An Impact of Model Parameter Uncertainty on Scheduling Algorithms

Radosław Rudek[1], Agnieszka Rudek[2], Andrzej Kozik[3] and Piotr Skworcow[4]
[1]*Institute of Business Informatics, Wrocław University of Economics, Wrocław, Poland*
[2]*Department of Systems and Computer Networks, Wrocław University of Technology, Wrocław, Poland*
[3]*Institute of Computer Engineering, Control and Robotics, Wrocław University of Technology, Wrocław, Poland*
[4]*De Montfort University, Water Software Systems, Leicester, U.K.*

Keywords: Scheduling, Flowshop, Learning, Heuristic, Robustness.

Abstract: This short paper presents a preliminary analysis of the impact of model parameter uncertainty on the accuracy of solution algorithms for the scheduling problems with the learning effect. We consider the maximum completion time minimization flowshop problem with job processing times described by the power functions dependent on the number of processed jobs. To solve the considered scheduling problem we propose heuristic (NEH based) and metaheuristic (simulated annealing) algorithms. The numerical experiments show that NEH and simulated annealing are robust for this problem with respect to model parameter uncertainty.

1 INTRODUCTION

Classical flowshop scheduling problems are perceived to be more interesting in a theoretical context than as a practical research (Gupta and Stafford, 2006). It follows from observations that algorithms constructed on the basis of the classical models usually provide unsatisfactory (unstable) solutions for real-life flowshop problems, since these models do not take into consideration additional factors such as the learning effect that is significant in practice (Biskup, 2008), (Lee and Wu, 2004), (Rudek, 2011).

A schedule for a real-life problem (e.g., in manufacturing or computer systems) is calculated on the basis of a model and values of its parameters. Due to the possible differences between estimated and real values of problem parameters (e.g., shape of the learning curve, job processing times), the algorithms that are efficient for the modelled problem do not have to be accurate for the real problem. Therefore, it is crucial to evaluate how values of parameters (uncertainty) affect the quality of solutions provided by such algorithms, thereby determine their usefulness.

Thus, in this paper, we will analyse the impact of values of parameters on the accuracy of solution algorithms for the scheduling problems with the learning effect. In particular, we will consider the maximum completion time minimization flowshop problem with job processing times described by the power functions dependent on the number of processed jobs.

This paper is organized as follows. Next section contains the problem formulation. Approximation algorithms with the analysis of their efficiency are given subsequently. The last section concludes the paper.

2 PROBLEM FORMULATION

There are given a set $J=\{1,\ldots,n\}$ of n jobs and m machines, namely $M=\{M_1,\ldots,M_m\}$. Each job j consists of a set $O=\{O_{1,j},\ldots,O_{m,j}\}$ of m operations. Each operation $O_{z,j}$ has to be processed on machine M_z ($z=1,\ldots,m$). Moreover operation $O_{z+1,j}$ may start only if $O_{z,j}$ is completed. It is assumed that machines have to process jobs in the same order, i.e., a permutation flowshop, and each machine can process one operation at a time. There are no precedence constraints between jobs, operations are non-preemptive and are available for processing at time 0 on M_1. Further, instead of operation $O_{z,j}$, we say job j on machine M_z.

Due to the learning effect the processing time $\tilde{p}_j^{(z)}(v)$ of job j processed as the vth in a sequence on machine M_z is described by a non-increasing positive function dependent on the number of previously processed operations $(v-1)$, i.e., on its position v in a sequence. The function $\tilde{p}_j^{(z)}(v)$ of the job processing time that models the learning effect is called the learning curve. Moreover, each job j is characterized by its normal processing $\tilde{p}_j^{(z)}$ time on machine M_z that is de-

Rudek R., Rudek A., Kozik A. and Skworcow P..
An Impact of Model Parameter Uncertainty on Scheduling Algorithms.
DOI: 10.5220/0004117603530356
In *Proceedings of the 14th International Conference on Enterprise Information Systems* (ICEIS-2012), pages 353-356
ISBN: 978-989-8565-10-5

fined as the time required to perform a job if the machine is not affected by learning, i.e., $p_j^{(z)} \triangleq \tilde{p}_j^{(z)}(1)$.

