

Credibility based chance constrained programming for parallel machine scheduling under linear deterioration and learning effects with considering setup times dependent on past sequences

Amir Sabripoor^a, Amirali Amirsahami^a and Rouzbeh Ghousi^{a*}

^aDepartment of Industrial Engineering, School of Engineering, Iran University of Science and Technology, Tehran, Iran

CHRONICLE ABSTRACT

Article history:

Received: December 1, 2022
Received in revised format: December 29, 2022
Accepted: March 4, 2023
Available online:
March 4, 2023

Keywords:

Parallel Machines Scheduling
Learning Effect
Deterioration effect
Past-Sequence-Dependent setup times
Augmented ϵ -constraint Method
VNS-NSGA II Hybrid Algorithm

The industry has expressed significant concern regarding the issue of parallel machines and the influence of learning and deterioration. This research investigates non-identical parallel machine scheduling, taking into account the simultaneous consideration of learning effects, deterioration, and past-sequence-dependent setup times. Due to the existence of uncertain parameters in real-world scenarios, the processing times and due dates are assumed to be triangular fuzzy numbers. A fuzzy nonlinear mathematical model with two objective functions is presented and solved using the fuzzy Chance Constraint Programming approach. The two objectives are the summation of earliness and tardiness, as well as makespan. To achieve an efficient near-optimal Pareto front for the problem, a hybrid NSGA-II and VNS multi-objective meta-heuristic is proposed and the results are discussed. Finally, the augmented ϵ -constraint method is utilized to address issues with small dimensions. The computational analysis demonstrates the effectiveness of this proposed algorithm in tackling problems, especially those with substantial dimensions.

© 2023 Growing Science Ltd. All rights reserved.

1. Introduction

The scheduling of parallel machines is a vital problem that has significant implications not only in the production environment but also in distributed computing and parallel information processing (Alchieri et al., 2018). This scheduling issue represents a critical challenge in production management and operations research due to its combinatorial nature and multiple constraints that must be considered, including machine availability, job processing times, and release dates (Alchieri et al., 2018; Meyr, 2002). In addition, real-world situations may feature additional variables, such as learning and deterioration effects, as well as machine delivery times, that render the problem even more intricate (Kim & Lee, 2021). Developing effective scheduling algorithms that take these factors into account is imperative for ensuring optimal production processes and meeting customer demand (Rostami & Sabripoor, 2022). Uncertainty about influential parameters represents one of the most fundamental issues in real-world problems and environments, where parameters such as processing times and delivery deadlines frequently exhibit a non-deterministic state. The use of fuzzy numbers can help to capture this uncertainty and ambiguity, although it is worth noting that other factors in the workplace can impact problem conditions and basic parameters. In a production environment, raw materials that are queued for processing are often subject to a “deterioration effect of jobs”, which may ultimately result in their complete loss (Abedi et al., 2020; Wang & Edwin Cheng, 2007; Wu et al., 2019). Conversely, the abilities and skills of production workers tend to improve over time, thereby accelerating the production process - an effect known as the “learning effect” (Mor et al., 2020; R. Wu et al., 2018). The setup times of machines which are sequence-dependent, constitute another critical parameter that impacts the completion time of tasks in a production system. This type of setup time is typically influenced by the conditions under which it is performed, with one of the

* Corresponding author.

E-mail address: ghousi@iust.ac.ir (R. Ghousi)

more common types being setup time dependent on the past sequence. It is essential to acknowledge that machines within the workplace may exhibit differences in their processing capabilities, stemming from variances in their specifications or brands (Liu et al., 2006; Siemieniuch et al., 2015). In addition to the discourse on workplace conditions, production goals are always a crucial consideration. These objectives can be viewed from the producer's perspective, such as minimizing the maximum completion time of tasks, or from the consumer's perspective, such as minimizing the total delay and early completion. In the literature, the latter two objectives are frequently merged into a single objective termed ET, which corresponds to the JIT concept in the production environment (Pinto et al., 2018). Based on our comprehensive analysis of resources and references, the problem of scheduling non-identical parallel machines that simultaneously considers the effects of task deterioration and learning in a fuzzy environment has yet to be investigated.

The structure of the sections is as follows: In *section 2*, the necessary basic principles for investigating the problem are presented. *Section 3*, while defining the problem, a Non-Linear Multi-Objective Mixed Integer Non-Linear Programming (MOMINLP) is first presented. Then, with the help of the credibility index and also chance-constrained programming (CCP), the presented model is made deterministic. *Section 4*, includes the development of metaheuristic hybrid algorithms for solving the problem under large dimensions. *Section 5* shows the computational results and finally, *section 6* includes conclusions and future studies.

2. Literature review

Parallel machine scheduling in deterministic environment without considering the effects of deterioration and learning has been the focus of many researchers, including (Brier & lia dwi jayanti, 2020; Prot et al., 2013; Sariçiçek & Çelik, 2011; Unlu & Mason, 2010; Vallada & Ruiz, 2011). In recent years, one paper used the SA and GA algorithms to study the problem of parallel machine scheduling in a deterministic environment (Wang et al., 2020).

Several researchers have studied this problem considering the effects of task deterioration. (Ji & Cheng, 2008; M. Liu et al., 2013) examined the problem of parallel machine scheduling with the goal of minimizing the total completion time under the condition that task deterioration follows a linear function. (Mazdeh et al., 2010) considered the problem of minimizing the sum of tardiness and the cost of machine deterioration in parallel machine scheduling. (Ruiz-Torres et al., 2013) studied the same problem under the condition that the task deterioration rate is a function of the job sequence, and solved it using SA. Additionally, (Ding et al., 2019; S. J. Yang, 2013; Zhang & Luo, 2013) investigated the problem of parallel machine scheduling under the conditions of task deterioration, and considering factors such as maintenance and rejection. In recent years, (Sekkal & Belkaid, 2020) solved the problem of parallel machine scheduling considering the effects of deterioration and resource constraints using the SA algorithm.

Considering the effect of learning alone in scheduling problems, (Lee et al., 2012) investigated the problem of uniform parallel machine scheduling, while (M. Liu, 2013) considered the joint effect of learning and delivery times of goods. (Przybylski, 2018) took into account the effect of integral-based learning, and (Expósito-Izquierdo et al., 2019) added sequence-dependent setup times. In recent years, the problem of single machine and parallel machine sequencing with learning effect has been solved and evaluated using three algorithms, SA, GSO, and SA-GSO. In another study, (Toksari & Güner, 2009) investigated the problem of scheduling identical parallel machines with the aim of minimizing the total earliness-tardiness and non-linear deterioration coefficient under deterministic environment and common due dates. Another research focuses on the parallel machine scheduling problem, aiming to minimize the total completion time under the influence of position-dependent learning and linear deterioration coefficient, which is solved using a traditional genetic algorithm. Despite numerous studies in the field of parallel machine scheduling under deterministic environments, only a few limited articles have been published in this area under uncertain conditions. (Al-Khamis & M'Hallah, 2011) solved the basic problem of scheduling tasks on parallel machines as a two-level mathematical model without considering the effect of learning and deterioration. (Peng & Liu, 2004) studied the basic problem of parallel machine scheduling under fuzzy processing times and developed three methods for modeling probabilistic problems, namely, EVP, CCP, and DCP, in a fuzzy environment with a new metric called credibility. Finally, they evaluated their proposed methods using simulation.

