DOI: 10.22032/dbt.57889

Simulation in Produktion und Logistik 2023
Bergmann, Feldkamp, Souren und Straßburger (Hrsg.)
Universitätsverlag Ilmenau, Ilmenau 2023
DOI (Tagungsband): 10.22032/dbt.57476

Automated Generation and Simulation of Hyper Heuristics for Stochastic Multi-Mode Multi-Project Resource-Constrained Project Scheduling Problems with Setup Times

Automatisierte Generierung und Simulation von Hyper-Heuristiken für stochastische Multi-Modus-Multi-Projekt-ressourcenbeschränkte Projekt- und Systemplanungsprobleme mit Umrüstzeiten

Rico Zöllner, Mathias Kühn, Konrad Handrich, Thorsten Schmidt, TU Dresden, Dresden (Germany), rico.zoellner@tu-dresden.de, mathias.kuehn@tu-dresden.de, konrad.handrich@tu-dresden.de, thorsten.schmidt@tu-dresden.de

Abstract: A simulation framework is presented which covers both generation and simulation of production planning and control problems which include transfer times and stochastic influences and therefore extend classical multi-mode multi-project RCPSPs. This allows for systematic and in-depth investigations of the quality and the behaviour of heuristics. In addition, the automated design of heuristics based on Boolean operators applied to relations of problem specific quantities leads on average to better results than a manual selection and adjustment of heuristic strategies.

1 Introduction

With the advent of Industry 4.0 and the continuous establishment of the associated innovative tools and techniques within the realm of AI, ML algorithms and human-machine-interaction, technical possibilities to increase productivity and efficiency previously lain beyond the horizon are opening up in daily production routines. At the same time, the targeted and effective application of the new techniques poses – due to their complexity and diversity – challenges for users in practice on the one hand and academic researchers on the other hand. Both tendencies show up impressively in the field of production planning and control (PPC). Recent developments in production tend towards shorter product life cycles, smaller batch sizes and distinct individual customer requirements, which entails widely spread product ranges. In addition, interdependencies and the influence of imprecise or missing process parameters and the relevance of the disturbance variables are increasing. To deal with the above tasks, hyper heuristics emerged as a promising option because finding the global optimum with justifiable efforts for PPC problems remains illusory even with improved computational power due to the size and the NP-hardness of such problems (see (Artigues et al., 2013; Snauwaert and Vanhoucke, 2023; Hartmann and Briskorn, 2022) for surveys on resource-constrained project scheduling problems (PSP) and (Đurasević and Jakobović, 2023; Guo et al., 2022; Zhang et al., 2021; Branke et al., 2016; Burke et al., 2013; Luo et al., 2022) for surveys focussing on algorithms). Moreover, even the appropriate application of heuristics causes serious difficulties because the number and the value range of the parameters to be taken into account as well as the diversity of the existing algorithms exceed what is intuitively comprehensible and there is still a lack of systematic, theoretically sound selection criteria (Burke et al., 2013).

Consequently, hyper heuristics address a central question of a hybrid PPC: How can AI (especially ML algorithms) be utilized to create heuristics for solving practically relevant PSPs in an automated way but also including human expertise provided by production environment?

Simulations play an important role in answering this research question in two respects (Kuck et al., 2016): Firstly, simulation results form the data basis (training data and test instances) on which ML algorithms build up hyper heuristics. Secondly, simple low-effort simulations can be part of the heuristic itself, i. e. methods which simulate a selected part of the problem (tree) in the ongoing production process can also be considered in order to make an updated heuristic (or even human) decision based on those results, cf. (Zhang et al. 2020). Such approaches are useful if the PPC problem is stochastically influenced.

This paper is organized as follows: Section 2 defines the scope of the considered problem class, namely stochastic multi-mode multi-project resource-constrained project scheduling problems with transfer times (MMMPRCPSPTT). The structure of hyper heuristics based on Boolean operations is explained in Section 3. The last two sections turn the focus from rather theoretical modelling to the practical implementation of the simulation framework (Section 4) and numerical results (Section 5).

