Kurdish Studies
Feb 2024
Volume: 12, No: 2, pp.2655-2675
ISSN: 2051-4883 (Print) | ISSN 2051-4891 (Online)
www.KurdishStudies.net

Received: December 2023 Accepted: January 2024
DOI: https://doi.org/10.58262/ks.v12i2.194

Enhancement of Job Shop Planning with Sequential Dependent Setups Using a Hybrid Method

Shun-Chi Yu[1]

Abstract

Job-shop scheduling is conventionally followed in the real world; therefore, this study focuses on ceramic substrate assembly lines concerning machine idle problem. Genetic algorithm (GA) is an efficient heuristic widely used in production scheduling environment. This study integrates a new type of crossover rule and fitness function into the traditional GA with a due slot and sequence-dependent setup for constraints in the job-shop scheduling system with minimal stocks and machine idle for cost-cutting. Accordingly, simulated data verify the effectiveness and robustness of the suggested method. The results indicate that the suggested approach can potentially replace GA for solving such issues.

Keywords: *combinatorial optimization; scheduling; integer programming; genetic algorithm.*

1 Introduction

Production management is mainly applied to utilize reserves efficiently for reducing production costs and satisfying the customer. Operation management consists of various measures to reduce tardiness, earliness, inventory, machine idle, and other production costs. Nevertheless, various methods emerge for resolving scheduling problems and have irregular solutions (González et al., 2022). Amidst the producing issues pondered, job shop management becomes the most regular productive environment under real-world conditions (Qu et al., 2022).

A job shop scheduling issue, solved by a B&B (branch and bound) approach, relied on one-machine scheduling problem, and the proved effectiveness of the method, refrained the approach from prompt choices (Carlier and Pinson, 1989). Fontes et al. (2022) used the particle swarm system with a fast lower bounding step to calculate the results of the heuristic responses for larger events and verified that the suggested method can be utilized to the attempted problems. Qu et al. (2022) suggested an electromagnetism-like mechanism algorithm to diminish the longest finishing time of the many-objective model job-shop scheduling problems and showed that the suggested approach outperforms others. Meloni et al. (2022) developed an innovative net-constructed model to solve a job shop scheduling issue under uncertainties with complicated blocking restrictions and demonstrated efficient problem solving. Zhang et al. (2022) showed a two-stage method based on a nervous system to solve an elastic job-shop scheduling problem related to machine breakdown with maximum finishing time. This verification examines the fundamental job-shop production problem with due slot and sequence-dependent setup to find real and relevant solutions in practical situations. Hence, job-shop production problem is NP-hard (Carlier and Pinson, 1989; Zhang et al., 2022).

[1] MBA Program, International College, Krirk University,Bangkok, Thailand

Regarding scheduling problems with machine idle, companies are concerned about the overhead costs in tasks processed without machines while machines wait for processing tasks. Duan and Wang (2021) studies machine idle arrangement and machine speed level selection using a heuristic multi-objective non-overpassed ranking genetic approach encoded by real numbers and batch transportation rules to solve the energy-saving optimization model with machine breakdown problem, implying that idle is money for all production or service scenarios (Ilk and Shang, 2022). Feng et al. (2022) developed a systematic approach integrating energetic model, experiment design, and multi-objective optimization method to efficiently solve CNC processing time with spindle idling. In practice, avoiding excessive machine idle becomes a critical issue.

Concerning scheduling problems with setup, tasks are processed according to machine eligibility, with the sequence setup by different machines being practically regularized (Adan, 2022; Barco and Filho, 2021; Li et al., 2021). The number of setups depends on the antecedent and the next task completed on the machine, which is regarded as sequence-dependent setups, which is an essential task for a mixed job-shop problem. The sequence setup feature allows machines to process operations with sequence-preparing setup and costs in production lines (Corominas et al., 2019). Mak et al. (2022) considered extending the setup between vaccination doses to address issues in arranging for two-dose COVID-19 vaccines when the supply is limited. Tasks with a sequential setup attract more attention in the production scenario owing to the various processing preparations of machines. Prakash et al. (2022) recommended an optimum data-cutting algorithm for airliner queueing on a single runway. Appropriate setup for aircraft or vessels to prepare for journey is an important issue in real airport or harbor situations.

Concerning due slots in scheduling problems, family task shifting (Liao et al., 2012), home-to-work transportation plan (Moccia et al., 2012), courier service, food delivery, dial-a-ride, and curbside pickup service (Khoo and Bonab, 2022) with a due slot must consider more complicated conditions in the real-world. Kohar and Jakhar (2021) used a branch-and-cut algorithm to effectively solve a multiple pickup, online food delivery problem regarding due slots. Di Puglia Pugliese et al. (2022) investigated the theoretical features of the common path tour problem considering due slots and suggested a dynamic programming method to solve it. On verification, a large set of novel milestone events were obtained, indicating the efficiency of the suggested solution method. Due slots are applied worldwide in multi-depot electric buses (Gkiotsalitis et al., 2022), pickup and delivery problem (Cherkesly and Gschwind, 2022), vehicle routing and delivery robots' problem (Chen et al., 2021; Hoogeboom et al., 2021), or hospital therapist scheduling problem (Jungwirth et al., 2022). Besides, early transfers of the soft due slot generate essential performance earnings, which are allowed out of the due slot with related penalties and deducted customer satisfaction (Wölck and Meisel, 2022). Due slot is a critical constraint and, therefore, considered in the job-shop production problem.

Adan (2022) used a mixed genetic algorithm (GA) to solve a parallel machine scheduling problem considering setup and stated that GA has fewer operators and parameters, and the recommended method has a local search factor to solve larger-scale problem sets. GA is widely used in several fields such as computer network information security technology (Qiang, 2022), cement manufacturing industry (Khan and Sinha, 2022) and robot technology (Su et al., 2022). Avgerinos et al. (2022) used a logic-based benders decomposition to solve an unparallel machine production problem considering a common due slot. Leng et al. (2022) presented a multi-objective novel learning method for rescheduling problems in automatic production systems considering sequence tardiness and costs in a flow shop scheduling problem. Yuan et

al. (2022) modeled an optimization of a laser-clinched hole scheduling problem using a response face approach and GA for minimizing the thinning rate and showed that the solution substance of the recommended method was better than that of traditional methods.

GA has an outstanding capability to solve production issues. However, GA has deficiencies, leading to numerous enhancements to GA. An adjusted GA solved a limited person-to-person contact, family medical care delivery scheduling and routing issue. Intrinsic parallelism in the evolutionary procedure is also an effective recommended method (Yadav and Tanksale, 2022). Khare and Agrawal (2019) diminished the mixed flow shop production problem for diminishing earliness and tardiness using an expired date slot considering sequence- dependent setups using three methods, including whale optimization approach, mixed squirrel search method, and discrete gray wolf optimization. Their method contradicted that proposed by Pan et al. (2017), whose approach was neither worser nor equal to other methods. Zhang et al. (2022) effectively solved double-aspect disconnection line equilibrium problems regarding part feature indexes using an advanced whale optimization method. Liu et al. (2022) applied a mixed multi-verse operator approach based on differential advancement to solve a city emergency supplies dispatch methods for epidemic outbursts problem. The result indicated the recommended algorithm can solve the attempted problem with efficiency. Li and Li (2022) adapted the GA using evolutionary algorithms to search for individuals in populations to solve an interval linear bilevel programming problem. Thus, both genetic particle swarm approach (Liu et al., 2021) and mixed genetic method (Türkyılmaz et al., 2022) can perform their outstanding efficiency by enhancement.

