Fuzzy Rough Total Weighted Tardiness Flow Shop Scheduling Model with Hurwicz Criterion

Achraf Touil, Abdelwahed Echchatbi
Laboratory of Engineering, of Industrial Management and Innovation,
Faculty of Sciences and Technology, Hassan 1st University, PO Box 577, Settat, Morocco
Email: (ac.touil;abdelwahed.echchatbi)@uhp.ac.ma

***Abstract*—In this paper, we investigate the total weighted tardiness permutation flow shop with fuzzy rough processing time and a distinct due date. A fuzzy rough model based on the Hurwicz criterion is established. By varying the value of θ, it can balance the optimistic and pessimistic levels of the decision-makers. Moreover, to solve this model, fuzzy rough simulation and a eagle strategy combined with the sine-cosine algorithm are integrated to produce a hybrid intelligent algorithm. Finally, numerical results are reported to demonstrate the efficiency and applicability of the proposed model.**

***Keywords*—Permutation Flow Shop (PFS), Total Weighted Tardiness (TWT), Fuzzy Rough Theory (FRT), Eagle Strategy(ES), Sine-Cosine Algorithm (SCA)**

I. Introduction

The permutation flow shop problem (PFSP) is considered to be the simplified version of the flow shop problem. This is where the sequence of jobs to the process will be the same on each machine. The PFSP has proved to be non-deterministic-polynomial-time(NP)-hard [1]. Both the motor and the semiconductor industries use this problem as apart of their production process [2]. It has also attracted the attention of many researchers and practitioners due to its theoretical complexity, as well as practical application.

Several scheduling criteria are studied in the literature. Among them, the total tardiness which coincides with customer satisfaction. Delays are likely to lead to an increase in costs. These costs would include penalty clauses, loss of customers, and a bad-standing reputation with new customers. To avoid these problems, the objective seeks to reduce the completion times of jobs to properly meet their due dates. According to the notation in [3] this type of problem is denoted as: $Fm|prum|\sum_{j\in J}\alpha_i.T_i$.

In the original total weighted tardiness scheduling problem, one will likely assume that the job's processing times are deterministic. However, this assumption is inappropriate in many practical scenarios. For example: in real production systems there are many imprecise or uncertain factors involved in the scheduling problems: including demand, processing time, and due date. This uncertainty in real production systems is usually manipulated through three approaches, according to the literature [4]. The first is the probability theory. The second is the fuzzy set theory, proposed by Zadeh [5]. The third is the rough set theory, initialized by Pawlak [6]. Regarding the PFSP under uncertainty, [7] demonstrated how to integrate the concept of the credibility measurement for the mentioned problem. As a result, three types of fuzzy models are presented, namely, the expected value model (EVM), the chance-constrained model (CCPM), and the dependent chance model (DCPM). A hybrid intelligent algorithm is then designed to solve the proposed models. The processing times have been considered as rough variables by [8]. The objective is to minimize the makespan while considering the same decision models as [7]. To solve this problem, they used a hybrid intelligent algorithm that combined the rough simulation and genetic algorithm.

There is only one source of uncertainty that is considered in these mentioned studies. However, scheduling problems may be subject to hybrid uncertainties such as an encounter with fuzziness and roughness simultaneously. For example, many accept that the job's processing times are triangular fuzzy number variables (a, b, c) from the viewpoint of the fuzzy theory. Though, the values of a, b, and c may emerge with incomplete or uncertain information. In a sense, they are rough characteristics. Thus decision-makers have to face the "fuzzy number with rough parameters". In this case, particularly, the job's processing times should be more appropriately represented as the fuzzy rough variable [4].

In this paper, a new model based on both the fuzzy rough theory and the Hurwicz criterion is proposed to deal with uncertainty in the total weighted tardiness flow-shop scheduling. A remarkable advantage of such criteria is to attempt to strike a balance between the many extremes posed by the optimistic and pessimistic criteria. In such a non-deterministic environment, traditional methods cannot provide a solution. To overcome this issue, researchers give a central focus for the well known hybrid intelligent algorithm (HIA). In HIA, a type of simulation (i.e. fuzzy, or rough simulation) is used for calculating the non-deterministic measure (i.e. credibility, or trust measures) while another method of metaheuristic such as genetic algorithm is adopted to find the quasi-optimal solution. As a second contribution, a new hybrid intelligent algorithm is proposed. It starts with a simulation technique that is based on a fuzzy rough simulation of the objective function. Then, the global search is performed with an eagle strategy method [9] according to Levy Flights while the local search is completed through Sine-Cosine Algorithm (SCA) [10]. Besides, local search methods are adopted from a performance improvement standpoint.

The rest of this paper is organized as follows: Section 2 provides some preliminaries about fuzzy rough theory. Section 3, the fuzzy rough total weighted tardiness flow shop problem based on the Hurwicz criterion is presented. Section 4, presents the proposed hybrid intelligent algorithm. Section 5 contains the computational results to illustrate the performance of the proposed model and algorithm. Finally, the conclusion is summarized in Section 6

II. Fuzzy Rough Theory Preliminaries

A fuzzy rough variable was initialized by Liu [4] as a rough variable defined on the universal set of fuzzy variables, or a rough variable taking "fuzzy variable" values. In this section, we will recall some basic concepts that will help establish the fuzzy rough model with Hurwicz criterion for flow shop problem with total weighted tardiness.

Definition 4. [4]. A fuzzy rough variable ξ is a mapping from a rough space $(\Lambda, \Delta, \mathcal{A}, \pi)$ to a collection of fuzzy variables.

