E-ISSN: 2717-3453 | P-ISSN: 2783-1442

Journal of Fuzzy Extension and Applications

www.journal-fea.com

J. Fuzzy. Ext. Appl. Vol. 3, No. 3 (2022) 249–262.

Paper Type: Research Paper

An Efficient Parallel Greedy Algorithm for Fuzzy Hybrid Flow Shop Scheduling with Setup Time and Lot Size: A Case Study in Apparel Process

Orhan Engin[1,*], Meral İşler[2]

[1] Department of Industrial Engineering, Faculty of Engineering, Konya Technical University, Konya, Turkey; oengin@ktun.edu.tr.

[2] Department of Fashion Design, Faculty of Architecture and Design, Selcuk University, Konya, Turkey; meralisler@selcuk.edu.tr.

Citation:

Engin, O., & İşler, M. (2022). An efficient parallel greedy algorithm for fuzzy hybrid flow shop scheduling with setup time and lot size: a case study in apparel process. *Journal of fuzzy extension and applications*, *3*(3), 249-262.

Received: 19/11/2021 Reviewed: 10/12/2021 Revised: 13/01/2022 Accepted: 06/02/2022

Abstract

This paper deals with the Fuzzy Hybrid Flow Shop (FHFS) scheduling inspired by a real apparel process. A Parallel Greedy (PG) algorithm is proposed to solve the FHFS problems with Setup Time (ST) and Lot Size (LS). The fuzzy model is used to define the uncertain setup and Processing Time (PT) and Due Dates (DDs). The setup and PTs are defined by a Triangular Fuzzy Number (TAFN). Also, the Fuzzy Due Date (FDD) is denoted by a doublet. The tardiness, the tardy jobs, the setup and Idle Time (IT), and the Total Flow (TF) time are minimized by the proposed PG algorithm. The effectiveness of the proposed PG algorithm is demonstrated by comparing it with the Genetic Algorithm (GeA) in the literature. A real-world application in an apparel process is done. According to the results, the proposed PG algorithm is an efficient method for FHFS scheduling problems with ST and LS in real-world applications.

Keywords: Hybrid flow shop, Fuzzy processing time and due date, Parallel greedy algorithm, Case study, Setup time, Lot size.

1 | Introduction

Apparel is a type of dynamic production in which appointed time is short, loss of labor force is most, and labor-focused structure is continued instead of developing technology in recent times. Hereby, productivity, planning, and scheduling in the apparel manufacturing process are significant due to these factors in production structure. The configuration of the workshop in the apparel manufacturing process is a typical Hybrid Flow Shop Scheduling Problem (HFSSP). HFSSP combines the properties of flow shop and parallels machine scheduling problems. In HFSSP, machines are arranged into s stage and in each stage k (k=1,…,s) there are m_k identical parallel machines. Job j (j=1,…,n) has to be processed on any one machine at each stage [1] and [2]. The HFSSP is known as NP-hard [1]. The HFSSP was first studied by Arthanary [3]. There are also some other papers on HFSSP: Engin and Döyen [4] considered an artificial immune algorithm for solving HFSSP. Tang et al. [5] developed a neural network to solve the HFSSP. Zandieh et al. [6] presented an immune system to solve the HFSSP with STs.

Corresponding Author: oengin@ktun.edu.tr

http://dx.doi.org/10.22105/jfea.2021.314312.1169

Allaoui and Artiba [7] investigated the two-stage HFSSP. Voß and Witt [8] are interested in a multi-project scheduling problem from an HFSSP. Alaykýran et al. [9] generated an ant colony to solve HFSSP. Kahraman et al. [10] developed a Genetic Algorithm (GeA) for solving HFSSP. Liao et al. [11] proposed a Particle Swarm Algorithm (PSA) for solving HFSSP. Chung and Liao [12] generated an artificial immune system for solving the HFSSP. Li et al. [13] proposed a hybrid variable neighborhood search method to solve the HFSSP. Marichelvam et al. [14] developed a cuckoo search method for HFSSP. Cui and Gu [15] proposed an artificial Bee Colony Algorithm (BCA) for minimizing the makespan on HFSSP. Akkoyunlu et al. [16] presented a harmony search method to solve the multiprocessor tasks HFSSP. Engin and Engin [17] developed a memetic algorithm to solve the multiprocessor HFSSP. Also, Engin et al. [18] proposed a memetic global and local search method to solve the multiprocessor HFSSP.