Recently, two articles have been published in the fuzzy environment, taking into account the effects of learning and decay of tasks and without considering the sequence-dependent setup times. The first article, written by (Rostami et al., 2015), investigates this problem for the first time. They propose an algorithm based on important features for solving optimization problems that consider the effects of learning and decay of tasks. In addition, (Arik & Toksari, 2018) also examine this problem in the fuzzy environment, considering the effects of decay and learning and solving it using a local search algorithm

According to our search and research in this field, so far no research has been conducted on scheduling parallel machines considering the learning effect and the decay effect of tasks simultaneously under conditions where processing times and delivery times are fuzzy and preparation times are dependent on the past sequence in the problem. In this article, the problem of scheduling non-identical parallel machines in a fuzzy environment with the aim of minimizing both the total earliness and tardiness (ET) and the maximum completion time (makespan) is investigated. In this problem, the learning effect and the decay effect simultaneously affect the problem conditions, which can be taken into account in the processing times of tasks. We assume that both the decay effect and the learning effect are functions of task processing priorities.

3. Mathematical model development

This section provides an overview of the problem and the fundamental principles of fuzzy programming, as well as the index used, and finally, the Augmented ϵ -constraint method is discussed.

3.1. Problem description

There are N tasks that must all be completed at time zero and failure is not allowed. In a production workshop, there are M machines that process tasks at different speeds (V_j). Each machine is operated by workers with a different skill level, represented by a negative coefficient β_j . The tasks also deteriorate over time at different rates, represented by another coefficient α_i . To start processing each task on each machine, we need to spend some preparation time, which is a sequence of past events (see (Hsu et al., 2011) for more details). Processing times and delivery deadlines are also fuzzy triangular numbers. The objective is to minimize the total earliness and tardiness (ET) and makepan. According to the standard introduced by (Graham et al., 1979), the problem can be formulated as follows:

$$Qm | LE, D, ST_{psd} | (ET, C_{max}) \tag{1}$$

where LE, D, and ST represent the effect of learning, task deterioration, and preparation time, respectively, which depend on past events. Since problem $Pm||C_{max}$ has been proven to be NP-hard by (Liao & Sheen, 2008), adding more difficult conditions to the problem still leaves it NP-hard.

3.2. Proposed Mathematical model

In this section, Fuzzy Multi-Objective Mixed Integer Non-linear Programming (FMOMINLP) is presented. Firstly, it is assumed that the sequence for each machine is divided into N hypothetical priorities denoted by r . Then, the task assignment process to these priorities is performed using the model. The model searches the solution space to ultimately reach the optimal solution. Table 1 summarizes the parameters and decision variables of the proposed model.

Table 1
Parameters and Decision Variables of FMOMINLP

Sets	i	Set of tasks	$i=1,2,\dots,N$
	k,r	Set of priority number	$k,r=1,2,\dots,N$
	j	Set of machines	$j=1,2,\dots,N$
Parameters	N	Total number of tasks	
	β_j	Learning coefficient of machine j	
	α_i	Degradation coefficient of task i	
	b_j	Machine j setup speed rate	
	\tilde{p}_i	Nominal processing time of task i	
	\tilde{d}_i	Deadline of task i	
Decision Variables	V_j	Machine j speed rate	
	\tilde{p}_{rj}	Actual processing time of priority r related to machine j	
	s_{rj}	Setup time of priority r related to machine j	
	\tilde{C}_{rj}	Completion time of priority r related to machine j	
	\tilde{C}_{max}	Maximum completion time	
	\tilde{T}_{rj}	Tardiness of priority r related to machine j	
	\tilde{E}_{rj}	Earliness of priority r related to machine j	
	x_{irj}	1 if task i is assigned to priority r related to machine j ; 0.	

MOMIP model:

$$\min \tilde{f}_1 = \sum_{r=1}^N \sum_{j=1}^M \tilde{T}_{rj} + \sum_{r=1}^N \sum_{j=1}^M \tilde{E}_{rj} \tag{2}$$

$$\min \tilde{f}_2 = \tilde{C}_{max} \tag{3}$$

S.t:

$$\tilde{p}_{rj} = \frac{1}{V_j} \left[\sum_{i=1}^N [x_{irj} \times ([\tilde{p}_i + \alpha_i \times (r - 1)] \times r^{\beta_j})] \right] \quad \forall r, j \tag{4}$$

$$s_{rj} = b_j \sum_{k=1}^{r-1} \tilde{p}_{kj} \quad \forall r, j \quad (5)$$

$$\begin{aligned} s_{1j} &= 0 & \forall j \\ \tilde{C}_{rj} &= \tilde{C}_{r-1,j} + s_{rj} + \tilde{p}_{rj} \quad \forall r, j \end{aligned} \quad (6)$$

$$\begin{aligned} \tilde{C}_{0j} &= 0 & \forall j \\ \tilde{C}_{\max} &\geq \tilde{C}_{n,j} \quad \forall j \end{aligned} \quad (7)$$

$$\begin{aligned} \tilde{T}_{rj} &\geq \left(\tilde{C}_{r,j} \times \sum_{i=1}^N x_{irj} \right) - \sum_{i=1}^N (\tilde{d}_i \times x_{irj}) & \forall r, j \\ \tilde{E}_{rj} &\geq \sum_{i=1}^N (\tilde{d}_i \times x_{irj}) - \left(\tilde{C}_{r,j} \times \sum_{i=1}^N x_{irj} \right) & \forall r, j \end{aligned} \quad (8)$$

$$\sum_{r=1}^N \sum_{j=1}^M x_{irj} = 1 \quad \forall i \quad (9)$$

$$\begin{aligned} \sum_{i=1}^N x_{i,r,j} &\leq 1 & \forall r, j \\ x_{I,r+1,j} &\leq \sum_{i=1}^N x_{irj} & \forall j, r (r \neq N), (I = 1, \dots, N), (I \neq i) \end{aligned} \quad (10)$$

$$\begin{aligned} \tilde{p}_{rj}, \tilde{C}_{rj}, \tilde{T}_{rj}, \tilde{E}_{rj}, \tilde{C}_{\max} & \quad \forall r, j \\ x_{irj} &\in \{0,1\} & \forall r, j, i \end{aligned} \quad (11)$$

Objective function (2) minimizes the total tardiness and earliness. *Objective function (3)* minimizes the maximum completion time of jobs. *Constraint (4)* is designed to calculate the actual processing times of jobs assigned to each priority on each machine. It is clear from this relationship that as the priority number increases, the impact of deterioration on each job increases and processing time increases. Also, as the value of r increases, the effect of learning plays a role and reduces the processing time of that priority. *Constraint (5)* is to calculate the sequence-dependent setup times on each machine. It is clear that on each machine, the setup time also increases with an increase in r . *Constraint (6)* is to calculate the completion time of each priority from each machine, which is equal to the completion time of the previous priority on the same machine plus the setup and actual processing times of that priority. *Constraint (7)* is to convert the Min Max objective function to the simplified Min and calculate the makespan. *Constraint (8)* calculates the tardiness and earliness of each priority. *Constraint (9)* requires that each job be assigned to only one priority of one machine. *Constraint (10)* are designed to calculate the absence of more than one job in each priority of each machine, as well as the absence of idle time on machines. That is, if a job is assigned to a priority of a machine, then no other job can be assigned to that priority from that machine, and there must be a job in its previous priority as well. Finally, *constraint (11)* defines decision variables. It should be noted that this model is a nonlinear model because it has a series of nonlinear constraints. In this model, *constraint (8)* is a nonlinear constraint due to the multiplication of two decision variables. So, the solving software does not guarantee reaching the global optimal solution.

