Multi-Objective Optimization of the Dynamic and Flexible Job Shop Scheduling Problem Under Workers Fatigue Constraints

Dorsaf Aribi[2][a], Olfa Belkahla Driss[1,2][b] and Hind Bril El Haouzi[3][c]

[1] *Univ. Manouba, ESCT, Campus Universitaire Manouba, 2010, Tunisia*

[2] *Univ. Manouba, ENSI, LARIA UR22ES01, Campus Universitaire Manouba, 2010, Tunisia*

[3] *Université de Lorraine, CNRS, CRAN, F-88000 Epinal, France*

Keywords: Dynamic Scheduling, Workers' Fatigue, Flexible Job-Shop Rescheduling, New Job Insertion, Machine Breakdown, Job Cancellation , Multi-Objective Optimization, NSGA-II.

Abstract: A massive number of studies has tackled the scheduling problem, but they mainly seek to solve the classic problem by reducing the real constraints of the environment like workers' fatigue, which may lead to defective production, and the occurrence of unexpected events that makes the initial scheduling obsolete. In this paper, we study the multi-objective dynamic flexible job shop-scheduling problem under workers' fatigue constraints (DFJSP-WF) through three unexpected events: job insertion, machine breakdown and job cancellation. First, a multi-objective model is established with objectives to minimize makespan and total weighted tardiness, earliness and rejected parts due to workers' errors, which depend on workers' fatigue. Second, to deal with this model, a non-dominated sorting genetic algorithm II (NSGA-II) is adapted. Computational results are presented using three sets of well-known benchmark literature instances.

1 INTRODUCTION

Production scheduling plays an essential role in the modern manufacturing system thanks to its great potential to promote efficiency and productivity. Different complexities often arise during production such as machine breakdown and rush orders. The occurrence of these unexpected events may disrupt the original production scheduling and hinder the realization of scheduling objectives. The consideration of such events makes the scheduling process dynamic and requires a rescheduling (Liang and Yu, 2001). The main aim of scheduling/rescheduling is to allow available resources to perform a number of tasks, over a given period of time, in the best possible way to optimize one or more objectives with respecting the existing constraints of the manufacturing environment (Jain and Elmaraghy, 1997).

In such an agile environment, workers and machines complement each other to accomplish the production. At each stage, operators play an important role in leading and participating in the whole process of production, controlling and supervising all kinds of equipment, completing all kinds of work, and communicating with the environment.The human factor plays an important role in ensuring production safety and the success of the operation.

In this research, we tackle the Dynamic Flexible Job Shop Scheduling Problem under Worker Fatigue Constraint (DFJSP-WF), which is a generalization of the classical Job shop scheduling problem (JSP). To solve this problem, we apply a bi-objective optimization based on Non-dominated sorting genetic algorithm II (NSGA-II). The two objectives studied are the completion time (makespan) and the total weighted tardiness, earliness and rejected pieces. The unexpected events considered are new job arrival, job cancellation and machine breakdown. Experimental results are presented in order to prove the effectiveness of the proposed algorithm.

The rest of the paper is organized as follows. Section 2 gives a critical survey of the current literature. In Section 3, the description and formulation of the proposed DFJSP-WF are depicted in detail. The proposed NSGA-II is introduced in Section 4. In Section 5, an example is presented to clarify the subject. To evaluate the performance of the proposed algorithm, comprehensive experiments are conducted and the re-

[a] https://orcid.org/0000-0002-0872-413X
[b] https://orcid.org/0000-0003-3077-6240
[c] https://orcid.org/0000-0003-4746-5342

Aribi, D., Belkahla Driss, O. and El Haouzi, H.
Multi-Objective Optimization of the Dynamic and Flexible Job Shop Scheduling Problem Under Workers Fatigue Constraints.
DOI: 10.5220/0011663800003393
In *Proceedings of the 15th International Conference on Agents and Artificial Intelligence (ICAART 2023) - Volume 3*, pages 301-308
ISBN: 978-989-758-623-1; ISSN: 2184-433X

sults are illustrated in Section 6. Finally, concluding remarks and future works are described in Section 7.