Definition 5. [4]. Let ξ be a fuzzy rough variable, defined on the rough space $(\Lambda, \Delta, \mathcal{A}, \pi)$. Then its expected value $E[\xi]$ defined as:

$$E[\xi] = \int_0^\infty Tr\{\lambda \in \Lambda | E[\xi(\lambda)] \geq r\} \,\mathrm{d}r - \int_\infty^0 Tr\{\lambda \in \Lambda | E[\xi(\lambda)] \leq r\} \,\mathrm{d}r \quad \text{(II.1)}$$

Definition 6. [4]. Let $\xi = (\xi_1, \xi_2, ..., \xi_n)$ be an n-dimensional fuzzy rough vector defined on the rough space $(\Lambda, \Delta, \mathcal{A}, \pi)$ and $f_j : \Re^n \to \Re$ be real-valued functions,$j = 1, 2, ..., q$. Then the chance $Ch\{f_j(\xi) \leq 0, j = 1, 2, ..., q\}(\alpha)$ of the fuzzy rough event characterized by$f_j(\xi) \leq 0, j = 1, 2, ..., q$ is a function from $[0, 1] to [0, 1]$ defined as:

$$Ch\{f_j(\xi) \leq 0, j = 1, 2, ..., q\}(\alpha) = sup\left\{\beta | Tr\left\{\lambda \in \Lambda | Cr\left\{\begin{array}{c} f_j(\xi(\lambda)) \leq 0 \\ j = 1, 2, ..., q \end{array}\right\} \geq \beta\right\} \geq \alpha\right\} \quad \text{(II.2)}$$

III. Fuzzy rough permutation flow shop problem

The Permutation Flow Shop Problem (PFSP) is a typical combinatorial optimization problem, which determines the processing sequence of jobs over machines to minimizes the total makespan, the total flow time, and can satisfy other objectives (such as total weighted tardiness). Assumptions and notation for are described as follows:

- The processing sequence of all jobs on each machine is the same, but has not been known;
- All jobs are available at time zero and the processing time is assumed to be a fuzzy rough variable. Moreover, the setup time is included in the processing time.
- Each machine can process at most one job at any time;
- Each job can be processed on at most one machine at any time;
- Each job must complete processing without preemptions.

A. Symbols and formulation

n: Denotes the number of jobs;
m: Denotes the number of machines;
$\xi_{i,j}$: Denotes the fuzzy rough processing time of the job i on machine j;
$\boldsymbol{\xi} = (\xi_{1,1}, \xi_{1,2}, ..., \xi_{n,m})$: Denotes the fuzzy rough vector;
d_i: Denotes the due date of the job;
v_i: Denotes the tardiness cost per unit time of the job finished after the due date;
$\boldsymbol{x} = (x_1, x_2, ..., x_n)$: The integer decision vector that represents the schedule

$$x_1 \to x_2 \to ... \to x_n \quad \text{(III.1)}$$

For simplicity, we use $\boldsymbol{\xi} = (\xi_{1,1}, \xi_{1,2}, ..., \xi_{n,m})$ to denote the fuzzy rough vector and note that the full schedule is represented by the integer decision vector $\boldsymbol{x} = (x_1, x_2, ..., x_n)$, which is representing a permutation of n jobs with $1 \leq x_i \leq n$ and $x_i \neq x_k$ for all $i \neq k, \;\; i, k = 1, 2, ..., n$.
Let $C(x_i, j, \boldsymbol{\xi})$ be the completion times of job x_i on the machine j. Then the completion times can be calculated by the following equations:

$$\begin{cases} C(x_1, 1, \boldsymbol{\xi}) = \xi_{x_1,1} \\ C(x_i, 1, \boldsymbol{\xi}) = C(x_{i-1}, 1, \boldsymbol{\xi}) + \xi_{x_i,1}, \;\; i = 2, ..., n \\ C(x_1, j, \boldsymbol{\xi}) = C(x_1, j-1, \boldsymbol{\xi}) + \xi_{x_1,j}, \;\; j = 2, ..., m \\ C(x_i, j, \boldsymbol{\xi}) = max\{C(x_{i-1}, j, \boldsymbol{\xi}), C(x_i, j-1, \boldsymbol{\xi})\} + \xi_{x_i,j} \\ \qquad i = 2, ..., n, j = 2, ..., m \end{cases} \quad \text{(III.2)}$$

Let $TWT(\boldsymbol{x}, \boldsymbol{\xi})$ denote the fuzzy rough total weighted tardiness of the schedule which is calculated as follows:

$$TWT(\boldsymbol{x}, \boldsymbol{\xi}) = \sum_{i=1}^{n} v_i * (C(x_i, j, \boldsymbol{\xi}) - d_i))^+ \quad \text{(III.3)}$$

where $v_i * (C(x_i, j, \boldsymbol{\xi}) - d_i)^+$ denotes the fuzzy rough tardiness cost when the job is completed after its due date d_i.

B. Fuzzy rough model with Hurwicz criterion

In an uncertain environment, the most well-known criteria are optimistic value criteria and pessimistic value criteria. By using optimistic criteria, a decision-maker can handle the maximum payoffs of alternatives and can choose a suitable alternative with the lowest cost (which would be the greatest outcome). By using pessimistic criteria the decision-maker only handles the minimum payoffs of alternatives, and can choose the alternative with the least bad outcome. This criterion is suggestive of a conversation decision-maker who, in a situation of an unfavorable outcome known as loss, ensures that we are aware of the maximum loss. Several other criteria are proposed to overcome the extreme cases of these two criteria. The Hurwicz criterion, proposed by [11], is one of the most well-known criteria. It attempts to balance extreme criteria by ensuring that both the optimistic and pessimistic criteria are averaged using the weights θ and $1-\theta$. It then associates to each action x the following index:

$$\theta \max(x) + (1-\theta)\min(x). \tag{III.4}$$

Then, we get the fuzzy rough environment under Hurwicz criterion as follows:

$$\theta \min_{f_{opt}} f_{opt}(\alpha,\beta) + (1-\theta)\max_{f_{pes}} f_{pes}(\alpha,\beta) \tag{III.5}$$

where $f_{opt}(\alpha,\beta)$ and $f_{pes}(\alpha,\beta)$ are the (α,β)-optimistic and (α,β)-pessimistic values defined as follows:

$$f_{opt}(\alpha,\beta) = \min_{f}\{f|Ch\{f(x,\xi)\le f\}(\alpha)\ge\beta\} \tag{III.6}$$

$$f_{pes}(\alpha,\beta) = \max_{f}\{f|Ch\{f(x,\xi)\ge f\}(\alpha)\ge\beta\} \tag{III.7}$$

