions.

5.3. Comparing Computation Effort

In order to evaluate whether the simulation effort and thus the computational effort can be reduced with the presented 2-phase algorithm, a comparison of the simulation effort between the strategy for generating the initial population and the randomized generation of the initial population is necessary. The strategy for optimization of the initial population is called copied in the following, the strategy for the random generation of the initial population is called random. The simulation effort SA_{ICR} to generate the initial population with an initial copy rate of $ICR > 0.0$ is higher in contrast to the simulation effort of the randomized population SA_{rand} with an initial copy rate of $ICR = 0.0$. In order to equate the simulation effort $SA_{ICR} = SA_{rand}$, the following assumptions for calculation of the simulation effort are made and demonstrated using a calculation example: The deterministic simulation effort SA_{det} for the first phase of the 2-phase algorithm with $pop = 100$, generation amount $GA_{det} = 100$, and $|W| = 10$ repetitions is $SA_{det} = 100{,}000$ ($GA_{det} \cdot pop \cdot |W|$). Selection of the 100 best solutions with respect to the objective value out of 10 repetitions $|W|$ with subsequent stochastic simulation with $|N| = 100$ scenarios to generate the stochastic initial population leads to a further simulation effort of $SA_{init} = 100{,}000(pop \cdot |N| \cdot |W|)$ simulations, so that the total simulation effort is $SA_{ICR} = 200{,}000(SA_{det} + SA_{init})$. To compare the simulation effort, for example, more randomized solutions can be generated $SA_{init} = SA_{rand} = 200pop \cdot 100|N| \cdot 10|W| = 200{,}000$. Another possibility is to leave the population size pop unchanged at $pop = 100$ $SA_{init} = 100pop \cdot 100|N| \cdot 10|W| = 100{,}000$ and additionally to perform a generation step $GA_{stoch} = 1$ within the framework of stochastic optimization at $SA_{stoch} = 1GA_{stoch} \cdot 100pop \cdot 100|N| \cdot 10|W| = 100{,}000$, so that the simulation effort with the randomized initial population also corresponds to $SA_{rand} = 200{,}000(SA_{init} + SA_{stoch})$. The latter option was chosen for comparison purposes, since it also uses the mode of action of the genetic algorithm, which requires a certain calculation time and thus makes the results more comparable. Figure 6 illustrates the performance of the different strategies for generating the initial population (randomized $ICR = 0.0$ and copied $ICR = 0.2$–1.0).
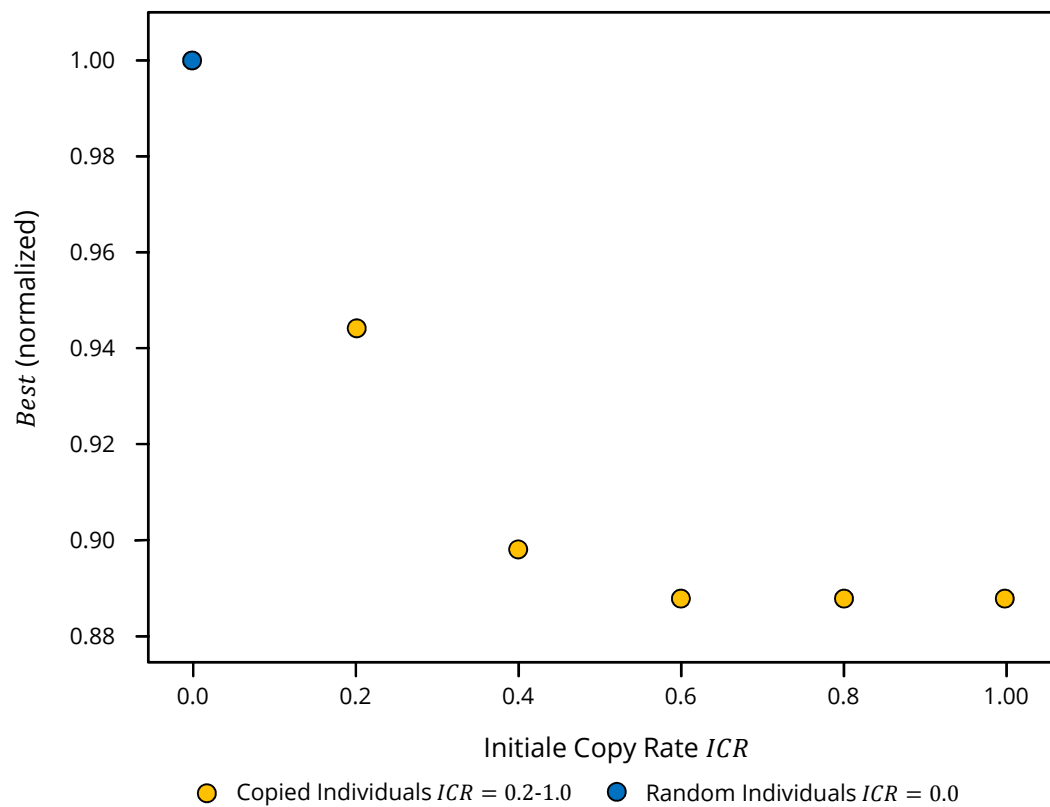
**Figure 6.** Example of how the initial copy rate works; worst value: $Best = 1.00$.

