Single machine total weighted tardiness problem with budgeted uncertainty set on the processing times

Eduardo Henrique Reis Montalvão
Departamento de Engenharia de Produção – Universidade Federal Fluminense
Rua Passo da Pátria, 156, São Domingos, 24.220-240, Niterói, RJ
montalvao.edu@gmail.com

Artur Alves Pessoa
Departamento de Engenharia de Produção – Universidade Federal Fluminense
Rua Passo da Pátria, 156, São Domingos, 24.220-240, Niterói, RJ
artur@producao.uff.br

RESUMO

Este trabalho apresenta um algoritmo que combina busca local iterada e programação dinâmica para tratar a versão robusta do problema de sequenciamento em uma máquina com atraso total ponderado. Um conjunto de incertezas orçado foi considerado com o objetivo de modelar a incerteza sobre os tempos de processamento das tarefas, bem como para controlar o nível de conservadorismo sobre os dados de entrada. Foram realizadas simulações sobre os resultados obtidos a fim de comparar as abordagens robusta e determinística em relação aos sequenciamentos otimizados. O algoritmo robusto apresentou melhores resultados quando há pouco atraso nas tarefas de uma determinada sequência, apresentando piores resultados se o atraso médio das tarefas for significativamente alto. No entanto, ao se avaliar as caudas das distribuições dos custos simulados, a solução robusta decresce mais rápido do que a abordagem determinística em todos os casos observados.

PALAVRAS CHAVE. Sequenciamento Robusto, Máquina Simples, Conjunto de Incertezas Orçado.

Tópicos (Programação Matemática, Otimização Combinatória, Outras Aplicações em PO)

ABSTRACT

In this paper, we developed an algorithm based on both iterated local search and dynamic programming to deal with the robust version of the single machine total weighted tardiness-scheduling problem. The budgeted uncertainty set was used to model the uncertainty over the jobs processing times and to control the degree of conservatism on data. Simulations were performed to compare both the robust and the deterministic approaches on the schedule optimization. The robust algorithm presents better results if there are few delayed jobs in a sequence, being moderately worse in cases where the average jobs tardiness is significantly high. However, considering the tails of the simulated cost distributions, the robust solution presents faster decrease in all observed cases.

KEYWORDS. Robust Scheduling. Single Machine. Budgeted Uncertainty Set.

Paper topics (Mathematical Programming, Combinatory Optimization, Other Applications in OR)

1. Introduction

Robust Optimization (RO) problems constitute an intense research field given their theoretical and practical relevance. A wide variety of optimization problems are often stated considering precisely defined numerical parameters, which in many cases are only possible to be actually known by the time the solution is effectively implemented. Thereby, rather than exact values of the input data, one may define a set of all their possible realizations, named *scenario set* [Kasperski and Zieliński 2016] or originally, *uncertainty set* [Ben-Tal and Nemirovsky 1998]. In fact, by taking into account the uncertainty of the data, a solution is said to be feasible if it is feasible for all realizations of the data in a predetermined uncertainty set [Poss and Raack 2013].

The most relevant frameworks and developments in the field of RO are relatively recent, especially on the aspects related to tractability, conservativeness, probability guarantees and flexibility. The main advantage about RO rather than others approaches to deal with data uncertainty in optimization like *Stochastic Optimization* is the knowledge about probabilistic behavior of the data, which requires to meet restricted conditions [Ben-Tal *et al* 2009], besides to be difficult to measure and obtain the actual probability function of data in many practical applications [Bertsimas and Sim 2003]. An example of stochastic optimization approach can be found in [Ranjbar *et al* 2012], where the authors study the problem $P_m \parallel C_{max}$ when the processing times are stochastic according to a normal distribution. Their goal is to found a schedule that maximizes the customer service level, which is the probability of the makespan not exceeding the due date. They proposed a non-linear formulation and developed exact branch-and-bound algorithms to the problem, as well as provide upper / lower bounds and dominance rules to accelerate B&B procedures. Opposed to this idea of knowing the probability distribution of the data, RO states that the unknown parameters belong to determined uncertainty sets and imposes that constraints must be feasible for all parameters values in the uncertainty set [Poss 2013].

In this sense, [Kouvelis and Yu 1997] proposed three different robustness criteria in order to evaluate the objective function values of the solution under any likely input parameter scenario, which are: the absolute robustness or worst-case scenario, the maximal regret or robust deviation and the relative robust deviation. In this paper, we seek for a solution minimizing cost in a worst-case scenario. Regarding to the uncertainty sets, [Gorissen *et al* 2015] develop a guide to construct different kinds of sets from a given uncertainty constraint formulation and provide additional references in theoretical investigations to each referred set.