In the real world manufacturing environment, ST and Processing Time (PT) and Due Date (DD) are very dynamic. In this research, the fuzzy model is used to define the uncertain ST and PT and DD for HFSSP with LS. In recent years, a fuzzy model is used intensively for scheduling problems from the literature. These are given below. Sakawa and Mori [19] considered Job Shop Problems (JSPs) with Fuzzy PT (FPT) and DD. They developed a GeA to solve these problems. Also, Sakawa and Kubota [20] investigated the multi-objective JSP with FPT and DD. Konno and Ishii [21] considered the fuzzy allowable open shop problem. Chanas and Kasperski [22] investigated the fuzzy single machine problem. Wang et al. [23] analyzed the ready and FPT scheduling problems. Temİzand Erol [24] studied fuzzy flow shop problems (FFSPs). Canbolat and Gundogar [25] proposed a fuzzy priority rule to solve the JSP. Peng and Liu [26] presented a methodology for fuzzy Parallel Machine Scheduling Problems (PMSP). Anglani et al. [27] considered a fuzzy identical PMSP. Petrovic et al. [28] developed a fuzzy model with LS, PT, and DD for JSP. Engin and Gözen [29] presented a fuzzy model for PMSP. They proposed a GeA and made a real-world application. Hu et al. [30] considered the fuzzy JSP. Lai et al. [31] evaluated the FFSP and to solve this problem a virus-evolutionary GeA is considered. Balin [32] developed a GeA approach embedded in a simulation model for solving the fuzzy PMSP. Engin et al. [33] presented a greedy method to solve the FFSP. Lei [34] presented a fuzzy flexible JSP and proposed a co-evolutionary GeA to solve the problem. Also, Lei and Guo [35] developed a swarm-based neighborhood search method for solving the fuzzy flexible JSP. Engin et al. [36] presented a fuzzy availability constraint for JSP. They developed a scatter search approach for solving these problems. Wang et al. [37] developed a hybrid artificial BCA to solve the fuzzy flexible JSP. Li and Pan [38] developed a hybrid discrete PSA for solving fuzzy JSP. Also, Li and Pan [39] presented a hybrid chemical reaction optimization method for solving the fuzzy JSP. Behnamian and Ghomi [40] are interested to a bi-objective FHFS. Also, Behnamian [41] presented a hybrid PSA method for solving the fuzzy PMSP. Palacios et al. [42] developed a PSA to minimize the makespan for a fuzzy open shop. Xu et al. [43] proposed a teaching learning-based method to solve the fuzzy flexible JSP. Wang et al. [44] proposed a distributed algorithm to solve the distributed FFSP. Yuan et al. [45] considered a fuzzy model for solving the FFSP. Emin Baysal et al. [46] proposed a BCA for solving distributed FFSP. Also, Baysal et al. [47] are interested to a multi-objective distributed FFSP. They proposed a BCA to solve this problem. Engin and Yılmaz [48] developed a GeA to solve fuzzy multi-objective multi-processor HFSSP. İşler and Engin [49] interested in the FHFS and a PG algorithm presented.

The contribution of this search is given below:

- *A version of the HFSSP with LS and fuzzy setup, PTs, and DDs is considered first.*
- *A PG algorithm is proposed for solving the HFSSP.*
- *This PG algorithm is used at an apparel process.*
- *The results are compared with the GeA at the literature.*

Engin and İşler |J. Fuzzy. Ext. Appl. 3(3) (2022) 249-262

The remaining parts of this study are defined as follows: in Section 2, FHFS with ST and LS is given. In Section 3, the developed PG algorithm is explained to solve FHFS with ST and LS problems. Section 4 describes the GeA. In Section 5, a case study of the apparel process is given. In Section 6, the conclusion is explained.

2| Hybrid Flow Shop with Fuzzy Environment

In the real shop floor systems, some parameters are exactly not known. Thus, a fuzzy model is used to determine these uncertain parameters such as setup and PTs. At the FHFS scheduling with ST and LS problems, each lot is consisting of a set of consecutive processed jobs from the same class. If a lot is assigned to one of the available machines, a setup is required at the beginning of the first job in that lot. In the literature, ST is accepted to be part of the PT [50] but in real shop floor systems, the ST is separated from the PTs.

The notation of the FHFS is presented as follows [50]:

j: jobs index, j = 1,…,J.

i: processing stages index, i = 1,…,10.

l: processing lots index, (for each month), l = 1,…,12.

d_l: DD for lots.

n_l: number of jobs at the lot l.

m_i: available machines at stage i.

$\tilde{a}_{l,i}$: ST for jobs at stage i in lot l.

$\tilde{p}_{j,i}$: PT of job j at stage i.

IT_i: Idle Time (IT) of machines at the stage i.

$\tilde{c}_{j,i}$: Completion Time (CT) of job j at stage i.

$\tilde{P}_{l,i}$: PT of all jobs in lot l at stage i.

$\tilde{C}_{l,i}$: CT of the last job in lot l at stage I.

$\tilde{C}_{nax}$: makespan.

In the real-world application, the fuzzy ST of the lot is represented by a Triangular Fuzzy Number (TAFN) and denoted by a triplet as shown in *Fig. 1* [51]-[53].

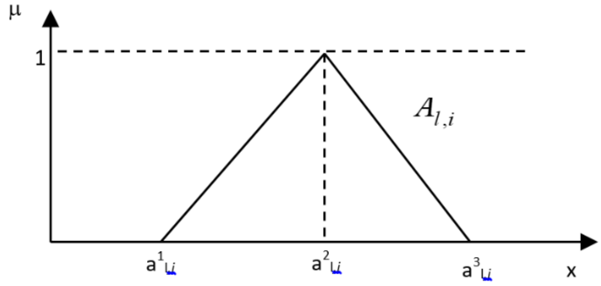

Fig. 1. Fuzzy ST.

The FPT of operation p is represented by a TAFN and denoted by a triplet (p1, p2, p3) as given in *Fig. 2* [20], [29], [51].

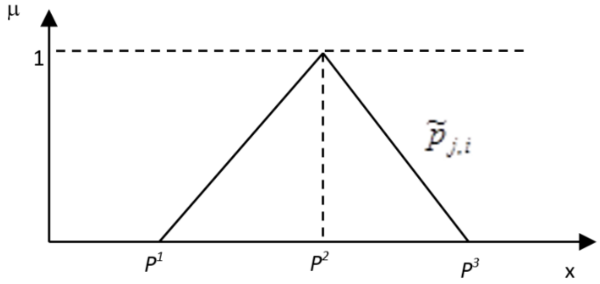

Fig. 2. FPT.