One possibility to evaluate the effectiveness of a copied initial population $ICR = 0.2$–1.0 is to compare the objective fulfillment of the best solution $Best$ per experiment (Figure 7). A condensed evaluation is the relative frequency with which an ICR achieves the best value for the objective value $Best$ per individual experiment. Of the 96 possible individual experiments $(2VarK_{lognorm} \cdot 8M \cdot 6f)$, a copied initial population ($ICR = 0.2$–1.0) achieves the best objective value $Best$ in 79.17% of the individual experiments. In 20.83% of the experiments, a randomized initial population achieves the best objective value ($ICR = 0.0$). The initial copy rate of $ICR = 0.4$ and $ICR = 0.6$ is more often used to achieve the best objective value $Best$ (22.5%) than the initial copy rates $ICR = 0.2, 0.8$, and $1.0 (12\%)$. The following conclusions can be drawn from the distribution of the frequency:

- An optimization of the initial population does not necessarily lead to the best result with respect to the objective value $Best$.
- A low and a too high initial copy rate lead less frequently to the best objective value $Best$. Therefore, it can be concluded that either too many randomized solutions ($ICR < 0.4$) or too many copied solutions ($ICR > 0.6$) do not sufficiently represent the correlation between deterministic and stochastic solution and that, therefore, for a diverse initial population, an equal ratio of copied and randomized solutions is most promising.

The relative saving $Best_{rel}$ between the copied population ($ICR = 0.2, 0.4, 0.6, 0.8$ and 1.0) compared to the randomized population ($ICR = 0.0$) is used to evaluate the performance of the initial copy rate ICR.

A positive value $Best_{rel}$ means a saving; a negative value $Best_{rel}$ is a loss. $Best_{rel}\%$ is calculated by the following:

$$Best_{rel}(\%) = \left(1 - \frac{Best(ICR = 0.2; \ 0.4; \ 0.6; \ 0.8; \ 1.0)}{Best(ICR = 0.0)}\right) \cdot 100 \qquad (12)$$
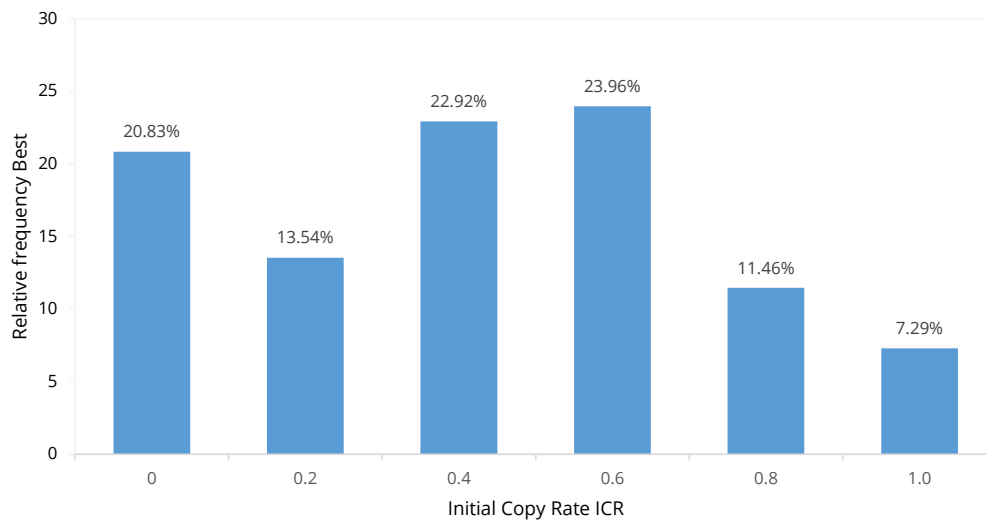
Figure 7. Comparison frequency *Best* between random and copied initial solutions.

With increasing initial copy rate ICR, the range of the relative saving potential $Best_{rel}$ increases (Table 2). From this, it can be concluded that, with initial copy rate ICR, the saving potential $Best_{rel}$ fluctuates significantly more and can be predicted less reliably. Thus, the initial copy rate ICR should be selected in such a way that the savings are maximized on average $\bar{x}(\%)$ with a high probability $ki_{0.95}(\%)$. When using the initial copy rate of ICR = 0.2–1.0, better results $Best_{rel}$ are obtained for 58.3% ($ICR = 1.0$) to 69.7% ($ICR = 0.4$) of the considered individual experiments than with an initial copy rate of $ICR = 0.0$. If the best initial copy rate ICR_{Best} was always selected, the results $Best_{rel}$ will be better in 79.17% of the individual experiments compared to randomized solutions ($ICR = 1.0$). In contrast, the results $Best_{rel}$ will be worse in 55.2% of the individual experiments if the worst initial copy rate ICR_{Worst} was selected. The average $\bar{x}$ best saving $Best_{rel}$ is achieved with an initial copy rate $ICR = 0.6$ ($\bar{x} = 0.42\%$), and the median $\tilde{x}$ best saving $Best_{rel}$ is achieved with an initial copy rate $ICR = 0.4$ ($\tilde{x} = 0.46\%$). These values should not be regarded as too low, since, as will be examined in Section 5.4, the overall optimization potential is approximately 4%. With both $ICR = 0.4$ and $ICR = 0.6$, the confidence interval is $ki_{0.95} = \pm 0.49\%$. Thus, there is no clear recommendation for the selection of an initial copy rate ICR, but $ICR = 0.4$ or $ICR = 0.6$ should be preferred. The following experiments uses $ICR = 0.6$.