2 RELATED WORK

2.1 Dynamic Flexible Job Shop Scheduling

Most scheduling problems are proved NP hard problems (Blazewicz et al., 1983). Clearly, dynamic scheduling is more complex and challenging to solve. Ze Tao and Xiaoxia Liu, (Tao and Liu, 2019) studied the dynamic scheduling problem of blocking job shop constrained by machines and workers based on genetic algorithm and simulated annealing algorithm (GASA). The performance of the method was examined based on two cases, to be rescheduled or minor adjusted according to their influence on the completion time, and the results show that the method proposed is effective and feasible. The study of Frutos and al. (Frutos et al., 2010) proposed a memetic algorithm based on NSGA-II to solve a Bi-objective FJSP. Authors integrated a simulated annealing stage as a local search procedure to minimize the makespan and the total operating cost. . (Chen et al., 2018) also used NSGA-II to study a multi-objective dynamic flexible job shop scheduling problem with machine breakdown to minimize makespan and total machine workload. The performance of two rescheduling strategies including transfer strategy and complete strategy is analyzed in terms of the scheduling efficiency and its stability. Earlier in 2015, (Shen and Yao, 2015) developed a multi-objective evolutionary algorithm (MOEA)-based proactive-reactive method in order to capture the dynamic and multi-objective nature of flexible job shop scheduling. Besides, Shen constructed a new mathematical model for the multi-objective dynamic flexible job shop scheduling problem (MODFJSP).

2.2 Fatigue in the Manufacturing Systems

Two resources such as workers and machines are constrained in the manufacturing process and this problem is called Dual-resource constrained job shop scheduling problem.

Dual-resource constrained scheduling has been given more and more attention in recent years (Dhiflaoui et al., 2018) (Gong et al., 2019) (Mraihi et al., 2022) (Farjallah et al., 2022). However, most of the objectives studied are profit/cost oriented. Only recently new models have been proposed in order to optimize the working conditions such as to reduce physical workload (Mossa et al., 2016). This shows an important gap in scheduling literature where the impact of scheduling decisions on human performance and health is usually ignored (Lodree et al., 2009). To fill this gap, a significant effort on modeling of human factors has to be made. We would focus on the integration of one of such factors like the human accumulated fatigue in scheduling decision-making processes.

In (Jaber et al., 2013), the authors evaluated fatigue F using the model defined by the equation 1, where R is the residual fatigue carried, λ is a fatigue exponent specifying fatigue rate, t_n is the production time of the cycle i and t_i is determined by projecting the value of $R(\tau_i)$ on the fatigue curve.

$$F_{i+1}(t) = R(\tau_i) + (1 - R(\tau_i))(1 - e^{-\lambda(t_n - t_i)}) \quad (1)$$

In (Ferjani et al., 2017), the authors associated a penalty coefficient d_j to each machine j to model the difficulty of work on that machine, such as $0 \leq dj \leq 1$. This coefficient d_j expresses the speed of fatigue accumulation. The increase of fatigue $\Delta_{i,j}$ generated during a new task on machine j, between its beginning t_i and the current time t, is expressed as follows:

$$\Delta_{i,j}(t_i, t) = (1 - F_i(t_i))(1 - e^{-d_j(t - t_i)}) \quad (2)$$

As a consequence, from time t to t_i, the level of fatigue is updated as follows:

$$F_i(t) = F_i(t_i) + \Delta_{i,j}(t_i, t) \quad (3)$$

(Zhang and Luo, 2020) had another vision. They presented a mathematical model of job fatigue time varying with time in the equation 4.

$$FT_i = \alpha FS[(1 + F_j) * \sum_{j=1}^{n} m_{ij} * mod_j],\ \forall i \in I; \forall j \in J \quad (4)$$

where I represents task set, $I = \{1, 2, 3, ...\}$; J represents the set of sequence numbers, $J = \{1, 2, 3, ...\}$; m_j represents each action element; F_j represents the fatigue index of each action; mod_{ij} represents the time required to complete each pose under the MOD method in seconds; n represents the number of occurrences of a single action to complete a task, $n = \{1, 2, 3...\}$; FS represents the fatigue state values of workers in different time periods; FT_i represents fatigue time in seconds and α represents confidence, $(0.95 < \alpha < 1)$.

3 DFJSP-WF

The DFJSP-WF considers only the flexibility of the machine. To optimize the worker's fatigue impact and system productivity, the Total Weighted Tardiness, Earliness and Rejected Parts ($TWETRP$) and makespan (C_{max}) are employed as objectives. In this section, the description and formulation of the proposed DFJSP-WF are depicted in detail.