The FDD is represented by the degree of satisfaction with respect to job CT and denoted by a doublet (d1, d2) as shown in *Fig. 3* [20], [29].

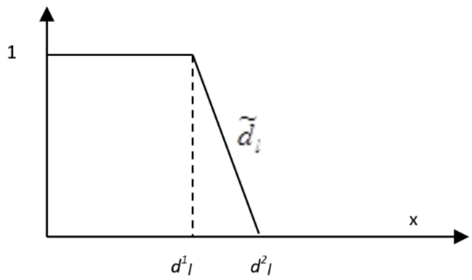

Fig. 3. Fuzzy DD.

For the TAFN defined by the three real numbers, (p1 ≤ p2 ≤p3), as in *Eq. (1)* [29]:

$$C_1(\tilde{A}) = \frac{a^1 + 2a^2 + a^3}{4}. \tag{2.a}$$

$$C_2(\tilde{A}) = a^2. \tag{2.b}$$

$$C_3(\tilde{A}) = a^3 - a^1. \tag{2.c}$$

A fuzzy set with linear decreasing membership function is given in *Eq. (3):*

$$\mu_D(x) = \begin{cases} 1 \ldots\ldots\ldots\ldots x < d^1 \\ \dfrac{x - d^2}{d^1 - d^2} \ldots\ldots d^1 < x \le d^2 \\ 0 \ldots\ldots\ldots d^2 < x \end{cases}. \tag{3}$$

For the fuzzy DD of each lot, when the fuzzy CT of the last job in lot *l* is expressed as TAFN, as an index showing the portion of that meets the fuzzy DD. The intersection between fuzzy CT and DD is given in *Fig. 4* [20], [27], [29].

Engin and İşler |J. Fuzzy. Ext. Appl. 3(3) (2022) 249-262

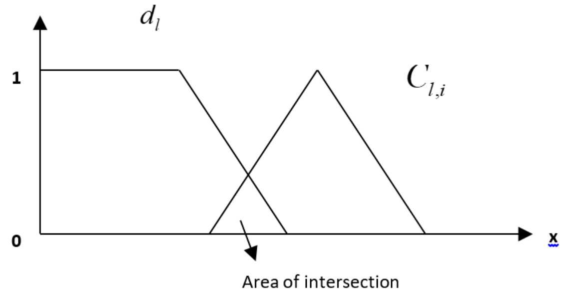

Fig. 4. Satisfaction Grade (SG).

The SG is defined in *Eq. (4)*:

$$SG_T(\tilde{C}_{l,i}) = \frac{\text{area}(\tilde{d}_l \cap \tilde{C}_{l,i})}{\text{area}(\tilde{C}_{l,i})}. \quad (4)$$

3 | PG Algorithm

Greedy Algorithm (GrA) was used by Binato et al. [54] for solving the JSP. Aiex et al. [55] proposed a GrA to solve the JSP. Ruiz and Stützle [56] developed an iterated GrA for Permutation Flow Shop Problem (PFSP). Baraz and Mosheiov [57] presented a GrA for flow shop problems. Li and Zhao [58] developed a GrA to solve the flexible flow shop scheduling problem. Kahraman et al. [59] proposed a PG algorithm to solve the multiprocessor HFSSP. Akgöbek et al. [60] proposed a PG algorithm to solve the open shop scheduling problem. Engin et al. [33] proposed a GrA for solving multi-objective FFSP. Pan and Ruiz [61] developed an iterated GrA for the mixed no-idle PFSP. Karabulut and Tasgetiren [62] proposed an iterated GrA for solving the traveling salesman problem. Fernandez-Viagas and Framinan [63] generated a bounded-search iterated GrA for the distributed PFSP.

GrA has an iterative search process. The solution is built in small steps. It uses two phases. These are destruction and construction.

3.1 | Destruction and Construction

At the destruction, a permutation π of n jobs are chosen randomly without repetition of k (number of a subgroup) jobs. These k jobs are then removed from π in the order in which they were chosen [59]. As a result, two subsequences are obtained.

Also at the construction phase, the jobs in π_R are reinserted into π_D. In this research, two constructive methods, Initial Position (IP), and End Position (EP) are used.

IP: the π_R is added to the beginning of the π_D.

EP: the π_R is added to the end of the π_D.

3.2 | The Proposed Parallel GrA for FHFS Problem

An efficient PG algorithm is proposed for solving the FHFS with ST and LS. The objectives are defined as follows [28], [29], [64].

SG of Average Tardiness (AT) is given in *Eq. (5):*

$$SG_{AT} = \frac{1}{n}\sum_{j=1}^{n} \tilde{C}_{l,i}. \quad (5)$$

SG of Number of Tardy (NT) jobs is given in *Eq. (6)*:

$$SG_{NT} = \sum_{j=1}^{n} u_j \qquad u_j=1 \text{ if } T_j>0 \text{ ,otherwise } u_j=0. \quad (6)$$