Table 2. Relative saving potential initial copy rate for generating the copied population compared to randomized population.

Statistic	0.2	0.4	0.6	0.8	1.0	0.2–1.0	*Best*	*Worst*
Ant (%)	66.6	69.7	64.6	62.5	58.3	66.4	79.17	55.2
$\bar{x}$ (%)	0.28	0.39	0.42	0.26	0.05	0.26	0.66	−0.37
$\tilde{x}$ (%)	0.37	0.46	0.44	0.41	0.32	0.40	0.58	0.22
σ (%)	2.20	2.42	2.40	2.59	3.15	2.55	2.30	3.06
$ki_{0.95}$ (%)	±0.45	±049	±0.49	±0.53	±0.64	±0.52	±0.47	±0.62

*5.4. Evaluation of the Overall Quality of The Algorithm*

To evaluate the performance of the overall concept, the objective value of the initial population is compared with the objective value of the last population of the optimization run. The maximum number of generations for deterministic and stochastic optimization is $gen = 100$ with a stop criterion of $gen = 20$ generations. The number of experiments is similar to that mentioned in Section 5.3. Since the results of the repetitions $|W|$ do not differ significantly, $|W|$ is set to $|W| = 1$ for the next experiments. With a coefficient of variation of $cv_{lognorm} = 0.9$, the optimization potential is between

0–1% in about 50% of the experiments (Figure 8). For the remaining experiments, an optimization potential between 1–5% is achieved.
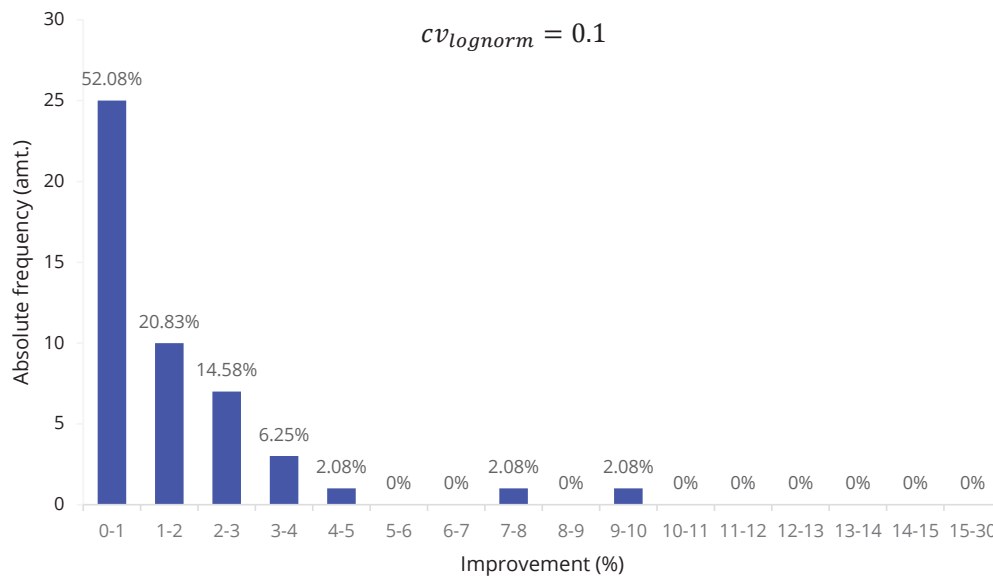
Figure 8. Performance of the genetic algorithm: comparison between best value initial population and best value last generation of optimization $cv_{lognorm} = 0.9$.

A higher optimization potential is achieved with a coefficient of variation of $cv_{lognorm} = 0.1$ (Figure 9). In about 50% of the experiments, the optimization potential is higher than 5%. It is remarkable that, in about 15% of the experiments, the optimization potential is between 15–30%. This represents that the optimization potential decreases with large stochastic influences, which represents that the actual process time deviates significantly from the expected value and that the solution is highly random. This corresponds almost to a complete lack of process knowledge so that an online optimization is probably more effective.
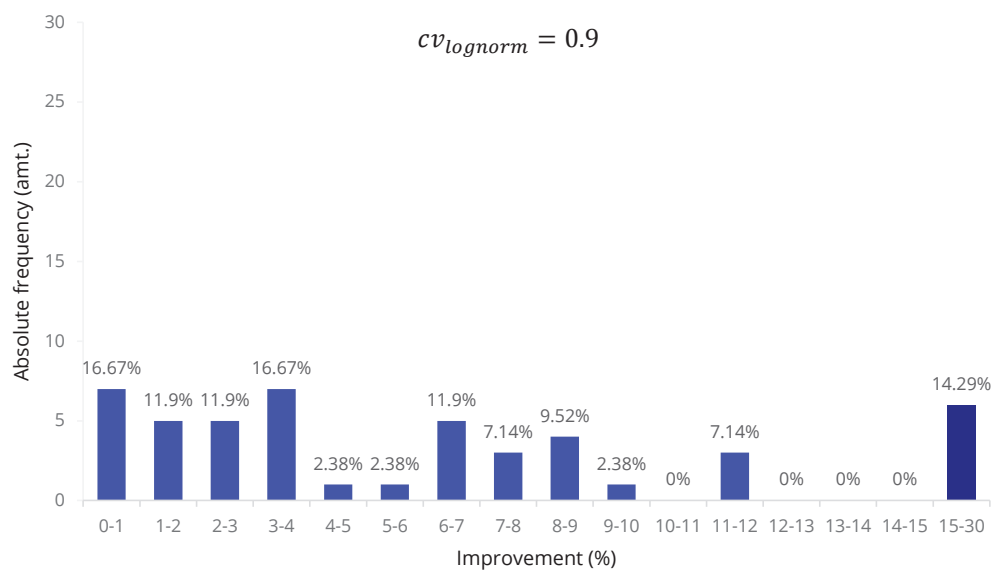