The parameter $\alpha,\beta\in(0,1]$ reflects the level of satisfying the event $Ch\{f(x,\xi)\le f\}$ or $Ch\{f(x,\xi)\ge f\}$. This means that the cost function will be below the (α,β)-optimistic value $f_{opt}(\alpha,\beta)$ with trust (α,β), and will reach upwards of (α,β)-pessimistic value $f_{pess}(\alpha,\beta)$ with trust (α,β) [4]. Therefore, by changing the value of θ, the Hurwicz criterion degenerate various criteria (e.g.,$\theta = 1$ led an optimistic criterion; $\theta = 0$ degenerate a pessimistic criterion).

Based on the above assumptions we can present the (α,β)-Cost model with Hurwicz criterion for TWT-PFSP as follow:

$$\begin{cases} \min \quad \theta \min\limits_{f_{opt}} f_{opt} + (1-\theta)\max\limits_{f_{pes}} f_{pes} \\ \text{Subject to:} \\ Ch\{TWT(x,\xi)\le f_{opt}\}(\alpha)\ge\beta \\ Ch\{TWT(x,\xi)\ge f_{pes}\}(\alpha)\ge\beta \\ 1\le x_i\le n, i=1,2,...,n \\ x_i\ne x_k, i\ne k \;\; i,k=1,2,...,n \\ x_i\in Z, \;\; i=1,2,...,n \end{cases} \tag{III.8}$$

Remark 1: According to Definition 6, if the fuzzy rough vector ξ degenerates to a rough vector, the (α,β)-Cost minimization model under the Hurwicz criterion becomes a simple the α-cost minimization model under the Hurwicz criterion with rough processing times as follow :

$$\begin{cases} \min \quad \theta \min\limits_{f_{opt}} f_{opt} + (1-\theta)\max\limits_{f_{pes}} f_{pes} \\ \text{Subject to:} \\ Tr\{TWT(x,\xi)\le f_{opt}\}\ge\alpha \\ Tr\{TWT(x,\xi)\ge f_{pes}\}\ge\alpha \\ 1\le x_i\le n, i=1,2,...,n \\ x_i\ne x_k, i\ne k \;\; i,k=1,2,...,n \\ x_i\in Z, \;\; i=1,2,...,n \end{cases} \tag{III.9}$$

Remark 2: According to the Definition 6, if the fuzzy rough vector ξ degenerates to a fuzzy vector, the (α,β)-Cost minimization model under the Hurwicz criterion becomes a simple the β-cost minimization model under the Hurwicz criterion with fuzzy processing times as follow :

$$\begin{cases} \min \quad \theta \min\limits_{f_{opt}} f_{opt} + (1-\theta)\max\limits_{f_{pes}} f_{pes} \\ \text{Subject to:} \\ Cr\{TWT(x,\xi)\le f_{opt}\}\ge\beta \\ Cr\{TWT(x,\xi)\ge f_{pes}\}\ge\beta \\ 1\le x_i\le n, i=1,2,...,n \\ x_i\ne x_k, i\ne k \;\; i,k=1,2,...,n \\ x_i\in Z, \;\; i=1,2,...,n \end{cases} \tag{III.10}$$

Remark 3: By setting the value of $\theta = 1$, the (α,β)-Cost model with Hurwicz criterion led to an optimistic model as follow:

$$\begin{cases} \min \quad \min\limits_{f_{opt}} f_{opt} \\ \text{Subject to:} \\ Ch\{TWT(x,\xi)\le f_{opt}\}(\alpha)\ge\beta \\ 1\le x_i\le n, i=1,2,...,n \\ x_i\ne x_k, i\ne k \;\; i,k=1,2,...,n \\ x_i\in Z, \;\; i=1,2,...,n \end{cases} \tag{III.11}$$

IV. Hybrid Intelligent Algorithm

In this section, we will design a hybrid intelligent algorithm to solve the (α,β)-cost model under the Hurwicz criterion, where fuzzy rough simulation and eagle strategy with the sine-cosine algorithm based on local search method are used.

A. Fuzzy Rough Simulation

Since total weighted tardiness $TWT(x,\xi)$ is a fuzzy rough variable, it will be difficult to compute the optimistic and pessimistic values to derive the Hurwicz criterion value by using analytical methods. As a kind of Monte Carlo methods, fuzzy rough simulation [12] provides an effective approximation. The main steps of fuzzy rough simulation are given according to [12]. Firstly, we give the simulation of the fuzzy rough function:

$$U_1 : x \to \min\left\{\overline{f}|Ch\left\{f(\boldsymbol{x},\boldsymbol{\xi})\le\overline{f}\right\}(\alpha)\ge\beta\right\} \tag{IV.1}$$

Generate $\underline{\lambda}_1,\underline{\lambda}_2,...,\underline{\lambda}_N$ from Δ and $\overline{\lambda}_1,\overline{\lambda}_2,...,\overline{\lambda}_N$ from Λ according the measure π. For any number v, let $\underline{N}(v)$ denote the number $\underline{\lambda}_k$ satisfying $Cr\{f(\boldsymbol{x},\boldsymbol{\xi}(\underline{\lambda}))\le v\}\le\beta$ for $k=1,2,...,N$, and $\overline{N}(v)$ denote the number of $\overline{\lambda}_k$ satisfying $Cr\left\{f(\boldsymbol{x},\boldsymbol{\xi}(\overline{\lambda}))\le v\right\}\ge\beta$ for $k=1,2,...,N$, with may be estimated by fuzzy simulation $Cr\{.\}$. Then find the minimal value v such that:

$$\frac{\underline{N}(v)+\overline{N}(v)}{2}\ge v \tag{IV.2}$$

This value is an estimation of $\overline{f}$. The process can be summarized in algorithm 1:

Algorithm 1 Fuzzy Rough simulation of U_1

Step1: Generate $\underline{\lambda_1}, \underline{\lambda_2}, ..., \underline{\lambda_N}$ from Δ according to the measure π .

