Author's copy of
Palacios, J. J., González-Rodríguez, I., Vela, C. R., & Puente, J. (2017).
Robust multiobjective optimisation for fuzzy job shop problems. Applied Soft
Computing, 56, 604-616.
DOI of published final version: 10.1016/j.asoc.2016.07.004

Robust Multiobjective Optimisation for Fuzzy Job Shop Problems

Juan José Palacios[a], Inés González-Rodríguez[b,*], Camino R. Vela[a], Jorge Puente[b]

[a]*Dept. of Computing, University of Oviedo,*
Campus de Viesques S/N, 33204, Gijón (Spain)
[b]*Dept. of Mathematics, Statistics and Computing, University of Cantabria,*
Av. Los Castros S/N, 39005, Santander (Spain)

Abstract

In this paper we tackle a variant of the job shop scheduling problem with uncertain task durations modelled as fuzzy numbers. Our goal is to simultaneously minimise the schedule's fuzzy makespan and maximise its robustness. To this end, we consider two measures of solution robustness: a predictive one, prior to the schedule execution, and an empirical one, measured at execution. To optimise both the expected makespan and the predictive robustness of the fuzzy schedule we propose a multiobjective evolutionary algorithm combined with a novel dominance-based tabu search method. The resulting hybrid algorithm is then evaluated on existing benchmark instances, showing its good behaviour and the synergy between its components. The experimental results also serve to analyse the goodness of the predictive robustness measure, in terms of its correlation with simulations of the empirical measure.

Keywords: Scheduling, Fuzzy Sets, Metaheuristics, Robustness, Multiobjective Optimisation

[*] Corresponding author

Email addresses: juanjose.palonso@gmail.com (Juan José Palacios), gonzalezri@unican.es (Inés González-Rodríguez), crvela@uniovi.es (Camino R. Vela), puente@uniovi.es (Jorge Puente)

Preprint submitted to Applied Soft Computing *July 18, 2016*

Following Elsevier Sharing Policy, this work is lincensed
under a Creative Commons Attribution-NonCommercial-
NoDerivatives 4.0 International License

1. Introduction

Operations scheduling is one of the most critical issues in manufacturing and production systems, as well as in information processing environments. The importance of scheduling as a research topic is undeniable, both as a source of interesting complex combinatorial optimisation problems and as a field with multiple real applications in industry, finance, welfare, etc. In particular, shop problems in their multiple variants— for instance, incorporating flexibility, operators or energy efficiency— can model many situations which naturally arise in manufacturing environments [30, 34, 52, 60].

Unfortunately, classical scheduling cannot model many practical situations due to the fact that project decisions usually have to be made in advance, when activity durations are still highly uncertain. A great variety of approaches have been considered to deal with these real-life situations, as can be seen for instance in [36]. Fuzzy sets have contributed to enhancing the applicability of scheduling, helping to bridge the gap between classical techniques and real-world user needs. They have been used both for handling flexible constraints and uncertain data [20, 65, 69]. They are also demonstrating to be an interesting tool for improving solution robustness, a much-desired property in real-life applications [43, 56].

When there is uncertainty in some of the input data, robustness becomes an important factor to be taken into account. The better-known approaches to robustness, based on min-max or min-max regret criteria, aim at constructing solutions having the best possible performance in the worst case motivated by practical applications where an anticipation of the worst case is crucial [2, 43, 68]. However, this kind of approach may be deemed as too conservative in some cases where the worst case is not that critical and instead an overall acceptable performance is preferred [42]. Here, we take an approach that might be more adequate in these situations.

Clearly, when the improvement in robustness must not be obtained at the cost of loosing performance quality in the solutions, we face a bi-objective scheduling problem. In general there is a growing interest in multiobjective optimisation for scheduling and, given its complexity, in the use of metaheuristic techniques to solve these problems, as can be seen in [5, 14, 26, 37] among others. In particular, the multiobjective fuzzy job shop problem is receiving an increasing attention, mostly to optimise objective functions related to makespan and due-date satisfaction. Existing proposals include genetic algorithms [27, 64], differential evolution algorithms [38], or hybrid strategies like the genetic simulated-annealing algorithm from [68]. Interestingly, the latter contemplates finding both robust and satisfactory schedules, although the robustness optimisation criterion is based on the worst-case approach.

In the single-objective case, it is common to hybridise evolutionary algorithms with local search to produce memetic algorithms, which benefit from the synergy between their components to provide a better search capacity. It is possible to find various multiobjective memetic algorithms (MOMAs) in the literature, some of them applied to manufacturing problems [12, 13, 39, 51, 62]. However, according to [49], the number of multiobjective local search algorithms proposed so far is still reduced. In fact, the main difficulty in designing multiobjective memetic algorithms is the implementation of the local search, which essentially is a single-objective optimisation technique. Some multiobjective memetic algorithms resort to scalarisation methods to guide the local search towards concrete areas of the Pareto front, for instance [39, 40, 51]. Other MOMAs use instead acceptance criteria for the local search based on Pareto-dominance [44, 47, 62]. In most cases, the local search method uses a hill-climbing strategy. On the other hand, in [41] a multiobjective flexible job shop scheduling problem is solved not with an evolutionary algorithm but a combination of tabu search, based on the TSAB from [53] and only for makespan minimisation, and path relinking.

In the following we tackle the fuzzy job shop problem, where uncertainty in task durations is modelled using fuzzy numbers. After introducing the problem in Section 2, in Section 3 we give a precise definition of two robustness measures based on the average behaviour across all possible cases: an a-priori measure to be evaluated in constant time from the predictive schedule, and an a-posteriori measure, to be evaluated at the moment of executing the schedule or, in the absence of a real execution, by a surrogate obtained with Monte-Carlo simulations. In Section 4 we propose a hybrid method to find non-dominated solutions with respect to the makespan—the total time needed to complete all jobs— and the a-priori robustness measure. This algorithm combines a multi-objective evolutionary algorithm with a new dominance-based tabu search as iterative improvement method. In Section 5, we report and analyse results of an experimental study which contemplates the synergy between the method components, the performance of the proposed method and the relation between both robustness measures. Finally, some conclusions are given in Section 6.

2. The fuzzy job shop scheduling problem

The *job shop scheduling problem*, also denoted *JSP*, consists in scheduling a set of jobs $\{J_1, \ldots, J_n\}$ on a set of physical resources or machines $\{M_1, \ldots, M_m\}$, subject to a set of constraints. There are *precedence constraints*, so each job J_i, $i = 1, \ldots, n$, consists of m tasks $\{\theta_{i1}, \ldots, \theta_{im}\}$ to be sequentially scheduled. Also, there are *capacity constraints*, whereby each task θ_{ij} requires the uninterrupted and exclusive use of one of the machines for its whole processing time p_{ij}. A feasible schedule is an allocation of starting times for each task such that all constraints hold. The objective is to find a schedule which is *optimal* according to some criterion, most commonly that the *makespan* is minimal.

2.1. Uncertain durations

In real-life applications, it is often the case that the exact processing time of tasks is not known in advance. However, based on previous experience, an expert may be able to estimate, for instance, an interval of possible values for the processing time or its most typical value, and he/she may even be able to assess whether some values in the interval appear to be more plausible than others. This naturally leads to modelling such durations using fuzzy intervals or fuzzy numbers, which have been extensively studied in the literature (cf. [22]). A *fuzzy interval* A is a fuzzy set on the reals (with membership function $\mu_A : \mathbb{R} \to [0, 1]$) such that its α*-cuts* $A_\alpha = \{u \in \mathbb{R} : \mu_A(u) \geq \alpha\}$, $\alpha \in (0, 1]$, are intervals. A fuzzy interval is a *fuzzy number* if its α-cuts (denoted $[\underline{a}_\alpha, \overline{a}_\alpha]$) are closed, its *support* $A_0 = \{u \in \mathbb{R} : \mu_A(u) > 0\}$ is compact (closed and bounded) and there is a unique modal value u^*, $\mu_A(u^*) = 1$. Clearly, real numbers can be seen as a particular case of fuzzy ones.

The simplest model of fuzzy interval is a *triangular fuzzy number* or *TFN*, using an interval $[a^1, a^3]$ of possible values and a modal value a^2 in it. For a TFN A, denoted $A = (a^1, a^2, a^3)$, the membership function takes the following triangular shape:

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

If TFNs are to be used to extend the job shop to handle uncertainty, two issues must be addressed: first, how the arithmetic operations of addition and maximum are to be extended to work with TFNs and, second, the precise meaning of “minimal makespan” when such makespan is a TFN.

2.1.1. Arithmetic of TFNs

In the fuzzy job shop, we essentially need two arithmetic operations on fuzzy numbers, the sum and the maximum. These are obtained by extending the corresponding operations on real numbers using the *Extension Principle*. However, computing the resulting expression is cumbersome, if not intractable; also, the set of TFNs is not always closed under the resulting operation. For the sake of simplicity and tractability of numerical calculations, it is fairly common in the literature, following [25], to approximate the results of these operations by interpolation, evaluating only the operation on the three defining points of each TFN. It turns out that the sum and its approximation coincide, so for any pair of TFNs A and B:

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

Regarding the maximum, we have:

$$\max(A, B) \approx \max_I(A, B) = (\max(a^1, b^1), \max(a^2, b^2), \max(a^3, b^3)). \tag{3}$$

The approximation max_I has been widely used in the scheduling literature, from earlier works [25, 46] to more recent ones [55, 68], to mention but a few. Additionally, some arguments can be given to support this approximation.

First, for any two fuzzy numbers A and B, if f is a bivariate continuous isotonic function, that is, $f : \mathbb{R}^2 \to \mathbb{R}$ such that for any $u \geq u'$ and $v \geq v'$ it holds that $f(u, v) \geq f(u', v')$, then $F = f(A, B)$ is another fuzzy number such that $F_\alpha = [f(\underline{a}_\alpha, \underline{b}_\alpha), f(\overline{a}_\alpha, \overline{b}_\alpha)]$. Computing $f(A, B)$ is then equivalent to computing f on every α-cut. In particular, the addition and the maximum are continuous isotonic functions, so they can be calculated by evaluating two sums or maxima of real numbers for every value $\alpha \in [0, 1]$. If seems then natural to approximate the maximum by the TFN that results from using linear interpolation, evaluating F_α only for certain values of α (as proposed for 6-point fuzzy numbers in [25]). Given that the defining values (a^1, a^2, a^3) of a TFN A are such that $A_0 = [a^1, a^3]$ and $A_1 = [a^2, a^2]$, $\max_I$ in (3) corresponds to such an interpolation for $\alpha = 0$ and $\alpha = 1$.

