Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 9970988, 12 pages
https://doi.org/10.1155/2022/9970988

Research Article

The Dynamic Flow Shop Scheduling Problem with JIT Philosophy and Common Due Date

Wen-Tso Huang,[1] **Ping-Shun Chen**,[2] **Gary Yu-Hsin Chen**,[3] **and Jr-Fong Dang**[4]

[1]*Department of Business Administration, Chung Yuan Christian University, Chung Li District, Taoyuan City 320314, Taiwan*
[2]*Department of Industrial and Systems Engineering, Chung Yuan Christian University, Chung Li District, Taoyuan City 320314, Taiwan*
[3]*Department of Logistics Management, National Kaohsiung University of Science & Technology, Yanchao District, Kaohsiung City 82445, Taiwan*
[4]*Undergraduate Program of Intellectual Creativity Engineering, National Chung Hsing University, 145 Xingda Road, Taichung 40227, Taiwan*

Correspondence should be addressed to Jr-Fong Dang; jrfongdang@gmail.com

Received 24 March 2021; Revised 10 November 2021; Accepted 2 December 2021; Published 7 February 2022

Academic Editor: Miguel A. Salido

Most studies on common due date problems discuss the topic in the context of solving the minimum penalty value for the single-machine or parallel machine scheduling environment. This study extends the problem of the common due date to the dynamic flow shop environment and proposes the enhanced heuristic algorithm to solve the minimum penalty value. The enhanced heuristic algorithm is characterized by designating mutant genomes of the child as the genomes located at the central location before mutation. The advantage is to integrate the successful experiences of the conventional common due date algorithm to improve the efficiency of the proposed heuristic algorithm. The performance of both algorithms is analyzed in terms of uniformly distributed job numbers, processing time, early penalty, and late penalty. The resulting outcomes indicate that the enhanced heuristic algorithm outperforms the conventional CDDA and EDD proposed by previous studies in the average penalties.

1. Introduction

The semiconductor manufacturing environment can be regarded as one of the most complicated production processes and usually operates hundreds of machines. Moreover, Liu et al. [1] stated that there are approximately 400 process operations in the process flow and the processing cycle may last for several months. To gain a competitive edge in semiconductor manufacturing, the enterprises intend to shorten the cycle time, reduce the production costs, and improve the quality by executing effective controls on the production process. Mönch et al. [2] claimed that production scheduling can be an appropriate approach to solving the mentioned problems. Furthermore, a manager would like to satisfy those chips that are in the processing list at a predetermined time point. That is, they are of a common due date. The objective can be to find the optimal schedule of the jobs, which minimizes the total penalty based on the due date and the earliness or tardiness of each job [3]. The earliness and tardiness objective functions are motivated by the just-in-time (JIT) philosophy. Moreover, JIT is applied in many back-end facilities due to the need to reduce holding costs and meet internal and external due dates. This leads to production cycles being optimized so that companies produce products in an efficient manner from their manufacturing facilities. For instance, service companies would adopt the JIT scheduling concept to reduce inventory costs and focus on the satisfaction of customers. In general, the cost of being early can be considered as a holding cost for finished goods, deterioration of perishable goods, and opportunity costs. The cost of being tardy may be the backlogging cost, which includes performance penalties, lost

sales, and lost goodwill [4]. Lauff and Werner [5] argued that although there are lots of studies investigating single-machine problems with earliness and tardiness penalties, a few articles deal with parallel machine problems and only very few ones treat shop scheduling problems with earliness and tardiness penalties.

We can see that previous studies mostly discussed the common due date of the single-machine or parallel machine scheduling; thus, this study extends the common due date problem to the dynamic flow shop scheduling environment. Because the dynamic flow shop scheduling environment is an NP-hard problem [6–9], we developed the enhanced genetic algorithm. By the conventional genetic algorithm, the specific shipping dates expected by different customers are encoded to the maternal gene sequence, and thus the children are generated through the genetic evolution of crossover and mutation. As a result, the most elite child with the solution of the minimum acceptable penalty value is selected to be the common due date. In order to optimize the number of evolutions and iterations, our proposed method adds iteration optimizing factors to force the children to mutate at each iteration and utilizes the evolved mutated children for decoding the common due date schedule. After comparing with the normally evolved children through the traditional genetic algorithm, any child with the minimum penalty value is selected to be the common due date for the jobs reported to customers. In this article, we utilize Monte Carlo simulation to generate numerical examples. The results show that the average penalties obtained by our proposed method are less than those obtained by the traditional genetic algorithm with a 95% confidence interval in terms of uniformly distributed job numbers, processing time, early penalty, and late penalty.

The remaining parts of this article can be organized as follows. The literature review is surveyed in Section 2. Section 3 presents our proposed algorithm and Section 4 utilizes the illustrative case study and further conducts the analysis based on the results. Finally, Section 5 concludes this study and provides the future direction of the research.

2. Literature Review

Moore [10] proposed the algorithm considering computational efficiency for large problems and formulated for sequencing jobs through a single facility to minimize the number of tardy jobs. Lawler [11] considered a sequence minimizing the maximum of the incurred costs. The cost can be incurred by completing the job at time t. Panwakar et al. [7] investigated the penalty function based on the due date value and the earliness or the tardiness of each job in the selected sequence. Ow and Morton [12] proposed their algorithm to minimize total early and tardy costs by a given set of jobs processed on a single machine. Two dispatch priority rules are proposed and tested for this NP-complete problem. Lee et al. [6] studied the case of job-dependent penalties under a certain condition on the ratio of processing times and weights. They developed dynamic programming algorithms to solve the solutions. Błazewicz et al. [13] presented the two-machine non-preemptive flow shop scheduling problem with a total weighted late work criterion and a common due date. The late work performance measure estimates the quality of the obtained solution with regard to the duration of late parts of tasks, not taking into account the quantity of this delay. Yang et al. [14] investigated the assignment of single-machine multiple common due dates and scheduling problems with consideration of position and resource allocation-dependent processing time. They proposed two algorithms that can be executed within polynomial time. Wang and Wang [15] studied a single-machine earliness-tardiness scheduling problem with due date assignment where the job processing time is dependent on its starting time and resource allocation. Liu et al. [16] proposed a new metaheuristic algorithm based on the architecture of the iterated greedy algorithm with enhanced destruction, construction, and new repair method to solve dynamic scheduling in the permutation flow shop with the new order arrival. Koulamas [17] considered the single-machine common due date assignment problem with generalized earliness and tardiness penalties and showed that the problem can be solved optimally in the polynomial time by dynamic programming. Li and Chen [9] addressed a single-machine scheduling and common due date assignment problem in which the actual processing time of a job is a linearly increasing function of the total basic processing times of already processed jobs. Sun et al. [18] studied the problem of two-machine no-wait flow shop scheduling with a learning effect and convex resource-dependent processing times. They provided a bicriteria analysis where the first criterion is to minimize the weighted sum of earliness, tardiness, and flow allowance costs, and the second criterion is to minimize the weighted sum of resource consumption cost. Fu et al. [19] draw a decision on the common due date, resource allocation, and the sequence of jobs by minimizing total earliness, tardiness, common due date cost, and total resource cost. Lu et al. [20] developed a hybrid multiobjective grey wolf optimizer (HMOGWO) to solve the multiobjective dynamic welding scheduling problem (MODWSP). Their model considers machines breakdown, jobs with poor quality, and job release delay to fulfill the needs of dynamic production. Zhou et al. [21] considered an event-triggered dynamic scheduling problem in the cloud manufacturing environment. They further utilized the case study to demonstrate its model capability in task execution time. Yan et al. [22] solved a real-time dynamic job-shop scheduling problem in a robotic cell where multiple jobs enter into the cell with unexpected arriving rates. They formulate the problem as a sophisticated mixed-integer programming (MIP) model and propose an exact iterative algorithm. The resulting outcomes validate the effectiveness and efficiency of the strengthened MIP model and the iterative algorithm.