2 Stochastic Multi-Mode Multi-Project Resource-Constrained Project Scheduling Problems with Setup Times

During the last years, multi-mode multi-project resource-constrained project scheduling problems (MMMPRCPSP) drew a great research interest because they arise in many logistic contexts (Issa and Tu, 2020; Gonçalves et al., 2008; Singh, 2014; Zhu et al., 2021; Pritsker et al., 1969), e. g. an SME or any customized manufacturer offering a certain range of products (projects) manufactured in a factory hall (containing the limited resources which can be both machines or employees) has some freedom in how the steps of creating a product can be executed (modes) within the working plan. Some practical relevant features required extensions, namely:

1. Stochastic influences: The durations of working processes deviate from time to time which means that job durations should rather be modelled by random variables of a certain distribution (see below) than by fixed values.

2. Transfer times: Between two different jobs, machines have to be retooled or reconfigured which causes additional time. Therefore, for each combination of two jobs a not necessarily deterministic transfer time increment is introduced.

After giving this intuitive outline, let us now turn to a more formal definition of a stochastic MMMPRCPSPTT.

- resources: $r = 1 \dots R$ with C_r independent capacity units, C_r is a discrete random variable of a given distribution
- projects: $p = 1 \dots P$ consisting of J_p jobs respectively

For each p fixed, we have

- jobs: $j = 1 \dots J_p$ with available modes $m = 1 \dots M_{pj}$ and the respective durations D_{pjm} as discrete random variables and required resource slots K_{pjmr} while being in process
- successor matrix $S_p \in \{0,1\}^{Jp \times Jp}$ defined by:

$$S_{pj_1j_2} := \begin{cases} 1, & \text{if job } j_2 \text{follows after job } j_1 \\ 0, & \text{otherwise} \end{cases}$$

- setup times $T_{p_1\hat{\jmath}p_2\tilde{\jmath}} :=$ time increment, if job $\hat{\jmath}$ of project p_1 is directly followed by job $\tilde{\jmath}$ of project p_2 on resource r (considered as discrete random variables as well)

In summary, a stochastic MMMPRCPSPTT is characterized by the tuple $\{R, P, (J_p), (M_{pj}), (D_{pjm}), (S_{pj_1j_2}), (K_{pjmr}), (T_{p_1\hat{\jmath}p_2\tilde{\jmath}})\}$ with $1 \leq p, p_1, p_2 \leq P$, $1 \leq r \leq R$, $1 \leq j, j_1 j_2 \leq J_p$, $1 \leq m \leq M_{pj}$, $1 \leq \hat{\jmath} \leq J_{p_1}$, $1 \leq \tilde{\jmath} \leq J_{p_2}$. Please note the following remarks:

- Each job is only started once and simultaneously in exactly one mode on all necessary resources and must not be paused.
- The slots of a resource can be occupied and switched independently.

Since we refrain here from a precise mathematical modelling, we will not further elaborate formal details and refer the interested reader to the literature (e. g. (Pritsker et al., 1969; Lova et al., 2006; Krüger and Scholl, 2009; Liu et al., 2023; Neumann, 2003; Artigues et al., 2003)). Comparing to classical MMMPRCPSPs, the defined stochastic MMMPRCPSPTT come closer to what we call reality but demand at the same time more efforts to find at least acceptable solutions within at least acceptable time horizons. Here, heuristics enter the game.

3 Hyper Heuristics Based on Boolean Operations

In our approach, the creation of hyper heuristics serving as priority rules is based on the choice and evaluation of quantities characterizing the problem and the process (see below for the complete list)

Global attributes:

`number_executable_modes`	`number_executable_projects`
`number_exectuable_jobs`	`foreach_setup_needed`

Process attributes:

DirectSuccessorCount	
DirectSuccessorSumMaxProcessingTime	
DirectSuccessorSumMinProcessingTime	
ProjectStartTime	ProjectFinishTime
Resources	Duration
SetupNeeded	SetupTime
TotalSuccessorCount	
TotalSuccessorSumMaxProcessingTime	
TotalSuccessorSumMinProcessingTime	
WaitingJobCount	WaitingTime

where for each of the process attributes the sum, the mean, the median, the span length and the standard deviation can be computed (regarding the current queue).