Figure 9. Performance of the genetic algorithm: comparison between best value initial population and best value last generation of optimization $cv_{lognorm} = 0.1$.

If we have a detailed look at the parameters (Figure 10), we see that there is a better improvement in general for the objective function project delay toward makespan. The model parameters have

no significant influence. The influence of the coefficient of variation of the distribution is significant. In the median, an improvement of approximately 4% occurs independently of the parameters.
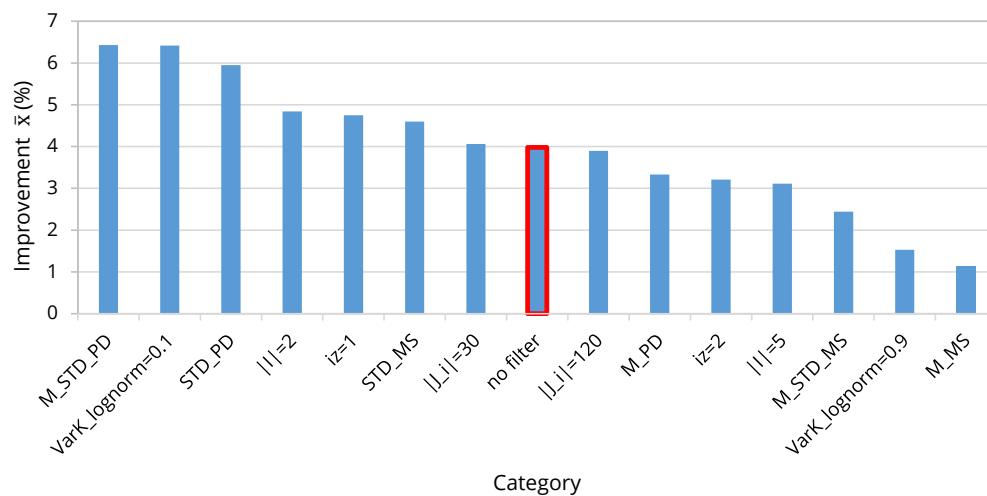
Figure 10. Performance of the proposed algorithm depending on the single parameters.

5.5. Comparison with Standard Priority Rules

For evaluation of the presented algorithm, we compare the performance of the $PR = GAWS$ with the $PR = SPR$ on 25 models M from the MPSPLIB, which essentially differ from the previously considered models by the instances. The experiments vary with respect to the coefficient of variance $cv_{lognorm}$ and the objective function f. The parameter initial copy rate ICR is still considered $ICR = 0.6$. The parameters of the GA are the same as in Section 5.4. Common SPRs serve as comparison criteria [49,50].

The presented algorithm achieves under stochastic influences without Pareto criterion in 50% better results than the considered SPR (Figure 11). The second best SPR MWRK (Most Work Remaining) achieves better results in 12% of the experiments, the third best SPR EDD (Earliest Due Date) achieves better results in 8% of the experiments, and the fourth best SPR FIFO achieves better results in 6% of the experiments. The remaining SPRs achieve the best results in 1% on average.

Looking at the three best PRs (GAWS, MWRK, and EDD) in Figure 12, it can be seen that all three PRs are represented on all ranks, with rank = 1 being the best PR and rank = 20 being the worst PR. GAWS is 95% among the 10 best PRs (sum of the relative frequency of GAWS from $rank = 1$ to $rank = 10$). SPR EDD is among the ten best SPRs in 70% of the experiments (sum of the relative frequency of EDD from $rank = 1$ to $rank = 5$). It is remarkable that the MWRK comes last ($rank = 20$) in almost 10% of the experiments.

In the following, the saving potential is quantified if the CPR GAWS ranks first (Table 3). On average, there is a saving of 3.3% compared to $rank = 2$ regarding the objective value. Compared to the average performance, there is a saving of 10.8%, and compared to the worst SPR, there is a saving of about 25%.

If CPR GAWS does not occupy rank = 1, there is an average loss compared to $rank = 1$ of -10.6% regarding the objective value (Table 4). Compared to the medium performance of the other SPRs, an average saving of 3.63% is still achieved. Comparing CPR GAWS to the worst SPR, an average saving of 18.9% is achieved. In summary, the presented CPR can achieve an average improvement over a randomly selected SPR.