Secondly, if $F = \max(A, B)$ denotes the maximum of two TFNs A and B and $G = \max_I(A, B)$ its approximated value, then $F = G$ if A and B do not overlap and, in any case, 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 may artificially increase the value of the actual maximum F, while maintaining the support and modal value, that is, $F_0 = G_0$ and $F_1 = G_1$.

Unless otherwise stated and for the sake of a simpler notation, we shall simply write max when referring to the interpolated maximum $\max_I$.

2.1.2. Ordering TFNs

For a given fuzzy schedule, the makespan C_{max} (the completion time of the last task to be executed) is a TFN; if several schedules are available, the "best" one would be the one with minimal makespan. Deciding which is the optimal schedule thus requires comparing fuzzy numbers, but unfortunately no natural total order exists for them. Numerous ranking methods have been and keep being proposed in the literature, and several have been used in the field of fuzzy scheduling (cf. [9] and [1, 20]). Here, we shall adopt an ordering based on expected values, analogously to what is done in stochastic scheduling approach.

Let A be a TFN, its expected value is given by:

$$E[A] = \frac{1}{4}(a^1 + 2a^2 + a^3). \tag{4}$$

This expression is obtained following different approaches: as the credibilistic expectation of the fuzzy variable underlying A [50], as the expected value of a fuzzy number based on random sets [35], as the generative expected value induced by the evidence A [11], as the centre of the mean value of A [23] and as the expected value of the so-called pignistic probability distribution that is the centroid of the set of probabilities $\mathcal{P}(\Pi_A)$ dominated by the possibility measure associated with A [18].

The expected value can be used to define an index-based ranking method; in particular, it induces a total ordering $\leq_E$ in the set of fuzzy intervals [25], where for any two fuzzy intervals A, B $A \leq_E B$ if and only if $E[A] \leq E[B]$. Notice that $\leq_E$ coincides with several other ranking methods from the literature which are not based on expected values, as highlighted in [57]. Additionally, a recent numerical study [9] suggests that, for TFNs, the ranking based on the expected value is very similar to seven more ranking methods, in the sense the ordering they induce in a sample of TFNs is strongly correlated (see [9] for further detail).

Interestingly, it is also possible to establish a relationship between the ranking method based on expected values and classical interval comparison in the light of imprecise probabilities [17, 19]. In particular, it comes down to using Hurwicz criterion for classical interval comparison on upper and lower expectations derived from A. This provides us with an interpretation for comparisons based on $\leq_E$ as those corresponding to a decision maker who keeps an equilibrium between pessimism and optimism.

In summary, with the expected value approach, not only are we taking an analogous stance to stochastic scheduling, but we are also modelling the behaviour of a moderate decision maker as well as taking into account many other ranking methods from the literature, either because they fully coincide with $\leq_E$ for TFNs or because they yield very similar orderings to it.

The problem that results from modelling uncertain processing times as TFNs will be referred to as *fuzzy job shop problem* or FJSP in short. It corresponds to what is called fuzzy JSP with fuzzy processing time in [1], where different extensions of JSP using fuzzy sets, all of them receiving the generic name "fuzzy job shop" in the literature, are classified and reviewed.

3. Robust scheduling

A fuzzy schedule does not provide exact starting times for each task. Instead, it gives a fuzzy interval of possible values for each starting time, provided that tasks are executed in the order determined by the schedule. In fact, it is impossible to predict what the exact time-schedule will be, because it depends on the realisation of the task's durations, which is not known yet. This idea is the basis for a semantics for fuzzy schedules from [31] by which solutions to the fuzzy job shop should be understood as a-priori solutions, also called baseline or predictive schedules in the literature [36]. When tasks are executed according to the ordering provided by the fuzzy schedule we shall know their real duration and, hence, obtain a real (executed) schedule, the a-posteriori solution with deterministic times. Clearly, it is desirable that a fuzzy solution yields reasonably good executed schedules at the moment of its practical use, in clear relation with the concept of schedule robustness.

Roughly speaking, according to [3] a schedule is said to be *robust* if it minimises the effect of executional uncertainties on its primary performance measure, the makespan in our case. This straightforward definition may, however, be subject to many different interpretations when it comes to specifying robustness measures [63]. In this work, we will consider only uncertainties in task processing times and distinguish between a-priori and a-posteriori robustness measures. We consider the *a-priori robustness* to be the one that can be measured in advance from the predictive schedule. We may say this is a prediction on the "real" robustness of the schedule. The *a-posteriori robustness* would then be the real robustness of the schedule, measured after executing it in a real environment. In the following, we give a precise definition of a measure for each kind of robustness.

It is worth noticing that the approach to robustness taken here is different from the better-known approach from combinatorial optimisation, based on min-max or min-max regret criteria, which aims at constructing solutions having the best possible performance in the worst case [2]. The study of such criteria is motivated by practical applications where an anticipation of the worst case is crucial and there already exist proposals to translate it to the fuzzy framework [43, 68]. However, the min-max approach may be deemed as too conservative in some cases where the worst case is not that critical and an overall acceptable performance is preferred [42]. It is in these situations where an approach such as the one proposed here might be more adequate. Also, more classical approaches measure robustness as a deviation from the optimal solution on each possible case, thus assuming that such optimal solution can indeed be found. Unfortunately, this is a somewhat unrealistic assumption when dealing with complex problems such as job shop (even in its deterministic version). Our approach takes the alternative stance of measuring deviations between the prediction and the real execution (be it optimal or not).

3.1. A-priori measure

Given that a predictive schedule provides a most plausible value for the makespan (C^2_{max}), we propose to interpret the a-priori robustness as the maximum deviation that the makespan of the executed schedule may suffer with respect to this value. Thus, we define the a-priori robustness measure, denoted Rob^D, as the maximum possible difference between the modal value C^2_{max} and the bounds of the support of the fuzzy interval C_{max}, as follows:

$$Rob^D = \max\{C^2_{max} - C^1_{max}, C^3_{max} - C^2_{max}\}. \tag{5}$$

Obviously, the smaller this difference, the better the robustness.

It is easy to see that Rob^D thus defined measures the maximum possible difference between the makespan of a real execution and the most likely estimated makespan value. Indeed, as mentioned above, the First Decomposition Theorem ensures that the predicted makespan C_{max} (a TFN obtained

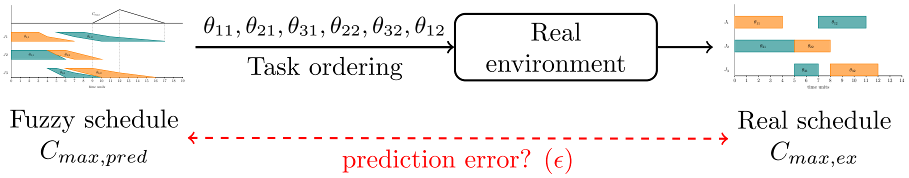
$\theta_{11}, \theta_{21}, \theta_{31}, \theta_{22}, \theta_{32}, \theta_{12}$
Task ordering
Real environment
Fuzzy schedule
$C_{max,pred}$
Real schedule
$C_{max,ex}$
prediction error? (ϵ)

Figure 1: Graphical representation of the a-posteriori robustness measure (ϵ-*robustness*).

using the sum and the approximated maximum) is such that its support contains all the exact makespan values resulting from all possible realisations of the schedule. Interestingly, for the particular case of triangular fuzzy numbers, Rob^D coincides with the entropy of C_{max} as defined in the setting of credibility theory [48].

3.2. A-posteriori measure

Regarding the a-posteriori robustness, we shall adopt the concept of ϵ-robustness proposed in [6] for stochastic scheduling, already adapted to the fuzzy open shop and fuzzy flexible job shop in [56] and [57] respectively. This definition states that a predictive schedule is considered to be robust if the quality of the eventually executed schedule is close to the quality of the predictive schedule. In particular, for the fuzzy job shop, a predictive schedule with makespan value $C_{max,pred}$ (a TFN) is ϵ-*robust* for a given ϵ if the objective value $C_{max,ex}$ of the eventually executed schedule (a real value) is such that:

$$(1-\epsilon) \leq \frac{C_{max,ex}}{E[C_{max,pred}]} \leq (1+\epsilon) \tag{6}$$

or, equivalently,

$$\frac{|C_{max,ex} - E[C_{max,pred}]|}{E[C_{max,pred}]} \leq \epsilon. \tag{7}$$

That is, the relative error of the estimation made by the predictive schedule (i.e. its expected makespan) is bounded by ϵ. Obviously, the smaller ϵ is, the better.

Figure 1 illustrates this definition of ϵ-*robustness*, where the idea is to measure the deviation between the performance predicted by the fuzzy schedule and the actual performance obtained during execution (assuming that operations are executed in the order established by the fuzzy schedule).

Notice however that this definition requires a real execution of the problem which may not always be available. In fact, in the literature it is very common to use synthetic problems for which no real execution exists. In this case we propose to run a Monte-Carlo simulation to provide a surrogate of the ϵ-robustness measure. Specifically, given a fuzzy instance, we may generate a sample of K possible realisations (also called scenarios) of that instance by assigning an exact duration to each task, that is K deterministic instances on which we can evaluate the ϵ-robustness of the solution. Now for each realisation $k = 1, \ldots, K$, let $C_{max,k}$ denote the exact makespan obtained by executing tasks according to the ordering provided by a predictive schedule. Then, the average ϵ-robustness of the predictive schedule, denoted $\overline{\epsilon}$, is calculated as:

$$\overline{\epsilon} = \frac{1}{K} \sum_{k=1}^{K} \frac{|C_{max,k} - E[C_{max}]|}{E[C_{max}]}, \tag{8}$$

where $E[C_{max}]$ is the expected makespan estimated by the predictive schedule. The process followed to compute the $\overline{\epsilon}$ value is illustrated in Figure 2.

