β-Robust Solutions for the Fuzzy Open Shop Scheduling

Juan José Palacios[1], Inés González-Rodríguez[2], Camino R. Vela[1], and Jorge Puente Peinador[1]

[1] Department of Computer Science, University of Oviedo, Spain
{palaciosjuan,puente,crvela}@uniovi.es http://di002.edv.uniovi.es/iscop
[2] Department of Mathematics, Statistics and Computing,
University of Cantabria, Spain
ines.gonzalez@unican.es

Abstract. We consider the open shop scheduling problem with uncertain durations modelled as fuzzy numbers. We define the concepts of necessary and possible β-robustness of schedules and set as our goal to maximise them. Additionally, we propose to assess solution robustness by means of Monte Carlo simulations. Experimental results using a genetic algorithm illustrate the proposals.

1 Introduction

Scheduling problems form an important body of research since the late fifties, with multiple applications in industry, finance and science [1]. In particular, the open shop scheduling problem models situations frequently appearing in testing components of electronic systems, in general repair facilities when repairs can be performed in an arbitrary order, as well as in certain medical diagnosis procedures. However, the open shop is NP-complete for a number of resources $m \geq 3$ and has a significantly large search space. Specific and efficient methods to solve it are necessary but still scarce, despite their increasing presence in the recent literature [2].

Traditionally, it has been assumed that problems are static and certain: all activities and their durations are precisely known in advance and do not change as the solution is being executed. However, for many real-world scheduling problems design variables are subject to perturbations or changes, causing optimal solutions to the original problem to be of little or no use in practice. Therefore, a common practical requirement is to obtain so-called *robust solutions*, which should still work satisfactorily when design variables change slightly, for instance, due to manufacturing tolerances.

A source of changes in scheduling problems is uncertainty in activity durations. There exists great diversity of approaches to dealing with this kind of uncertainty [3]. Perhaps the best-known is stochastic scheduling, although fuzzy sets and possibility theory provide an interesting alternative, with a tradeoff between the expressive power of probability and their associated computational

A. Laurent et al. (Eds.): IPMU 2014, Part I, CCIS 442, pp. 447–456, 2014.

complexity and knowledge demands. Indeed, fuzzy sets have been used in different manners in scheduling, ranging from representing incomplete or vague states of information to using fuzzy priority rules with linguistic qualifiers or preference modelling (cf. [4]).

The approaches to proactive robustness are several and varied. For instance, in stochastic settings, heuristic rules are used to include time buffers or slacks between activities in a baseline schedule [5]. In combinatorial optimisation, min-max or min-max regret criteria are applied to construct solutions having the best possible performance in the worst case [6], an approach already translated to the fuzzy framework [7],[8]. However, this may be deemed as too conservative when the worst case is not crucial and an overall acceptable performance is preferred. This is the basis for the β-robustness approach in stochastic scheduling [9], taking into account the subjective aspect of robustness through a target level specified by the decision maker so the goal is to maximise the likelihood that a solutions's actual performance is not worse than the target. This technique can be related to chance-constrained programming in linear optimisation, which has also been extended to fuzzy and fuzzy stochastic coefficients (cf. [10]).

The open shop problem with uncertainty constitutes a relatively new and complex research line. While there are many contributions to solve fuzzy job shop problems (we can cite, among others, [11],[12], [13] or [14]), the literature on fuzzy open shop is still scarce. Among the few existing proposals, a heuristic approach is proposed in [15] to minimise the expected makespan for an open shop problem with stochastic processing times and random breakdowns; in [16] the expected makespan of an open shop with fuzzy durations is minimised using a genetic algorithm hybridised with local search. Finally, in the framework of multiobjective approach, a possibilistic mixed-integer linear programming method is proposed in [17] for an OSP with setup times, fuzzy processing times and fuzzy due dates to minimise total weighted tardiness and total weighted completion times and in [18] a goal programming model based on lexicographic multiobjective optimisation of both makespan and due-date satisfaction is adopted and solved using a particle swarm algorithm.

