Sampling method for the robust single machine scheduling with uncertain parameters

Paweł Rajba[1][0000−0003−4252−5545]

Institute of Computer Science, University of Wrocław,
Joliot-Curie 15, 50-383 Wrocław,
pawel@cs.uni.wroc.pl

Abstract. Many real problems are defined in an uncertain environment where different parameters such as processing times, setup times, release dates or due dates are not known at the time of determining the solution. As using deterministic approach very often provides solutions with poor performance, several approaches have been developed to embrace the uncertainty and the most of the methods are based on: stochastic modeling using random variables, fuzzy modeling or bound form where values are taken from a specific interval. In the paper we consider a single machine scheduling problem with uncertain parameters modeled by random variables with normal distribution. We apply the sampling method which we investigate as an extension to the tabu search algorithm. Sampling provides very promising results and it is also a very universal method which can be easily adapted to many other optimization algorithms, not only tabu search. Conducted computational experiments confirm that results obtained by the proposed method are much more robust than the ones obtained using the deterministic approach.

Keywords: single machine scheduling · uncertain parameters · stochastic scheduling · normal distribution · tabu search · sampling method.

1 Introduction

Research on optimization problems for the last decades has been primarily focusing on deterministic models where we assume that problem parameters are specific and well defined. Unfortunately in many production processes we can observe different levels of uncertainty what has a direct impact on their smooth execution. For instance in many businesses delivering goods with no delays has a direct financial consequences. Unfortunately, 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, solving such problems effectively requires very often a thorough knowledge of the process or production system.

Depending on the nature of the problem and the level of our knowledge about the measured parameters, uncertainty can be modeled in different ways. For instance, approximated values can be taken in case the variance is very small, we

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77980-1_45

can use ranges of values or fuzzy representation in case we have limited understanding of the parameters' variation, finally, we can leverage random variables with specific probabilistic distributions and this approach we investigated in our paper. In literature scheduling based on random variables with probabilistic distributions is recognized as *stochastic scheduling*. Over the recent decades many different problems and their variants were investigated, during this time also many good reviews have been introduced. Basics of stochastic scheduling one can find in Pinedo [19] and more extensive reviews dedicated to methods solving scheduling problems in stochastic models are presented in Cai et. al. [8], Dean [10] and Vondrák [26].

There are different ways how randomness is considered and key ones are: uncertain problem parameters and machine breakdowns. The single machine scheduling problem where different problem parameters like processing times or due dates are uncertain is also approached in different ways. One way is to develop a scheduling policy. Rothkopf in [22] introduced WSEPT (weighted shortest expected processing time, ordering jobs with nonincreasing ratio $w_j/E[\tilde{p}_j]$) rule proved to be optimal for single machine scheduling with identical release dates in [23]. The approach is still being investigated and recently in [29] an optimal policy EWCT (the expected weighted completion time) for single-machine scheduling with random resource arrival times has been introduced. Earlier Cai in [7] showed that for single-machine scheduling with processing times modeled by random variables with exponential distribution and cost functions and due dates with any distribution, the WSEPT policy (weighted shortest expected processing time first) is optimal. More one can find in [15], [13] or [24]. The another way to solve the problem is to, instead of developing a scheduling policy, hire metaheuristics like tabu tabu or simulated annealing. This approach was investigated in Bożejko et al. [1], [2], [3], [4] and Rajba et. al. [20] where effective methods were proposed for single machine scheduling problem where parameters are modelled with random variables with the normal distribution. In [1], [2] and [20] additionally Erlang distribution was investigated and those papers cover $\sum w_i U_i$ and $\sum w_i T_i$ problem variants. The main goal of [3] and [4] was to introduce techniques to shorten the computational time (i.e. elimination criteria and random blocks) keeping the robustness of the determined solutions on a good level. An interesting approach has been also presented in Urgo et. al. [25] where a variant of stochastic single machine scheduling problem is considered with release times and processing times as uncertain parameters and solved using the classic branch-and-bound method. The other dimension is to define the appropriate stochastic objective function (what is related to stochastic dominance, see [12], [16], [19]). For the most of discussed research so far the goal is to minimize the average (expected) value, the variance, or some combination of those. It is worth to indicate that there is also another significant area of robust scheduling where to goal is to minimize and control the worst-case scenario (see [9], [14], [27] and [28]), however usually in those scenarios uncertainty is modelled by the bounded form instead of the stochastic approach.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77980-1_45

Random machine breakdowns is considered in the literature mainly in the following two variants: preemptive-resume mode where partially done work is continued after repair (see [5], [18]) and preemptive-repeat mode where partially done work is discarded and job needs to be started again (see [6], [11]). However, as in this paper we focus on uncertain parameters, we conclude machine breakdowns with this short introduction.