Step2: Generate $\overline{\lambda_1}, \overline{\lambda_2}, ..., \overline{\lambda_N}$ from Λ according to the measure π.

Step3: Find the minimal value v such that (16) .

Step4: Return v .

By the similar way, we can estimate the second fuzzy rough function which:

$$U_2 : x \rightarrow \max \left\{ \overline{f} | Ch \left\{ f(\boldsymbol{x}, \boldsymbol{\xi}) \geq \overline{f} \right\} (\alpha) \geq \beta \right\} \quad \text{(IV.3)}$$

At last, we obtain the estimation of the Hurwicz criterion as follow:

$$V = \theta \times U_1 + (1 - \theta) \times U_2 \quad \text{(IV.4)}$$

B. Brief introduction to Eagle Strategy

Eagle Strategy (ES) is a two-stage optimization strategy that was presented by [13], it mimics both the behavior and hunting patterns of eagles in nature. Eagles forage using two factors: the first would be random search performed by flying freely and the second would be intensive search which is used to catch prey when sighted. In ES's two-part strategy, the first stage involves exploring the search space globally through Levy flight. The second step is employed if it finds a promising solution; this would be known as a local optimizer (such as hill-climbing and the downhill simplex method). One of the many advantages of this process is that it uses a balanced system between global search (which is generally slow) and local search (which is much quicker). It should be noted that another advantage is that this process is not an algorithm, rather a methodology or strategy.

C. Brief introduction to Sine-Cosine Algorithm

The Sine Cosine Algorithm (SCA) is a population-based stochastic search algorithm proposed by [10], it is inspired by the behavior of the mathematical functions known as sine and cosine. It is an optimization technique that is used to solve complex engineering optimization problems [14], [15]. The core mechanism of the SCA consists of two basic phases: initialization and updating solutions. The initial population of agent's solutions, represented by n dimension, is randomly generated. At iteration t, the position of agent — solution is specified by $X^{i,t} = \left[X_1^{i,t}, X_2^{i,t}, ..., X_n^{i,t}\right]$. The solutions are updated based on the sine or cosine function as the following equations:

$$X^{i,t+1} = \begin{cases} X^{i,t} + r_1 \times sin(r_2) \times |r_3 \times P^{i,t} - X^{i,t}| & ,r_4 \leq 0.5 \\ X^{i,t} + r_1 \times cos(r_2) \times |r_3 \times P^{i,t} - X^{i,t}| & ,r_4 > 0.5 \end{cases} \quad \text{(IV.5)}$$

$$r_1 = a \times (1 - \frac{g}{G_{max}}) \quad \text{(IV.6)}$$

Where a is a constant, G_{max} is the maximum number of iterations and g is the current iteration. Where $P^{i,t}$ is the destination solution, $X^{i,t}$ is the current solution, $|.|$ is used to indicate the absolute value. There are four main parameters in SCA:r_1, r_2, r_3, and r_4, these are all classified as random variables. The parameter r_1 indicates the next movement direction that could be in the space between the solution destination or outside of it. In [10] the authors alter the parameter r_1 according to equation IV.6 to balance both exploration and the exploitation. The parameter r_2 defines how far the movement is away or towards the destination. The parameter r_3 gives random weights for destination in order to stochastically emphasize ($r_3 > 1$) or deemphasize ($r_3 < 1$) the effect of desalination in defining the distance. Finally, the parameter r_4 switches in a balanced manner between both the sine and the cosine factors in Eq. IV.5.

D. The proposed Hybrid Intelligent Algorithm (HIA)

The eagle strategy and the Sine-Cosine algorithm were designed to solve continuous optimization problems; it is not possible to apply it directly to discrete problems such as TWT-PFSP. To address the concerns mentioned previously, they are incorporated with several methods mainly: representation, population initialization, and generate solutions. Besides, the fuzzy rough simulation is used the compute the fitness. All these steps will be discussed in the following subsections. Algorithm 2 describes the pseudo-code of the proposed hybrid intelligent algorithm where R is the number of rounds, and N_p is the population size.

*1) **Solution representation**:* To properly apply ES-SCA to the TWT-PFS problem would be through constructing a direct relationship between the vector solution of ES-SCA and the job sequence itself. In this paper, we will be using the Largest-Order-Value (LOV) mechanism, which was explained in the literature from [21], to map the ES-SCA solution $X_i = [x_{i,1}, x_{i,2}, ..., x_{i,n}]$ to job solution/permutation vector. The basic idea of this rule is ranked X_i by descending order to obtain a sequence $\phi_i = [\phi_{i,1}, \phi_{i,2}, ..., \phi_{i,n}]$. Then, the job permutation π_i is calculated by the following formula $\phi_{i,k} = k$ where the dimension k varies from 1 to n.

