Local search for flexible due-date satisfaction in fuzzy job shop scheduling

Abstract

We consider the job shop scheduling problem with fuzzy sets modelling uncertain durations and flexible due dates. With the goal of maximising due-date satisfaction under uncertainty, we define a neighbourhood structure for local search, analyse its theoretical properties and provide a neighbour-estimation procedure. Additionally, a simple hill-climbing local search using the defined neighbourhood is incorporated to an existing genetic algorithm, so the resulting memetic algorithm is run on a set of benchmarks. The obtained results further illustrate the potential of our proposal.

1 Introduction

Scheduling problems are pervasive in a growing number of application domains. One of the most relevant problems in this family is the job shop, since it is considered to be a good model for many practical applications as well as posing a challenge due to its complexity [Pinedo, 2016]. Even though the most common objective is to find solutions with minimum makespan, due-date satisfaction is receiving increasing attention in recent years [González *et al.*, 2013; Kuhpfahl and Bierwirth, 2016]. On-time fulfilment emerges as a primary goal in modern pull-oriented supply chain systems and keeping job due dates is a prerequisite for serving customers within the promised delivery time and avoiding out-of-stocks or delay-compensation costs. Hence the importance of considering due-date satisfaction measures.

Traditionally, it has been assumed that scheduling takes place in static and certain environments. However, for many real-world scheduling problems design variables are subject to perturbations or changes, causing optimal solutions to the original "ideal" problem to be of little or no use in practice. It is also common to handle all constraints as sharp, while in some cases there is certain flexibility and constraints are better expressed in terms of preference, so it is possible to satisfy them to a certain degree.

A source of uncertainty in scheduling is activity durations. Within the great diversity of approaches to this issue, fuzzy sets and possibility theory provide an interesting framework, with a tradeoff between the expressive power of probability and its associated computational complexity and knowledge demands. Fuzzy sets can also be used to model flexibility or gradeness in certain management constraints such as due dates [Dubois *et al.*, 2003].

The variant of job shop scheduling problem with fuzzy durations and, optionally, fuzzy due dates, is called fuzzy job shop. Most contributions in the literature concentrate on minimising the project's makespan, but some authors have addressed the maximisation of due-date satisfaction, either on its own or in a multiobjective setting, combined with makespan [Abdullah and Abdolrazzagh-Nezhad, 2014].

In this paper, we intend to advance in the study of the fuzzy job shop scheduling problem, and in particular, in local search methods to maximise due-date satisfaction when uncertain task durations and flexible due dates are fuzzy sets.

2 The Fuzzy Job Shop Problem

The classical *job shop scheduling problem*, *JSP* in short, consists in scheduling a set of jobs $\{J_1, \ldots, J_n\}$ on a set $\{M_1, \ldots, M_m\}$ of physical resources or machines, subject to a set of constraints. There are *precedence constraints*, so each job $J_j, j = 1, \ldots, n$, consists of m tasks $\{\theta_{j1}, \ldots, \theta_{jm}\}$ to be sequentially scheduled. Also, there are *capacity constraints*, whereby each task θ_{jk} requires the uninterrupted and exclusive use of one of the machines for its whole processing time p_j. Additionally, each job J_j has a due date d_j by which it is desirable that the job be completed. A solution to this problem is a schedule, i.e. an allocation of starting times for each task, which is *feasible* (in the sense that all precedence and resource constraints hold) as well as *optimal* according to some criterion, in our case, maximal due-date satisfaction.

2.1 Fuzzy Durations and Flexible Due Dates

In real-life applications, it is difficult, if not impossible, to foresee in advance the exact time it will take to process a task. It is reasonable however to have some incomplete knowledge about the duration, possibly based on previous experience. The crudest representation of such uncertain knowledge would be a human-originated confidence interval and, 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 $\widehat{a} = (a^1, a^2, a^3)$, 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 func-

tion takes the following triangular shape:

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

Triangular fuzzy numbers (or, more generally, fuzzy intervals) are widely used in scheduling as a model for uncertain processing times [Abdullah and Abdolrazzagh-Nezhad, 2014; Dubois *et al.*, 2003; Palacios *et al.*, 2016].

The sum and maximum of fuzzy numbers, needed to handle them in the job shop, are usually defined by extending the corresponding operations on real numbers. The resulting addition is pretty straightforward, so for any pair of TFNs $\widehat{a}$ and $\widehat{b}$ we have $\widehat{a} + \widehat{b} = (a^1 + b^1, a^2 + b^2, a^3 + b^3)$. Computing the extended maximum is not that simple and the set of TFNs is not even closed under this operation. Hence, it is common in the fuzzy scheduling literature to approximate the maximum of two TFNs as $\max(\widehat{a}, \widehat{b}) \approx (\max\{a^1, b^1\}, \max\{a^2, b^2\}, \max\{a^3, b^3\})$. Besides its extended use, several arguments can be given in favour of this approximation (cf. [Palacios *et al.*, 2016]).

Finally, it is possible to compute the *expected value* of a TFN $\widehat{a}$ as

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

Notice that the expected value is linear with respect to addition and multiplication by a scalar quantity [Heilpern, 1992].