In this paper, we intend to advance in the study of the fuzzy open shop problem, and in particular, in the search of robust solutions. In analogy to stochastic scheduling, we shall define the concepts of β_*-robust and β^*-robust schedules in terms of necessity and possibility, so the objective will then be to maximise such robustness. Then, we shall propose to perform an additional analysis of the obtained solutions using a Monte-Carlo simulation method based on the semantics of fuzzy schedules from [13]. Finally, we adapt the genetic algorithm from [19] and provide experimental results to illustrate our proposals.

2 The Fuzzy Open Shop Problem

The *open shop scheduling problem*, or *OSP* in short, consists in scheduling a set of n jobs $J_1, \dots, J_n$ to be processed on a set of m physical resources or machines $M_1, \dots, M_m$. Each job consists of m tasks or operations, each requiring the exclusive use of a different machine for its whole processing time without preemption, i.e. all operations must be processed without interruption. In total, there are mn operations, $\{o_{ij}, 1 \leq i \leq n, 1 \leq j \leq m\}$. A solution to this problem is a *schedule*–an allocation of starting times for all operations– which is *feasible*, in the sense that all constraints hold, and is also optimal according to some criterion, most commonly minimising the makespan C_{max}, that is, the completion time of the last operation (and therefore, of the whole project).

In real-life applications, it is often the case that the exact time it takes to process a task is not known in advance. However, based on previous experience, an expert may have some knowledge (albeit uncertain) about the duration. The crudest representation of such knowledge would be a human-originated confidence interval; if some values appear to be more plausible than others, then a natural extension is a fuzzy interval or fuzzy number. The simplest model is a *triangular fuzzy number* or *TFN*, denoted $A = (a^1, a^2, a^3)$, given by an interval $[a^1, a^3]$ of possible values and a modal value $a^2 \in [a^1, a^3]$, so its membership function takes a triangular shape:

$$\mu_A(x) = \begin{cases} \frac{x-a^1}{a^2-a^1} & : a^1 \leq x \leq a^2 \\ \frac{x-a^3}{a^2-a^3} & : a^2 < x \leq a^3 \\ 0 & : x < a^1 \text{ or } a^3 < x \end{cases} \tag{1}$$

Triangular fuzzy numbers and more generally fuzzy intervals have been extensively studied in the literature (cf. [20]) and widely used in scheduling.

In the open shop, we essentially need two operations on fuzzy numbers, the sum and the maximum. For any bivariate continuous isotonic function f and any two fuzzy numbers A and B, if $A_\alpha = [\underline{a}_\alpha, \overline{a}_\alpha]$ denotes the α-cut, the result $f(A, B)$ is a fuzzy number F such that $F_\alpha = [f(\underline{a}_\alpha, \underline{b}_\alpha), f(\overline{a}_\alpha, \overline{b}_\alpha)]$, that is, computing the function is equivalent to computing it on every α-cut. In particular, this is true for both the addition and the maximum. However, evaluating two sums or two maxima for every value $\alpha \in [0, 1]$ is cumbersome if not intractable in general. For the sake of simplicity and tractability of numerical calculations, we follow [11] and approximate the results of these operations by a linear interpolation evaluating only the operation on the three defining points of each TFN (an approach also taken, among others, in [12], [18] or [21]). The approximated sum coincides with the actual sum, so for any pair of TFNs A and Bs:

$$A + B = (a^1 + b^1, a^2 + b^2, a^3 + b^3) \tag{2}$$

Regarding the maximum, for any two TFNs A, B, if $F = \max(A, B)$ denotes their maximum and $G = (\max\{a^1, b^1\}, \max\{a^2, b^2\}, \max\{a^3, b^3\})$ the approximated value, it holds that $\forall \alpha \in [0, 1]$, $\underline{f}_\alpha \leq \underline{g}_\alpha, \overline{f}_\alpha \leq \overline{g}_\alpha$. The approximated maximum G is thus a TFN which artificially increases the value of the actual maximum F, although it maintains the support and modal value. This approximation can be trivially extended to the case of more than two TFNs.