Such as other optimization problems, Scheduling has also a wide source of uncertainties that may affect real scheduling schemes. Machine failures and setup times, working environment conditions, labor productivity, changes in tasks' order or due dates among other unavailability factors might turn down the yields of an optimal schedule. Therefore, robust schedules have a practical interest since they provide hedge against system adverse conditions. In spite of its relevance in real problems, robust schedule has not been turned into practical frameworks whereas even simple scheduling problems become *NP*-hard as soon as the uncertainty set contains more than one scenario [Alalou and Croce 2008], [Daniels and Kouvelis 1995], [Yang and Yu 2002]. Complementarily, [Kouvelis and Yu 1997] has proved that robust discrete optimization problems are harder to solve, in the computational complexity sense, than their deterministic counterparts are. However, these negative aspects do not undermine efforts in providing solutions to robust schedules through the development of heuristic or linear approaches instead of strictly combinatorial algorithms.

[Alalou and Della Croce 2008] provide algorithmic and computational complexities for a set of well-known non-preemptive polynomial-time single machine-scheduling problem. By considering the scenario-based framework, in which all possible data realizations must be explicitly described, and the absolute robustness criterion, the authors develop complexity proofs for the robustness versions of the $1|prec|f_{max}$, $1||\sum w_j C_j$ and $1||\sum U_j$ (see [Graham *et al* 1979] for notations' description). For each problem, they considered the uncertainty on the parameters processing time, release date, due date and weight of job j under each possible scenario. In the same context, [Naji *et al* 2015] propose an artificial scenarios based approach to construct and identify robust schedules in an environment of unrelated parallel machines, which objective is minimizing makespan (C_{MAX}) when the splitting is tolerated. The uncertainty of processing time was taking by mean of discrete scenarios and the robustness was evaluated according to a worst-case strategy. [Yang and Yu 2002] evaluate the robust problem of minimizing the sum of completion time in a single machine environment considering a finite number of generic scenarios. The authors prove that this problem is NP-Complete and provide polynomial time heuristics and a dynamic programming to find optimal solutions. [Farias Jr. *et al* 2010] analyze the machine environment $1|| \sum w_j C_j$ and develop a large family of valid inequalities for the convex hull of the set of feasible robust schedules, considering the min-max criterion in a scenario strategy. [Mastrolilli *et al* 2013] adopt worst-case scenario strategy and model uncertainty in the single machine-scheduling problem minimizing the weighted sum of completion times. They presented a polynomial-time algorithm based on dynamic programming for when the number of scenarios and the values of the instances are bounded by some constant.

On the other hand, [Lu *et al* 2014] explore the uncertainty over the processing times in a constraint-based assumption, that is, each parameter may take any value between given lower and upper bounds, independent of the values of the other problem parameters. The authors proposed and compared the results of a mixed integer linear program (MILP), a simple iterative improvement (SII) heuristic and simulated annealing (SA) heuristic in min-max criteria to minimize the worst-case of C_{max} in a robust single machine-scheduling problem. They describe the uncertainty on the processing time by using intervals and adopted the budget parameter Γ as described in [Bertsimas and Sim 2004] aiming to control the degree of solution conservatism. In their MILP formulation, [Lu *et al* 2014] were capable to find the optimal solution for all instances with 50 and 100 jobs, whereas only five of eleven instances with 200 jobs had their optimal solutions obtained in the maximum computing time. Among heuristics, SA outperforms SII for all instances sizes, besides having found better solutions in nine of eleven instances with 200 jobs, surpassing the MILP yields. They also perform a simulation over the instances showing that, as Γ increases, the central tendency of simulated C_{max} is more significant, i.e., leads to less variability. In a similar way, we model the uncertain on the processing times in the constraint-based framework by considering the budgeted uncertainty set, which has advantages on both computational complexity and probabilistic guarantees, leading to a very large use of this set in discrete optimization problems under uncertainty [Pessoa *et al* 2015].

In this paper, the problem of minimizing the robust total weighted tardiness sum in a single machine environment is addressed. Encouraged by the positive results brought by [Bertsimas and Sim, 2003] we consider the robust counterpart of the $1|| \sum w_j T_j$ assuming the budgeted uncertainty set for the processing times. In this set, defined in [Bertsimas and Sim 2004], a positive integer Γ controls the degree of conservatism by limiting the number of jobs that might vary simultaneously from their respective nominal values. To solve this problem, we propose an algorithm based upon iterated local search combined with a dynamic programming procedure. Then, we compare the obtained robust schedules with the optimal deterministic schedules obtained through the algorithm stated in [Rodrigues *et al* 2008] using Monte Carlo simulation, in order to evaluate the yields of the robust approach. The optimality of the deterministic solutions to the problem considered in this paper has been proven by [Tanaka *et al* 2008].

Besides the introductory Section, this paper presents the following organization: Section 2 describes the single machine robust total weighted tardiness problem, as well as the notation to be used in all other parts of this work. Moreover, we provide a description on the budgeted uncertainty set and how we consider it in the processing time of the jobs. In Section 3, we present the min-max algorithm implemented to minimize the maximum total weighted tardiness given all possible realizations of the processing times in a given scenario. Additionally, we also describe the simulation performed for both deterministic and robust solutions to evaluate the quality of the robust scheduling. Section 4 is dedicated to the results and to the comparative analysis on the simulation output data. Finally, Section 5 points out some conclusions, further investigations and future developments regarding this problem.

