Robust single machine scheduling with random blocks in an uncertain environment

Wojciech Bożejko[1], Paweł Rajba[2], and Mieczysław Wodecki[3]

[1] Department of Automatics, Mechatronics and Control Systems,
Faculty of Electronics, Wrocław University of Technology,
Janiszewskiego 11-17, 50-372 Wrocław, Poland
wojciech.bozejko@pwr.wroc.pl, ORCID 0000-0002-1868-8603
[2] Institute of Computer Science, University of Wrocław
Joliot-Curie 15, 50-383 Wrocław, Poland
pawel@cs.uni.wroc.pl, ORCID 0000-0003-4252-5545
[3] Telecommunications and Teleinformatics Department
Faculty of Electronics, Wrocław University of Science and Technology
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
mieczyslaw.wodecki@pwr.wroc.pl, ORCID 0000-0001-8188-4503

Abstract. While scheduling problems in deterministic models are quite well investigated, the same problems in an uncertain environment require very often further exploration and examination. In the paper we consider a single machine tabu search method with block approach in an uncertain environment modeled by random variables with the normal distribution. We propose a modification to the tabu search method which improves the robustness of the obtained solutions. The conducted computational experiments show that the proposed improvement results in a much more robust solutions than the ones obtained in the classic block approach.

Keywords: single machine scheduling, uncertain parameters, normal distribution, tabu search, block approach

1 Introduction

Uncertainty occurs in many production processes and has a direct impact on their smooth execution. For instance it is important in construction domain to deliver goods with no delays, but it is not easy to meet this requirement as the transportation time depends on many external factors like weather conditions, traffic jams, driver's condition and many others. Moreover, effective solving practical problems and taking the best approach requires also thorough knowledge of the process or production system and values of all parameters. For example an uncertain data of the duration of activities (operations) can be measured and in result: approximated as deterministic ones in case the variance is small enough, modeled by an appropriate probabilistic distribution or determined the membership function for the fuzzy representation. So, as in practice it is difficult to clearly determine the process parameters, quite often safe ones are taken (e.g.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50436-6_39

assume longer transportation time) what is an opportunity for further improvements.

Research on scheduling problems carried out for many years is related primarily to deterministic models where the key assumption is that parameters are well defined. For those, mostly belonging to the class of strongly NP-hard problems, a number of very effective approximate algorithms have been developed. Solutions determined by these algorithms are very often only slightly worse from the optimal ones. In practice, however, as already mentioned, some parameters (e.g. operation times) may differ during the process execution from the initially assumed values. This can cause that the actual cost of execution is much bigger than expected what leads to either losing optimality or even acceptability (feasibility) of solutions.

In order to close that gap in recent years more and more research has been conducted on developing methods which find more robust solution resistant to data disturbance. Uncertain parameters are usually represented by random variables or fuzzy numbers and extensive review of methods and algorithms for solving optimization problems with random parameters is presented by Vondrák in monograph [12] and newer of Soroush [10], Xiaoqiang et al. [14], Urgo and Vancza [11], Zhang et al. [15] and Bożejko et al. [4] and [6].

In this paper we consider a single machine scheduling problem with due dates in two variants where either job execution times or due dates are represented by independent variables with normal distribution. We also present some properties of the problem (so-called block elimination properties) accelerating the review of neighborhoods in local search algorithms. The main goal is to compare the robustness of the block-based tabu search algorithm in the classic and the proposed random model and show the superiority of the latter one.

2 Deterministic scheduling problem

Let $\mathcal{J} = \{1, 2, \ldots, n\}$ be a set of jobs to be executed on a single machine. At any given moment a machine can execute exactly one job and all jobs must be executed without preemption. For each task $i \in \mathcal{J}$ let p_i be a *processing time*, d_i be a *due date* and w_i be a cost for tardy jobs.