3. Methodology

The common due date algorithm [4] indicates that the advantage completion dates with the lowest total penalty cost are all sets at the center of job numbers sequences. Therefore, the proposed GA_E designates the optimizing

factors mutant genomes of the children as the centrally located genomes of the offspring before mutation in the dynamic flow shop scheduling. This advantage intends to integrate the successful experiences of the common due date algorithm to improve the efficiency of GA_E.

3.1. Notations and the Mathematical Model. The notations used in this article are given as follows: P, the total penalty cost; C_j, the completion time of job j; CDD_j, the common due date of job j; EP_j, the early penalty cost of job j; LP_j, the late penalty cost of job j;

The objective function of our model is to minimize the total penalty cost and can be formulated as follows:

$$\text{Minimize } P = \sum_{j=1}^{n} \left[\left(CDD_j - C_j \right) \cdot EP_j \cdot E_j + \left(C_j - CDD_j \right) \cdot LP_j \cdot L_j \right], \quad \text{where } E_j + L_j = 1, \quad E_j \in \{0,1\}, L_j \in \{0,1\}, \forall j = 1, \ldots n. \tag{1}$$

In (1), we sum up the penalty of each job. Note that $E_j + L_j = 1$ means that the early penalty cost and late penalty cost of job j cannot exist simultaneously. The early penalty cost of job j can be calculated by $(CCD_j - C_j) \cdot EP_j$, and we can obtain the late penalty cost of job j by $(C_j - CCD_j) \cdot LP_j$. In our solution procedure, we apply the enhanced genetic algorithm, common due date algorithm, and early due date rule to solve our model.

3.2. Enhanced Genetic Algorithm. This study proposes the enhanced genetic algorithm (refer to GA_E) performed by the following 9 steps:

Step 1. Population size: Set the job order number (n) under the dynamic flow shop scheduling environment as the number of population size.

Step 2. Parent selection: Take the work order with the maximum delay penalty value as the last genome of the maternal gene sequences and arrange the remaining gene sequences randomly. After arrangement, if the last delay penalty value appears the same and the work order can be selected from multiple options, then the last job is randomly arranged in priority and the remaining jobs are randomly arranged.

Step 3. Rule of crossover: Arrange the parents in a descending sequence based on the total penalty cost from high to low. The parents with the largest total penalty value are crossed over with those with the second largest value; the parents with the third maximum value are mated with those with the fourth maximum value, and so on. They iteratively cross over with each other until the same number of children is generated.

Step 4. Crossover value (child): The child from the crossover is generated as follows: add up the parents of every pair and calculate the mean value of the total penalty, and then conduct unconditional rounding up. If the roundup value is repeated, assign the next unrepeated order to replace the value. For example, the mean value of the parents 1 and 2 is 3.5; thus, the children will be rounded up in value 4. This roundup value is the offspring of this crossover. Stop the procedure when all children are completely generated.

Step 5. Mutation probability of children: Set the mutation probability as $1/n$ (for instance, if there are four generations of children, set the mutation probability as $1/n = 1/4 = 0.25$). The children with the lowest total penalty cost before mutation are selected for mutation.

Step 6. Selection of the centrally located genomes of the children before mutation: The GA_E designates the mutant genomes of the children as the centrally located genomes of the offspring before its mutation. For example, the genome at the central location of children 1 before mutation is 5 and the genome at the central location of children 2 after mutation is random value 1.

Step 7. Application of the backward-forward heuristic: Find out the minimum total penalty value of this job order scheduling by applying the backward-forward heuristic [4]. In the backward phase, the initial job sequence is developed. The sequential job assignments start from the last position and proceed backward toward the first position. The assignments are completed when the first position is assigned to a job. Perform the forward pass on the job sequence found in the backward phase that is the "best" sequence at this stage. The forward pass progresses from the job in position 1 toward the job in position N. Let k define the lag between two jobs that are exchanged in the sequence. For example, jobs occupying positions 1 and 3 have lag k equal to 2.

Step 8. Select the minimum total penalty value obtained at Step 7, and its scheduling is the optimal solution for the common due date problem.

Step 9. Termination rule: repeat Step 1 to Step 8 and continue crossover for $N/2$ times. If one observes no improvement in regard to the currently attempted optimal solution, then stop GA_E algorithm. The lowest total penalty cost during the crossover is considered the optimal solution.

3.3. The Encoding and Decoding of the GA_E. In Figure 1(a), given n jobs, an order is encoded by a permutation of integers between 1 and m our GA_E. In Figure 1(b), the genetic evolution of crossover rules can be seen in Step 3 and Step 4. In Figure 1(c), when the optimum sequence is found, we first decode it back to a route where the sequence is represented and the job n is processed on machines. Then, we decode a route back to a schedule as Gantt chart. We can derive the specific completion time C_j according to equation (1) with the minimum penalty value. The resulting outcomes can be generated as the common due date for the jobs, which can be provided for customers.

4. Parameter Design and Case Study

4.1. Numerical Results

4.1.1. Parameter Design. Based on [4], Table 1 shows three scenarios with its parameter setup. We assume all parameters following the uniform distribution on a specific interval. For instance, in scenario 1, the processing time of machine 1 has a uniform distribution on the interval [6, 23].