Following (Mosheiov and Sidney, 2003), in this paper, we focus on a problem, where the processing time of job j processed as the vth on machine M_i is described by:

$$\tilde{p}_j^{(z)}(v) = p_j^{(z)} v^{\alpha_j^{(z)}}, \quad (1)$$

where $p_j^{(z)}$ and $\alpha_j^{(z)}$ are the normal processing time and the learning index, respectively, of job j on machine M_z. Moreover, we will analyse the problem with the special cases of (1), where $\alpha_j^{(z)} = \alpha$ for $j = 1, \ldots, n$ and $z = 1, \ldots, m$.

For the m-machine *permutation* flowshop problems the schedule of jobs on the machines can be unambiguously defined by their sequence (permutation). Let $\pi = \langle \pi(1), ..., \pi(i), ..., \pi(n) \rangle$ denote the sequence (permutation) of the n jobs where $\pi(i)$ is the job in position i of π. Also, let Π be the set of all job permutations. Thus, for each job $\pi(i)$, i.e., scheduled in the ith position in π, we can determine its completion time $C_{\pi(i)}^{(z)}$ on machine M_z as follows:

$$C_{\pi(i)}^{(z)} = \max\left\{C_{\pi(i)}^{(z-1)}, C_{\pi(i-1)}^{(z)}\right\} + \tilde{p}_{\pi(i)}^{(z)}(i), \quad (2)$$

where $C_{\pi(1)}^{(0)} = C_{\pi(0)}^{(z)} = 0$ for $z = 1, \ldots, m$ and $C_{\pi(i)}^{(1)} = \sum_{l=1}^{i} \tilde{p}_{\pi(l)}^{(1)}(l)$ is the completion time of a job placed in position i in the permutation π on M_1. On this basis, the maximum completion time (makespan) for the given π can be defined as $C_{\max}(\pi) = C_{\pi(n)}^{(m)}$.

The objective is to find such a schedule π^* of jobs on the machines that minimizes the maximum completion time (makespan): $\pi^* \triangleq \arg\min_{\pi \in \Pi} \left\{C_{\max}(\pi)\right\}$. For convenience, the problem according to the three field notation scheme $X \mid Y \mid Z$ will be denoted as $Fm|\tilde{p}_j(v) = p_j v^{\alpha_j}|C_{\max}$ and its special case ($\alpha_j^{(z)} = \alpha$) as $Fm|\tilde{p}_j(v) = p_j v^{\alpha}|C_{\max}$.

3 ALGORITHMS

In this section, we will briefly describe the algorithms that are analysed in the further part of this paper. Namely, we present the extensive search algorithm (ESA), the random schedule algorithm (RND), the shortest processing time (SPT) rule, NEH (Nawaz et al., 1983) and simulated annealing (SA) (Kirkpatrick et al., 1983). Note that the problem $Fm|\tilde{p}_j(v) = p_j v^{\alpha_j}|C_{\max}$ is strongly NP-hard even without the learning effect for $m \geq 3$, and it seems to be strongly NP-hard for $m = 2$ with the learning effect.

The extensive search algorithm (ESA) is an exact method that searches the total solution space, which size is $O(n!)$.

The random schedule algorithm (RND) provides a random schedule (permutation) as a solution; its complexity is $O(n)$.

The shortest processing time (SPT) rule constructs the solution according to the non-decreasing order of the normal processing times of jobs on machine M_1, i.e., $p_j^{(1)}$; its computational complexity is $O(n \log n)$.

The NEH algorithm (Algorithm 1) is based on the method introduced by (Nawaz et al., 1983). It starts with an initial solution $\pi_{initial}$ that determines the order of jobs that are subsequently inserted into the resulting solution π^* such that the criterion value $C_{\max}(\pi^*)$ is minimized. The computational complexity of this algorithm is $O(mn^3)$.

Algorithm 1: NEH.

1: Determine the initial sequence of jobs in $\pi_{initial}$ and set $\pi^* := \emptyset$
2: Get the first job j from $\pi_{initial}$
3: Insert j in such a position in π^* for which $C_{\max}(\pi^*)$ is minimal
4: Remove j from $\pi_{initial}$
5: If $\pi_{initial} \neq \emptyset$ Then go to Step 2
6: The permutation π^* is the given solution

Algorithm 2: SA.