*2) **Population initialization**:* The initial population determines the performance of metaheuristics, especially the proposed strategy ES-SCA. To construct a good initial population, many researchers use heuristics specifically developed for the case of the PFS problem; For the total weighted tardiness criteria, an adapted version of the NEH heuristic, called NEHEDD, is developed to initially sort jobs through using the earliest due date (EDD) rule [16]. Recently [17] introduced several tie-breaking mechanisms to NEHEDD namely :

- First tie (FT): The tie of the first occurrence.
- Last tie (LT): The tie of the last occurrence.
- Total idle time (IT1): The tie with minimum idle time (including front delays and excluding back delays), i.e., $IT1 = \sum_{j=1}^{m}(e_{l,j} - \sum_{j=1}^{m} p_{l,j})$
- Total completion time (CT): The tie with minimum total completion time, i.e., $CT = \sum_{i=1}^{l} e_{i,m}$

Algorithm 2 The proposed algorithm for solving the (α, β)-Cost model under Hurwicz criterion

Initialize the parameters and use NEHWEDD to produce 10% agents algorithm 3, and the rest of the agents are generated randomly according to Equation.(IV.7);
while $(R > r)$ **do**
  Generate a set of agents — solutions X for global exploration using the Levy flight according to Equation (IV.8);
  Convert each agent X_i of the set X to a job permutation π_i by using the LOV rule;
  For each permutation, calculate the total weighted tardiness $(TWT(\boldsymbol{x}, \boldsymbol{\xi}))$ according to Equation (IV.4) using fuzzy rough simulation (Algorithms 1);
  Update the best solution obtained so far $(P = X^*)$
Inner loop :
  Generate randomly a set of agents around this promising solution, where feasibility must be offered;
Carry out an intensive local search via the Sine-Cosine algorithm
  Execute the local search (Algorithm 4)
  for $i = 1 \ \ to \ \ NP$ **do**
    Update the position of search agents using Equation. (IV.6)
  end for
  Update r_1 using Equation.(IV.9) and r_2, r_3 and r_4
  Convert each agent X_i of the set X to a job permutation π_i by using the LOV rule;
  For each permutation, calculate the total weighted tardiness $(TWT(\boldsymbol{x}, \boldsymbol{\xi}))$ according to Equation (IV.4) using fuzzy rough simulation (Algorithms 1);
  Update the best solution obtained so far $(P = X^*)$
  if (a better solution is found) **then**
    Update the current best
  end if
End Inner loop
  Update $r = r + 1$
end while
Report the best solution as the optimal schedule

- Total earliness time (ET): The tie with maximum total earliness time, i.e., $ET = \sum_{i=1}^{l} max\left\{d_i - e_{i,m}, 0\right\}$
- Makespan(MS): The tie with maximum makespan, i.e., . $MS = e_{i,m}$

Based on advanced tests, IT1 tie-breaking mechanism [17] lead to better results than others tie-breaking. In this paper, the NEHWEDD with IT1 is used to generate 10% of N. The pseudo-code is shown in Algorithm 3. The rest of 90% of N of solutions generated randomly according to the following equation:

$$X_{p,j} = X_j^{max} - RND() \times (X_j^{max} - X_j^{min}) \qquad \text{(IV.7)}$$

where $p = 1, 2, ..., NP$, $j = 1, 2, ..., n$, and r is a uniform random number between 0 and 1.

*3) **Generate new solutions:***

Algorithm 3 NEHWEDD heuristic

Function NEHWEDD(TB_{IT1})
α : Jobs ordered by non-decreasing $\frac{d_i}{w_i}$ where $\alpha = [\alpha_1, \alpha_i, ..., \alpha_n]$
$\pi : \{\alpha_1\}$
for $k = 2$ to n **do**
  Test job α_k in any possible position of π
  π : Permutation obtained by inserting α_k in the position of π which less total tardiness breaking ties according to IT1 tie-breaking mechanism;
end for
EndFunction

a) Exploration phase using Lévy Flights: As mentioned above, the ES is a two-stage strategy, and we can use different algorithms at different stages. In the first stage, ES uses the so-called Lévy flights, which represent a kind of non-Gaussian stochastic process whose step sizes are distributed based on a Levy stable distribution to generate new solutions. When a new solution is produced, the following Levy flight is applied:

$$X_i^{t+1} = X_i^t + \alpha \oplus Levy(\beta) \qquad \text{(IV.8)}$$

Here, α is the step size that is relevant to the scales of the problem generally chosen as $\alpha = 0.01$. The step length $Levy(\lambda)$ can be calculated by using Mantegna's algorithm [18].

b) Exploitation phase using SCA based local search: For the second stage, we can use various efficient meta-heuristic algorithms like the sine-cosine algorithm to do a strenuous local search. We know the SCA is a global search algorithm, but it can easily be tuned to do an efficient local search by limiting new solutions locally around the most promising region. As mentioned above in the parameter r_1 given in equation (1) balanced the exploration and exploitation. Since, if the ranges of sine and cosine functions are in $(1, 2]$ and $[-2, -1)$ then the SCA explores the search space. However, it exploits the search space. In this paper, we proposed a modified version of the parameter r_1 as follow:

$$r_1 = (2 - 3 \times \frac{\log(g)}{\log(G_{max})}) \qquad \text{(IV.9)}$$

In SCA, at each iteration t, agents update their positions around the global optimal location, which results in a strong local aptitude. For further improvement of computational performance, a local search-based method is applied to the destination agent. Such a combination provides excellent results however; the performance of the local search depends on the neighborhood operation utilized. For the PFS problem, there are three neighborhood operations used in the literature: Insertion, Interchange, and Swap move. Several studies [19], [20] show that Insert move is more efficient than both Interchange and Swap move. Thus in this paper, Insert is used as the neighborhood structure (which was developed by [21] for the proposed model). The algorithm 4 provides the local search-based method.