4.2. The GA_E Algorithm Procedure. In scenario 1, from the simulated results of the uniform distribution shown in Table 1, we can obtain the simulated datasets of the dynamic flow shop scheduling shown in Table 2. To clearly explain our proposed algorithm, we detail the solution procedure of the simulation result of scenario 1 as follows.

Step 1. Simulate the population size under scenario 1, as shown in Table 3.

Step 2. Take job order 1 with the maximum delay penalty value as the last genome of the initial population genetic sequences and arrange the remaining genetic sequences randomly.

Step 3. Arrange the parents in a descending sequence based on the total penalty cost from high to low. The member with the largest total penalty value is crossed over with that with the second largest value (i.e., parents 1 and 2), and the member with the third maximum value is crossed over with that with the fourth largest value (i.e., parents 3 and 4), and so on.

Step 4. Generate children from the crossover. Add up the parents of every pair and calculate the mean value of the total penalty, and then conduct unconditional rounding up. This roundup value is the crossover of the child in this iteration. Stop the procedure until all children are completely produced, as shown in Table 3.

Step 5. Set the mutation probability of child as $1/n$: For instance, for scenario 1, if there are six generations of offspring after mating, set the mutation probability as $1/n = 1/6 = 0.1667$). Next, from iteration 2, select the children member 24 with the lowest total penalty cost of \$1344 before mutation as a candidate for mutation, as shown in Table 4.

Step 6. Select 4 as the centrally located genome of the children (member 24, 3-5-6-4-1-2) before mutation, and the centrally located genome, 1, of the child (member 26, 2-5-3-1-6-4) after mutation. The total penalty cost is reduced to \$1313. So far, the GA_E algorithm shown in Table 4 completes the iterated application of scenario 1 for the second time.

Step 9. Termination rule: after iterative crossover for $6/2 = 3$ times in scenario 1, the simulation results indicate during crossover that, among all the members, parent member 6 is the one with the lowest total penalty cost—the order processing sequence is 3-5-6-2-4-1, the optimum solution is \$1267, and the optimum delivery date set by GA_E is 112 days.

In scenario 2, from the simulated results in Tables 5 and 6, it is found that, during the crossover, among all the child members, the children member 34 after mutation is the one with the lowest total penalty cost, its order processing sequence is 5-1-6-7-2-4-3-8, the optimum solution is \$418, and the optimum shipping date set by GA_E is 72 days. In scenario 3, from the simulated results in Tables 7 and 8, it is found that, during crossover, among all the child members, the children member 20 is the one with the lowest total penalty cost, the order processing sequence is 5-3-4-1-2, the optimum solution is \$350, and the optimum delivery date set by GA_E is 84 days.

4.3. The Common Due Date Algorithm

4.3.1. The Common Due Date Algorithm Procedure. Sule [4] proposed the general common due date algorithm (CDDA) in the following steps:

Step 1. Select the early and late penalty values of all unscheduled job orders and find out the minimum one.

Step 2. As for N job orders, arrange the one with the maximum early penalty value in the first place for production in priority and similarly arrange the one with the minimum late penalty value in the last place for production. Among the work orders with early penalty values, those taking a long time for processing are arranged forward, and among the work orders with late penalty values, those taking a long time for processing are arranged backward. For any work order with the same early and late penalty cost, the arrangement can be optional, either the earliest scheduling or the latest scheduling.

Step 3. Respectively, calculate the total penalty value for the common due date of each job order according to the scheduling determined in Step 2.

Step 4. Find out the minimum total penalty value of this work order scheduling in the backward-forward heuristic [4].

Step 5. Select the minimum total penalty value obtained in Step 4, and its scheduling is the optimal solution for the common due date.

4.3.2. The Numerical Results of the Common Due Date Algorithm. The initial scheduling of the specific shipping

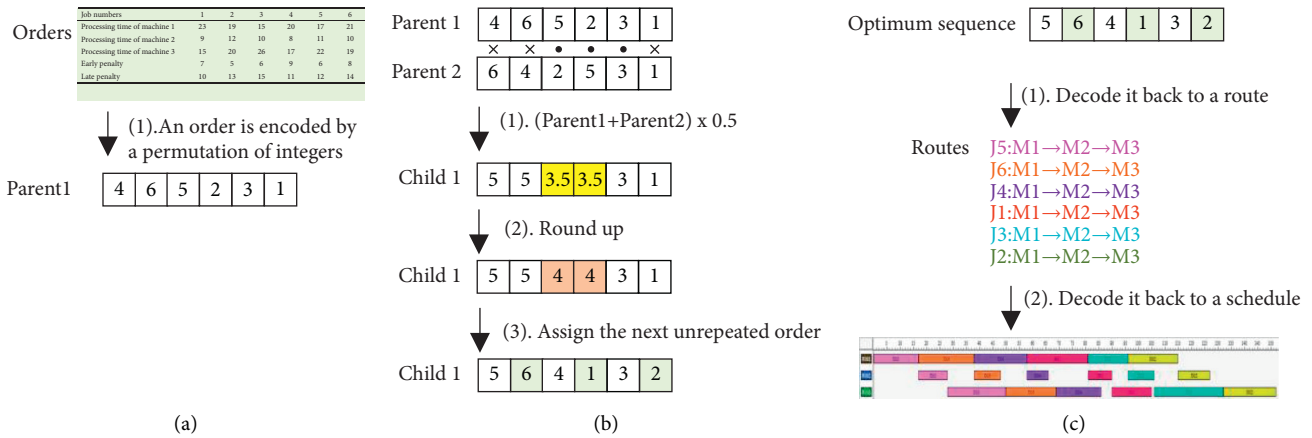

Figure 1: The encoding and decoding of the enhanced genetic algorithm. (a) Encoding. (b) Crossover. (c) Decoding.

Table 1: Parameter design for the dynamic flow shop scheduling.

	Scenario 1	Scenario 2	Scenario 3
Job numbers	[1, 6]	[6, 15]	[5, 10]
Processing time of machine 1	[6, 23]	[6, 15]	[6, 23]
Processing time of machine 2	[3, 12]	[3, 8]	[3, 10]
Processing time of machine 3	[3, 26]	[3, 15]	[3, 21]
Early penalty	[5, 9]	[1, 5]	[2, 9]
Late penalty	[10, 15]	[5, 10]	[5, 15]

Table 2: Simulation data of scenario 1.