Because of the optimal property and effectiveness of the swarm approach, the method has been widely adapted in many research fields over the decades (Kennedy et al., 2001). This approach outperforms the current algorithms and can solve a broad scope of optimal issues (Aljarah et al., 2016). The whale optimization method has been used as an evolutionary calculation skill and remarkably solved uninterrupted optimal problems using competition, population cooperation, and individual improvement (Rana et al., 2020). Whale optimization algorithm (WOA) is sufficiently competitive with other modern meta-heuristic methods (Mirjalili and Lewis, 2016). WOA uses swarm intelligence to solve scheduling problems (Rana et al. 2020); however, it has limitations, depending on the various issues that need to be resolved (Aljarah et al., 2016). Accordingly, the WOA has evolved (Baizid et al., 2015; Rana et al., 2020).

This investigation analyzes the job shop producing issue considering sequential setup, due period, and machine idle, as this issue is more prevalent and general than numerous other issues. Thus, this issue is NP-hard and valuably resolved in literature. A new method named finite-adaptive genetic algorithm (FAGA) based on GA can solve a practical issue. Furthermore, this study adapts the whale optimization method for competition. This research adapts the code-converting rule seeking the purposive set to be the heuristic schedule, which prevents it from being trapped in the local optimal solution. The suggested approach elucidates the effectiveness of retrieving an improved solution using the excellent fitness situations generated by repetitive iterations. The FAGA quickly selects the smallest earliness, tardiness, and machine idle expenses for examining the fitness function before transmutation. This formula can boost children effectively check the following schedule to choose. The elucidation expands the solution capability of the original GA and WOA and overcomes the numerous limitations of subsequent research. This study is composed of several sections. Part 1 examines documents on job shop, sequence setups, due slot, and machine idle. Part 2 proposes the integer programming formulations. Part 3 presents the suggested approach. Part 4 resolves practical problems and uses relevant methods to verify robustness and effectiveness. Part 5 offers the closure.

2 Problem Definition

The relevant definitions and notations are as follows:

Definition:

$$JS(M_z) \mid d_i = [d_i^{\mathrm{q}}, d_i^{\mathrm{r}}], set_{ijz} \mid A \times \sum_{i=1}^{n} E_i + B \times \sum_{i=1}^{n} T_i + C \times \sum_{z=1}^{m} I_z$$

where $JS(M_z)$ refers a job-shop scenario regarding z machines; d_i is the due slot; set_{ijz} implies the sequential setups on every machine; $A \times \sum_{i=1}^{n} E_i + B \times \sum_{i=1}^{n} T_i + C \times \sum_{z=1}^{m} I_z$ refers to the function for minimizing the total cost of weighted earliness, tardiness, and machine idle.

Notation

n = All number of tasks
m = All number of machines
J_i = Task of number i
J_{ij} = The j-th processing time of J_i
M_z = Machine of number z
$J_{ijz} = J_{ij}$ is treated on M_z
Op_i = Serial of operations to finish J_i
$Strt_{ijz}$ = Beginning time of J_{ijz}
set_{ijz} = Sequential subordinate setup of J_{ij} on M_z
pr_{ijz} = Processing time of J_{ijz}
d_i^{q} = Due slot lower bound of J_i
d_i^{r} = Due slot upper bound of J_i

$$Z_{ijstz} = \begin{cases} 1, & \text{if } J_{ij} \text{ is treated before } J_{st} \text{ on } M_z \\ 0, & \text{if } J_{ij} \text{ is treated after } J_{st} \text{ on } M_z \end{cases}$$

F_i = Finishing time (Flow time) of J_i
$C\max$ = Maximal finishing time of J_i
I_z = Idle (Machine idle time) of M_z
F_{ijz} = Finishing time of J_{ijz}
E_i = Finishing time before due date of J_i
T_i = Finishing time after due date of J_i
A = Weight of earliness cost
B = Weight of lateness cost
C = Weight of machine idle cost

2.1 Mathematical Model

Objective Equations

$$Min\ A\times\sum_{i=1}^{n}E_i+B\times\sum_{i=1}^{n}T_i+C\times\sum_{z=1}^{m}I_z\ ;i=1,2,..n,z=1,2,...,m \quad (1)$$

Equ.(1): this function diminishes the total cost of weighted earliness, lateness and total machine idle.

Subject to:

$$E_i=Max(d_i^q-F_i,0)\ \ ;i=1,2,..n, \quad (2)$$

Equ.(2): earliness is not the smaller of $(d_i^q-F_i)$ and 0.

$$T_i=Max(F_i-d_i^r,0)\ \ ;i=1,2,..n, \quad (3)$$

Equ.(3): lateness is not the smaller of $(F_i-d_i^r)$ and 0.

$$F_i=Max(F_{ijz})\ ;i=1,2,..n;\ j=1,2,...,Op_i;\ z=1,2,...,m \quad (4)$$

Equ.(4): finished time of J_i is the maximal finishing time on each machine.

$$C\max=Max(F_i)\ ;i=1,2,..n, \quad (5)$$

Equ.(5): maximal finished time of J_i is the maximal finishing time on each machine.

$$I_z=C\max-\sum_{i=1}^{n}\sum_{j=1}^{Op}\sum_{z=1}^{m}(F_i+set_{ijz})\ ;i=1,2,..n;j=1,2,...,Op_i;z=1,2,...,m \quad (6)$$

Equ.(6): machine idle of M_z is the maximal finishing time minus total processing time and set up.

$$F_{ijz}=Strt_{ijz}+set_{ijz}+pr_{ijz}\ ;i=1,2,..n;\ j=1,2,...,Op_i;\ z=1,2,...,m \quad (7)$$

Equ.(7): finished time of J_{ijz} is the total time of its beginning time, sequential setup and treating time.

$$F_{i(j-1)}\le Strt_{ij}\ ;i=1,2,..n;\ j=1,2,...,Op_i \quad (8)$$

Equ.(8): regarding task J_i, the beginning time of the j-th process oversteps the finishing time of the (j-1)-th serial.

$$F_{ijz}\times Z_{ijstz}\le Strt_{stz}\ ;\ i=1,2,..n;\ j=1,2,...,Op_i;\ z=1,2,...,m;\ s=0,1,...,n;\ t=1,2,...,Op_o;\ i\ne s \quad (9)$$

Equ.(9): when $Z_{ijstz}=1$, J_{ijz} is treated before J_{stz}, and thus the beginning time of J_{stz} is

larger than the finishing time of J_{ijz}; meanwhile, when $Z_{ijstz}=0$, J_{stz} is treated before J_{ijz}, and the beginning time of J_{stz} oversteps 0.

$$F_{stz}-(F_{stz}+1)\times Z_{ijstz}\le Strt_{ijz}\ ;\ i=1,2,..n;\ j=1,2,...,Op_i;\ z=1,2,...,m\ ;$$
$$s=0,1,...,n$$
$$t=1,2,...,Op_o\ ;\ i\ne s \quad (10)$$

Equ.(10): when $Z_{ijstz}=1$, J_{ijz} is treated before J_{stz}, and thus the beginning time of J_{ijz} is larger than 0; meanwhile, when $Z_{ijstz}=0$, J_{stz} is treated before J_{ijz}, and thus the beginning time of J_{ijz} is larger than the finishing time of J_{stz}.

