

Research Article

A Study of Flexible Flow Shop Scheduling Problem with Various Heterogeneous Labors

Rong-Hwa Huang ¹, Shao-Jung Chang,¹ and Shun-Chi Yu²

¹Department of Business Administration, Fu Jen Catholic University, New Taipei City, Taiwan, China

²Bachelor's Program in Business Management, Fu Jen Catholic University, New Taipei City, Taiwan, China

Correspondence should be addressed to Rong-Hwa Huang; 026299@mail.fju.edu.tw

Received 22 January 2021; Revised 27 May 2021; Accepted 17 June 2021; Published 12 July 2021

Academic Editor: Mingwei Lin

Copyright © 2021 Rong-Hwa Huang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This study considers the necessity of hiring heterogeneous labors. So far, many studies focus more on manufacturing of equipment and control systems in the intelligent production planning. In fact, the regular labors' processing time may be affected by the external factors and the disabled labors by the mentally handicapped. Therefore, this study sets the processing time of the two types of labors as fuzzy sets. The extra processing time from overtime generated by physical deterioration of old-aged labors is equal to the processing time of regular labors multiplied by the physical deterioration rate of old-aged labors on machine. The coefficient of the cost function is the stepwise function of cost structure. Besides, the dispatching rule based on floating time utilizes ant colony optimization to minimize the makespan. The data test results indicate that the proposed algorithm can efficiently dispatch and schedule the operations, with the average improvement ratio about 11.76%, and demonstrates high capability for the intelligent production planning

1. Introduction

In recent years, the value of old-age population over the age of 65 in the production line attracts more attention, as companies have been faced with the lack of skilled labors. Operations scheduling pursues to optimize resources and thus improves production efficiency and customers' satisfaction. Belabid et al. [1] indicated that each job, represented by a set of operations, generates the path to be followed in the production scenario and considers a set $N = \{1, \dots, n\}$ of n independent jobs that must be treated on a set of $M = \{1, \dots, m\}$ machines within the same order [2, 3]. Many alike production treatments can be summarized as a flexible flow shop scheduling problem, representing the presence of parallel machines in some or all stages; thus, each stage of each job can be treated on several similar machines, and the processing time of each machine is different [4, 5]. The problem definition of this study is to solve the flexible flow shop scheduling problem with various heterogeneous labors, consisting of the regular, the disabled, and old-aged labors,

for better arranging human power and limited resources. Thus, many studies focus more on manufacturing of equipment and control systems in the intelligent production planning, although it is observed that in some cases the processing times of operations can be controlled by allocating available resources [6]. Basically, the controlled processing times focuses on machine operational capacity or device parameter for processing work-in-piece; however, the uncertainty processing time in this study considers the different processing conditions like machine amounts, heterogeneous labors, or jobs with different materials. Notwithstanding, well-constructed medical care and health networks have extended the added value of the old-aged labors. The United Nations adopted the Convention on the Rights of Persons with Disabilities (CRPD) in 2006 [7]. Generally, the type of production management combines approaches of distributing operational productions to decrease makespan cost, inventory cost, wages, and other operational costs. In order to minimize those costs, the employment of the disabled and older labors becomes an

alternative production choice. Meanwhile, many useful methods exist for solving scheduling problems and have various results [8].

As far as all the scheduling problems are concerned, flexible flow shop scheduling is seen as the most popular production condition in the real manufacturing industry, such as in the pharmaceutical factory, mouldmaking, steel mill, and accessories shop processing scenarios. In flexible flow shop problems, various jobs go through a pre-determined path and the sequence of using machines are the same for all jobs [9]. A well-arranged heterogeneous labor in the flow shop scheduling can efficiently decrease the excessive costs compared with competitors with the intelligent production planning. That is, the flow shop is an important way of production for daily necessities. The flow shop scheduling involving multiple agents has been investigated by more and more researchers because of its importance in realistic manufacturing and service systems [10, 11].

Baker [12] indicated that common approach taken by most researchers is to use either an integer linear program or mixed integer linear program to find a minimum makespan solution. Huq et al. [13] pointed out that optimization algorithms did not always produce reasonable results when studying job shop and flow shop problems [14]. Generally, n jobs must be processed through m available machines in a flow shop scheduling problem. The jobs follow the pattern flow of machines. If the jobs follow the same sequence through machines, the problem is defined as a permutation flow shop [15]. Therefore, a multiobjective problem considering flexibility of job processing machines in a flexible flow shop (FFS) is considered using manufacturing simulation; results show the optimal resource assignment and optimal job sequence with efficiency and productivity. In addition, this method can minimize makespan, the number of late jobs, total flow time, and total weighted flow time in the flexible flow shop.

As for heterogeneous labors, most studies considered unrelated machines for solving the attempted scheduling problems. Adapting an exact branch-and-price algorithm to a parallel machine scheduling problem on unrelated machines with sequence- and machine-dependent setup times, and parallelizing the adapted algorithm by implementing a distributed-memory parallelization with a master/worker approach, the conducting extensive computational experiments showed that the efficiency of parallelization approach can lead to superlinear speedup but can vary substantially between instances [16–18]. Researchers have developed increasingly efficient dispatching methods, heuristics, and algorithms for solving different scheduling problems. Scheduling considering learning effect is applied to solve a comparison of two stage-based hybrid algorithms for a batch scheduling problem in hybrid flow shop [19]. Blum and Miralles [20] proposed an algorithm based on beam search for solving the Assembly Line Worker Assignment and Balancing Problem (ALWABP) with the objective of minimizing the cycle time when given a fixed number of workstations and workers. The result proved the efficiency and effectiveness of the method to solve the heterogeneous stations and workers.

In regard to uncertainty processing time, Daniels and Kouvelis [21] applied the best worst-case performance compared to the corresponding optimal solution over all potential realization of job processing time, which was a more appropriate guide to schedule selection, besides the vague job processing times using triangular and trapezoidal fuzzy numbers in job shop production systems, in which the fuzzy makespans and fuzzy mean flow times are then calculated for greater decision-making information [22]. Joo et al. [23] developed an iterative job sequencing algorithm for solving a multilevel job scheduling problem in a single machine under uncertainty processing time that can be used for a given order sequence. Compared with the heuristic method and the unknown processing time fitting the practical situation, the results confirm that the proposed metaheuristic method with or without the iterative job sequencing algorithm indicates good performance [24]. It is reasonable to develop novel models and algorithms based on uncertainty processing time so as to solve uncertainty job shop scheduling problems [25–29]. Besides, stochastic processing time is common in literature [30, 31]. Rodrigues et al. [32] posited that many techniques cope with uncertainty, namely, deterministic models with inventory buffers, robust optimization, and stochastic programming, and thus models incorporating conditional value-at-risk measures are mostly considered. A fuzzy number which is essentially a generalized interval can represent this processing time interval exactly and naturally, though processing times can be more accurately represented as intervals with the most possible completion time somewhere amid the interval. Silva and de Souza [33] demonstrated an approximate dynamic programming with integer programming model, which outperformed the efficiency and effectiveness especially on solving the attempted problem, under uncertainties inherent to the surgical processes. Thus, this study applies fuzzy processing time for approximating the real world, instead of crisp value.

As to algorithms, Huang and Lin [34] adopted a new bee colony optimization algorithm using idle-time-based filtering scheme based on the processes of the Forward Pass of bee's foraging behaviors. That is, less machine idle time efficiently saves resources and production costs. Besides, Engin et al. [35] demonstrated an efficient genetic algorithm (GA) program using a full factorial experimental design to solve the hybrid flow shop scheduling with multiprocessor task problem. Computational results indicated that the proposed approach efficiently solves the problem of makespan. Nevertheless, many algorithms suffer limitations owing to the varied nature of problems that can be encountered. Zhen et al. [36] formulated a mixed integer linear programming model and developed an improved particle swarm optimization algorithm (IPSO) to solve the problem raised by hybrid electric vehicles (HEVs) with the advantages of low energy consumption and pollution, while their disadvantage mainly lies in their limited continuous driving range. The IPSO outperformed other approaches with efficiency and effectiveness. Thus, Dorigo et al. [37] developed the ant system using a population of cooperating agents and communicating via a distributed and shared memory and

proved that the proposed algorithm can be applied to combinatorial optimization problems. Ant colony optimization (ACO) has an excellent ability to solve scheduling problems in the literature [14, 38, 39].