3.3. Fuzzy Basics

Fuzzy theory was first introduced by (Zadeh, 1965). In this section, the prerequisites for fuzzy numbers are defined, which will be used in subsequent sections (Tanaka, 1992). These definitions and properties are as follows:

Proposition 1: If $\tilde{\xi}_1 = (a, b, c)$ and $\tilde{\xi}_2 = (d, e, f)$ are two triangular fuzzy numbers, then their sum is also a triangular fuzzy number (Chanas & Kasperski, 2001):

$$\tilde{\xi}_1 \oplus \tilde{\xi}_2 = (a + d, b + e, c + f) \quad (12)$$

Proposition 2: If $\tilde{\xi}$ is a fuzzy number, r is a crisp number, and Cr represents credibility, Pos represents possibility, and Nec represents necessity, then the relationship between the three measurement tools, i.e., possibility, obligation, and credibility, for an event $\tilde{\xi} \leq r$ is as follows (B. Liu & Liu, 2002):

$$Cr\{\tilde{\xi} \leq r\} = \frac{1}{2} \left(Pos\{\tilde{\xi} \leq r\} + Nec\{\tilde{\xi} \leq r\} \right) \quad (13)$$

Proposition 3: According (Zhu & Zhang, 2009) have proven, if $\lambda > 0.5$ is a triangular fuzzy number and r is a crisp number, then the product and sum of two fuzzy numbers will be less than or equal to r :

$$Cr\{\tilde{\xi} \leq r\} \geq \lambda \Leftrightarrow r \geq (2 - 2\lambda)\xi^{(2)} + (2\lambda - 1)\xi^{(3)} \tag{14}$$

where $\xi^{(2)}$ and $\xi^{(3)}$ are the middle value and right side value of the triangular fuzzy number $\tilde{\xi}$, respectively.

Proposition 4: If $A = (a_1, a_2, a_3)$ and $B = (b_1, b_2, b_3)$ are two triangular fuzzy numbers, then their distance, denoted by $d(A, B)$, is a triangular fuzzy number $C = (c_1, c_2, c_3)$ whose values c_i can be calculated using the following equations (Chen & Wang, 2008):

$$c_i = |\Gamma(A) - \Gamma(B)| + s_i \quad i = 1, 2, 3 \tag{15}$$

where in Eq. (4):

$$\begin{cases} \Gamma(A) = \frac{a_1 + 4a_2 + a_3}{6} \\ \Gamma(B) = \frac{b_1 + 4b_2 + b_3}{6} \end{cases} \tag{16}$$

$$s_i = (a_i - \Gamma(A) + b_i - \Gamma(B)) / 2 \quad i = 1, 2, 3 \tag{17}$$

3.4. Credibility-based on Chance Constrained Programming (CCP)

In this section, the model presented above is transformed into a deterministic model by using credibility measurement tools and consulting with CCP. In the model below, the concept of fuzzy chance constrained programming is used to make it deterministic (Peng & Liu, 2004). The difference with the probabilistic case is that instead of using a probability function, a credibility value function for an uncertain event is used. An important point in the model below is that when calculating ET, one should be careful to calculate $\left(\tilde{C}_{r,j} \times \sum_{i=1}^N x_{irj}\right) - \sum_{i=1}^N (\tilde{d}_i \times x_{irj})$ for tasks with a late start and $\sum_{i=1}^N (\tilde{d}_i \times x_{irj}) - \left(\tilde{C}_{r,j} \times \sum_{i=1}^N x_{irj}\right)$ for tasks with an early start. Therefore, to establish such conditions, the concept of fuzzy distance between two numbers expressed in proposition 4 has been used. In the model below, two new variables \bar{f}_1 and \bar{f}_2 are defined to model the problem with the help of CCP, which are the upper bounds of objective functions 1 and 2, respectively. Also, instead of two variables \tilde{E}_{rj} and \tilde{T}_{rj} , a new variable called $\tilde{E}\tilde{T}_{rj}$, which is a combination of these two variables, has been used. The two parameters λ_1 and λ_2 are also the confidence levels of the credibility functions of the first and second objective functions.

Credibility-based CCP model:

$$\min (\bar{f}_1, \bar{f}_2) \tag{18}$$

subject to

$$Cr\left\{\sum_{r=1}^N \sum_{j=1}^M \tilde{E}\tilde{T}_{rj} \leq \bar{f}_1\right\} \geq \lambda_1 \tag{19}$$

$$Cr\{\tilde{C}_{n,j} \leq \bar{f}_2\} \geq \lambda_2 \quad \forall j \tag{20}$$

$$\tilde{E}\tilde{T}_{rj} = d\left(\left(\tilde{C}_{r,j} \times \sum_{i=1}^N x_{irj}\right), \sum_{i=1}^N (\tilde{d}_i \times x_{irj})\right) \quad \forall r, j \tag{21}$$

Constrains

(4), (5), (6),
(9) & (10)

$$\tilde{p}_{rj}, \tilde{C}_{rj}, \tilde{C}_{max} \quad \forall r, j \quad (22)$$

$$x_{irj} \in \{0,1\} \quad \forall r, j, i \quad (23)$$

In this model, the objective function (18) minimizes the two objective functions in order. *Constraints (19) and (20)* are used to define chance constraints in fuzzy problems, considering the credibility value of the two objective functions greater than a predetermined confidence level using credibility tools and seeking the minimum values of \tilde{f}_i that satisfy these equations. These two constraints can be transformed into deterministic constraints with the help of the results in proposition 3. In *equation (21)*, the right-hand side expression represents the distance between two fuzzy numbers, i.e., $\tilde{C}_{r,j}$ and \tilde{d}_i . The distance value for these two numbers can be calculated using the equations in proposition 4, resulting in a triangular fuzzy number. It should be noted that in the above model, it is assumed that the confidence levels are values greater than 0.5, i.e., $\lambda_i > 0.5$.

3.5. Augmented ε -constraint method

In a multi-objective optimization problem, the Pareto front defines the set of solutions that achieve Pareto optimality. Thus, there is no single optimal solution, but rather a set of optimal solutions representing different trade-offs among the objectives (Jabir et al., 2015; Khoshabi et al., 2020; Petchrompo et al., 2022). In fact, this means that the performance of each objective function cannot be improved without worsening the results of at least one other objective function in the problem (Ghanbari et al., 2022; Spielhofer et al., 2022).

The ε -constraint method is a popular approach suggested and frequently employed for tackling problems with multiple objectives. (Hosseini et al., 2016; Javadi et al., 2020; Laumanns et al., 2006; Z. Yang et al., 2014). This method has some disadvantages, for example, when the number of objectives increases, Sensibility, to the initial conditions, etc (Hemmatesfe et al., 2017; Stanovov et al., 2020). becomes complicated. In order to resolve this problem (Mavrotas, 2009), introduced the augmented ε -constraint method. Numerous studies and articles provide evidence for the efficacy and enhancement resulting from this approach (Esmaili et al., 2011; G & K, 2020; Huy et al., 2023; Torabzadeh et al., 2022; Xing et al., 2019). According to this method, using lexicographic optimization, a payoff table is formed and the two-objective model presented is transformed into a new problem, which is shown in *model (24)*.

$$\begin{aligned} & \min (Z_1(x) - \text{eps} \times (s_2/r_2 + s_3/r_3 + \dots + s_p/r_p)) \\ & \text{subject to} \\ & Z_2(x) = e_2 - s_2 \\ & Z_3(x) = e_3 - s_3 \\ & \dots \\ & Z_p(x) = e_p - s_p \\ & x \in S, s_i \in R^+ \end{aligned} \quad (24)$$

It should be noted that the range of objective $Z_i(x)$ is calculated from the payoff table ($Z_i(x) \in [Z_i(x)^{min}, Z_i(x)^{max}]$), and similarly, the range of my objective is calculated from the payoff table ($r_i = Z_i(x)^{max} - Z_i(x)^{min}$). In addition, the range of objective $Z_i(x)$ should be divided by q_i, e_i^j which is calculated from *equation (25)*.

$$e_i^j = Z_i(x)^{max} - \frac{r_i}{q_i} \times j \quad j = 0, 1, \dots, q_i \quad (25)$$

Additionally, a small value of ε is considered between 10^{-6} and 10^{-3} .