SG of total ST is given in *Eq. (7):*

$$SG_{ST} = \sum a_{l,i}. \quad (7)$$

SG of total IT is given in *Eq. (8)*:

$$SG_{IT} = \sum IT_i. \quad (8)$$

SG of Total Flow (TF) time is given in *Eq. (9)*:

$$SG_{TF} = \sum \tilde{C}_{l,i}. \quad (9)$$

The SG of all objectives (ϕ) is given in *Eq. (10)*:

$$\phi = (SGAT + SGNT + SGST + SGIT + SGTF) / 5. \quad (10)$$

The pseudocode of the proposed PG algorithm is defined as follows:

```
Set initial parameters:
   Jobs number.
   Lots number.
   fuzzy ST.
   FPT.
   fuzzy DD.
   lot sizes.
Set the PG algorithm value:
   population size.
   subgroups number.
   iterations number.
   greedy ratio.
   construction method.
Create a population randomly:
   calculate SG_AT , SG_NT , SG_ST , SG_IT , SG_TF,  ϕ and
For each generation subpopulations do:
   create two subpopulations randomly.
   Destruction.
   construction (generate new string).
   calculate SG_AT , SG_NT , SG_ST , SG_IT , SG_TF,  ϕ  and
For each solutions local search do:
   calculate SG_AT , SG_NT , SG_ST , SG_IT , SG_TF,  ϕ and
   While stopping criteria meet
End.
```

3.3 | The Parameter of the Proposed PG Algorithm for FHFS Problem

The performance of the PG algorithm depends on the initial parameter sets. To determine these parameter sets, a full factorial experimental design is made. Five parameters (factors) each with different possible values are given in *Table 1*.

J. Fuzzy. Ext. Appl
Engin and İşler |J. Fuzzy. Ext. Appl. 3(3) (2022) 249-262

Table 1. The parameters and range.

Parameters	Range
Initial population size	15; 30
Subset size	2; 3; 4; 5; 6; 7; 8; 9; 10; 11; 12; 13
Iteration number	250
Greedy ratio	0.1; 0.2: 0.3: 0.4: 0.5: 0.6: 0.7: 0.8: 0.9
Construction method	IP, EP

The Best Parameter Set (BPS) found is given below (*Table 2*).

Table 2. BPS value.

Parameter	Value
Initial population size	15
Sub set size	2
Iteration number	250
Greedy ratio	0.4
Construction method	EP

4 | Genetic Algorithm

GeA is a population-based stochastic search method. The GeA was first proposed by Goldberg [65]. Genetic is one of the best-known algorithms for solving scheduling problems [66]. The Genetic is a population-based algorithm. GeA maintains a set of populations in each generation [67].

Better solutions can be produced by genetic operators such as reproduction, crossover, and mutation methods. The new solutions are constructed by these operators. There are a few studies on GeA to solve fuzzy scheduling problems. Engin and Gözen [29] proposed a GeA for solving PMSP with FPT. The efficiency of the proposed PG algorithm is demonstrated by comparing it with the GeA of Engine and Gözen [29]. The BPS of Engin and Gözen [29]'s GeA are used to solve FHFS problems with ST and LS problems. It is given in *Table 3*.

Table 3. The BPS of Engin and Gözen [29]'s GeA.

Selection Method	Selection Ratio	Crossover Method	Crossover Ratio	Mutation Method	Mutation Ratio
Tournament	0.2	Position Based Crossover	0.1	Arbitrary three job change	0.3

The details of these parameters (*Table 1*) are in the Engine and Gözen [29]'s paper in the literature.

5 | A Case Study in Apparel Process

In the textile and apparel industry, there is a seasonal change and plenty of models. Also, the unit time of operations in the models may be different from each other. The apparel process consists of parallel workstations doing the same job. The station layout of the enterprise where the application is made is as in *Fig. 5*.

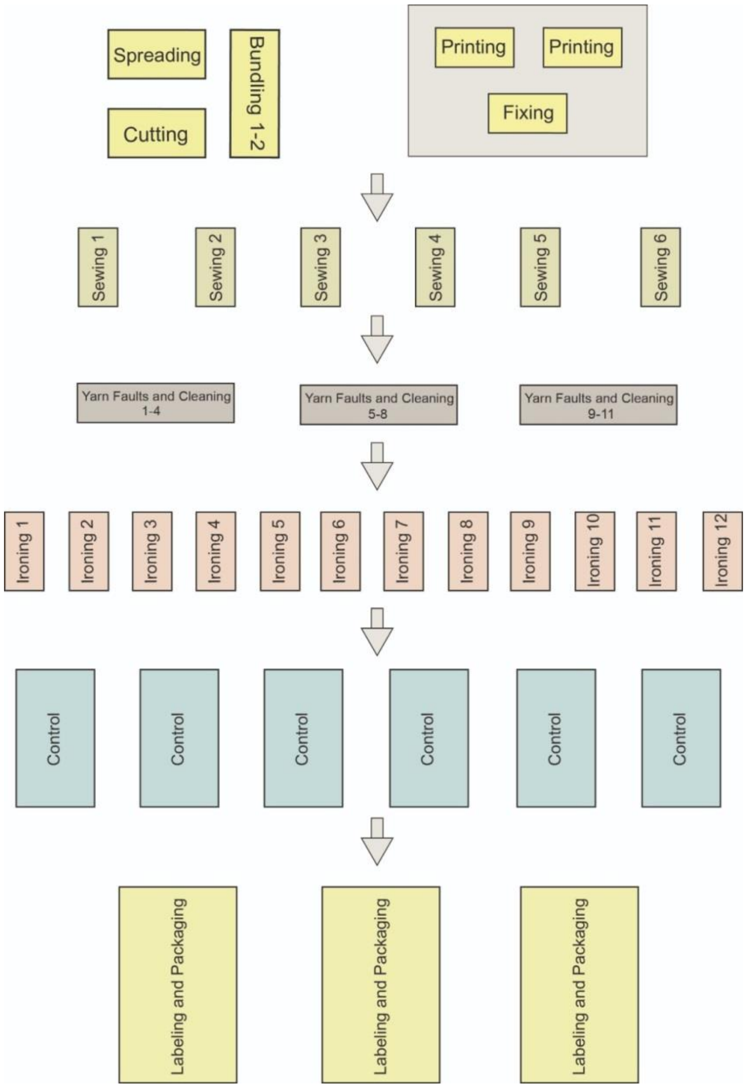

Fig. 5. Apparel manufacturing workstation layout.