For complete evaluation of the presented CPR GAWS, it is necessary to assess the fulfillment of individual project objectives according to the Pareto criterion (Table 5). For this purpose, the solutions are divided into solution fronts (Section 4.2). CPR GAWS is on average in 90% of the experiments in the first front. There are no significant deviations when considering individual parameters. The only difference shows the parameter instance *iz*. A performance analysis of the other SPRs shows that no

SPR dominates another SPR with regard to its occurrence in the first front. The SPR FIFO is often in the first front. As far as frequency is concerned, no other SPR comes close to the presented CPR GAWS. This shows the real potential of CPR GAWS.
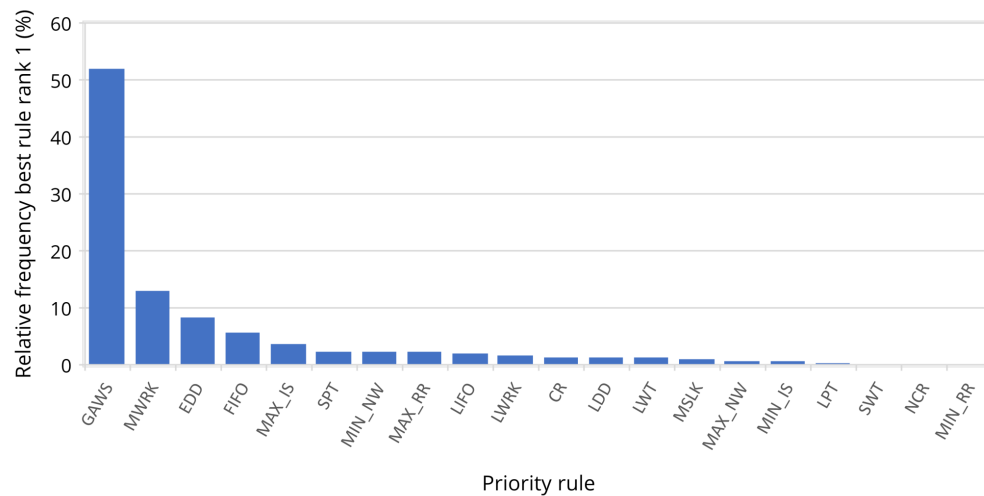
Figure 11. Performance of the proposed algorithm compared to Standard Priority Rules (SPRs): *MWRK*: Most Work Remaining; *EDD*: Earliest Due Date; *FIFO*: First In First Out; *MAXIS*: Maximum Immediate (Direct) Successors; *MAXRR*: Maximum Resource Request; *LIFO*: Last In First Out; *LWRK*: Least Work Remaining; *CR*: Critical Ration; *NCR*: None Critical Ratio; *LDD*: Least Due Date; *LWT*: Longest Wait Time; *MSLK*: Minimum Slack; *MAXNW*: Maximum Total Successors; *MINIS*: Minimum Immediate (Direct) Successors; *LPT*: Longest Processing Time; *SWT*: Shortest Wait Time (cumulative); *NCR*: None Critical Ration; *MINRR*: Minimal Resource Request.
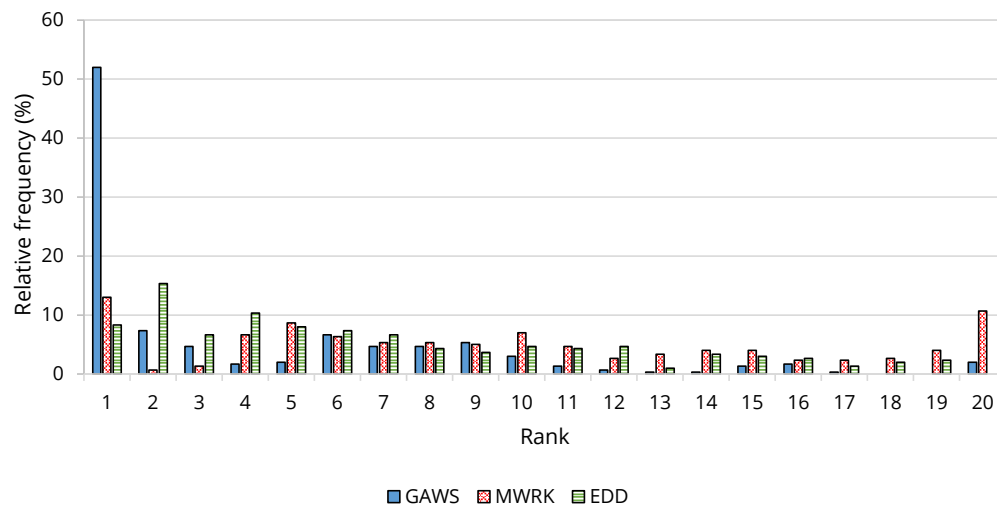
Figure 12. Comparison of the performance of the three priority rules.

Table 3. Improvement with GAWS = rank 1.