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 uncertain and modeled by independent variables with the normal distribution. We investigate the sampling method which is an extension to the tabu search algorithm and offers a probabilistic approach of finding solutions. To the best of author's knowledge, this technique was introduced in [21] for the first time and it wasn't studied in the scheduling literature. In this paper we introduce the following novel contribution:

- We apply the sampling method to the considered single machine scheduling problem with uncertain processing times and due dates,
- We introduce 3 optimization rules which makes samples more tailored and significantly decreases required samples' sizes keeping the robustness on a good level,
- We conduct an extensive experimental evaluation of the proposed method.

As verified in the computational experiments by applying the proposed method we obtain much more robust solutions than the ones obtained in the classic deterministic approach, moreover, the experiments also confirmed that applying optimization rules significantly reduces the samples' sizes.

The rest of the paper is structured as follows: in Section 2 we describe a classic deterministic version of the problem, then in Section 3 we introduce a randomized variant of the one. In Section 4 we present the method and optimization rules what is the main contribution of the paper, and in Section 5 a summary of computational experiments is described. Finally, in Section 6 conclusions and future directions close the paper.

2 Deterministic scheduling problem

Let $\mathcal{J} = \{1, 2, \ldots, n\}$ be a set of jobs to be executed on a single machine with conditions that (1) at any given moment a machine can execute exactly one job and (2) all jobs must be executed without preemption. For each $i \in \mathcal{J}$ we define p_i as a *processing time*, d_i as a *due date* and w_i as a cost for a delay.

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

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

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

We investigate the following ways of calculating cost function:

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77980-1_45

– sum of weights for tardy jobs,
– the total weighted tardiness.

Therefore we introduce the delay indicator

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

and cost factor

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

Then, the cost function for the permutation π is either

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

or

$$\sum_{i=1}^{n} w_{\pi(i)} T_{\pi(i)}. \tag{2}$$

Finally, the goal is to find a permutation $\pi^* \in \Pi$ which minimizes either

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

or

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

(depending on the considered variant)

3 Probabilistic model

In this section we introduce the randomization of the problem described in Section 2. We investigate two variants: (a) uncertain processing times and (b) uncertain due dates.

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. $\pi = (1, 2, \ldots, n)$.

3.1 Random processing times

Random processing times are represented by random variables with the normal distribution $\tilde{p}_i \sim N(p_i, c \cdot p_i))$ ($i \in \mathcal{J}$, c determine the disturbance level and

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77980-1_45

will be defined later) while due dates d_i and weights w_i are deterministic. Then, completion times $\tilde{C}_i$ are random variables

$$\tilde{C}_i \sim N\left(p_1 + p_2 \ldots + p_i, c \cdot \sqrt{p_1^2 + \ldots + p_i^2}\right). \tag{3}$$

Furthermore, the delay's indicators are random variables

$$\tilde{U}_i = \begin{cases} 0 & \text{for} \quad \tilde{C}_i \leqslant d_i, \\ 1 & \text{for} \quad \tilde{C}_i > d_i, \end{cases} \tag{4}$$

and the cost's factors are random variables

$$\tilde{T}_i = \begin{cases} 0 & \text{for} \quad \tilde{C}_i \leqslant d_i, \\ \tilde{C}_i - d_i & \text{for} \quad \tilde{C}_i > d_i. \end{cases} \tag{5}$$

For each permutation $\pi \in \Pi$ the cost in the random model is defined as a random variable:

$$\widetilde{W}(\pi) = \sum_{i=1}^{n} w_i \tilde{U}_i, \tag{6}$$

or

$$\widetilde{W}(\pi) = \sum_{i=1}^{n} w_i \tilde{T}_i. \tag{7}$$

3.2 Random due dates

Random due dates are represented by random variables with the normal distribution $\tilde{d}_i \sim N(d_i, c \cdot d_i)$, $i \in \mathcal{J}$ while processing times p_i and weights w_i are deterministic. Delay's indicators are random variables

$$\tilde{U}_i = \begin{cases} 0 & \text{dla} \quad C_i \leqslant \tilde{d}_i, \\ 1 & \text{dla} \quad C_i > \tilde{d}_i, \end{cases} \tag{8}$$

and the cost's factors are random variables

$$\tilde{U}_i = \begin{cases} 0 & \text{dla} \quad C_i \leqslant \tilde{d}_i, \\ C_i - \tilde{d}_i & \text{dla} \quad C_i > \tilde{d}_i. \end{cases} \tag{9}$$

Cost functions are the same as for the variant with random processing times.