A PG algorithm is developed for solving the FHFS problem with ST and LS. Also, the developed PG algorithm is tested on the apparel process. The apparel process consists of the steps in *Fig. 5*.

While some steps in the apparel process have ST, some do not. In this case study, spreading, cutting and sewing have ST.

For our application, the definitions of process steps and machines or job station numbers are given in *Table 4*.

Table 4. Apparel manufacturing process steps and machines or job station numbers.

Stage	Process Steps	Available Machines or Job Station Numbers
1	Spreading	1
2	Cutting	1
3	Bundling	2
4	Screen Printing	2
5	Fixing	1
6	Sewing	6
7	Yarn Faults And Clearing	11
8	Ironing	12
9	Control	6
10	Labeling and Packaging	3

The algorithm is implemented in Borland Delphi. In addition, the algorithm is executed with an Intel Xeon E5420, 2.5 GHz PC.

The FHFS problem with ST and LS is solved by the proposed PG algorithm and GeA. The deviation percent is found as in *Eq. (11)*.

$$\text{Deviation percent} = \frac{\widetilde{C}_{max}(GeA) - \widetilde{C}_{max}(PG)}{\widetilde{C}_{max}(PG)}. \quad (11)$$

A case study is done at the apparel process, for scheduling, six months customer orders are taken. For each month, 12 lots are scheduled with FDD. For six months, 72 lots are scheduled. The fitness function and SG of five objectives are given in *Table 5*.

In *Table 5*, the best $\tilde{C}_{max}$ for five months is found by the proposed PG algorithm. GeA found the best values for only four months. Also, the deviation percent of $\tilde{C}_{max}$ values for five months at the proposed PG algorithm are better than GeA. For the SG_{AT}, the proposed PG algorithm calculates better values for 3 months than GeA. Only four months GeA calculates the better value and for 2 months the proposed PG algorithm and GeA calculate the same SG_{AT}. For the SG_{NT}, the proposed PG algorithm calculates better values for 4 months than GeA. Only 2 months, GeA calculates a better value than the PG algorithm. For the SG_{ST}, the proposed PG algorithm calculates better values for one month than GeA. For 5 months the proposed PG algorithm and GeA calculate the same S_{ST}. For the SG_{IT}, the proposed PG algorithm calculates better values for 4 months than GeA. Only 2 months GeA calculates the better value. For the SG_{TF}, the proposed PG algorithm calculates better values for only one month than GeA. For 4 months GeA calculates the better value and for 1month the proposed PG algorithm and GeA calculate the same SG_{TF} values.

Table 5. Fitness values of the aggregated SG and objectives.

Month	Fitness Value	SGAT		SGNT		SGST		SGIT		SGTF		φ		$\widetilde{C_{max}}$(second)		Dev.$\widetilde{C_{max}}$	Number of Trady Jobs	
		PG	GeA	PG	GeA	PG	GeA	PG	GeA	PG	GeA	PG	GeA	PG	GeA		PG	GeA
1	Best	1	1	1	1	0.992	0.992	0.730	0.653	1	1	0.951	0.921	763560	812430	6.40	0	1
	Average	0.975	0.975	0.783	0.716	0.992	0.992	0.773	0.699	0.908	0.854	0.886	0.847	851957	85502	0.95	0.96	0.50
	Std. dev.	0.013	0.015	0.250	0.252	0.00	0.00	0.40	0.058	0.092	0.199	0.043	0.071	33185	33046	-	0.18	0.50
2	Best	0.985	0.966	1	0.5	0.993	0.993	0.736	0.721	0.940	0.963	0.931	0.828	879030	917520	4.37	2	2
	Average	0.942	0.932	0.566	0.293	0.993	0.993	0.978	0.743	0.730	0.815	0.806	0.755	927466	940255	1.37	1.93	1.86
	Std. dev.	0.026	0.055	0.172	0.250	0.00	0.00	0.041	0.037	0.123	0.086	0.051	0.069	17240	13953	-	0.78	0.81
3	Best	0.923	0.835	1	0.00	0.976	0.975	0.280	0.308	0.689	1	0.778	0.623	545100	622230	14.14	1	1
	Average	0.890	0.857	0.733	0.00	0.977	0.975	0.296	0.295	0.755	0.899	0.729	0.602	583364	665579	14.09	0.5	0.7
	Std. dev.	0.035	0.029	0.440	0.00	0.00	0.00	0.020	0.015	0.103	0.115	0.082	0.029	25455	23461	-	0.0	0.25
4	Best	0.972	1	1	1	0.91	0.991	0.785	0.747	0.728	1	0.890	0.947	585690	566070	GA better	0	0
	Average	0.972	0.984	0.783	0.883	0.991	0.991	0.755	0.702	0.754	1	0.847	0.912	605554	581157	GA better	0.53	0.50
	Std. dev.	0.018	0.028	0.25	0.215	0.00	0.00	0.33	0.03	0.220	0	0.021	0.048	18271	9047		0.50	0.50
5	Best	1	1	1	1	0.977	0.976	0.324	0.280	1	1	0.860	0.851	1266600	1434450	13.25	0	0
	Average	0.995	1	0.966	1	0.977	0.976	0.286	0.262	1	1	0.852	0.847	1288485	1459033	13.23	0	0
	Std. dev.	0.00	0.00	0.182	0.00	0.00	0.00	0.019	0.21	0.00	0.00	0.004	0.004	26797	11800	-	0	0
6	Best	1	1	1	1	0.982	0.82	0.289	0.328	1	1	0.870	0.813	894030	1010340	13.00	1	1
	Average	0.861	0.753	0.833	0.333	0.981	0.981	0.338	0.363	0.885	0.888	0.780	0.664	970433	1037901	6.95	0.66	0.73
	Std. dev.	0.093	0.117	0.380	0.46	0.00	0.00	0.033	0.020	0.086	0.17	0.093	0.132	30230	-		0.49	0.45