Regarding due dates, there is often certain flexibility. Consider the case where there is a preferred delivery date d^1, but some delay may be allowed up to a later date d^2. Satisfying the due-date constraint thus becomes a matter of degree. A fuzzy set $\widetilde{d} = (d^1, d^2)$ can be used to model such gradual satisfaction level with a decreasing membership function:

$$\mu_{\widetilde{d}}(x) = \begin{cases} 1 & : x \leq d^1 \\ \frac{x-d^2}{d^1-d^2} & : d^1 < x \leq d^2 \\ 0 & : d^2 < x \end{cases} \quad (3)$$

This expresses a flexible threshold "less than", representing the satisfaction level $sat(t) = \mu_{\widetilde{d}}(t)$ for the ending date t of the job [Dubois *et al.*, 2003].

2.2 The Disjunctive Graph Model Representation

For the deterministic job shop with makespan minimisation as objective, the disjunctive graph representation provides the basis for many heuristic solving methods [Błażewicz *et al.*, 2000]. Consequently, variants of the model that cope, among others, with crisp due-date tardiness or lateness and with uncertain durations have been proposed [Kuhpfahl and Bierwirth, 2016; González *et al.*, 2013; González Rodríguez *et al.*, 2008b; Pinedo, 2016]. Here we build on these to provide a model for the fuzzy job shop with flexible due dates.

A disjunctive graph is a directed graph $G = (V, A \cup D)$ representing a problem instance. Each node in the set V either represents a task or is a start *s* or end node e_j, $1 \leq j \leq n$, corresponding to dummy tasks with null processing times. Arcs in A are called *conjunctive arcs* and represent job-precedence constraints, including arcs from node *s* to the first task of each job and arcs from the last task of each job j to the corresponding sink node e_j. Arcs in D are called *disjunctive arcs* and represent resource constraints, so D is partitioned into subsets D_k, $D = \cup_{k=1,\ldots,m} D_k$, where D_k includes an arc for each pair of tasks requiring machine M_k. All arcs are weighted with the processing time of the task at the source node (a TFN in our case).

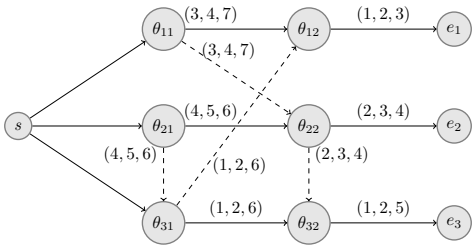

Figure 1: Solution graph $G(\sigma)$ for the processing order $\sigma = (\theta_{11}\ \theta_{21}\ \theta_{31}\ \theta_{22}\ \theta_{12}\ \theta_{32})$.

Finding a solution to the fuzzy JSP essentially consists in establishing partial task processing orders on all machines, represented by a linear processing order σ or an acyclic subgraph $G(\sigma)$ of G, $G(\sigma) = (V, A \cup R(\sigma))$, where $R(\sigma) = \cup_{k=1\ldots m} R_k(\sigma)$, $R_k(\sigma)$ being a hamiltonian selection of D_k. This is often referred to as *solution graph*. A feasible schedule (starting and completion times of all tasks) may be easily computed by simply propagating constraints in $G(\sigma)$ using the sum and maximum of TFNs.

In the deterministic case, critical paths (longest paths from the start node to a sink node) play an important role in the design of heuristic methods. Extending the notion of criticality to the problem with fuzzy durations is not trivial, wit diverse proposals co-existing in the literature (cf. [Dubois *et al.*, 2003; Fortemps, 1997]). Here we follow [González Rodríguez *et al.*, 2008b] and, given a solution graph $G(\sigma)$, consider three *parallel solution graphs* $G^i(\sigma)$, $i = 1, 2, 3$, with identical structure to $G(\sigma)$ but where the cost of any arc (x, y) is p^i_x, the i-th component of the processing time $\widehat{p}_x$ for the task at the source node x. Weights in each parallel graph $G^i(\sigma)$ are deterministic, so a critical path in $G^i(\sigma)$ for a job j is the longest path from node s to node e_j. The set of *critical paths* in $G(\sigma)$ is defined as the union of critical paths in $G^i(\sigma)$, $i = 1, 2, 3$. *Critical nodes and arcs* are those in a critical path. Unlike the deterministic case, the union set of critical paths from s to e_j for all $j = 1, \ldots, n$ is not a tree.

Figure 1 contains an example of solution graph for a toy problem with 3 jobs and 2 machines. It is easy to see that, for job J_3, $(s\ \theta_{21}\ \theta_{22}\ \theta_{32}\ e_3)$ is critical in $G^1(\sigma)$ and $G^2(\sigma)$, while the critical path to e_3 in the third parallel graph is $(s\theta_{21}\theta_{31}\theta_{32}e_3)$. This illustrates the fact that the set of critical paths to a sink node does not necessarily have a tree structure.