These are nested via combination of Boolean operators such as conjunction or disjunction. A priority rule is built up as a tree of elementary rules of the form:

```
if: quantity1 ~ parameter1
then: rule1
else: rule2
```

where $\mathrm{rule}_{1/2} \in \{\mathrm{FIFO}, \mathrm{LIFO}, \mathrm{MWKR}, \mathrm{SPT}\}$ (first-in-first-out, last-in-first-out, most-work-remaining, shortest-processing-time) and ~ symbolizes a binary relation, i. e. $\sim \in \{\leq, <, >, \geq\}$. More formally, the set of all priority rules Z can be constructed recursively by setting FIFO, LIFO, MWKR, SPT $\in Z$ and by claiming that the rule defined by

```
if: condition
then: rule1
else: rule2
```

is element of Z for all $\mathrm{rule}_{1/2} \in Z$ and all valid conditions of the form

$$(\mathrm{quantity}_1 \sim \mathrm{parameter}_1)\ \mathrm{o}_1\ (\mathrm{quantity}_2 \sim \mathrm{parameter}_2)\ \mathrm{o}_2\ \ldots$$

where o_i symbolize Boolean operations. In summary, a heuristic depends on its tree structure, on the chosen set of quantities, the respective parameter values, the respective order relation and the chosen Boolean operations which yields such a variety that ML algorithms become necessary to handle it.

4 Simulation Framework

In this section, we present our simulation framework following the 'life cycle' of an optimization process (see Fig. 1 for a scheme).

Step 1: Generation of project templates

For each of the P projects, we use a classical Kolisch programme (Kolisch, 1996; Kolisch and Sprecher, 1997) to create a deterministic MMRCPSP.

Step 2: Extension process

We have developed an extension process both to incorporate random variables and setup times as well as to unify all MMRCPSPs of step 1 to a single MMMPRCPSPTT. In more detail, a random number generator scatters logarithmically normal distributed random numbers around the deterministic values taken as expectation values of each MMRCPSP and the standard deviations are evaluated via a global, manual adjustment of the coefficient of variation (CV). The setup times are generated analogously where a uniform expectation value and a uniform standard deviation enter by manual input. The unification means essentially unifying the working plans ($S_{pj_1j_2}$), assigning the resources and reevaluation their capacities to ensure principal solvability. Note that the extension process as well allows the generation of fully or partially deterministic MMMPRCPSPTT.

Step 3: Simulation

A given heuristic according to Section 3 (encoded e. g. in a json file) serves together with a stochastic MMMPRCPSPTT according to Section 2 as input for the Java-based simulation framework, whose core is implemented using a DESMO-J framework for discrete-event modelling and simulation (Lechler and Page, 1999).

Each realization of the stochastic problem instance is executed independently. The simulator comprises two events: 1) start of the first job, which initializes the simulation, and 2) start of the successor jobs, fired each time a running job is finished. The new jobs are selected from the list of currently available jobs and resources, according the given heuristic. As soon as all the jobs are executed, the framework returns the value of the specified objective functions, e. g. cycle time, project delay or make span.

Step 4: Optimization

We envisage a fast simulation to be potentially applied in real-time scenarios, and we use a standard genetic algorithm (Wilhelmstötter, 2023) which starts with a randomly generated initial population.

In Section 5 we show some facets selected from steps 2, 3 and 4. As an extra aspect intended to further populate the topic we are going to establish a library of problems (analogue to MPSPLib (Homberger, 2023)) providing access to stochastic MMMPRCPSPTT files (incl. search function and information about each file), allowing for submission of solutions (incl. visualization of the solution (see Fig. 2), check for correctness and ranking of all solutions).

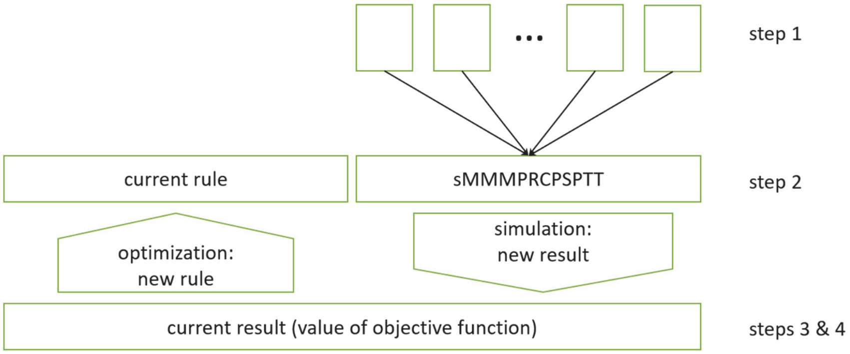

Figure 1: Scheme of simulation framework, the problem creation (steps 1 & 2) followed by an iterative optimization process (steps 3 & 4), where simulation and optimization alternate