6 | Conclusions

In this paper, an FHFS problem with ST and LS problem is considered. This problem is inspired by an apparel process. A PG algorithm is developed to solve the FHFS problems. A fuzzy model is used to determine the uncertain ST, PT, and DD. The ST and PT are presented by a TAFN and the DD is denoted by a doublet. The objectives are considered, SG of AT, NT, ST, IT, TF, and for all objectives is ϕ. The proposed PG algorithm is compared with GeA in the literature and tested on an apparel process. The proposed PG algorithm calculates the best SG_{AT}, SG_{NT}, SG_{ST}, SG_{IT}, SG_{TF}, and ϕ of all objectives. According to the results, the proposed PG algorithm is an efficient method for FHFS scheduling with ST and LS in real-world problems.

In the future, the proposed PG algorithm can be used for the fuzzy multi-processor HFSSP with ST and LS scheduling problems.

Acknowledgments

The authors are grateful to the anonymous referees and the issues editor for their valuable comments.

Conflicts of Interest

The authors declare that no conflict of interest.

References

[1] Gupta, J. N. (1988). Two-stage, hybrid flowshop scheduling problem. *Journal of the operational research society*, *39*(4), 359-364.

[2] Carlier, J., & Neron, E. (2000). An exact method for solving the multi-processor flow-shop. *RAIRO-operations research*, *34*(1), 1-25.

[3] Arthanary, T. S. (1971). An extension of two machine sequencing problem. *Opsearch*, *8*, 10-22.

[4] Engin, O., & Döyen, A. (2004). A new approach to solve hybrid flow shop scheduling problems by artificial immune system. *Future generation computer systems*, *20*(6), 1083-1095.

[5] Tang, L., Liu, W., & Liu, J. (2005). A neural network model and algorithm for the hybrid flow shop scheduling problem in a dynamic environment. *Journal of intelligent manufacturing*, *16*(3), 361-370.

[6] Zandieh, M., Ghomi, S. F., & Husseini, S. M. (2006). An immune algorithm approach to hybrid flow shops scheduling with sequence-dependent setup times. *Applied mathematics and computation*, *180*(1), 111-127.

[7] Allaoui, H., & Artiba, A. (2006). Scheduling two-stage hybrid flow shop with availability constraints. *Computers & operations research*, *33*(5), 1399-1419.

[8] Voß, S., & Witt, A. (2007). Hybrid flow shop scheduling as a multi-mode multi-project scheduling problem with batching requirements: a real-world application. *International journal of production economics*, *105*(2), 445-458.

[9] Alaykýran, K., Engin, O., & Döyen, A. (2007). Using ant colony optimization to solve hybrid flow shop scheduling problems. *The international journal of advanced manufacturing technology*, *35*(5), 541-550.

[10] Kahraman, C., Engin, O., Kaya, I., & Kerim Yilmaz, M. (2008). An application of effective genetic algorithms for solving hybrid flow shop scheduling problems. *International journal of computational intelligence systems*, *1*(2), 134-147.

[11] Liao, C. J., Tjandradjaja, E., & Chung, T. P. (2012). An approach using particle swarm optimization and bottleneck heuristic to solve hybrid flow shop scheduling problem. *Applied soft computing*, *12*(6), 1755-1764.

[12] Chung, T. P., & Liao, C. J. (2013). An immunoglobulin-based artificial immune system for solving the hybrid flow shop problem. *Applied soft computing*, *13*(8), 3729-3736.

[13] Li, J. Q., Pan, Q. K., & Wang, F. T. (2014). A hybrid variable neighborhood search for solving the hybrid flow shop scheduling problem. *Applied soft computing*, *24*, 63-77.

[14] Marichelvam, M. K., Prabaharan, T., & Yang, X. S. (2014). Improved cuckoo search algorithm for hybrid flow shop scheduling problems to minimize makespan. *Applied soft computing*, *19*, 93-101.

[15] Cui, Z., & Gu, X. (2015). An improved discrete artificial bee colony algorithm to minimize the makespan on hybrid flow shop problems. *Neurocomputing*, *148*, 248-259.

[16] Akkoyunlu, M. C., Engın, O., & Büyüközkan, K. (2015, May). A harmony search algorithm for hybrid flow shop scheduling with multiprocessor task problems. *2015 6th international conference on modeling, simulation, and applied optimization (ICMSAO)* (pp. 1-3). IEEE.

[17] Engin, O., & Engin, B. (2018). Hybrid flow shop with multiprocessor task scheduling based on earliness and tardiness penalties. *Journal of enterprise information management*, *31*(6), 925-936.

[18] Engin, B. E., & Engin, O. (2020). A new memetic global and local search algorithm for solving hybrid flow shop with multiprocessor task scheduling problem. *SN applied sciences*, *2*(12), 1-14.