Job numbers	1	2	3	4	5	6
Processing time of machine 1	23	19	15	20	17	21
Processing time of machine 2	9	12	10	8	11	10
Processing time of machine 3	15	20	26	17	22	19
Early penalty	7	5	6	9	6	8
Late penalty	10	13	15	11	12	14

Table 3: Calculation of crossover genetic sequence for children.

Parent 1	4	6	5	2	3	1
Parent 2	6	4	2	5	3	1
(Parent 1 + parent 2) × 0.5	5	5	3.5	3.5	3	1
Children sequence	5	5	4[1]	4	3	1
Children sequence after roundup	5	6[2]	4	1	3	2

1 The mean value of the parents 1 and 2 is 3.5; thus, the children will be rounded up in value 4. 2 Because children's sequence 5 overlaps with the first genetic sequence 5, they are assigned to 6.

Table 4: Iterations in scenario 1 after 30 generations.

Iteration	Member	Parent	Penalty ($)	Children	Member	Due date	Penalty ($)
1	1 2	4-6-5-**2**-3-1 6-4-2-**5**-3-1	1859 1845	5-6-4-**1**-3-2	19	105	1805
	3 4	5-4-2-**3**-6-1 3-5-4-**6**-2-1	1543 1345	4-5-2-**6**-1-3	20	109	1685
	5 6	3-6-2-**5**-4-1 3-5-6-**2**-4-1	1319 **1267*** **(optimal)**	3-5-6-**4**-1-2	21	109	1344
	—	—	—	2-5-6-**3**-4-1	22	118	1320
2	7 8	6-2-4-**3**-5-1 4-5-2-**6**-3-1	1603 1705	5-4-3-**6**-1-2	23	112	1496
	9 10	3-4-5-**6**-2-1 3-6-4-**2**-5-1	1487 1467	3-5-6-**4**-1-2	24	109	1344
	11 12	5-2-4-**6**-3-1 3-5-2-**4**-6-1	1441 1372	4-5-3-**6**-1-2	25	115	1619
	—	—	—	2-5-3-**1**-6-4	26	114	1313
3	13 14	4-3-5-**6**-2-1 4-2-5-**6**-3-1	1763 1573	4-3-5-**6**-1-2	27	112	1568
	15 16	2-6-4-**3**-5-1 5-2-4-**3**-6-1	1522 1421	4-5-6-**3**-1-2	28	115	1713
	17 18	6-5-2-**4**-3-1 2-5-3-**4**-6-1	1419 1291	4-5-3-**1**-6-2	29	111	1505
	—	—	—	3-2-6-**1**-4-5	30	105	1328

Table 5: Simulation data of scenario 2.

Job numbers	1	2	3	4	5	6	7	8
Processing time of machine 1	6	13	7	12	8	10	9	11
Processing time of machine 2	4	3	6	5	8	7	8	3
Processing time of machine 3	14	12	6	3	9	10	11	15
Early penalty	1	2	5	2	1	4	3	5
Late penalty	10	8	7	9	7	5	6	5

Table 6: Iterations in scenario 2 after 39 generations.

Iteration	Member	Parent	Penalty ($)	Child	Member	Due date	Penalty ($)
	1 2	4-3-2-5-1-8-7-6 3-5-7-1-2-8-4-6	775 754	4-5-6-3-2-8-7-1	25	65	881
	3 4	8-5-7-4-2-3-1-6 5-7-4-2-3-1-8-6	706 659	7-6-8-3-4-2-5-1	26	62	943
1	5 6	1-3-2-4-5-7-8-6 4-1-5-3-2-8-7-6	656 602	3-2-4-5-6-8-1-7	27	67	979
	7 8	1-2-3-4-5-7-8-6 1-4-2-3-5-7-8-6	593 576	1-3-4-5-6-7-8-2	28	60	750
	—	—		5-1-8-7-2-3-4-6	29	77	459
	9 10	6-7-5-4-3-2-1-8 2-6-7-5-4-3-1-8	864 662	4-7-6-5-8-3-1-2	30	74	819
	11 12	7-5-4-2-1-3-6-8 3-1-2-7-6-4-5-8	639 634	5-3-4-6-7-8-1-2	31	65	1007
2	13 14	4-1-3-2-7-6-5-8 2-1-3-4-5-7-6-8	599 491	3-1-4-5-6-7-8-2	32	60	790
	15 16	2-1-7-3-4-5-6-8 5-1-6-7-2-3-4-8	472 451	4-1-7-5-3-6-8-2	33	62	632
	—	—		5-1-6-7-2-4-3-8	34	72	418* (optimal)
	17 18	6-3-1-2-5-4-7-8 5-4-3-2-1-7-6-8	656 655	6-4-2-3-5-7-8-1	35	67	919
	19 20	1-6-7-5-4-3-2-8 2-5-3-1-6-7-4-8	590 577	2-6-5-3-7-8-4-1	36	66	809
3	21 22	7-2-1-3-4-6-5-8 1-5-6-7-4-3-2-8	545 545	4-5-6-7-8-1-2-3	37	73	863
	23 24	7-6-1-2-3-4-5-8 2-1-6-7-5-4-3-8	437 435	5-4-6-7-8-1-2-3	38	73	859
	—	—		7-6-5-3-2-1-4-8	39	65	750

Table 7: Simulation data of scenario 3.

Job numbers	1	2	3	4	5
Processing time of machine 1	6	15	20	23	14
Processing time of machine 2	3	9	4	10	5
Processing time of machine 3	3	9	21	17	14
Early penalty	8	6	5	9	2
Late penalty	5	9	10	15	13

dates for three scenarios in Table 9 is obtained. As for scenario 2 in Table 10, the order processing sequence is 5-1-2-4-7-6-8-3, the optimum solution is $440, and the optimum delivery date set by CDDA is 77 days. By comparing the performance of CDDA and GA_E in three scenarios in Table 11, the results indicate that the optimum solutions of GA_E is smaller than those of CDDA, and the penalty value respectively is reduced 2.08%, 5%, and 47.29%.

4.3.3. Statistics and Informatics. This section presents statistics and informatics of the numerical results by simulation dataset (e.g., scenarios 1, 2, and 3) corresponding to the average penalties and standard deviation of due dates between CDDA and GA_E and the mean plot of penalties. Table 12 shows that the average penalties are 1261.789 dollars for the CDDA method and 913.889 dollars for the GA_E method. Furthermore, the average penalties obtained by the GA_E method lead to a 28% reduction compared to that obtained by the CDDA method.