Let σ be a task processing order and for every task x with processing time $\widehat{p}_x$, let $PM_x(\sigma)$ and $SM_x(\sigma)$ denote respectively the tasks preceding and succeeding x in the machine sequence provided by σ (in $R(\sigma)$), and let PJ_x and SJ_x de-

note respectively the predecessor and successor tasks of x in the job sequence (in A). We define the *head* of task x $\widehat{r}_x(\sigma)$ as the starting time of x, a TFN given by:

$$\widehat{r}_x(\sigma) = \max\{\widehat{r}_{PJ_x}(\sigma) + \widehat{p}_{PJ_x}, \widehat{r}_{PM_x(\sigma)}(\sigma) + \widehat{p}_{PM_x(\sigma)}\}, \tag{4}$$

where $\widehat{r}_s(\sigma) = 0$. It corresponds to the length of a longest path in $G(\sigma)$ from s to x. We also define the *tail* of task x relative to j, denoted $\widehat{q}_{x,j}(\sigma)$, as:

$$\widehat{q}_{x,j}(\sigma) = \max\{\widehat{q}_{SJ_x,j}(\sigma) + \widehat{p}_{SJ_x}, \widehat{q}_{SM_x(\sigma),j}(\sigma) + \widehat{p}_{SM_x(\sigma)}\}, \tag{5}$$

where $\widehat{q}_{e_j,l}(\sigma) = 0$ if $l = j$ and $-\infty$ otherwise and $\widehat{q}_{SM_x(\sigma),j}(\sigma) = -\infty$ if x is the last task to be processed in its machine. The tail represents the time left after x until a job is completed, i.e., the length of a longest path in $G(\sigma)$ from the successors of x to the job's sink node. Notice that the tail of a task x w.r.t. j needs to be computed even if x does not belong to job J_j and it is trivial, $-\infty$, when there is no path from x to e_j. When there is no confusion regarding the processing order, we may simply write $\widehat{r}_x$ and $\widehat{q}_{x,j}$.

Let $\widehat{c}_x(\sigma)$ and $\widehat{c}_j(\sigma)$ denote respectively the completion times of a task x and job j (or simply $\widehat{c}_x$ and $\widehat{c}_j$). Clearly, $\widehat{c}_x = \widehat{r}_x + \widehat{p}_x$ and $\widehat{c}_j = \widehat{r}_{e_j}$. For each parallel graph $G^i(\sigma)$, r^i_x is the length of the longest path from node s to node x and $q^i_{x,j} + p^i_x$ is the length of the longest path from node x to node e_j if this path exists. Hence, for all i, $r^i_x + p^i_{x,j} + q^i_x$ is the length of the longest path from node s to node e_j through node x in $G^i(\sigma)$; it is a lower bound of the i-th component of the $\widehat{c}_j$, being equal to c^i_j if node x belongs to a critical path in $G^i(\sigma)$ for J_j. The following properties ensue trivially from the definition.

Proposition 1. *If an arc $v = (x, y)$ is critical for some job j, then $\exists i, r^i_x + p^i_x = r^i_y$. A task x is critical for some job j if and only if $\exists i, r^i_x + p^i_x + q^i_{x,j} = c^i_j$.*

2.3 Due-Date Satisfaction for Fuzzy Schedules

The schedule obtained from a task processing order σ is fuzzy in the sense that the starting and completion times of all tasks are TFNs, interpreted as possibility distributions on the values that the times may take. In particular, every job's completion time is no longer a real number but a TFN $\widehat{c}$, so the degree to which $\widehat{c}$ satisfies a due-date constraint $\widetilde{d}$ may no longer be measured by direct evaluation of $\mu_{\widetilde{d}}$.

The most common approach to this problem is to use the so-called *agreement index* [Abdullah and Abdolrazzagh-Nezhad, 2014], which essentially corresponds to the degree of subsethood of the completion time into the due date. However, treating both the due date and the completion time equally can be objected to since they have very different meanings, with one of them modelling uncertainty while the other one models flexibility. Also, fuzzy schedules can be seen as predictive or a-priori schedules, found when the duration of tasks is not exactly known and a set of possible scenarios must be taken into account [González Rodríguez *et al.*, 2008a]. At this stage, the degree of due-date satisfaction can only be approximated; only after tasks are executed according to the ordering provided by the schedule are real durations known and, hence, can the actual due-date satisfaction degree be measured. The agreement index thus corresponds to an estimate of the due-date satisfaction degree that would be obtained after execution, but it tends to be excessively optimistic (and innacurate) since it ignores the information provided by the portion of $\widehat{c}$ outside the support of the due date $\widetilde{d}$.

We propose instead to measure the *expected satisfaction degree (ESD)*, defined as the degree to which the expected completion time $E[\widehat{c}]$ satisfies the due date $\widetilde{d}$:

$$ESD(\widehat{c}, \widetilde{d}) = \mu_{\widetilde{d}}(E[\widehat{c}]). \tag{6}$$

This estimate ranges between 0, when the due date is not expected to be satisfied at all, and 1, when it is expected that the due date is fully satisfied. In the case that the completion time is crisp (absence of uncertainty), ESD becomes the usual measure of flexible due-date satisfaction given by the membership function.

Having built a schedule from σ, the expected satisfaction degree $ESD(\widehat{c}_j(\sigma), \widetilde{d}_j)$, denoted ESD_j for short, measures to what degree the job's flexible due date $\widetilde{d}_j$ is satisfied in this schedule, $j = 1, \ldots, n$. The overall value of due-date satisfaction for the schedule can be obtained as the average expected satisfaction degree $ESD_{avg} = \frac{1}{n}\sum_{j=1,\ldots,n} ESD_j$, a value that needs to be maximised. Accordingly, an optimal task ordering is one that yields a schedule with maximal ESD_{avg}.