Given a task processing order π, the schedule (starting and completion times of all tasks) may be computed as follows. For every task x with processing time

p_x, let $S_x(\pi)$ and $C_x(\pi)$ denote respectively the starting and completion times of x, let $PM_x(\pi)$ and $SM_x(\pi)$ denote the predecessor and successor tasks of x in the machine sequence provided by π, and let $PJ_x(\pi)$ and $SJ_x(\pi)$ denote respectively the predecessor and successor tasks of x in the job sequence provided by π. Then the starting time $S_x(\pi)$ of x is a TFN given by:

$$S_x(\pi) = \max(S_{PJ_x(\pi)} + p_{PJ_x(\pi)}, S_{PM_x(\pi)} + p_{PM_x(\pi)}), \tag{3}$$

Clearly, $C_x(\pi) = S_x(\pi) + p_x(\pi)$. If there is no possible confusion regarding the processing order, we may simplify notation by writing S_x and C_x. The completion time of the last task to be processed according to π thus calculated will be the makespan, denoted $C_{max}(\pi)$ or simply C_{max}. We obtain a *fuzzy schedule* in the sense that the starting and completion times of all tasks and the makespan are fuzzy intervals, interpreted as possibility distributions on the values that the times may take. However, notice that the task processing ordering π that determines the schedule is deterministic; there is no uncertainty regarding the order in which tasks are to be processed.

3 Robust Schedules

The usual objective of deterministic scheduling of minimising the makespan could, in principle, be translated to the fuzzy framework as minimising the expected makespan $E[C_{max}]$. However, minimising the expected makespan may be criticised, since it reduces the information provided by a fuzzy makespan to a single value, thus loosing part of the information. Neither does it address the practical requirement of solution robustness. Therefore we propose instead to find the equivalent to what has been called in the stochastic framework β-robust schedules [9,22], schedules with a certain confidence level that the performance will be within a given threshold.

The membership function μ_D of a fuzzy duration D may be interpreted as a possibility distribution on the real numbers [23,24], representing the set of more or less plausible, mutually exclusive values of a variable y (in our case, the underlying uncertain duration). Since a degree of possibility can be viewed as an upper bound of a degree of probability, μ_D also encodes a whole family of probability distributions.

It is well known that for a given interval $I \subseteq \mathbb{R}$, the *possibility* and *necessity measure* that $D \in I$ are respectively given by $\Pi(D \in I) = \sup_{y \in I} \mu_D(y)$ and $N(D \in I) = \inf_{y \in I} 1 - \mu_D(y) = 1 - \sup_{y \notin I} \mu_D(x) = 1 - \Pi(D \notin I)$, so necessity and possibility are dual measures which provide lower and upper bounds for the probability that y is in I given the information 'y is D': $N(D \in I) \leq Pr(D \in I) \leq \Pi(D \in I)$. In particular, for $A = (a^1, a^2, a^3)$ a TFN, the *necessity* and the *possibility* that A is less than a given real number r are given by:

$$N(A \leq r) = \begin{cases} 0, & r \leq a^2, \\ \frac{r-a^2}{a^3-a^2}, & a^2 \leq r \leq a^3, \\ 1, & a^3 < r \end{cases} \Pi(A \leq r) = \begin{cases} 0, & r \leq a^1, \\ \frac{x-a^1}{a^2-a^1}, & a^1 \leq r \leq a^2, \\ 1, & a^2 < r \end{cases} \tag{4}$$

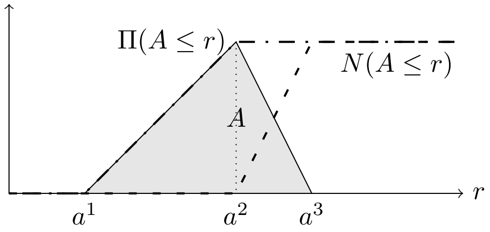

Fig. 1. Necessity $N(A \leq r)$ and possibility $\Pi(A \leq r)$ for varying values of $r \in \mathbb{R}$

Clearly, for any value r, $N(A \leq r) \leq \Pi(A \leq r)$. Figure 1 illustrates both measures.

Assuming we have a target or threshold for the makespan $C^\star$, we may want to maximise the confidence that C_{max} will "for sure" be less than this threshold. In our setting, this means to maximise the necessity degree that C_{max} is less than $C^\star$.