4.4. The Most Compared Algorithms and Their Statistics. This article further applies the earliest due date (EDD) rule to see our model performance. The EDD rule is that jobs are processed according to the due date, the earliest due date first. We assume that the total processing time is set up to be the due date once this article employs the EDD rule. In addition, this article utilizes the Monte Carlo simulation method to generate the validation data, the processing time, early penalty, and late penalty, as shown in Tables 2, 5, 7. It simulates 30 replications of datasets, including the processing time of machines from 3 to 26, early penalty with a range between 1 and 9 dollars, and a late penalty with a range between 5 and 15 dollars. The resulting outcomes derived by GA_E, CDDA, and EDD can be found in Tables 13–15. One would know that analysis of variance (ANOVA) is a statistical technique to check if the means of two or more groups are significantly different from each other. In this article, we compare the performance of these three methodologies using ANOVA. In scenario 1, the average penalty derived by GA_E model is $1328.77, by CDDA model is

Table 8: Iterations in scenario 3 after 30 generations.

Iteration	Member	Parent	Penalty ($)	Child	Member	Due date	Penalty ($)
1	1 2	3-2-**5**-4-1 2-5-**3**-4-1	834 813	3-4-**5**-1-2	19	84	444
	3 4	5-3-**2**-4-1 4-2-**5**-3-1	750 666	5-3-**4**-1-2	20	84	350* **(optimal)**
	5 6	5-4-**3**-2-1 3-5-**4**-2-1	443 386	4-5-**1**-2-3	21	67	600
	—	—	—	1-5-**4**-2-3	22	70	937
2	7 8	2-4-**5**-3-1 4-2-**3**-5-1	732 708	3-4-**5**-1-2	23	84	444
	9 10	4-5-**2**-3-1 5-4-**2**-3-1	600 536	5-1-**2**-3-4	24	53	1226
	11 12	4-3-**5**-2-1 3-5-**4**-2-1	526 386	4-5-**1**-2-3	25	67	600
	—	—	—	4-5-**3**-2-1	26	85	498
3	13 14	2-3-**5**-4-1 4-1-**3**-5-2	831 773	3-2-**4**-5-1	27	85	653
	15 16	2-4-**5**-3-1 3-2-**4**-5-1	732 653	3-4-**5**-1-2	28	84	444
	17 18	5-2-**4**-3-1 3-4-**5**-2-1	614 462	4-3-**5**-1-2	29	85	508
	—	—	—	2-5-**4**-1-3	30	79	593

Table 9: The initial sequences obtained by following CDDA of scenarios 1, 2 and 3.

Scenario 1 and job number is 6	2-5-3-1-6-4
Scenario 2 and job number is 8	5-1-2-4-7-6-8-3
Scenario 3 and job number is 5	5-3-2-1-4

Table 10: Penalties over the range of due dates by following CDDA.

Scenario	Job number	Due date	Penalty ($)
1	2	51	3851
	5	73	2597
	3	99	1583
	1	114	1313
	6	133	1294*
	4	150	1651
2	5	25	1967
	1	39	1281
	2	51	825
	4	54	741
	7	67	520
	6	77	440*
	8	92	455
	3	98	521
3	5	33	1845
	3	59	873
	2	68	685
	1	71	664*
	4	105	868

Table 11: The comparisons of the CDDA and GA_E.

	Scenario 1		Scenario 2		Scenario 3	
	CDDA	GA_E	CDDA	GA_E	CDDA	GA_E
Job numbers	6		8		5	
Optimal sequences	2-5-3-1-6-4	3-5-6-2-4-1	5-1-2-4-7-6-8-3	5-1-6-7-2-4-3-8	5-3-2-1-4	5-3-4-1-2
Due date	133	112	77	72	71	84
Penalty ($)	1294*	1267	440	418*	664	350*
		−2.08%↓		−5.0%↓		−47.29%↓

Table 12: Average and standard deviation of the penalties.

	CDDA	GA_E
Average penalties over the range of due dates ($)	1261.789	913.889
Standard deviation over the range of due dates ($)	864.950	431.665

Table 13: The 30 replications simulation results of scenario 1.

Runs	GA_E algorithm			Common due date algorithm			EDD rule		
	(1) Optimum sequence	(2) Common due date (days)	(3) Penalty ($)	(4) Optimum sequence	(5) Common due date (days)	(6) Penalty ($)	(7) Optimum sequence	(8) Common due date (days)	(9) Penalty ($)
1	3-2-6-5-4-1	112	1254	5-1-2-3-6-4	117	1558	4-1-5-6-2-3	45	4560
2	2-3-5-6-1-4	118	1197	6-3-4-1-2-5	108	1650	4-1-5-6-2-3	67	3350
3	2-6-3-5-4-1	118	1358	5-4-1-3-2-6	111	1757	4-1-5-6-2-3	93	2362
4	5-3-6-2-1-4	115	1286	6-1-3-4-5-2	112	1716	4-1-5-6-2-3	112	1982
5	3-2-6-5-1-4	112	1249	1-2-5-6-3-4	115	1658	4-1-5-6-2-3	132	2022
6	3-2-6-4-1-5	107	1294	6-4-1-3-2-5	115	1905	4-1-5-6-2-3	158	2542
7	5-6-2-3-1-4	115	1390	6-1-5-4-3-2	111	1879	4-1-5-6-2-3	67	3350
8	2-3-5-6-4-1	118	1202	5-1-6-4-2-3	107	1862	4-1-5-6-2-3	45	4560
9	5-2-6-4-3-1	106	1452	6-2-3-1-4-5	113	1454	4-1-5-6-2-3	158	2542
10	3-6-2-5-4-1	112	1319	3-6-4-5-1-2	109	1502	4-1-5-6-2-3	132	2022
11	6-5-3-2-1-4	118	1442	5-1-6-3-4-2	116	1629	4-1-5-6-2-3	93	2362
12	6-2-5-3-4-1	120	1463	4-3-6-2-1-5	110	1585	4-1-5-6-2-3	45	4560
13	3-2-6-1-4-5	105	1328	6-5-3-1-4-2	113	1508	4-1-5-6-2-3	67	3350
14	2-3-6-4-5-1	113	1296	1-5-6-2-3-4	112	1712	4-1-5-6-2-3	158	2542
15	6-2-3-5-1-4	120	1434	6-4-1-5-2-3	114	2067	4-1-5-6-2-3	45	4560
16	5-2-1-6-3-4	109	1580	6-1-2-5-3-4	117	1852	4-1-5-6-2-3	112	1982
17	3-2-5-6-1-4	112	1187	5-4-2-6-3-1	107	1597	4-1-5-6-2-3	132	2022
18	2-3-5-4-1-6	116	1287	6-3-5-4-2-1	115	1466	4-1-5-6-2-3	67	3350
19	3-2-6-1-5-4	105	1330	6-2-1-3-5-4	114	1604	4-1-5-6-2-3	158	2542
20	2-5-3-6-4-1	118	1226	1-5-6-4-2-3	109	1820	4-1-5-6-2-3	93	2362
21	6-2-3-5-4-1	120	1439	2-6-4-3-1-5	113	1482	4-1-5-6-2-3	67	3350
22	2-5-3-1-6-4	114	1313	2-1-6-5-4-3	114	1659	4-1-5-6-2-3	132	2022
23	6-5-3-2-4-1	118	1447	6-4-5-1-2-3	106	2007	4-1-5-6-2-3	112	1982
24	2-6-3-5-1-4	118	1353	4-5-3-6-1-2	115	1619	4-1-5-6-2-3	45	4560
25	5-3-6-2-4-1	115	1291	4-5-1-2-3-6	111	2042	4-1-5-6-2-3	158	2542
26	2-3-5-4-6-1	116	1267	1-5-6-4-3-2	109	1774	4-1-5-6-2-3	112	1982
27	3-2-5-4-1-6	110	1277	1-6-5-4-2-3	112	1819	4-1-5-6-2-3	67	3350
28	5-6-2-3-4-1	115	1395	1-3-6-2-4-5	113	1511	4-1-5-6-2-3	132	2022
29	2-5-3-6-1-4	118	1221	6-1-2-3-4-5	121	1724	4-1-5-6-2-3	93	2362
30	3-2-6-4-5-1	107	1286	4-1-3-6-5-2	113	1858	4-1-5-6-2-3	112	1982
		Avg.	1328.77($)		Avg.	1709.20($)		Avg.	2835.93($)