4. Developed Metaheuristic Algorithms

In this section, first the developed NSGA-II algorithm stages for the problem under investigation in this paper are explained, and then the combination algorithm with Variable Neighborhood Search (VNS) will be discussed. The implementation stages of the NSGA-II algorithm can be described as follows:

I. Representation of Solutions (Genotype Space)

The type of solution representation in this paper is considered as two-row matrices with n columns, where n represents the total number of available tasks. The first row of the matrix indicates the machine number that each task (column number) is processed on, and the second row represents the processing priority of each task on that machine. For example, *Figure 1* illustrates the real space of the problem and its chromosome representation for 5 tasks and 2 parallel machines.

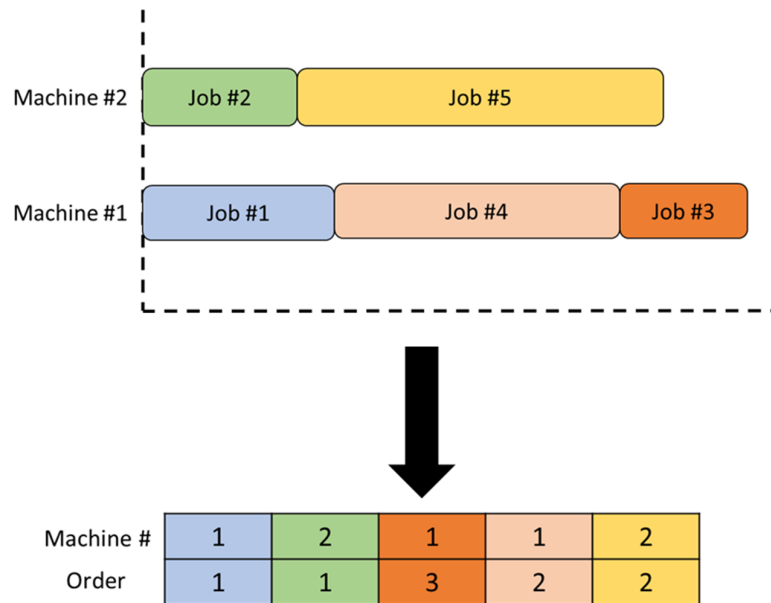


Fig. 1. Representation of solution

II. Definition of Fitness Function

To determine the value of each created solution (chromosome), a specific fitness function based on the Pareto rank and crowding distance needs to be developed. However, before that, necessary processing has been carried out to determine the precise value of each objective function due to the fuzzy nature of the objective functions. After calculating the value for each solution i (using the mathematical equations of objective functions provided in section 3.3), the crowding distance of each solution can also be calculated.

III. Implementation of Crossover Operation

In order to perform the crossover operation, a single-point crossover operator has been used. First, one of the columns of the chromosome is randomly selected, and then the columns before the selected column are taken from parent one and the columns after are taken from parent two to form the first offspring.

IV. Implementation of Mutation Operation

To perform mutation on chromosomes, first $P_{mut} * n$ columns are randomly selected (although this value is rounded up to the nearest integer). Then, the values of the selected columns' components are randomly chosen according to a uniform relationship, adopting an integer value between 1 and the maximum value of each row. $P_{mut} = 0.03$ is the value in this algorithm.

V. Genome Encoding

Given the selected representation structure, almost all problem constraints will always be satisfied. However, constraints related to the order of priority values on each machine may be violated after crossover and mutation operations. To solve this problem, repair strategy is employed where after each execution of crossover and mutation operators, infeasible solutions are repaired. The following steps are taken to repair the solutions:

Step 1: The priority values assigned to the tasks of machine j are sorted in ascending order.

Step 2: The rank of each task in the sorted arrangement of Step 1 is considered as its new priority value.

Step 3: The above procedure is applied to all machines to repair all solutions.

VI. Termination Criterion

One of the well-known conditions in literature is the condition of achieving the maximum predetermined number of repetitions, which is also considered in this paper.

VII. Combination with VNS

In this section, we present the features of the VNS algorithm for its integration with the NSGA-II algorithm. This combination can be classified as a Low-level Co-evolutionary Hybridization (LCH) combination. In this combination, after generating solutions using crossover operators, a subset of chromosomes will be influenced by the VNS algorithm with two neighborhood structures (NSS) to extract a better solution (local optimum near them). To evaluate the solutions obtained by the VNS algorithm, a one-sided intensification approach based on the concept of exploiting is used. Exploiting means that 50% of the time, VNS attempts to find solutions with the lowest primary objective function, and 50% of the time, it tries to find solutions with the lower secondary objective function. Therefore, it is expected that the optimal Pareto front will be drawn more tightly, and a better dispersion criterion will be demonstrated.

In this algorithm, two neighborhood structures are proposed as follows:

First structure: Randomly select two machines, randomly select one job on each machine, swap the two jobs with each other. Second structure: Randomly select two machines, randomly select one job on each machine, swap the two jobs with each other, randomly select another job on each machine, swap the two jobs with each other again. Also, a stopping criterion is defined by specifying a fixed number of iterations. Two options for the number of iterations, 50 and 100, are presented, and 100 is selected as the better option in the parameter settings. Moreover, considering the combination of VNS algorithm with NSGA-II algorithm, other parameters should also be defined, including: VNS algorithm start time (VNSstart), which in this study, will be called after one-third of all NSGA-II iterations, then the VNS algorithm will be invoked. The step length of the VNS algorithm (VNSstep), which means that after starting the VNS algorithm, it will be executed three times. The number of chromosomes (VNSnumber), which means that in each VNS execution, a percentage of the total number of generated offspring will be used for local improvement. This parameter is set to 15% in the parameter settings.

Proposed Hybrid Algorithm	
Inputs:	$N, P_c, P_m, P_v, Max_Gen, VNSnumber;$
1:	Generate <i>Initial_Pop</i> ;
2:	Evaluate fitness values of the <i>Initial_Pop</i> ;
3:	Assign rank base on pareto dominance sort;
4:	for $i = 1$ to Max_Gen do
5:	for $j = 1$ to $\text{round}[(P_c \times N)/2]$
6:	Select two individuals: (X_1, X_2) randomly;
7:	Apply crossover: $(X_1, X_2) \rightarrow (X'_1, X'_2)$;
8:	if $i \geq \text{round}[(P_v \times Max_Gen)]$
9:	Select $VNSnumber$ crossovered individuals randomly;
10:	for each selected individuals (X)
11:	$NSS = 1$;
12:	until $NSS \leq 2$
13:	Calculate the first and second objective function;
14:	Apply current NSS;
15:	for $k = 1$ to 100
16:	Apply local search;
17:	Save the <i>Best_Sol</i> ;
18:	endfor
19:	if <i>Best_Sol</i> is better than X ; then
20:	Replace X with <i>Best_Sol</i> ;
21:	otherwise
22:	$NSS = NSS + 1$;
23:	endif
24:	enduntil
25:	endfor
26:	endif
27:	endfor
28:	for $j = 1$ to $\text{round}[(P_m \times N)]$
29:	Select an individual: X randomly;
30:	Apply mutation: $X \rightarrow X''$;
31:	endfor
32:	Combine offsprings and parents $\{P \cup Q\}$;
33:	Assign rank based on pareto dominance sorting algorithm;
34:	Calculate the crowded distance of individual base on rank and crowded distance;
35:	endfor
36:	Output: Extract the best pareto front;
37:	end

Fig. 2. illustrates the pseudocode implementation of the VNS algorithm within the main body of NSGA-II