This study examines the flexible flow shop scheduling with various heterogeneous labors using ant colony optimization, as this problem is more popular and real than most other problems. In fact, the regular labors' processing time may be affected by the external factors and the disabled labors by the mentally handicapped. Therefore, this study sets the processing time of the two types of labors as fuzzy sets. Besides, this problem is complicated and worthy of verifying. This study compared the feasible solutions that are acceptable by different papers using the same effectiveness improvement rate proposed by this study. Singh and Singh [40] demonstrate 0.8% and 5.64% of the average relative effectiveness improvement rates of HGWOSCA against PSO and GWO, respectively. Chou et al. [41] show 61.65% and -1.68% of the average relative error rates of MABC against PSO and GA, respectively. Gupta and Srivastava [42] posited their testing results that demonstrate that ACO attains global minima in lesser number of iterations with a lower value

$$FF_c \left| Z_{ijW_hM_k}^{rw}, Z_{ijW_hM_k}^{dw}, Z_{ijW_hM_k}^{ow}, e_r, e_d, e_o \right| C_{\max}(e_r, e_d, e_o) \cdot (C_o \cdot m - S_d \cdot e_d - S_o \cdot e_o), \quad (1)$$

where FF_c represents a flexible flow shop environment involving k machines, $Z_{ijW_hM_k}^{rw}$, $Z_{ijW_hM_k}^{dw}$, $Z_{ijW_hM_k}^{ow}$ represent the regular, disabled, and old-aged labor, respectively, e_r, e_d, e_o represent the quantity of the regular, disabled, and old-aged labor, respectively, and $C_{\max}(e_r, e_d, e_o) \cdot (C_o \cdot m - S_d \cdot e_d - S_o \cdot e_o)$ demonstrates that the objective is to minimize the product of makespan considering the number of hired regular, disabled, and old-aged labors and deduction of subsidies of the disabled and old-aged labors from unit cost generated by incremental makespan.

2.2. Notification. n = total number of jobs

m = total number of machines

w = total number of stations

o = total number of operations to finish

J_i = job of number i , $i = 1, 2, \dots, n$

O_{ij} = the j th operation of J_i

W_h = the h th station of w

W_hM_k = machine of number k in the h th station

$O_{ijW_hM_k} = O_{ij}$ is processed on M_k in the h th station

$FFH_{W_hM_k}$ = number of free float time on M_k in the h th station

$IFH_{W_hM_k}$ = number of interfering float time on M_k in the h th station

$U_{W_hM_k}$ = assigned number of operations on M_k in the h th station

when compared with PSO. However, each method has a distinct applicable area and is favorable to a particular domain based on its features. The motivation of using ACO in this study is the consideration of various heterogeneous labors, fuzzy processing time, and stepwise function of cost structure, not just comparisons among methods. Besides, the dispatching rule based on floating time utilizes ant colony optimization to minimize the makespan. Thus, ACO is proposed to solve this complex problem. This study applies an efficient dispatching rule, fuzzy processing time, and three heterogeneous labors using ACO to search for minimization of makespan. Also, the dispatching rule can prompt ants efficiently identify the next route. This performance reinforces the solution capacity of ACO and outperforms most of trade-off problems for future studies.

2. Problem definition

Related definitions and notifications are as follows.

2.1. Definition.

rw = the amount of regular labor

dw = the amount of disabled labor

ow = the amount of old-aged labor

e_r = number of hired regular labor

e_d = number of hired disabled labor

e_o = number of hired old-aged labor

E = number of total hired heterogeneous labors

$W_h e_r$ = number of hired regular labors in the h th station

$W_h e_d$ = number of hired disabled labors in the h th station

$W_h e_o$ = number of hired old-aged labors in the h th station

C_0 = unit cost generated by incremental makespan

S_d = the coefficient of subsidy per hour for hiring a disabled labor

S_o = the coefficient of subsidy per hour for hiring an old-aged labor

$C_{\max}(e_r, e_d, e_o)$ = makespan considering number of hired regular, old-aged, and disabled labor

A_i = number of operations to finish J_i

B = a big number

f = the physical deterioration rate of old-aged labors

$S_{ijW_hM_k}$ = starting time of $O_{ijW_hM_k}$ in the h th station

$p_{ijW_hM_k}$ = processing time of $O_{ijW_hM_k}$ in the h th station

$F_{ijW_hM_k}$ = completion time of $O_{ijW_hM_k}$ in the h th station

$p_{ijW_hM_k, rw}$ = processing time for regular labor to process $O_{ijW_hM_k}$ in the h th station

$p_{ijW_hM_k, dw}$ = processing time for disabled labor to process $O_{ijW_hM_k}$ in the h th station

$p_{ijW_hM_k, ow}$ = processing time for old-aged labor to process $O_{ijW_hM_k}$ in the h th station

u_{ijow} = the extra processing time from overtime generated by physical deterioration of old-aged labors

$X_{ijW_hM_k} = \begin{cases} 1, & \text{if } O_{ij} \text{ is processed on } M_k \text{ in the } h\text{th station,} \\ 0, & \text{otherwise} \end{cases}$

$Y_{ijlqW_hM_k} = \begin{cases} 1, & \text{if } O_{ij} \\ & \text{is processed before } O_{lq} \text{ on } M_k \text{ in the } h\text{th station,} \\ 0, & \text{otherwise} \end{cases}$

$Z_{ijW_hM_k}^{rw} = \begin{cases} 1, & \\ & \text{if } M_k \text{ is processed by regular labor in the } h\text{th station,} \\ 0, & \text{otherwise} \end{cases}$

$Z_{ijW_hM_k}^{dw} = \begin{cases} 1, & \text{if } M_k \text{ is processed by disabled labor} \\ & \text{in the } h\text{th station,} \\ 0, & \text{otherwise} \end{cases}$

$Z_{ijW_hM_k}^{ow} = \begin{cases} 1, & \text{if } M_k \text{ is processed by old - aged labor} \\ & \text{in the } h\text{th station,} \\ 0, & \text{otherwise} \end{cases}$

T_{ijrw} = the unit cost for processing O_{ij} by hired regular labors

T_{ijdw} = the unit cost for processing O_{ij} by hired disabled labors

T_{ijow} = the unit cost for processing O_{ij} by hired old-aged labors

2.3. *Mathematical Model.* Formulation is as follows:

$$\begin{aligned} \min Z &= C_{\max}(e_r, e_d, e_o) \cdot (C_o \cdot m - S_d \cdot e_d - S_o \cdot e_o), \\ \min Z &= C_{\max}(e_r, e_d, e_o) \cdot (C_o \cdot m - S_d \cdot e_d - S_o \cdot e_o), \end{aligned} \quad (2)$$

subject to

$$C_{\max}(e_r, e_d, e_o) = \max_{1 \leq k \leq m} (F_{ijW_hM_k}), \quad \forall i = 1, 2, \dots, n; j = 1, 2, \dots, Q_i; h = 1, 2, \dots, w; k = 1, 2, \dots, m, \quad (3)$$

$$F_{ijW_hM_k} = S_{ijW_hM_k} + p_{ijW_hM_k}, \quad \forall i = 1, 2, \dots, n; j = 1, 2, \dots, Q_i; h = 1, 2, \dots, w; k = 1, 2, \dots, m, \quad (4)$$

$$F_{i(j-1)W_hM_k} \leq S_{ijW_hM_k}, \quad \forall i = 1, 2, \dots, n; j = 1, 2, \dots, Q_i; h = 1, 2, \dots, w; k = 1, 2, \dots, m, \quad (5)$$

$$p_{ijW_hM_k} = \begin{cases} p_{ijW_hM_k, rw} & \text{if } Z_{ijW_hM_k}^{rw} = 1, \\ p_{ijW_hM_k, ow} & \text{if } Z_{ijW_hM_k}^{ow} = 1, \\ p_{ijW_hM_k, dw} & \text{if } Z_{ijW_hM_k}^{dw} = 1, \end{cases} \quad \forall i = 1, 2, \dots, n; j = 1, 2, \dots, Q_i; h = 1, 2, \dots, w; k = 1, 2, \dots, m, \quad (6)$$