Definition 1. Let Σ be the set of feasible task processing orders. A task processing order $\sigma_0 \in \Sigma$ is said to be *optimal* for ESD_{avg} if and only if $ESD_{avg}(\sigma_0) = \max\{ESD_{avg}(\sigma) : \sigma \in \Sigma\}$.

The resulting job shop problem, with fuzzy processing times and fuzzy due dates, and where the objective is to maximise the aggregated expected satisfaction degree ESD_{avg} can be denoted $J|\widehat{p}_j, \widetilde{d}_j|ESD_{avg}$ according to the three-field notation from [Graham *et al.*, 1979].

3 Local Search to Maximise ESD_{avg}

Local search methods, either on their own or combined with other metaheuristic methods, provide some of the most competitive solutions to different variants of the job shop [Bierwirth and Kuhpfahl, 2017; González *et al.*, 2015; Peng *et al.*, 2015; Puente *et al.*, 2010].

Generally speaking, the local search starts from a given solution and at each step it selects a promising element from a neighbourhood structure to replace the current solution, until a stopping criterion is met. In simple hill climbing, the chosen neighbour is the first one improving the objective value of the current solution. The search stops when it reaches a solution without improving neighbours. This strategy is very fast compared to other local search strategies, making it appealing for large neighbourhoods.

Two key points of this algorithm are the definition of the neighbourhood structure and the method used to estimate a neighbour's quality. The next subsections describe, respectively, a new neighbourhood structure for *FJSP* to maximise ESD_{avg} (including some properties thereof) and a procedure for ESD_{avg} estimation.

3.1 Neighbourhood structure

For the classical job shop, a well-known neighbourhood, which relies on the concept of critical path, is that proposed in [Van Laarhoven *et al.*, 1992], later extended to the fuzzy case in [González Rodríguez *et al.*, 2008b]. In this structure, neighbours are obtained by reversing single critical arcs.

Based on these, we propose a new neighbourhood structure for ESD_{avg} maximisation. The intuition is that, for ESD_{avg} to increase in a neighbouring solution, it must be the case that at least one of the expected satisfaction degree values ESD_j improves. This occurs only if the due date d_j is not yet fully satisfied and the completion time $\widehat{c}_j(\sigma)$ of the corresponding job is reduced or, equivalently, the length of the longest path from the start node s to the end node e_j in the solution graph is reduced.

Let CP_j denote the set of critical paths for job $J_j, j = 1, \ldots, n$. An improvement in $\widehat{c}_j(\sigma)$ (and hence in ESD_{avg}) can only be obtained by reversing machine arcs belonging to paths in CP_j for some $j = 1, \ldots, n$. Furthermore, since $ESD_j \leq 1$ for $j = 1, \ldots, n$, reducing the completion time of a job such that $ESD_j = 1$ (i.e., its due date $\widetilde{d}_j$ is already fully satisfied) cannot improve ESD_{avg} either.

Proposition 2. *Let σ a feasible processing order, let v an arc that is not critical for any job J_j such that $ESD_j(\sigma) < 1$ in $G(\sigma)$ and let $\sigma_{(v)}$ denote the processing order obtained from σ after reversing arc v in $G(\sigma)$. Then*

$$\forall j, \quad ESD_j(\sigma_{(v)}) \leq ESD_j(\sigma). \tag{7}$$

Proof. Let J_j be a job such that $ESD_j(\sigma) < 1$ (if $ESD_j(\sigma) = 1$, it has reached its upper bound and the result follows trivially). If $v = (x, y)$ is not critical for j then, for all $i \in 1, 2, 3$, the longest paths form s to e_j in $G^i(\sigma)$ remain as paths in $G^i(\sigma_{(v)})$ too, so $\forall i\ c^i_j(\sigma_{(v)}) \geq c^i_j(\sigma)$. Therefore, $E[c_j(\sigma_{(v)})] \geq E[c_j(\sigma)]$ and, since $\mu_{\widetilde{d}}$ is a decreasing function, $ESD_j(\sigma_{(v)}) \leq ESD_j(\sigma)$. □

This suggests the following definition:

Definition 2. (Neighbourhood $\mathcal{N}_{ESD}$) Let σ a feasible operation processing order. The *neighbourhood structure* obtained from σ is given by $\mathcal{N}_{ESD}(\sigma) = \{\sigma_{(v)} : v \in R(\sigma)$ is critical for some $1 \leq j \leq n$ with $ESD_j(\sigma) < 1\}$.

3.2 Some Desirable Properties

The new neighbourhood $\mathcal{N}_{ESD}$ has two highly desirable properties: feasibility and connectivity.