3 Finite-Adaptive Genetic Algorithm

3.1 Procedure of FAGA

Encoding

Choong et al. (2011) developed three sets for task, phase, and machine ($y = 1, 2, \ldots, r$) identified at every phrase. This investigation adapts a finite forward scheduling rule as an array of genes in which the phases are presented in rows and tasks present in columns.

$$A_{m\times n}=\begin{bmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & & \vdots \\ a_{m,1} & \cdots & a_{m,n} \end{bmatrix} \quad (11)$$

The element $a_{m,n}(m=1,\ 2,\ ...,\ k,\ n=1,\ 2,\ ...,\ j)$ is a real number among the interval (1, M_y). $a_{m,n}$ represents which machine treats task j. The decimal refers to the treating rank; the lower the decimal, the earlier the task is treated. A coding array means a chromosome in which $k\times j$ genes are listed as follows.

$$[a_{1,1},\ a_{1,2},\ ...,\ a_{1,j},\ a_{2,1},\ a_{2,2},...,\ a_{k,j}] \quad (12)$$

In the finite forward scheduling rule, the coding array comprises three procedures to treat eight tasks and two machines.

$$A=\begin{bmatrix} 1.698 & 1.486 & 2.649 & 2.237 & 2.379 & 2.505 & 2.174 & 2.591 \\ 1.898 & 1.236 & 1.981 & 1.243 & 1.603 & 1.538 & 1.345 & 2.184 \\ 2.007 & 1.584 & 1.589 & 2.321 & 2.248 & 2.85 & 1.325 & 1.406 \end{bmatrix} \quad (13)$$

Row 1 indicates the first procedure treating conditions. $a_{3,1}=2.007$ indicates that J_1 is treated in machine 2, $a_{3,2}=1.584$ reveals that $a_{3,2}=1.584$ is treated in machine 1, and $a_{3,3}=1.589$ indicates that J_3 is treated in machine 1. The decimal $a_{3,1}$ is 0.007 not lesser than 0.589 of $a_{3,3}$, implying that $a_{3,1}$ is treated earlier than $a_{3,3}$.

Initial Population

GA's initial population is the beginning solution. Typically, targeted generation or searching

for the minimal cost can produce a beginning solution. The scope of the beginning population impacts the substance of the solutions and efficiency. An excellent scheme of utilizing the narrowest due slot of tasks for beginning solution was applied.

The Fitness Function

The function of fitness examines the grade of chromosomes and vanishes disadvantaged answers. ϑ indicates a reciprocal fixed number. The formula of fitness for the FAGA is listed below:

$$g(x) = (w_1 \cdot \sum_{i=1}^{n} E_i + w_2 \cdot \sum_{i=1}^{n} T_i + w_3 \cdot \sum_{z=1}^{m} I_z) \cdot \vartheta \quad (14)$$

Choice

An originator from comfortable chromosomes was selected with relevant mutation and crossover. Every chromosome possesses a value of fitness where the possibility for mutation and crossover is identified. This study uses a fitness value f_g selected from each chromosome C_g from fitness function *g(x)* in this step. The probability *Prg* of each chromosome in a population is calculated proportionally. The probabilities of the chromosomes sequenced by the fitness value are accepted until a randomly identified value is obtained. The last chromosome added to the sum is selected as a parent chromosome.

Crossover

This method fractionally exchanges the chosen chromosomes. Based on the problem characteristics, the two worst genes, with their due slots, are competed to fetch the worst of them; the indexing of this dot is identified as a cut dot in the parent that owns the worst gene. After this cut dot, the genes with three genes of the two parents are exchanged. The exchange rate is set to 0.8 using a two-point method if the due slots are the same. Figure 1 shows the crossover approach to solve the attempted issue.

Figure 1: Example of the Swapping Method of Swift Order Crossover.

Parent 1	**7**	**5**	**1**	**3**	**9**	**6**	**2**	**4**	**8**
Children 1				3	9	6			
Children 2				1	7	5			
Parent 2	4	8	2	1	7	5	3	9	6

After the exchange, the new chromosomes of children 1 and 2 are produced swiftly (Figure 2):

Figure 2: Example of the Swapped Chromosomes.

Parent 1	**7**	**5**	**1**	**3**	**9**	**6**	**2**	**4**	**8**
Children 1	8	7	2	3	9	6	1	5	4
Children 2	9	4	3	1	7	5	8	6	2
Parent 2	4	8	2	1	7	5	3	9	6

Mutation

Mutation targets emerged varied and multiple children. Q_m is the transmutation ratio. The suggested method yields a probability number. If the number is not above the transmutation rate, a transmutation occurs. The uniformly distributed mutation rate between 1 and 0, was given to 0.2. If the rate is 1, the suggested approach reproduces a spacing of actual numbers $(1, M_y + 1)$.

The suggested method produces queueing with fewer machine idle expenses, while certain tasks incur overhead earliness and lateness expenses. FAGA precludes extra expenses compared with the original GA.

Cease Criteria

The suggested method then ceases using the cease criteria, for example, a determined digit for convergence or cycles of iteration. The pseudo codes are as follows: ***Input:***

Population Size, p
Maximum number of iterations, MAXIMUM
Output:
Global best solution, Xbt
start
Generate finite-adaptive initial population of n chromosomes Xi (i=1, 2, ….., p)
Set iteration counter a=0
Compute the FAGA fitness value of each chromosome
while (a<MAXIMUM)
Select a pair of chromosomes from initial population based on fitness
Apply crossover operation on selected pair with crossover probability
Apply mutation of the offspring with mutation probability
Replace old population with newly generated population
Increment the current iteration a by 1.
end while
return the best solution, Xbt
end

4 Data Verifying and Analyses

This part demonstrates a resolution of a practical issue and smaller- and larger-scale virtual data verifications to verify FAGA's resolution capability. Solid proof based on the verification results shows that FAGA outperforms other approaches and the traditional GA. An example of a ceramic substrate component line in terms of just enough shipment and expense of in-production work for many project genres (Let J_1 is genre A, J_2 is genre B and J_3 is genre C). Every machine in the ceramic substrate component line has serial setups, and entire tasks should be completed in due slots. The costs of in-production work and machine idle are also essential for total production scenarios. Real-world scheduling only considers the experience of scheduling managers, which is time-consuming. Therefore, this study focuses on a ceramic substrate component line with due slot issue.

Traditionally, the scheduling manager spends much time calculating the machine idle of J_1, J_2 and J_3, which are 159, 121 and 108, respectively. Considering the optimum solution, the objective function is arduously computed as $0.5 \times (0 + 0 + 0) + 0.25 \times (0 + 0 + 112) + 0.25 \times (159 + 121 + 108) = 108.5$. Thus, the objective function of the suggested approach is quickly

computed as 0.5 × (0 + 0 + 0) + 0.25 × (0 + 28 + 0) + 0.25 × (155 + 127 + 116) = 127.5. The results present the actual cost deduction as well as effectiveness of the component line.

4.1 Data Generation

This study inspects the issue of $JS(M_z) \mid d_i = [d_i^{\mathrm{q}}, d_i^{\mathrm{r}}], set_{ijz} \mid A \times \sum_{i=1}^{n} E_i + B \times \sum_{i=1}^{n} T_i + C \times \sum_{z=1}^{m} I_z$. The posterior parameters are included in generating the verification data, where all tasks (n), all devices (m), task treating slot (pr_{ijz}), sequence-dependent setup (set_{ijz}), due slot (d_i^q, d_i^r), and weight of expense ($A, B,$ and C). Indicators relevant to FAGA and GA methods contain the following crossover rate, genetic operations, mutating rate (Pm), cycles (*loop*), and number of parents in a cycle (x). For the WOA approach, the indicators contain (a), which linearly descended from 2 to 0 when the cycles persist, random value (r), interval value (l), multiplier (b), cycles (*loop*) as well as whales in loop (x).