$$F_{ijW_hM_k} - B(1 - Y_{ijlqW_hM_k}) \leq S_{lqW_hM_k}, \quad \forall i = 1, 2, \dots, n; j = 1, 2, \dots, Q_i; h = 1, 2, \dots, w; k = 1, 2, \dots, m; l = 1, 2, \dots, n; q = 1, 2, \dots, Q_o; i \neq l, \quad (7)$$

$$F_{lqW_hM_k}(1 - Y_{ijlqW_hM_k}) - B \cdot Y_{ijlqW_hM_k} \leq S_{ijW_hM_k}, \quad \forall i = 1, 2, \dots, n; j = 1, 2, \dots, Q_i; h = 1, 2, \dots, w; k = 1, 2, \dots, m; l = 1, 2, \dots, n; q = 1, 2, \dots, Q_o; i \neq l, \quad (8)$$

$$\sum_k X_{ijW_hM_k} = 1, \quad \forall i = 1, 2, \dots, n; j = 1, 2, \dots, Q_i; h = 1, 2, \dots, w; k = 1, 2, \dots, m, \quad (9)$$

$$P_{ijW_hM_k, ow} = p_{ijW_hM_k, rw} + u_{ijow}, \quad \forall i = 1, 2, \dots, n; j = 1, 2, \dots, Q_i; h = 1, 2, \dots, w; k = 1, 2, \dots, m, \quad (10)$$

$$u_{ijow} = p_{ijW_hM_k, rw} \cdot f, \quad \forall i = 1, 2, \dots, n; j = 1, 2, \dots, Q_i; h = 1, 2, \dots, w; k = 1, 2, \dots, m. \quad (11)$$

The meanings of all objectives and constraints are explained as follows:

In (2), this model minimizes the product of makespan considering the number of hired regular, old-aged, and disabled labors and deduction of subsidies of the disabled and old-aged labors from unit cost generated by

incremental makespan on machine M_k in the h th station.

(3), makespan considering the number of hired regular, old-aged, and disabled labors represents the maximum completion time among all the machines on machine M_k in the h th station.

In (4), completion time of $O_{ijW_hM_k}$ is the sum of its starting time and processing time on machine M_k in the h th station.

In (5), for job, $O_{ijW_hM_k}$ is the starting time of the j th operation exceeds the completion time of the $(j-1)$ th operation on machine M_k in the h th station.

In (6), the processing time of $O_{ijW_hM_k}$ is decided by different labors on machine M_k in the h th station.

In (7), $O_{ijW_hM_k}$ is processed before $O_{lqW_hM_k}$, and thus, the completion time of $O_{ijW_hM_k}$ is smaller than the starting time of $O_{lqW_hM_k}$ on machine M_k in the h th station.

In (8), $O_{ijW_hM_k}$ is processed after $O_{lqW_hM_k}$, and thus, the completion time of $O_{lqW_hM_k}$ is smaller than the starting time of $O_{ijW_hM_k}$ on machine M_k in the h th station.

In (9), O_{ij} is only processed on one machine once in the h th station.

In (10), completion time of O_{ij} is the sum of its starting time and extra processing time from overtime generated by physical deterioration of old-aged labors on machine M_k in the h th station.

In (11), extra processing time from overtime generated by physical deterioration of old-aged labors is equal to the processing time of regular labors multiplied by the physical deterioration rate (100/85) of old-aged labors on machine M_k in the h th station.

3. ACO, Fuzzy Processing Time, and Dispatching Rule

3.1. Ant Colony Optimization. Ant colony optimization is derived from the ant system (AS) first introduced by [43]. The development and applications of ACO grew rapidly since then. Dorigo et al. [44] applied AS to solve the traveling salesman problem, quadratic assignment problem, and scheduling problem. Subsequently, Dorigo and Gambardella [45] improved AS to create ant colony system (ACS). Finally, Dorigo and Di Caro [46] combined the properties of AS and ACS and developed ant colony optimization (ACO). A detailed introduction to ACO can be found in [47].

ACO simulates the behavior of ants searching for food. Each ant uses exploitation and exploration for route selection. The state transition rule is used to balance exploitation and exploration behaviors:

$$s = \begin{cases} \arg \max_{u \in J_k(r)} \{ [\tau(r, u)]^\alpha \cdot [\eta(r, u)]^\beta \}, & \text{if } q \leq q_0, \\ P_k(r, s), & \text{otherwise,} \end{cases} \quad (12)$$

$$P_k(r, s) = \begin{cases} \frac{[\tau(r, s)]^\alpha \cdot [\eta(r, s)]^\beta}{\sum_{u \in J_k(r)} [\tau(r, u)]^\alpha \cdot [\eta(r, u)]^\beta}, & s \in J_k(r), \\ 0, & \text{otherwise.} \end{cases} \quad (13)$$

The upper half of (12) comprises artificial ant exploitation behavior. When $q \leq q_0$, the ant will choose to progress to a node with the largest quantity of pheromones. q denotes a random number, and q_0 represents a constant. Both q and q_0 are real numbers between 0 and 1. Otherwise, the artificial ant uses exploration. As in (13), the quantity of pheromones determines the likelihood of an ant moving to the adjacent nodes. Ants perform route selection using the state transition rule. τ denotes the quantity of pheromones between two nodes (r, u) . Furthermore, η represents the greedy heuristic, which equals the inverse of the distance between nodes (r, u) . $J_k(r)$ denotes the set of unexplored adjacent nodes. Furthermore, β is a constant which controls the relative importance of pheromone amount and greedy heuristic.

When an ant chooses a complete route, the route can be transferred into a feasible schedule and objective value. The ant then leaves pheromone based on the objective value. The local pheromone update of ants is as shown in the following:

$$\tau(r, s) := (1 - \rho) \cdot \tau(r, s) + \rho \cdot \Delta\tau(r, s). \quad (14)$$

Local pheromone update prevents ants from solely selecting the best-known route, not exploring potential better routes, and falling into a local optimum. ρ denotes local pheromone evaporation rate and comprises a number between 0 and 1. $\Delta\tau(r, s)$ represents the amount of pheromone increased between nodes (r, s) , as shown in (16). Q is a positive constant. Finally, L_k represents the objective value derived from the ant selected route.

Following all ants in the present iteration select their route, the ant selecting the best route will initiate a global pheromone update to increase the influence of good route in the succeeding iterations. The following equation shows the global pheromone update:

$$\tau(r, s) := (1 - \sigma) \cdot \tau(r, s) + \sigma \cdot \Delta\tau(r, s), \quad (15)$$

$$\Delta\tau(r, s) = \begin{cases} \frac{Q}{L_k}, & \text{if } (r, s) \in \text{global - best - tour,} \\ 0, & \text{otherwise,} \end{cases} \quad (16)$$

where σ denotes the global pheromone evaporation rate and comprises a number between 0 and 1.

3.2. Fuzzy Processing Time. Concerning fuzzy processing time, Balin [30] posited various factors concerned in practical scheduling problems that were often uncertain or imprecise in nature. This is especially real when human-made factors are comprised of the problems. A fuzzy number is virtually a generalized interval that can indicate the processing time interval actually and naturally [48]. Fuzzy makespans and fuzzy mean flow times are then computed for greater decision-making message. By retaining the fuzziness throughout the analysis process, the decision-maker holds the whole message useful for subsequent sensitivity analysis of the capability criteria. Thus, this study utilizes fuzzy processing time for approximating the practical world, instead of crisp value.

In order to meet the actual production situation, the working time is set as a fuzzy set. The most likely time for the completion of a job is called the most likely time (MLT), the minimum time required for a job to complete successfully is called the optimistic time (OT), and the longest time is called the most pessimistic time (PT). And the most optimistic time for the old-aged and disabled labors may be smaller than that of the most pessimistic time of the regular ones. Figure 1 showed the graph of the fuzzy processing time range.

In terms of cost structure, this study assumes that fixed and intangible costs have been depreciated and amortized and only considers labors' salary. In practice, when the completion time exceeds the expected completion due date, extra costs such as the overtime cost of the workers and the penalty cost of the tardiness will be derived. Therefore, the cost coefficient is set to a stepwise type, that is, when the maximum completion time exceeds the completion due date. A certain percentage of overtime will lead to an increase in the cost factor; this assumption complies with the practical situation.