Comparing GAWS Rank = 1 to			
Statistic	**Rank 2**	**MW Rank**	**Rank 20**
$\overline{x}$ (%)	3.31	10.82	24.86
$\widetilde{x}$ (%)	2.66	8.49	17.64
σ (%)	2.88	7.56	20.65
$ki_{0.95}$ (%)	± 0.46	± 1.20	± 3.27

Table 4. Deterioration with GAWS $\neq$ Rank 1.

Comparing GAWS Rank $\neq$ 1 to			
Statistic	Rank 2	MW Rank	Rank 20
$\overline{x}$ (%)	-10.60	3.63	18.90
$\widetilde{x}$(%)	-3.22	2.91	14.68
σ (%)	19.15	5.43	15.66
$ki_{0.95}$ (%)	± 3.03	± 0.86	± 2.48

Table 5. Comparison of objective fulfillment with Pareto criterion.

Filter	GAWS = 1st Front (%)	2nd Best PR = 1st Front	2nd Best PR = 1st Front (%)
All experiments	91	FIFO	65
$cv_{lognorm} = 0.1$	95	FIFO	61
$cv_{lognorm} = 0.9$	88	FIFO	69
M_PD	94	FIFO	74
STD_PD	90	MSLK	42
$M\_PD\_STD\_PD$	92	FIFO	86
M_MS	92	FIFO	72
STD_MS	82	LWRK	42
M_MS_STD_MS	98	FIFO	88
$|\mathbf{I}| = 2$	89	FIFO	55
$|\mathbf{I}| = 5$	95	MAX_NW	88
$|\mathbf{J_i}| = 30$	95	MAX_NW	64
$\mathbf{J_i}| = 120$	89	FIFO	71
$\mathbf{iz} = 1$	77	FIFO	72
$\mathbf{iz} = 2$	100	MWRK	58

6. Conclusions and Outlook

In this paper, we present and study a 2-phase genetic algorithm for the efficient generation of project-specific composite priority rules for short-term production control of the stochastic resource-constrained multi-project scheduling problem. Computational results with our presented discrete-event simulation-framework PyScOp show that, with the same simulation effort, the proposed algorithm achieves better initial solutions compared to complete randomly generated initial solutions on average. The best initial population is based on half copied best and half randomly selected solutions. The experiments also show that, with our algorithm, the objective fulfillment compared to priority rules used in practice is much better. Our solutions are in 90% of the experiments in the Pareto front. The best priority rule FIFO is in about 65% of the experiments in the Pareto front. All experiments show that the performance of the algorithm depends on the model parameter.

We see further research in the parameter optimization of the GA to improve the generated composite priority rules. The computational effort can be further reduced if additional strategies for generating the initial population are applied. This includes generation of the start solution with known priority rules or with solutions from similar models. In general, the model-attribute-objective effects need to be further analyzed in order to set model-dependent parameters that can improve both computation time and objective values.

Abbreviations

The following abbreviations are used in this manuscript:

APS	Advanced Panning and Scheduling
CDR	Composite Dispatching Rule
CPR	Composite Priority Rule
CPPS	Cyber Physical Production System
GA	Genetic Algorithm
GAWS	Genetic Algorithm Weighted Sum
ICR	Initial Copy Rate
MES	Manufacturing Execution System
MPSPLIB	Multi-Project Scheduling Library
PPC	Project Planning and Control
PSPLIB	Project Scheduling Linrary
RCPSP	Resource Constrained Project Scheduling Problem
SH	Sequencing Heuristic
SPR	Simple Priority Rules
SRCMPSP	Stochastic Resource Constrained Multi-Project Scheduling Problem