Based on the real-world scenario, the scope of total task numbers is a uniformly distributed range betwixt U[3, 50], and the total number of devices equals the total number of tasks. Sequence-dependent setup set_{ijz} obeying U[30, 10] is a uniformly distributed range between 30 and 10. Task treating slot pr_{ijz} obeying U[100, 50] is as well a uniformly distributed range between 100 and 50 indicating that task treating slots overstep setups but maintain reasonable scope.

To match the practical processing time, it is designed as a fuzzy set. The most likely finishing time for a task is named the most likely time, the minimal time required for a task to finish successfully is named the affirmative time; and the longest time is called the saddest time. Defuzzification must be processed before the assignment phase. Then, the fuzzy set is transformed into a crisp value, and the defuzzied procedure uses the fuzzy number by triangularly ordering situation, as suggested by (Dubois and Prade, 1988) and (Singh and Singh, 2017). The formula is as follows:

$$C(\tilde{A}) = \frac{a_1 + 2a_2 + a_3}{4} \quad (14)$$

Regarding the due slot, this investigation adopts from Zheng et al. (1993) and utilizes lateness indicator (LI) and due slot range (SR) to set up the slot date. The slot date scope is $[(1 - LI - SR/2)\sum_{i=1}^{m} P_i(1+(M-1)*0.2), (1 - LI + SR/2)\sum_{i=1}^{m} P_i(1+(M-1)*0.2)]$. Accordingly, four genres of conditions are presented with possible due slot settings (Table 1).

Table 1: Indicators of Slot Date.

Genre	LI	SR
1	0.2	0.6
2	0.2	1.0
3	0.5	0.6
4	0.5	1.0

As regards the weights of expense, the real-world business decision making considers design. Companies' weights of expense differ according to their superiority. More earliness weight of expense restricts firms from early finishing of tasks and deducts inventory costs; more delay weight of expense restricts from late shipment and satisfies consumer demands; a lower machine idle weight of expense restricts from expense of in-production work. This investigation uses several weight situations: $C_1 = 0.50$, $C_2 = 0.25$, $C_3 = 0.25$; $C_1 = 0.25$, $C_2 = 0.50$, $C_3 = 0.25$; $C_1 = 0.25$, $C_2 = 0.25$, $C_3 = 0.50$. The benefit of using three situations, rather

than only one, is that all practical situations can be totally assessed.

The parameters used in the suggested method are verified in the pre-verification phase (Table 2 and 3). The entire verifications are encoded in Lingo® 18.0 and Java® scripts and executed on a laptop with Intel® Core® i7-1165G7, 2.8GHz, 2.8GHz, 512GB SSD, 16GB DDR4 SDRAM. Accordingly, the outputs are analyzed and compared.

Table 2: Indicators of WOA.

Indicator	a	r	l	b	*loop*	x
Value	1	0.8	0.2	0.6	100	10

Table 3: Indicators of the Suggested Method.

Indicator	genetic operations	crossover rate	mutation rate	*Pm*	*loop*	x
Value	2	0.1	0.2	0.1	100	10

4.2 Data Verification for Smaller-Scale Issues

Effectiveness Verification

Due to the overtreating time, smaller-scale issues are indicated as issues from four to nine tasks. Therefore, this investigation uses three distinct weight situations and four due slot genres. Those settings intersect mutually, generating 12 conditions for each condition of the entire number of tasks. Every design is checked by GA, integer programming (IP), FAGA, and WOA for 30 arbitrarily formed issues. Computer processing times (in seconds) and the average solutions show the effectiveness and robustness of the suggested FAGA method (Table 4).

Table 4: Effectiveness Verification Outcome of Smaller-Scale Issues (A=0.50, B=0.25, C=0.25)

Genre	n	m	IP		GA		FAGA		WOA	
			Avg.	CPU time	Avg.	CPU time	Avg.	CPU time	Avg.	CPU time
1	4	4	324.79	1.816	355.17	0.087	331.69	0.086	354.33	0.082
	5	5	585.27	18.284	616.80	0.092	603.66	0.093	605.54	0.093
	6	6	867.67	37.642	1271.17	0.093	984.98	0.095	1239.41	0.094
	7	7	1462.35	50.543	1824.18	0.093	1625.56	0.094	1738.41	0.085
	8	8	2071.68	8378.415	2869.51	1.346	2517.08	1.335	2543.35	1.345
	9	9	2768.11	14606.918	3508.14	2.465	3185.09	2.427	3422.21	2.463
2	4	4	289.62	2.044	338.42	0.083	311.19	0.081	311.98	0.077
	5	5	522.10	3.879	561.80	0.091	533.52	0.087	556.86	0.086
	6	6	875.02	18.668	1234.93	0.097	1052.35	0.090	1199.15	0.096
	7	7	1496.72	337.239	1942.93	0.105	1661.20	0.095	1656.36	0.106
	8	8	1895.04	10502.344	2479.05	1.214	2137.23	1.204	2381.37	1.215
	9	9	2975.69	13470.519	3389.37	2.028	3065.12	2.009	3455.05	2.003
3	4	4	383.14	2.560	404.21	0.090	395.18	0.084	403.51	0.087
	5	5	722.01	5.573	783.89	0.096	752.46	0.087	770.50	0.093
	6	6	946.68	55.438	1310.65	0.091	1039.69	0.088	1286.51	0.085
	7	7	1568.09	754.585	2038.63	0.096	1817.92	0.084	1973.78	0.096
	8	8	2184.51	8055.084	2855.38	1.227	2435.89	1.220	2511.51	1.232
	9	9	2717.28	9275.630	3460.08	2.321	3258.72	2.310	3392.31	2.322
4	4	4	329.12	2.731	377.00	0.084	351.75	0.080	355.51	0.083
	5	5	581.31	8.041	634.94	0.093	628.00	0.091	635.55	0.103
	6	6	1032.27	154.469	1315.49	0.094	1121.31	0.090	1311.52	0.094
	7	7	1353.31	564.756	2078.20	0.106	1615.66	0.111	1906.08	0.110
	8	8	2322.14	5262.902	2827.55	1.338	2566.20	1.318	2763.09	1.330
	9	9	2707.95	10141.202	3445.91	2.339	2870.29	2.336	3130.09	2.350

Based on the due slot genres and weightage of expense situations, the computer processing times of GA, FAGA, and WOA are under three seconds without distinct disparity. This study

confirms that GA, FAGA, and WOA display similar remarkable resolution effectiveness in verifications of smaller-scale issues. This investigation examined whole due slot genres with relevant weightage of expense situations but has presented only one table owing to space constraint.

The average resolutions of FAGA outperform those of GA and WOA. About the improving percent ratio of FAGA, the effectiveness improving ratio is computed below.

Effectiveness of improving ratio: $(1 - \frac{Outcome_{FAGA} - Outcome_{IP}}{Outcome_{GA} - Outcome_{IP}}) \times 100\%$

$(1 - \frac{Outcome_{FAGA} - Outcome_{IP}}{Outcome_{WOA} - Outcome_{IP}}) \times 100\%$

Table 5: Effectiveness of Improving Percentage of FAGA vs GA in Smaller-Scale Issues.