Every sequence of jobs execution can be presented as a permutation $\pi = (\pi(1), \pi(2), \ldots, \pi(n))$ of items from the set $\mathcal{J}$.

Let Π be the set of all permutations of the set $\mathcal{J}$. For every permutation $\pi \in \Pi$ we define

$$C_{\pi(i)} = \sum_{j=1}^{i} p_{\pi(j)}$$

as a completion time of a job $\pi(i)$.

The cost of jobs' execution determined by the permutation π is as follows

$$\sum_{i=1}^{n} w_{\pi(i)} U_{\pi(i)}. \tag{1}$$

DOI: 10.1007/978-3-030-50436-6_39

where

$$U_{\pi(i)} = \begin{cases} 0 & \text{for} \quad C_{\pi(i)} \leqslant d_{\pi(i)}, \\ 1 & \text{for} \quad C_{\pi(i)} > d_{\pi(i)}. \end{cases}$$

We consider the optimization problem where the goal is to find a permutation $\pi^* \in \Pi$ which minimizes the cost of jobs' execution:

$$W(\pi^*) = \min_{\pi \in \Pi} \left(\sum_{i=1}^{n} w_{\pi(i)} U_{\pi(i)} \right).$$

3 Probabilistic jobs times

In order to simplify the further considerations we assume w.l.o.g. that at any moment the considered solution is the natural permutation, i.e. $\pi = (1, 2, \ldots, n)$. Moreover, if X is a random variable, then F_X denotes its cumulative distribution function.

In this section we consider a TWT problem with uncertain parameters. We investigate two variants: (a) uncertain processing times and (b) uncertain due dates.

3.1 Random processing times

Random processing times are represented by random variables with the normal distribution $\tilde{p}_i \sim N(p_i, c \cdot p_i))$, $i \in \mathcal{J}$. Other parameters, i.e. due dates d_i and costs w_i are deterministic. Then completion times $\tilde{C}_i$ are random variables:

$$\tilde{C}_i \sim N\left(p_1 + p_2 \ldots + p_i, c \cdot \sqrt{p_1^2 + \ldots + p_i^2}\right) \qquad (2)$$

and delays are random variables

$$\tilde{U}_i = \begin{cases} 0 & \text{dla} \quad \tilde{C}_i \leqslant d_i, \\ 1 & \text{dla} \quad \tilde{C}_i > d_i. \end{cases} \qquad (3)$$

For each permutation $\pi \in \Pi$ the cost in the deterministic model is defined as $W(\pi) = \sum_{i=1}^{n} w_{\pi(i)} U_{\pi(i)}$ (see (1)). A corresponding cost in the random model is defined as the following random variable:

$$\widetilde{W}(\pi) = \sum_{i=1}^{n} w_i \tilde{U}_i. \qquad (4)$$

In order to compare the costs of permutations from the set Π we introduce the following *comparison function* to calculate the value:

$$\mathcal{W}(\pi) = w_i E(\tilde{U}_i) \qquad (5)$$

where $E(\tilde{U}_i)$ is the expected value of the random variable $\tilde{U}_i$.

DOI: 10.1007/978-3-030-50436-6_39

3.2 Random due dates

Random due dates are represented by random variables with the normal distribution $\tilde{d}_i \sim N(d_i, c \cdot d_i))$, $i \in \mathcal{J}$. Other parameters, i.e. processing times p_i and costs w_i are deterministic. Delay indication is a random variable

$$\tilde{U}_i = \begin{cases} 0 & \text{dla} \quad C_i \leqslant \tilde{d}_i, \\ 1 & \text{dla} \quad C_i > \tilde{d}_i. \end{cases} \tag{6}$$

In this variant of the problem we apply the comparison function (5) defined in the previous section.

The *TWT* problem in both variants (i.e. with random processing times and random due dates) is to find a permutation for which the comparison function (5) is minimal in the set Π. We denote the probabilistic version of the problem as *TWTP*. As the deterministic version, the problem belongs to the class of *NP*-hard problems.