3.1 Problem Description

The DFJSP-WF is described as follows. There is a set of n independent jobs $J = \{J_1, J_2, \dots, J_n\}$, a set of m machines $M = \{M_1, M_2, \dots, M_m\}$, and a set of m workers $W = \{W_1, W_2, \dots, W_m\}$. A job J_i has a sequence of r_i operations $\{O_{i1}, O_{i2}, \dots, O_{ir_i}\}$ to be processed one after another according to the precedence constraint. Each operation O_{ij}, namely, the jth operation of J_i, must be executed on a given machine M_k chosen from the given machine set M_{ij} with a specific processing time t_{ijk}. Each machine M_k must be operated by the given worker W_k. Workers will experience fatigue and recovery during production. Worker fatigue accumulates during a continuous period of work while alleviates during a continuous period of rest.

For the dynamic impact, we considered three events: a new job arrival, a job cancellation and a machine breakdown. Each event is characterized with an occurrence time. At each event, we are supposed to provide a rescheduling.
For the first case, a new job arrival, rescheduling is a necessary procedure for a flexible job shop when newly arrived priority jobs must be inserted into an existing schedule. In the second case, job cancellation, it is rescheduled if the remainder task is heavy, otherwise, the remainder jobs will be processed corresponding to initial scheduling. And for the last event, machine breakdown, according to (Tao and Liu, 2019), there are two different situations of machine breakdown according to the fault repairing time:

- Major fault : the fault repairing time is long $\Rightarrow$ a rescheduling is recommended
- Minor fault : the fault repairing time is short $\Rightarrow$ a rescheduling becomes not necessary because it has little effect on the completion time.

To clarify the proposed problem, the following assumptions are considered:

1. Preemption of jobs is not allowed.
2. All raw material and production resources (machine and worker) are available at time 0.
3. Processing time is deterministic and predefined.
4. The worker cannot be interrupted during processing.
5. Worker has no fatigue at time 0.
6. The processing routing of each job and the processing time of each operation are known when the job is inserted.
7. Each machine can only process one operation at a time.

Table 1: Model parameters and decision variables.

Notation	Description
n	The number of jobs
m	The number of machines/ The number of workers
r_i	Total number of operations for job J_i
O_{ij}	The jth operation of job J_i
M_{ij}	The set of compatible machines for O_{ij}
t_{ijk}	The processing time of operation O_{ij} on the machine M_k
L	A big positive number
X_{ijk}	If the operation O_{ij} is processed on M_k, $X_{ijk} = 1$; otherwise, $X_{ijk} = 0$, $i \in \{1, 2, .., n\}, j \in \{1, 2, .., r_i\}, M_p \subseteq M_{ij}$
$Y^k_{ij,i'j'}$	If the operation O_{ij} is processed adjacently before $O_{i'j'}$ on M_k, $Y^k_{ij,i'j'} = 1$; otherwise, $Y^k_{ij,i'j'} = 0$, $i, i' \in \{1, 2, .., n\}$, $j \in \{1, 2, .., r_i\}$, $j' \in \{1, 2, .., r'_i\}$, $M_p \subseteq M_{ij} \cap Mi'j'$

The used parameters and decision variables are listed in Table 1.

3.2 Mathematical Model of DFJSP-WF

The first objective is to process jobs in Just-In-Time, which means minimizing late deliveries but also avoiding early ones. We also want to consider the effect of such strategy on the worker and particularly on his reliability. This can be formulated by eq. (5):

$$TWTERP = \sum_{i=1}^{n} (\alpha_{1i} * u_i + \alpha_{2i} * e_i + \sum_{j=1}^{m} \alpha_{3j} * r_j) \quad (5)$$

where α_{1i} is the job i penalty cost per unit of time of an early delivery, α_{2i} is its penalty cost per unit of time of a late delivery, and α_{3j} is the penalty cost of rejected deliveries due to quality issues related to reference j. u_i is the tardiness period of job i, e_i is its earliness period, and r_j is the number of rejected parts due to human error after processing a job j which is modeled using eq. (6). The parameter C is for error

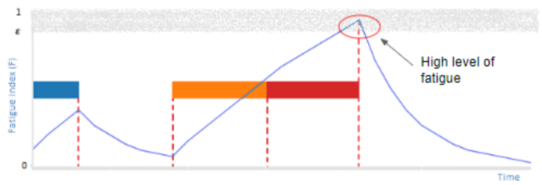

Figure 1: Fatigue evolution during jobs processing.

calibration. θ_j designates the tasks difficulty. Both C and θ_j are estimated using a mapping with data provided by the Human Error Assessment and Reduction Technique (HEART) (El Mouayni et al., 2019).