[19] Sakawa, M., & Mori, T. (1999). An efficient genetic algorithm for job-shop scheduling problems with fuzzy processing time and fuzzy duedate. *Computers & industrial engineering*, *36*(2), 325-341.

[20] Sakawa, M., & Kubota, R. (2000). Fuzzy programming for multiobjective job shop scheduling with fuzzy processing time and fuzzy duedate through genetic algorithms. *European Journal of operational research*, *120*(2), 393-407.

[21] Konno, T., & Ishii, H. (2000). An open shop scheduling problem with fuzzy allowable time and fuzzy resource constraint. *Fuzzy sets and systems*, *109*(1), 141-147.

[22] Chanas, S., & Kasperski, A. (2001). Minimizing maximum lateness in a single machine scheduling problem with fuzzy processing times and fuzzy due dates. *Engineering applications of artificial intelligence*, *14*(3), 377-386.

[23] Wang, C., Wang, D., Ip, W. H., & Yuen, D. W. (2002). The single machine ready time scheduling problem with fuzzy processing times. *Fuzzy sets and systems*, *127*(2), 117-129.

[24] Temİz, İ., & Erol, S. (2004). Fuzzy branch-and-bound algorithm for flow shop scheduling. *Journal of intelligent manufacturing*, *15*(4), 449-454.

[25] Canbolat, Y. B., & Gundogar, E. (2004). Fuzzy priority rule for job shop scheduling. *Journal of intelligent manufacturing*, *15*(4), 527-533.

[26] Peng, J., & Liu, B. (2004). Parallel machine scheduling models with fuzzy processing times. *Information sciences*, *166*(1-4), 49-66.

[27] Anglani, A., Grieco, A., Guerriero, E., & Musmanno, R. (2005). Robust scheduling of parallel machines with sequence-dependent set-up costs. *European journal of operational research*, *161*(3), 704-720.

[28] Petrovic, S., Fayad, C., Petrovic, D., Burke, E., & Kendall, G. (2008). Fuzzy job shop scheduling with lot-sizing. *Annals of operations research*, *159*(1), 275-292.

[29] Engin, O., & Gözen, Ş. E. R. I. F. E. (2009). Parallel machine scheduling problems with fuzzy processing time and fuzzy duedate: an application in an engine valve manufacturing process. *Journal of multiple-valued logic & soft computing*, *15*(2), 107-123.

[30] Hu, Y., Yin, M., & Li, X. (2011). A novel objective function for job-shop scheduling problem with fuzzy processing time and fuzzy due date using differential evolution algorithm. *The international journal of advanced manufacturing technology*, *56*(9), 1125-1138.

[31] Lai, P. J., & Wu, H. C. (2011). Evaluate the fuzzy completion times in the fuzzy flow shop scheduling problems using the virus-evolutionary genetic algorithms. *Applied soft computing*, *11*(8), 4540-4550.

[32] Balin, S. (2011). Parallel machine scheduling with fuzzy processing times using a robust genetic algorithm and simulation. *Information sciences*, *181*(17), 3551-3569.

[33] Engin, O., Yilmaz, M. K., Akkoyunlu, M. C., Baysal, M. E., & Sarucan, A. (2012). A greedy algorithm for multiobjective fuzzy flow-shop scheduling problem. In *uncertainty modeling in knowledge engineering and decision making* (pp. 189-194). Selcuk University Scientific Research Coordination Centre (BAP). https://doi.org/10.1142/9789814417747_0031

[34] Lei, D. (2012). Co-evolutionary genetic algorithm for fuzzy flexible job shop scheduling. *Applied soft computing*, *12*(8), 2237-2245.

[35] Lei, D., & Guo, X. (2012). Swarm-based neighbourhood search algorithm for fuzzy flexible job shop scheduling. *International journal of production research*, *50*(6), 1639-1649.

[36] Engin, O., Yilmaz, M. K., BAYSAL, M., & Sarucan, A. (2013). Solving fuzzy job shop scheduling problems with availability constraints using a scatter search method. *Journal of multiple-valued logic & soft computing*, *21*(3-4) 317-334.

[37] Wang, L., Zhou, G., Xu, Y., & Liu, M. (2013). A hybrid artificial bee colony algorithm for the fuzzy flexible job-shop scheduling problem. *International journal of production research*, *51*(12), 3593-3608.

[38] Li, J. Q., & Pan, Y. X. (2013). A hybrid discrete particle swarm optimization algorithm for solving fuzzy job shop scheduling problem. *The international journal of advanced manufacturing technology*, *66*(1), 583-596.

[39] Li, J. Q., & Pan, Q. K. (2013). Chemical-reaction optimization for solving fuzzy job-shop scheduling problem with flexible maintenance activities. *International journal of production economics*, *145*(1), 4-17.

[40] Behnamian, J., & Ghomi, S. F. (2014). Multi-objective fuzzy multiprocessor flowshop scheduling. *Applied soft computing*, *21*, 139-148.

[41] Behnamian, J. (2014). Particle swarm optimization-based algorithm for fuzzy parallel machine scheduling. *The international journal of advanced manufacturing technology*, *75*(5), 883-895.

[42] Palacios, J. J., González-Rodríguez, I., Vela, C. R., & Puente, J. (2014). Robust swarm optimisation for fuzzy open shop scheduling. *Natural computing*, *13*(2), 145-156.