4 Blocks in random model

4.1 Random processing times and due dates

Each permutation $\pi \in \Pi$ is decomposed into m ($m \leqslant n$) subpermutations $\widetilde{B}_1, \ldots, \widetilde{B}_m$, called random blocks for π, which satisfy the following criteria:

1. $\widetilde{B}_k = (s_k, s_k + 1, \ldots, l_k - 1, l_k)$, $l_{k-1} + 1 = s_k \leqslant l_k$, $k = 1, \ldots, m$, $l_0 = 0$, $l_m = n$.
2. All jobs $j \in B_k$ satisfy either the condition
$$P(\tilde{d}_j \geqslant \widetilde{C}_{l_k}) \geqslant 1 - \epsilon \tag{7}$$
or the condition
$$P(\tilde{d}_j \leqslant \widetilde{S}_{s_k} + \tilde{p}_j) \geqslant 1 - \epsilon. \tag{8}$$
3. $\widetilde{B}_k$ is maximal subsequence of π where all the jobs satisfy either (7) or (8).

We distinguish two types of blocks:

- E-Random Blocks, denoted as $\widetilde{B}_k^E$, the ones satisfying condition (7),
- T-Random Blocks, denoted as $\widetilde{B}_k^T$, the ones satisfying condition (8).

Theorem 1. *Let π be a permutation with a distinguished random block $\widetilde{B}$, i.e. $\pi = (1, 2, ..., s_k, s_k + 1, \ldots, l_k - 1, l_k, \ldots, n)$ where $A = (1, \ldots, s_k - 1)$, $B = (s_k, \ldots, l_k)$ and $C = (l_k + 1, \ldots, n)$. Estimated value of comparison function can be calculated as follows:*

$$\begin{aligned} W_{ABC} &= W_A + W_B + W_C \\ &= \sum_{i=1}^{s_k - 1} w_i E(\tilde{U}_i) + \sum_{i=s_k}^{l_k} w_i E(\tilde{U}_i) + \sum_{i=l_k+1}^{n} w_i E(\tilde{U}_i) \\ &= \sum_{i \in \pi} w_i E(\tilde{U}_i). \end{aligned}$$

Now let $\mathcal{B}$ be a set of all permutations of $B = (s_k, \ldots, l_k)$ and for each $B' \in \mathcal{B}$ we define $W_{AB'C} = W_A + W'_B + W_C$ in the same way as W_{ABC}.

Then we have the following

a) if B is a random E-block, then for each $B' \in \mathcal{B}$ $W'_B \leqslant \sum_{i=s_k}^{l_k} w_i \cdot \epsilon$,
b) if B is a random T-block, then for each $B' \in \mathcal{B}$ $W'_B \geqslant \sum_{i=s_k}^{l_k} w_i \cdot (1-\epsilon)$

what in result gives as that for each $B' \in \mathcal{B}$ there is a fixed upper bound for $W_{AB'C}$.

Proof. Let's consider the following 2 cases.
A. B is random E-block. Then we have:

$$W_B = \sum_{i=s_k}^{l_k} w_i E(\tilde{U}_i) = \sum_{i=s_k}^{l_k} w_i P(\tilde{C}_i > \tilde{d}_i) = \sum_{i=s_k}^{l_k} w_i (1 - P(\tilde{C}_i \leqslant \tilde{d}_i)).$$