$$r_j(t) = CF_t^{\theta_j} \tag{6}$$

Following the recommendations of the literature, it is assumed that general fatigue is governed by an exponential distribution. Thus, the fatigue index between two successive period t and $t+1$ is modeled using eq. (7) :

$$F_{t+1} = w_t(1-(1-F_t)e^{-\lambda_k}) + (1-w_t)F_t e^{-\mu_k} \tag{7}$$

where F_t is a float designating the fatigue index ($0 \leq F \leq 1$) at the end of the period $t \in [0, T-1]$, T is an integer designating the scheduling horizon. λ_k is the fatigue rate when processing a job k. The parameter μ_k is the recovery rate after processing a job k. The parameter w_t designates the worker's state. It is equal to 1 if worker is busy, 0 if idle. Depending on the schedule, the worker may have some rest windows to recover from fatigue. Similarly, periods with intensive work may occur leading to excessive fatigue and error probability raise as shown in Figure 1. On the other hand, job put-off due to fatigue recovery may lead to delivery tardiness. Therefore, a trade-off should be found.

The second objective of makespan (C_{max}) is modeled as the matter of routine and the formulation is shown as eq. (8).

$$C_{max} = \max_{i,j}\{C_{ij}\}\ ,\ i \in \{1,2,\ldots,n\},\ j \in \{1,2,\ldots,m\} \tag{8}$$

In order to solve the problem in hand, a bi-objective model is formulated as follows:

$$minimize \begin{cases} f_1 = TWTERP \\ f_2 = C_{max} \end{cases} \tag{9}$$

Subject to:

$$C_{i(j+1)} \geq C_{ij} + \sum_{k=1}^{m} t_{i(j+1)k} X_{i(j+1)k}\ , \quad i \in \{1,2,..,n\}, j \in \{1,2,..,r_i\} \tag{10}$$

$$C_{i(j+1)} + (1 - Y^p_{ij,i'j'})L \geq C_{ij} + \sum_{k=1}^{m} t_{i'j'k} X_{i'j'k}\ , \quad i,i' \in \{1,2,..,n\}, j \in \{1,2,..,r_i\}, \quad j' \in \{1,2,..,r_i'\},\ p \in 1,2,..m \tag{11}$$

$$\sum_{k=1}^{m} X_{ijk} = 1\ , i \in \{1,2,..,n\}, j \in \{1,2,..,r_i\} \tag{12}$$

$$Y^p_{ij,i'j'} Y^p_{i'j',ij} = 0\ , \quad i,i' \in \{1,2,..,n\}, j \in \{1,2,..,r_i\}, \quad j' \in \{1,2,..,r_i'\}, M_p \subseteq M_{ij} \cap M_{i'j'} \tag{13}$$

$$(1 - X_{ijp} X_{i'j'p}) Y^p_{ij,i'j'} = 0\ , \quad i,i' \in \{1,2,..,n\}, j \in \{1,2,..,r_i\}, \quad j' \in \{1,2,..,r_i'\}, M_p \subseteq M_{ij} \cap M_{i'j'} \tag{14}$$

The constraint (10) guarantees the precedence relation of job operations. Constraint (11) indicates that a machine can process only one operation at a time. Constraint (12) ensures that each operation should only be processed once. Constraint (13) avoids the conflicted sequence of operations assigned to a machine. Constraint (14) is network constraint of the decision variables X and Y, which avoid the conflicted values between them.

4 NSGA-II FOR DFJSP-WF

The NSGA-II procedure is one of the popularly used EMO procedures which attempt to find multiple Pareto-optimal solutions for a multi-objective optimization problem and has the following three features: an elitist principle, an explicit diversity preserving mechanism and non-dominated solutions to be emphasized.

4.1 Coding and Decoding

Let's consider that we have to schedule 4 jobs composed respectively by $\{3,3,2,3\}$ operations. The operation sequence should be encoded in the chromosome. The individual is constituted of a random permutation of 1 to the number of operations (11 in our case). With a simple mathematical equation, each chromosome is transformed to the job number as illustrated in figure 2.

10	6	5	11	1	3	2	7	8	4	9

$\Downarrow$ Decoding

4	2	2	4	1	1	1	4	3	2	3
O_{41}	O_{21}	O_{22}	O_{42}	O_{11}	O_{12}	O_{13}	O_{43}	O_{31}	O_{23}	O_{32}

Figure 2: Coding and decoding procedure.