Definition 1. *A schedule with makespan C_{max} is said to be* necessarily β_*-robust *w.r.t. a threshold $C^\star$ if and only if $\beta_* = N(C_{max} \leq C^\star)$. Analogously, the schedule is said to be* possibly β^*-robust *w.r.t. $C^\star$ iff $\beta^* = \Pi(C_{max} \leq C^\star)$. β_* and β^* are respectively the* degrees of necessary and possible robustness *w.r.t. the threshold $C^\star$.*

Clearly, if a schedule is β^*-robust and β_*-robust w.r.t. the same threshold, and $\beta = Pr(C_{max} \leq C^\star)$, we have that $\beta_* \leq \beta \leq \beta^*$.

The degree of necessary robustness represents the degree of confidence that the makespan will certainly be less than the threshold. In the following, we will consider that the objective will be to find a schedule maximising this confidence level, so the resulting problem may be denoted $O|fuzz\,p_i|\beta_*(C^\star)$ following the three-field notation [25]. Obviously, by maximising the degree of necessary robustness we are also maximising the possible robustness of the schedule.

4 Monte-Carlo Simulation Assessment

Assuming we have solved the above optimisation problem and have obtained a β_*-robust schedule w.r.t. $C^\star$, is there a means of assessing the actual robustness of such schedule? In other words, does the concept of β_*-robustness really capture the desired high-level characteristic of robustness? Here, we propose a method for an empirical assessment of solutions to the $O|fuzz\,p_i|\beta_*(C^\star)$ problem, based on using Monte-Carlo simulations and inspired by the semantics for fuzzy schedules from [13].

In [13] fuzzy schedules are interpreted as *a-priori solutions*, found when the duration of tasks is not exactly known. In this setting, it is impossible to predict what the exact time-schedule will be, because it depends on the realisation of the tasks' durations, which is not known yet. Each fuzzy schedule corresponds to a precise ordering of tasks and it is not until tasks are executed according to this ordering that we know their real duration and, hence, know the exact schedule, the *a-posteriori solution* with exact job completion times and makespan. The practical interest of a solution to the fuzzy open shop would then lie in the ordering of tasks that it provides a priori using the available incomplete information, which should ideally yield good schedules in the moment of its practical use. Its behaviour could therefore be evaluated on a family of K deterministic open shop problems, representing K possible a posteriori realisations of the fuzzy problem. These may be simulated by generating an exact duration $\hat{p}_x$ for each task at random according to a probability distribution which is coherent with the fuzzy duration p_x.

Given a solution to the fuzzy open shop, consider the task processing order π it provides. For a deterministic version of the problem, let $\hat{\boldsymbol{p}}$ be the matrix of precise durations, such that $\hat{p}_{ij}$, the a-posteriori duration of operation o_{ij}, is coherent with the constraint imposed by the fuzzy duration p_{ij}. The ordering π can be used to process the operations, where the duration of each operation o_{ij} is taken to be $\hat{p}_{ij}$. This yields a time-schedule with precise starting and completion times for all tasks and, in particular, a real makespan $C_{max}(\pi, \hat{\boldsymbol{p}})$, which may be under or above the threshold $C^{\star}$. If instead of a single deterministic instance we consider the whole family of K deterministic problems, each with a duration matrix, we obtain K makespan values; the proportion κ of those values among the K which are actually below the threshold $C^{\star}$ gives us an empirical measure of the robustness of π. If the β_*-robustness is a good measure of the schedules robustness, then a schedule with high β_* should correspond to a high κ.

5 Genetic Algorithm

To solve the optimisation problem $O|fuzz\ p_i|\beta_*(C^{\star})$, we propose to use the genetic algorithm (GA) from [19]. In principle, to do so it would only be necessary to substitute the fitness function therein by the β_*-robustness degree of the schedule represented by each chromosome. However, such a straightforward approach has a serious drawback: the initial population, generated at random, consists of poor schedules, with high makespan values which, most likely, will yield a value $\beta_* = 0$ for any reasonable threshold $C^{\star}$, thus making it impossible for the GA to evolve.

In order to overcome this drawback, we propose to adapt the GA to use an "adaptive" threshold, with successive approximations $C^{\star}_0 > C^{\star}_1 > \dots$ until $C^{\star}$ is reached. Given the first population, a first threshold $C^{\star}_0$ is obtained as the most pessimistic value of the best makespan in this population, making sure that there will be chromosomes with non-zero fitness values (in fact, the individual with the best makespan will have fitness 1), thus allowing the GA to evolve. The threshold can then be updated along successive generations with new more demanding values $C^{\star}_g$ linearly decreasing from $C^{\star}_0$ to $C^{\star}$. This smooth updating allows the GA to evolve to robust solutions w.r.t. iteratively smaller thresholds. Finally, in order to give the GA the chance of obtaining β_*-robust solutions w.r.t.