2. Problem statement

Consider a set $J = \{1, \dots, n\}$ of jobs to be processed in a single machine without preemption. It is possible to process only one job at a time in the machine. All jobs are available for processing in time $t = 0$, each job j requires a processing time p_j, and has a due date d_j and a positive weight w_j. The tardiness of a job j in relation to its due date is defined as $T_j = \max\{C_j - d_j\}$, where C_j is the completion time of job j. The deterministic version of this scheduling problem consists in sequencing the jobs in the machine to minimize $\sum_{j=1}^{n} w_j T_j$. If π is a permutation of the jobs in the set S of feasible schedules and p is the n-tuple of the processing times, it is possible to rewrite this this problem as $\min_{\pi \in S} f(\pi, p)$, where $f(\pi, p) = \sum_{j=1}^{n} w_j T_j$. As proved in [Lawler 1977], this problem is strongly *NP-hard*. In case of $w_j = 1, j \in \{1, \dots, n\}$, the problem become *NP-hard* in the ordinary sense [Du and Leung, 1990], albeit may be solved in pseudo-polynomial time [Lawler 1977].

In this paper, we focus on robust counterpart of this problem, using the budgeted uncertainty set referred to as robust $1||\sum w_j T_j$ for short. Instead of providing optimal solutions given deterministic sets of input data, we are interested in model the uncertainty in processing times by a finite set $U \subset \mathbb{R}^n$. Consequently, the robust problem aims to minimize $f(\pi, p)$ over all $p \in U$ or, in a formal way, $\min_{\pi \in S} F(\pi, U)$ where $F(\pi, U) = \max_{p \in U} f(\pi, p)$ is the robust cost of permutation π. Based on the structured approach provided by [Bertsimas and Sim 2003], we consider the following uncertainty set definition in order to model the deviation on the processing times:

$$U^{\Gamma} \equiv \left\{p \in \mathbb{R}^n : \; p_j = \overline{p_j} + \delta_j \widehat{p_j}, j \in \{1, \dots, n\}, \delta_j \in [-1,1], \; \sum_{j=1}^{n} |\delta_j| \leq \Gamma \right\}.$$

Where $\overline{p_j}$ and $\widehat{p_j}$ are respectively referred to as the mean processing time of job j and its deviation, and the parameter Γ adjusts the robustness of the solution by controlling the number of processing times which vary at the same time to their maximum values. Based upon an independence assumption amongst variables, one may think intuitively that the chance of many coefficients change simultaneously in a real scheduling is quite unlikely. The Section 3 describes the heuristic procedures and the dynamic programming associated to deal with this min-max problem.

3. The Algorithm

Let $\pi = \{\pi_1, \pi_2, \dots, \pi_n\}$ be a permutation of the job indices $1, \dots, n$. Essentially, the procedure herein has two phases. The algorithm computes its first solution as a permutation π of jobs obtained according to the well-known Earliest Due Date (EDD). Then, in the minimization phase, the algorithm performs a local search over a neighborhood defined by Generalized Pairwise Interchanges (GPI) moves as stated in [Della Croce 1995]. These moves consist as both exchanges of pairs of jobs (not necessarily adjacent) and removal-and-insertion moves of a job. This local search phase occurs until no more improvements are possible in the current robust solution. This heuristic procedure was firstly proposed in [Rodrigues *et al* 2008] for parallel machine weighted tardiness problem ($P||\sum w_j T_j$) and we adapted it to the robust $1||\sum w_j T_j$.

Algorithm 1 presents the steps of the adapted local search heuristic. In the algorithm, π^* represents the best current solution. The function $opt(\pi)$ computes $F(\pi, U)$ for a given schedule π, using dynamic programming to find optimally the worst-case solution for this permutation. Then, it returns both the robust cost of π and the set K of Γ augmented jobs, i.e. jobs whose $\delta_j \neq 0$, used to compute this cost. Alternatively, $opt_{fix}(\pi', K)$ calculates the cost of schedule π' considering a fixed set K. Parameters N, q and L control the loops over a given initial permutation, the number of 2-change moves performed in each perturbation and the frequency in which a current solution is superseded for a totally random permutation, respectively. If there are not more improvements applying GPI moves, the algorithm executes q randomly perturbations through the pairwise moves over the current solution in order to run off from local optima. During the execution of the proposed algorithm, each new GPI move performed along π is firstly evaluated by computing $opt_{fix}(\pi', K)$ for the modified permutation π' using the set K of augmented jobs found in the calculation of $opt(\pi)$. This calculation is used as a fast underestimation of the value of $opt(\pi')$. If this cost does not improve the robust cost of π, then the move is rejected. Otherwise, an exact evaluation of the move is proceeded, causing a rejection of the move if the exact robust cost no longer improves the current solution. A preliminary experiment with an instance of 40 jobs showed that this mechanism reduces the overall running time by a factor of 43.