4 Sampling

We have introduced sampling for the first time in [21] for the flowshop problem. In this paper we apply the same idea, but with additional optimizations and tailored for the single machine scheduling problem.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77980-1_45

Let's first recall that since permutations' costs defined in (6) and (7) are random variables, we need some way to be able to compare different solutions, so in each tabu search algorithm's iteration when we are testing different candidate solutions from the neighbourhood, we are able to find the best one and improve the current global best solution. The sampling method is a way how to make this comparison and the main idea is as follows.

Let us consider a problem instance $\{(\tilde{p}_i, w_i, d_i)\}$ (or $\{(p_i, w_i, \tilde{d}_i)\}$, respectively for uncertain due dates variant) and the examined candidate solution, a permutation π. As $\{\tilde{p}_i\}$ ($\{\tilde{d}_i\}$, respectively) are defined as random variables and we don't know the actual values that may come, the main idea of sampling is to generate samples of disturbed data based on $\{\tilde{p}_i\}$ ($\{\tilde{d}_i\}$, respectively) and simulate the potential different scenarios evaluating those disturbed candidate solutions. More formally we can describe the method as follows.

Algorithm 1: Sampling overview

1: Generate l vectors $\{(\overline{p}_i^k)\} = \{(\overline{p}_1^k, \ldots, \overline{p}_n^k)\}$ based on $\{\tilde{p}_i\}$ what gives l deterministic instances $\{(\overline{p}_i^k, w_i, d_i)\}$, $i \in \{1, \ldots, n\}$, $k \in \{1, \ldots, l\}$.
2: For each deterministic instance $(\overline{p}_i^k, w_i, d_i)$ a cost is calculated based on the candidate solution π. By that we obtain a sample $\{W_1^\pi, \ldots, W_l^\pi\}$.
3: We calculate a mean $\overline{x}$ and a standard deviation s from the sample which are used in the comparison criteria W by tabusearch.

The above listing is applicable for the random processing times variant. We can easily obtain a version for the random due dates by generating and using samples for $\tilde{d}_i$ instead of $\tilde{p}_i$.

In the basic scheme we use the fixed size of a sample (parameter l) and we investigate values dependent on the jobs' number: $0.25n, 0.5n, \ldots, 2n$.

Let's first specify the formula for criteria W. We investigated two main options: either only mean or some kind of combination of the mean and the standard deviation. During the analysis it turned out that the comparison criteria efficiency depends on the considered problem variant and even though for the most analyzed variants the best was $W = \overline{x}$, for instance for the one the best comparison criteria was $W = 50 \cdot \overline{x} + s$. However, as the differences were very small, for the simplicity reasons we assumed everywhere

$$W = \overline{x}.$$

Next, we wanted to learn something about the sample $\{W_1^\pi, \ldots, W_l^\pi\}$. Unfortunately it turned out that the distribution of the samples are not representing the normal distribution (according to the Shapiro-Wilk test), so we calculated how many different values are produced taking the large sample size (we took $10 \cdot n$).

Looking at Table 1 we can make a few observations. First, what is the most significant, the standard deviation from the number of different values is very

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77980-1_45

Table 1. Number of different values in samples obtained in Step 2 in Algorithm 1

		w_iU_i, $\tilde{p}_i$		w_iU_i, $\tilde{d}_i$		w_iT_i, $\tilde{p}_i$		w_iT_i, $\tilde{d}_i$	
N	Factor	Mean	StdDev	Mean	StdDev	Mean	StdDev	Mean	StdDev
40	0.05	1,9	1,0	4,4	2,3	36,8	2,1	37,0	1,6
	0.10	2,6	1,4	8,0	2,5	37,0	2,1	38,1	1,1
	0.15	3,4	1,7	11,2	2,4	37,2	2,0	39,2	0,7
	0.20	3,9	1,7	14,1	2,3	37,2	2,0	39,7	0,4
	0.25	4,7	1,8	16,7	2,2	37,2	2,0	39,9	0,3
	0.30	5,3	1,9	18,6	2,1	37,3	1,9	39,9	0,2
50	0.05	1,9	1,1	5,4	2,5	45,9	2,5	46,6	1,8
	0.10	2,7	1,6	10,4	2,8	46,2	2,5	48,5	1,0
	0.15	3,5	1,8	14,8	2,7	46,3	2,5	49,5	0,5
	0.20	4,2	2,0	18,4	2,5	46,4	2,4	49,8	0,3
	0.25	4,9	2,0	21,3	2,3	46,5	2,4	49,9	0,3
	0.30	5,6	2,1	23,7	2,2	46,6	2,3	49,9	0,2
100	0.05	3,6	2,6	13,0	4,8	90,7	4,4	92,5	3,3
	0.10	5,8	4,3	21,3	4,0	91,5	4,4	97,5	1,6
	0.15	7,4	4,6	27,9	3,3	92,0	4,5	99,5	0,6
	0.20	9,0	5,0	33,5	2,9	92,2	4,6	99,8	0,4
	0.25	9,9	4,7	38,4	2,6	92,3	4,6	99,8	0,4
	0.30	11,5	4,7	42,3	2,4	92,6	4,5	99,9	0,3