References

1. Bischoff, J.; Taphorn, C.; Wolter, D.; Braun, N.; Fellbaum, M.; Goloverov, A.; Ludwig, S.; Hegmanns, T.; Prasse, C.; Henke, M.; et al. *Erschließen der Potenziale der Anwendung von Industrie 4.0 im Mittelstand*; BMWi: Berlin, Germany, 2015.
2. Huber, W. *Industrie 4.0 Kompakt—Wie Technologien Unsere Wirtschaft und Unsere Unternehmen Verändern*; Springer Fachmedien Wiesbaden: Wiesbaden, Germany, 2018; doi:10.1007/978-3-658-20799-1. [CrossRef]
3. Lödding, H. *Verfahren der Fertigungssteuerung: Grundlagen, Beschreibung, Konfiguration*; Springer: Berlin/Heidelberg, Germany, 2016.
4. Glonegger, M. *Berücksichtigung Menschlicher Leistungsschwankungen bei der Planung von Variantenfließmontagesystemen*; Forschungsberichte IWB; Utz, Herbert: München, Germany, 2014; Volume 292.
5. Marczinski, G. Einsatzgebiete von ERP-, APS- und MES-Lösungen. *ERP Manag.* **2008**, *2008*, 62–64.
6. Niehus, M.R. Adaptive Produktionssteuerung für Werkstattfertigungssysteme durch Fertigungsbegleitende Reihenfolgebildung. Ph.D. Thesis, Technische Universität München, München, Germany, 2016.
7. Hansmann, K.W. *Industrielles Management*; 8., völlig überarb. und erw. aufl. ed.; Oldenbourg: München, Germany, 2006.
8. Nyhuis, P.; Mayer, J.; Kuprat, T. Die Bedeutung von Industrie 4.0 als Enabler für logistische Modelle. In *Industrie 4.0*; Kersten, W., Ed.; Schriftenreihe der Hochschulgruppe für Arbeits- und Betriebsorganisation e.V. (HAB), Gito: Berlin, Germany, 2014; pp. 79–100.
9. Peßl, E. *Digitale Produktion: Studie über Status, Hemmnisse und Anforderungen österreichischer Produzierender Klein- und Mittelunternehmen und Analyse der Software-Hersteller von MES-Systemen*; Inst. Industrial Management/Industriewirtschaft FH JOANNEUM Kapfenberg: Kapfenberg, Austria, 2013.
10. Branke, J.; Nguyen, S.; Pickardt, C.W.; Zhang, M. Automated Design of Production Scheduling Heuristics: A Review. *IEEE Trans. Evol. Comput.* **2016**, *20*, 110–124. [CrossRef]
11. Hildebrandt, T.; Heger, J.; Scholz-Reiter, B. Towards improved dispatching rules for complex shop floor scenarios. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*; Pelikan, M., Branke, J., Eds.; ACM: New York, NY, USA, 2010; pp. 257–264. [CrossRef]
12. Pritsker, A.A.B.; Waiters, L.J.; Wolfe, P.M. Multiproject Scheduling with Limited Resources: A Zero-One Programming Approach. *Manag. Sci.* **1969**, *16*, 93–108. [CrossRef]
13. Lova, A.; Tormos, P. Analysis of Scheduling Schemes and Heuristic Rules Performance in Resource-Constrained Multiproject Scheduling. *Ann. Oper. Res.* **2001**, *102*, 263–286.:1010966401888. [CrossRef]
14. Confessore, G.; Giordani, S.; Rismondo, S. A market-based multi-agent system model for decentralized multi-project scheduling. *Ann. Oper. Res.* **2007**, *150*, 115–135. [CrossRef]
15. Homberger, J. A multi-agent system for the decentralized resource-constrained multi-project scheduling problem. *Int. Trans. Oper. Res.* **2007**, *14*, 565–589. [CrossRef]

16. Homberger, J. MPSPLIB: Multi Project Scheduling Problem Library. Hochschule für Technik Stuttgart. 2008. Available online: www.mpsplib.com (accessed on 17 October 2020).

17. Kolisch, R.; Sprecher, A. PSPLIB—A project scheduling problem library. *Eur. J. Oper. Res.* **1996**, *96*, 205–216. [CrossRef]

18. Blazewicz, J.; Lenstra, J.; Rinnooy Kan, A. Scheduling subject to resource constraints: Classification and complexity. *Discret. Appl. Math.* **1983**, *5*, 11–24. [CrossRef]

19. Ballestín, F. When it is worthwhile to work with the stochastic RCPSP? *J. Sched.* **2007**, *10*, 153–166. [CrossRef]

20. Kolisch, R.; Hartmann, S. Heuristic Algorithms for the Resource-Constrained Project Scheduling Problem: Classification and Computational Analysis. In *Project Scheduling: Recent Models, Algorithms and Applications*; Węglarz, J., Ed.; Springer US: Boston, MA, USA, 1999; pp. 147–178. [CrossRef]

21. März, L.; Krug, W.; Rose, O.; Weigert, G. *Simulation und Optimierung in Produktion und Logistik*; Springer: Berlin, Germany, 2011; Volume 1. [CrossRef]

22. Herroelen, W. Project Scheduling—Theory and Practice. *Prod. Oper. Manag.* **2005**, *14*, 413–432. [CrossRef]

23. Schade, K. Stochastische Optimierung. In *Stochastische Optimierung*; Schade, K., Ed.; Stochastic Programming, Vieweg+Teubner: Wiesbaden, Germany, 2012; pp. 45–72. [CrossRef]

24. Ashtiani, B.; Leus, R.; Aryanezhad, M.B. A Novel Class of Scheduling Policies for the Stochastic Resource-Constrained Project Scheduling Problem. *SSRN Electron. J.* **2008**. [CrossRef]

25. Stork, F. Stochastic Resource-Constrained Project Scheduling. Ph.D. Thesis, Technische Universitat Berlin, Berlin, Germany, 2001; doi:10.14279/DEPOSITONCE-398. [CrossRef]

26. Ashtiani, B.; Leus, R.; Aryanezhad, M.B. New competitive results for the stochastic resource-constrained project scheduling problem: Exploring the benefits of pre-processing. *J. Sched.* **2011**, *14*, 157–171. [CrossRef]

27. Deb, K.; Jain, H. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. *IEEE Trans. Evol. Comput.* **2014**, *18*, 577–601. [CrossRef]

28. Kück, M.; Broda, E.; Freitag, M.; Hildebrandt, T.; Frazzon, E.M. Towards adaptive simulation-based optimization to select individual dispatching rules for production control. In Proceedings of the 2017 Winter Simulation Conference (WSC), Las Vegas, NV, USA, 3–6 December 2017; pp. 3852–3863. [CrossRef]

29. Grundstein, S.; Freitag, M.; Scholz-Reiter, B. A new method for autonomous control of complex job shops—Integrating order release, sequencing and capacity control to meet due dates. *J. Manuf. Syst.* **2017**, *42*, 11–28. [CrossRef]

30. Fink, A.; Homberger, J. Decentralized Multi-Project Scheduling. In *Handbook on Project Management and Scheduling*; Schwindt, C., Zimmermann, J., Eds.; Springer International Publishing: Cham, Switzerland, 2015; Volume 1, pp. 685–706.