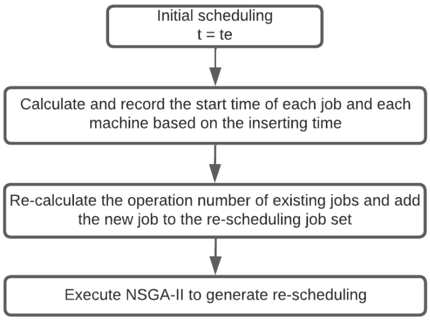

Figure 3: Re-scheduling process after new job arrival.

4.2 Evolutionary Operators of NSGA-II

Two-point crossover operator is used where two crossover points are picked randomly from the parent chromosomes. The bits in between the two points are swapped between the parent organisms. The repairment phase is to adjust the appearance time of the jobs after crossover to guarantee the feasibility of the solution.
For the mutation, partial transfer mutation operator is applied. Mutation is a genetic operator used to maintain diversity from one generation to the next. The partial transfer operator is about choosing two positions in the solution. The gene in the first position will be in the last and the genes in between are shifted with one position.

4.3 Re-Scheduling Process

In the scenario of dynamic scheduling, it is important to check which operations have been finished and which operations have not been processed or are being processed but are incomplete at time t when the dynamic events occur. The re-scheduling process for these events is described in the next three sections.

4.3.1 New Job Insertion

At an instant t_e, a new job arrive so, a re-scheduling is needed. We start by listing the unprocessed operations then we add the operations of the new job and re-execute the NSGA-II to obtain a new PF. The process is depicted in figure 3.

4.3.2 Job cancellation

When an order is canceled, we need to verify if the cancellation time is less than $2/3$ of the completion time. If it's the case then, a rescheduling is necessary. The process is shown in figure 4.

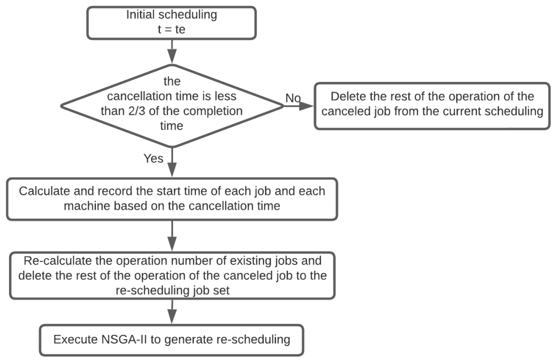

Figure 4: Re-scheduling process after a job cancellation.

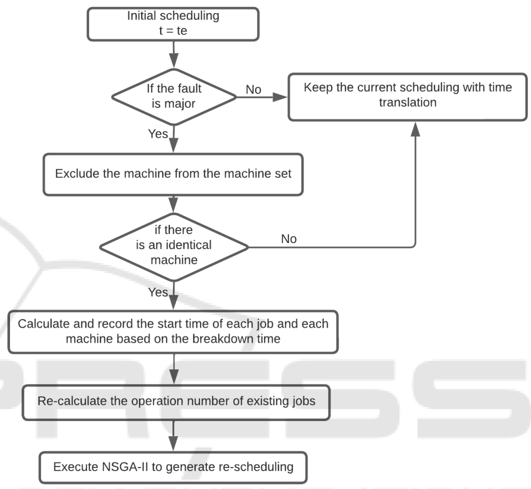

Figure 5: Re-scheduling process after a machine breakdown.

4.3.3 Machine Breakdown

When a machine breakdown occur, we first need to verify if it is rather a minor or a major fault. Second, we should verify if there is an identical machine to replace the defaulted one. If the two conditions are verified, then NSGA-II is executed and a new PF is generated. The steps details are shown in figure 5.

5 ILLUSTRATIVE EXAMPLE

In this section, an illustrative example is provided to show the efficiency of the algorithm on static and dynamic FJSP under worker fatigue constraint.

5.1 Static FJSP

To better understand the different steps of the algorithm, we would introduce an instance IP of the

Table 2: Job–machine information of the instance IP.

Job	Operations	Corresponding machine	Processing time	Due date	Penalty
J1	O11	M1,M2	3,5	20	0.24
	O12	M5	4		
	O13	M3,M4	4,6		
J2	O21	M5	10	35	0.2
	O22	M1,M2	5,7		
	O23	M3,M4	6,8		
J3	O31	M3,M4	7,9	55	0.3
	O32	M5	6		
J4	O41	M1,M2	6,10	60	0.15
	O42	M5	5		
	O43	M3,M4	7,8		

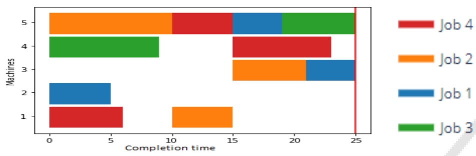

Figure 6: GANTT chart of the optimal makespan value equal to 25 for IP.