1: Determine initial solution π_{init} and $\pi = \pi^* = \pi_{init}$, $T = T_0$
2: For $i = 1$ to *Iterations*
3: Choose π' by a random interchange of two jobs in π
4: Assign $\pi = \pi'$ with probability $P(T, \pi', \pi) = \min\left\{1, exp\left(-\frac{C_{\max}(\pi') - C_{\max}(\pi)}{T}\right)\right\}$
5: If $C_{\max}(\pi) < C_{\max}(\pi^*)$ Then $\pi^* = \pi$
6: $T = \frac{T}{1+\lambda T}$
7: The permutation π^* is the given solution

The presented simulated annealing (SA) algorithm (Algorithm 2), that is based on (Kirkpatrick et al., 1983), starts with an initial solution $\pi_{initial}$ and generates in each iteration a new permutation π' based on the current solution π by interchanging of two randomly chosen jobs in π. This new solution π' replaces π (i.e., $\pi = \pi'$) with the following probability $P(T, \pi', \pi) = \min\left\{1, exp\left(-\frac{C_{\max}(\pi') - C_{\max}(\pi)}{T}\right)\right\}$, where T is the temperature that decreases in a logarithmical manner $T = \frac{T}{1+\lambda T}$, and the values of the initial temperature T_0 and of the parameter λ are chosen empirically. The solution π^* with the minimal found criterion value $C_{\max}(\pi^*)$ is also stored. The algorithm stops after *Iterations* steps, thus, its overall computational complexity is $O(Iterations \cdot mn)$.

4 NUMERICAL ANALYSIS

In practice, a schedule for a real-life problem (e.g., in manufacturing systems) is calculated on the basis of a model and values of its parameters. Due to the possible differences between estimated and real values of problem parameters (e.g., shape of the learning curve, job processing times), the algorithms that are efficient for the modelled problem do not have to be accurate for the real problem. Therefore, it is crucial to evaluate how uncertain values of parameters affect the quality of solutions provided by such algorithms. Some of the analysed algorithms were described in (Rudek, 2011).

Let REAL denote the flowshop problem $Fm|\tilde{p}_j(v) = p_j v^{\alpha_j}|C_{\max}$, where job processing times are described by (1) and the values of the job parameters ($p_j^{(z)}$, $\alpha_j^{(z)}$) are precise. However, in practice it is usually difficult to obtain such accurate values and solution methods are based on uncertain (estimated) values. Following this, let ESTIM denote the flowshop scheduling problem, where the exact values of job parameters are unknown. In this case, job parameters are estimated, and we assume that job processing times are described by $\widehat{\tilde{p}}_j^{(z)}(v) = \widehat{p}_j^{(z)} v^{\widehat{\alpha}}$, where $\widehat{p}_j^{(z)}$ and $\widehat{\alpha}$ are the estimated values of $p_j^{(z)}$ and $\alpha_j^{(z)}$, respectively.

In the further part of this section, we provide the numerical analysis of the presented algorithms concerning the impact of the imprecise model on their efficiency. It is done according to the following steps. First, we draw values of job parameters for the problem REAL. Next, we solve the problem REAL using an algorithm A, which find a schedule π with criterion value $C_{\max}(\pi)$. Based on the parameters of the problem REAL, we draw or determine values of parameters for the problem ESTIM ($Fm|\tilde{p}_j(v) = p_j v^{\alpha}|C_{\max}$), which simulates their estimation. Next, we use the algorithm A to calculate a schedule $\widehat{\pi}$ for the problem ESTIM. For this schedule, we calculate the corresponding criterion value $C_{\max}(\widehat{\pi})$ for the problem REAL. The difference $C_{\max}(\widehat{\pi}) - C_{\max}(\pi)$ informs about the usefulness of the algorithm A in case of imprecise values of job parameters. Algorithms with smaller differences are more stable (robust), than those with greater.

The values of parameters for the problem REAL are generated as follows. For each pair of $n \in \{10, 25, 50\}$ and $m \in \{2, 3\}$, 100 random instances are generated from the uniform distribution in the following ranges of parameters: $p_j^{(z)} \in [1, 10]$, $\alpha_j^{(z)} \in [-0.51, -0.15]$ for $j = 1, \ldots, n$ and $z = 1, \ldots, m$. In all experiments in this paper, p_j are integers and α_j are rational values with accuracy of two decimal place, e.g., for $\alpha_j^{(z)} \in [-0.51, -0.15]$ it is $\alpha_j^{(z)} \in \{-0.51, -0.50, -0.49, \ldots, -0.15\}$. The values of $\alpha_j^{(z)} \in [-0.51, -0.15]$ corresponds to the learning curves in range between 70% and 90%, which are most common in practice (Biskup, 2008).