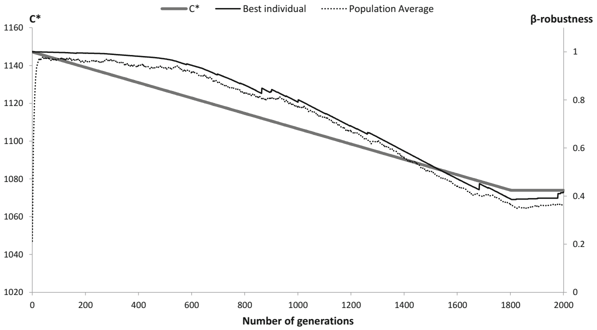

Fig. 2. Evolution of the best and mean solution of GA and the $C_g^\star$ values for the instance j8-per10-1 averaged across 10 runs

$C^\star$, in the last generations of the algorithm the $C^\star$ value is used to compute the β_*-robustness degree as fitness function.

6 Experimental Results

For the experimental study we shall use the test bed given in [16], where the authors follow [11] to generate a set of fuzzy instances from well-known open shop benchmark problems. Given a deterministic instance, each deterministic processing time t is transformed into a symmetric fuzzy processing time $p(t)$ with modal value $p^2 = t$ and where values p^1, p^3 are taken as random integer values such that the resulting TFN is symmetric w.r.t. p^2 and its maximum range of fuzziness is 30% of p^2. The original benchmark consists of 6 families, denoted *J3, J4,..., J8*, of sizes from 3×3 to 8×8, containing 8 or 9 instances each. In this work we only consider the largest instances: i.e. the 9 instances of size 7×7 and the 8 instances of size 8×8.

In a real problem, the target value $C^\star$ would be provided by an expert with a reasonable knowledge of the problem. However, as we are using synthetic problems, such expert is not available and in consequence the target values must be set following some criterium. In our case, we have taken the best known solution $A = (a^1, a^2, a^3)$ for each instance [18] and we have defined $C^\star = a^2 + TF \times (a^3 - a^2)$, where TF is a given tightness factor. To obtain the best possible performance, a parametric analysis (not reported here due to the lack of space) was conducted using $TF = 0.75$. The resulting parameter values were: population size=100, crossover probability=0.7, mutation probability=0.05, and number of generations=2000 from which the last 200 use the $C^\star$ value. The GA has been run with these parameters 10 times on each problem instance. Figure 2 shows the convergence pattern for *j8-per10-1*, one of the largest instances, with the remaining instances presenting a similar behaviour. The figure shows the evolution along 2000 generations of the fitness value of the best individual together with the mean fitness of the population and the $C_g^\star$ threshold used at each generation g to compute the β_*-robustness. As expected, we can appreciate that the algorithm's behaviour is sensitive to the $C^\star_g$ values. Initially, a less-demanding $C^\star_0$ allows the GA to evolve properly so the average quality of the population improves. After the first generations, $C^\star_g$ decreases becoming more demanding and in consequence, despite the fact that the population continues evolving, the robustness deteriorates for some generations (notice that for the same solution, its robustness value is dependent on the threshold $C^\star$). Finally, in the last iterations the goal $C^\star$ remains fixed and robustness values improve again thanks to the algorithm's evolution.

To empirically measure the robustness of the schedules obtained by the GA, we follow the Monte-Carlo simulation assessment introduced in Section 4 and generate samples of $K = 1000$ deterministic problems for each fuzzy instance, with random a-posteriori durations following a probability distribution which is coherent with the TFNs that model the fuzzy durations. We have then obtained the makespan values for each deterministic problem using the ordering provided by the GA on the fuzzy instanc, and we have finally computed the proportion κ out of the K deterministic makespan values which are below the threshold $C^\star$. Table 1 shows, for each fuzzy instance, the threshold $C^\star$, the β_* value of the best, average and worst solution across 10 runs, the CPU time (Runtime) in seconds, and the proportion κ obtained in the simulation for the best solution (κ-robustness). We can appreciate that even for the worst solutions $\beta_* > 0$, so in all solutions the possible β^*-robustness is 1. Moreover, the obtained "real" robustness values (κ) are always 1 or very close to 1, even in those instances