Before the assignment phase, defuzzification must be performed in advance, and the fuzzy set is converted into a crisp value, and then the subsequent operations are performed. The defuzzification process uses the triangular fuzzy number ordering condition proposed by [40]. The formula is as follows:

$$\begin{aligned} C(\tilde{A}) &= \frac{a_1 + 2a_2 + a_3}{4}, \\ C(\tilde{A}) &= \frac{a_1 + 2a_2 + a_3}{4}. \end{aligned} \quad (17)$$

3.3. Dispatching Rule. Concerning dispatching rule, this study calculates the makespan considering all hired regular labors using ACO, identifying the earliest and latest time span, and proposes a novel dispatching rule based on backward scheduling adopting the free float time (FF) and interfering float time (IF) to fit the lowest costs synchronously:

$$\begin{aligned} D_{1k} &= \sum_{i=1}^n \sum_{j=1}^o \frac{X_{ijW_hM_k} FF_{ijW_hM_k}}{f_{W_hM_k}}, \quad \forall k = 1, 2, \dots, m, \\ D_{2k} &= \frac{FFH_{W_hM_k}}{f_{W_hM_k}}, \quad \forall k = 1, 2, \dots, m, \\ D_{3k} &= \sum_{i=1}^n \sum_{j=1}^o \frac{X_{ijW_hM_k} IF_{ijW_hM_k}}{f_{W_hM_k}}, \quad \forall k = 1, 2, \dots, m, \\ D_{4k} &= \frac{IFH_{W_hM_k}}{f_{W_hM_k}}, \quad \forall k = 1, 2, \dots, m, \end{aligned} \quad (18)$$

where D_{1k} represents the average free float time of machine k in the h th station, and its calculation method is "the sum of free float time for machining in machine k , divided by the number of operations with free floats on machine k in the

h th station"; D_{2k} means "the ratio of the number of operations with free floats on machine k to the total number of operations on that machine in the h th station"; D_{3k} represents the average interfering float time of the machine k in the h th station, and the calculation method is "the sum of interfering float time for machining in machine k , divided by the number of operations with interfering floats on machine k in the h th station"; D_{4k} means "The ratio of the number of operations with interfering floats on machine k in the h th station to the total number of operations on that machine." The assignment is carried out by stages, and each stage t is screened according to D_{tk} from large to small in descending order. If D_{1k} in the first stage is the same, the second stage is performed, and so on. In order to keep the completeness of the assignment process, if the value is still the same in the fourth stage, D_{5k} is performed, and D_{5k} indicates that one machine is arbitrarily selected from the machine in the h th station that was not yet assigned with labors. After being assigned in the five stages above, the dispatching rule offers initial solution of the ant colony algorithm, which can avoid excessive calculating times and difficulty in obtaining the best solution under large scale of production, and searches for the better solution within polynomial time. Even if the best solution is not obtained, the better solution would not deviate too much from the minimum.

In terms of stepwise function of cost structure, the ratio of cost coefficient is based on the minimum hourly wage of the current Labor Standard Laws in Taiwan. The hourly wage is set at 15 units according to the cost ratio. The government subsidy factor for the old-aged and disabled labors is set to an hourly subsidy of 4.5 units and 7 units, respectively. The stepwise function of cost structure factor is set at 36 hours as the due date, for example. Figure 2 shows when the completion time exceeded the due date, which is 36 hours, about more than 11%, which equals to 4 hours, will increase the cost factor by 4 units.

The stepwise cost function is mainly affected by two factors: "extra unit cost from overtime" and "extra multiple from overtime." First, "overtime" is defined as the extra time of completion time of the total hired regular labors. That is, completion time is used as a benchmark, which means that working time beyond this benchmark is considered "overtime."

The "extra unit cost from overtime" is the additional cost factor per unit time for overtime work. Therefore, the cost value of the regular labor's completion time will not be affected by the "extra unit cost from overtime," which mainly affects the rise of each level in the step function graph. The "extra multiple from overtime" is defined as 100% of the total completion time. After hiring a heterogeneous labor, each time a certain percentage is exceeded, the "extra unit cost from overtime" rate will be stepped. The formula is incremented by 1 time, and the ratio can be adjusted according to the actual situation. The "extra cost of overtime" is represented by the cost of "overtime" times "extra unit cost from overtime." The total cost for production is the result of the "unit hour cost for production" times "completion time of the total hired regular labors" and "extra cost of overtime." (Algorithm 1) [43]

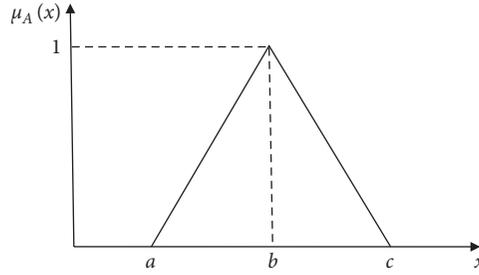


FIGURE 1: The graph of the fuzzy processing time range.

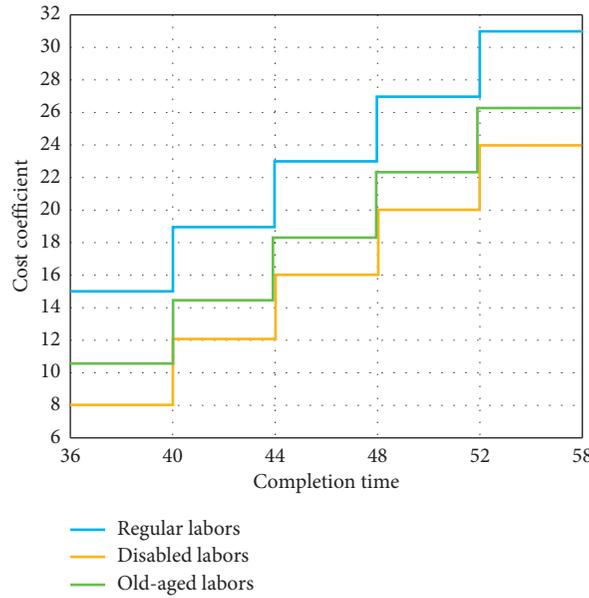


FIGURE 2: Stepwise function of cost structure.

3.4. *Algorithm.* This section presents small-scale and large-scale simulated data tests to confirm the solution capacity of ACO. Substantial evidence based on test results demonstrates that ACO solves the attempted problem with efficiency.

4. Data Testing and Analysis

4.1. *Data Generation.* This study examines the following subject scheduling problem of $FF_c | Z_{ijW_k M_k}^{rw}, Z_{ijW_k M_k}^{dw}, Z_{ijW_k M_k}^{ow}, e_r, e_d, e_o | C_{\max}(e_r, e_d, e_o) \cdot (C_o \cdot m - S_d \cdot e_d - S_o \cdot e_o)$. The following parameters are considered in generating test data: FF_c represents a flexible flow shop environment involving k machines; $Z_{ijW_h M_k}^{rw}, Z_{ijW_h M_k}^{dw}, Z_{ijW_h M_k}^{ow}$ represent the regular, disabled, and old-aged labor, respectively; e_r, e_d, e_o represent the quantity of the regular, disabled, and old-aged labor, respectively; $C_{\max}(e_r, e_d, e_o) \cdot (C_o \cdot m - S_d \cdot e_d - S_o \cdot e_o)$ demonstrates that the objective is to minimize the product of makespan considering number of hired regular, disabled, and old-aged labors and deduction of subsidies of the disabled and old-aged labors from unit cost generated by incremental makespan. Parameters related to ACO algorithms include the following: α is the weight

coefficient of pheromone influence, β is the weight coefficient of visibility function, σ is a global pheromone evaporation coefficient, ρ is a local pheromone evaporation coefficient, q_0 is exploration/exploitation control ratio, and z is number of ants in each iteration and iterations (loop).

The parameter settings in the proposed algorithm are decided during pretest, as listed in Table 1. All tests are performed on a personal notebook with Intel R Core TM i5-3210M CPU @ 2.50 GHz, 4.0 GB RAM, and the OS version is Windows 10 R 64 bit. Finally, the test results are compared and analyzed.