Data Type	n	m	$A = 0.50,\ B = 0.25,\ C = 0.25$	$A = 0.25,\ B = 0.50,\ C = 0.25$	$A = 0.25,\ B = 0.25,\ C = 0.50$
1	4	4	77.27%	32.97%	32.94%
	5	5	41.67%	31.24%	88.47%
	6	6	70.93%	39.10%	79.55%
	7	7	54.89%	15.41%	84.36%
	8	8	44.17%	33.28%	60.85%
	9	9	43.65%	74.81%	92.02%
2	4	4	55.80%	38.40%	89.18%
	5	5	71.23%	31.79%	84.12%
	6	6	50.73%	55.66%	14.71%
	7	7	63.14%	46.52%	75.21%
	8	8	58.53%	41.21%	76.63%
	9	9	78.38%	86.21%	77.09%
3	4	4	42.87%	55.31%	84.89%
	5	5	50.80%	48.41%	58.38%
	6	6	74.45%	32.38%	63.86%
	7	7	46.90%	27.52%	93.41%
	8	8	62.53%	40.45%	66.66%
	9	9	27.11%	22.67%	78.46%
4	4	4	52.74%	69.90%	61.32%
	5	5	12.95%	67.05%	64.78%
	6	6	68.56%	48.10%	63.35%
	7	7	63.81%	17.51%	59.32%
	8	8	51.71%	78.23%	55.71%
	9	9	78.00%	73.02%	70.04%
Avg. Improvement			55.95%	46.13%	69.80%
Total Improvement					57.30%

Tables 5 and 6 present the effectiveness of improving ratio of the target values between FAGA, GA and WOA in smaller-scale issues. The total improving percentage ratio is 57.30%. and 45.51%, respectively, in Tables 9 and 10. The results indicate that FAGA outperforms GA and WOA.

Table 6: Effectiveness of Improving Percentage of FAGA vs WOA in Smaller-Scale Issues.

Genre	n	m	$A = 0.50,\ B = 0.25,\ C = 0.25$	$A = 0.25,\ B = 0.50,\ C = 0.25$	$A = 0.25,\ B = 0.25,\ C = 0.50$
1	4	4	77.27%	8.50%	13.11%
	5	5	41.67%	12.29%	85.89%
	6	6	70.93%	14.93%	63.41%
	7	7	54.89%	12.89%	66.30%
	8	8	44.17%	13.14%	25.28%
	9	9	43.65%	53.77%	88.71%
2	4	4	55.80%	15.32%	37.18%
	5	5	71.23%	9.24%	76.69%
	6	6	50.73%	17.38%	10.14%
	7	7	63.14%	25.57%	10.82%
	8	8	58.53%	37.80%	45.42%
	9	9	78.38%	79.64%	77.42%
3	4	4	42.87%	54.28%	70.49%
	5	5	50.80%	18.00%	53.37%
	6	6	74.45%	28.79%	47.32%
	7	7	46.90%	6.00%	98.73%
	8	8	62.53%	35.00%	38.67%
	9	9	27.11%	6.81%	53.50%
4	4	4	52.74%	72.84%	21.91%
	5	5	12.95%	56.93%	61.33%
	6	6	68.56%	34.15%	37.79%
	7	7	63.81%	13.98%	10.00%
	8	8	51.71%	63.55%	21.98%
	9	9	78.00%	72.08%	55.36%
Avg. Improvement			55.95%	31.79%	48.78%
Total Improvement					45.51**%**

Robustness Verification

An arbitrarily produced issue is examined 30 cycles testing FAGA, GA, and WOA with three weightage of expense situations regarding from four to nine tasks. Integer programming is listed one time to contrast the outcomes. This investigation computes the best, worst, standard deviation, average resolution, and computer processing time (in seconds) in every condition.

The robustness analysis of FAGA confirms the effectiveness verification in smaller-scale issues. The entire computer processing times are resolved under three seconds without vary much from that of GA and WOA. This investigation verifies that FAGA can resolve the issue with efficiency.

The best resolutions of FAGA are close to IP solution in robustness verification. Besides, FAGA prevails over GA and WOA regarding the best, worst, and mean solutions. To verify even more how FAGA performs better than GA and WOA regarding robustness, this investigation computes the robustness improving percentage listed below.

Robustness improving percentage: $(1 - \frac{\mathrm{Result}_{\sigma^2(FAGA)}}{\mathrm{Result}_{\sigma^2(GA)}}) \times 100\%$ $(1 - \frac{\mathrm{Result}_{\sigma^2(FAGA)}}{\mathrm{Result}_{\sigma^2(WOA)}}) \times 100\%$

Table 7: Robustness Improving Percentage of FAGA vs GA in Smaller-Scale Issues.

Genre	n	m	$A = 0.50,\ B = 0.25,\ C = 0.25$	$A = 0.25,\ B = 0.50,\ C = 0.25$	$A = 0.25,\ B = 0.25,\ C = 0.50$
1	4	4	12.78%	42.11%	19.38%
	5	5	8.28%	12.71%	16.58%
	6	6	23.01%	7.33%	8.79%
	7	7	6.38%	31.00%	13.45%
	8	8	38.17%	37.18%	13.63%
	9	9	1.58%	5.21%	3.86%
2	4	4	7.22%	32.64%	25.72%
	5	5	11.24%	9.00%	22.07%
	6	6	19.92%	10.43%	10.24%
	7	7	2.59%	10.33%	13.34%
	8	8	18.70%	25.22%	10.05%
	9	9	25.79%	15.63%	9.57%
3	4	4	6.70%	44.18%	22.01%
	5	5	9.71%	19.81%	22.96%
	6	6	7.37%	16.33%	20.77%
	7	7	9.02%	25.64%	20.18%
	8	8	26.74%	9.10%	9.85%
	9	9	4.31%	16.11%	17.72%
4	4	4	7.33%	35.91%	29.23%
	5	5	5.56%	24.79%	22.91%
	6	6	3.82%	7.61%	21.87%
	7	7	9.60%	18.96%	20.89%
	8	8	10.68%	9.80%	10.94%
	9	9	13.12%	5.41%	14.72%
Avg. Improvement			12.07%	19.68%	16.70%
Total Improvement					16.15%

Tables 7 and 8 present the outputs of the robustness of FAGA compared with the other approaches.

Table 8: Robustness Improving Percentage of FAGA vs WOA in Smaller-Scale Issues.

Genre	n	m	$A = 0.50,\ B = 0.25,\ C = 0.25$	$A = 0.25,\ B = 0.50,\ C = 0.25$	$A = 0.25,\ B = 0.25,\ C = 0.50$
1	4	4	21.28%	41.09%	20.17%
	5	5	8.96%	10.83%	19.42%
	6	6	7.81%	6.51%	14.47%
	7	7	3.32%	27.99%	20.12%
	8	8	13.23%	37.36%	10.60%
	9	9	2.34%	10.53%	6.03%
2	4	4	5.70%	30.92%	32.35%
	5	5	5.00%	13.10%	20.89%
	6	6	4.85%	8.19%	13.39%
	7	7	12.82%	10.53%	11.97%
	8	8	1.69%	17.83%	10.04%
	9	9	8.28%	19.00%	15.39%
3	4	4	9.88%	39.50%	25.68%
	5	5	13.74%	9.60%	30.36%
	6	6	3.52%	16.27%	24.38%
	7	7	1.63%	27.93%	19.15%
	8	8	7.95%	9.49%	12.32%
	9	9	6.86%	17.18%	15.14%
4	4	4	5.82%	35.79%	33.06%
	5	5	6.83%	21.96%	25.05%
	6	6	2.34%	9.55%	19.34%
	7	7	5.04%	16.30%	23.85%
	8	8	11.73%	11.31%	8.55%
	9	9	4.76%	13.83%	14.57%
Avg. Improvement			7.31%	19.27%	18.59%
Total Improvement					15.06%

Tables 7 and 8 indicate that the robustness improving percentages are affirmative and

remarkable in every verification set. The total improving percentages are 16.15%. and 15.06%, individually, in Tables 7 and 8. These outputs obviously present enhancements of FAGA relevant to GA and WOA in resolving smaller-scale scheduling issues.