Applying our assumption that B fulfills (7) (i.e. B is a random E-block) as well as by definition of $\widetilde{C}_i$ and the problem formulation where every realization of $\widetilde{C}_i$ will be less or equal than realization of $\widetilde{C}_{l_k}$ we obtain that

$$P(\tilde{C}_i \leqslant \tilde{d}_i) \leqslant P(\widetilde{C}_{l_k} \leqslant \tilde{d}_i)$$

for all $i \in \widetilde{B}_k$. Having that we can proceed as follows:

$$W_B = \sum_{i=s_k}^{l_k} w_i (1 - P(\widetilde{C}_{l_k} \leqslant d_i)) \leqslant \sum_{i=s_k}^{l_k} w_i (1 - 1 + \epsilon) = \sum_{i=s_k}^{l_k} w_i \cdot \epsilon$$

what leads us to the conclusion that for each permutation $B' \in \mathcal{B}$ we have

$$W'_B \leqslant \sum_{i=s_k}^{l_k} w_i \cdot \epsilon.$$

B. B is random T-block. Then we have:

$$W_B = \sum_{i=s_k}^{l_k} w_i E(\tilde{U}_i) = \sum_{i=s_k}^{l_k} w_i P(\tilde{C}_i > \tilde{d}_i).$$

By definition of S_{s_k} and $\tilde{C}_i$ $(i \in B)$ we can easily observe the following:

$$\tilde{C}_i = \widetilde{S}_{s_k} + \tilde{p}_{s_k} + \tilde{p}_{s_k+1} + \tilde{p}_{s_k+2} + \ldots + \tilde{p}_i \geqslant \widetilde{S}_{s_k} + \tilde{p}_i.$$

Having that and applying our assumption that B fulfills (8) (i.e. B is a random T-block) we obtain that

$$P(\tilde{d}_i < \tilde{C}_i) \geqslant P(\tilde{d}_i < S_k + p_i) \geqslant 1 - \epsilon$$

what implies that

$$P(\tilde{d}_i < \tilde{C}_i) \geqslant 1 - \epsilon$$

for all $i \in \widetilde{B}_k$. Having that we can proceed as follows:

$$W_B = \sum_{i=s_k}^{l_k} w_i P(\tilde{C}_i > \tilde{d}_i) \geqslant \sum_{i=s_k}^{l_k} w_i (1-\epsilon)$$

what leads us to the conclusion that for each permutation $B' \in \mathcal{B}$ we have

$$W'_B \geqslant \sum_{i=s_k}^{l_k} w_i \cdot (1-\epsilon).$$

That concludes the proof.

The above theorem also holds in case where we consider only random processing times or only random due dates and each case the proof is analogous.

4.2 Improving robustness by applying the derived theorem

It is easy to show the following. For the variant with random processing times $\tilde{p}_i$ we have:

$$E(\widetilde{U}_{\pi(i)}) = P(\widetilde{C}_{\pi(i)} > d_{\pi(i)}) = 1 - F_{\widetilde{C}_{\pi(i)}}(d_{\pi(i)}).$$

and for the variant with random due dates $\tilde{d}_i$ we have:

$$E(\widetilde{U}_{\pi(i)}) = P(C_{\pi(i)} > \widetilde{d}_{\pi(i)}) = F_{\widetilde{d}_{\pi(i)}}(C_{\pi(i)}).$$

Combining the above with assumptions expressed in (7) and (8) and adapted to respective variants, we apply the following rules to modify the base tabu search method. For the variant with random processing times $\tilde{p}_i$:

a) if B is a random E-block, then $W_B = \sum_{i=s_k}^{l_k} w_i \cdot (1 - F_{\widetilde{C}_{\pi(i)}}(d_{\pi(i)}))$,
b) if B is a random T-block, then $W_B = \sum_{i=s_k}^{l_k} w_i \cdot F_{\widetilde{C}_{\pi(i)}}(d_{\pi(i)})$.

For the variant with random due dates $\tilde{d}_i$:

a) if B is a random E-block, then $W_B = \sum_{i=s_k}^{l_k} w_i \cdot F_{\widetilde{d}_{\pi(i)}}(C_{\pi(i)})$,
b) if B is a random T-block, then $W_B = \sum_{i=s_k}^{l_k} w_i \cdot (1 - F_{\widetilde{d}_{\pi(i)}}(C_{\pi(i)})$.