DFJSP-WF, presented in table 2 , that we will consider for the rest of this section. The proposed instance IP consists of 4 jobs, 11 operations and 5 machines. The optimal solution based on Makespan value is presented in figure 6.

5.2 Dynamic FJSP

We considered the selected solution presented in figure 7 as the initial scheduling. Three scenarios of unexpected events are studied in the following.

(1) New Job Arrival

At $t = 15$, a new order arrive. The new job data is presented in table 3. In figure 8, two solutions are represented. The first solution is selected from the Pareto front delivered by our approach based on CR (Complete Rescheduling) strategy with $C_{max} = 38$ and the second one is based on RS (Right-Shift) strategy with $C_{max} = 46$. We can obviously see that the first strategy deliver a better result.

(2) Job Cancellation

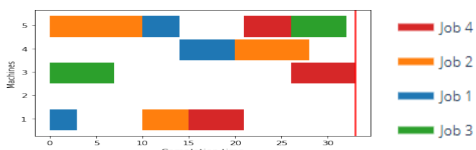

Figure 7: GANTT chart of the selected solution for rescheduling.

Table 3: New job arrival data.

Job	Operations	Corresponding machine	Processing time	Penalty	Due date
J5	O51	M3,M4	3,4		
	O52	M5	5	0.2	70
	O53	M1,M2	5,4		

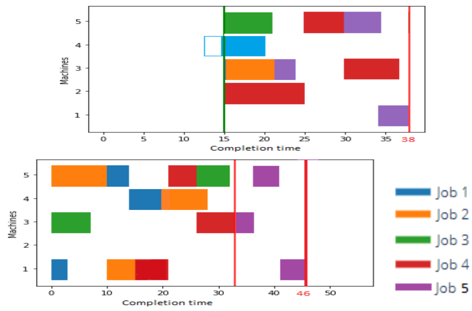

Figure 8: GANTT charts of the rescheduling after J_5 insertion with 2 strategies.

The second scenario is the cancellation of job 2 at $t = 10$ and we obtain the solution with a makespan equal to 28 that is presented in figure 9. We observe that the makespan value is lower that the obtained in the initial scheduling.

(3) Machine Breakdown

The last considered scenario is machine 1 breakdown at $t = 12$. Machine 1 and machine 2 are identical so, we can generate a rescheduling. One of the obtained solutions by our approach is presented in figure 10.

6 SIMULATION AND ANALYSIS

In the proposed algorithm, there are several input parameters that we set as follow: 1) population size $N = 40$; 2) crossover probability $\rho_c = 0.8$; and 3) mutation probability $\rho_m = 0.05$. For the calculation of fatigue rate of each worker, we consider the following parameters: machine m difficulty rate $\theta_m \in [0.01, 0.5]$; error calibration $C = 1$; fatigue rate $\lambda_i \in [0.01, 0.1]$ and recovery rate $\mu_i \in [0.1, 0.5]$ after processing job i. The effectiveness of the NSGA-II for DFJSP under fatigue constraint is analyzed by performing experiments with static FJSP benchmarks. These instances

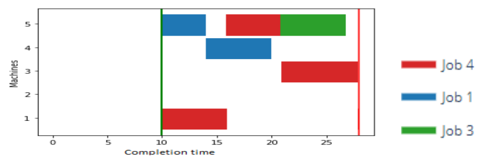

Figure 9: GANTT chart of rescheduling after job 2 cancellation.

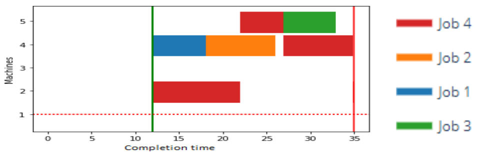

Figure 10: GANTT chart of rescheduling after machine 1 breakdown.

Table 4: Results of applying NSGA-II to benchmark instances.