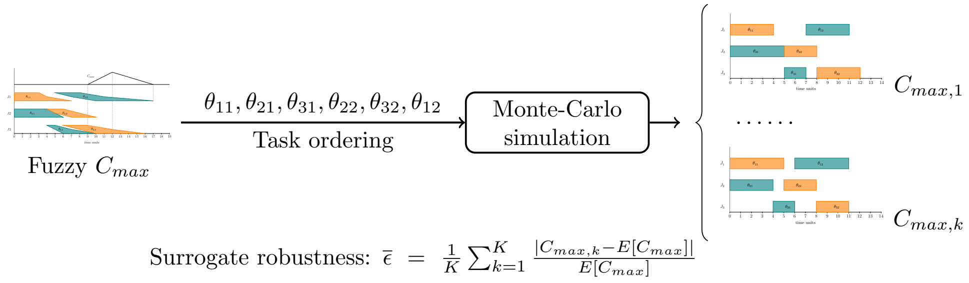
Fuzzy C_{max}
$\theta_{11}, \theta_{21}, \theta_{31}, \theta_{22}, \theta_{32}, \theta_{12}$
Task ordering
Monte-Carlo simulation
$C_{max,1}$
......
$C_{max,k}$
Surrogate robustness: $\overline{\epsilon} = \frac{1}{K}\sum_{k=1}^{K} \frac{|C_{max,k} - E[C_{max}]|}{E[C_{max}]}$

Figure 2: Graphical representation of the surrogate ϵ-robustness measure.

Clearly, a crucial factor in this method is the way in which we determine crisp durations for the tasks. This is done by simulating crisp durations for tasks following a probability distribution that is consistent with the possibility distribution μ_A defined by each fuzzy duration A. In [56], the authors use a simple renormalisation technique for possibility-probability transformation, consistent in dividing the membership function μ_A by the area under the triangle. However, this technique can be objected to: according to [18], it is arbitrary, since the obtained probability may fail to belong to $\mathcal{P}(\mu_A)$, the set of probability measures dominated by μ_A. In consequence, here we shall consider two alternative approaches to obtain a probability distribution for the simulation.

The first approach consists in considering the uniform probability distribution that is bounded by the support of the TFN; we shall refer to resulting simulation as “Scenario I”. This transformation is motivated by several results from the literature (see [4, 21]) that justify the use of TFNs as fuzzy counterparts to uniform probability distributions and model-free approximations of probability distributions with bounded support.

The second method consists in taking the probability distribution obtained from each fuzzy duration after applying the pignistic transformation obtained by considering cuts as uniformly distributed probabilities [24]. This is the probability one would obtain from the membership function of a fuzzy duration applying a generalised version of the Insufficient Reason Principle by Laplace. The probability distribution thus obtained is much more “focused” on the modal value, in the sense that it gives high probability to values close to the mode and very low probability to values at the extreme of the support interval. We shall refer to the simulation that results from this transformation as “Scenario II”

4. The multiobjective approach

It is easy to find in the literature many works on scheduling that assume a deterministic setting and aim at optimising classical objective functions such as makespan or tardiness. However, when there is uncertainty in some of the input data, solution robustness becomes an important factor to be taken into account. Indeed, an optimal solution found for an ideal deterministic scenario (for instance, assuming that all durations take their modal value) may be of little or no use when it is executed if changes in the input data affect its real performance. Therefore, our aim in this work is to optimise both a performance or quality function as well as the robustness of the solution with respect to that function.

Clearly, robustness measures such as those defined in Section 3 are dependent on the performance function. Therefore we opt for an optimisation strategy that allows us to optimise both the quality function and the associated robustness measure simultaneously. Specifically, there will be an objective function related to performance, the expected makespan $E[C_{max}]$, and an objective function related to robustness, the a-priori measure Rob^D. Ideally, the surrogate of the a-posteriori measure $\overline{\epsilon}$ should be used, but the

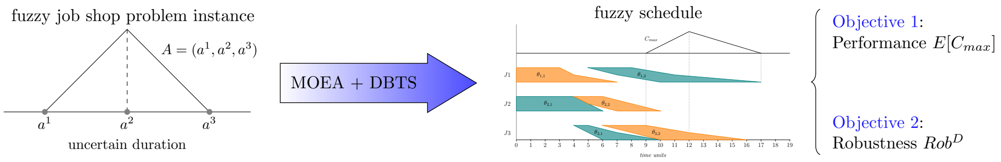

Figure 3: Flow of information of the proposed algorithm.

simulations required to compute its value translate into an excessive computational cost, in contrast with the constant time required by Rob^D.

To optimise the two objective functions, we shall take a dominance-based approach. In general, for a minimisation problem with f_i, $i = 1, \ldots, n$ objective functions, a solution s is said to be *dominated* by a solution s', denoted $s' \succ s$ iff for each objective function f_i, $f_i(s') \leq f_i(s)$ and there exists at least one objective function such that $f_i(s') < f_i(s)$. Our goal will then be to find non-dominated solutions to the FJSP with respect to $E[C_{max}]$ and Rob^D. To achieve this, we propose a dominance-based hybrid method, combining a multiobjective evolutionary algorithm (MOEA) with a dominance-based tabu search (DBTS). Figure 3 illustrates the flow of information of the resulting method, which receives as input a problem instance with fuzzy numbers as task durations and returns a fuzzy schedule, for which the two objective functions $E[C_{max}]$ and Rob^D are evaluated.

4.1. Multiobjective evolutionary algorithm

Our MOEA is based on the well-known NSGA-II template for a dominance-based evolutionary algorithm [15]. Roughly speaking, an initial population P_0 is randomly created and evaluated and then the algorithm iterates over a number of generations, keeping a set of non-dominated solutions. At each iteration i a new population $\text{Off}(P_i)$ is built from the current one P_i by applying the genetic operators of selection and recombination and a replacement strategy is applied to obtain the next generation P_{i+1}. Finally, the algorithm stops if no solution belonging to the set of non-dominated solutions is removed from this set after max_{MOEA} iterations.

4.1.1. Representation

Solutions are codified into chromosomes using permutations with repeated elements, as introduced in [7] for the JSP. This is a permutation of the set of tasks, each being represented by its job number. For example, if we have a problem with 3 jobs: $J_1 = \{\theta_{11}, \theta_{12}\}$, $J_2 = \{\theta_{21}, \theta_{22}, \theta_{23}, \theta_{24}\}$, $J_3 = \{\theta_{31}, \theta_{32}, \theta_{33}\}$, then a topological order of tasks $\pi = \{\theta_{21}, \theta_{11}, \theta_{22}, \theta_{31}, \theta_{23}, \theta_{32}, \theta_{33}, \theta_{24}, \theta_{12}\}$ is represented by the sequence $v = (2\ 1\ 2\ 3\ 2\ 3\ 3\ 2\ 1)$. With this encoding, every feasible processing order has a unique associated permutation.

A given chromosome is evaluated by generating an associated schedule and then computing the pair of fitness values formed by the makespan expected value and the a-priori robustness. To do this, each task is scheduled using an insertion strategy following the sequence given by the chromosome [58]. More precisely, given a task θ_{ij} that is to be scheduled, we define a *feasible insertion interval* as a time interval $[t_S, t_E]$ in which the machine required by θ_{ij} is idle and such that θ_{ij} can be processed within that time interval without violating precedence constraints. In our case, this means that for all three components $k = 1, 2, 3$ of the TFN it holds that $t_S^k + p_{ij}^k \leq t_E^k$, and $t_S^k \geq C_{i(j-1)}^k$, where $C_{i(j-1)}$ denotes the completion time of the predecessor in the job $\theta_{i(j-1)}$ (if $j = 0$, $C_{i(j-1)}$ is taken to be the TFN $(0, 0, 0)$). Then, task θ_{ij} is scheduled with starting time S_{ij}, where S_{ij} is the smallest t_S that can be found, and hence its completion time is $C_{ij} = S_{ij} + p_{ij}$.

4.1.2. Genetic operators

In the selection phase all chromosomes are randomly grouped into pairs, and then each of these pairs is mated to obtain two offspring. For the mating we have considered three of the most common crossover operators for the *JSP*: the Job Order Crossover (JOX), the Generalized Order Crossover (GOX) and the Generalized Position Crossover (GPX) [8]. In order to preserve the diversity of individuals inside the population and prevent the algorithm from getting stuck in local optima, three different mutation strategies are also introduced: swap, reverse and insertion mutation [66].

4.1.3. Replacement strategy

The replacement strategy is a key factor in MOEA algorithms. It establishes how population P_i of size N and population $\text{Off}(P_i)$ that results from applying selection and mating to P_i are combined to generate the new population P_{i+1} for the next iteration of the algorithm. Here we adopt a strategy based on the non-dominated sorting approach with diversity preservation from [15]. Initially, for each individual j in the pool $P_i \cup \text{Off}(P_i)$ a non-domination rank ($rank(j)$) and a crowding distance ($dist(j)$) are calculated. The former sorts the pool into different non-domination levels while the latter estimates the density of solutions in the area of the non-domination level where the individual lies. Population P_{i+1} is then formed by the best N individuals from the pool $P_i \cup \text{Off}(P_i)$ according to the lexicographical order defined by $(rank, dist)$. That is, solutions belonging to a lower (better) non-domination rank are preferred and, between two solutions in the same non-dominance level, we prefer the solution located in the less crowded region.

In order to provide greater diversity to the algorithm, we have included an additional step in the above strategy. Specifically, we propose to start by removing from the pool of individuals $P_i \cup \text{Off}(P_i)$ those which are repeated, in the sense that there exists in the pool at least another individual having identical values for all objective functions. Only after this elimination is the above strategy based on $(rank, dist)$ applied. In the case that such elimination causes the pool to contain less than N individuals, all the non-repeated individuals pass onto the next generation P_{i+1}, which is later completed with the best repeated individuals according to their rank level and crowding distance.

4.2. Dominance-based tabu search

Tabu search (TS) is an advanced local search technique, proposed in [28] and [29], which may select non-improving neighbours in order to escape from local optima and promotes the exploration of new promising regions of the search space by maintaining a so-called tabu list of moves. Due to the sensitivity of TS to its starting solution, it is common practice to use it in a multi-start fashion, with several runs from different starting solutions which are either generated randomly or using a problem-specific heuristic that yields good starting points. Alternatively, TS is often used in combination with other metaheuristics in such a way that the tabu search provides exploitation while the other metaheuristic provides exploration. Indeed, tabu search combined with additional techniques is the basis for two of the state-of-the-art approaches to deterministic JSP [54, 70].

Given the above, we propose to combine the MOEA from Section 4.1 with a dominance-based tabu search procedure (DBTS), in such a way that DBTS is applied to every individual in the population right after its evaluation, as illustrated in Figure 4. We shall refer to the resulting multiobjective memetic algorithm as *MTEA* (the acronym for multiobjective tabu evolutionary algorithm).

