Postprint: Flexible Job Shop Scheduling with AGVs Considering Behavioral Effects

Authors: Xu Yunqin, Ye Chunming, Cao Lei

Date: 2018-07-09T00:00:00+00:00

Abstract

To investigate the impact of behavioral effects on the flexible job shop scheduling problem with AGVs, a mathematical model considering both learning and deterioration effects with the objective of minimizing makespan is constructed. Based on the characteristics of this model, an encoding scheme based on operations, machines, and AGV assignment is adopted, and a whale algorithm with chaotic local search strategy is proposed for solving it. Numerical example results verify the effectiveness and feasibility of the algorithm. Experiments compare the differences in processing time under four scenarios: learning effect only, deterioration effect only, combined learning-deterioration effect, and no effect considered. Simultaneously, the influence of processing task batch size and AGV quantity on processing time is studied. Through experiments, it is demonstrated that AGV resource constraints and the various effects jointly influence the unit batch completion time.

Full Text

Preamble

Research on Flexible Job-Shop Scheduling Problem with AGV Constraints and Behavioral Effects

Xu Yunqin, Ye Chunming, Cao Lei
(School of Business, University of Shanghai for Science & Technology, Shanghai 200093, China)

Abstract: To investigate the impact of behavioral effects on the flexible job-shop scheduling problem with AGV constraints, this paper constructs a mathematical model aimed at minimizing completion time, considering both learning and deterioration effects. Based on the model' s characteristics, we adopt an encoding scheme based on process sequences, machine allocation, and AGV assignment, and propose a whale optimization algorithm with chaotic local search

strategy. Numerical examples validate the algorithm' s effectiveness and feasibility. Experiments compare processing time differences across four scenarios: learning effect only, deterioration effect only, combined learning-deterioration effect, and no effect. The study also examines how processing task batch size and AGV quantity affect processing time. Results demonstrate that AGV resource constraints and various behavioral effects jointly influence unit batch completion time.

Key words: AGV; FJSP; learning effects; deteriorating effects; whale optimization algorithm

0 Introduction

AGV (Automated Guided Vehicle) is a material handling mobile robot controlled by intelligent computer systems, widely applied in manufacturing. In actual production, workpiece processing times are affected by behavioral effects. Generalized behavioral effects improve production efficiency by studying human behavior patterns and machine processing laws, including learning effects, deterioration effects, etc. This research examines how worker learning effects and workpiece deterioration effects influence processing time to help enterprises arrange production plans more rationally.

When machines repeatedly process the same part, processing time per part decreases as production volume increases, producing a learning effect. When workpiece start times are delayed, processing time per part increases, creating a deterioration effect. Literature has proposed scheduling models with deterioration and learning effects and resource allocation, generalizing the model as $P_{ij}^{r} = P_{ij} \times \max\{f(r), b_j \times c\}$ (where r represents available resources for job j, c is the deterioration rate, and $f(r)$ is the learning effect function). Other studies have examined no-wait flow shop scheduling with learning-deterioration effects using local search methods to minimize total tardiness, while some have addressed flexible job-shop scheduling with simultaneous learning-deterioration effects using improved genetic algorithms combined with variable neighborhood search. Parameter settings significantly affect algorithm performance, prompting the use of Taguchi' s robust design method to define optimal parameter values. Some research has considered flexible job-shop scheduling with learning effects and setup times, proposing genetic algorithms combined with variable neighborhood search and iterative local search with adaptive parameter adjustment strategies. Others have studied parallel machine scheduling with learning-deterioration effects using ant colony optimization.

Our literature review reveals minimal research on flexible job-shop scheduling with AGV constraints and behavioral effects. Current scholars typically study scheduling with learning-forgetting effects or learning-deterioration effects, but researchers often use fixed learning rates, forgetting rates, and deterioration rates for comparative analysis against scenarios without effects, neglecting to analyze the differential impacts of varying these factors. This paper not only examines how different effect factors influence processing time but also analyzes

the impacts of task batch size and AGV quantity on flexible job-shop scheduling with AGV constraints.