Regarding the setting of the weights of measuring criteria, the real business decision process is taken into consideration. Companies requirements in terms of criteria measurement weights vary according to their priorities. According to the feature of the attempted problem, this study adopts various scale number of machines and jobs. The ranges from small to large-scale number are as follows: scale (1): $w = 3, n = 5$; scale (2): $w = 4, n = 10$; scale (3): $w = 5, n = 15$; and scale (4): $w = 6, n = 20$. Due to limitation of content, this study lists $w = 3, n = 5$ for small scale and $w = 6, n = 20$ for demonstration of large scale, respectively. The item “E” in Table 2 indicates the number of hired heterogeneous labors,

- (1) Procedure ACO
- (2) Select an initial point
 - Iteration $t = 1$
 - Spread pheromones for the first iteration
- (3) Do until reaching the stopping criteria
- (4) For each ant
 - (5) For each job
 - (6) If job $\neq 1$ then evaluate the initial point end if
 - (7) Load possible ending points in the last day for job j
 - (8) Generate the graph for the problem
 - (9) For each condition of dispatching rule
 - (10) Computes the visibility η for each possible move
 - (11) Draw the next job to be processed considering the probabilities.
 - (12) Updated the sparse graphs
 - (13) Next condition of dispatching rule
 - (14) Next job
 - (15) Apply the local search procedure
 - (16) Save the solution if it is the best one
 - (17) Computes the quantity of pheromones to be spread by the ant i in each arc, job and day.
 - (18) Next ant
 - (19) update the pheromones using the variation and $\sigma \ t = t + 1$
- (20) Loop
- (21) Return the best solution found

ALGORITHM 1: Algorithm of ACO.

TABLE 1: Parameters setting of algorithm.

Parameter	α	β	σ	ρ	q_0	z	Loop
Value	1	1	0.1	0.1	0.5	10	200

and “R,” “O,” and “D” in dispatching rule represent the assigned regular, old-aged, disabled labor, respectively, and the item “ms” means microsecond of CPU.

As shown in Table 2 and Figure 3, the assigned sequence considering degree of slack is $W_3M_2 \rightarrow W_1M_3 \rightarrow W_2M_2 \rightarrow W_1M_2 \rightarrow W_2M_1 \rightarrow W_1M_1 \rightarrow W_3L_1$. When hiring a heterogeneous labor, the best assignment is RRR RR RD. That is, the total cost is deduced from 6352.50 to 6070.37 when a heterogeneous labor is hired; when hiring two, the best assignment is RRD RR RD; when hiring three, the best assignment is RRD RD RO, and the results demonstrate that the hiring of heterogeneous labors surely decreases costs with efficiency. Since the lowest cost is 5262.88, the cost increases with the increasing amount of heterogeneous labors hired. Therefore, under the condition of $w = 3$, $n = 5$, and $m = 7$, it is recommended to hire 3 heterogeneous labors including 2 disabled labors to operate in W_1M_3 and W_2M_2 , 1 old-aged labor to operate in W_3M_2 , and regular labors to operate in W_1M_1 , W_1M_2 , W_2M_1 , and W_3M_1 , which generates the most efficiency of saving costs. The results reveal that production manager can choose the better way to hire appropriate amount of heterogeneous labors for cost-saving and to minimize completion time using the proposed dispatching rule. Besides, ACO solves the attempted problem with efficiency. The arrangement of labors, jobs, and machines is both effective and efficient for companies to apply in the real world.

As shown in Table 3 and Figure 4, the assigned sequence considering degree of slack is $W_1M_2 \rightarrow W_6M_2 \rightarrow W_6M_1 \rightarrow W_2M_1 \rightarrow W_6M_4 \rightarrow W_2M_3 \rightarrow W_4M_1 \rightarrow W_3M_3 \rightarrow W_5M_3 \rightarrow W_6M_3 \rightarrow W_4M_2 \rightarrow W_4M_3 \rightarrow W_3M_1 \rightarrow W_1M_3 \rightarrow W_3M_2 \rightarrow W_5M_1 \rightarrow W_2M_4 \rightarrow W_1M_1 \rightarrow W_5M_2 \rightarrow W_2M_2$. When hiring a heterogeneous labor, the best assignment is RDR RRRR RRR RRR RRR RRRR, when hiring two, the best assignment is RDR RRRR RRR RRR RRR RDRR, when hiring three, the best assignment is ROR RRRR RRR RRR RRR DDRR, and so on; the results demonstrate that the hiring of heterogeneous labors surely decreases costs with efficiency. Since the lowest cost is 156719.61, the cost increases with the hired amount of heterogeneous labors. Therefore, under the condition of $w = 6$, $n = 20$, $m = 20$, it is recommended to hire 9 heterogeneous labors including 6 disabled labors to operate in W_2M_1 , W_2M_3 , W_3M_3 , W_4M_1 , W_5M_3 , and W_6M_4 , 3 old-aged labors to operate in W_1M_2 , W_6M_1 , and W_6M_2 , and regular labors to operate in W_1M_1 , W_1M_3 , W_2M_2 , W_2M_4 , W_3M_1 , W_3M_2 , W_4M_2 , W_4M_3 , W_5M_1 , W_5M_2 , and W_6M_3 , which generates the most efficiency of saving costs. This study just demonstrates the amount of hired heterogeneous labors from 0~7 and 20, due to the limit of content. The results reveal that production manager can choose the better way to hire appropriate amount of heterogeneous labors for cost-saving and to minimize completion time using the proposed dispatching rule.

TABLE 2: The optimal solution and makespan with $w = 3, n = 5$.

Amount of hired heterogeneous labors (E)	Arrangement of labors e_r, e_d, e_o	Dispatching rule			Cmax	Optimum solution (Total cost)	CPU time (ms)
		($w = 1, m = 3$)	($w = 2, m = 2$)	($w = 3, m = 2$)			
0	(7, 0, 0)	RRR	RR	RR	60.5	6352.5	—
1	(6, 0, 1)	RRR	RR	RO	61.56	6186.78	296.86
	(6, 1, 0)	RRR	RR	RD	61.94	6070.37	296.87
2	(5, 0, 2)	RRO	RR	RO	63.37	6083.76	265.62
	(5, 1, 1)	RRD	RR	RO	64.56	6036.36	297.34
3	(5, 2, 0)	RRD	RR	RD	62.62	5698.42	281.24
	(4, 0, 3)	RRO	RO	RO	63.86	5842.96	281.25
	(4, 1, 2)	RRO	RD	RO	64.46	5736.72	249.99
	(4, 2, 1)	RRD	RD	RO	60.84	5262.88	265.76
4	(4, 3, 0)	RRD	RD	RD	66.36	5573.82	296.86
	(3, 0, 4)	ROO	RO	RO	64.68	5626.94	218.74
	(3, 1, 3)	RDO	RO	RO	65.48	5532.64	249.99
	(3, 2, 2)	RDO	RD	RO	66.9	5485.96	249.99
	(3, 3, 1)	RDD	RD	RO	67.97	7306.78	249.99
	(3, 4, 0)	RDD	RD	RD	68.36	7177.28	249.99
	(2, 0, 5)	ROO	OO	RO	67.85	7497.7	297.21
	(2, 1, 4)	ROO	DO	RO	68.09	7354.1	296.87
5	(2, 2, 3)	RDO	DO	RO	67.9	7163.14	328.11
	(2, 3, 2)	RDO	DD	RO	68.41	7045.97	328.11
	(2, 4, 1)	RDD	DD	RO	68.71	6905.1	265.81
	(2, 5, 0)	RDD	DD	RD	69.86	6845.79	296.87
	(1, 0, 6)	OOO	OO	RO	69.77	7395.09	484.36
	(1, 1, 5)	DOO	OO	RO	68.68	7107.86	281.24
6	(1, 2, 4)	DOO	DO	RO	68.01	6869.26	312.49
	(1, 3, 3)	DDO	DO	RO	71.02	6994.98	281.24
	(1, 4, 2)	DDO	DD	RO	69.64	6684.96	234.74
	(1, 5, 1)	DDD	DD	RO	68.67	6420.88	415.25
	(1, 6, 0)	DDD	DD	RD	72.38	6586.49	389.7
	(0, 0, 7)	OOO	OO	OO	71.17	7224.01	292.84
	(0, 1, 6)	OOO	OO	DO	71.17	7046.08	283.21
	(0, 2, 5)	DOO	OO	DO	71.17	6868.15	311.78
7	(0, 3, 4)	DOO	DO	DO	71.17	6690.22	319
	(0, 4, 3)	DDO	DO	DO	78.53	9384.04	285.18
	(0, 5, 2)	DDO	DD	DO	71.46	6359.72	316.43
	(0, 6, 1)	DDD	DD	DO	73.82	8452.39	287.57
	(0, 7, 0)	DDD	DD	DD	77.17	8643.32	338.08

Note. The grey shading represents the minimal makespan among all combinations.