$1709.20, and by EDD model is $2835.93 in Table 13. Based on the data, the statistic can be 61.97 and p value is less than 0.05. It means that a significant difference exists among the three models. Moreover, we further apply Tukey's honestly significant difference (HSD) test to see all possible pairwise comparisons while keeping the familywise error rate low. The resulting outcomes can be arranged in Table 16. Table 13 indicates that GA_E outperforms the other two models in penalties. Following the same procedure, our proposed model results in improved penalties compared to those obtained by CDDA and EDD models of scenarios 2 and 3 in Tables 17 and 18. Obviously, our proposed solution procedure can be applied to solve the scheduling problem in an efficient manner. The manufacturing companies would benefit substantially from the application of our proposed model.

Table 14: The 30 replications simulation results of the scenario 2.

Runs	GA_E algorithm			Common due date algorithm			EDD rule		
	(1) Optimum sequence	(2) Common due date (days)	(3) Penalty ($)	(4) Optimum sequence	(5) Common due date (days)	(6) Penalty ($)	(7) Optimum sequence	(8) Common due date (days)	(9) Penalty ($)
1	5-4-2-1-3-7-6-8	79	465	3-4-1-7-6-2-8-5	64	833	3-4-1-5-6-2-7-8	19	1198
2	1-2-4-5-3-6-7-8	73	459	3-4-8-1-5-7-6-2	71	907	3-4-1-5-6-2-7-8	27	926
3	2-4-1-5-3-8-7-6	79	491	8-5-3-4-7-1-2-6	66	889	3-4-1-5-6-2-7-8	43	558
4	5-2-4-1-3-7-8-6	74	461	4-8-3-5-6-7-1-2	66	934	3-4-1-5-6-2-7-8	52	450
5	2-1-4-5-3-6-7-8	73	445	3-1-7-2-6-4-8-5	66	684	3-4-1-5-6-2-7-8	62	410
6	1-5-4-2-6-7-8-3	77	491	7-2-3-6-1-4-5-8	70	608	3-4-1-5-6-2-7-8	74	470
7	4-1-7-5-2-3-6-8	74	431	6-7-1-2-5-4-3-8	73	538	3-4-1-5-6-2-7-8	85	635
8	4-7-5-1-2-3-8-6	81	507	6-8-3-7-4-5-1-2	62	967	3-4-1-5-6-2-7-8	100	860
9	1-5-4-2-7-6-3-8	67	482	4-3-6-2-1-8-5-7	72	894	3-4-1-5-6-2-7-8	27	926
10	5-1-6-7-2-4-3-8	72	418	7-8-5-3-4-6-2-1	61	868	3-4-1-5-6-2-7-8	74	470
11	4-7-5-2-1-3-6-8	81	498	2-8-4-6-1-7-5-3	77	774	3-4-1-5-6-2-7-8	19	1198
12	4-1-7-2-5-3-8-6	74	463	6-8-5-1-7-3-2-4	76	777	3-4-1-5-6-2-7-8	52	450
13	5-1-2-4-7-3-8-6	67	430	7-6-5-8-3-4-2-1	68	764	3-4-1-5-6-2-7-8	43	558
14	1-2-4-5-7-6-8-3	77	484	1-4-6-7-2-5-8-3	68	647	3-4-1-5-6-2-7-8	100	860
15	2-1-4-5-3-8-7-6	77	464	7-1-5-3-2-4-8-6	69	485	3-4-1-5-6-2-7-8	19	1198
16	4-5-7-2-3-1-6-8	80	544	5-7-3-8-6-2-1-4	67	927	3-4-1-5-6-2-7-8	85	635
17	1-7-4-2-5-3-6-8	71	444	2-6-4-8-3-7-5-1	70	834	3-4-1-5-6-2-7-8	27	926
18	5-1-6-8-2-4-3-7	76	466	4-2-1-6-7-5-3-8	75	563	3-4-1-5-6-2-7-8	74	470
19	2-4-5-1-3-7-8-6	81	505	2-1-5-3-6-7-8-4	67	626	3-4-1-5-6-2-7-8	85	635
20	1-2-4-5-3-8-7-6	77	478	8-3-2-4-1-5-7-6	67	755	3-4-1-5-6-2-7-8	19	1198
21	2-1-5-4-3-6-8-7	73	477	8-2-3-7-5-6-4-1	68	818	3-4-1-5-6-2-7-8	52	450
22	4-5-7-1-2-3-6-8	80	464	7-8-5-3-1-2-4-6	72	723	3-4-1-5-6-2-7-8	43	558
23	1-7-4-5-2-3-8-6	70	445	4-1-6-5-7-3-8-2	66	624	3-4-1-5-6-2-7-8	62	410
24	5-1-2-4-3-7-6-8	60	485	6-7-8-2-3-4-5-1	71	816	3-4-1-5-6-2-7-8	27	926
25	1-2-5-4-3-8-6-7	75	474	1-4-7-3-6-8-2-5	62	749	3-4-1-5-6-2-7-8	43	558
26	2-1-5-4-3-7-8-6	74	457	8-1-2-6-7-4-3-5	76	570	3-4-1-5-6-2-7-8	85	635
27	4-1-7-5-8-3-6-2	77	522	3-1-4-8-6-5-7-2	64	801	3-4-1-5-6-2-7-8	27	926
28	1-5-2-4-7-6-8-3	77	471	6-4-8-5-1-3-2-7	74	765	3-4-1-5-6-2-7-8	19	1198
29	2-4-1-5-3-7-6-8	75	451	7-1-2-3-5-4-6-8	69	457	3-4-1-5-6-2-7-8	62	410
30	5-1-4-7-2-3-6-8	66	429	5-8-3-4-7-6-2-1	66	891	3-4-1-5-6-2-7-8	74	470
		Avg.	**470.03($)**		**Avg.**	**749.6($)**		**Avg.**	**714.15**($)