5 Computational experiments

In this section we present the results of the robustness property comparison between the tabu search method with blocks and the tabu search method with blocks and theorem applied in a way described in Section 4. All tests are executed with a modified version of tabu search method described in [1]. The algorithm has been configured with the following parameters: $\pi = (1, 2, \ldots, n)$ is an initial permutation, n is the length of tabu list and n is the number of algorithm iterations where n is the tasks number.

Both methods have been tested on instances from OR-Library ([8]) where there are 125 examples for $n = 40$, 50 and 100 (in total 375 examples). For each example and each parameter $c = 0.02$, 0.04, 0.06 and 0.08 (expressing 4 levels of data disturbance) 100 randomly disturbed instances were generated according to the normal distribution defined in Section 3.1 (in total 400 disturbed instances per example). The full description of the method for disturbed data generation can be found in [5].

All the presented results in this section are calculated as the relative coefficient according to the following formula:

$$\delta = \frac{W - W^*}{W^*} \cdot 100\% \tag{9}$$

which expresses by what percentage the investigated solution W is worse than the reference (best known) solution W^*. Details of calculating robustness of the investigated methods can also be found in [5].

An algorithm without applied theorem we denote by $\mathcal{AD}$ and the one with applied theorem by $\mathcal{AP}$.

5.1 Results

In Tables 1 and 2 we present a complete summary of the computational experiments results. Values from columns $\mathcal{AD}$ and $\mathcal{AP}$ in both tables represent a relative distance between solutions established by a respective algorithm and the best known solution. The distance is based on (9) and it is an average of all solutions calculated for the disturbed data on a respective disturbance level expressed by the parameter c. Value from column *IF* (what stands for *improvement factor*) expresses the relative distance (also based on (9)) between the results obtained by $\mathcal{AD}$ and the results obtained by $\mathcal{AP}$.

Table 1. Relative distance between robustness coefficient of algorithm $\mathcal{AD}$ (or respectively $\mathcal{AP}$) and the reference value for random p_i on different disturbance levels (0.02–0.08)

N	40			50			100		
c	$\mathcal{AD}$	$\mathcal{AP}$	IF	$\mathcal{AD}$	$\mathcal{AP}$	IF(%)	$\mathcal{AD}$	$\mathcal{AP}$	IF
0.02	757.9	25.6	2863%	820.6	24.3	3275%	3625.5	11.5	31558%
0.04	1776.3	24.6	7112%	2132.2	25.8	8177%	5146.6	12.2	41952%
0.06	2442.4	26.8	9022%	3013.2	25.4	11762%	6488	13.5	47831%
0.08	2821.2	29.4	9509%	5656.3	28.7	19627%	7957.5	14.8	53661%
Avg	**1949.5**	**26.6**	**7233%**	**2905.6**	**26.0**	**11060%**	**5804.4**	**13.0**	**44525%**

Table 2. Relative distance between robustness coefficient of algorithm $\mathcal{AD}$ (or respectively $\mathcal{AP}$) and the reference value for random p_i on different disturbance levels (0.02–0.08)

N	40			50			100		
c	$\mathcal{AD}$	$\mathcal{AP}$	IF	$\mathcal{AD}$	$\mathcal{AP}$	IF(%)	$\mathcal{AD}$	$\mathcal{AP}$	IF
0.02	3000.2	70.8	4136%	4661.6	31.8	14547%	11812.1	20.1	58795%
0.04	4719.2	124.9	3678%	7948.9	181.8	4273%	17830.2	141.3	12517%
0.06	6303.4	271.6	2220%	8191.3	284.6	2777%	15180.5	264	5649%
0.08	5654.6	341.6	1555%	8109.1	511.8	1484%	7235.6	322.2	2145%
Avg	**4919.3**	**202.2**	**2332%**	**7227.7**	**252.5**	**2762%**	**13026.2**	**186.6**	**6880%**