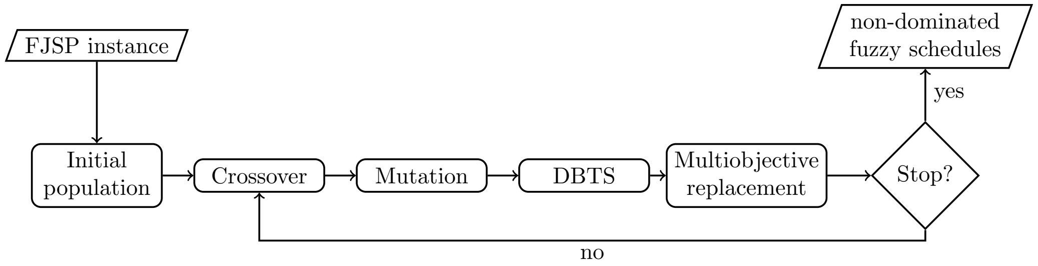

Figure 4: Flow chart of the hybrid algorithm MTEA.

4.2.1. Neighbourhood structure

Clearly, a central element of any tabu search procedure is the definition of the neighbourhood structure. Several neighbourhoods have been proposed in the literature for the JSP; here we adopt the structure for makespan optimisation initially proposed in [67] and extended to the fuzzy case in [33]. This structure is based on reversing critical arcs in a graph representation of a schedule S and exhibits some nice properties, in particular, it always generates feasible neighbours, avoiding the need of repair procedures.

4.2.2. General schema

The DBTS algorithm used herein follows a similar schema to many TS algorithms in the literature. In the first step, the initial solution, here provided by the MOEA, is evaluated. Then the TS iterates for a number of steps, so at each iteration, the neighbourhood of the current solution is calculated and the best neighbour is then selected as new solution (in our case, such best neighbour will be selected based on dominance-based criteria, as explained below). Then the arc that was reversed to generate the best neighbour is stored as tabu, so any neighbour is considered to be tabu (unless an aspiration criterion holds) if it has been generated by reversing a tabu arc. Additionally, we use a dynamic length schema for the tabu list and a cycle checking mechanism as proposed in [16]. The TS procedure finishes after a given number of iterations max_{DBTS} without improvement, returning the best solution found so far.

A drawback of the procedure as described above is that it requires evaluating the entire neighbourhood of the current solution in order to find the best neighbour, with the consequent computational cost. For the sake of efficiency, we introduce a mechanism to estimate for each neighbour the value of both objective functions, so the choice of best neighbour is no longer based on the actual objective values, but on their estimates. Specifically, we use here the method proposed in [61] to estimate the neighbour's fuzzy makespan as the length of the longest path passing through the nodes affected by the move, since it has proved to be both fast and accurate, always providing a lower bound for the expected makespan.

4.2.3. Dominance-based selection

A very important issue when applying tabu search to a multiobjective setting is to establish a selection criterion for the best neighbour. In general, given the solution s and its neighbourhood, there is not a single "best" neighbour, since the dominance relation $\succ$ defines a partial order. In the literature we can find different approaches to this issue. For instance, in [39] and [40] the authors propose to scalarise the objective function vector to guide the search. However, according to [49], the scalarising approach has the inconvenience that "in the combinatorial case, a number of efficient solutions known as non-supported efficient solutions, are not optimal for any weighted-sum aggregation function". Other authors propose instead to define acceptance criteria based on a dominance relation; for instance, in [10] or [45] the local search provides a set of candidate solutions by keeping an archive of non-dominated ones.

Here we propose a dominance-based approach which bears certain similarities to the PAES (Pareto Archived Evolution Strategy) from [45]. As PAES, it starts from a single initial solution and performs the selection based on dominance. However, our approach is different in many other aspects. For instance, we do not keep and return an archive of limited size of non-dominated solutions; instead, our local search procedure provides a single (hopefully improved) output solution, what is called "one-point iteration" in [47]. In addition, thanks to the estimation procedure we are using we can explore the whole neighbourhood looking for the most promising neighbour. Finally, the selection of the best neighbour is based on dominance and with additional criteria somehow related to crowding, but always relative to the neighbourhood. The combination of the "one-point iteration" strategy with the estimation procedure allows to apply the local search to every individual in the population at each iteration; this would not be possible for very interesting alternatives such as the Pareto Local Search [59], given their greater computational load.

For a given solution s and its neighbourhood $\mathcal{N}(s)$, we define three sets, forming a *dominance-based neighbourhood partition*:

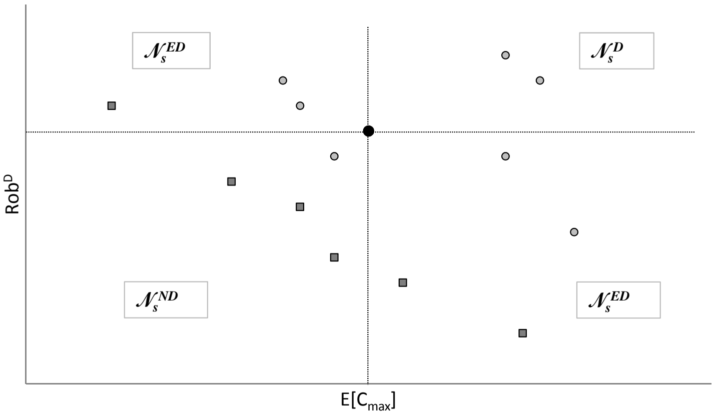

Figure 5: Example of dominance-based neighbourhood partition.

- $\mathcal{N}^{ND}(s) = \{n_i \in \mathcal{N}(s) : n_i \succ s\}$ the subset of neighbours dominating s;
- $\mathcal{N}^{D}(s) = \{n_i \in \mathcal{N}(s) : s \succ n_i\}$, the subset of neighbours dominated by s;
- $\mathcal{N}^{ED}(s) = \mathcal{N}(s) - \mathcal{N}(s)^{ND} - \mathcal{N}(s)^{D}$, the subset of neighbours that neither dominate nor are dominated by s.

Obviously, these three sets are disjoint and form a partition of the neighbourhood $\mathcal{N}(s)$.

Now let $N_s = \{n_i \in \mathcal{N}(s) : \nexists n_j \in \mathcal{N}(s), n_j \succ n_i\}$ be the subset of non-dominated neighbours and let $N_s^{ND} = \mathcal{N}^{ND}(s) \cap N_s$, $N_s^{D} = \mathcal{N}^{D}(s) \cap N_s$ and $N_s^{ED} = \mathcal{N}^{ED}(s) \cap N_s$ denote the intersection of the non-dominated neighbours with the three sets above. Clearly, it cannot be the case that $N_s^{ND} \neq \emptyset$ and $N_s^{D} \neq \emptyset$ simultaneously.

Figure 5 illustrates these definitions depicting the projection of a solution and its neighbourhood in the objective space. The black dot represents the current solution s, the dark squares represent the estimates of the non-dominated neighbours in N_s and the grey dots represent the estimates of the dominated neighbours in $\mathcal{N}(s) - N_s$, while the dotted lines partition $\mathcal{N}(s)$ in three regions corresponding to the three sets above, $\mathcal{N}^{ND}(s)$, $\mathcal{N}^{D}(s)$ and $\mathcal{N}^{ED}(s)$. In this case, $N_s^{D} = \emptyset$.

We now define the *set of candidate neighbours* as the following subset of the non-dominated ones:

$$N'_s = \begin{cases} N_s^{ND}, & \text{if } N_s^{ND} \neq \emptyset, \\ N_s^{ED}, & \text{if } N_s^{ND} = \emptyset \wedge N_s^{ED} \neq \emptyset, \\ N_s^{D}, & \text{otherwise} \end{cases} \tag{9}$$

The rationale behind this definition is to isolate from N_s those neighbours that improve the current solution, provided they exist, or at least, those which are not worse than the current solution in terms of the dominance relation.

Finally, to continue the iterative process the DBTS selects a single element from $N'_s \subseteq N_s$ based on the following criterion: normalise the objective values of all the neighbours in N'_s and then select as the "best" element the one that minimises the Euclidean distance to the origin $(0, 0)$.

Regarding the stopping criterion and the dynamic length schema for the tabu list, we consider that there is an improvement only in the case where DBTS moves to a solution $n_i \in N_s^{ND}$.

5. Experimental results

In this section we provide an empirical evaluation of the proposed algorithm and an analysis on the obtained results in terms of makespan and robustness. Experiments are made on instances available in the literature for the FJSP [32]. Specifically we use a set of 12 instances generated by fuzzyfying 12 well-known benchmark problems for deterministic job shop which are considered hard to solve: FT10 (size 10×10), FT20 (20×5), La21, La24, La25 (15×10), La27, La29 (20×10), La38, La40 (15×15), and ABZ7, ABZ8, ABZ9 (20×15). We take a fuzzy instance generated from each original problem following [25], so task durations become symmetric TFNs where the modal value is the original duration. This ensures that the optimal solution to the original problem in terms of makespan provides a lower bound (LB) for the expected makespan of the fuzzified version. All the experiments reported in this section correspond to a C++ implementation running on a PC with Xeon processor at 2,2Ghz and 24 Gb RAM running Linux (SL 6.0.1).

5.1. Experimental framework

As a result of a preliminary parametric analysis, the parameter setup for the MTEA is as follows: population size 100, GOX crossover with probability

1.0, inversion mutation with probability 0.1, $max_{\text{MOEA}} = 25$ as stopping criterion for the evolutionary algorithm and $max_{\text{DBTS}} = 100$ as stopping criterion for the local search. Given the stochastic nature of the algorithm, it is run 10 times on each instance, so 10 different sets of non-dominated solutions (one per run) are stored in order to obtain representative data.

In the literature we find many proposals to compare different configurations or setups of a multiobjective algorithm. In this work we adopt two well-known indicators: the hypervolume (HV) and the unary additive ϵ indicator ($I_{\epsilon+}$) [66, 71]. Additionally, to avoid problems derived from the different scales of the objective functions, we normalise their values. For a set of solutions S and an objective function f_i, let $f_i^-(S)$ and $f_i^+(S)$ denote respectively a lower and an upper bound of f_i in S, then the objective value $f_i(s)$ of each solution $s \in S$ is normalised as follows:

$$f_i(s) = \frac{f_i(s) - f_i^-(S)}{f_i^+(S) - f_i^-(S)}. \tag{10}$$

In our case, the lower bound for the expected makespan $E[C_{max}]$ will be given by the best-known solution of the deterministic counterpart of each instance, whereas the lower bound for the new proposed robustness measure Rob^D will be taken to be 0. As for the upper bound, it is calculated with the following expression:

$$f_i^+(S) = \max\{f_i(s) : s \in S\} + 0.05 * (\max\{f_i(s) : s \in S\} - \min\{f_i(s) : s \in S\})$$

By taking this upper bound, we prevent the solutions from having a value equal to 1, which can be troublesome when computing comparison metrics such as the hypervolume. The lower and upper bounds used for the union of all sets of solutions reached for the three methods are provided in Table 1.