Table 1. Results of the GA and the *a-posteriori* analysis across the largest instances of the Brucker data set

Instance	$C^\star$	β_*-robustness Best	 Average	 Worst	Runtime	κ-robustness
j7-per0-0	1105.25	0.3682	0.2258	0.1082	9.2s.	0.9830
j7-per0-1	1140.00	0.7439	0.6231	0.4789	9.0s.	1.0000
j7-per0-2	1136.75	0.5493	0.4364	0.3147	9.0s.	0.9980
j7-per10-0	1099.50	0.7500	0.5294	0.2895	8.6s.	1.0000
j7-per10-1	1075.75	0.7319	0.5383	0.1972	8.9s.	1.0000
j7-per10-2	1079.75	0.6408	0.4701	0.2351	9.2s.	1.0000
j7-per20-0	1028.50	0.6477	0.5667	0.4524	9.0s.	1.0000
j7-per20-1	1075.00	0.7541	0.5041	0.1509	9.0s.	1.0000
j7-per20-2	1059.50	0.6288	0.3657	0.1508	9.1s.	1.0000
j8-per0-1	1106.50	0.3750	0.2164	0.0473	13.6s.	0.9190
j8-per0-2	1115.75	0.4696	0.2561	0.1735	13.8s.	0.9630
j8-per10-0	1110.00	0.9054	0.5723	0.3273	13.5s.	1.0000
j8-per10-1	1074.00	0.5714	0.4162	0.2692	13.7s.	0.9830
j8-per10-2	1059.25	0.4179	0.2601	0.0753	13.9s.	0.9850
j8-per20-0	1062.75	0.6433	0.4975	0.3994	13.6s.	1.0000
j8-per20-1	1048.00	0.7164	0.5445	0.4133	13.6s.	1.0000
j8-per20-2	1059.00	0.5444	0.4451	0.3299	13.6s.	0.9960

where β_* is smaller (e.g. j7-per0-0). This could be explained by the conservative character of the necessary robustness. In fact, in all cases where the fuzzy schedule has $\beta_* \geq 0.6$, the makespan values for all deterministic simulations are below the threshold $C^\star$.

7 Conclusions

We have tackled a variant of the open shop scheduling problem where uncertainty in durations is modelled using triangular fuzzy numbers. We have defined necessary and possible β-robustness in terms of scheduling and we have proposed as objective function to maximize the most pessimistic measure which is the necessary β-robustness. Moreover, we have proposed a method to empirically assess the actual robustness of the solutions. We have tested our approach using a genetic algorithm from the literature using an adaptive threshold of makespan values that overcomes the drawback of a likely random search by the GA. Based in the promising results, in the future we intend to improve on the β-robustness by adapting to the fuzzy framework the definition of α-β-robustness, that is, for a given confidence level β (ideally close to 1), try to minimise the threshold α for which this confidence is obtained (as in [22] for stochastic scheduling). We also intend to consider some kind of multiobjective approach that maximises robustness and minimises makespan.

Acknowledgements. This research has been supported by the Spanish Government under research grants FEDER TIN2010-20976-C02-02 and MTM2010-16051 and by the Principality of Asturias (Spain) under grant Severo Ochoa BP13106.

References

1. Pinedo, M.L.: Scheduling. Theory, Algorithms, and Systems, 3rd edn. Springer (2008)
2. Sha, D.Y., Cheng-Yu, H.: A new particle swarm optimization for the open shop scheduling problem. Computers & Operations Research 35, 3243–3261 (2008)
3. Herroelen, W., Leus, R.: Project scheduling under uncertainty: Survey and research potentials. European Journal of Operational Research 165, 289–306 (2005)
4. Dubois, D., Fargier, H., Fortemps, P.: Fuzzy scheduling: Modelling flexible constraints vs. coping with incomplete knowledge. European Journal of Operational Research 147, 231–252 (2003)
5. Van de Vonder, S., Demeulemeester, E., Herroelen, W.: Proactive heuristic procedures for robust project scheduling: An experimental analysis. European Journal of Operational Research 189, 723–733 (2008)
6. Aissi, H., Bazgan, C., Vanderpooten, D.: Min-max and min-max regret versions of combinatorial optimization problems: A survey. European Journal of Operational Research 197, 427–438 (2009)
7. Wang, J.: A fuzzy robust scheduling approach for product development projects. European Journal of Operational Research 152, 180–194 (2004)