Algorithm 4 Insert-based local search

Convert individual destination $X_{destination}$ to a job permutation $\pi_{i,0}$ according to the LOV rule;
Randomly select u and v, where $u \neq v$; $\pi_i = Insert(\pi_{i,0}, u, v)$;
Set loop=1;
repeat
  Randomly select u and v, where $u \neq v$; $\pi_{i-1} = Insert(\pi_i, u, v)$;
  if $TWT(\pi_{i-1}, \boldsymbol{\xi}) < TWT(\pi_{i-1}, \boldsymbol{\xi})$ **then**
    $\pi_{i-1} = \pi_{i-1}$;
    $loop++$;
  end if
until $(loop < n \times (n-1))$
if $TWT(\pi_i, \boldsymbol{\xi}) < TWT(\pi_{i,0}, \boldsymbol{\xi})$ **then**
  $\pi_{i,0} = \pi_i$;
end if

V. Numerical Results

In this section, we illustrate the numerical results to (i) to demonstrate the application of the model, (ii) to test the robustness of the proposed algorithm. We assume that there are 10 jobs and 5 machines. The processing times are addressed by fuzzy rough variables are presented in Table 1, and Table 2 shows the corresponding due date time and tardiness penalty cost of each job. We set 1000 cycles in fuzzy rough simulation and the confidence levels $(\alpha, \beta) = (0.8, 0.9)$.

A. Analysis of robustness

The proposed algorithm (ES-SCA-LS) is compared with the standard Sine-Cosine algorithm (SCA) [10]. The robustness of both algorithms is tested with different values of two parameters including population size and maximum iteration. Table 3 reports the obtained computational results as follow :

- **Column 1 & 2**: The algorithm parameters $(N_{pop}; G_{max})$
- **Column 3**: The obtained objective value (TWT) by ES-SCA-LS and SCA
- **Column 4**: The relative error computed according to equation 23 :

$$\frac{V_{actual} - V_{optimal}}{V_{optimal}} * 100\% \qquad \text{(V.1)}$$

where $V_{optimal}$ represents the optimal objective value (i.e minimum) of all objective values obtained with different parameters and V_{actual} is the objective value obtained for a given experiment with the given parameters.

It follows from columns 4 and 6 of table 3 that the error does not exceed 14.91% and 46.00% for ES-SCA-LS and SCA respectively which implies that the designed algorithm is robust to the parameter settings when solving the problem considered in this paper.

Figures 1-3 provide the convergence curves of ES-SCA-LS for a given N_{pop} and G_{max}.

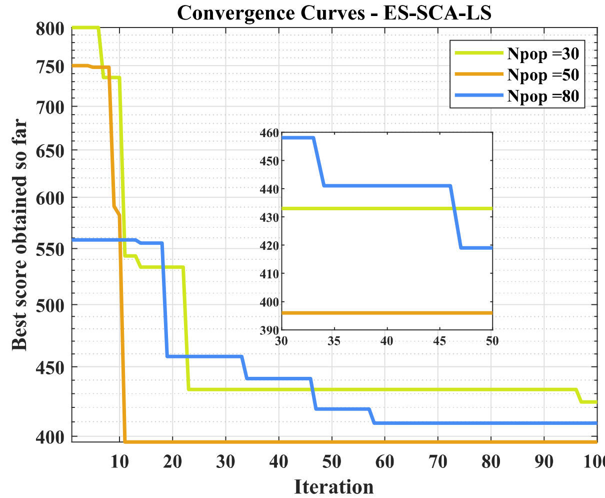

Fig. 1. Convergence Curve of $G_{max} = 100$

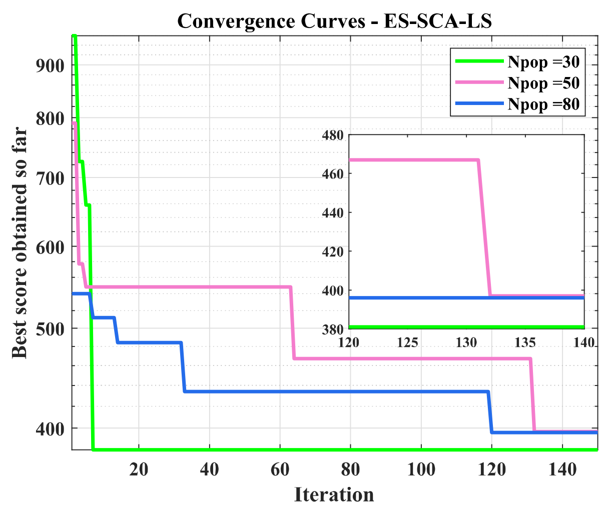

Fig. 2. Convergence Curve of $G_{max} = 150$

B. Analysis of sensitivity

The proposed model is dependent on the values of Hurwicz parameter θ, thus it is, therefore, useful to study the sensitivity of the optimal objectives concerning these parameters. We will discuss this issue using different values. Table 4 reports the obtained optimal values as follow :

- **Column 1**: The confidence levels (α, β)
- **Column 2**: The considered values of Hurwicz parameter
- **Column 3**: The pessimistic value
- **Column 4**: The Optimistic value
- **Column 5**: The value of Hurwicz Criterion

It follows from Table 4 and figure 4 that the pessimistic value decreases, while the optimistic and Hurwicz values increases as the Hurwicz parameter increases.