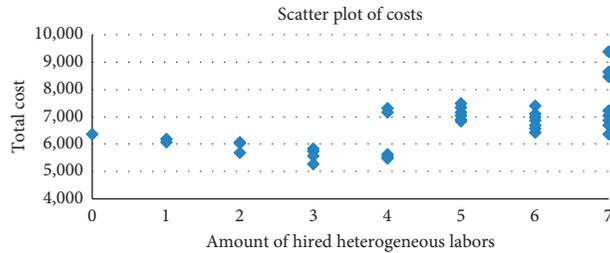


FIGURE 3: Scatter plot of cost structure with $(w) = 3$ and $(n) = 5$.

The results again reveal that production manager can choose the better way to hire appropriate number of heterogeneous labors for cost-saving and to minimize completion time using the proposed dispatching rule. Besides, ACO solves the attempted problem with efficiency. The

arrangement of labors, jobs, and machines is both effective and efficient for companies to utilize in the real world.

As shown in Table 4, the improvement ratios of optimum are up to 17.15%, which demonstrates the percentage of cost-down under the proposed dispatching rule, with

TABLE 3: The optimal solution and makespan with $w = 6, n = 20$.

Amount of hired heterogeneous labors (E)	Arrangement of labors e_r, e_d, e_o	Dispatching rule						Cmax	Optimum solution (Total cost)	CPU time (ms)
		($w = 1, m = 3$)	($w = 2, m = 4$)	($w = 3, m = 3$)	($w = 4, m = 3$)	($w = 5, m = 3$)	($w = 6, m = 4$)			
0	(20, 0, 0)	RRR	RRRR	RRR	RRR	RRR	RRRR	590.5	177150	—
1	(19, 0, 1)	ROR	RRRR	RRR	RRR	RRR	RRRR	587.29	173544.2	5315.63
	(19, 1, 0)	RDR	RRRR	RRR	RRR	RRR	RRRR	589.25	172650.25	5218.08
2	(18, 0, 2)	ROR	RRRR	RRR	RRR	RRR	RORR	605.47	176190.32	5331.14
	(18, 1, 1)	ROR	RRRR	RRR	RRR	RRR	RDRR	599.96	173087.74	5295.04
3	(18, 2, 0)	RDR	RRRR	RRR	RRR	RRR	RDRR	599.43	171436.27	5376.2
	(17, 0, 3)	ROR	RRRR	RRR	RRR	RRR	OORR	611.46	175182.57	5263.86
4	(17, 1, 2)	ROR	RRRR	RRR	RRR	RRR	DORR	615.22	174721.77	5375.03
	(17, 2, 1)	ROR	RRRR	RRR	RRR	RRR	DDRR	600.21	168957.71	5302.12
5	(17, 3, 0)	RDR	RRRR	RRR	RRR	RRR	DDRR	612.15	170789.15	5534.99
	(16, 0, 4)	ROR	ORRR	RRR	RRR	RRR	OORR	613.08	172887.15	5481.58
6	(16, 1, 3)	ROR	DRRR	RRR	RRR	RRR	OORR	595.37	166405.22	5319.75
	(16, 2, 2)	ROR	DRRR	RRR	RRR	RRR	DORR	602.28	166831.56	5470.35
7	(16, 3, 1)	ROR	DRRR	RRR	RRR	RRR	DDRR	602.74	165450.76	5358.12
	(16, 4, 0)	RDR	DRRR	RRR	RRR	RRR	DDRR	599.82	163151.72	5531.05
8	(15, 0, 5)	ROR	ORRR	RRR	RRR	RRR	OORO	616.05	170954.57	5417.81
	(15, 1, 4)	ROR	ORRR	RRR	RRR	RRR	OORD	618.81	170172.75	5599.63
9	(15, 2, 3)	ROR	DRRR	RRR	RRR	RRR	OORD	617.06	168149.53	5392.19
	(15, 3, 2)	ROR	DRRR	RRR	RRR	RRR	DORD	619.57	167282.55	5545.63
10	(15, 4, 1)	ROR	DRRR	RRR	RRR	RRR	DDRD	612.53	163851.11	5641.47
	(15, 5, 0)	RDR	DRRR	RRR	RRR	RRR	DDRD	612.25	162246.25	5452.24
11	(14, 0, 6)	ROR	OROR	RRR	RRR	RRR	OORO	620.49	169393.09	5592.88
	(14, 1, 5)	ROR	ORDR	RRR	RRR	RRR	OORO	613.28	165890.89	5470.4
12	(14, 2, 4)	ROR	ORDR	RRR	RRR	RRR	OORD	614.12	164584.16	5654.9
	(14, 3, 3)	ROR	DRDR	RRR	RRR	RRR	OORD	618.42	164189.18	5562.88
13	(14, 4, 2)	ROR	DRDR	RRR	RRR	RRR	DORD	631.46	166073.32	5736.43
	(14, 5, 1)	ROR	DRDR	RRR	RRR	RRR	DDRD	613.77	159886.43	5522.81
14	(14, 6, 0)	RDR	DRDR	RRR	RRR	RRR	DDRD	623.56	160878.48	5609.58
	(13, 0, 7)	ROR	OROR	RRR	ORR	RRR	OORO	625.11	167842.71	5539.26
15	(13, 1, 6)	ROR	OROR	RRR	DRR	RRR	OORO	625.54	166392.31	5747.1
	(13, 2, 5)	ROR	ORDR	RRR	DRR	RRR	OORO	627.68	165393.02	5707.66
16	(13, 3, 4)	ROR	ORDR	RRR	DRR	RRR	OORD	622.06	162358.31	5529.52
	(13, 4, 3)	ROR	DRDR	RRR	DRR	RRR	OORD	628.93	162577.76	5670.48
17	(13, 5, 2)	ROR	DRDR	RRR	DRR	RRR	DORD	634.34	162391.04	5532.89
	(13, 6, 1)	ROR	DRDR	RRR	DRR	RRR	DDRD	635.2	161023.83	5863.24
18	(13, 7, 0)	RDR	DRDR	RRR	DRR	RRR	DDRD	635.44	159495.44	5635.21
	(0, 0, 20)	OOO	OOOO	OOO	OOO	OOO	OOOO	713.87	207022.88	8159.69
19	(0, 1, 19)	OOO	ODOO	OOO	OOO	OOO	OOOO	715.98	205844.25	8298.79
	(0, 2, 18)	OOO	ODOO	OOO	OOO	ODO	OOOO	713.39	203316.72	8401.87
20	(0, 3, 17)	DOO	ODOO	OOO	OOO	ODO	OOOO	714.9	201958.4	8321.09
	(0, 4, 16)	DOO	ODOD	OOO	OOO	ODO	OOOO	714.99	200197.76	8190.65
21	(0, 5, 15)	DOO	ODOD	OOO	OOO	DDO	OOOO	719.76	199733.96	8045.88
	(0, 6, 14)	DOO	ODOD	ODO	OOO	DDO	OOOO	721.78	256232.97	8086.76
22	(0, 7, 13)	DOD	ODOD	ODO	OOO	DDO	OOOO	720.54	253990	7999.67
	(0, 8, 12)	DOD	ODOD	DDO	OOO	DDO	OOOO	722.76	252966.35	8213.06
23	(0, 9, 11)	DOD	ODOD	DDO	OOD	DDO	OOOO	721.33	250662.52	8260.78
	(0, 10, 10)	DOD	ODOD	DDO	ODD	DDO	OOOO	724.77	250045.31	8469.36
24	(0, 11, 9)	DOD	ODOD	DDO	ODD	DDO	OODO	725.78	248580.68	8293.31
	(0, 12, 8)	DOD	ODOD	DDO	ODD	DDD	OODO	723.89	246123.96	8562.87
25	(0, 13, 7)	DOD	ODOD	DDD	ODD	DDD	OODO	722.77	243935.21	8583.97
	(0, 14, 6)	DOD	ODOD	DDD	DDD	DDD	OODO	726.89	243509.16	8756.91
26	(0, 15, 5)	DOD	ODDD	DDD	DDD	DDD	OODO	725.76	241316.2	8861.01
	(0, 16, 4)	DOD	ODDD	DDD	DDD	DDD	OODD	724.35	239034.51	8684.7
27	(0, 17, 3)	DOD	ODDD	DDD	DDD	DDD	OODD	727.54	238269.68	8798.53
	(0, 18, 2)	DOD	ODDD	DDD	DDD	DDD	DODD	729.49	237084.25	8689.65
28	(0, 19, 1)	DOD	ODDD	DDD	DDD	DDD	DDDD	731.13	235787.81	8759.75
	(0, 20, 0)	DDD	ODDD	DDD	DDD	DDD	DDDD	730.98	233914.24	8860.41