Theorem 3. *Let $\sigma \in \Sigma$ be a feasible task processing order; the reversal of a critical arc $v = (x, y) \in R(\sigma)$ produces a feasible processing order, i.e., $\sigma_{(v)} \in \Sigma$. In consequence, $\mathcal{N}_{ESD}(\sigma) \subset \Sigma$.*

Proof. If $v = (x, y)$ is critical, by Proposition 1, $\exists i, r^i_y = r^i_x + p^i_x$. Suppose by contradiction that $G(\sigma_{(v)})$ has a cycle. Since $G(\sigma)$ has no cycles and the only change in $\sigma_{(v)}$ w.r.t. σ has been the processing order between x and y, having a cycle in $G(\sigma_{(v)})$ means that there must exist an alternative path from x to y in $G(\sigma)$. Given the definition of the head $\widehat{r}_y$ in (4), $\forall i$ we have $r^i_y \geq r^i_{PJ_y} + p^i_{PJ_y}$. This, together with the existence of an alternative path from x to y means that

$$\forall i, r^i_y \geq r^i_{SJ_x} + p^i_{SJ_x} + p^i_{PJ_y} \geq r^i_x + p^i_x + p^i_{SJ_x} + p^i_{PJ_y}$$

But being critical, there is at least one component k where $r^k_y = r^k_x + p^k_x$, and for this k we have $r^k_x + p^k_x \geq r^k_x + p^k_x + p^k_{SJ_x} + p^k_{PJ_y}$, which is absurd, because all task durations are strictly positive. □

Feasibility means that the local search is automatically limited to the subspace of feasible task orders, thus avoiding feasibility checks and repairing strategies for neighbours. This increases the efficiency of the local search procedure and avoids the loss of feasible solutions that is usually encountered with feasibility checking procedures (cf. [Dell' Amico and Trubian, 1993]).

To prove connectivity, we require the following preliminary result.

Lemma 4. *Given $\sigma \in \Sigma$ a feasible task processing order, $G(\sigma) = (V, A \cup R(\sigma))$ its solution graph and σ_0 an optimal processing order, let*

$$V_{ESD}(\sigma, \sigma_0) = \{v = (x, y) \in R(\sigma) : v \text{ is critical for some } j \text{ with } ESD_j < 1, (y, x) \in \overline{R(\sigma_0)}\} \tag{8}$$

where $\overline{R(\sigma_0)}$ denotes the transitive closure of $R(\sigma_0)$; i.e., $V_{ESD}(\sigma, \sigma_0)$ is the set of critical arcs (x, y) in $\mathcal{N}_{ESD}(\sigma)$ such that there exists a path from y to x in $R(\sigma_0)$. If $V_{ESD}(\sigma, \sigma_0) = \emptyset$ then σ is optimal w.r.t. ESD_{avg}.

Sketch of proof We first prove that, if σ is not optimal for ESD_{avg}, there exists at least a critical arc in $R(\sigma)$ for some j such that $ESD_j < 1$. This is done by contradiction, supposing that either there are no critical arcs for any job in $R(\sigma)$ or every arc $v \in R(\sigma)$ that is critical for some j is such that $ESD_j(\sigma) = 1$ and showing that in both cases σ must be optimal. Secondly we prove also by contradiction that if σ is not optimal for ESD_{avg}, there exists at least a critical arc for some j with $ESD_j(\sigma) < 1$ such that $(y, x) \in \overline{R(\sigma_0)}$. □

Theorem 5. *$\mathcal{N}_{ESD}$ verifies the connectivity property: for every non-optimal task processing order σ we may build a finite sequence of transitions of $\mathcal{N}_{ESD}$ leading from σ to a globally optimal processing order σ_0.*

Proof. Let σ_0 be any optimal processing order and let the sequence $\{\lambda_k\}_{k \geq 0}$ of processing orders be given by $\lambda_0 = \sigma$ and $\lambda_{k+1} = \lambda_{k(v)}$ with $v \in V_{ESD}(\lambda_k, \sigma_0)$. The reversal of an arc in $V_{ESD}(\lambda_k, \sigma_0)$ is a move from $\mathcal{N}_{ESD}$ so, by Theorem 3, $\forall k\ \lambda_k \in \Sigma$ (i.e., all task processing orders in the sequence are feasible solutions). Let us prove that the above sequence is finite. For any processing order $\sigma \in \Sigma$, we define

$$M(\sigma, \sigma_0) = \{v = (x, y) \in R(\sigma) : (y, x) \in \overline{R(\sigma_0)}\}$$
$$\overline{M(\sigma, \sigma_0)} = \{v = (x, y) \in \overline{R(\sigma)} : (y, x) \in \overline{R(\sigma_0)}\}$$

Clearly, $V_{ESD}(\lambda_k, \sigma_0) \subset M(\sigma, \sigma_0) \subset \overline{M(\sigma, \sigma_0)}$. Let $\|M(\sigma, \sigma_0)\|$ and $\|\overline{M(\sigma, \sigma_0)}\|$ denote their cardinalities. By definition of λ_k, if $\|\overline{M(\lambda_k, \sigma_0)}\| > 0$ then $\|\overline{M(\lambda_{k+1}, \sigma_0)}\| =$

$\|\overline{M(\lambda_k, \sigma_0)}\| - 1$. Therefore, for $k^\star = \|\overline{M(\sigma, \sigma_0)}\|$, we have $\|\overline{M(\lambda_{k^\star}, \sigma_0)}\| = 0$. Since $V_{ESD}(\sigma, \sigma_0) \subseteq M(\sigma, \sigma_0) \subseteq \overline{M(\sigma, \sigma_0)}$, this implies that $V_{ESD}(\lambda_{k^\star}, \sigma_0) = \emptyset$ so, by Lemma 4, $\lambda_{k^\star}$ is optimal for ESD_{avg}. □

Connectivity is important because it ensures the non-existence of starting points from which local search cannot reach a global optimum. It also ensures asymptotic convergence in probability to a globally optimal order. Additionally, it opens the possibility of designing an exact branch and bound method for fuzzy job shop.