4.3 Data Verification for Larger-scale Issues

In real-world task arrangement issues, the competition between the larger-scale issues of the sum of machines and tasks and the smaller-scale issues ought to be verified. Therefore, complying with the smaller-scale verification, this investigation as well examined larger-scale issues. According to the pre-verification, this investigation recognizes that as the sum of tasks increases, the computer processing time for solving IP also increases quickly. Given 10 tasks, the computer processing time of integer programming exceeds 150 hours, and the best solution remains unverified. Because of this restriction, integer programming solutions are neglected in the verification of larger-scale issues. This investigation compares the resolving efficiency of FAGA with that of GA and WOA in larger-scale issues.

Effectiveness Verification

The number of larger-scale issues ranges from 20 to 60. Four due slot situations, crossed with three weight-of-expense conditions and combined with all tasks, may establish 60 larger-scale issues. Besides, 30 instances are arbitrarily produced and verified using FAGA, GA and WOA for every instance. The mean resolution and computer processing time (in seconds) are presented to compare the efficiency and effectiveness of FAGA, GA and WOA.

Based on these due date genres with weightage of expense situations, no distinct disparities present betwixt the computer processing times of FAGA, GA and WOA. This outcome indicates that FAGA, GA and WOA solve similar issues efficiently.

The entire mean outcomes of FAGA overstep those of GA and WOA. Consequently, the effectiveness improving percentages are computed and presented in Tables 9 and 10 to indicate the improvement in effectiveness of FAGA.

Table 9: Effectiveness Improving Percentage of FAGA vs GA in Larger-Scale Issues.

Genre	*n*	*m*	$A = 0.50,\ B = 0.25,\ C = 0.25$	$A = 0.25,\ B = 0.50,\ C = 0.25$	$A = 0.25,\ B = 0.25,\ C = 0.50$
1	20	20	12.92%	11.70%	13.90%
	30	30	14.55%	20.29%	5.96%
	40	40	17.64%	15.36%	18.61%
	50	50	14.45%	16.40%	7.99%
	60	60	5.47%	14.31%	5.98%
2	20	20	16.27%	10.40%	21.40%
	30	30	16.36%	13.28%	9.86%
	40	40	17.67%	16.11%	7.81%
	50	50	10.83%	19.86%	8.33%
	60	60	20.76%	10.41%	5.98%
3	20	20	10.83%	17.29%	18.07%
	30	30	12.52%	23.55%	17.67%
	40	40	8.28%	7.15%	18.06%
	50	50	14.24%	16.21%	15.58%
	60	60	11.94%	22.41%	7.56%
4	20	20	10.42%	6.79%	18.44%
	30	30	15.89%	18.69%	7.97%
	40	40	13.90%	13.35%	16.08%
	50	50	16.24%	29.93%	10.30%
	60	60	19.34%	20.80%	16.76%
Avg. Improvement			14.03%	16.21%	12.61%
Total Improvement					14.28%

Table 10: Effectiveness Improving Percentage of FAGA vs WOA in Larger-Scale Issues.

Genre	*n*	*m*	$A = 0.50,\ B = 0.25,\ C = 0.25$	$A = 0.25,\ B = 0.50,\ C = 0.25$	$A = 0.25,\ B = 0.25,\ C = 0.50$
1	20	20	12.65%	9.00%	8.54%
	30	30	7.05%	14.59%	7.01%
	40	40	2.21%	16.56%	11.86%
	50	50	6.04%	2.81%	5.06%
	60	60	23.49%	8.28%	8.83%
2	20	20	10.63%	14.12%	8.46%
	30	30	6.86%	9.52%	10.40%
	40	40	3.25%	12.32%	12.26%
	50	50	8.63%	13.17%	6.10%
	60	60	52.32%	7.60%	5.83%
3	20	20	10.43%	9.22%	10.36%
	30	30	8.21%	21.69%	5.63%
	40	40	7.70%	13.27%	12.89%
	50	50	33.57%	11.29%	9.49%
	60	60	-17.59%	8.69%	7.19%
4	20	20	10.29%	4.39%	9.00%
	30	30	13.48%	5.87%	11.67%
	40	40	22.05%	8.61%	12.96%
	50	50	7.86%	32.20%	4.08%
	60	60	5.84%	14.08%	6.27%
Avg. Improvement			11.75%	11.86%	8.70%
Total Improvement					10.77 %

Tables 9 and 10 reveal that in any due date types and weight-of-expense situations, FAGA obviously oversteps GA and WOA. The total improving percentages are 14.28% and 10.77%, separately, in the larger-scale issue verification shows that FAGA can efficiently solve issues in either the smaller-scale or larger-scale. The outcome proves the enhancement achieved by FAGA in smaller-scale and larger-scale issues.

Robustness Verification

Regarding robustness verifications of larger-scale issues, an instance is randomly produced for every issue, and 30 repetitive verifications are executed by GA, FAGA, and WOA, separately. The best, worst, average solutions, standard deviation, and computer processing time computed by a repetitive method are contrasted.

Accordingly, robustness verification indicates that there is no definite disparity in the resolving time among GA, FAGA, and WOA. Among these issues, FAGA fares better best, worst, mean resolution, and lesser standard error than those of GA and WOA. Thus, this investigation verifies that FAGA can solve problems more robustly than GA and WOA.

Particularly, the mean resolutions of FAGA approximate the best resolutions than worst ones. Owing to the characteristic of the FAGA method, descendants are developed to ferret out a better chromosome in the narrowest due date when searching for solutions. Based on this approach, a better chromosome is more possibly to be selected. Besides, in rare situations, offspring are not easily entrapped in the local optimum resolution and stay in the worst resolutions regions, allowing them to search for a better response when seeking solutions.

According to the rarity and severity of this outcome, the worst resolution constantly generates greater deviation from the average than the best resolution.

Referring to the verification of a smaller-scale issue, robustness improving percentages are still computed for larger-scale issues (Tables 11 and 12).

Table 11: Robustness Improving Percentage of FAGA vs GA in Larger-Scale Issues.

Genre	**n**	**m**	$A = 0.50,\ B = 0.25,\ C = 0.25$	$A = 0.25,\ B = 0.50,\ C = 0.25$	$A = 0.25,\ B = 0.25,\ C = 0.50$
1	20	20	5.84%	26.53%	35.11%
	30	30	20.25%	31.93%	26.62%
	40	40	16.29%	19.01%	22.73%
	50	50	11.65%	39.76%	8.08%
	60	60	10.66%	23.53%	19.03%
2	20	20	14.38%	11.23%	15.80%
	30	30	26.26%	16.63%	15.77%
	40	40	28.64%	19.21%	24.51%
	50	50	33.60%	24.64%	17.24%
	60	60	11.95%	13.86%	18.69%
3	20	20	46.34%	12.56%	15.30%
	30	30	28.75%	22.94%	26.81%
	40	40	21.10%	20.16%	20.38%
	50	50	27.46%	30.88%	16.21%
	60	60	4.98%	26.78%	23.18%
4	20	20	21.19%	21.27%	22.44%
	30	30	24.95%	36.00%	22.88%
	40	40	25.82%	6.92%	31.30%
	50	50	18.52%	31.16%	31.22%
	60	60	16.44%	9.43%	19.59%
Avg. Improvement			20.75%	22.22%	21.65%
Total Improvement					21.54 **%**

Table 12: Robustness Improving Percentage of FAGA vs GA in Larger-Scale Issues.