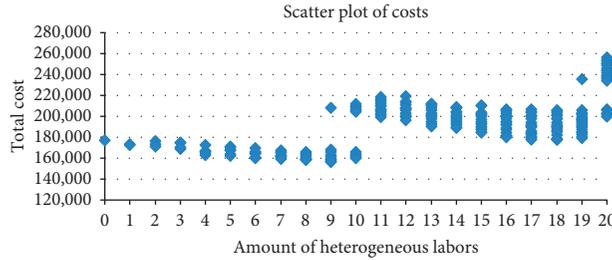


FIGURE 4: Scatter of cost structure with $(w) = 6$ and $(n) = 20$.

TABLE 4: The improvement ratios of optimal solution.

	Scale (1)	Scale (2)	Scale (3)	Scale (4)
Workstation (w)	3	4	5	6
Job (n)	5	10	15	20
Machine (m)	7	12	17	20
Best combination of dispatching	Regular*4, disabled*2, old-aged*1	Regular*8, disabled*2, old-aged*2	Regular*11, disabled*4, old-aged*2	Regular*11, disabled*6, old-aged*3
CPU time (ms)	10396.6	197351.64	468900.35	1618725.21
Optimum of all regular labors hired	6352.5	50715	114686.25	177150
Optimum of heterogeneous labors hired	5262.88	46931.62	102196.77	156719.61
Improvement ratios (%)	17.15%	7.46%	10.89%	11.53%

TABLE 5: The t -test of various conditions optimal solution.

Scale (1)	Scale (2)	Scale (3)	Scale (4)
t Stat	-44.404***	-83.939***	-194.167***
$P(T \leq t)$ one-tail	0.000	0.000	0.000

Note: * denotes $p < 0.05$; ** denotes $p < 0.01$; *** denotes $p < 0.001$.

tolerable makespans compared with all regular workers hired. Test results show that improvement ratios are definite and outstanding in each test scale. As shown in Table 5, the optimum solutions are proved significantly difference in this study for solving the attempted problems.

5. Conclusion

This study examines the flexible flow shop scheduling problem, which is well-known for its strong NP-hard. The problem we study can be depicted as $FF_c | Z_{ijW_k M_k}^{rw}, Z_{ijW_k M_k}^{dw}, Z_{ijW_k M_k}^{ow}, e_r, e_d, e_o | C_{\max}(e_r, e_d, e_o) \cdot (C_o \cdot m - S_d \cdot e_d - S_o \cdot e_o)$. The problem involves fuzzy numbers, stepwise cost function, float time, and heterogeneous labors in a flexible flow shop, in which the objective is to minimize the product of makespan considering costs of hired regular, disabled, and old-aged labors and deduction of subsidies for the disabled and old-aged labors from unit cost emerged by incremental makespan.

The extra processing time from overtime generated by physical deterioration of old-aged labors is equal to the processing time of regular labors multiplied by the physical deterioration rate of old-aged labors on machine. The coefficient of the cost function is the stepwise function of cost structure. Besides, the dispatching rule based on floating

time utilizes ant colony optimization to minimize the makespan. Numerous studies indicate that ACO has excellent capacity to solve flexible flow shop scheduling problems, and this study demonstrates its effectiveness in solving the attempted problems.

This study significantly surmounts and grants ACO to solve the flexible flow shop scheduling problem and then compares various scales of solutions for saving costs. According to the test results and analysis, this study infers the following.

- (1) The advantage of this study is to downsize the complexity of the combination, assigning heterogeneous labors for keeping efficiency and costs of production. The effectiveness improvement ratio of hiring heterogeneous labors compared with regular ones in $w = 3$ and $n = 5$ is 17.15%, and 7.46% in $w = 4$ and $n = 10$, 10.89% in $w = 5$ and $n = 15$, and 11.53% in $w = 6$ and $n = 20$. This examines that the proposed method significantly surpasses the dispatching rule in terms of effectiveness and costs.
- (2) By screening the degree of float time of machines, assigning appropriate labors to operate can really reduce the computation time and the impact of hiring heterogeneous labors on the total completion time. The results indicate that the effectiveness and efficiency of the proposed method clearly transcends than that of manual production scheduling.
- (3) Under the various scales set in this study, ACO has excellent solving efficiency, which matches the capability of intelligent production planning. The data test results indicate that the proposed algorithm can efficiently dispatch and schedule the operations, with

the average improvement ratio about 11.76%, and demonstrates high capability for the intelligent production planning.

- (4) The suggestions for the future studies are using different dispatching rules or total flow time that might better match the real-world situations.

Above all, this study indicates that ACO outperforms advantages over both effectiveness and cost-saving. Thus, ACO has the potential to solve complicated problems in the practical scenario and becomes a significant heuristic algorithm for this kind of problem, responding to both production situations and future research.

Data Availability

No data were used to support this study.

Conflicts of Interest

All the authors declare that there are no conflicts of interest regarding the publication of this paper.