3.3 ESD Estimation Procedure

Evaluating all neighbours in $\mathcal{N}_{ESD}$ has a high computational cost. However this can be reduced if an upper bound of ESD_{avg} can be easily evaluated, allowing to discard neighbours that are for sure not improving.

For a processing order σ and tasks x and y, let $P_{\sigma,j}(x \vee y)$ denote the set of all paths to end node e_j in $G(\sigma)$ containing x or y, $P_{\sigma,j}(x \wedge y)$ denote the set of all paths to e_j in $G(\sigma)$ containing both x and y and let $P_{\sigma,j}(\neg x)$ denote de set of all paths to e_j in $G(\sigma)$ not containing x. Also, for a given set P_j of paths to node e_j, let $D[P_j]$ denote the TFN such that $D^i[P_j]$ is the length of the longest path in P_j in the parallel graph G^i if $P_j \neq \emptyset$ and $-\infty$ otherwise, $i = 1, 2, 3$.

Proposition 6. *Let σ be a task processing order and let $\sigma_{(v)}$ be the order that results after reversing $v = (x, y)$ an arc in $G(\sigma)$. Then, the completion time for job j in the new solution is given by:*

$$c_j(\sigma_{(v)}) = \max\{D[P_{\sigma_{(v)},j}(x \vee y)], D[P_{\sigma,j}(\neg x)]\} \quad (9)$$

Proof. For every $i = 1, 2, 3$ it holds that $c_j^i(\sigma_{(v)}) = \max\{D^i[P_{\sigma_{(v)},j}(x \vee y)], D^i[P_{\sigma_{(v)},j}(\neg x \wedge \neg y)]\}$. Since the only arcs that change from $G(\sigma)$ to $G(\sigma_{(v)})$ are $(PM_x(\sigma), x)$, (x, y), $(y, SM_y(\sigma))$, those paths to e_j not containing x nor y remain unchanged, i.e., $c_j^i(\sigma_{(v)}) = \max\{D^i[P_{\sigma_{(v)},j}(x \vee y)], D^i[P_{\sigma,j}(\neg x \wedge \neg y)]\}$. Now, for every path to e_j in $G(\sigma)$ containing y but not containing x, either it starts in y or it contains (PJ_y, y) so in any case the subpath to y is identical in both $G(\sigma)$ and $G(\sigma_{(v)})$. If the path to e_j does not contain SM_y, it is still a path to e_j in $G(\sigma_{(v)})$ and if it does contain the arc (y, SM_y), then substituting (x, y) by $(y, x), (x, SM_y)$ we obtain a longer path to e_j in $G(\sigma_{(v)})$. Therefore, $D^i[P_{\sigma,j}(\neg x \wedge y)] \leq D^i[P_{\sigma_{(v)},j}(x \vee y)]$ and we may rewrite the above expression as $c_j^i(\sigma_{(v)}) = \max\{D^i[P_{\sigma_{(v)},j}(x \vee y)], D^i[P_{\sigma,j}(\neg x)]\}$. □

Corollary 7. *Let $\widehat{r}'_x$ and $\widehat{r}'_y$ denote respectively the heads of x and y and let $\widehat{q}'_{x,j}$ and $\widehat{q}'_{y,j}$ denote the tails of x and y relative to j after reversing arc $v = (x, y)$ in $G(\sigma)$. Then,*

$$LB(\widehat{c}_j) = \begin{cases} \max\{\widehat{r}'_x + \widehat{p}_x + \widehat{q}'_{x,j}, \widehat{r}'_y + \widehat{p}_y + \widehat{q}'_{y,j}\} \\ \qquad\qquad \textit{if } P_{\sigma_{(v)},j}(x \vee y) \neq \emptyset \\ \widehat{r}_{e_j}(\sigma) \quad \textit{otherwise} \end{cases} \quad (10)$$

provides a lower bound *for the completion time c_j of job j obtained after reversing (x, y) in the sense that for $i = 1, 2, 3$ $LB(\widehat{c}_j)^i \leq c_j^i$ and hence $E[LB(\widehat{c}_j)] \leq E[\widehat{c}_j]$). Consequently,*

$$UB_j = ESD(LB(\widehat{c}_j), \widetilde{d}_j) \quad (11)$$

provides an upper bound *for the expected satisfaction degree for job j and $UB = \frac{1}{n}\sum_{j=1,\ldots,n} UB_j$ provides an* upper bound *for the average expected satisfaction degree ESD_{avg}.*

This upper bound of ESD_{avg}, inspired by the work of [Taillard, 1994] for the classical job shop with makespan minimisation, can be calculated in time $O(n)$.