These lower and upper bounds are used also as reference points for measuring the hypervolume indicator. To compute the ϵ-indicator, $I_{\epsilon+}$, it is necessary to have a reference set of non-dominated solutions RF. Ideally, this set RF should be the optimal Pareto front PO^*, but in our case this is not known because the proposed benchmark has not been solved yet to optimality even for the single objective C_{max} and the robustness metric is proposed in this work for the first time. In consequence, we follow an standard approach and approximate RF by the non-dominated elements in the union of all sets of solutions [66].

Instance	$f^{-}(S)$		$f^{+}(S)$	
	$E[C_{max}]$	Rob^{D}	$E[C_{max}]$	Rob^{D}
ABZ7	656	0	895	54
ABZ8	645	0	865	49
ABZ9	661	0	955	47
FT10	930	0	1205	73
FT20	1165	0	1570	100
La21	1046	0	1340	108
La24	935	0	1225	95
La25	977	0	1230	84
La27	1235	0	1605	105
La29	1152	0	1550	104
La38	1196	0	1590	99
La40	1222	0	1615	108

Table 1: Lower and upper bounds for each objective in the proposed instances

5.2. Analysis of MTEA performance

In order to asses the performance of our hybrid algorithm MTEA, we shall compare it with each of its components: the MOEA algorithm with no DBTS and the DBTS alone, both starting from solutions generated at random. For the sake of fairness in comparisons, DBTS has been iteratively applied for as many iterations as the number of individuals evaluated by MTEA and MOEA evolves a population (of size 100, the same as MTEA) for the same running time as DBTS.

The results obtained by MOEA and DBTS run separately as well as by MTEA can be seen in Tables 2 and 3. Table 2 shows, for each instance, the average hypervolume (HV) and ϵ-indicator ($I_{\epsilon+}$) across the 10 sets of non-dominated solutions obtained by each algorithm, with standard deviation values between brackets. For each problem instance, after testing for normality with a Kolmogorov-Smirnov test, we have run T-tests for pair-wise comparisons between algorithms on these results; we highlight in bold those values in the table which are significantly better than their counterparts according to these tests. Table 3 reports for each instance the average number of solutions contained in each non-dominated set and the average runtime in seconds for each method. We can see that both MOEA and MTEA clearly improve DBTS both in terms of $I_{\epsilon+}$ and HV. More importantly, the hybrid

Inst.	DBTS		MOEA		MTEA	
	HV	$I_{\epsilon+}$	HV	$I_{\epsilon+}$	HV	$I_{\epsilon+}$
ABZ7	0.347 (0.014)	0.260 (0.011)	0.561 (0.033)	0.115 (0.031)	**0.612** (0.023)	**0.072** (0.022)
ABZ8	0.274 (0.014)	0.307 (0.023)	0.447 (0.036)	0.173 (0.032)	**0.539** (0.018)	**0.068** (0.018)
ABZ9	0.300 (0.012)	0.289 (0.010)	0.483 (0.038)	0.174 (0.027)	**0.565** (0.028)	**0.098** (0.025)
FT10	0.412 (0.015)	0.167 (0.016)	0.508 (0.036)	0.111 (0.040)	**0.555** (0.018)	**0.066** (0.018)
FT20	0.264 (0.023)	0.328 (0.017)	**0.592** (0.022)	**0.057** (0.014)	**0.594** (0.018)	**0.055** (0.019)
La21	0.439 (0.010)	0.218 (0.010)	0.559 (0.034)	0.159 (0.019)	**0.621** (0.048)	**0.121** (0.043)
La24	0.435 (0.012)	0.170 (0.013)	0.542 (0.041)	0.105 (0.044)	**0.578** (0.022)	**0.060** (0.020)
La25	0.392 (0.021)	0.216 (0.019)	0.521 (0.023)	0.122 (0.028)	**0.568** (0.018)	**0.079** (0.022)
La27	0.315 (0.012)	0.291 (0.023)	**0.557** (0.040)	**0.099** (0.024)	**0.574** (0.023)	**0.095** (0.022)
La29	0.291 (0.013)	0.298 (0.014)	0.483 (0.043)	0.158 (0.052)	**0.581** (0.030)	**0.078** (0.031)
La38	0.317 (0.018)	0.305 (0.028)	0.461 (0.046)	0.208 (0.059)	**0.626** (0.021)	**0.041** (0.022)
La40	0.427 (0.015)	0.208 (0.010)	0.560 (0.033)	**0.102** (0.017)	**0.616** (0.029)	**0.083** (0.030)
Avg.:	*0.351*	*0.255*	*0.523*	*0.132*	*0.586*	*0.076*

Table 2: Average indicator value and the standard deviation (in brackets) obtained by MOEA, DBTS and MTEA

algorithm MTEA outperforms MOEA in terms of HV as well as $I_{\epsilon+}$, except for 2 or 3 cases (depending on whether HV or $I_{\epsilon+}$ is considered) for which the improvement is not statistically significant. Furthermore, the hybrid MTEA is able to obtain these results in less than half the runtime of DBTS and MOEA. To complement Tables 2 and 3, Figure 6 depicts pairwise comparisons between algorithms by means of empirical attainment functions (EAFs) [71], plotting the difference in EAF values between each pair of algorithms. The figure corresponds to instance La40, but the behaviour in the remaining instances is similar. Comparing MOEA with DBTS, we can see that the MOEA clearly dominates DBTS in all regions. The EAFs plots also indicate that MTEA dominates MOEA, but with higher probability in the region with better makespan than in the region with better robustness, where there is some probability that solutions obtained by MOEA dominate those obtained with MTEA. It is also clear that the sets of solutions obtained with DBTS are dominated by the solutions of MTEA with a probability very close to 1 in every region. This shows the potential of the combination of both strategies in MTEA, due to a synergy effect.

Additionally, to measure the real impact of the local search step in MTEA,

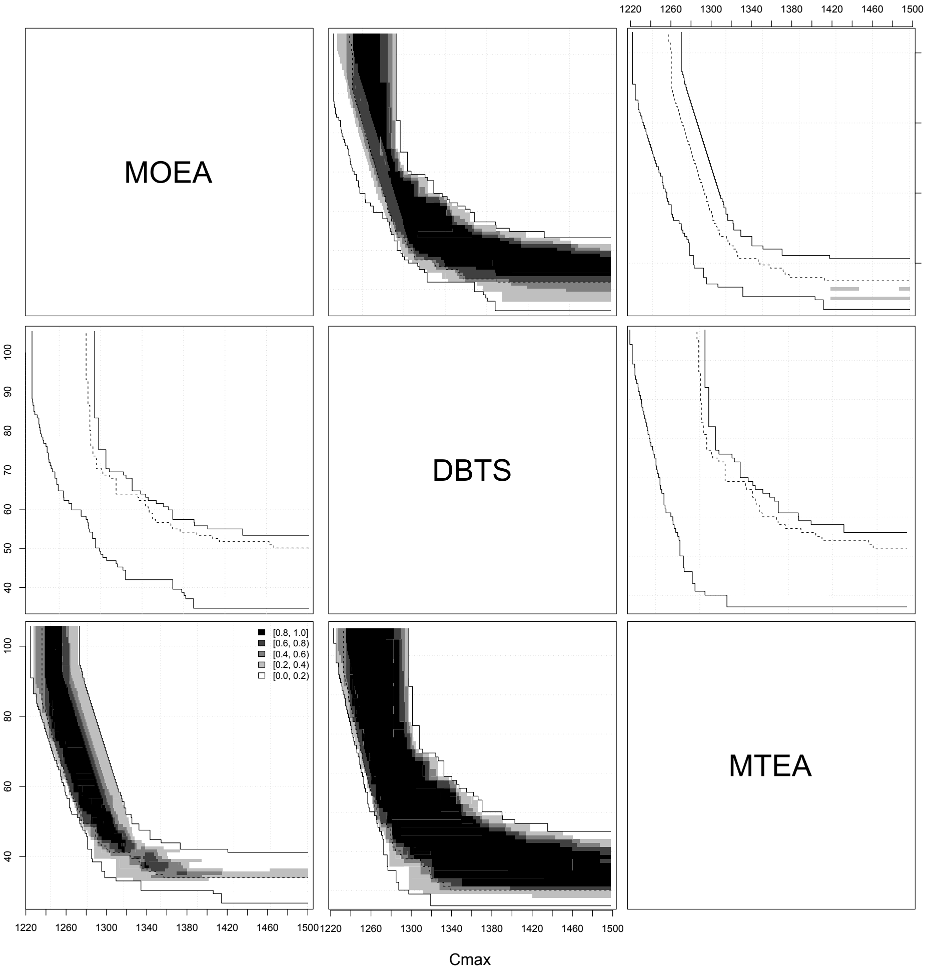

Figure 6: Comparison of MOEA, DBTS and MTEA using EAF differences for instance La40.

Instance	Avg. #solutions			Avg. runtime (s)		
	DBTS	MOEA	MTEA	DBTS	MOEA	MTEA
ABZ7	7.5	28.5	23.8	607.34	607.34	231.08
ABZ8	7.0	22.9	18.9	616.77	616.77	235.42
ABZ9	6.5	26.5	18.5	525.86	525.86	208.40
FT10	5.9	17.7	16.4	73.42	73.42	35.19
FT20	4.2	35.7	30.7	385.64	385.64	156.22
LA21	8.6	31.0	21.8	165.01	165.01	70.53
La24	8.6	34.6	24.8	162.85	162.85	69.36
La25	7.5	21.8	22.8	147.58	147.58	62.59
La27	7.3	42.7	33.8	422.51	422.51	170.24
La29	6.3	42.7	27.2	482.26	482.26	186.76
La38	8.1	38.6	22.6	354.56	354.56	128.42
La40	10.4	56.3	38.6	375.88	375.88	134.48
Average:	*7.3*	*33.3*	*25.0*	*359.97*	*359.97*	*140.72*

Table 3: Average number of solutions in non-dominated fronts and average runtime for MOEA, DBTS and MTEA.

we have considered variants thereof with different ratios of application of the intensification mechanism, namely applying DBTS only to the best individual of the population and to 10%, 25%, 50% and 75% of the remaining population. The obtained results are depicted in Figure 7. It shows the average HV values across all instances for each setup (in black), together with the CPU time (in gray) required by each variant of MTEA. We can see how the HV value increases as local search is applied to a greater proportion of the population, and so does the CPU time. However, the above experimental results suggest that the increase in computational cost incurred by intensification is still acceptable. Indeed, MTEA outperforms DBTS and MOEA when the local search is applied to all individuals, still requiring just 40% of the CPU time of any of the other algorithms.