The values of the normal processing times for ESTIM are $\widehat{p}^{(z)} = p_j^{(z)}(1+\Delta_p)$, where Δ_p is the estimation error, which allows us to control precision of parameters for the analysis; it simulates the estimation process. The values of Δ_p and $\widehat{\alpha}$ are provided for particular experiments in Table 1.

Let $A_R = \{\text{ESA}, \text{ESA}_{\max}, \text{RND}, \text{SPT}, \text{NEH}, \text{SA}\}$ denote the algorithms that calculate the schedule for the problem REAL, where $\text{ESA}_{\max}$ is the algorithm that calculates the schedule with the maximum possible criterion value (opposite to ESA). ESA and $\text{ESA}_{\max}$ clearly show the place of the errors provided by the analysed algorithms in reference to the optimum and the worst criterion values. Note that the algorithms RND provide the same solution (schedule) for REAL and ESTIM. On the other hand, let $A_E = \{\widehat{\text{ESA}}, \widehat{\text{SPT}}, \widehat{\text{NEH}}, \widehat{\text{SA}}\}$ denote the corresponding algorithms from A_R that calculate the schedule for the problem ESTIM.

The initial solution for NEH and SA is a random permutation (in this case natural) and values of the parameters of SA were chosen empirically as follows: $Iterations = 1000000$, $T_0 = 1000000$ and $\lambda = 0.01$.[1]

The algorithms are evaluated, for each instance I, according to the relative error $\delta_A(I) = \left(\frac{C_{\max}(\pi_I^A)}{C_{\max}(\pi_I^*)} - 1\right) \cdot 100\%$, where $C_{\max}(\pi_I^A)$ denotes the criterion value provided by algorithm $A \in \{A_R, A_E\}$ for instance I and $C_{\max}(\pi_I^*)$ is the optimal solution of instance I (if $n = 10$) or the best found solution of instance I (if $n \geq 25$) provided by the considered algorithms. The optimal solution is provided by ESA for the problem REAL. The results concerning the percentage values of mean, minimum and maximum relative errors and mean criterion values $\bar{C}_{\max}$ (rounded to integer) provided by the analysed algorithms are presented in Table 1.

First, we discuss the results provided by the heuristic and metaheuristic algorithms for the prob-

[1] All algorithms were coded in C++ and simulations were run on PC, Intel® Core™i7–2600K Processor and 8GB RAM.

Table 1: The impact of model parameter uncertainty on the errors of the algorithms for $p_j^{(z)} \in [1,10]$, $\alpha_j^{(z)} \in [-0.51,-0.15]$, $\Delta_p \in [-0.25,0.25]$, $\widehat{\alpha} = -0.322$.