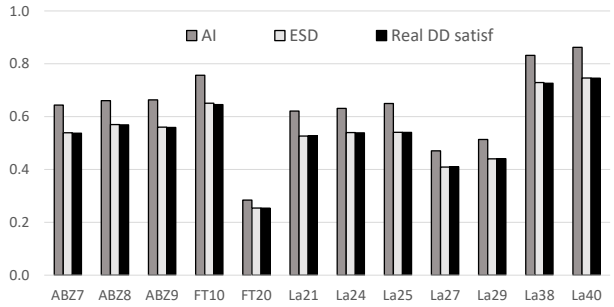

Figure 2: ESD_{avg}, AI_{avg} and real due-date satisfaction in average under Scenario I

4 Experimental Results

The performance of local search algorithms depends heavily on the initial solution provided and this is why they are usually run with multi-starts. When the starting points are provided and evolved in a genetic algorithm, we obtain a *memetic algorithm* (MA). MAs combine the intensification provided by the local search with the diversification provided by the population-based algorithm and have proved to be very powerful in solving scheduling problems [Palacios *et al.*, 2015; 2016]. Thus, to analyse the new objective function and the proposed neigbourhood, we shall use the MA framework proposed in [Palacios *et al.*, 2017b] for the fuzzy job shop with due date satisfaction, but replacing the original objective function (based on aggreement index) and neighbourhood structure used in that work with ESD_{avg} and the new neighbourhood $\mathcal{N}_{ESD}$ proposed here. We also adopt the same set of benchmark instances, fuzzy versions of well-known problems: 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). Experiments have been run on a PC with Xeon processor at 2,2Ghz and 24 Gb RAM with Linux (SL 6.0.1), using a C++ implementation.

A first set of experiments is run to analyse if the proposed objective function ESD_{avg} is more accurate an estimate of the actual average due-date satisfaction degree after execution than the average agreement index AI_{avg} from the literature. To this end, for each problem instance, we consider the 30 fuzzy schedules obtained with 30 runs of the MA from [Palacios *et al.*, 2017b] maximising AI_{avg}. For each schedule we compute the value ESD_{avg}. Then, in absence of real executions of these schedules (due to the instances being synthetic), we obtain 1000 MonteCarlo simulations of task durations (i.e., 1000 possible deterministic realisations of each

Instance	No Estimations ESD_{avg}	Time	With Estimations ESD_{avg}	Time
ABZ7	0.553 (0.013)	117.4	0.552 (0.021)	87.8
ABZ8	0.578 (0.015)	135.2	0.581 (0.015)	94.7
ABZ9	0.563 (0.026)	138.0	0.571 (0.025)	106.0
FT10	0.637 (0.008)	4.6	0.636 (0.009)	3.5
FT20	0.246 (0.013)	5.8	0.247 (0.013)	4.1
La21	0.543 (0.017)	22.9	0.545 (0.019)	15.9
La24	0.559 (0.023)	22.7	0.554 (0.017)	16.0
La25	0.563 (0.011)	18.3	0.558 (0.016)	12.9
La27	0.391 (0.018)	47.1	0.386 (0.025)	28.2
La29	0.435 (0.031)	49.5	0.438 (0.024)	27.0
La38	0.730 (0.026)	51.4	0.738 (0.023)	31.4
La40	0.754 (0.027)	46.6	0.762 (0.019)	34.1

Table 1: ESD_{avg} values obtained by MA with and without the estimation filtering, and runtime (in seconds).

Instance	LS	GA	MA
ABZ7	0.325 (0.016)	0.511 (0.023)	0.552 (0.021)
ABZ8	0.339 (0.017)	0.513 (0.014)	0.581 (0.015)
ABZ9	0.326 (0.016)	0.511 (0.030)	0.571 (0.025)
FT10	0.466 (0.030)	0.621 (0.007)	0.636 (0.009)
FT20	0.154 (0.011)	0.232 (0.015)	0.247 (0.013)
La21	0.398 (0.018)	0.485 (0.015)	0.545 (0.019)
La24	0.388 (0.013)	0.510 (0.037)	0.554 (0.017)
La25	0.386 (0.013)	0.506 (0.022)	0.558 (0.016)
La27	0.222 (0.013)	0.316 (0.020)	0.386 (0.025)
La29	0.213 (0.014)	0.356 (0.035)	0.438 (0.024)
La38	0.540 (0.017)	0.672 (0.028)	0.738 (0.023)
La40	0.544 (0.016)	0.673 (0.032)	0.762 (0.019)

Table 2: ESD_{avg} values obtained by MA and its the main components: GA and LS

fuzzy instance) using a probability distribution that is consistent with the possibility distribution given by each fuzzy processing time. In fact, two different probability distributions provide two different Scenarios (I and II) as proposed in [Palacios *et al.*, 2017a]. The 30 task processing orders provided by the 30 fuzzy schedules are evaluated on each possible realisation, so the actual average due-date satisfaction degree obtained after execution can be measured under both scenarios. As expected the AI_{avg} value always overestimates the real due-date satisfaction (17.1% higher under Scenario I and 16.9% under Scenario II in average). On the other hand, ESD_{avg} provides a more accurate estimate (0.4% higher under Scenario I and 0.1% under Scenario II in average). Results under Scenario I can be seen in Figure 2.

Now we move on to evaluating the MA maximising ESD_{avg} and using the new neighbourhood $\mathcal{N}_{ESD}$. A preliminary parametric analysis (not reported here due to lack of space) is carried out to find the best setup for the algorithm, trying different genetic operators as well as population sizes, stopping criterion and percentage of the population to which local search is applied. The obtained parameter values are population size 100, JOX crossover [Ono *et al.*, 1996], swap mutation, crossover and mutation probabilities 0.9 and 0.05 respectively, and $maxIter$=25 as stopping criterion, with the local search being applied to all individuals in the population at each iteration.