Instance	n x m	f1 = Cmax		f2 = TWTERP	
		min	max	min	max
Mk01	10 x 6	68	141	210,64	329,38
Mk02	10 x 6	58	120	378,13	525,37
Mk03	15 x 8	333	550	4106,78	5025,73
Mk04	15 x 8	111	199	858,34	1161,8
Mk05	15 x 4	247	356	1055,36	1465,68
Mk06	10 x 15	183	326	904,17	1169,54
Mk07	20 x 5	265	454	2916,51	3805,47
Mk08	20 x 10	677	932	7926,86	9220,55
Mk09	20 x 10	600	857	8405,23	9648,48
Mk10	20 x 15	518	771	8988,36	10263,5
mt10c1	10 x 11	1195	2156	5317.42	7322.71
mt10cc	10 x 12	1110	2245	5849,1	8419,51
mt10x	10 x 11	1130	2118	4592,01	7116,53
mt10xx	10 x 12	1159	2052	4643.93	6781.03
mt10xxx	10 x 13	1179	2083	4676.2	6559.03
mt10xy	10 x 12	1137	2314	5034.14	7886.35
mt10xyz	10 x 13	1185	2171	3699.67	5332.56
Case1	4 x 5	16	94	1	26
Case2	10 x 7	28	197	7	62
Case3	10 x 10	22	110	16	59
Case4	10 x 15	44	188	16	94

are constructed by (Brandimarte, 1993), (Chambers and Barnes, 1996) and (Kacem et al., 2002).
Since these instances were formulated in order to minimize the makespan, we have to add a weight (penalty) and a due date for each job. We make $\alpha_{1i} = \alpha_{2i} \in U[0.01, 0.25]$. According to (Singer and Pinedro, 1998), the due date of job j is set as in eq. (15) a factor $f = 1.3$.

$$d_i = r_i + \lfloor f \sum_j \max_m (p_{ijm}) \rfloor \tag{15}$$

The extreme objective values are listed in Table 4. To demonstrate the effectiveness of the designed NSGA-II, we start by visualizing the PF obtained. As shown in figure 12, the PFs shape illustrates a good convergence and diversity. This is mainly due to the convergence mechanisms used (ranking and elitism).

In figure 11, the Hypervolume metric evolution is presented. We observe that HV is grater than 0.5 for all instances and it reach out to 0.85 which demonstrate the convergence and diversity of the solution set.

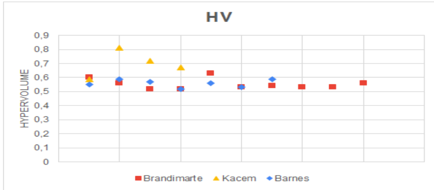

Figure 11: Hypervolume evolution relative to parameters changing.

7 CONCLUSION

The purpose of this study is to improve production effectiveness by planning the appropriate production schedule. The main objectives are to define the Dynamic Flexible Job Shop Scheduling Problem under fatigue constraint and develop an optimization method to minimize the amount of time required to manufacture a set of clients' orders with the least possible default. Job fatigue, as an integral part of human factors engineering, has an important impact on work. The excessive work pressure will not only result in the workers' resentment but will also lead to low work efficiency. For this reason, we are devoted to minimizing workers' fatigue in order to reduce its impact on production. The optimization methods must be practical in the sense that it can effectively deal with a large number of operations to be scheduled on several flexible machines. We also focused on the rescheduling after the interruption of unexpected events. The occurrence of these unexpected events may disrupt the original production scheduling and affect the realization of scheduling objectives. In this study, we considered three types of events new job arrival, job cancellation and machine breakdown. For each type, we introduced a rescheduling strategy. The goal of this study has been successfully achieved by adapting the effective multi-objective evolutionary algorithm, NSGA-II with different strategies for rescheduling. The effectiveness of this algorithm has been proved by experiments on well-known benchmarks from the literature.

The research presented in this paper paves the way for further exploration of exciting areas of research in the future. We would like to extend the problem beyond flexibility only on machines to double-flexibility on human and material resources. However, further improvements to the algorithm are needed. Since reinforcement learning (RL) has made remarkable achievements in recent years, some RL-based ap-

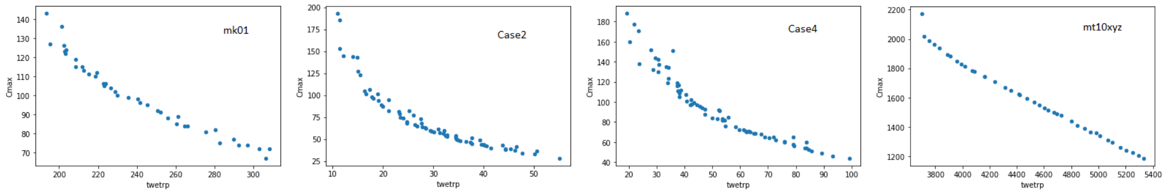

Figure 12: PF of Mk01, Case2, Case4 and mt10xyz.

proaches will be employed to enhance the performance of existing heuristic algorithms to effectively solve the DFJSP-WF.