5. Computational results

To validate and achieve the efficiency of the mathematical model and algorithm presented in the previous section, in this section, randomly generated problems are solved using these methods to discuss and investigate the performance of the

presented methods. The reason for generating problems randomly is that the problem under study in this paper has not been investigated in the literature with all the conditions defined simultaneously, so there is no sample problem for it. In randomly generated problems, fuzzy processing times are selected for each operation randomly from uniform distribution function $P^{(2)} = U(21, 40)$ for the center number, $P^{(1)} = U(0.8, 1) \times P^{(2)}$ for the left number, and $P^{(3)} = U(1, 1.2) \times P^{(2)}$ for the right number. Fuzzy triangular delivery times are also selected randomly from the uniform distribution function $d^{(2)} = \eta \times U(21, 40)$ for the center number, $d^{(1)} = U(0.8, 1) \times d^{(2)}$ for the left number, and $d^{(3)} = U(1, 1.2) \times d^{(2)}$ for the right number, where in the equations mentioned, $\eta = U\left(\left[\frac{n}{m} - 1\right], \left[\frac{n}{m} + 1\right]\right)$ is an integer number from a uniform distribution with the specified limits. The learning rate, machine speed, and machine setup time rate are selected from the uniform distribution function $\beta_j = U(-0.4, -0.3)$, $V_j = U(0.8, 1.2)$ and $b_j = U(0.1, 0.3)$ respectively. The decay rate is selected from the uniform distribution function $\alpha_i = U(0.1, 0.3)$. In this paper, the confidence level of the first and second objective functions is considered 0.95 and 0.9, respectively. The problems created in this paper are designed in three categories of small, medium, and large sizes. The mathematical programming model is modeled and solved using GAMS software. In the NSGA-II algorithm, the number of iterations is considered 200. The mutation rate, crossover rate, and population size parameters are considered $P_m = 0.15$, $P_c = 0.9$, $N_p = 20$ respectively, after adjusting the parameters. The NSGA-II algorithm is coded using MATLAB software. The workflow is as follows: first, a problem is created, then it is solved for each level using the mathematical model, and the Pareto optimal front is obtained. Then, the same problem is solved using the NSGA-II algorithm and the Pareto optimal front is obtained. Finally, the results are compared and the efficiency of the proposed methods is investigated.

The workflow is as follows: first, a problem is created, then a mathematical model is solved for each level and the optimal Pareto front is obtained. Then the same problem is solved using the NSGA-II algorithm. It should be noted that the mathematical model does not guarantee reaching the global optimal solution due to its nonlinearity, so some points on the Pareto front obtained by the mathematical model may be dominated by the solutions obtained from the metaheuristic algorithm. Therefore, two criteria are used here to evaluate the solutions. The first criterion is the error of the Pareto points obtained by algorithm k from the optimal Pareto front obtained by another algorithm (GDk). This index is calculated using equations (23) and (24). It should be noted that in these equations, the value is only applicable to the points that are dominated by other points. In other words, d_i exists in cases where the value of d_i becomes a positive number.

$\min_j \{(f_i^k(x) - f_i^j(x)) + (f_2^k(x) - f_2^j(x))\}$ is the number of created Pareto solutions. In these formulas, $|PF_{known}|$, is the index corresponding to the points created by algorithm k and j is the index corresponding to the points created by another algorithm.

$$GD_k = \frac{\sqrt{\sum_{i=1}^n d_i^2}}{|PF_{known}|} \quad (23)$$

$$d_i = \max\{0, \min_j \{(f_i^k(x) - f_i^j(x)) + (f_2^k(x) - f_2^j(x))\}\} \quad (24)$$

The second criterion is the coverage index. If we denote the Pareto front obtained from the mathematical model by X and the Pareto front obtained from NSGA-II by Y, then the criterion C(X,Y) represents the number of points where the front Y dominates the front X.

Table 2 shows the results of comparing the mathematical model and NSGA-II for small-sized problems. The results indicate that NSGA-II has a much higher speed compared to the developed epsilon constraint in generating the Pareto front. However, in this category of problems, the mathematical programming model creates better solutions in terms of both the GD and coverage criteria. A significant point in this table is that for problems with three machines, some of the Pareto points of the metaheuristic algorithm outperform some of the Pareto points of the mathematical model (according to the value of C), indicating that the nonlinear model has failed to create the global Pareto front in this category of problems.

Except for problems with small sizes, all other problems can only reach a suitable Pareto front through NSGA-II. Therefore, to evaluate the obtained solutions using the metaheuristic approach, it is necessary to compare them with solutions that are very close to the Pareto optimal solutions. In this paper, this very good solution is obtained with the help of a large number of NSGA-II algorithm runs and a large population size, and is referred to as the best-known solution (BKS). For obtaining the BKS, the number of iterations is set to 1000 and the population size is set to 100. Table 3 shows the results of these evaluations.

Table 2

Comparison of AUGMECON and NSGA-II results

m	n	AUGMECON					NSGA-II				
		ONVG ¹	RT ²	ART ³	GD ⁴	C ⁵	ONGV	RT	ART	GD	C
2	5	13	437	33.61	4.8	1	8	5.06	0.632	5.4	1
	7	8	1038	129.75	2.1	5	8	5.54	0.692	4.5	0
	9	13	2074	159.53	0	13	10	12.68	1.268	33.1	0
3	7	13	4001	307.76	9.3	3	11	7.43	0.675	14.2	2
	9	15	6985	465.66	5.9	6	10	14.24	1.424	12.7	3

Table 3

The results of solving medium and large size problems

m	n	N _p	Running Time	ONVG	Spacing	Max (ET) -Min (ET)	Max (C max) -Min (C max)	Mean of ET	Mean of Cmax	GD	Error
4	13	20	37.62	14	0.00140	79.27	35.53	376.57	113.60	3.02	0.77%
	16	20	44.28	27	0.00111	751.35	93.93	1150.92	163.22	10.37	0.89%
	19	20	59.87	19	0.00102	299.30	48.17	800.03	144.55	7.79	0.96%
5	13	20	41.13	24	0.00138	287.93	46.71	452.40	103.27	6.54	1.41%
	16	20	54.87	21	0.00158	627.88	81.06	986.77	135.54	7.69	0.77%
	19	20	74.53	25	0.00106	489.32	72.21	940.02	145.88	13.85	1.46%
	25	20	114.41	28	0.00090	951.71	103.53	2329.48	181.09	14.54	0.62%
	30	20	349.86	25	0.00083	523.09	49.80	1703.78	174.86	11.10	0.65%
6	35	20	476.15	27	0.00111	1008.62	94.03	2985.26	203.91	19.33	0.65%
	40	20	541.23	28	0.00054	1196.96	77.01	4359.86	207.63	16.52	0.38%
7	45	20	729.23	28	0.00062	1502.42	94.45	4718.47	210.27	32.48	0.69%
	50	20	1131.16	29	0.00070	1977.74	122.80	5689.53	234.68	28.72	0.50%
8	60	20	1365.51	29	0.00030	1377.30	63.99	6832.49	226.21	27.50	0.40%
	70	20	1580.19	29	0.00064	3203.35	142.83	11350.72	263.35	48.86	0.43%
Avg.	-	-	471.43	25.21	0.00094	1019.73113	80.43319	3191.16479	179.14789	17.73592	0.76%

In Table 3, the number of population members, solution times, the number of Pareto points created by the algorithm, the spacing metric, the ranges, and average objective functions, GD metric, and finally the average errors of the generated solutions are presented. The average error metric actually calculates the ratio of GD compared to the distance between the average point of two objective functions from the coordinate origin. This metric is used because the values of GD are proportional to the values of objective functions. Therefore, if the values of objective functions are numerically large, then the value of GD also becomes large, which cannot be a suitable metric alone for comparison. The error value is calculated from *equation (25)*:

$$Error = \frac{GD}{\sqrt{(mean ET)^2 + (mean C max)^2}} \quad (25)$$

The spacing metric is also calculated using *equations (26) and (27)*, which indicate the regularity of solutions in the Pareto front. Before calculating the metric, however, the objective function values are normalized linearly to adopt values between zero and one.

$$Spacing = \sqrt{\frac{1}{|PF_{Known}| - 1} \sum_{i=1}^{|PF_{Known}|} (\bar{d} - d_i)^2} \quad (26)$$

¹Number of optimal Pareto front members created

²Running time

³Average running time

⁴Calculated for each algorithm using *equation (23)*

⁵Shows the coverage index for each algorithm

$$d_i = \min_j (|f_1^i(x) - f_1^j(x)| + |f_2^i(x) - f_2^j(x)|) \tag{27}$$

In this table, all metrics have been calculated relative to BKS. The results show that as the size of the problems increases, the solution time increases, the spacing metric decreases, and the error rate remains almost unaffected by the problem size.