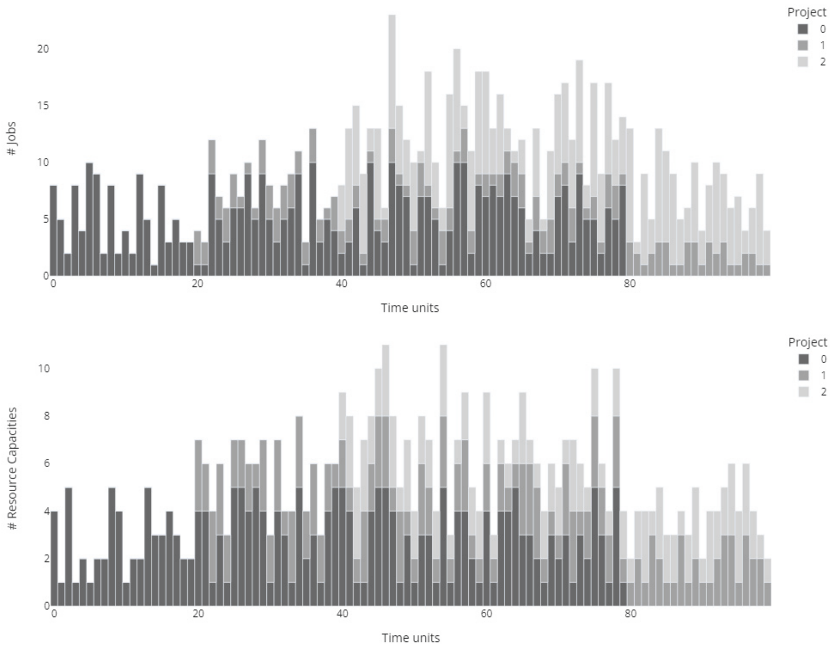

Figure 2: Visualization of a solution by exhibiting the project processing (upper panel) and the resource occupation (lower panel).

5 Selected Numerical Results

The simulation uncovers interesting relations between FIFO, LIFO, MWKR and SPT applied to a large sample of stochastic MMMPRCPSPTT: the cycle times depend nearly linearly on each other (see Fig. 4) which allows for estimating the cycle time by another reducing the effort of the simulations.

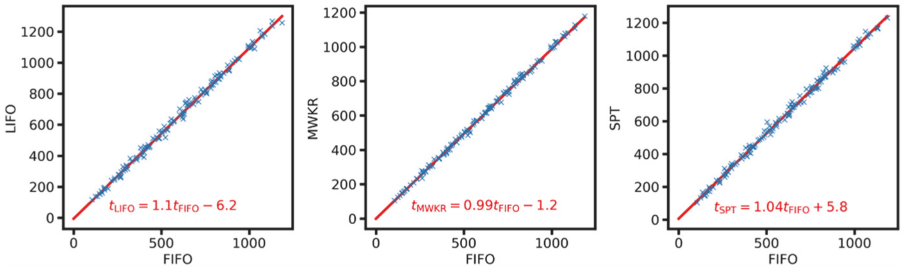

Figure 3: Cycle time of LIFO (left), MWKR (middle) and SPT (right) as a function of the FIFO cycle time with the respective linear best approximation displayed as thin line. The underlying sample covers 140 deterministic MMMPRCPSPTT with J=30…90 jobs, R=4…10 resources and P=1…200 projects.

Considering stochastic MMMPRCPSPs means facing the new quality that the objective functions are random variables as well and therefore display a distributional behaviour which could be subject to further investigations. As a glimpse, Fig. 5 exhibits the distribution if an elementary priority rule is applied. Fig. 6 shows by comparing coefficients of variation that the values of the cycle time as objective function spread to a lesser extent than the input variables.