Genre	n	m	$A = 0.50,\ B = 0.25,\ C = 0.25$	$A = 0.25,\ B = 0.50,\ C = 0.25$	$A = 0.25,\ B = 0.25,\ C = 0.50$
1	20	20	22.99%	2.21%	25.97%
	30	30	18.20%	21.53%	8.21%
	40	40	7.07%	13.51%	6.09%
	50	50	9.27%	22.66%	15.36%
	60	60	-2.90%	7.61%	3.45%
2	20	20	1.35%	2.33%	7.17%
	30	30	24.33%	10.21%	11.27%
	40	40	19.88%	19.57%	4.93%
	50	50	8.78%	18.76%	15.17%
	60	60	12.02%	1.75%	23.28%
3	20	20	29.87%	6.30%	11.73%
	30	30	27.58%	14.33%	25.68%
	40	40	20.11%	19.77%	11.56%
	50	50	6.59%	34.08%	4.55%
	60	60	6.96%	11.84%	10.85%
4	20	20	2.19%	8.47%	8.37%
	30	30	7.93%	17.47%	30.05%
	40	40	18.26%	9.80%	15.70%
	50	50	18.00%	16.29%	11.95%
	60	60	14.04%	11.04%	9.98%
Avg. Improvement			13.62%	13.48%	13.07%
Total Improvement					13.39 **%**

Tables 11 and 12 reveal that for all verification issues, FAGA solves issues more robustly than the traditional GA and WOA. The total enhancements of robustness are 21.54% and 13.39%, individually, and the proof indicates that FAGA can solve issues more robustly than traditional GA and WOA in larger-scale task shop production issues. The outcome refers to the smaller-scale robustness verification and verifies the FAGA's robustness.

5 Conclusion

This investigation verifies the task shop production issue, which is known for its complicacy. The verified issue can be presented as $JS(M_z) \mid d_i = [d_i^{\mathrm{q}}, d_i^{\mathrm{r}}], set_{ijz} \mid \mathrm{A} \times \sum_{i=1}^{n} E_i + \mathrm{B} \times \sum_{i=1}^{n} T_i + \mathrm{C} \times \sum_{z=1}^{m} I_z$. The problem involves due slot and sequence-dependent setup in a task shop scheduling problem, wherein the objective function is to diminish all weights-of-expense from earliness, lateness, and machine idle. Studies have revealed that GA and WOA have a remarkable ability to solve task shop production issues, yet their robustness and effectiveness are finite.

This study develops and arranges FAGA to resolve the task shop production issue, and then compares with the IP, GA and WOA resolutions. Based on the verification outcomes and analyses, this investigation concludes as follows:

1. The total effectiveness of improving percentage of objective value in smaller-scale issue verifying by FAGA versus GA is 57.30%, if either FAGA or GA are contrasted with IP, and 45.51% if compared with WOA. In a larger-scale issue verifying FAGA versus GA is 14.28% if both FAGA and GA are compared and 10.77% in a larger-scale problem verification if both FAGA and WOA are contrasted. This shows that FAGA obviously oversteps GA and WOA regarding effectiveness.
2. The total robustness improving percentage in the smaller-scale issue verification of FAGA versus GA is 16.15%, and 15.06%, when contrasted with WOA. In a larger-scale issue, verification by FAGA versus GA is 21.54%, and 13.39% when compared with WOA. This confirms that FAGA's robustness outperforms that of GA and WOA.
3. Under the conditions of this study, FAGA has excellent solving efficiency with fewer overhead costs of earliness, lateness and machine idle compared with GA and WOA.

The limitation of this study is its focus on task shop scheduling issue. Future studies can seek other topics such as flow shop or open shop production issue. Future research can also adopt other heuristics to solve the attempted problems. Overall, this study reveals that FAGA oversteps GA and WOA regarding either robustness or effectiveness. FAGA has the potential to substitute for GA and WOA as an outstanding hybrid method for such issues, in response to both real-world scenarios and future studies.

Data Availability Statement

The data used to support the findings of this investigation are included within the article.

Disclosure Statement

The authors confirm that there is no conflict of interest to declare for this publication.

Acknowledgments

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors. The authors would like to thank the editor-in-chief, guest

editors and anonymous reviewers for their comments that help improve the quality of this work.

Funding

This work was not supported by any organization.

References

Adan, J. (2022). A hybrid genetic algorithm for parallel machine scheduling with setup times. Journal of Intelligent Manufacturing, 33, 2059–2073. https://doi.org/10.1007/s10845-022-01959-4

Aljarah, I., Faris, H. & Mirjalili, S. (2016). Optimizing connection weights in neural networks using the whale optimization algorithm. Soft Computing, 22, 1–15.

Avgerinos, I., Mourtos, I., Vatikiotis, S. & Zois, G. (2022). Scheduling unrelated machines with job splitting, setup resources and sequence dependency. International Journal of Production Research, 1–23.

Baizid, K., Yousnadj, A., Meddahi, A., Chellali, R. & Iqbal, J. (2015). Time scheduling and optimization of industrial robotized tasks based on genetic algorithms. Robotics and Computer-Integrated Manufacturing, 34, 140–150.

Barco, C. F. & Filho, M. G. (2021). The effect of transfer lot size on manufacturing lead time: a stochastic analysis. European Journal of Industrial Engineering, 15(6), 825–851.

Carlier, J. & Pinson, E. (1989). An algorithm for solving the job-shop problem. Management Science, 5, 146–176.

Cherkesly, M. & Gschwind, T. (2022). The pickup and delivery problem with time windows, multiple stacks, and handling operations. European Journal of Operational Research, 301(2), 647–666.

Chen, C., Demir, E. & Huang, Y. (2021). An adaptive large neighborhood search heuristic for the vehicle routing problem with time windows and delivery robots. European Journal of Operational Research, 294(3), 1164–1180.

Choong, F., Phon-Amnuaisuk, S. & Alias, M. Y. (2011). Metaheuristic methods in hybrid flow shop scheduling problem. Expert Systems with Applications, 38(9), 10787–10793.

Corominas, A., García-Villoria, A., González, N. A. & Pastor, R. (2019). A multistage graph-based procedure for solving a just-in-time flexible job-shop scheduling problem with machine and time-dependent processing costs. Journal of the Operational Research Society, 70(4), 620–633.

Di Puglia Pugliese, L., Ferone, D., Festa, P. & Guerriero, F. (2022). A generalized shortest path tour problem with time windows. Computational Optimization and Applications, 1–22.

Duan, J. & Wang, J. (2021). Energy-efficient scheduling for a flexible job shop with machine breakdowns considering machine idle time arrangement and machine speed level selection. Computers and Industrial Engineering, 161, 107677.

Dubois, D. & Prade, H. (1988). Representation and combination of uncertainty with belief functions and possibility measures. Computational Intelligence, 4(3), 244–264.

Feng, C., Guo, H., Zhang, J., Huang, Y. & Huang, S. (2022). A systematic method of optimization of machining parameters considering energy consumption, machining time, and surface roughness with experimental analysis. The International Journal of Advanced Manufacturing Technology, 119(11), 7383–7401.

Fontes, D. B. M. M., Mahdi Homayouni, S. & Goncalves, J. F. (2022). A Hybrid Particle Swarm Optimization and Simulated Annealing Algorithm for the Job Shop Scheduling Problem with Transport Resources. European Journal of Operational Research, 306(3), 1140–1157. doi: https://doi.org/10.1016/j.ejor.2022.09.006

Gkiotsalitis, K., Iliopoulou, C. & Kepaptsoglou, K. (2022). An exact approach for the multi-depot electric bus scheduling problem with time windows. European Journal of Operational Research, 306(1), 189–206. https://doi.org/10.1016/j.ejor.2022.07.017

González, M. A., Rasconi, R. & Oddi, A. (2022). Metaheuristics for multiobjective optimization in energy-efficient job shops. Engineering Applications of Artificial Intelligence, 115, 105263.