REFERENCES

Blazewicz, J., Lenstra, J. K., and Kan, A. R. (1983). Scheduling subject to resource constraints: classification and complexity. *Discrete applied mathematics*, 5(1):11–24.

Brandimarte, P. (1993). Routing and scheduling in a flexible job shop by tabu search. *Annals of Operations research*, 41(3):157–183.

Chambers, J. B. and Barnes, J. W. (1996). Tabu search for the flexible-routing job shop problem.

Chen, C., Ji, Z., and Wang, Y. (2018). Nsga-ii applied to dynamic flexible job shop scheduling problems with machine breakdown. *Modern Physics Letters B*, 32:1840111.

Dhiflaoui, M., Nouri, H. E., and Driss, O. B. (2018). Dual-resource constraints in classical and flexible job shop problems: A state-of-the-art review. volume 126, pages 1507–1515. Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 22nd International Conference.

El Mouayni, I., Demesure, G., EL Haouzi, H., Charpentier, P., and Siadat, A. (2019). Jobs scheduling within industry 4.0 with consideration of worker's fatigue and reliability using greedy randomized adaptive search procedure. *IFAC-PapersOnLine*, 52:85–90.

Farjallah, F., Nouri, H. E., and Belkahla Driss, O. (2022). *Multi-start Tabu Agents-Based Model for the Dual-Resource Constrained Flexible Job Shop Scheduling Problem*, pages 674–686. Springer International Publishing, Cham.

Ferjani, A., Ammar, A., Pierreval, H., and Elkosantini, S. (2017). A simulation-optimization based heuristic for the online assignment of multi-skilled workers subjected to fatigue in manufacturing systems. *Computers & Industrial Engineering*, 112:663–674.

Frutos, M., Olivera, A., and Tohmé, F. (2010). A memetic algorithm based on a nsgaii scheme for the flexible job-shop scheduling problem. *Annals of Operations Research*, 181:745–765.

Gong, G., Chiong, R., Deng, Q., Han, W., Like, Z., Lin, W., and Li, K. (2019). Energy-efficient flexible flow shop scheduling with worker flexibility. *Expert Systems with Applications*, 141:112902.

Jaber, M., Givi, Z., and Neumann, W. (2013). Incorporating human fatigue and recovery into the learning–forgetting process. *Applied Mathematical Modelling*, 37(12):7287–7299.

Jain, A. K. and Elmaraghy, H. (1997). Production scheduling/rescheduling in flexible manufacturing. *International Journal of Production Research*, 35(1):281–309.

Kacem, I., Hammadi, S., and Borne, P. (2002). Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 32(1):1–13.

Liang, W. and Yu, H. (2001). Learning based dynamic approach to job-shop scheduling. In *2001 International Conferences on Info-Tech and Info-Net. Proceedings*, volume 3, pages 274–279.

Lodree, E. J., Geiger, C. D., and Jiang, X. (2009). Taxonomy for integrating scheduling theory and human factors: Review and research opportunities. *International Journal of Industrial Ergonomics*, 39(1):39–51.

Mossa, G., Boenzi, F., Digiesi, S., Mummolo, G., and Romano, V. (2016). Productivity and ergonomic risk in human based production systems: A job-rotation scheduling model. *International Journal of Production Economics*, 171:471–477.

Mraihi, T., Driss, O. B., and EL-Haouzi, H. B. (2022). *A New Variant of the Distributed Permutation Flow Shop Scheduling Problem with Worker Flexibility*, pages 587–597. Springer International Publishing, Cham.

Shen, X.-N. and Yao, X. (2015). Mathematical modeling and multi-objective evolutionary algorithms applied to dynamic flexible job shop scheduling problems. *Information Sciences*, 298:198–224.

Singer, M. and Pinedro, M. (1998). A computational study of branch and bound techniques for minimizing the total weighted tardiness in job shops. *IIE Transactions*, 30(2):109–118.

Tao, Z. and Liu, X. (2019). Dynamic scheduling of dual-resource constrained blocking job shop. In Yu, H., Liu, J., Liu, L., Ju, Z., Liu, Y., and Zhou, D., editors, *Intelligent Robotics and Applications*, pages 447–456, Cham. Springer International Publishing.

Zhang, K. and Luo, Y. (2020). Effects of worker fatigue on assembly line balancing. In *2020 IEEE 11th International Conference on Software Engineering and Service Science (ICSESS)*, pages 254–257.