TABLE I
FUZZY ROUGH PROCESSING TIMES

Jobs	Machine1	Machine2	Machine3	Machine4	Machine5
Job1	$(\rho-1,\rho,\rho+1)$; $\rho=([50,60],[50,66])$	$(\rho-3,\rho,\rho+3)$; $\rho=([15,17],[13,18])$	$(\rho-4,\rho,\rho+4)$; $\rho=([11,13],[11,15])$	$(\rho-2,\rho,\rho+2)$; $\rho=([20,24],[20,25])$	$(\rho-1,\rho,\rho+1)$; $\rho=([18,20],[18,23])$
Job2	$(\rho-2,\rho,\rho+2)$; $\rho=([15,20],[15,25])$	$(\rho-5,\rho,\rho+5)$; $\rho=([40,43],[40,45])$	$(\rho-3,\rho,\rho+3)$; $\rho=([23,25],[20,25])$	$(\rho-7,\rho,\rho+7)$; $\rho=([33,36],[34,37])$	$(\rho-2,\rho,\rho+2)$; $\rho=([17,19],[17,24])$
Job3	$(\rho-6,\rho,\rho+6)$; $\rho=([45,65],[45,70])$	$(\rho-1,\rho,\rho+1)$; $\rho=([30,55],[30,60])$	$(\rho-3,\rho,\rho+3)$; $\rho=([31,37],[31,40])$	$(\rho-3,\rho,\rho+3)$; $\rho=([17,18],[17,24])$	$(\rho-5,\rho,\rho+5)$; $\rho=([80,95],[80,100])$
Job4	$(\rho-2,\rho,\rho+2)$; $\rho=([10,24],[10,30])$	$(\rho-10,\rho,\rho+10)$; $\rho=([77,83],[77,91])$	$(\rho-5,\rho,\rho+5)$; $\rho=([22,25],[22,28])$	$(\rho-3,\rho,\rho+3)$; $\rho=([44,48],[43,49])$	$(\rho-11,\rho,\rho+11)$; $\rho=([72,78],[69,80])$
Job5	$(\rho-2,\rho,\rho+2)$; $\rho=([15,17],[15,20])$	$(\rho-1,\rho,\rho+1)$; $\rho=([84,86],[82,90])$	$(\rho-8,\rho,\rho+8)$; $\rho=([65,70],[65,75])$	$(\rho-8,\rho,\rho+8)$; $\rho=([48,52],[48,52])$	$(\rho-30,\rho,\rho+30)$; $\rho=([80,95],[80,100])$
Job6	$(\rho-1,\rho,\rho+1)$; $\rho=([34,42],[34,45])$	$(\rho-1,\rho,\rho+1)$; $\rho=([11,15],[10,16])$	$(\rho-11,\rho,\rho+11)$; $\rho=([60,65],[58,65])$	$(\rho-5,\rho,\rho+5)$; $\rho=([29,33],[27,37])$	$(\rho-25,\rho,\rho+25)$; $\rho=([80,85],[77,92])$
Job7	$(\rho-4,\rho,\rho+4)$; $\rho=([19,22],[19,24])$	$(\rho-2,\rho,\rho+2)$; $\rho=([16,17],[16,19])$	$(\rho-8,\rho,\rho+8)$; $\rho=([27,29],[27,31])$	$(\rho-6,\rho,\rho+6)$; $\rho=([23,26],[23,28])$	$(\rho-6,\rho,\rho+6)$; $\rho=([29,32],[29,35])$
Job8	$(\rho-5,\rho,\rho+5)$; $\rho=([28,31],[28,35])$	$(\rho-12,\rho,\rho+12)$; $\rho=([36,39],[36,41])$	$(\rho-30,\rho,\rho+30)$; $\rho=([115,120],[100,124])$	$(\rho-1,\rho,\rho+1)$; $\rho=([15,17],[15,20])$	$(\rho-3,\rho,\rho+3)$; $\rho=([19,21],[17,24])$
Job9	$(\rho-4,\rho,\rho+4)$; $\rho=([28,30],[28,32])$	$(\rho-3,\rho,\rho+3)$; $\rho=([38,40],[35,45])$	$(\rho-12,\rho,\rho+12)$; $\rho=([124,128],[124,128])$	$(\rho-5,\rho,\rho+5)$; $\rho=([11,12],[11,14])$	$(\rho-20,\rho,\rho+20)$; $\rho=([93,102],[87,115])$
Job10	$(\rho-11,\rho,\rho+11)$; $\rho=([24,27],[23,28])$	$(\rho-15,\rho,\rho+15)$; $\rho=([114,119],[105,121])$	$(\rho-7,\rho,\rho+7)$; $\rho=([19,21],[19,24])$	$(\rho-25,\rho,\rho+25)$; $\rho=([91,94],[84,94])$	$(\rho-10,\rho,\rho+10)$; $\rho=([48,52],[40,59])$

TABLE II
DUE DATE TIMES AND PENALTY COEFFICIENT

Jobs	Due date	Tardiness penalty cost
J1	468	2
J2	325	2
J3	923	7
J4	513	4
J5	850	4
J6	690	3
J7	602	4
J8	350	3
J9	873	3
J10	245	2

TABLE III
COMPUTATIONAL RESULTS

N_p	G_{max}	**ES-SCA**		**SCA**	
		TWT	**Error**	**TWT**	**Error**
30	100	424	**14,9%**	378	0,0%
	150	381	3,3%	450	19,0%
	200	388	5,1%	411	8,7%
50	100	395	7,0%	552	**46,0%**
	150	397	7,6%	463	22,5%
	200	407	10,3%	404	6,9%
80	100	409	10,8%	442	16,9%
	150	396	7,3%	433	14,6%
	200	369	0,0%	399	5,6%

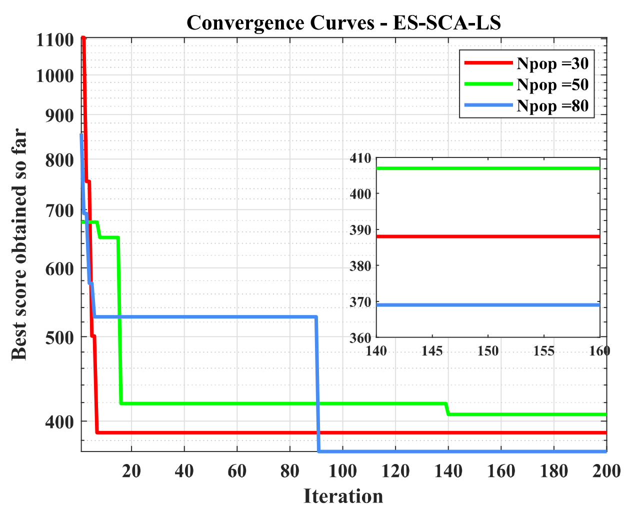

Fig. 3. Convergence Curve of $G_{max} = 200$

TABLE IV
SENSITIVITY ANALYSIS OF HURWICZ PARAMETER

(α,β)	θ	TWT_{op}	TWT_{pess}	TWT_{HC}
(0.8,0.9)	0,3	1361,0	0,0	408,3
	0,5	727,0	121,0	424,0
	0,7	715,0	432,0	630,1
	0,9	714,0	527,0	695,3