8. Kasperski, A., Kule, M.: Choosing robust solutions in discrete optimization problems with fuzzy costs. Fuzzy Sets and Systems 160, 667–682 (2009)
9. Daniels, R.L., Carrillo, J.E.: β-robust scheduling for single-machine systems with uncertain processing times. IIE Transactions 29, 977–985 (1997)
10. Aiche, F., Abbas, M., Dubois, D.: Chance-constrained programming with fuzzy stochastic coefficients. Fuzzy Optimization and Decision Making 12, 125–152 (2013)
11. Fortemps, P.: Jobshop scheduling with imprecise durations: a fuzzy approach. IEEE Transactions of Fuzzy Systems 7, 557–569 (1997)
12. Sakawa, M., Kubota, R.: Fuzzy programming for multiobjective job shop scheduling with fuzzy processing time and fuzzy duedate through genetic algorithms. European Journal of Operational Research 120, 393–407 (2000)
13. González Rodríguez, I., Puente, J., Vela, C.R., Varela, R.: Semantics of schedules for the fuzzy job shop problem. IEEE Transactions on Systems, Man and Cybernetics, Part A 38(3), 655–666 (2008)
14. Puente, J., Vela, C.R., González-Rodríguez, I.: Fast local search for fuzzy job shop scheduling. In: Proc. of ECAI 2010, pp. 739–744. IOS Press (2010)
15. Alcaide, D., Rodriguez-Gonzalez, A., Sicilia, J.: A heuristic approach to minimize expected makespan in open shops subject to stochastic processing times and failures. International Journal of Flexible Manufacturing Systems 17, 201–226 (2006)
16. González-Rodríguez, I., Palacios, J.J., Vela, C.R., Puente, J.: Heuristic local search for fuzzy open shop scheduling. In: Proc. FUZZ-IEEE 2010, pp. 1858–1865. IEEE (2010)
17. Noori-Darvish, S., Mahdavi, I., Mahdavi-Amiri, N.: A bi-objective possibilistic programming model for open shop scheduling problems with sequence-dependent setup times, fuzzy processing times, and fuzzy due-dates. Applied Soft Computing 12, 1399–1416 (2012)
18. Palacios, J.J., González-Rodríguez, I., Vela, C.R., Puente, J.: Swarm lexicographic goal programming for fuzzy open shop scheduling. Journal of Intelligent Manufacturing (2013)
19. Palacios, J.J., Puente, J., Vela, C.R., González-Rodríguez, I.: A genetic algorithm for the open shop problem with uncertain durations. In: Mira, J., Ferrández, J.M., Álvarez, J.R., de la Paz, F., Toledo, F.J. (eds.) IWINAC 2009, Part I. LNCS, vol. 5601, pp. 255–264. Springer, Heidelberg (2009)
20. Dubois, D., Prade, H.: Possibility Theory: An Approach to Computerized Processing of Uncertainty. Plenum Press, New York (1986)
21. Niu, Q., Jiao, B., Gu, X.: Particle swarm optimization combined with genetic operators for job shop scheduling problem with fuzzy processing time. Applied Mathematics and Computation 205, 148–158 (2008)
22. Wu, C.W., Brown, K.N., Beck, J.C.: Scheduling with uncertain durations: Modeling β-robust scheduling with constraints. Computers & Operations Research 36, 2348–2356 (2009)
23. Dubois, D., Prade, H.: Fuzzy Sets and Systems: Theory and Applications. Academic Press, New York (1980)
24. Zadeh, L.A.: Fuzzy sets as a basis for a theory of possibility. Fuzzy Sets and Systems 1, 3–28 (1978)
25. Graham, R., Lawler, E., Lenstra, J., Rinnooy Kan, A.: Optimization and approximation in deterministic sequencing and scheduling: a survey. Annals of Discrete Mathematics 4, 287–326 (1979)