n	m	Algorithms	$\widehat{C}_{\max}$	Errors		
				Mean	Min	Max
10	2	ESA	36	0.00	0.00	0.00
		$\text{ESA}_{\max}$	54	44.35	21.37	74.72
		RND	44	19.58	4.05	46.09
		SPT	39	5.20	0.00	18.63
		NEH	37	1.60	0.00	8.09
		SA	36	0.00	0.00	0.00
		$\widehat{\text{ESA}}$	38	3.09	0.00	16.41
		$\widehat{\text{SPT}}$	39	5.60	0.15	21.73
		$\widehat{\text{NEH}}$	38	3.45	0.00	17.11
		$\widehat{\text{SA}}$	38	3.05	0.00	16.41
	3	ESA	41	0.00	0.00	0.00
		$\text{ESA}_{\max}$	62	49.64	28.56	80.89
		RND	52	21.85	6.16	48.14
		SPT	45	10.31	0.54	34.50
		NEH	43	2.20	0.00	10.15
		SA	41	0.01	0.00	0.44
		$\widehat{\text{ESA}}$	43	4.31	0.00	16.21
		$\widehat{\text{SPT}}$	46	10.93	0.25	30.58
		$\widehat{\text{NEH}}$	43	5.21	0.57	19.90
		$\widehat{\text{SA}}$	43	4.18	0.00	16.21
25	2	RND	82	19.57	7.95	32.41
		SPT	75	6.65	0.10	19.38
		NEH	72	2.34	0.12	5.82
		SA	70	0.00	0.00	0.00
		$\widehat{\text{SPT}}$	75	6.74	0.32	18.47
		$\widehat{\text{NEH}}$	73	4.93	0.55	12.76
		$\widehat{\text{SA}}$	73	3.90	0.26	13.73
	3	RND	84	22.71	3.30	41.19
		SPT	77	12.77	4.96	29.46
		NEH	71	3.51	0.51	7.61
		SA	70	0.00	0.00	0.00
		$\widehat{\text{SPT}}$	78	12.90	5.05	28.37
		$\widehat{\text{NEH}}$	75	7.53	2.11	19.88
		$\widehat{\text{SA}}$	75	6.10	1.12	18.19
50	2	RND	129	19.32	9.61	31.39
		SPT	119	9.35	0.20	20.08
		NEH	113	3.09	0.12	7.18
		SA	109	0.00	0.00	0.00
		$\widehat{\text{SPT}}$	118	9.44	0.43	20.15
		$\widehat{\text{NEH}}$	116	6.47	0.51	13.04
		$\widehat{\text{SA}}$	113	4.19	0.42	11.86
	3	RND	141	20.41	10.33	31.11
		SPT	133	14.21	5.87	28.94
		NEH	122	3.78	1.02	7.41
		SA	117	0.00	0.00	0.00
		$\widehat{\text{SPT}}$	134	13.93	6.46	30.50
		$\widehat{\text{NEH}}$	125	7.66	2.95	15.01
		$\widehat{\text{SA}}$	125	5.96	1.40	14.29

lem REAL, for which the exact values of model parameters are known. It can be seen in Table 1 that SA finds solutions with criterion values close to the optimum. On the other hand, the differences between the mean relative errors provided by SA and NEH is about 3.5% and for the maximum errors 10%; for SPT it is about 14% for mean and 35% for maximum errors. The random solution is usually equally between the optimal and the worst case ($n = 10$) and provides mean and maximum errors (in reference to SA) about 20% and 45%, respectively.

However, if the applied algorithms are based on uncertain values of model parameters (solve the problem ESTIM), then their accuracy decreases in reference to the criterion value found by the algorithms, which are based on exact values (solve the problem REAL). It can be seen in Table 1 (for $n = 10$), that SA is more robust with respect to model parameter uncertainty than ESA. Namely, solutions obtained for ESTIM by $\widehat{\text{SA}}$ have lower criterion values (in reference to REAL) than provided by $\widehat{\text{ESA}}$. Note that the mean relative errors of NEH and SA increase about 3-5% if model parameters are uncertain. The exception is SPT, which is robust to the analysed model parameter uncertainty, however, it provides solutions with relative errors greater than $\widehat{\text{NEH}}$ and $\widehat{\text{SA}}$. Note that the considered algorithms calculate schedules that are significantly lower than a random solution (RND).

From the numerical analysis follows that NEH and SA can be efficiently applied to solve the considered real-life problem even if the model parameters are uncertain.

5 CONCLUSIONS

In this paper, we analysed the impact of model parameter uncertainty on the accuracy of solution algorithms for the makespan minimization flowshop scheduling problem with job processing times described by the power functions dependent on the number of processed jobs. We showed that the considered algorithms are efficient even if the values of problem parameters are not precisely identified.

REFERENCES

Biskup, D. (2008). A state-of-the-art review on scheduling with learning effects. *European Journal of Operational Research*, 188:315–329.

Gupta, J. N. D. and Stafford, J. E. F. (2006). Flowshop scheduling research after five decades. *European Journal of Operational Research*, 169:699–711.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220:671–680.

Lee, W.-C. and Wu, C.-C. (2004). Minimizing total completion time in a two-machine flowshop with a learning effect. *International Journal of Production Economics*, 88:85–93.

Mosheiov, G. and Sidney, J. B. (2003). Scheduling with general job–dependent learning curves. *European Journal of Operational Research*, 147:665–670.

Nawaz, M., Enscore, J. E. E., and Ham, I. A. (1983). A heuristic algorithm for *m*-machine, *n*-jobs Flow-shop sequencing problem. *Omega*, 11:91–95.

Rudek, R. (2011). Computational complexity and solution algorithms for flowshop scheduling problems with the learning effect. *Computers & Industrial Engineering*, 61:20–31.