Since the intensification step (DBTS) is applied to each individual in the population, MTEA might be suspected to essentially be a multi-start local search. Figure 8 illustrates the actual impact of the evolutionary part in MTEA, showing the evolution of MTEA compared to a multi-start local search using DBTS. More specifically, if N_p is the population size of MTEA, DBTS is applied to N_p randomly generated individuals, then to other N_p ran-

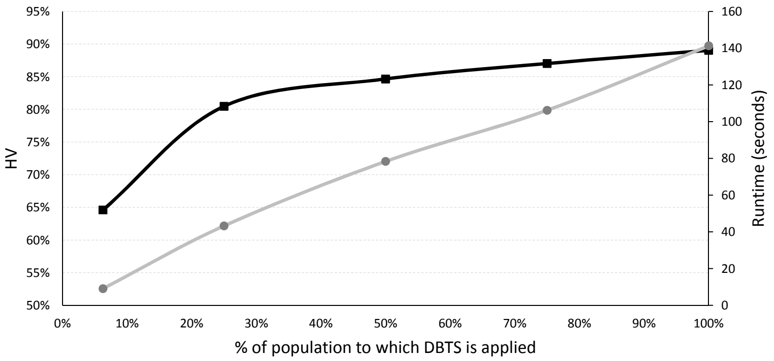

Figure 7: Impact of the intensification step in MOEA: HV (in black) and runtime (in gray) obtained when DBTS applied to different ratios of the population.

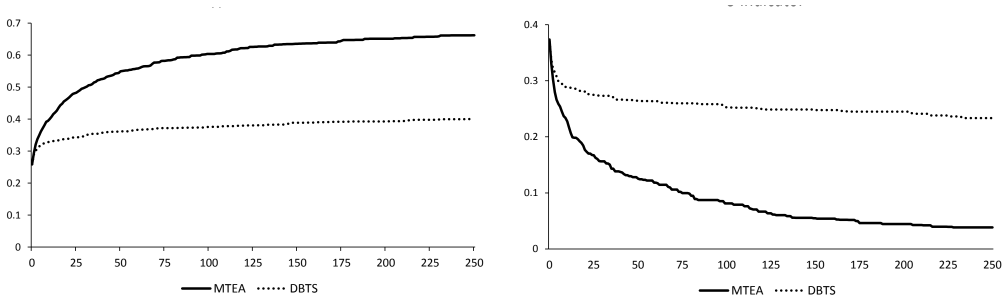

Figure 8: Evolution of the performance indicators HV (left) and $I_{\epsilon+}$ (right) for both MTEA and multistart-DBTS on instante La38

dom individuals and so forth, until the total number of DBTS applications is the same as for MTEA. The set of non-dominated solutions is also updated in the same manner as in MTEA, so the only difference between MTEA and multi-start MTEA lies in the evolution-related operators. Figure 8 shows for instance La38 the evolution along all generations of the HV and $I_{\epsilon+}$ values for MTEA together with the evolution, every N_p new starts, of the same indicators for the Pareto archive of multi-start DBTS. We can appreciate that, even though multi-start DBTS improves its set of non-dominated solutions, the convergence of MTEA is clearly superior. A similar behaviour can be observed on the remaining instances, showing that MOEA has a great impact in the hybrid algorithm's performance.

Regarding the overall performance of MTEA, looking again at Table 2 we notice that the standard deviation values for $I_{\epsilon+}$ are very low, so the solution obtained in an arbitrary execution of the algorithm is a good representative of its average behaviour, suggesting that MTEA is a stable algorithm. Also, the average value of $I_{\epsilon+}$ is below 0.1, which means that a little translation suffices for the solution obtained in a arbitrary execution to weakly dominate the reference set. Furthermore, the average cardinality of the non-dominated sets obtained by MTEA is 25 which means that 25% of the final population (with size 100) is in that set. Even though this is not a definite quality indicator, it constitutes an added value to the obtained results. The fact that MOEA obtains even larger sets of non-dominated solutions (with average cardinality of 33), might be explained because the NSGA-II template is designed to keep a high diversity in the population, while adding DBTS increases the algorithm's convergence.

Unfortunately, straightforward comparisons with other methods are not possible, since the a-priori robustness measure Rob^D is an original contribution of this work. For further reference on the quality of MTEA we propose to compare the obtained sets of non-dominated solutions with the best results of the single-objective memetic algorithm (MA) for the FJSP from [61], which was designed to optimise the makespan and proved to be competitive with other approaches from the literature. Figures 9 and 10 portray for each instance the ten sets of points in the objective space corresponding to the sets of non-dominated solutions of MTEA, where grey circles represent the elements of each set, joined by grey dotted lines, and a black bold line represents the approximation of the Pareto front which results from taking the non-dominated elements in the union of all these sets. We can also see the best expected makespan (a cross) and the average expected makespan (a vertical dotted line) obtained by MA on each instance. Notice that the X-axis corresponds to the expected makespan, while the Y-axis corresponds to the robustness value. The Rob^D value of the best solutions from MA has been calculated from the TFNs provided in the original paper.

As expected, the solutions obtained by MA are very good in terms of makespan but they do not perform that well in terms of robustness. For those instances where MTEA does not produce solutions as good in makespan as MA, we have on the other hand that the robustness Rob^D values for all solutions produced by MA are so poor that these solutions never dominate any of the solutions produced by MTEA. For the remaining instances, there are always solutions from MTEA that either dominate or are equal to the best solutions obtained by MA. Regarding the average behaviour, in most instances MTEA is able to find solutions that are better in expected makespan than

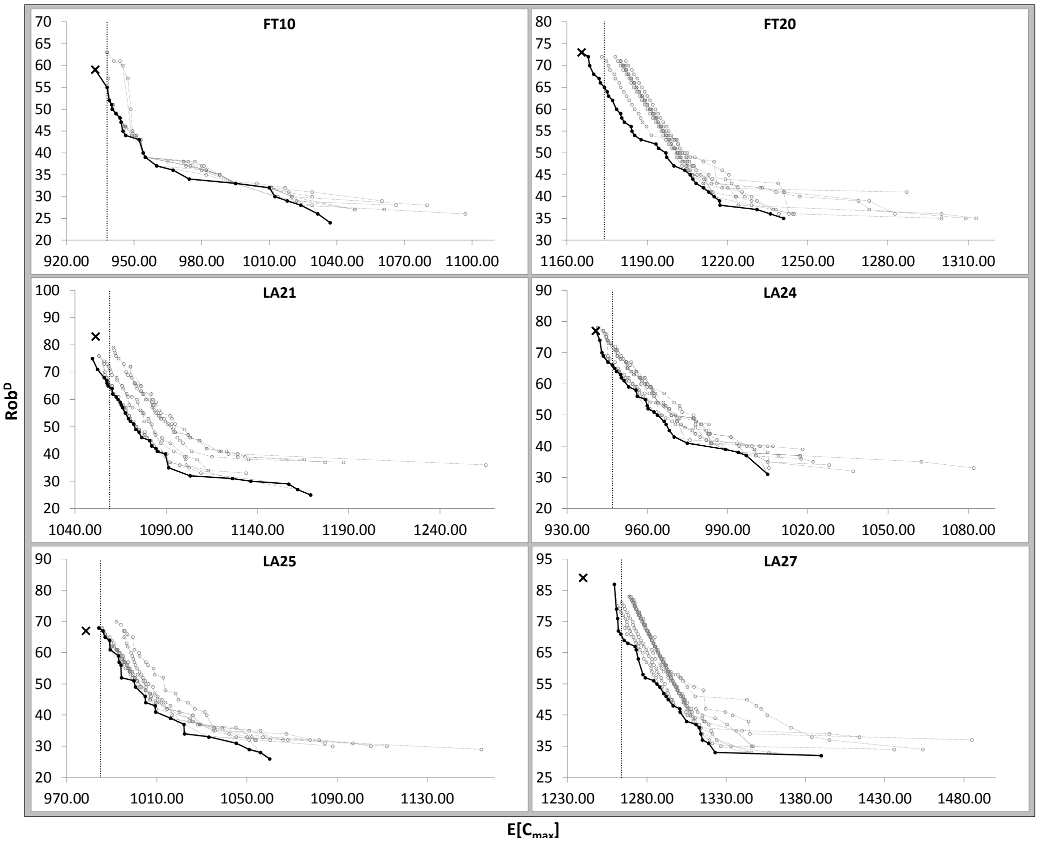

Figure 9: Detailed performance of MTEA and comparison with a single-objective MA from [61] for instances FT10, FT20, La21, 24, 25 and 27

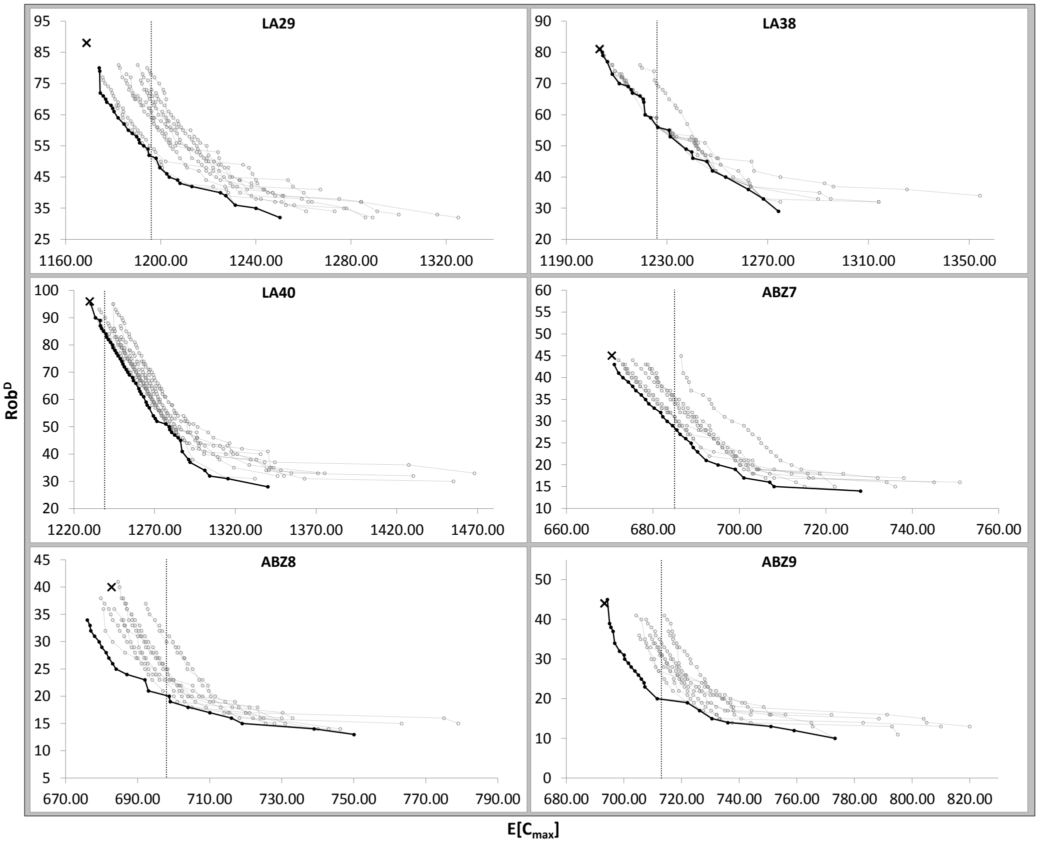

Figure 10: Detailed performance of MTEA and comparison with a single-objective MA from [61] for instances La29, 38, 40, ABZ7, 8 and 9