Algorithm 1 Iterated local search for robust $1||\sum w_j T_j$ (Adapted from [Rodrigues *et al* 2008])

- $j \leftarrow 0$; $l \leftarrow 0$; $\pi^* \leftarrow$ a permutation following the EDD rule
- $(C^*, K^*) \leftarrow opt(\pi^*)$
- **While $j < 3 * N$**
  - **If $l = 0$ or $l = L$ then**
    - $\pi \leftarrow$ a random permutation; $l \leftarrow 0$
  - $(C, K) \leftarrow opt(\pi)$; $C' \leftarrow -\infty$
  - **While $C' < C$**
    - **For each** GPI move over π
      - $\pi' \leftarrow$ GPI move over π
      - $(C', K) \leftarrow opt_{fix}(\pi', K)$
      - **If** $C' < C$ **Then** $(C', K') \leftarrow opt(\pi')$
      - **If** $C' < C$ **Then** $\pi \leftarrow \pi'$; $C \leftarrow C'$; $K \leftarrow K'$
    - **If** $C < C^*$ **Then** $\pi^* \leftarrow \pi'$; $C^* \leftarrow C$; $K^* \leftarrow K$
  - Perform q randomly chosen 2-change moves over π
  - $\pi \leftarrow$ 2-change move over π
  - $j \leftarrow j + 1$; $l \leftarrow l + 1$

Given a permutation π, the corresponding robust cost and the associated set K of augmented jobs are calculated as follows. For each $\kappa = 1, \ldots, n$, and $\gamma = 0, \ldots, \min\{\gamma, \kappa\}$, let $\hat{P}_{\kappa,\gamma}$ be the set of all possible sums of γ deviations from jobs $\pi_1, \ldots, \pi_\kappa$, that is,

$$\hat{P}_{\kappa,\gamma} = \left\{\textstyle\sum_{j \in S} \hat{p}_j \mid S \subseteq \{1, \ldots, \kappa\},\ |S| = \gamma\right\}.$$

Let also $\bar{p}(\pi, \kappa) = \sum_{l=1}^{\kappa} \bar{p}_{\pi_l}$. We define $f^*(\kappa, \gamma, s)$ as the largest cost of the partial schedule $(\pi_1, \ldots, \pi_\kappa)$ augmenting exact γ jobs so that the sum of deviations of the augmented jobs is exactly s. The proposed dynamic programming algorithm calculates the value of $f^*(\kappa, \gamma, s)$ for each $\kappa = 1, \ldots, n$, $\gamma = 0, \ldots, \min\{\gamma, \kappa\}$ and $s \in \hat{P}_{\kappa,l}$ using the following recursion:

$$f^*(\kappa,\gamma,s) = \begin{cases} w_\kappa * \max\{0\,, \bar{p}_{\pi_1} - d_{\pi_1}\} & if\ \kappa = 1, \gamma = 0\ and\ s = 0 \\ w_\kappa * \max\{0\,, \bar{p}_{\pi_1} + \hat{p}_{\pi_1} - d_{\pi_1}\} & if\ \kappa = 1, \gamma = 1\ and\ s = \hat{p}_{\pi_1} \\ \min\{f^*(\kappa-1, \gamma-1, s-\hat{p}_{\pi_\kappa})\,, f^*(\kappa-1,\gamma,s)\} + & \\ \quad w_\kappa * \max\{0, \bar{p}(\pi,\kappa) + s - d_{\pi_\kappa}\} & if\ \kappa > 1 \end{cases}$$

After each GPI move and before the computation of the robust cost by dynamic programming, the cost of the current solution is calculated considering the augmented jobs of the previous schedule as this gives a lower bound on the robust cost of the solution. If the obtained cost does not improve, the exact cost evaluation can be skipped. This procedure was adopted in order to avoid unnecessary dynamic programming calculations and, consequently, reduce the CPU time to compute an entire robust solution.

4. Computational experiments

The experiments consider only instances with $n = 100$ jobs. We set $N = 100,\ q = 3$, $L = 10$ and $\Gamma = 10$. We performed 300 perturbations over the current solutions in order to change the search space to the algorithm. For each new sequence randomly generated, we evaluate 1.565 x 10^6 GPI moves in the current sequence. We consider a scenario in which the jobs' processing times are augmented only 50% of to their nominal values, that is $\hat{p}_j = \lceil 0.5 * \bar{p}_j \rceil$. The reported times were obtained with an Intel® Core™ i7 3.4 GHz processor. The algorithm was developed in C++ language.

The experiments were performed on available instances for this problem in OR-Library. [Crauwels *et al* 1998] have generated this set of instances as follows: integers value for both processing times p_j and weights w_j were obtained from uniform distributions over [1, 100] and [1, 10], respectively. The relative range of due dates (RDD) and the average tardiness factor (TF) are likely to define the problem "hardness". That way, after having computed $P = \sum_{j=1}^{n} p_j$ and selected from the set $\{0.2, 0.4, 0.6, 0.8, 1.0\}$ the values for RDD and TF, all due dates d_j are generated from the uniform distribution $[P(1 - TF - RDD/2), P(1 - TF + RDD/2)]$. The authors generated five problems for all the 25 possible combinations of RDD and TF, yielding 125 instances to a given n. Since our objective is evaluating the behavior of the robust schedules against their deterministic versions regarding to tardiness, in this analysis we discard those groups of instances in which deterministic costs are zero.