Finally, by considering the VNS algorithm and running the algorithm for the case of 16 jobs and 4 machines, Table 4 compares the results with each other. As can be seen from the claim made, the dispersion value of the partial solutions in the one-sided intensification approach is higher than the normal approach.

Table 4
Comparison of the quality of results between simple and combined states

Metrics	Basic NSGA-II		Hybrid NSGA-II and VNS	
	Mean	SD	Mean	SD
Running Time (Sec)	112.85	53.19	634.76	22.74
ONVG Metric	25.67	1.15	27.67	2.31
Spacing Metric	0.00072	0.00006	0.00056	0.00008
Max (ET)-Min (ET)	709.60	17.74	799.03	50.16
Max (C max)-Min (C max)	96.54	6.73	109.54	3.49
Mean of ET	1099.58	9.73	1133.36	20.12
Mean of C max	164.10	1.34	157.39	2.56

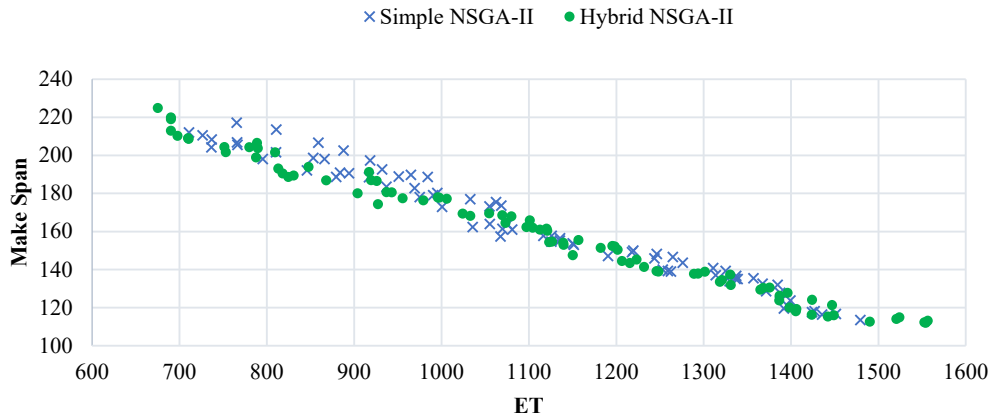


Fig. 3. Partition fronts for two simple and combined algorithms for 16 jobs and 4 machines.

Furthermore, to analyze the partial solutions created in each of the approaches, Fig. 3 shows the Pareto front resulting from the combination of all partial solutions in three repetitions. As can be seen from the figure, the one-sided intensification approach demonstrates significantly better performance, especially in the corner regions of the Pareto front.

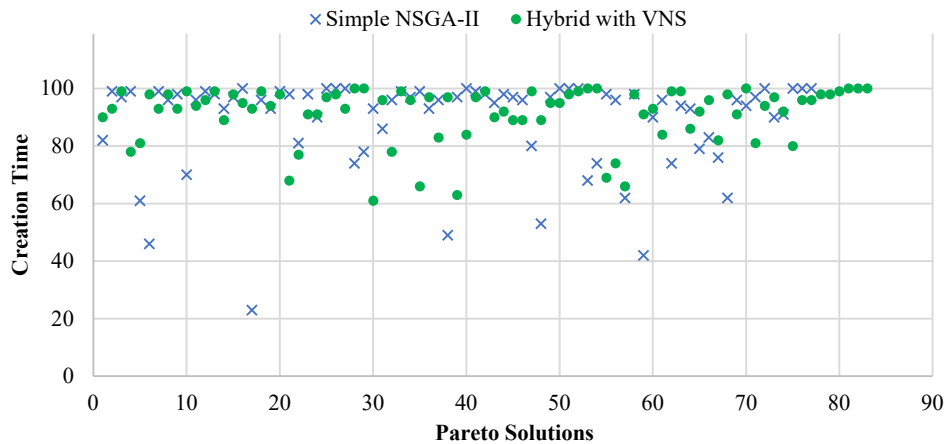


Fig. 4. Convergence process of Pareto-optimal solutions for various algorithms

On the other hand, to analyze the convergence speed, Fig. 4 is proposed. In this plot, the vertical axis represents the time of birth for each of the final Pareto solutions, and the horizontal axis represents the number of final Pareto solutions. Therefore, the more the plot tends towards a 100% ratio, the slower the convergence and the slower the stabilization of final Pareto solutions. As can be seen from the plot, a significant percentage of partial solutions in each of the approaches have a birth time of more than 80% of the algorithm's execution time, indicating a lack of early convergence in the proposed model.

6. Conclusions and future research directions

In this paper, we examine the problem of scheduling non-identical parallel machines in a fuzzy environment, considering job degradation, learning effects, and order-dependent setup time effects that simultaneously affect processing time. This article first presents a preliminary definition of the problem and a literature review. A mixed-integer nonlinear mathematical model is then proposed to represent order processing time and delivery dates using triangular fuzzy numbers. Minimizes two objective functions, total early and total late, along with maximum job completion time. The multi-objective fuzzy model is then transformed into a deterministic model using the method of Chance Constrained Programming (CCP) with a measure of reliability that has properties close to the concept of probability. Multi-objective fuzzy models are transformed into deterministic models that can be coded in solver software. Since the proposed mathematical model cannot solve large-dimensional problems in a reasonable amount of time, we use the NSGA-II metaheuristic algorithm to solve larger-sized problems. Furthermore, to enhance the power of the NSGA-II algorithm, a hybrid state of this algorithm is presented in the form of a multi-objective genetic algorithm based on the concept of one-way reinforcement.

To evaluate the proposed methods in this article, 20 completely random problems were generated in three sizes, namely small, medium, and large. For small-sized problems that both methods have the ability to solve in a reasonable time, Pareto sets were created by two methods and compared with each other. The results showed that in small-sized problems, the mathematical model creates relatively better solutions than NSGA-II, although in some problems, the mathematical model has weaker performance than NSGA-II. By solving medium and large-sized problems using NSGA-II and comparing the obtained solutions with the mathematical model, it is shown that the proposed hybrid NSGA-II algorithm outperforms the mathematical model in terms of solution quality and computational time.

For further research, considering the effect of non-linear deterioration instead of linear and measuring its difference in the problem use other methods to solve small-scale multi-objective problems. The use of another metaheuristic algorithm and their comparison with the method of solution presented in this paper.