31. Chand, S.; Singh, H.; Ray, T. Evolving heuristics for the resource constrained project scheduling problem with dynamic resource disruptions. *Swarm Evol. Comput.* **2019**, *44*, 897–912. [CrossRef]

32. Hildebrandt, T.; Freitag, M. Bessere Prioritätsregeln für komplexe Produktionssysteme mittels multi-kriterieller simulationsbasierter Optimierung. In *Simulation in Production and Logistics 2015*; Rabe, M., Clausen, U., Eds.; Fraunhofer Verlag: Stuttgart, Germany, 2015; pp. 309–318.

33. Branke, J.; Hildebrandt, T.; Scholz-Reiter, B. Hyper-heuristic Evolution of Dispatching Rules: A Comparison of Rule Representations. *Evol. Comput.* **2015**, *23*, 249–277. [CrossRef]

34. Nguyen, S.; Zhang, M.; Tan, K.C. Surrogate-Assisted Genetic Programming With Simplified Models for Automated Design of Dispatching Rules. *IEEE Trans. Cybern.* **2017**, *47*, 2951–2965. [CrossRef]

35. Jorapur, V.S.; Puranik, V.S.; Deshpande, A.S.; Sharma, M. A Promising Initial Population Based Genetic Algorithm for Job Shop Scheduling Problem. *J. Softw. Eng. Appl.* **2016**, *9*, 208–214. [CrossRef]

36. Werner, F. A Survey of Genetic Algorithms for Shop Scheduling Problems. In *Heuristics: Theory and Application*; Siar, P., Ed.; Nova Science Publishers: Hauppauge, NY, USA, 2013; pp. 161–222.

37. Nguyen, S.; Zhang, M.; Johnston, M.; Tan, K.C. Automatic Design of Scheduling Policies for Dynamic Multi-objective Job Shop Scheduling via Cooperative Coevolution Genetic Programming. *IEEE Trans. Evol. Comput.* **2014**, *18*, 193–208. [CrossRef]

38. Nguyen, S.; Zhang, M.; Johnston, M.; Tan, K.C. A Computational Study of Representations in Genetic Programming to Evolve Dispatching Rules for the Job Shop Scheduling Problem. *IEEE Trans. Evol. Comput.* **2013**, *17*, 621–639. [CrossRef]

39. Omar, M.; Baharum, A.; Hasan, Y. A job-shop scheduling problem (JSSP) using genetic algorithm (GA). In Proceedings of the 2nd im TG T Regional Conference, Penang, Malaysia, 13–15 June 2006.

40. Goldberg, D.E. *Genetic Algorithm in Search, Optimization, and Machine Learning*; Addison-Wesley: Reading, MA, USA, 1989; Volume XIII.

41. Python Software Foundation. Python. 2019. Available online: www.python.org (accessed on 17 October 2020).

42. Kühn, M. PyScOp. 2019. Available online: https://tlscm.mw.tu-dresden.de/scm/git/PyScOp_2.0 (accessed on 17 October 2020).

43. Schmidt, T.; Kühn, M.; Genßler, P.R. Design of Project-oriented Calculation Models for Job Priorities by Using a Customized Genetic Algorithm. In *Simulation in Produktion und Logistik 2017*; Wenzel, S., Peter, T., Eds.; Kassel University Press: Kassel, Germany, 2017; pp. 99–108.

44. Law, A.M.; Kelton, W.D. *Simulation Modeling and Analysis*, 3rd ed.; internat. ed., [nachdr.] ed.; McGraw-Hill Series in Industrial Engineering und Management Science; McGraw-Hill: Boston, MA, USA, 2000.

45. Sexton, R.S.; Dorsey, R.E.; Johnson, J.D. Optimization of neural networks: A comparative analysis of the genetic algorithm and simulated annealing. *Eur. J. Oper. Res.* **1999**, *114*, 589–601. [CrossRef]

46. Das, I.; Dennis, J.E. Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems. *SIAM J. Optim.* **1998**, *8*, 631–657. [CrossRef]

47. Freitag, M.; Hildebrandt, T. Automatic design of scheduling rules for complex manufacturing systems by multi-objective simulation-based optimization. *CIRP Ann.* **2016**, *65*, 433–436. [CrossRef]

48. Hildebrandt, T.; Branke, J. On Using Surrogates with Genetic Programming. *Evol. Comput.* **2015**, *23*, 343–367. [CrossRef]

49. Haupt, R. A survey of priority rule-based scheduling. *Spektrum* **1989**, *11*, 3–16. [CrossRef]

50. Vanhoucke, M. *Integrated Project Management Sourcebook*; Springer International Publishing: Cham, Switzerland, 2016; doi:10.1007/978-3-319-27373-0. [CrossRef]

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.