Inst.	Avg. $E[C_{max}]$		Inst.	Avg. $E[C_{max}]$	
	MA	MTEA		MA	MTEA
FT10	938.00	**938.60**	La29	1196.00	**1185.45**
FT20	**1174.00**	1177.85	La38	1226.00	**1205.15**
La21	**1059.00**	1061.05	La40	**1239.00**	1240.53
La24	947.00	**943.10**	ABZ7	685.00	**676.10**
La25	**985.00**	986.83	ABZ8	698.00	**685.23**
La27	**1264.00**	1266.35	ABZ9	713.00	**708.58**

Table 4: Average makespan values of solutions from MA and MTEA.

the average obtained by MA. In fact, it is common to get more than one set of solutions containing at least one solution improving the average performance of MA. Table 4 contains for all the instances the average expected makespan value of solutions of MA and the corresponding value of the solutions of the MTEA located in the extreme of the fronts corresponding to this objective. We can see that there are not significant differences between both methods as far as makespan values are concerned. In 6 of the 12 instances, MTEA provides better average expected makespan while in the other 6 differences are below 0.5%. Therefore, in a single run of the MTEA we can expect the non-dominated set of solutions to contain at least one element which, regarding the makespan objective, is better than or similar to the solution we can obtain with MA. Notice however that this similarity in expected makespan optimisation does not translate into similarity in robustness. Indeed, the Rob^D values of solutions given by MA are between twice and trice those we can expect to find in one run of MTEA by focusing on the other extreme of the resulting front. Since MTEA is a multiobjective algorithm that considers Rob^D, it is expected to improve MA (focused only on $E[C_{max}]$) on this aspect. What is perhaps less expected is that the multiobjective MTEA also improves MA in terms of its single objective, the expected makespan.

5.3. Quality of the a-priori robustness measure

One of the main contributions of this work is to include solution robustness as an objective of the optimisation process, by means of the definition of a new a-priori robustness measure. With the goal of assessing if this a-priori measure accurately represents the a-posteriori robustness, we perform a series of experiments to check the degree of correlation between both measures

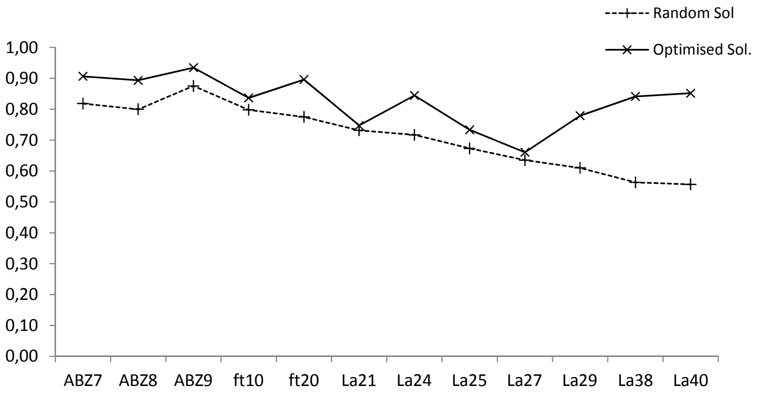

Figure 11: Detailed correlation index of a-priori (Rob^D) and a-posteriori ($\bar{\epsilon}$) robustness on random and optimised solutions computed by MTEA under Scenario I

Rob^D and ϵ-robustness (in fact, its surrogate $\bar{\epsilon}$).

First, for each instance we randomly generate a set of non-dominated solutions and measure both the average a-priori Rob^D and a-posteriori $\bar{\epsilon}$ robustness, where the latter is obtained under "Scenario I". The results show a moderate correlation index R^2 in the largest La instances (between 0.55 and 0.70) and a high positive correlation in the other ones, with an average correlation index across all instances of 0.71. Based on this, the proposed a-priori measure Rob^D may be considered to be a good predictive approach to the actual executed robustness.

The same experiment is repeated, but this time using the sets of non-dominated solutions obtained with MTEA. In this case the correlation between both measures is much stronger, with an average correlation index across all instances of 0.83. This is illustrated in Figure 11, which shows for all instances the correlation index of both robustness measures with random and optimised solutions.

The obtained results suggest that the simultaneous optimisation of the a-priori robustness and the expected makespan enhances the quality of this a-priori measure as an estimator of the real robustness. This supports the idea of optimising both objectives simultaneously, as we have done in this work.

Finally, the same experiments are run under "Scenario II". In this case, the obtained average correlation indices are 0.83 and 0.85 respectively. This small gap shows that the optimisation proposed in this work has a greater impact in terms of robustness when there is a greater likelihood of deviations from the most probable durations, which is the case of "Scenario I". Notice nonetheless that the correlation index after optimisation is quite high under both simulation scenarios, which illustrates the high fiability of the proposed metric Rob^D as a predictive estimate of the schedule's executed robustness.

6. Conclusions

We have considered a variant of the job shop problem which incorporates uncertainty in operation durations in the form of fuzzy numbers. In an attempt to reduce the gap between academic and real-world problems, we have proposed to optimise not just the time needed to perform all tasks, known as makespan, but also the robustness of the solutions, understood as an overall acceptable performance under variations in the input data.

To solve the resulting problem, we have proposed a new hybrid algorithm, named MTEA, which combines a multiobjective evolutionary algorithm (MOEA) with a dominance-based tabu search (DBTS) by applying the latter to every newly generated chromosome. An important contribution to the resulting algorithm is the use of a new dominance-based method as criterion to select the "best" neighbour of a current solution at each iteration of DBTS. The experimental results show that, even though the DBTS evaluated on random solutions has worse performance than the MOEA, it provides a good intensification mechanism, with MTEA obtaining better results than MOEA or DBTS run independently, thanks to the synergy between both components of the hybrid method. The experimental results also show that MTEA is competitive in terms of makespan optimisation with a single-objective memetic algorithm from the literature (MA), with the sets of non-dominated solutions of MTEA containing solutions similar and even slightly better in average than those of MA. Additionally, as expected, the non-dominated fronts of solutions produced by MTEA contain much more robust solutions. Finally, given that the evaluation of the fitness function is a critical operation in any evolutionary algorithm, we have proposed a new predictive measure for the robustness, Rob^D, that can be evaluated in constant time. The existing correlation between Rob^D and the a-posteriori robustness measured on real executions has been assessed using two different Monte-Carlo simulations. We have also seen that the reliability of the prediction clearly benefits from the optimisation process when larger deviations from the most probable durations are more likely to happen.

Acknowledgements

This research has been supported by the Spanish Government under Grants FEDER TIN2013-46511-C2-2-P and MTM2014-55262-P.

References

[1] S. Abdullah and M. Abdolrazzagh-Nezhad. Fuzzy job-shop scheduling problems: A review. *Information Sciences*, 278:380–407, 2014.

[2] H. Aissi, C. Bazgan, and D. Vanderpooten. Min-max and min-max regret versions of combinatorial optimization problems: A survey. *European Journal of Operational Research*, 197:427–438, 2009.

[3] H. Aytung, M. A. Lawley, K. McKay, M. Shantha, and R. Uzsoy. Executing production schedules in the face of uncertainties: A review and some future directions. *European Journal of Operational Research*, 161:86–110, 2005.

[4] C. Baudrit and D. Dubois. Practical representations of incomplete probabilistic knowledge. *Computational Statistics & Data Analysis*, 51:86–108, 2006.

[5] J. Behnamian and S. Fatemi Ghomi. Multi-objective fuzzy multiprocessor flowshop scheduling. *Applied Soft Computing*, 21:139–148, 2014.

[6] J. Bidot, T. Vidal, and P. Laboire. A theoretic and practical framework for scheduling in stochastic environment. *Journal of Scheduling*, 12:315–344, 2009.

[7] C. Bierwirth. A generalized permutation approach to jobshop scheduling with genetic algorithms. *OR Spectrum*, 17:87–92, 1995.

[8] C. Bierwirth, D. C. Mattfeld, and H. Kopfer. On permutation representations for scheduling problems. In *PPSN IV: Proceedings of the 4th International Conference on Parallel Problem Solving from Nature*, pages 310–318, London, UK, 1996. Springer-Verlag.

[9] M. Brunelli and J. Mezei. How different are ranking methods for fuzzy numbers? A numerical study. *International Journal of Approximate Reasoning*, 54:627–639, 2013.

[10] T. Buer and G. Pankratz. Grasp with hybrid path relinking for bi-objective winner determination in combinatorial transportation auctions. *BuR – Business Research*, 3:192–213, 2010.

[11] S. Chanas and M. Nowakowski. Single value simulation of fuzzy variable. *Fuzzy Sets and Systems*, 25:43–57, 1988.

[12] T.-C. Chiang, H.-C. Cheng, and L.-C. Fu. NNMA: An effective memetic algorithm for solving multiobjective permutation flow shop scheduling problems. *Expert Systems with Applications*, 38:5986–5999, 2011.

[13] M. Chica, O. Cordón, S. Damas, and J. Bautista. Multiobjective memetic algorithms for time and space assembly line balancing. *Engineering Applications of Artificial Intelligence*, 25:254–273, 2012.