1.1 Problem Description

The flexible job-shop scheduling problem with AGV constraints involves n jobs processed on m machines with AGVs responsible for inter-machine transportation, where each machine is operated by one worker. Each part has multiple operations that can be processed on multiple machines. There are r AGVs available, with $r < m + 2$. The following assumptions apply:

a) AGV transportation routes are fixed, AGV transportation is non-delayed, and different AGVs do not interfere with each other.

b) AGV tasks involve immediately transporting workpieces from the output buffer of the previous operation' s machine to the input buffer of the next operation' s machine.

c) At time zero, all equipment and machines are available.

d) The time from a workpiece entering a machine' s input buffer to starting processing is instantaneous and negligible.

e) Each machine can process only one part at a time, and each part cannot be interrupted during processing.

f) Machines follow a First-Come-First-Served (FCFS) processing sequence constraint.

g) Different jobs have no precedence constraints; the same job must complete the previous operation before proceeding to the next.

h) All workpieces to be processed are placed at loading station M_0, and completed workpieces are placed at unloading station M_{m+1}.

1.2.1 Learning Effect Model

The learning effect refers to workers becoming more proficient and reducing processing time as they produce more units. Since learning effects vary by individual, processing times for the same workpiece differ among workers, making it necessary to study learning effects' impact on processing time.

Current research lacks studies on parameter values for learning effect models, which can cause significant experimental deviations. Therefore, this paper adopts the concise DeJong learning effect model:

$$P_{ij}^{r} = P_{ij} \times [M + (1 - M) \times r^{-\alpha}]$$

where P_{ij}^{r} represents the actual processing time, P_{ij} is the theoretical processing time (without learning effect), r indicates the job' s processing position (the r-th time processing this job), α is the learning factor, l is the learning rate, and M is the incompressible factor: when $M = 0$, it becomes the Biskup model; when $M = 1$, it represents the case without learning effect.

According to references [11, 12], employee learning effects are influenced by four factors: initial employee capability, capability ceiling, learning ability, and repetition frequency. Initially, employees have varying initial capabilities related to work experience and pre-job training. Initial capability for a position is expressed as the ratio of employee processing time to standard processing time —the stronger the initial capability, the smaller the ratio; the more complex the work, the larger the ratio.

We construct an initial capability matrix $P_{initial}$ (where $p_{initial_{ij}}$ represents employee i' s initial capability for processing job j, with m employees independently operating m machines and n jobs):

$$P_{initial} = \begin{bmatrix} p_{initial_{11}} & p_{initial_{12}} & \cdots \& p_{initial_{1n}} \\ p_{initial_{21}} & p_{initial_{22}} & \cdots \& p_{initial_{2n}} \\ \vdots & \vdots & \ddots \\ p_{initial_{m1}} & p_{initial_{m2}} & \cdots \& p_{initial_{mn}} \end{bmatrix} \vdots$$

Learning ability relates to observation, memory, comprehension, and attention —stronger learning ability means smaller learning rate l; simpler work means larger l. We construct the employee learning rate matrix L (where l_{ij} represents employee i' s learning rate for job j):

$$L = \begin{bmatrix} l_{11} & l_{12} & \cdots \& l_{1n} \\ l_{21} & l_{22} & \cdots \& l_{2n} \\ \vdots & \vdots & \ddots \\ l_{m1} & l_{m2} & \cdots \& l_{mn} \end{bmatrix} \vdots$$

As repetition frequency increases, processing time stabilizes when employee capability reaches its ceiling. The capability ceiling relates to job difficulty and employee ability—more difficult jobs have lower ceilings; weaker learning ability results in lower ceilings. $P_{final_{ij}}$ represents employee i' s maximum capability for processing job j.

Based on the four influencing factors of learning effects, we modify the DeJong learning model as:

$$P_{ij}^{r} = \begin{cases} P_{ij} \times [M + (1 - M) \times r^{-\alpha}] & \text{if } P_{ij} \times [M + (1 - M) \times r^{-\alpha}] \geq P_{final_{ij}} \\ P_{final_{ij}} & \text{if } P_{ij} \times [M + (1 - M) \times r^{-\alpha}] < P_{final_{ij}} \end{cases}$$

where $\alpha = -\frac{\lg l}{\lg 2}$ and $r \in N^*$.

To verify that initial capability, capability ceiling, learning ability, and repetition frequency are important factors affecting skill changes, assume two employees processing the same batch of the same operation. The skill curve changes are shown in [Figure 2: see original paper]. Employee 2 has stronger initial skills than Employee 1 but weaker learning ability. After 5 repetitions, Employee 2' s skills become weaker than Employee 1' s, and both skills remain constant at $r = 6$.