We can easily observe that applying the theorem into the method improves results very significantly for all the investigated cases. We can also observe that robustness coefficients for random d_i are generally worse than ones for random p_i what can be explained by the fact that for random d_i there are more fluctuations in disturbed data than for random p_i what is implied by the disturbed data generation method. The other observation is related to the results on different disturbance levels (parameter c). With the bigger value of c we might expect the worse robustness coefficient. Surprisingly, we can observe a difference between results for random p_i and random d_i. For random p_i the rule generally works both for $\mathcal{AD}$ and $\mathcal{AP}$, only for $\mathcal{AP}$ there is a swap between $c = 0.02$ and $c = 0.04$ for $N = 40$ and a swap between $c = 0.04$ and $c = 0.06$ for $N = 50$, nevertheless those values are all very close to each other, so those swaps are actually meaningless. For d_i the situation is different. The rule still works for $\mathcal{AP}$, but for $\mathcal{AD}$ we are not able to observe any trend. That can be considered as an advantage of $\mathcal{AP}$ as it behaves in a more predictive and stable way than $\mathcal{AD}$ does.

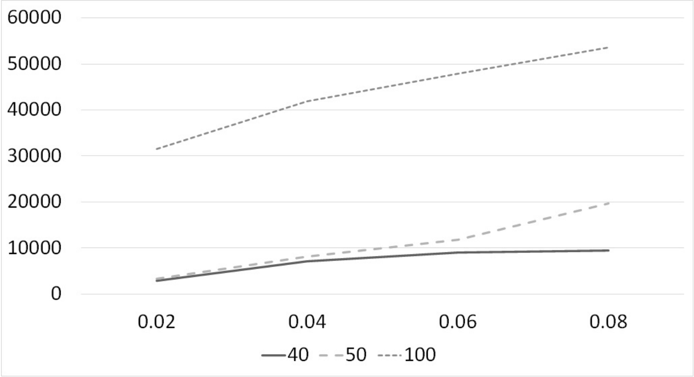

Fig. 1. Comparison of the robustness level with reference to the main methods

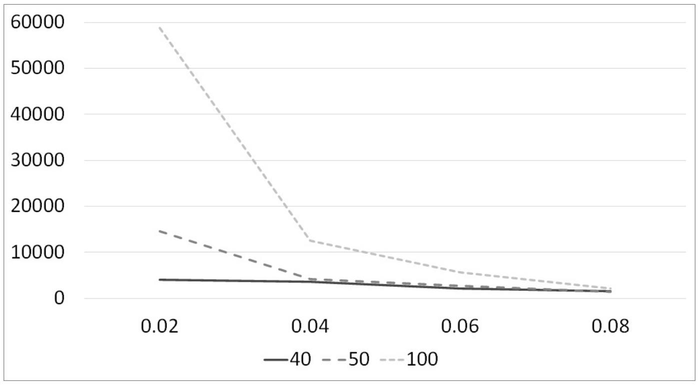

Fig. 2. Comparison of the robustness level with reference to the main methods

It is also worth noting the difference of trends for random p_i and random d_i of the comparison between $\mathcal{AD}$ and $\mathcal{AP}$ (column *IF*) on different disturbance levels which are visualized on Figures 1 and 2. While for random p_i we can see that with the increase of the parameter c the gap between $\mathcal{AD}$ and $\mathcal{AP}$ is also increasing, for random d_i we can observe exactly the opposite. On the other hand the order of magnitude of *improvement factor* (*IF*) is the same for both random p_i and random d_i what shows that the level of improvement introduced by the presented Theorem works similarly in both considered scenarios.