The results obtained in this paper were compared to those reached by the deterministic optimal solution. Thus, in order to evaluate the quality of the robust scheduling, we run 10,000 simulations on both deterministic and robust solutions of each instance considering a random triangular distribution for processing times. In practice, this probability density function is more effective on turning into parameter estimates (minimum, maximum and most likely values) the decision-maker's subjective viewpoints than other probabilities functions [Stein and Keblis 2009]. The most likely values are the nominal processing times of the jobs, and the maximum and minimum values are symmetric in a range of [-50%; +50%] around $\bar{p}_j$.

Table 1 summarizes the results of robust and simulated costs to all groups of instances. All data are presented as average values of their respective groups. The percentages shown in table are deviations from the average cost of deterministic optimal solutions. The two last columns point out the average total CPU time and the average time to reach the best solution for each instance group, respectively. It is important to notice that the robust costs of the robust heuristic solutions are clearly better when compared to the robust costs of the deterministic optimal solutions, which may be observed in all instances groups evaluated. Indeed, those findings are relevant and are not so obvious since the optimization on robust schedules does not ensure optimality.

Another important observation refers to the simulated data. For instance, in an 85%-percentile, one may observe that 62% of groups presents their heuristic robust costs smaller than the robust cost of deterministic optimal solutions. For a 95%-percentile, 90% of the groups have their heuristic robust costs smaller, and at the 99%-percentile we reach 100% of the instances groups following this behavior. Despite the good results obtained, one could perceive the disadvantage in robust approach if the intended objective is to optimize average cost, once the deterministic simulated average cost had a better performance for most instances with larger values of TF. On the other hand, it is not possible to determine whether the underperformance of robust schedules is due to the "hardiness" associated to the instances or whether the robust method is not the best approach to this instance group. In contrast, groups 1, 6 and 15, in which

TF is smaller, the robust approach shows itself with advantages on deterministic solutions. Indeed, in real situations the decision-maker would be willing to deal with small delays rather than infeasible or too delayed schedules.

Figures 1 to 3 presents a graphical view of the simulations performed on both robust heuristic and deterministic solutions. A vertical straight dashed line on the same color of its respective distribution indicate the average value to each kind of simulated solution (robust heuristic and deterministic optimal). The abscissa axis represents the percentage distance from the average cost of the optimal deterministic solution for the indicated group. Thus, Figure 1 illustrates an instance group where the simulated values for the heuristic robust solutions are not only better in the average but also have a short tail than simulated costs of deterministic ones. Figure 2 points out a case where simulated costs of both solutions are almost equivalent. In Figure 3, it is possible to notice that the distributions of the simulated costs are very different from each other, prevailing a clear advantage to the heuristic when the distribution tails are confronted.

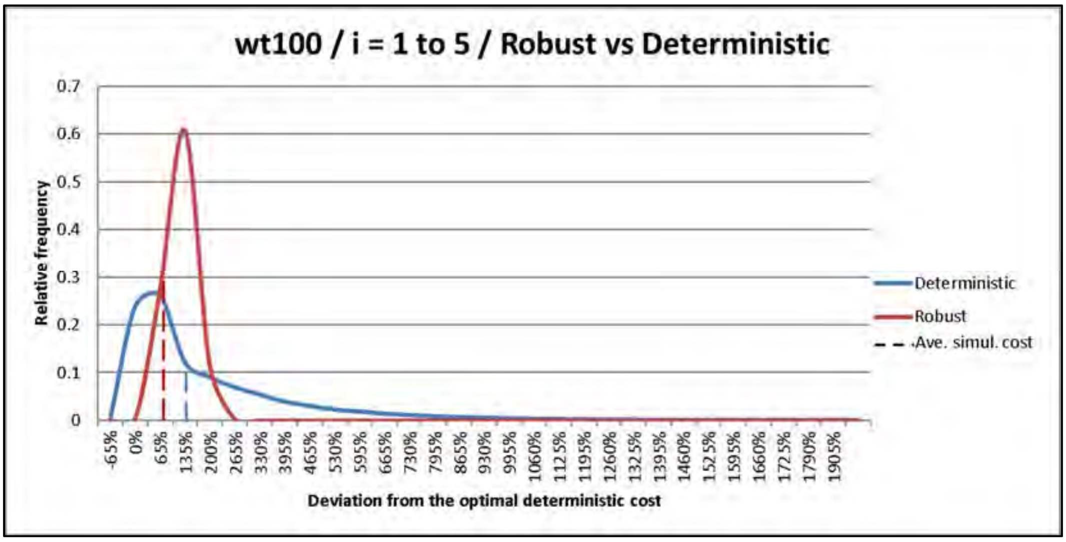

Figure 1 – Robust solution simulations for instance group 1.