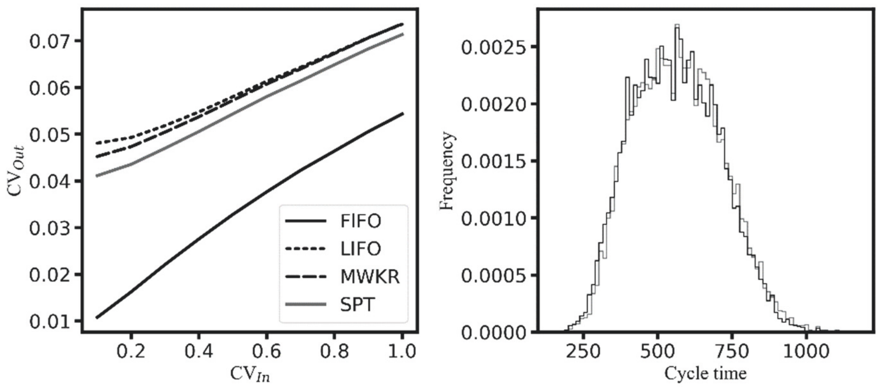

Figure 4:. Left: CV of the cycle time (CVOut) as a function of the CVIn of all durations and capacities (see Section 2). The fact CVOut ≪ CVIn underlines that the values of the objective function are more concentrated than the input values. Right: Distribution of the cycle time achieved using one of the elementary rules for 10000 realizations of one stochastic MMMPRCPSP with 1 project, 10 jobs and 4 resources. For enhanced visibility only the distributions of FIFO and SPT are displayed.

Let us consider again the cycle time for the priority rule

```
if: number_executable_modes ≤ W or mean_duration ≤ D
then: MWKR
else: FIFO
```

where the parameters W and D serve as target variables of the optimization.

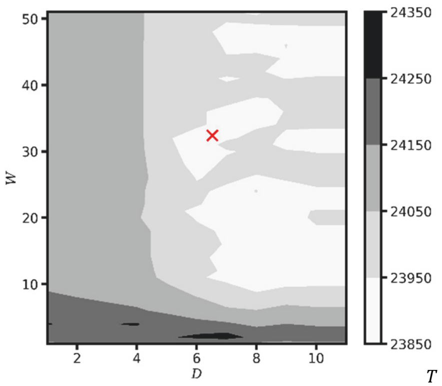

Figure 6: Contour plot of the cycle time T as a function of the parameters D and W of the priority rule given above. The minimum of T_{min} = 23894 marks the red cross. The horizontal and vertical stripes were found to be characteristic in systematic investigations and represent areas in which no minimum is to be expected, so that these can be excluded from the search for optimal parameters.

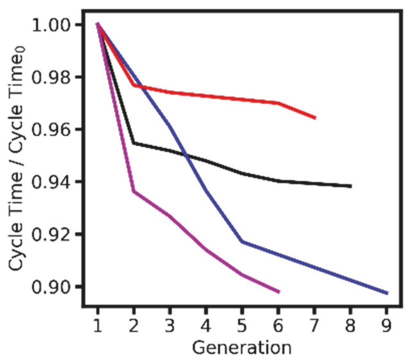

Figure 7: Evolution of the scaled cycle time (mean) throughout the application of the genetic algorithm for four selected stochastic MMMPRCSPSTT (distinguished by coloured graphs). Although the number of generations is rather small and the total number of individuals was limited by 2000, there are improvements from about 3 % to about 10 %.

Clearly, the optimization is not only limited to varying the parameters of a fixed priority rule (as above) but also includes the variation of all components. Fig. 7

exhibits the evolution of the cycle time as objective function over the generation for some selected problem instances.

6 Summary and Outlook

As shown above, a comprehensive simulation tool allows for manoeuvring through the field of stochastic MMMPRCPSPTT, delivers new insights and uncovers previously hidden relations. However, this is to be considered as a starting point for further investigations, e. g. systematic studies to elaborate probability distributions of objective functions. Another important avenue leads to practical applications in terms of real-time simulations and human biases.

Acknowledgement

The work of MK and RZ is supported by DFG project 418727532. The authors gratefully thank Dmytro Pukhkaiev and Richard Lux for their comments and the valuable contributions to the project.

References

Artigues, C.; Demassey, S.; Néron, E.: Resource-Constrained Project Scheduling: Models, Algorithms, Extensions and Applications. New York, NY: John Wiley & Sons 2013.

Artigues, C.; Michelon, P.; Reusser, S.: Insertion techniques for static and dynamic resource-constrained project scheduling. European Journal of Operational Research 149 (2003) 2, pp. 249–267.

Branke, J.; Nguyen, S.; Pickardt, C.W.; Zhang, M.: Automated Design of Production Scheduling Heuristics: A Review. IEEE Transactions on Evolutionary Computation 20 (2016) 1, pp. 110–124.

Burke, E.K.; Gendreau, M.; Hyde, M.; Kendall, G.; Ochoa, G.; Özcan, E.; Qu, R.: Hyper-heuristics: a survey of the state of the art. Journal of the Operational Research Society 64 (2013) 12, pp. 1695–1724.