Jungwirth, A., Desaulniers, G., Frey, M. & Kolisch, R. (2022). Exact branch-price-and-cut for a hospital therapist scheduling problem with flexible service locations and time-dependent location capacity. INFORMS Journal on Computing, 34(2), 1157–1175.

Kennedy, J., Eberhart, R. C. & Shi, Y. (2001). Swarm intelligence, San Francisco, Morgan Kaufmann Publishers.

Khan, M. N. & Sinha, A. K. (2022). Development of a sustainable supply chain network for the cement manufacturing industry using real-coded genetic algorithm. Soft Computing, 1–21.

Khare, A. & Agrawal, S. (2019). Scheduling hybrid flowshop with sequence-dependent setup times and due windows to minimize total weighted earliness and tardiness. Computers and Industrial Engineering, 135, 780–792.

Hoogeboom, M., Adulyasak, Y., Dullaert, W. & Jaillet, P. (2021). The robust vehicle routing problem with time window assignments. Transportation Science, 55(2), 395–413.

Ilk, N. & Shang, G. (2022). The impact of waiting on customer-instigated service time: Field evidence from a live-chat contact center. Journal of Operations Management, 68(5), 487–514.

Kohar, A. & Jakhar, S. K. (2021). A capacitated multi pickup online food delivery problem with time windows: a branch-and-cut algorithm. Annals of Operations Research, 1–22.

Khoo, T. S. & Bonab, M. B. (2022). Solving multi-objective pickup and delivery with time windows using mediocre evolutionary distributed microservices re-optimization algorithm. Applied Soft Computing, 109128.

Leng, J., Wang, X., Wu, S., Jin, C., Tang, M., Liu, R., Vogl, A. & Liu, H. (2022). A multi-objective reinforcement learning approach for resequencing scheduling problems in automotive manufacturing systems. International Journal of Production Research, 1–20.

Li, H. & Li, H. (2022). GA/PD: a genetic algorithm based on problem decomposition for solving interval linear bilevel programming problems. Engineering Optimization, 1–16.

Li, Y., Li, X., Gao, L., Zhang, B., Pan, Q. K., Tasgetiren, M. F. & Meng, L. (2021). A discrete artificial bee colony algorithm for distributed hybrid flowshop scheduling problem with sequence-dependent setup times. International Journal of Production Research, 59(13), 3880–3899.

Liao, C. J., Chao, C. W. & Chen, L. C. (2012). An improved heuristic for parallel machine weighted flowtime scheduling with family set-up times. Computers & Mathematics with Applications, 63(1), 110–117.

Liu, H., Sun, Y., Pan, N., Li, Y., An, Y. & Pan, D. (2022). Study on the optimization of urban emergency supplies distribution paths for epidemic outbreaks. Computers & Operations Research, 146, 105912.

Liu, M., Liu, Z., Chu, F., Dolgui, A., Chu, C. & Zheng, F. (2022). An optimization approach for multi-echelon supply chain viability with disruption risk minimization. Omega, 112, 102683.

Liu, Z., Wang, J., Zhang, C., Chu, H., Ding, G. & Zhang, L. (2021). A hybrid genetic-particle swarm algorithm based on multilevel neighbourhood structure for flexible job shop scheduling problem. Computers & Operations Research, 135, 105431.

Mak, H. Y., Dai, T. & Tang, C. S. (2022). Managing two-dose COVID-19 vaccine rollouts with limited supply. Production and Operations Management, 31(12), 4424–4442. https://doi.org/10.1111/poms.13862

Meloni, C., Pranzo, M. & Samà, M. (2022). Evaluation of VaR and CVaR for the makespan in interval valued blocking job shops. International Journal of Production Economics, 247, 108455.

Mirjalili, S. & Lewis, A. (2016). The whale optimization algorithm. Advances in Engineering Software, 95, 51–67.

Moccia, L., Cordeau, J. F. & Laporte, G. (2012). An incremental tabu search heuristic for the generalized vehicle routing problem with time windows. Journal of the Operational Research Society, 63(2), 232–244. https:// doi.org/ 10. 1057/ jors. 2011. 25

Pan, Q. K., Gao, L., Li, X. Y. & Gao, K. Z. (2017). Effective metaheuristics for scheduling a hybrid flowshop with sequence-dependent setup times. Applied Mathematics and Computation, 303, 89–112.

Prakash, R., Desai, J. & Piplani, R. (2022). An optimal data-splitting algorithm for aircraft sequencing on a single runway. Annals of Operations Research, 309(2), 587–610.

Qiang, X. J. (2022). Computer application under the management of network information security technology using genetic algorithm. Soft Computing, 1–6.

Qu, M., Zuo, Y., Xiang, F. & Tao, F. (2022). An improved electromagnetism-like mechanism algorithm for energy-aware many-objective flexible job shop scheduling. The International Journal of Advanced Manufacturing Technology, 119(7), 4265–4275.

Rana, N., Latiff, M. S. A., Abdulhamid, S. I. M. & Chiroma, H. (2020). Whale optimization algorithm: a systematic review of contemporary applications, modifications and developments. Neural Computing and Applications, 32(20), 16245–16277.

Singh, N. & Singh, S. B. (2017). A modified mean gray wolf optimization approach for benchmark and biomedical problems. Evolutionary Bioinformatics, 13, 1–28. 1176934317729413.

Su, J., Jiang, C. & Li, Y. (2022). Multi-source and multi-objective path planning based on genetic optimized long short-term memory neural network model. The International Journal of Advanced Manufacturing Technology, 1–11.

Türkyılmaz, A., Senvar, O., Ünal, İ. & Bulkan, S. (2022). A hybrid genetic algorithm based on a two-level hypervolume contribution measure selection strategy for bi-objective flexible job shop problem. Computers & Operations Research, 141, 105694.

Wölck, M. & Meisel, S. (2022). Branch-and-price approaches for real-time vehicle routing with picking, loading, and soft time windows. INFORMS Journal on Computing, 34(4), 2192–2211. https://doi.org/10.1287/ijoc.2021.1151

Yadav, N. & Tanksale, A. (2022). An integrated routing and scheduling problem for home healthcare delivery with limited person-to-person contact. European Journal of Operational Research, 303, 1100–1128.

Yuan, H., Pan, C., Song, L., Zhao, G. & Zheng, C. (2022). Modeling and optimization of laser shock hole-clinching using response surface methodology and genetic algorithm. The International Journal of Advanced Manufacturing Technology, 1–16.

Zhang, G., Lu, X., Liu, X., Zhang, L., Wei, S. & Zhang, W. (2022). An effective two-stage algorithm based on convolutional neural network for the bi-objective flexible job shop scheduling problem with machine breakdown. Expert Systems with Applications, 203, 117460.

Zhang, Y., Zhang, Z., Guan, C. & Xu, P. (2022). Improved whale optimisation algorithm for two-sided disassembly line balancing problems considering part characteristic indexes. International Journal of Production Research, 60(8), 2553–2571.

Zheng, W. X., Nagasawa, H. & Nishiyama, N. (1993). Single-machine scheduling for minimizing total cost with identical, asymmetrical earliness and tardiness penalties. International Journal of Production Research, 31, 1611–1620.