small among all considered cases and it varies from below 1 to 5 (however majority of values are around 1-2). Second, the average value for a specific problem variant (i.e. defined by the number of jobs, the random parameter and the cost criteria) is also quite stable. It is interesting that the average value is around number of jobs for the $\sum w_iT_i$ criteria and it is much smaller for the $\sum w_iU_i$ criteria. Moreover, for that criteria it is still much smaller both for random processing times and for random due dates.

We use those observations to define the first optimization rule to reduce the size of samples keeping the robustness coefficient on a good level.

Optimization 1 *Based on the number of different values in samples presented in Table 1 we state that for the problem variant with the cost criteria $\sum w_iT_i$, the size of the sample S is enough and there is no need to generate new samples when $|S| = n$.*

Unfortunately, based on the initial evaluation, for the cost variant $\sum w_iU_i$ the values are to small define the similar rule that brings any positive contribution.

Another approach for generating not too big samples is looking into the confidence intervals for average. Along with generating successive values we can analyze how confidence intervals change for sequence of samples. As we don't know the sample $W_1^\pi, \ldots, W_l^\pi$ distribution we apply the following variant of the theory

$$\overline{x} - \mu_\alpha \frac{s}{\sqrt{l}} < m < \overline{x} + \mu_\alpha \frac{s}{\sqrt{l}}$$

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77980-1_45

where l is a sample size (at least 30), $\overline{x}$ is the sample mean, s is the sample standard deviation and μ_α is the value of random variable $N(0,1)$ under the condition:

$$\Phi(\mu_\alpha) = 1 - \frac{\alpha}{2}$$

what, assuming the standard significance level $\alpha = 5\%$, provides $\mu_\alpha = 1,96$.

Now we are ready to introduce the second optimization rule.

Optimization 2 *Let $S_1, S_2, \ldots, S_k$ $(k > n/2)$ be a sequence of samples where $S_i = S_{i-1} \cup \{new\ element\}$ and $CI_1, CI_2, \ldots, CI_k$ be the sequence of the confidence intervals obtained from $S_1, S_2, \ldots$. Let $len(CI)$ denote the length of the confidence interval CI. We state that the size of the sample $|S| = k$ is enough and there is no need to generate new samples when the lengths of the last $n/2$ confidence intervals are more less the same, i.e.:*

$$\sum_{i=k,k-1,\ldots,k-n/2} |len(CI_i) - len(CI_{i-1})| \leqslant 3$$

Of course the values $n/2$ and 3 are arbitrary and they are based on some initial evaluation of different values.

Making an initial evaluation of confidence intervals for average we observed that the later iteration is, the smaller confidence intervals are. This observation triggered to introduce one more optimization rule which is very simple, but quite strong and by default it reduces the total sum of samples' sizes by half.

Optimization 3 *Let i be the iteration number in the tabu search algorithm execution and let's assume tabu search is executing n iterations. Then in the i-th iteration the sample size is determined by the following formula:*

$$|S| = (n - i + 1) \cdot 2;$$

Obviously the above formula can be easily adapted for any number of the tabu search iterations.

All the optimization rules are applied into the tabu search as follows:

- Optimization 3 is defining the upper bound for the sample size and provides the guarantee on the overall execution time.
- If the any of the rules defined by Optimization 1 and Optimization 2 is fulfilled, we stop generating more items in the sample before Optimization 3 rule holds. We can't estimate at which stage those rules are fulfilled as they are depended on the actual samples' values. However, the initial investigation shows that applying those makes a significant reduce in the total sum of samples' sizes.

5 Robustness of the solutions

In this section we present the results of the robustness property comparison between the tabu search method with and without the sampling method applied.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77980-1_45

All the tests are executed using 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 ([17]) where there are 125 examples for $n = 40$, 50 and 100 (in total 375 examples). For each example and each parameter $c = 0.05$, 0.1, 0.15, 0.2, 0.25 and 0.3 (expressing 6 levels of data disturbance) 100 randomly disturbed instances were generated according to the normal distribution defined in Section 3 (in total 600 disturbed instances per example). The full description of the method for disturbed data generation can be found in [3].

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

$$\delta = \frac{W - W^*}{W^*} \cdot 100\% \tag{10}$$

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 [3].