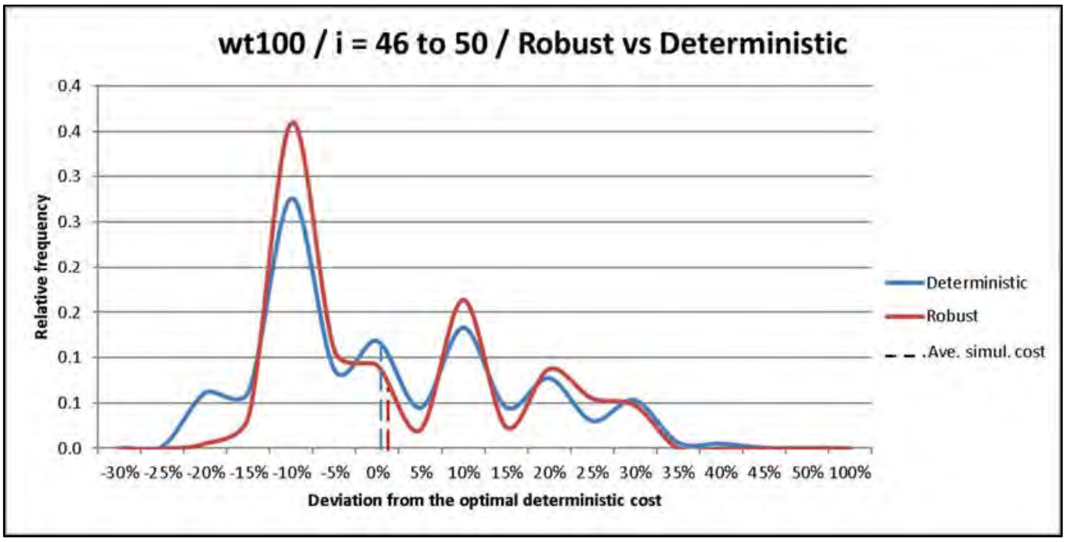

Figure 2 – Robust solution simulations for instance group 10.

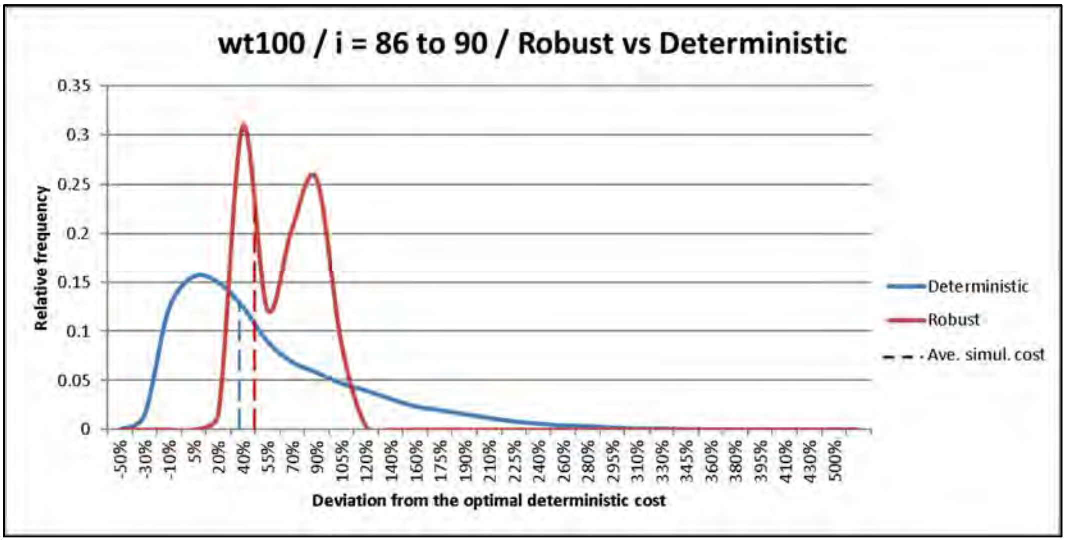

Figure 3 – Robust solution simulations for instance group 16.

5. Conclusion and future developments

In this paper we presented a heuristic approach to deal with the robust version of the single machine total weighted tardiness scheduling problem. The proposed algorithm considered the uncertain nature of data over the processing times and the budgeted uncertainty set was assumed to model the variability and to control the level of conservatism on the values. The heuristic presented good performance in computing robust costs when compared to those obtained straight from the optimal deterministic solutions. Furthermore, as denoted in the simulated results, the robust algorithm is better on average if there are few delayed jobs in a schedule and is moderately worse on most of the cases. However, by considering the probabilistic guarantee from which a solution will not surpass a certain limit, the simulated results evince the better performance of the robust schedule.

For the next steps, we will develop further investigations by considering slack in the processing times, by evaluating the simulation's behavior to other probability distributions rather than the triangular and by comparing the proposed algorithm with other similar approaches in literature for the same problem. Moreover, we intend to apply other values of parameter Γ in the uncertainty set to evaluate eventual improvements or worsening in the obtained solutions. Finally, we will try to derive a theoretical probabilistic guarantee on the deviation from the robust cost since the well-known theorem of [Bertsimas and Sim 2004] only applies to linear objective functions.