5.2 Parallelization consideration

The tabu search method is well paralleling. According to the classification proposed by Voß [13], four models of the parallel tabu search method can be considered: SSSS (Single Starting point Single Strategy), SSMS (Single Starting point Multiple Strategies), MSSS (Multiple Starting point Single Strategy), MSMS (Multiple Starting point Multiple Strategies). They refer to the classic classification of Flynn's parallel architectures [7]. In the version proposed in this paper where block approach is applied it is natural to use MSSS or MSMS diversification strategies, because the block mechanism is quite easily parallelized (each block can be considered separately [3]). The use of 'tree' strategies using the same start solution is also possible provided the fact that the process of searching the solution space at a later stage is diversified (e.g. by using different tabu length for individual threads or through a mechanism of dynamic tabu list length change [1]).

6 Conclusions

In the paper we considered a single machine tabu search method with block approach in an uncertain environment modeled by random variables with the normal distribution. We proposed a theorem which allows to modify the base tabu

DOI: 10.1007/978-3-030-50436-6_39

search method in a way which improves the robustness of calculated solutions. Computational experiments conducted on disturbed data confirmed substantial predominant of the method after applying the proposed theorem.

Acknowledgments

This work was partially funded by the National Science Centre of Poland, grant OPUS no. 2017/25/B/ST7/02181 and a statutory subsidy 049U/0032/19.

References

1. Bożejko, W., Grabowski, J., & Wodecki, M. (2006). Block approach–tabu search algorithm for single machine total weighted tardiness problem. Computers & Industrial Engineering, 50(1–2), 1–14.
2. Bożejko, W., Rajba, P., & Wodecki, M. (2014). Stable Scheduling with Random Processing Times. In Advanced Methods and Applications in Computational Intelligence (pp. 61–77). Springer, Heidelberg.
3. Bożejko, W., Uchroński, M., & Wodecki, M. (2016). Parallel metaheuristics for the cyclic flow shop scheduling problem. Computers & Industrial Engineering, 95, 156–163.
4. Bożejko, W., Rajba, P., & Wodecki, M. (2017). Stable scheduling of single machine with probabilistic parameters. Bulletin of the Polish Academy of Sciences Technical Sciences, 65(2), 219–231.
5. Bożejko, W., Rajba, P., & Wodecki, M. (2018, July). Robustness of the Uncertain Single Machine Total Weighted Tardiness Problem with Elimination Criteria Applied. In International Conference on Dependability and Complex Systems (pp. 94–103). Springer, Cham.
6. Bożejko, W., Hejducki, Z., & Wodecki, M. (2019). Flowshop scheduling of construction processes with uncertain parameters. Archives of Civil and Mechanical Engineering, 19(1), 194–204.
7. Flynn, M. J. (1995). Computer architecture: Pipelined and parallel processor design. Jones & Bartlett Learning.
8. OR-Library
`http://www.brunel.ac.uk/~mastjjb/jeb/info.html` Accessed 13 April 2020
9. Shang, C., & You, F. (2018). Distributionally robust optimization for planning and scheduling under uncertainty. Computers & Chemical Engineering, 110, 53–68.
10. Soroush, H. M. (2013). Scheduling stochastic jobs on a single machine to minimize weighted number of tardy jobs. Kuwait Journal of Science, 40(1), 123–147.
11. Urgo, M., & Váncza, J (2019). A branch-and-bound approach for the single machine maximum lateness stochastic scheduling problem to minimize the value-at-risk. Flexible Services and Manufacturing Journal 31, 472-–496.
12. Vondrák, J. (2005). Probabilistic methods in combinatorial and stochastic optimization (Doctoral dissertation, Massachusetts Institute of Technology).
13. Voß, S. (1993). Tabu search: Applications and prospects. In Network Optimization Problems: Algorithms, Applications and Complexity (pp. 333–353).
14. Cai, X., Wu, X., & Zhou, X. (2014). Optimal stochastic scheduling (Vol. 4). New York: Springer.
15. Zhang, L., Lin, Y., Xiao, Y., & Zhang, X. (2018). Stochastic single-machine scheduling with random resource arrival times. International Journal of Machine Learning and Cybernetics, 9(7), 1101–1107.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50436-6_39