A classic version of the algorithm we denote by $\mathcal{AD}$, the one with applied sampling with the fixed sample size by $\mathcal{AP}^F$ and the one with applied sampling with the sample sized based on optimization rules by $\mathcal{AP}^O$.

5.1 Results

In Tables 2 and 3 we present a complete summary of results for cost criteria $\sum w_i U_i$ and in Table 4 and 5 we present a complete summary of results for cost criteria $\sum w_i T_i$. Values from columns $\mathcal{AD}$, $\mathcal{AP}^F$ and $\mathcal{AP}^O$ in all tables represent a relative distance between solutions established by a respective algorithm and the best known solution. The distance is based on (10) and it is the average of all solutions calculated for the disturbed data on a respective disturbance level expressed by the parameter c. For $\mathcal{AP}^F$ values are broken down for different sample sizes (which is based on number of jobs) and we can observe how those values are changing depending on the sample size. For the cost criteria $\sum w_i U_i$ the highlighted column $(0.75n)$ represents in the author's opinion the best choice when it comes to balance between the obtained robustness coefficient value and the sample size. The version of the algorithm with optimizations applied is presented only for random p_i and cost criteria $\sum w_i T_i$ as only for this variant optimizations brought the expected improvements.

Looking at the results we can quickly conclude that by applying sampling method we obtain significantly more robust solutions than in the deterministic approach. Column *IF (2n)* represents how much relatively presented approach

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77980-1_45

is better and actually the level of improvement depends on the problem variant. For random p_i improvements are enormous and they start from ca. 400% and they reaches values over 7000% for the cost criteria $\sum w_i U_i$ and they start from ca. 900% and reaches over 32000% for the cost criteria $\sum w_i T_i$. For random d_i those values are smaller, but still showing great improvements.

We can also notice that by applying optimizations 1, 2 and 3 we obtain a better ratio between the sample size and solution robustness. As in this approach there is no longer a fixed sample size, the actual sample size has been calculated during performing the tests and on average the sample size was $0,78n$ with very small deviations. Comparing values presented in the column $\mathcal{AP}^O$ to similar variant in the $\mathcal{AP}^F$ version $(0,75n)$ we observe that for all cases either values are similar or version with optimizations provides much better robustness keeping more less the same samples' size on average.

In general results follow the expectations, i.e. the bigger disturbance factor, the worse robustness coefficient, the bigger sample size, the better robust coefficient (to some degree), however, there are some exceptions from those rules what we plan to investigate further in the future research.

Table 2. Results for random p_i and $\sum w_i U_i$ cost criteria. Values are the relative errors between algorithm being compared to the best known solution

N	Factor	$\mathcal{AD}$	$\mathcal{AP}^F$								IF (2n)
			0,25n	0,5n	**0,75n**	1n	1,25n	1,5n	1,75n	2n	
40	0,05	58,0	28,1	24,7	**13,2**	13,4	12,2	12,1	11,8	12,0	382%
	0,1	172,7	68,2	33,5	**22,6**	20,6	18,7	17,6	16,6	16,9	920%
	0,15	505,0	281,3	128,9	**51,6**	48,0	52,7	50,3	51,9	52,2	867%
	0,2	827,7	320,0	228,0	**93,8**	86,8	78,8	79,0	69,0	71,6	1055%
	0,25	1213,6	781,8	197,4	**119,0**	153,9	133,1	132	98,7	97,2	1148%
	0,3	1299,4	578,4	473,8	**156,8**	162,2	163,2	145,7	164,8	110,3	1078%
50	0,05	70,7	21,2	17,4	**16,5**	15,7	15,4	14,3	13,4	12,6	461%
	0,1	610,0	82,3	105,3	**76,2**	56,1	54,8	39	39,8	29,0	2002%
	0,15	553,7	171,9	100,1	**81,4**	62,7	42,9	40,8	34,9	21,4	2484%
	0,2	2064,7	529,8	390,9	**181,0**	192,4	121,7	96,6	104	97,8	2011%
	0,25	2248,8	394,3	271,0	**200,3**	126,3	117,0	114,0	92,3	114,2	1868%
	0,3	1752,5	315,2	202,4	**111,4**	110,3	108,8	59,2	81,1	53,2	3195%
100	0,05	546,4	186,2	160,8	**110,8**	150,1	151,9	100,2	58,8	55,6	882%
	0,1	717,1	74,4	44,1	**33,5**	37,0	12,6	38,9	22,3	14,2	4932%
	0,15	1585,8	252,0	79,7	**58,4**	58,5	57,3	82,7	68,4	42,3	3648%
	0,2	1670,8	247,6	102,9	**120,5**	54,7	56,5	48,8	51,4	38,9	4199%
	0,25	1551,4	165,0	98,2	**70,6**	69,4	51,2	40,3	38,2	36,6	4133%
	0,3	2199,2	186,2	74,5	**48,9**	48,5	37,2	31,8	33,9	29,4	7377%