References

- Abedi, M., Chiong, R., Noman, N., & Zhang, R. (2020). A multi-population, multi-objective memetic algorithm for energy-efficient job-shop scheduling with deteriorating machines. *Expert Systems with Applications*, 157, 113348. <https://doi.org/10.1016/j.eswa.2020.113348>
- Al-Khamis, T., & M'Hallah, R. (2011). A two-stage stochastic programming model for the parallel machine scheduling problem with machine capacity. *Computers and Operations Research*, 38(12), 1747–1759. <https://doi.org/10.1016/j.cor.2011.01.017>
- Alchieri, E., Dotti, F., & Pedone, F. (2018). Early scheduling in parallel state machine replication. *SoCC 2018 - Proceedings of the 2018 ACM Symposium on Cloud Computing*, 82–94. <https://doi.org/10.1145/3267809.3267825>
- Arik, O. A., & Toksari, M. D. (2018). Multi-objective fuzzy parallel machine scheduling problems under fuzzy job deterioration and learning effects. *International Journal of Production Research*, 56(7), 2488–2505. <https://doi.org/10.1080/00207543.2017.1388932>
- Brier, J., & lia dwi jayanti. (2020). *Sequence-dependent group scheduling problem on unrelated-parallel machines*. 21(1), 1–9. <http://journal.um-surabaya.ac.id/index.php/JKM/article/view/2203>
- 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. [https://doi.org/10.1016/S0952-1976\(01\)00011-2](https://doi.org/10.1016/S0952-1976(01)00011-2)
- Chen, S., & Wang, C. (2008). Fuzzy distance of trapezoidal fuzzy numbers and application. *International Journal of Innovative Computing, Information and Control*, 4(6), 1445–1454.
- Ding, J., Shen, L., Lü, Z., & Peng, B. (2019). Parallel machine scheduling with completion-time-based criteria and sequence-dependent deterioration. *Computers and Operations Research*, 103, 35–45. <https://doi.org/10.1016/j.cor.2018.10.016>
- Esmaili, M., Amjady, N., & Shayanfar, H. A. (2011). Multi-objective congestion management by modified augmented ϵ -constraint method. *Applied Energy*, 88(3), 755–766. <https://doi.org/10.1016/j.apenergy.2010.09.014>
- Expósito-Izquierdo, C., Angel-Bello, F., Melián-Batista, B., Alvarez, A., & Báez, S. (2019). A metaheuristic algorithm and simulation to study the effect of learning or tiredness on sequence-dependent setup times in a parallel machine scheduling problem. *Expert Systems with Applications*, 117, 62–74. <https://doi.org/10.1016/j.eswa.2018.09.041>
- G, M., & K, F. (2020). *An improved version of the augmented epsilon-constraint method (AUGMECON2) for finding the exact Pareto set in Multi-Objective Integer Programming problems*. 105034.
- Ghanbari, H., Fooeik, A. M. L., Eskorouchi, A., & Mohammadi, E. (2022). Investigating the effect of US dollar, gold and oil prices on the stock market. *Journal of Future Sustainability*, 2(3), 97–104. <https://doi.org/10.5267/j.jfs.2022.9.009>
- Graham, R. L., Lawler, E. L., Lenstra, J. K., & Kan, A. H. G. R. (1979). Optimization and approximation in deterministic sequencing and scheduling: A survey. In *Annals of Discrete Mathematics* (Vol. 5, Issue C, pp. 287–326). [https://doi.org/10.1016/S0167-5060\(08\)70356-X](https://doi.org/10.1016/S0167-5060(08)70356-X)
- Hemmat Esfe, M., Hajmohammad, H., Toghraie, D., Rostamian, H., Mahian, O., & Wongwises, S. (2017). Multi-objective

- optimization of nanofluid flow in double tube heat exchangers for applications in energy systems. *Energy*, 137, 160–171. <https://doi.org/10.1016/j.energy.2017.06.104>
- Hosseini, S. A., Amjady, N., Shafie-khah, M., & Catalão, J. P. S. (2016). A new multi-objective solution approach to solve transmission congestion management problem of energy markets. *Applied Energy*, 165, 462–471. <https://doi.org/10.1016/j.apenergy.2015.12.101>
- Hsu, C. J., Kuo, W. H., & Yang, D. L. (2011). Unrelated parallel machine scheduling with past-sequence-dependent setup time and learning effects. *Applied Mathematical Modelling*, 35(3), 1492–1496. <https://doi.org/10.1016/j.apm.2010.09.026>
- Huy, T. H. B., Dinh, H. T., & Kim, D. (2023). Multi-objective framework for a home energy management system with the integration of solar energy and an electric vehicle using an augmented ϵ -constraint method and lexicographic optimization. *Sustainable Cities and Society*, 88(July 2022), 104289. <https://doi.org/10.1016/j.scs.2022.104289>
- Jabir, E., Panicker, V. V., & Sridharan, R. (2015). Multi-objective Optimization Model for a Green Vehicle Routing Problem. *Procedia - Social and Behavioral Sciences*, 189, 33–39. <https://doi.org/10.1016/j.sbspro.2015.03.189>
- Javadi, M., Lotfi, M., Osorio, G. J., Ashraf, A., Nezhad, A. E., Gough, M., & Catalao, J. P. S. (2020). A Multi-Objective Model for Home Energy Management System Self-Scheduling using the Epsilon-Constraint Method. *Proceedings - 2020 IEEE 14th International Conference on Compatibility, Power Electronics and Power Engineering, CPE-POWERENG 2020*, 175–180. <https://doi.org/10.1109/CPE-POWERENG48600.2020.9161526>
- Ji, M., & Cheng, T. C. E. (2008). Parallel-machine scheduling with simple linear deterioration to minimize total completion time. *European Journal of Operational Research*, 188(2), 342–347. <https://doi.org/10.1016/j.ejor.2007.04.050>
- Khoshabi, P., Nejati, E., Ahmadi, S. F., Chegini, A., Makui, A., & Ghousi, R. (2020). Developing a multi-criteria decision making approach to compare types of classroom furniture considering mismatches for anthropometric measures of university students. *PLoS ONE*, 15(9 September), 1–25. <https://doi.org/10.1371/journal.pone.0239297>
- Kim, H. J., & Lee, J. H. (2021). Scheduling uniform parallel dedicated machines with job splitting, sequence-dependent setup times, and multiple servers. *Computers and Operations Research*, 126, 105115. <https://doi.org/10.1016/j.cor.2020.105115>
- L.A.Zadeh. (1965). Fuzzy Sets. *Department of Electrical Engineering and Electronics Research Laboratory, University of California, Berkeley, California A*, 353, 1. <https://doi.org/10.1061/9780784413616.194>
- Laumanns, M., Thiele, L., & Zitzler, E. (2006). An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method. *European Journal of Operational Research*, 169(3), 932–942. <https://doi.org/10.1016/j.ejor.2004.08.029>
- Lee, W. C., Chuang, M. C., & Yeh, W. C. (2012). Uniform parallel-machine scheduling to minimize makespan with position-based learning curves. *Computers and Industrial Engineering*, 63(4), 813–818. <https://doi.org/10.1016/j.cie.2012.05.003>
- Liao, L. W., & Sheen, G. J. (2008). Parallel machine scheduling with machine availability and eligibility constraints. *European Journal of Operational Research*, 184(2), 458–467. <https://doi.org/10.1016/j.ejor.2006.11.027>
- Liu, B., & Liu, Y. K. (2002). Expected value of fuzzy variable and fuzzy expected value models. *IEEE Transactions on Fuzzy Systems*, 10(4), 445–450. <https://doi.org/10.1109/TFUZZ.2002.800692>
- Liu, G., Shah, R., & Schroeder, R. G. (2006). Linking work design to mass customization: A sociotechnical systems perspective. *Decision Sciences*, 37(4), 519–545. <https://doi.org/10.1111/j.1540-5414.2006.00137.x>
- Liu, M. (2013). Parallel-machine scheduling with past-sequence-dependent delivery times and learning effect. *Applied Mathematical Modelling*, 37(23), 9630–9633. <https://doi.org/10.1016/j.apm.2013.05.025>
- Liu, M., Zheng, F., Wang, S., & Xu, Y. (2013). Approximation algorithms for parallel machine scheduling with linear deterioration. *Theoretical Computer Science*, 497, 108–111. <https://doi.org/10.1016/j.tcs.2012.01.020>
- Mavrotas, G. (2009). Effective implementation of the ϵ -constraint method in Multi-Objective Mathematical Programming problems. *Applied Mathematics and Computation*, 213(2), 455–465. <https://doi.org/10.1016/j.amc.2009.03.037>
- Mazdeh, M. M., Zaerpoor, F., Zareei, A., & Hajinezhad, A. (2010). Parallel machines scheduling to minimize job tardiness and machine deteriorating cost with deteriorating jobs. *Applied Mathematical Modelling*, 34(6), 1498–1510. <https://doi.org/10.1016/j.apm.2009.08.023>
- Meyr, H. (2002). Simultaneous lotsizing and scheduling on parallel machines. *European Journal of Operational Research*, 139(2), 277–292. [https://doi.org/10.1016/S0377-2217\(01\)00373-3](https://doi.org/10.1016/S0377-2217(01)00373-3)
- Mor, B., Mosheiov, G., & Shapira, D. (2020). Flowshop scheduling with learning effect and job rejection. *Journal of Scheduling*, 23(6), 631–641. <https://doi.org/10.1007/s10951-019-00612-y>
- Peng, J., & Liu, B. (2004). Parallel machine scheduling models with fuzzy processing times. *Information Sciences*, 166(1–4), 49–66. <https://doi.org/10.1016/j.ins.2003.05.012>
- Petchrompo, S., Wannakrairot, A., & Parlikad, A. K. (2022). Pruning Pareto optimal solutions for multi-objective portfolio asset management. *European Journal of Operational Research*, 297(1), 203–220. <https://doi.org/10.1016/j.ejor.2021.04.053>
- Pinto, J. L. Q., Matias, J. C. O., Pimentel, C., Azevedo, S. G., & Govindan, K. (2018). *Lean Manufacturing and Kaizen*. https://doi.org/10.1007/978-3-319-77016-1_2
- Prot, D., Bellenguez-Morineau, O., & Lahlou, C. (2013). New complexity results for parallel identical machine scheduling problems with preemption, release dates and regular criteria. *European Journal of Operational Research*, 231(2), 282–287. <https://doi.org/10.1016/j.ejor.2013.05.041>
- Przybylski, B. (2018). A new model of parallel-machine scheduling with integral-based learning effect. *Computers and Industrial Engineering*, 121(May), 189–194. <https://doi.org/10.1016/j.cie.2018.05.035>
- Rostami, M., Pilerood, A. E., & Mazdeh, M. M. (2015). Multi-objective parallel machine scheduling problem with job deterioration and learning effect under fuzzy environment. *Computers and Industrial Engineering*, 85, 206–215. <https://doi.org/10.1016/j.cie.2015.03.022>
- Rostami, M., & Sabriipoor, A. (2022). Optimization of green hybrid flow shop scheduling problem with considering batch delivery system. *Modern Research in Decision Making*, 7(4), 126–155. http://journal.saim.ir/article_701950.html
- Ruiz-Torres, A. J., Paletta, G., & Pérez, E. (2013). Parallel machine scheduling to minimize the makespan with sequence