[14] S. Dabia, E.-G. Talbi, T. van Woensel, and T. De Kok. Approximating multi-objective scheduling problems. *Computers & Operations Research*, 40:1165–1175, 2013.

[15] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[16] M. Dell' Amico and M. Trubian. Applying tabu search to the job-shop scheduling problem. *Annals of Operational Research*, 41:231–252, 1993.

[17] S. Destercke and I. Couso. Ranking of fuzzy intervals seen through the imprecise probabilistic lens. *Fuzzy Sets and Systems*, In press, 2014.

[18] D. Dubois. Possibility theory and statistical reasoning. *Computational Statistics & Data Analysis*, 51:47–69, 2006.

[19] D. Dubois. The role of fuzzy sets in decision sciences: Old techniques and new directions. *Fuzzy Sets and Systems*, 184:3–28, 2011.

[20] D. Dubois, H. Fargier, and P. Fortemps. Fuzzy scheduling: Modelling flexible constraints vs. coping with incomplete knowledge. *European Journal of Operational Research*, 147:231–252, 2003.

[21] D. Dubois, L. Foulloy, G. Mauris, and H. Prade. Probability-possibility transformations, triangular fuzzy sets and probabilistic inequalities. *Reliable Computing*, 10:273–297, 2004.

[22] D. Dubois and H. Prade. *Possibility Theory: An Approach to Computerized Processing of Uncertainty.* Plenum Press, New York (USA), 1986.

[23] D. Dubois and H. Prade. The mean value of a fuzzy number. *Fuzzy Sets and Systems*, 24:279–300, 1987.

[24] D. Dubois, H. Prade, and S. Sandri. On possibility/probability transformations. In *Fuzzy Logic*, volume 12 of *Theory and Decision Library*, pages 103–112. Kluwer Academic, 1993.

[25] P. Fortemps. Jobshop scheduling with imprecise durations: a fuzzy approach. *IEEE Transactions of Fuzzy Systems*, 7:557–569, 1997.

[26] M. Gen and L. Lin. Multiobjective evolutionary algorithm for manufacturing scheduling problems. *Journal of Intelligent Manufacturing*, 25:849–866, 2014.

[27] O. A. Ghrayeb. A bi-criteria optimization: minimizing the integral value and spread of the fuzzy makespan of job shop scheduling problems. *Applied Soft Computing*, 2(3):197–210, 2003.

[28] F. Glover. Tabu search–part I. *ORSA Journal on Computing*, 1(3):190–206, 1989.

[29] F. Glover. Tabu search–part II. *ORSA Journal on Computing*, 2(1):4–32, 1990.

[30] M. A. González, A. Oddi, R. Rasconi, and R. Varela. Scatter search with path relinking for the job shop with time lags and setup times. *Computers & Operations Research*, 60:37–54, 2015.

[31] I. González Rodríguez, J. Puente, C. R. Vela, and R. Varela. Semantics of schedules for the fuzzy job shop problem. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, 38(3):655–666, 2008.

[32] I. González Rodríguez, C. R. Vela, A. Hernández-Arauzo, and J. Puente. Improved local search for job shop scheduling with uncertain durations. In *Proceedings of the Nineteenth International Conference on Automated Planning and Scheduling (ICAPS-2009)*, pages 154–161, Thesaloniki, 2009. AAAI Press.

[33] I. González Rodríguez, C. R. Vela, J. Puente, and R. Varela. A new local search for the job shop problem with uncertain durations. In *Proceedings of the Eighteenth International Conference on Automated Planning and Scheduling (ICAPS-2008)*, pages 124–131, Sidney, 2008. AAAI Press.

[34] M. Guzek, J. Pecero, B. Dorronsoro, and P. Bouvry. Multi-objective evolutionary algorithms for energy-aware scheduling on distributed computing systems. *Applied Soft Computing*, 24:432–446, 2014.

[35] S. Heilpern. The expected value of a fuzzy number. *Fuzzy Sets and Systems*, 47:81–86, 1992.

[36] W. Herroelen and R. Leus. Project scheduling under uncertainty: Survey and research potentials. *European Journal of Operational Research*, 165:289–306, 2005.

[37] H. Hoogeveen. Multicriteria scheduling. *European Journal of Operational Research*, 167:592–623, 2005.

[38] Y. Hu, M. Yin, and X. Li. A novel objective function for job-shop scheduling problem with fuzzy processing time and fuzzy due date using differential evolution algorithm. *International Journal of Advanced Manufactoring Technology*, 56:1125–1138, 2011.

[39] H. Ishibuchi, Y. Hitotsuyanagi, N. Tsukamoto, and Y. Nojima. Use of biased neighborhood structures in multiobjective memetic algorithms. *Soft Computing*, 13(8–9):795–810, 2009.

[40] A. Jaszkiewicz. Do multiple-objective metaheuristics deliver on their promises? A computational experiment on the set-covering problem. *IEEE Transactions on Evolutionary Computation*, 7(2):133–143, 2003.

[41] S. Jia and Z.-H. Hu. Path-relinking tabu search for the multi-objective flexible job shop scheduling problem. *Computers & Operations Research*, 47:11–26, 2014.

[42] R. Kalaï, C. Lamboray, and D. Vanderpooten. Lexicographic α-robustness: An alternative to min-max criteria. *European Journal of Operational Research*, 220:722–728, 2012.

[43] A. Kasperski and M. Kule. Choosing robust solutions in discrete optimization problems with fuzzy costs. *Fuzzy Sets and Systems*, 160:667–682, 2009.

[44] J. Knowles and D. Corne. M-PAES: A memetic algorithm for multiobjective optimization. In *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, volume 1, pages 325–332, 2000.

[45] J. D. Knowles and D. W. Corne. Approximating the nondominated front using the Pareto archived evolution strategy. *Evolutionary Computation*, 8(2):149–172, 2000.

[46] M. Kuroda and Z. Wang. Fuzzy job shop scheduling. *International Journal of Production Economics*, 44:45–51, 1996.

[47] A. Lara, G. Sánchez, C. A. Coello Coello, and O. Schütze. HCS: A new local search strategy for memetic multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 14(1):112–132, 2010.

[48] P. Li and B. Liu. Entropy of credibility distributions for fuzzy variables. *IEEE Transactions on Fuzzy Systems*, 16(1):123–129, 2008.

[49] A. Liefooghe, J. Humeau, S. Mesmoudi, L. Jourdan, and E.-G. Talbi. On dominance-based multiobjective local search: design, implementation and experimental analysis on scheduling and traveling salesman problems. *Journal of Heuristics*, 18(2):317–352, 2012.

[50] B. Liu and Y. K. Liu. Expected value of fuzzy variable and fuzzy expected value models. *IEEE Transactions on Fuzzy Systems*, 10:445–450, 2002.

[51] Y. Mei, K. Tang, and X. Yao. Decomposition-based memetic algorithm for multiobjective capacitated arc routing problem. *IEEE Transactions on Evolutionary Computation*, 15(2):151–165, 2011.

[52] R. Mencía, M. Sierra, C. Mencía, and R. Varela. Memetic algorithms for the job shop scheduling problem with operators. *Applied Soft Computing*, 34:94–105, 2015.

[53] E. Nowicki and C. Smutnicki. A fast taboo search algorithm for the job shop scheduling problem. *Management Science*, 42:797–813, 1996.

[54] E. Nowicki and C. Smutnicki. An advanced tabu search algorithm for the job shop problem. *Journal of Scheduling*, 8(2):145–159, 2005.

[55] J. J. Palacios, M. A. González, C. R. Vela, I. González-Rodríguez, and J. Puente. Genetic tabu search for the fuzzy flexible job shop problem. *Computers & Operations Research*, 54:74–89, 2015.

[56] J. J. Palacios, I. González-Rodríguez, C. R. Vela, and J. Puente. Robust swarm optimisation for fuzzy open shop scheduling. *Natural Computing*, 13(2):145–156, 2014.

[57] J. J. Palacios, I. González-Rodríguez, C. R. Vela, and J. Puente. Coevolutionary makespan optimisation through different ranking methods for the fuzzy flexible job shop. *Fuzzy Sets and Systems*, 278:81–97, 2015.

[58] J. J. Palacios, C. R. Vela, I. González-Rodríguez, and J. Puente. Schedule generation schemes for job shop problems with fuzziness. In T. Schaub, G. Friedrich, and B. O'Sullivan, editors, *Proceedings of ECAI 2014*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pages 687–692. IOS Press, 2014.

[59] L. Paquete, T. Schiavinotto, and T. Stützle. On local optima in multiobjective combinatorial optimization problems. *Annals of Operations Research*, 156:83–97, 2007.

[60] M. L. Pinedo. *Scheduling. Theory, Algorithms, and Systems.* Springer, third edition, 2008.

[61] J. Puente, C. R. Vela, and I. González-Rodríguez. Fast local search for fuzzy job shop scheduling. In *Proceedings of ECAI 2010*, pages 739–744. IOS Press, 2010.

[62] B. Qian, L. Wang, D.-X. Huang, and X. Wang. Scheduling multiobjective job shops using a memetic algorithm based on differential evolution. *International Journal of Advanced Manufacturing Technology*, 35:1014–1027, 2008.

[63] B. Roy. Robustness in operational research and decision aiding: A multi-faceted issue. *European Journal of Operational Research*, 200:629–638, 2010.

[64] M. Sakawa and R. Kubota. 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.

[65] R. Słowiński and M. Hapke, editors. *Scheduling Under Fuzziness*, volume 37 of *Studies in Fuzziness and Soft Computing.* Physica-Verlag, 2000.

[66] E.-G. Talbi. *Metaheuristics. From Design to Implementation.* Wiley, 2009.

[67] P. Van Laarhoven, E. Aarts, and K. Lenstra. Job shop scheduling by simulated annealing. *Operations Research*, 40:113–125, 1992.

[68] B. Wang, Q. Li, X. Yang, and X. Wang. Robust and satisfactory job shop scheduling under fuzzy processing times and flexible due dates. In *Proc. of the 2010 IEEE International Conference on Automation and Logistics*, pages 575–580, 2010.

[69] B. K. Wong and V. S. Lai. A survey of the application of fuzzy set theory in production and operations management: 1998–2009. *International Journal of Production Economics*, 129:157–168, 2011.

[70] C. Y. Zhang, P. Li, Y. Rao, and Z. Guan. A very fast TS/SA algorithm for the job shop scheduling problem. *Computers & Operations Research*, 35:282–294, 2008.

[71] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, 2003.