VI. CONCLUSION

In this paper, we investigate the total weighted tardiness permutation flow-shop problem under uncertainty. The main contributions of this paper are discussed below. First, the proposed model uses the fuzzy rough theory and the Hurwicz criterion. The fuzzy rough theory represents a powerful alternative to other classical theories, such as random, fuzzy, and rough. Moreover, the Hurwicz criterion is more flexible than optimistic or pessimistic criteria. Therefore, the proposed model can provide better theoretical directions for decision-makers. Second, a hybrid intelligence based on a fuzzy rough simulation and a discrete eagle strategy combined with a sine-cosine algorithm is developed to solve the model. Furthermore,

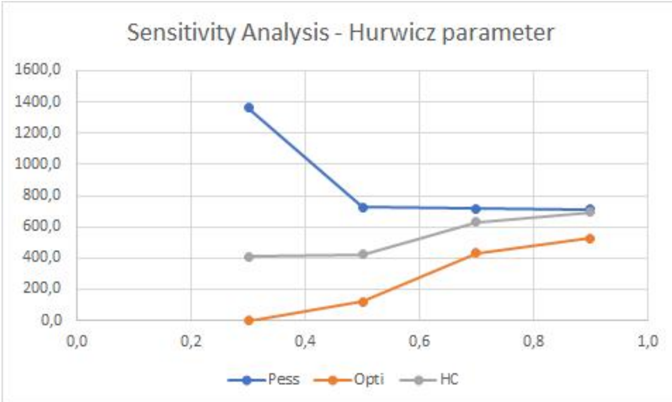

Fig. 4. Hurwicz parameter sensitivity analysis

the computational experiments demonstrate that the proposed strategy is more competitive and efficient to solve the model than the standard SCA.

References

[1] Q.-K. Pan and L. Wang, "Effective heuristics for the blocking flowshop scheduling problem with makespan minimization," *Omega*, vol. 40, no. 2, pp. 218–229, 2012.

[2] A. R. Kan, *Machine scheduling problems: classification, complexity and computations*. Springer Science & Business Media, 2012.

[3] M. Pinedo, "Scheduling: theory, algorithms and systems, 1995."

[4] B. Liu and B. Liu, *Theory and practice of uncertain programming*. Springer, 2002.

[5] L. A. Zadeh, "Fuzzy sets," *Information and control*, vol. 8, no. 3, pp. 338–353, 1965.

[6] Z. Pawlak, "Rough sets," *International Journal of Parallel Programming*, vol. 11, no. 5, pp. 341–356, 1982.

[7] J. Peng and K. Song, "Fuzzy flow-shop scheduling models based on credibility measure," in *Fuzzy Systems, 2003. FUZZ'03. The 12th IEEE International Conference on*, vol. 2. IEEE, 2003, pp. 1423–1427.

[8] J. Peng and K. Iwarnura, "Flow-shop scheduling models with parameters represented by rough variables," *Tsinghua Science and Technology*, vol. 8, no. 1, pp. 55–59, 2003.

[9] A. Charnes and W. W. Cooper, *Management models and industrial applications of linear programming*. JSTOR, 1961, vol. 1.

[10] S. Mirjalili, "Sca: a sine cosine algorithm for solving optimization problems," *Knowledge-Based Systems*, vol. 96, pp. 120–133, 2016.

[11] L. Hurwicz, "A class of criteria for decision-making under ignorance," *Cowles Comission Paper*, vol. 356, 1951.

[12] B. Liu, "Fuzzy rough theory," in *Uncertainty Theory*. Springer, 2004, pp. 349–367.

[13] X.-S. Yang and S. Deb, "Eagle strategy using lévy walk and firefly algorithms for stochastic optimization," in *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*. Springer, 2010, pp. 101–111.

[14] R. Sindhu, R. Ngadiran, Y. M. Yacob, N. A. H. Zahri, and M. Hariharan, "Sine–cosine algorithm for feature selection with elitism strategy and new updating mechanism," *Neural Computing and Applications*, vol. 28, no. 10, pp. 2947–2958, 2017.

[15] O. E. Turgut, "Thermal and economical optimization of a shell and tube evaporator using hybrid backtracking search—sine–cosine algorithm," *Arabian Journal for Science and Engineering*, vol. 42, no. 5, pp. 2105–2123, 2017.

[16] Y.-D. Kim, "Heuristics for flowshop scheduling problems minimizing mean tardiness," *Journal of the Operational Research Society*, vol. 44, no. 1, pp. 19–28, 1993.

[17] V. Fernandez-Viagas and J. M. Framinan, "Neh-based heuristics for the permutation flowshop scheduling problem to minimise total tardiness," *Computers & Operations Research*, vol. 60, pp. 27–36, 2015.

[18] R. N. Mantegna, "Fast, accurate algorithm for numerical simulation of levy stable stochastic processes," *Physical Review E*, vol. 49, no. 5, p. 4677, 1994.

[19] J. Grabowski and M. Wodecki, "A very fast tabu search algorithm for the permutation flow shop problem with makespan criterion," *Computers & Operations Research*, vol. 31, no. 11, pp. 1891–1909, 2004.

[20] E. Nowicki and C. Smutnicki, "A fast tabu search algorithm for the permutation flow-shop problem," *European Journal of Operational Research*, vol. 91, no. 1, pp. 160–175, 1996.

[21] B. Qian, L. Wang, R. Hu, D. Huang, and X. Wang, "A de-based approach to no-wait flow-shop scheduling," *Computers & Industrial Engineering*, vol. 57, no. 3, pp. 787–805, 2009.