Table 15: The 30 replications simulation results of the scenario 3.

Runs	GA_E algorithm			Common due date algorithm			EDD rule		
	(1) Optimum sequence	(2) Common due date (days)	(3) Penalty ($)	(4) Optimum sequence	(5) Common due date (days)	(6) Penalty ($)	(7) Optimum sequence	(8) Common due date (days)	(9) Penalty ($)
1	5-3-2-1-4	71	664	3-5-1-2-4	62	835	1-2-5-3-4	12	2864
2	3-5-4-2-1	84	386	4-1-5-3-2	67	745	1-2-5-3-4	39	1811
3	4-5-3-1-2	85	480	4-1-2-3-5	62	983	1-2-5-3-4	54	1451
4	3-4-5-1-2	84	444	2-5-4-3-1	79	668	1-2-5-3-4	80	1217
5	5-4-3-1-2	85	416	4-2-1-5-3	62	731	1-2-5-3-4	105	1367
6	5-3-4-2-1	84	368	2-4-5-3-1	79	732	1-2-5-3-4	39	1811
7	3-5-4-1-2	84	332	4-3-2-5-1	80	582	1-2-5-3-4	12	2864
8	4-5-3-2-1	85	498	1-5-4-3-2	70	1100	1-2-5-3-4	54	1451
9	4-3-5-2-1	85	526	1-2-4-3-5	71	1329	1-2-5-3-4	54	1451
10	5-3-4-1-2	84	350	3-4-5-1-2	84	444	1-2-5-3-4	12	2864
11	3-4-1-5-2	73	556	4-3-1-5-2	74	620	1-2-5-3-4	105	1367
12	4-3-5-1-2	85	508	3-5-2-1-4	68	703	1-2-5-3-4	105	1367
13	2-5-4-1-3	79	593	2-1-4-3-5	71	1173	1-2-5-3-4	39	1811
14	5-4-1-2-3	67	536	5-1-2-4-3	53	1186	1-2-5-3-4	54	1451
15	4-3-2-5-1	80	582	2-4-5-1-3	79	657	1-2-5-3-4	12	2864
16	2-3-4-5-1	85	704	4-1-5-2-3	67	706	1-2-5-3-4	105	1367
17	5-4-2-3-1	73	536	4-2-3-5-1	83	708	1-2-5-3-4	105	1367
18	3-5-2-1-4	68	703	2-4-1-5-3	68	769	1-2-5-3-4	12	2864
19	5-2-4-1-3	79	539	3-2-5-1-4	68	769	1-2-5-3-4	80	1217
20	3-1-5-4-2	90	614	3-1-4-2-5	76	759	1-2-5-3-4	12	2864
21	3-4-1-2-5	73	560	1-5-3-2-4	65	1157	1-2-5-3-4	80	1217
22	2-3-4-1-5	85	673	4-1-2-5-3	62	785	1-2-5-3-4	105	1367
23	5-4-2-1-3	73	476	3-2-4-5-1	85	653	1-2-5-3-4	39	1811
24	2-4-5-1-3	79	657	2-3-4-5-1	85	704	1-2-5-3-4	105	1367
25	5-2-3-1-4	77	703	4-5-3-2-1	85	498	1-2-5-3-4	54	1451
26	4-3-1-5-2	74	620	1-2-3-4-5	66	1476	1-2-5-3-4	12	2864
27	2-5-4-3-1	79	668	3-5-2-4-1	68	768	1-2-5-3-4	39	1811
28	4-5-1-2-3	67	600	2-3-5-1-4	74	796	1-2-5-3-4	105	1367
29	5-4-3-2-1	85	434	4-5-1-2-3	67	600	1-2-5-3-4	54	1451
30	3-5-1-4-2	90	592	5-1-4-2-3	70	757	1-2-5-3-4	12	2864
		Avg.	543.93($)		Avg.	813.1($)		Avg.	1842($)

Table 16: Pairwise comparison results of scenario 1.

Group 1	Group 2	Mean difference	P value	Reject
CDDA	EDD	1126.7333	0.001	True
CDDA	GA_E	−380.4333	0.0223	True
EDD	GA_E	−1570.1667	0.001	True

Table 17: Pairwise comparison results of scenario 2.

Group 1	Group 2	Mean difference	P value	Reject
CDDA	EDD	−30.5333	0.7729	False
CDDA	GA_E	−279.5667	0.001	True
EDD	GA_E	−249.0333	0.001	True

Table 18: Pairwise comparison results of scenario 3.

Group 1	Group 2	Mean difference	P value	Reject
CDDA	EDD	1028.9000	0.001	True
CDDA	GA_E	−269.1667	0.0316	True
EDD	GA_E	−1298.0667	0.001	True

5. Conclusion

By the simulation results derived in this study, we can see that employing the modified genetic algorithm can improve the optimum solution quality of common due date for the minimum total penalty value in the dynamic flow shop scheduling. Furthermore, the modified genetic algorithm can solve the complicated problem of common due date in the dynamic flow shop scheduling through a more simplified method, as shown in Section 4.2. At last, through iterative experiments, the modified genetic algorithm is proved to be superior to the CDDA approach proposed by Suel [4]. This article considers the advantage of genetic algorithm to design the termination rule, crossover, and mutation to generate diversified children at the initial stage; thus, the problems can be solved in a simple, timesaving manner. Furthermore, our studies compare the performance of the proposed GA_E EDD and CDDA. The resulting outcomes show that our model outperforms CDDA and EDD. The manufacturing companies would implement our solution procedure to gain the benefits.