Finally, in Table 6 we present aggregated on all disturbance levels the percentage for how many instances $\mathcal{AP}^F$ gives not worse solution than $\mathcal{AD}$ assuming that the samples size S=2n. We can quickly see that for all problem variants (except the one for random p_i and cost criteria $\sum w_i T_i$) all results are almost

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77980-1_45

Table 3. Results for random d_i and $\sum w_i U_i$ cost criteria. Values are the relative errors between algorithm being compared to the best known solution

N	Factor	$\mathcal{AD}$	$\mathcal{AP}^F$								IF (2n)
			0,25n	0,5n	**0,75n**	1n	1,25n	1,5n	1,75n	2n	
40	0,05	1191,6	451,2	297,0	**197,2**	200,6	200,8	191,1	197,3	207,2	475%
	0,1	2891,4	1062,7	881,5	**709,1**	785,1	734,1	645,6	684,4	653,9	342%
	0,15	4460,0	1757,4	1193	**1152,6**	1128,3	1109	1132,8	1171,1	1095,6	307%
	0,2	2953,8	1480,0	1209,8	**1073,5**	1034,5	1055,4	1045,7	1016,3	1012,6	191%
	0,25	2457,6	1304,8	1153,3	**1129,5**	1100,4	1100,1	1111,4	1103,4	1087,4	126%
	0,3	1391,3	874,9	777,9	**732,8**	731,9	718,8	716,7	714,8	712,3	95%
50	0,05	3045,5	873,6	586,3	**512,9**	487,1	566,6	563,6	494,7	583,2	422%
	0,1	1128,6	472,6	370,5	**361,4**	323,7	317,9	338,6	321,5	323,2	249
	0,15	3387,1	1210,3	1112,1	**1067,2**	1000,9	955,5	928,4	918,9	963,1	251%
	0,2	2217,6	1171,6	943,7	**943,0**	908,5	892,4	876,5	858,3	877,1	152%
	0,25	2462,2	1501,2	1328,2	**1247,6**	1225,1	1195	1178,3	1181,6	1190,2	106%
	0,3	1758,1	1171,4	1036,6	**1002,8**	980,0	989,8	965,6	977,5	980,5	79%
100	0,05	3898,6	640,9	404,4	**373,9**	340,4	261,0	256,0	308,5	261,6	1390%
	0,1	1671,4	464,3	424	**379,8**	369,8	355,1	351,0	338,3	337,0	395%
	0,15	1537,7	576,4	530,6	**495,8**	477,5	485,9	497,1	504,7	473,5	224%
	0,2	1445,3	729,8	688,3	**676,7**	663,3	670,3	641,9	644,7	641,0	125%
	0,25	1180,5	735,7	700,8	**674,2**	664,6	659,2	650,5	652,4	640,3	84%
	0,3	784,6	548,8	525,5	**506,9**	503,2	503,7	499,7	497,0	494,1	58%

Table 4. Results for random p_i and $\sum w_i T_i$ cost criteria. Values are the relative errors between algorithm being compared to the best known solution

N	Factor	$\mathcal{AD}$	$\mathcal{AP}^O$	$\mathcal{AP}^F$							IF (2n)
			0,78n	0,25n	0,5n	0,75n	1n	1,25n	1,5n	1,75n	2n	
40	0,05	452,8	28,3	58,5	57,1	37,4	28,1	22,2	23,2	23,9	21,7	1982%
	0,1	851	31,3	250,6	163	133	51,2	121,8	42,2	37,5	35,2	2320%
	0,15	1532,6	49,2	571	226,8	94,9	101,1	70,9	79,3	49,9	52,2	2836%
	0,2	2012	98,3	762	311,2	126,1	108,8	97,6	105,2	86,3	68,5	2836%
	0,25	4693,6	900,9	5013,9	2073,4	1793,9	1737,6	1662,8	1628,9	456,4	458,4	923%
	0,3	5238,1	1014,7	6297,6	1440,9	1065,3	1044,7	983	932,2	280,5	279,8	1772%
50	0,05	373,1	10,4	49,6	45,1	21	10	9,7	9,9	9,8	9,5	3831%
	0,1	990,6	17,7	123,2	63,7	32,4	21,1	19,4	16,9	16,6	15,8	6186%
	0,15	2666	192,9	501,2	368,9	237,7	101,3	100,4	95,3	100,7	98,1	2617%
	0,2	4524	87,5	1068,9	484,4	148,4	92,3	89,6	83,1	74,9	73,9	6018
	0,25	12048,7	528,2	1921,3	1334,8	855,2	553,5	498	476,7	480,6	382	3053%
	0,3	7585,5	115,8	1099,6	633,6	283,2	113,5	154,5	119,1	117,7	44,5	16931%
100	0,05	830,9	6,9	19	7,5	7,1	7,7	7,5	7,5	6,9	7,3	11345%
	0,1	2452,8	13,7	39,5	13,3	12	8,9	10,8	5,3	8,8	6,4	37991%
	0,15	4095	62,1	222,2	56	63,3	70,6	46,2	31,7	40,1	42,2	9613%
	0,2	12436,8	78,8	336,3	78,3	144,9	67,3	59,6	64,9	42,3	38,5	32173%
	0,25	9104,2	114	305,4	161,2	135	80,5	85,4	82,6	75,6	59,4	15214%
	0,3	11229,8	93,3	501,7	167,2	145,7	129,3	123,6	113,3	86,5	87,3	12756%