1.2.2 Deterioration Effect Model

In actual production, when a workpiece' s start time is sufficiently late or its position is far back in the sequence, processing time cannot increase indefinitely. However, most existing deterioration effect research neglects this aspect. Therefore, this paper improves upon the general linear deterioration model:

$$P_{ij}^* = P_{ij} + b_j \times t$$

where P_{ij} is the basic processing time, t is the workpiece' s start time, and b_j is the deterioration factor related only to the job.

The improved model is:

$$P_{ij}^* = P_{ij} + b_j \times \min\{\max\{t - t_{min}, 0\}, t_{max} - t_{min}\}$$

where t_{min} and t_{max} represent the start and end times of deterioration, respectively.

1.2.3 Learning-Deterioration Effect Model

While learning effects shorten processing time through skill proficiency, deterioration effects lengthen processing time due to delayed start times. Since processing time is influenced by both factors simultaneously, it is necessary to comprehensively consider their combined impact.

This paper adopts Cheng' s (2017) learning-deterioration model [14]: $P_{ij}^r = P_{ij} + b_j \times t \times r^{\alpha}$, combined with our improved learning effect model to obtain:

$$P_{ij}^r = \begin{cases} P_{ij} \times [M + (1-M) \times r^{-\alpha}] + b_j \times \min\{\max\{t - t_{min}, 0\}, t_{max} - t_{min}\} & \text{if } P_{ij} \times [M + (1-M) \times r^{-\alpha}] \\ P_{final_{ij}} + b_j \times \min\{\max\{t - t_{min}, 0\}, t_{max} - t_{min}\} & \text{if } P_{ij} \times [M + (1-M) \times r^{-\alpha}] \end{cases}$$

2.1 Algorithm Encoding and Decoding

This paper uses a humpback whale position to represent a candidate scheduling solution with dimension $3L$, consisting of a transport sequence vector Tr, a machine vector M, and a robot vector R, where each dimension of these three

vectors corresponds to each other. The particle dimension is $\sum_{i=1}^{n} L_i$ (where n is the number of jobs and L_i is the total number of operations for job i). The Tr vector components are less than or equal to the total number of jobs, the M vector components are less than or equal to the total number of processing machines (including loading and unloading stations), and the R vector components are less than or equal to the number of AGVs. shows an encoding example.

Since the Whale Optimization Algorithm' s position update formulas inevitably produce non-integer values, and in FJSP operation sequences must be integers and machines and AGVs must be assigned integer values, we adjust iteration results according to the following rules:

a) **Tr Vector Adjustment**: Algorithm formulas inevitably produce decimal components. Given FJSP' s operation precedence constraints and integer operation requirements, we correct iteration results using the method shown in [Figure 4: see original paper]. First, sort the iterated Tr vector values from smallest to largest to obtain a new vector, then rearrange the original values according to the new vector' s order to obtain a corrected feasible operation sequence.

b) **M Vector Adjustment**: Algorithm formulas inevitably produce decimal components. For FJSP' s integer machine requirement (assuming full flexibility), we correct iteration results by first rounding up the iterated M vector components, then applying boundary constraints: values exceeding the upper bound (total number of machines) are set to the upper bound, and values below the lower bound are set to 1. For partial flexibility, first obtain the corrected Tr vector to determine the operation sequence, then update each component of the iterated M vector to the machine number in the corresponding operation' s optional machine set that is closest to the original value.

c) **R Vector Adjustment**: The adjustment rules follow the same procedure as the M vector (assuming full flexibility).

2.2 Population Initialization

Based on the encoding scheme above, the initialization process involves: first, randomly generating the transport sequence; second, using a greedy method to assign machines; and third, heuristically generating the R vector, drawing on heuristic adjustment ideas from reference [15] to reduce idle time.

Greedy Machine Assignment: a) The randomly generated Tr vector determines the operation processing sequence. b) Calculate the earliest completion time for each operation and assign it to the machine with the earliest completion. c) If multiple machines have equal completion times, assign the operation to the machine with shorter waiting time. If waiting times are equal, randomly assign the operation to one of the machines.