One would note that during the process of obtaining the optimal solution, the study determines that the expected delivery date arising from permutation and combination of delivery date must be set at the central location of combination, a significant characteristic to be adopted to the enhanced genetic algorithm for solving the common due date problem in the three-stage dynamic flow shop. The following three points are proposed for future researchers to study. From the conclusion, it is known that the genetic algorithm is suitable for large population sizes and job orders. From the experiments, we find that, in the case of small population size and job orders, the CDDA can be directly used to avoid wasting too much calculating time and cost. When designing a new genetic algorithm, attempt to diversify the children after crossover. Furthermore, the simple crossover would generate homogeneous children without much variety. After several crossovers are iterated, similar children are more likely to be generated, but no optimum solution is provided. Hence, the variety or mutation rate should be increased during the crossover process to produce more diversified offspring. More combination modes shall be added in terms of child mutation. The homogeneous models may only be applicable in the case that the job orders are in small number; when the work orders are in large number, the children after mutation will easily become identical with those generated through general crossover; therefore, the importance of mutation cannot be accentuated and the significance of mutation is lost.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This research was funded by the Ministry of Science and Technology of Taiwan under Grant 110-2221-E-005-087-.

References

[1] J. Liu, F. Qiao, and W. Kong, "Scenario-based multi-objective robust scheduling for a semiconductor production line," *International Journal of Production Research*, vol. 57, no. 21, pp. 6807–6826, 2019.

[2] L. Mönch, J. W. Fowler, S. Dauzére-Pérès, S. J. Mason, and O. Rose, "A survey of problems, solution techniques, and future challenges in scheduling semiconductor manufacturing operations," *Journal of Scheduling*, vol. 14, pp. 583–599, 2011.

[3] M. Liu, S. Wang, F. Zheng, and C. Chu, "Algorithms for the joint multitasking scheduling and common due date assignment problem," *International Journal of Production Research*, vol. 55, no. 20, pp. 6052–6066, 2017.

[4] D. R. Sule, *Industrial Scheduling*, International Thomson Publishing Inc, MA, USA, 1997.

[5] V. Lauff and F. Werner, "Scheduling with common due date, earliness and tardiness penalties for multimachine problems: a survey," *Mathematical and Computer Modelling*, vol. 40, no. 5-6, pp. 637–655, 2004.

[6] C.-Y. Lee, S. L. Danusaputro, and C.-S. Lin, "Minimizing weighted number of tardy jobs and weighted earliness-tardiness penalties about a common due date," *Computers & Operations Research*, vol. 18, no. 4, pp. 379–389, 1991.

[7] S. S. Panwalkar, M. L. Smith, and A. Seidmann, "Common due date assignment to minimize total penalty for the one machine scheduling problem," *Operations Research*, vol. 30, no. 2, pp. 391–399, 1982.

[8] J. Błazewicz, E. Pesch, M. Sterna, and F. Werner, "The two-machine flow-shop problem with weighted late work criterion and common due-date," *European Journal of Operational Research*, vol. 165, pp. 408–415, 2005.

[9] S.-S. Li and R.-X. Chen, "Common due date assignment and cumulative deterioration scheduling on a single machine," *Engineering Optimization*, vol. 49, no. 6, pp. 976–989, 2017.

[10] J. M. Moore, "AnnJob, one machine sequencing algorithm for minimizing the number of late jobs," *Management Science*, vol. 15, no. 1, pp. 102–109, 1968.

[11] E. L. Lawler, "Optimal sequencing of a single machine subject to precedence constraints," *Management Science*, vol. 19, no. 5, pp. 544–546, 1973.

[12] P. S. Ow and T. E. Morton, "The single machine early/tardy problem," *Management Science*, vol. 35, no. 2, pp. 177–191, 1989.

[13] J. Błazewicz, E. Pesch, M. Sterna, and F. Werner, "The two-machine flow-shop problem with weighted late work criterion and common due-date," *European Journal of Operational Research*, vol. 165, pp. 408–415, 2005.

[14] S.-J. Yang, H.-T. Lee, and J.-Y. Guo, "Multiple common due dates assignment and scheduling problems with resource allocation and general position-dependent deterioration effect," *International Journal of Advanced Manufacturing Technology*, vol. 67, no. 1–4, pp. 181–188, 2013.

[15] X.-Y. Wang and J.-J. Wang, "Single-machine due date assignment problem with deteriorating jobs and resource-

dependent processing times," *International Journal of Advanced Manufacturing Technology*, vol. 67, no. 1–4, pp. 255–260, 2013.

[16] W. Liu, Y. Jin, and M. Price, "New meta-heuristic for dynamic scheduling in permutation flowshop with new order arrival," *International Journal of Advanced Manufacturing Technology*, vol. 98, no. 5–8, pp. 1817–1830, 2018.

[17] C. Koulamas, "Common due date assignment with generalized earliness/tardiness penalties," *Computers & Industrial Engineering*, vol. 109, pp. 79–83, 2017.

[18] X. Sun, X.-N. Geng, J.-B. Wang, and F. Liu, "Convex resource allocation scheduling in the no-wait flowshop with common flow allowance and learning effect," *International Journal of Production Research*, vol. 57, no. 6, pp. 1873–1891, 2018.

[19] G. Fu, M. Liu, J.-J. Wang, and Y.-Y. Lu, "No-wait two-machine permutation flow shop scheduling problem with learning effect, common due-date and controllable job processing times," *International Journal of Production Research*, vol. 56, no. 6, pp. 2361–2369, 2018.

[20] C. Lu, L. Gao, X. Li, and S. Xiao, "A hybrid multi-objective grey wolf optimizer for dynamic scheduling in a real-world welding industry," *Engineering Applications of Artificial Intelligence*, vol. 57, pp. 61–79, 2017.

[21] L. Zhou, L. Zhang, B. R. Sarker, Y. Laili, and L. Ren, "An event-triggered dynamic scheduling method for randomly arriving tasks in cloud manufacturing," *International Journal of Computer Integrated Manufacturing*, vol. 31, no. 3, pp. 318–333, 2017.

[22] P. Yan, S. Q. Liu, T. Sun, and K. Ma, "A dynamic scheduling approach for optimizing the material handling operations in a robotic cell," *Computers & Operations Research*, vol. 99, pp. 166–177, 2018.