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77980-1_45

Table 5. Results for random d_i and $\sum w_i T_i$ cost criteria. Values are the relative errors between algorithm being compared to the best known solution (all values for $\mathcal{AD}$ and $\mathcal{AP}^F$ should be multiplied by 10^3)

N	Factor	$\mathcal{AD}$ $(\cdot 10^3)$	$\mathcal{AP}^F$ $(\cdot 10^3)$								IF (2n)
			0,25n	0,5n	0,75n	1n	1,25n	1,5n	1,75n	2n	
40	0,05	10,5	5,4	4,2	1,8	1,9	1,8	1,8	1,7	1,5	595%
	0,1	26,4	11	7,5	7,0	7,1	6,5	6,6	6,4	6,2	325%
	0,15	35,2	18,8	15,1	14,7	14	13,6	13,2	13,5	13,2	165%
	0,2	56,2	32,6	26,4	25,5	24,6	24,1	24,3	23,7	23,1	142%
	0,25	51,9	32	28,5	27,8	27,8	26,8	27,1	27	26,4	96%
	0,3	21,1	15,5	14,2	13,3	13,3	13	13,1	12,9	13	62%
50	0,05	10,3	1,9	1,4	1,1	0,9	1,0	0,8	0,8	0,8	1164%
	0,1	68,2	23,9	18	15,6	16,6	14,7	15,3	15,8	14,4	372%
	0,15	55,1	27,7	22	21	19,8	19,8	19,5	19,6	19,8	178%
	0,2	103,6	54	46,3	42,8	42,6	42,3	43	42,3	42,1	146%
	0,25	221,2	136,3	121,4	116,3	116,4	115,8	114,1	114,2	114,7	92%
	0,3	25,8	16,5	14,4	14	13,7	13,8	13,6	13,5	13,6	89%
100	0,05	26	3,4	2,8	2,4	2,1	2,1	2	1,9	1,8	1319%
	0,1	97,4	28,6	24,2	22,6	21,8	21,8	21,3	20,8	20,1	384%
	0,15	576,1	283,7	270,6	254,1	246,2	244,9	241,7	242,4	241,1	138%
	0,2	2345,7	1506,1	1402	1332,5	1301,1	1308,8	1302,7	1296,5	1288,4	82%
	0,25	657,5	454,3	425,6	400,8	395,2	392,5	394,9	395,1	390,8	68%
	0,3	37,7	29,6	27,5	26,2	25,8	25,9	25,6	25,5	25,5	47%

95% and higher and the bigger n is, the better percentage we get. Moreover, for random d_i and cost criteria $\sum w_i U_i$ all values are very close or even equal to 100%. For random p_i and cost criteria $\sum w_i T_i$, even if a little smaller, we also get very strong result where all results are above 74%. This another perspective also shows predominance of the proposed sampling method.

Table 6. The percentage for how many instances $\mathcal{AP}^F$ gives not worse solution than $\mathcal{AD}$ assuming S=2n

n	p_i		d_i	
	$\sum w_i U_i$	$\sum w_i T_i$	$\sum w_i U_i$	$\sum w_i T_i$
40	94,7%	77,9%	98,8%	94,9%
50	96,8%	74,3%	99,2%	97,7%
100	99,9%	80,0%	100,0%	99,9%

6 Conclusions

In this paper we proposed the sampling method with a set of optimizations which can be applied to tabu search and other similar methods in order to improve

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77980-1_45

the robustness of solutions calculated in an uncertain environment modeled by random variables with the normal distribution. Based on the performed computational experiments we can conclude that the proposed method provides substantially more robust solutions than the ones obtained by the deterministic approach and the proposed optimization rules reduces the samples' sizes generated during the algorithm's execution

As there are several ways how the described method can be further investigated, we can see the following ways for continuation. Obviously, as there are several results which don't follow the expected trends, we plan to investigate the topic further and understand better the nature of those exceptions and by that, hopefully, improve the method and make it more tailored where applicable. There is also a question how strong is that method comparing to other methods solving the same problem based on stochastic and fuzzy description. The other area of investigation might be also to verify how much the input data distribution is important in the final results as the obtained samples doesn't reflect the distribution of the input data. Please note that this might be both advantage and disadvantage depending on the properties we would like to obtain in the end. Finally, this method is very universal and can be applied to many other types of optimization problems, so we plan also to follow this direction.