References

- [1] J. Belabid, S. Aqil, and K. Allali, "Solving permutation flow shop scheduling problem with sequence-independent setup time," *Journal of Applied Mathematics*, vol. 2020, Article ID 7132469, 11 pages, 2020.
- [2] O. B. Driss and H. Bargaoui, "Multi-agent model based on tabu search for the permutation flow shop scheduling problem," in *Proceedings of the International Symposium on Distributed Computing and Artificial Intelligence*, S. Omatu, H. Bersini, J. Corchado, S. Rodriguez, P. Pawlewski, and E. Bucciarelli, Eds., vol. 290, pp. 519–527, Springer, Salamanca, Spain, June 2014.
- [3] L. Gui, L. Gao, and X. Y. Li, "Anomalies in special permutation flow shop scheduling problems," *Chinese Journal of Mechanical Engineering*, vol. 33, no. 46, pp. 1–7, 2020.
- [4] J. Y. Shen and Y. G. Zhu, "Uncertain flexible flow shop scheduling problem subject to breakdowns," *Journal of Intelligent and Fuzzy Systems*, vol. 32, no. 1, pp. 1–8, 2016.
- [5] X. L. Wu, X. L. Shen, and Q. Cui, "Multi-objective flexible flow shop scheduling problem considering variable processing time due to renewable energy," *Sustainability*, vol. 10, pp. 841–871, 2018.
- [6] E. Nowicki and S. Zdrzałka, "A survey of results for sequencing problems with controllable processing times," *Discrete Applied Mathematics*, vol. 26, no. 2-3, pp. 271–287, 1990.
- [7] United Nations, "Convention on the rights of persons with disabilities (CRPD)," 2019, <https://www.un.org/development/desa/disabilities/convention-on-the-rights-of-persons-with-disabilities.html>. 2019.
- [8] A. Mishra and S. Deb, "Assembly sequence optimization using a flower pollination algorithm-based approach," *Journal of Intelligent Manufacturing*, vol. 30, no. 2, pp. 461–482, 2019.
- [9] H. Hamers, F. Klijn, and M. Slikker, "Implementation of optimal schedules in outsourcing with identical suppliers," *Mathematical Methods of Operations Research*, vol. 89, pp. 173–187, 2019.
- [10] Y. Fu, H. Wang, G. Tian, Z. Li, and H. Hu, "Two-agent stochastic flow shop deteriorating scheduling via a hybrid multi-objective evolutionary algorithm," *Journal of Intelligent Manufacturing*, vol. 30, no. 5, pp. 2257–2272, 2019.
- [11] J. Seif, M. Dehghanimohammadabadi, and A. J. Yu, "Integrated preventive maintenance and flow shop scheduling under uncertainty," *Flexible Services and Manufacturing Journal*, vol. 32, pp. 852–887, 2020.
- [12] K. R. Baker, "Introduction to Sequencing and Scheduling," Wiley, New York, NY, USA, 1974.
- [13] F. Huq, K. Cutright, and C. Martin, "Employee scheduling and makespan minimization in a flow shop with multi-processor work stations: a case study," *Omega-International Journal of Management Science*, vol. 32, no. 2, pp. 121–129, 2004.
- [14] A. Pooya and M. Pakdaman, "Optimal control model for finite capacity continuous MRP with deteriorating items," *Journal of Intelligent Manufacturing*, vol. 30, no. 5, pp. 2203–2215, 2019.
- [15] M. S. Nagano, A. S. Komesu, and H. H. Miyata, "An evolutionary clustering search for the total tardiness blocking flow shop problem," *Journal of Intelligent Manufacturing*, vol. 30, no. 4, pp. 1843–1857, 2019.
- [16] C. Lu, L. Gao, J. Yi, and X. Li, "Energy-efficient scheduling of distributed flow shop with heterogeneous factories: a real-world case from automobile industry in China," *IEEE Transactions on Industrial Informatics*, p. 1, 2020.
- [17] C. Lu, L. Gao, W. Gong, C. Hu, X. Yan, and X. Li, "Sustainable scheduling of distributed permutation flow-shop with non-identical factory using a knowledge-based multi-objective memetic optimization algorithm," *Swarm and Evolutionary Computation*, vol. 60, pp. 1–24, 2021.
- [18] M. Skutella, M. Sviridenko, and M. Uetz, "Unrelated machine scheduling with stochastic processing times," *Mathematics of Operations Research*, vol. 41, no. 3, pp. 1–14, 2016.
- [19] M. Sakawa and R. Kubota, "Two-objective fuzzy job shop scheduling through genetic algorithm," *Electronics and Communications in Japan Part III-Fundamental Electronic Science*, vol. 84, no. 4, pp. 60–68, 2001.
- [20] C. Blum and C. Miralles, "On solving the assembly line worker assignment and balancing problem via beam search," *Computers & Operations Research*, vol. 38, no. 1, pp. 328–339, 2011.
- [21] R. L. Daniels and P. Kouvelis, "Robust scheduling to hedge against processing time uncertainty in single-stage production," *Management Science*, vol. 41, no. 2, pp. 363–376, 1995.
- [22] C. S. Mccahon and E. S. Lee, "Job sequencing with fuzzy processing times," *Computers & Mathematics with Applications*, vol. 19, no. 7, pp. 31–41, 1990.
- [23] B. J. Joo, S. O. Shim, T. J. Chua, and T. X. Cai, "Multi-level job scheduling under processing time uncertainty," *Computers and Industrial Engineering*, vol. 120, pp. 480–487, 2018.
- [24] N. Xie and N. Chen, "Flexible job shop scheduling problem with interval grey processing time," *Applied Soft Computing*, vol. 70, pp. 513–524, 2018.
- [25] R. H. Huang and T. H. Yu, "An effective ant colony optimization algorithm for multi-objective job-shop scheduling with equal-size lot-splitting," *Applied Soft Computing*, vol. 57, pp. 642–656, 2017.
- [26] R. S. Hansmann, T. Rieger, and U. T. Zimmermann, "Flexible job shop scheduling with blockages," *Mathematical Methods of Operations Research*, vol. 79, pp. 135–161, 2014.
- [27] E. Pourjavad and R. V. Mayorga, "A comparative study and measuring performance of manufacturing systems with mamdani fuzzy inference system," *Journal of Intelligent Manufacturing*, vol. 30, no. 3, pp. 1085–1097, 2019.

- [28] D. Wang, J. Zhu, X. Wei, T. C. E. Cheng, Y. Yin, and Y. Wang, "Integrated production and multiple trips vehicle routing with time windows and uncertain travel times," *Computers & Operations Research*, vol. 103, pp. 1–12, 2019.
- [29] Y. Zhang, Z. J. M. Shen, and S. Song, "Exact algorithms for distributionally β -robust machine scheduling with uncertain processing times," *INFORMS Journal on Computing*, vol. 30, no. 4, pp. 662–676, 2018.
- [30] S. Balin, "Parallel machine scheduling with fuzzy processing times using a robust genetic algorithm and simulation," *Information Sciences*, vol. 181, no. 17, pp. 3551–3569, 2011.
- [31] S. Chanas and A. Kasperski, "On two single machine scheduling problems with fuzzy processing times and fuzzy due dates," *European Journal of Operational Research*, vol. 147, no. 2, pp. 281–296, 2003.
- [32] F. Rodrigues, A. Agra, M. Christiansen, L. M. Hvattum, and C. Requejo, "Comparing techniques for modelling uncertainty in a maritime inventory routing problem," *European Journal of Operational Research*, vol. 277, no. 3, pp. 831–845, 2019.
- [33] T. A. O. Silva and M. C. de Souza, "Surgical scheduling under uncertainty by approximate dynamic programming," *Omega-International Journal of Management Science*, vol. 95, pp. 1–37, 2020.
- [34] Y. M. Huang and J. C. Lin, "A new bee colony optimization algorithm with idle-time-based filtering scheme for open shop-scheduling problems," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5438–5447, 2011.
- [35] O. Engin, G. Ceran, and M. K. Yilmaz, "An efficient genetic algorithm for hybrid flow shop scheduling with multiprocessor task problems," *Applied Soft Computing*, vol. 11, no. 3, pp. 3056–3065, 2011.
- [36] L. Zhen, Z. Xu, C. L. Ma, and L. Y. Xiao, "Hybrid electric vehicle routing problem with mode selection," *International Journal of Production Research*, vol. 58, no. 2, pp. 562–576, 2019.
- [37] C. A. Dorigo, M. Dorigo, V. Manjezzo, and M. Trubian, "Ant system for job-shop scheduling," *Belgian Journal of Operations Research*, vol. 34, pp. 39–53, 1994.
- [38] I. Kucukkoc, Z. Li, A. D. Karaoglan, and D. Z. Zhang, "Balancing of mixed-model two-sided assembly lines with underground workstations: a mathematical model and ant colony optimization algorithm," *International Journal of Production Economics*, vol. 205, pp. 228–243, 2018.
- [39] L. C. Lu and T. W. Yue, "Mission-oriented ant-team ACO for min-max MTSP," *Applied Soft Computing*, vol. 76, pp. 436–444, 2019.
- [40] N. Singh and S. B. Singh, "A novel hybrid GWO-SCA approach for optimization problems," *Engineering Science and Technology, An International Journal*, vol. 20, pp. 1586–1601, 2017.
- [41] K. P. Chou, C. T. Lin, and W. C. Lin, "A self-adaptive artificial bee colony algorithm with local search for TSK-type neuro-fuzzy system training," in *Proceedings of the 2019 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1502–1509, IEEE, Wellington, New Zealand, June 2019.
- [42] A. Gupta and S. Srivastava, "Comparative analysis of ant colony and particle swarm optimization algorithms for distance optimization," *Procedia Computer Science*, vol. 173, pp. 245–253, 2020.
- [43] M. Dorigo, *Optimization, learning and natural algorithms*, PhD Thesis, Dipartimento di Elettronica, Politecnico di Milano, Milan, Italy, 1992.
- [44] M. Dorigo, V. Maniezzo, and A. Colnori, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 26, no. 1, pp. 29–41, 1996.
- [45] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, pp. 53–66, 1997.
- [46] M. Dorigo and G. Di Caro, "The ant colony optimization metaheuristic," *New Ideas Optimization*, vol. 18, pp. 11–57, 1999.
- [47] M. Dorigo and T. Stützle, *Ant Colony Optimization*, MIT Press, Cambridge, UK, 2004.
- [48] I. Mladenović, M. Milovančević, and S. Sokolov-Mladenović, "Retraction note: Analyzing of innovations influence on economic growth by fuzzy system," *Quality Quantity*, vol. 53, pp. 1297–1304, 2019.