Table 1 - Results of robust scheduling on single machine total weighted tardiness problem (Average values for each instance group).

Instance group	TF	RDD	Instance Range	Deterministic Cost (DC)	Robust Determ. Cost (% on DC)	Robust Cost (% on DC)	Simulated Determ. Cost (% on DC)	Simulated Robust Cost (% on DC)	Simulation percentiles (Calculated on DC) Pr[85%] Determ.	Pr[85%] Robust	Pr[95%] Determ.	Pr[95%] Robust	Pr[99%] Determ.	Pr[99%] Robust	Time RO(s)	Time Best RO(s)
1	0.2	0.2	1-5	5,343.8	718.0%	**211.2%**	122.6%	**55.8%**	323.8%	**93.7%**	597.9%	**117.6%**	956.3%	**147.8%**	201.1	54.9
2	0.4	0.2	6-10	52,570.0	121.1%	**58.6%**	16.6%	**13.9%**	62.2%	**34.1%**	99.6%	**44.2%**	146.8%	**55.8%**	355.2	203.7
3	0.6	0.2	11-15	185,027.8	62.4%	**29.5%**	**7.3%**	10.3%	**33.2%**	34.0%	55.1%	**46.2%**	82.8%	**55.0%**	515.1	338.8
4	0.8	0.2	16-20	433,824.6	31.2%	**17.3%**	**3.3%**	6.8%	**25.6%**	31.6%	**38.3%**	39.0%	51.3%	**44.3%**	869.6	600.0
5	1.0	0.2	21-25	665,021.4	13.2%	**8.8%**	**1.3%**	1.6%	**31.3%**	33.7%	42.5%	**38.7%**	50.3%	**42.4%**	1,020.9	774.1
6	0.2	0.4	26-30	256.6	12,036.0%	**2,771.1%**	2,600.6%	**1,360.6%**	6,175.1%	**1,694.4%**	12,329.4%	**1,954.5%**	21,521.8%	**2,290.2%**	173.5	34.9
7	0.4	0.4	31-35	24,792.8	331.4%	**105.5%**	54.3%	**48.1%**	150.9%	**88.2%**	265.9%	**110.0%**	419.6%	**128.4%**	362.5	228.0
8	0.6	0.4	36-40	132,402.4	100.8%	**40.1%**	**13.6%**	18.5%	**53.5%**	55.7%	85.5%	**68.3%**	122.7%	**77.6%**	556.7	227.2
9	0.8	0.4	41-45	374,993.8	38.2%	**20.2%**	**4.5%**	7.6%	**27.2%**	29.3%	39.8%	**35.7%**	53.6%	**40.9%**	799.8	484.0
10	1.0	0.4	46-50	691,626.8	14.5%	**9.6%**	**1.6%**	2.3%	**18.7%**	20.7%	27.3%	**25.6%**	34.8%	**29.1%**	1,009.7	612.0
11	0.4	0.6	56-60	12,955.0	597.9%	**164.2%**	111.1%	**98.6%**	297.1%	**192.1%**	528.0%	**219.1%**	834.7%	**243.0%**	382.3	227.7
12	0.6	0.6	61-65	85,544.2	163.6%	**56.8%**	**24.7%**	35.0%	75.3%	**63.3%**	119.9%	**75.9%**	171.0%	**85.8%**	590.0	347.3
13	0.8	0.6	66-70	315,179.2	42.4%	**19.4%**	**5.5%**	11.1%	**36.8%**	40.3%	51.5%	**44.8%**	66.0%	**49.2%**	763.4	449.0
14	1.0	0.6	71-75	607,101.8	17.0%	**10.1%**	**1.8%**	3.3%	10.4%	**9.0%**	15.6%	**11.6%**	21.3%	**14.4%**	1,032.2	496.3
15	0.4	0.8	81-85	656.6	11,718.1%	**2,028.1%**	2,009.7%	**1,265.3%**	4,777.3%	**1,621.5%**	9,334.4%	**1,987.3%**	15,700.9%	**2,322.9%**	533.5	204.9
16	0.6	0.8	86-90	67,259.2	233.2%	**71.9%**	**38.8%**	49.4%	105.8%	**76.6%**	172.1%	**85.2%**	248.3%	**93.5%**	670.8	402.3
17	0.8	0.8	91-95	295,368.4	47.5%	**21.1%**	**6.6%**	9.2%	25.1%	**21.5%**	38.0%	**25.0%**	52.1%	**28.5%**	832.3	412.7
18	1.0	0.8	96-100	576,902.0	19.8%	**10.8%**	**2.0%**	2.8%	20.4%	**19.4%**	27.4%	**24.0%**	34.5%	**27.5%**	884.5	461.2
19	0.6	1.0	111-115	132,623.0	123.2%	**45.6%**	**21.0%**	25.8%	71.0%	**52.1%**	106.3%	**58.4%**	143.7%	**65.4%**	759.9	393.1
20	0.8	1.0	116-120	300,435.0	44.0%	**18.7%**	**6.4%**	13.0%	**28.5%**	38.3%	**41.9%**	44.9%	55.6%	**49.7%**	755.8	455.8
21	1.0	1.0	121-125	486,114.2	20.5%	**13.0%**	**3.1%**	4.0%	22.5%	**21.1%**	31.4%	**24.4%**	39.9%	**27.6%**	878.6	362.8

References

Aloulou, M. A. and Croce, F. D. (2008). Complexity of single machine scheduling problems under scenario-based uncertainty. *Operations Research Letters*, 36: 338–342.