[43] Xu, Y., Wang, L., Wang, S. Y., & Liu, M. (2015). An effective teaching–learning-based optimization algorithm for the flexible job-shop scheduling problem with fuzzy processing time. *Neurocomputing*, *148*, 260-268.

[44] Wang, K., Huang, Y., & Qin, H. (2016). A fuzzy logic-based hybrid estimation of distribution algorithm for distributed permutation flowshop scheduling problems under machine breakdown. *Journal of the operational research society*, *67*(1), 68-82.

[45] Yuan, F., Xu, X., & Yin, M. (2019). A novel fuzzy model for multi-objective permutation flow shop scheduling problem with fuzzy processing time. *Advances in mechanical engineering*, *11*(4), 1-9.

[46] Emin Baysal, M., Sarucan, A., Büyüközkan, K., & Engin, O. (2020, July). Distributed fuzzy permutation flow shop scheduling problem: a bee colony algorithm. *International conference on intelligent and fuzzy systems* (pp. 1440-1446). Springer, Cham.

[47] Baysal, M. E., Sarucan, A., Büyüközkan, K., & Engin, O. (2022). Artificial bee colony algorithm for solving multi-objective distributed fuzzy permutation flow shop problem. *Journal of intelligent & fuzzy systems*, *42*(1), 439-449.

[48] Engin, O., & Yılmaz, M. K. (2022). A fuzzy logic based methodology for multi-objective hybrid flow shop scheduling with multi-processor tasks problems and solving with an efficient genetic algorithm. *Journal of intelligent & fuzzy systems*, *42*(1), 451-463.

[49] İşler, M., & Engin, O. (2021, August). Fuzzy hybrid flow shop scheduling problem: an application. *International conference on intelligent and fuzzy systems* (pp. 623-630). Springer, Cham.

[50] Yimer, A. D., & Demirli, K. (2009). Fuzzy scheduling of job orders in a two-stage flowshop with batch-processing machines. *International journal of approximate reasoning*, *50*(1), 117-137.

[51] Bortolan, G., & Degani, R. (1985). A review of some methods for ranking fuzzy subsets. *Fuzzy sets and Systems*, *15*(1), 1-19. https://doi.org/10.1016/0165-0114(85)90012-0

[52] Kane, L., Sidibe, H., Kane, S., Bado, H., Konate, M., Diawara, D., & Diabate, L. (2021). A simplified new approach for solving fully fuzzy transportation problems with involving triangular fuzzy numbers. *Journal of fuzzy extension and applications*, *2*(1), 89-105.

[53] Kané, L., Diakité, M., Kané, S., Bado, H., Konaté, M., & Traoré, K. (2021). The new algorithm for fully fuzzy transportation problem by trapezoidal fuzzy number (a generalization of triangular fuzzy number). *Journal of fuzzy extension and application*, *2*(3), 204-225.

[54] Binato, S., Hery, W. J., Loewenstern, D. M., & Resende, M. G. (2002). A GRASP for job shop scheduling. In *Essays and surveys in metaheuristics* (pp. 59-79). Springer, Boston, MA.

[55] Aiex, R. M., Binato, S., & Resende, M. G. (2003). Parallel GRASP with path-relinking for job shop scheduling. *Parallel computing*, *29*(4), 393-430.

[56] Ruiz, R., & Stützle, T. (2007). A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. *European journal of operational research*, *177*(3), 2033-2049.

[57] Baraz, D., & Mosheiov, G. (2008). A note on a greedy heuristic for flow-shop makespan minimization with no machine idle-time. *European journal of operational research*, *184*(2), 810-813.

[58] Li, X., & Zhao, H. (2009). Greedy algorithm solution of flexible flow shop scheduling problem. *IJCSNS*, *9*(11), 177.

[59] Kahraman, C., Engin, O., Kaya, I., & Öztürk, R. E. (2010). Multiprocessor task scheduling in multistage hybrid flow-shops: a parallel greedy algorithm approach. *Applied soft computing*, *10*(4), 1293-1300.

[60] Akgöbek, Ö., Kaya, S., Değirmenci, Ü., & Engin, O. (2011). Parallel greedy metaheuristic algorithm for solving open shop scheduling problem. *NWSA eJ. New World Sci. Acad. Eng. Sci.*, *6*(1), 421-427.

[61] Pan, Q. K., & Ruiz, R. (2014). An effective iterated greedy algorithm for the mixed no-idle permutation flowshop scheduling problem. *Omega*, *44*, 41-50.

[62] Karabulut, K., & Tasgetiren, M. F. (2014). A variable iterated greedy algorithm for the traveling salesman problem with time windows. *Information sciences*, *279*, 383-395.

[63] Fernandez-Viagas, V., & Framinan, J. M. (2015). A bounded-search iterated greedy algorithm for the distributed permutation flowshop scheduling problem. *International journal of production research*, *53*(4), 1111-1123.

[64] Engin, O., Kahraman, C., & Yilmaz, M. K. (2009). A scatter search method for multiobjective fuzzy permutation flow shop scheduling problem: a real world application. In *computational intelligence in flow shop and job shop scheduling* (pp. 169-189). Springer, Berlin, Heidelberg.

[65] Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley Longman Publishing Co., Inc.

[66] Reeves, C. R. (1995). A genetic algorithm for flowshop sequencing. *Computers & operations research*, *22*(1), 5-13.

[67] Raja, M. A. Z., Sabir, Z., Mehmood, N., Al-Aidarous, E. S., & Khan, J. A. (2015). Design of stochastic solvers based on genetic algorithms for solving nonlinear equations. *Neural computing and applications*, *26*(1), 1-23.