To evaluate the impact of the neighbour estimation procedure from Section 3.3 in the efficiency of the search, Table 1 reports the average and standard deviation (between brackets) values for ESD_{avg} and CPU time (in seconds) obtained across 30 runs of MA on each instance with and without using UB to filter non-improving neighbours and avoid unnecessary evaluations. Using UB clearly translates in a reduction of running time: MA is nearly 43% slower in average when UB is not used as filtering mechanism, with an increase in runtime up to 83% for La29. However, both approaches obtain similar quality regarding objective function, with no significant statistical difference. This shows the great potential of the ESD_{avg} estimation procedure, which yields the same performance in considerably shorter time.

A final set of experiments evaluates the synergy between the genetic algorithm (GA) and the local search procedure (LS). To this end, the GA is run independently for as long as the MA takes to converge, and LS is run as a multi-start local search with as many restarts as the average number of evaluations performed by MA on each instance. The results in Table 2 show that the GA by itself does not outperform MA in any of the instances, being 10.3% worse in average. A statistical test confirms significant differences between both methods. A similar behaviour is encountered with the multi-start LS. In the absence of good starting points provided by the GA, results get worse even with slightly longer runtimes. A nice synergy effect can be appreciated combining both strategies, with MA obtaining much better results in the same runtime as GA and with the same number of evaluations as LS. That is, MA benefits from the exploration of GA and also from the intensification of LS.

5 Conclusions

We have tackled the job shop scheduling problem with uncertain durations and flexible due dates modelled as fuzzy numbers. We have proposed a new measure of due-date satisfaction and we have defined a new neighbourhood structure purpose-built for this new objective. It has been shown that the neighbourhood presents a good theoretical behaviour and we have provided an estimation mechanism that allows to discard non-improving neighbours. Experimental results have been presented for the new neighbourhood embedded in an existing MA framework from the literature. These results avail the proposal of the new objective function for due-date satisfaction under uncertainty as well as the good behaviour of the resulting MA, thanks to the synergy between the genetic algorithm and the local search and the gain in efficiency provided by the estimation procedure.

Acknowledgements

Finantial support acknowledgement omitted for blind review.

References

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

[Bierwirth and Kuhpfahl, 2017] C. Bierwirth and J. Kuhpfahl. Extended GRASP for the job shop scheduling problem with total weighted tardiness objective. *European Journal of Operational Research*, 261:835–848, 2017.

[Błażewicz *et al.*, 2000] J. Błażewicz, E. Pesch, and M. Sterna. The disjunctive graph machine representation of the job shop scheduling problem. *European Journal of Operational Research*, 127:317–331, 2000.

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

[Dubois *et al.*, 2003] 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.

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

[González *et al.*, 2013] Miguel A. González, C.R. Vela, I. González-Rodríguez, and R. Varela. Lateness minimization with tabu search for job shop scheduling problem with sequence dependent setup times. *Journal of Intelligent Manufacturing*, 24(4):741–754, 2013.

[González *et al.*, 2015] M. A. González, C. R. Vela, and R. Varela. Scatter search with path relinking for the flexible job shop scheduling problem. *European Journal of Operational Research*, 245:35–45, 2015.

[González Rodríguez *et al.*, 2008a] 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.

[González Rodríguez *et al.*, 2008b] 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.

[Graham *et al.*, 1979] R.L. Graham, E.L. Lawler, J.K. Lenstra, and A.H.G. Rinnooy Kan. Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of Discrete Mathematics*, 4:287–326, 1979.

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

[Kuhpfahl and Bierwirth, 2016] J. Kuhpfahl and C. Bierwirth. A study on local search neighbourhoods for the job shop scheduling problem with total weighted tardiness objective. *Computers & Operations Research*, 261:44–57, 2016.

[Ono *et al.*, 1996] I. Ono, M. Yamamura, and S. Kobayashi. A genetic algorithm for job-shop scheduling problems using job-based order crossover. In *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on*, pages 547–552. IEEE, 1996.

[Palacios *et al.*, 2015] 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.

[Palacios *et al.*, 2016] J. J.Palacios, J. Puente, C. R. Vela, and I. González-Rodríguez. Benchmarks for fuzzy job shop problems. *Information Sciences*, 329:736–752, 2016.

[Palacios *et al.*, 2017a] J. J. Palacios, I. González-Rodríguez, C. R. Vela, and J. Puente. Robust multi-objective optimisation for fuzzy job shop problems. *Applied Soft Computing*, 56:604–616, 2017.

[Palacios *et al.*, 2017b] J. J. Palacios, C. R. Vela, I. González-Rodríguez, and J. Puente. A memetic algorithm for due-date satisfaction in fuzzy job shop scheduling. In *IWINAC2017 International Work-Conference on the Interplay Between Natural and Artificial Computation. LNCS 10337*, pages 135–145. Springer, June 2017.

[Peng *et al.*, 2015] B. Peng, Z. Lü, and T.C.E. Cheng. A tabu search/path relinking algorithm to solve the job shop scheduling problem. *Computers & Operations Research*, 53:154–164, 2015.

[Pinedo, 2016] M. L. Pinedo. *Scheduling. Theory, Algorithms, and Systems.* Springer, Fifth edition, 2016.

[Puente *et al.*, 2010] 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.

[Taillard, 1994] E. D. Taillard. Parallel taboo search techniques for the job shop scheduling problem. *ORSA Journal on Computing*, 6(2):108–117, 1994.

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