References

1. 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.
2. 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.
3. 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.
4. Bożejko W., Rajba P., Wodecki M. (2020) Robust Single Machine Scheduling with Random Blocks in an Uncertain Environment. In: Krzhizhanovskaya V. et al. (eds) Computational Science – ICCS 2020. ICCS 2020. Lecture Notes in Computer Science, vol 12143. Springer, Cham
5. Cai, X., & Zhou, S. (1999). Stochastic scheduling on parallel machines subject to random breakdowns to minimize expected costs for earliness and tardy jobs. Operations Research, 47(3), 422-437.
6. Cai, X., Sun, X., & Zhou, X. (2004). Stochastic scheduling subject to machine breakdowns: The preemptive-repeat model with discounted reward and other criteria. Naval Research Logistics (NRL), 51(6), 800-817.
7. Cai, X., & Zhou, X. (2005). Single-machine scheduling with exponential processing times and general stochastic cost functions. Journal of Global Optimization, 31(2), 317-332.
8. Cai, X., Wu, X., & Zhou, X. (2014). Optimal stochastic scheduling (Vol. 4). New York: Springer.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77980-1_45

9. Daniels, R. L., & Carrillo, J. E. (1997). β-Robust scheduling for single-machine systems with uncertain processing times. IIE transactions, 29(11), 977-985.
10. Dean, B. C. (2005). Approximation algorithms for stochastic scheduling problems (Doctoral dissertation, Massachusetts Institute of Technology).
11. Glazebrook, K. D. (2005). Optimal scheduling of tasks when service is subject to disruption: the preempt-repeat case. Mathematical Methods of Operations Research, 61(1), 147-169.
12. Hadar, J., & Russell, W. R. (1969). Rules for ordering uncertain prospects. The American economic review, 59(1), 25-34.
13. Jang, W., & Klein, C. M. (2002). Minimizing the expected number of tardy jobs when processing times are normally distributed. Operations Research Letters, 30(2), 100-106.
14. Kuo, C. Y., & Lin, F. J. (2002). Relative robustness for single-machine scheduling problem with processing time uncertainty. Journal of the Chinese Institute of Industrial Engineers, 19(5), 59-67.
15. Leus, R., & Herroelen, W. (2005). The complexity of machine scheduling for stability with a single disrupted job. Operations Research Letters, 33(2), 151-156.
16. Levy, H. (1992). Stochastic dominance and expected utility: Survey and analysis. Management science, 38(4), 555-593.
17. OR-Library
http://www.brunel.ac.uk/ mastjjb/jeb/info.html Accessed 11 May 2020
18. Pinedo, M., & Rammouz, E. (1988). A note on stochastic scheduling on a single machine subject to breakdown and repair. Probability in the Engineering and Informational Sciences, 2(1), 41-49.
19. Pinedo, M. L. (2016). Scheduling: Theory, Algorithms, and Systems, Springer International Publishing.
20. Rajba, P., & Wodecki, M. (2012). Stability of scheduling with random processing times on one machine. Applicationes Mathematicae, 2(39), 169-183.
21. Rajba, P., & Wodecki, M. (2017, June). Sampling Method for the Flow Shop with Uncertain Parameters. In IFIP International Conference on Computer Information Systems and Industrial Management (pp. 580-591). Springer, Cham.
22. Rothkopf, M. H. (1966). Scheduling independent tasks on parallel processors. Management Science, 12(5), 437-447.
23. Rothkopf, M. H. (1966). Scheduling with random service times. Management Science, 12(9), 707-713.
24. 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.
25. 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.
26. Vondrák, J. (2005). Probabilistic methods in combinatorial and stochastic optimization (Doctoral dissertation, Massachusetts Institute of Technology).
27. Yang, J., & Yu, G. (2002). On the robust single machine scheduling problem. Journal of Combinatorial Optimization, 6(1), 17-33.
28. Yue, F., Song, S., Jia, P., Wu, G., & Zhao, H. (2020). Robust single machine scheduling problem with uncertain job due dates for industrial mass production. Journal of Systems Engineering and Electronics, 31(2), 350-358.
29. 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 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77980-1_45