Ben-Tal, A. and Nemirovsky, A. (1998). Robust convex optimization. *Mathematics of Operations Research*, 23: 769–805.

Ben-Tal, A., El Ghaoui, L. and Nemirovski, A. S. (2009). Robust optimization. Princeton University Press, United Kingdom.

Bertsimas, D. and Sim, M. (2003). Robust discrete optimization and network flows. *Mathematical Programming*, 98: 49–71.

Bertsimas, D. and Sim, M. (2004). The price of robustness. *Operations Research*, 52: 35–53.

Crauwels, H.A.J., Potts, C.N. and Van Wassenhove L.N. (1998). Local search heuristics for the single machine total weighted tardiness scheduling problem. *Informs Journal on Computing* 10: 341–350.

Daniels, R. L. and Kouvelis, P. (1995). Robust scheduling to hedge against processing time uncertainty in single-stage production. *Management Science*, 41: 363–376.

Della Croce, F. (1995). Generalized pairwise interchanges and machine scheduling. *European Journal of Operational Research*, 83: 310–319.

Du, J. and Leung J. (1990). Minimizing total tardiness on one processor is NP-Hard. *Mathematics of Operations Research*, 15: 483–495.

Farias Jr, I. R. D., Zhao, H. and Zhao, M. (2010). A family of inequalities valid for the robust single machine scheduling polyhedron. *Computers & Operations Research*, 37: 1610–1614.

Gorissen, B. L., Yanıkoğlu, I. and Den Hertog, D. (20015). A practical guide to robust optimization. *Omega*, 53: 124–137.

Graham, R., Lawler, E., Lenstra, J.K. and Rinnooy Kan A.H.G. (1979). Optimization and approximation in deterministic sequencing and scheduling: A survey. In: *Annals of Discrete Mathematics*, p. 287–326, North–Holland.

Kouvelis, P. and Yu, G. (1997). Robust Discrete Optimization and its Applications. Kluwer Academic Publisher, USA.

Lawler, E. (1977). A pseudopolynomial algorithm for sequencing jobs to minimize total tardiness. *Annals of Research Letters*, 1: 331–342.

Lu, C-C., Ying, K-C. and Lin, S-W. (2014). Robust single machine scheduling for minimizing total flow time in the presence of uncertain processing times. *Computers & Industrial Engineering*, 74: 102–110.

Naji, W., Espinouse, M.-L. and Cung V.-D. (2015). Towards a robust scheduling on unrelated parallel machines: A scenarios based approach, in Le Thi, H. A., Pham Dinh, T. and Nguyen, N. T. (eds.). Modelling, Computation and Optimization in Information Systems and Management Sciences, Advances in Intelligent Systems and Computing. Springer International Publishing.

Mastrollili, M., Mutsanas, N. and Svensson, O. (2013). Single machine scheduling with scenarios. *Theoretical Computer Science*, 477: 57–66.

Pessoa, A. A., Pugliese, L. D. P., Guerriero, F. and Poss, M. (2015). Robust constrained shortest path problems under budgeted uncertainty. *Networks*, 66: 98–111.

Poss, M. and Raack, C. (2013). Affine recourse for the robust network design problem: Between static and dynamic routing. *Networks*, 61: 180–198.

Poss, M. (2013). Robust combinatorial optimization with variable budgeted uncertainty. *4OR – A Quarterly Journal of Operations Research*, 11: 75–92.

Ranjbar, M., Davari, M. and Leus, R. (2012). Two branch-and-bound algorithms for the robust parallel machine-scheduling problem. *Computers & Operations Research*, 29: 1652–1660.

Rodrigues, R., Pessoa, A. A., Uchoa, E. and Aragão, M. P. (2008). Heuristic algorithm for the parallel machine total weighted tardiness scheduling problem. *Relatórios de Pesquisa em Engenharia de Produção*, 8(10): 1–11.

Soyster, A. L. (1973). Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research*, 21: 1154–1157.

Stein, W. E. and Keblis M. F. (2009). A new method to simulate triangular distribution. *Mathematical and Computer Modelling*, 49: 1143–1147.

Tanaka, S., Fujikuma, S., Araki, M. (2008). An exact algorithm for single-machine scheduling without machine idle time. *Journal of Scheduling*, 12(6): 575-593.

Yang, J. and Yu, G. (2002). On the robust single machine scheduling problem. *Journal of Combinatorial Optimization*, 6:17–33.