- dependent deteriorating effects. *Computers and Operations Research*, 40(8), 2051–2061. <https://doi.org/10.1016/j.cor.2013.02.018>
- Sarıçiçek, I., & Çelik, Ç. (2011). Two meta-heuristics for parallel machine scheduling with job splitting to minimize total tardiness. *Applied Mathematical Modelling*, 35(8), 4117–4126. <https://doi.org/10.1016/j.apm.2011.02.035>
- Sekkal, N., & Belkaid, F. (2020). A multi-objective simulated annealing to solve an identical parallel machine scheduling problem with deterioration effect and resources consumption constraints. In *Journal of Combinatorial Optimization* (Vol. 40, Issue 3). Springer US. <https://doi.org/10.1007/s10878-020-00607-y>
- Siemieniuch, C. E., Sinclair, M. A., & Henshaw, M. J. C. (2015). Global drivers, sustainable manufacturing and systems ergonomics. *Applied Ergonomics*, 51(November), 104–119. <https://doi.org/10.1016/j.apergo.2015.04.018>
- Spielhofer, R., Schwaab, J., & Grêt-Regamey, A. (2022). How Spatial Policies Can Leverage Energy Transitions - Finding Pareto-Optimal Solutions for Wind Turbine Locations with Evolutionary Multi-Objective Optimization. *SSRN Electronic Journal*, 142(August 2022), 220–232. <https://doi.org/10.2139/ssrn.4220641>
- Stanovov, V., Akhmedova, S., & Semenkin, E. (2020). Combined fitness–violation epsilon constraint handling for differential evolution. *Soft Computing*, 24(10), 7063–7079. <https://doi.org/10.1007/s00500-020-04835-6>
- Tanaka, S. (1992). *3 Fuzzy Mathematical Programming. I.*
- Toksari, M. D., & Güner, E. (2009). Parallel machine earliness/tardiness scheduling problem under the effects of position based learning and linear/nonlinear deterioration. *Computers and Operations Research*, 36(8), 2394–2417. <https://doi.org/10.1016/j.cor.2008.09.012>
- Torabzadeh, S. A., Nejati, E., Aghsami, A., & Rabbani, M. (2022). A dynamic multi-objective green supply chain network design for perishable products in uncertain environments, the coffee industry case study. *International Journal of Management Science and Engineering Management*, 17(3), 220–237. <https://doi.org/10.1080/17509653.2022.2055672>
- Unlu, Y., & Mason, S. J. (2010). Evaluation of mixed integer programming formulations for non-preemptive parallel machine scheduling problems. *Computers and Industrial Engineering*, 58(4), 785–800. <https://doi.org/10.1016/j.cie.2010.02.012>
- Vallada, E., & Ruiz, R. (2011). A genetic algorithm for the unrelated parallel machine scheduling problem with sequence dependent setup times. *European Journal of Operational Research*, 211(3), 612–622. <https://doi.org/10.1016/j.ejor.2011.01.011>
- Wang, X., & Edwin Cheng, T. C. (2007). Single-machine scheduling with deteriorating jobs and learning effects to minimize the makespan. *European Journal of Operational Research*, 178(1), 57–70. <https://doi.org/10.1016/j.ejor.2006.01.017>
- Wang, X., Li, Z., Chen, Q., & Mao, N. (2020). Meta-heuristics for unrelated parallel machines scheduling with random rework to minimize expected total weighted tardiness. *Computers and Industrial Engineering*, 145(100), 106505. <https://doi.org/10.1016/j.cie.2020.106505>
- Wu, R., Li, Y., Guo, S., & Xu, W. (2018). Solving the dual-resource constrained flexible job shop scheduling problem with learning effect by a hybrid genetic algorithm. *Advances in Mechanical Engineering*, 10(10), 1–14. <https://doi.org/10.1177/1687814018804096>
- Wu, X., Shen, X., & Li, C. (2019). The flexible job-shop scheduling problem considering deterioration effect and energy consumption simultaneously. *Computers and Industrial Engineering*, 135(June), 1004–1024. <https://doi.org/10.1016/j.cie.2019.06.048>
- Xing, X., Yan, Y., Zhang, H., Long, Y., Wang, Y., & Liang, Y. (2019). Optimal design of distributed energy systems for industrial parks under gas shortage based on augmented ϵ -constraint method. *Journal of Cleaner Production*, 218, 782–795. <https://doi.org/10.1016/j.jclepro.2019.02.052>
- Yang, S. J. (2013). Unrelated parallel-machine scheduling with deterioration effects and deteriorating multi-maintenance activities for minimizing the total completion time. *Applied Mathematical Modelling*, 37(5), 2995–3005. <https://doi.org/10.1016/j.apm.2012.07.029>
- Yang, Z., Cai, X., & Fan, Z. (2014). Epsilon constrained method for constrained multiobjective optimization problems: Some preliminary results. *GECCO 2014 - Companion Publication of the 2014 Genetic and Evolutionary Computation Conference*, 1181–1185. <https://doi.org/10.1145/2598394.2610012>
- Zhang, M., & Luo, C. (2013). Parallel-machine scheduling with deteriorating jobs, rejection and a fixed non-availability interval. *Applied Mathematics and Computation*, 224, 405–411. <https://doi.org/10.1016/j.amc.2013.07.003>
- Zhu, H., & Zhang, J. (2009). A credibility-based fuzzy programming model for APP problem. *2009 International Conference on Artificial Intelligence and Computational Intelligence, AICI 2009*, 1, 455–459. <https://doi.org/10.1109/AICI.2009.204>