Đurasević, M.; Jakobović, D.: Heuristic and metaheuristic methods for the parallel unrelated machines scheduling problem: a survey. Artificial Intelligence Review 56 (2023) 4, pp. 3181–3289.

Gonçalves, J.F.; Mendes, J.; Resende, M.: A genetic algorithm for the resource constrained multi-project scheduling problem. European Journal of Operational Research 189 (2008) 3, pp. 1171–1190.

Guo, H.; Liu, J.; Zhuang, C.: Automatic design for shop scheduling strategies based on hyper-heuristics: A systematic review. Advanced Engineering Informatics 54 (2022), pp. 101756.

Hartmann, S.; Briskorn, D.: An updated survey of variants and extensions of the resource-constrained project scheduling problem. European Journal of Operational Research 297 (2022) 1, pp. 1–14.

Homberger, J., 2023: mpsplib.com | MULTI PROJECT SCHEDULING PROBLEM LIBRARY |. Online available at: http://mpsplib.com/, last updated 07.05.2023, last checked 07.05.2023.

Issa, S.; Tu, Y.: A survey in the resource-constrained project and multi-project scheduling problems. Journal of Project Management 5 (2020) 2, pp. 117–138.

Kolisch, R.: Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation. European Journal of Operational Research 90 (1996) 2, pp. 320–333.

Kolisch, R.; Sprecher, A.: PSPLIB - A project scheduling problem library. European Journal of Operational Research 96 (1997) 1, pp. 205–216.

Krüger, D.; Scholl, A.: A heuristic solution framework for the resource constrained (multi-)project scheduling problem with sequence-dependent transfer times. European Journal of Operational Research 197 (2009) 2, pp. 492–508.

Kuck, M.; Ehm, J.; Hildebrandt, T.; Freitag, M.; Frazzon, E.M.: Potential of data-driven simulation-based optimization for adaptive scheduling and control of dynamic manufacturing systems. In: 2016 Winter Simulation Conference (WSC), 2016,

Lechler, T.; Page, B.: DESMO-J: An object oriented discrete simulation framework in Java Proceedings of the 11th European Simulation Symposium (1999), pp. 46–50.

Liu, Y.; Jin, S.; Zhou, J.; Hu, Q.: A branch-and-bound algorithm for the unit-capacity resource constrained project scheduling problem with transfer times. Computers & Operations Research 151 (2023), pp. 106097.

Lova, A.; Tormos, P.; Barber, F.: Multi-mode resource constrained project scheduling: scheduling schemes, priority rules and mode selection rules. Inteligencia Artificial 10 (2006) 30.

Luo, J.; Vanhoucke, M.; Coelho, J.; Guo, W.: An efficient genetic programming approach to design priority rules for resource-constrained project scheduling problem. Expert Systems with Applications 198 (2022), pp. 116753.

Neumann, K.: Project scheduling with changeover times—Modelling and applications 2003.

Pritsker, A.A.; Waiters, L.J.; Wolfe, P.M.: Multiproject Scheduling with Limited Resources: A Zero-One Programming Approach. Management Science 16 (1969) 1, pp. 93–108.

Singh, A.: Resource Constrained Multi-project Scheduling with Priority Rules & Analytic Hierarchy Process. Procedia Engineering 69 (2014), pp. 725–734.

Snauwaert, J.; Vanhoucke, M.: A classification and new benchmark instances for the multi-skilled resource-constrained project scheduling problem. European Journal of Operational Research 307 (2023) 1, pp. 1–19.

Wilhelmstötter, F., 2023: Jenetics: Java Genetic Algorithm Library. Online available at: https://jenetics.io/, last updated 03.05.2023, last checked 08.05.2023.

Zhang, F.; Mei, Y.; Nguyen, S.; Zhang, M.: Guided Subtree Selection for Genetic Operators in Genetic Programming for Dynamic Flexible Job Shop Scheduling. In: Genetic Programming, (2020), pp. 262–278.

Zhang, F.; Mei, Y.; Nguyen, S.; Zhang, M.: Programming for Production Scheduling (2021).

Zhu, L.; Lin, J.; Li, Y.-Y.; Wang, Z.-J.: A decomposition-based multi-objective genetic programming hyper-heuristic approach for the multi-skill resource constrained project scheduling problem. Knowledge-Based Systems 225 (2021), pp. 107099.