Heuristic Generation and Adjustment of R Vector: a) **Generation**: Heuristically and uniformly assign AGVs to each transport operation in the

sequence to obtain the R vector. b) **Adjustment**: After an AGV transports workpiece i to a machine, check if workpiece j is waiting in that machine's buffer. If so, the AGV immediately transports workpiece j next, moving workpiece j' s transport operation after the current one to reduce the AGV' s idle time. If not, the AGV proceeds to the next transport operation in the sequence. Based on the R vector, adjust the transport sequence from left to right, ultimately minimizing the AGV' s idle time.

2.3 Position Update Strategy

The Whale Optimization Algorithm (WOA) [16], proposed by Australian scholar Mirjalili et al. in 2016, is a novel swarm intelligence optimization algorithm inspired by humpback whale hunting behavior. Through collective searching, encircling, and attacking prey, the algorithm achieves optimization. Although WOA has demonstrated superior solution accuracy and convergence speed compared to PSO, DE, Gravitational Search Algorithm (GSA), and GWO, it still suffers from difficulties balancing global and local search and tends to fall into local optima [17]. Improvements addressing these issues remain underexplored.

WOA' s position update results from three interacting behaviors: "searching for prey," "encircling prey," and "spiral predation."

a) **Searching for Prey**: Whales navigate by randomly selecting individual positions from the population.

$$\begin{cases} X_{t+1} = X_{rand} - A \cdot D \\ D = |C \cdot X_{rand} - X_t| \end{cases}$$

where X_t and X_{t+1} represent a whale' s position at iterations t and $t+1$, and A, C are coefficient vectors. When $A \leq 1$, the random individual X_{rand} is considered prey, and whales approach it; otherwise, they move away to search for better prey.

Coefficient vectors A and C are calculated as:

$$\begin{cases} A = 2a \cdot r - a \\ C = 2r \end{cases}$$

where a decreases monotonically from 2 to 0 with iteration count: $a = 2 - 2t/T_{max}$; r is a random vector in [0,1].

b) **Encircling Prey**: Whales use the population' s best individual position to navigate and shrink the encirclement.

$$\begin{cases} X_{t+1} = X_{best} - A \cdot D \\ D = |C \cdot X_{best} - X_t| \end{cases}$$

where X_{best} is the global best individual (prey location or position closest to prey). When $A > 1$, whales move away from random individuals toward X_{best}.

c) **Spiral Predation**: While approaching X_{best}, whales may also move along a logarithmic spiral path with probability p.

$$\begin{cases} X_{t+1} = D' \cdot e^{bl} \cdot \cos(2\pi l) + X_{best} \\ D' = |X_{best} - X_t| \end{cases}$$

where D' is the distance between the whale' s current position and the best individual, b is a constant for constructing the spiral path, and l is a random number in $[-1, 1]$.

2.4 Chaotic Search Strategy

In WOA, whales navigate toward prey using the best individual' s coordinates while spiraling and shrinking the encirclement. Although this accelerates convergence, it can cause the population to cluster rapidly, leading to premature convergence [17]. Chaotic search, with its ergodic and stochastic properties, can escape local optima to find global optima to some extent. However, chaotic search cannot learn from experience and suffers from search blindness [18, 19]. To combine WOA' s fast optimization with chaotic search' s ergodicity, we introduce chaos into WOA.

The chaotic search process works as follows: first, map the whale' s spatial position to the chaotic interval (-1, 1); then perform ergodic search using a chaotic self-mapping function; next, convert the obtained chaotic variable sequences into solution vector sequences to find the optimal solution vector; finally, update the whale' s spatial position.

a) Convert whale individual spatial positions to the chaotic interval (-1, 1):

$$Z_i = \frac{2(x_i - a_i)}{b_i - a_i} - 1$$

where (x_i, a_i, b_i) represents the i-th dimension component.

b) Use the logistic self-mapping function [20, 21], which has better ergodicity than the classical logistic map, to generate multiple chaotic variable sequences:

$$Z_{i+1} = 1 - 2Z_i^2$$

where $i = 0, 1, 2, ...$; when $Z_i \in (-1, 1)$ and $Z_0 \neq 0, \pm 0.5$, chaos occurs.

c) Convert the obtained chaotic variable sequences into multiple solution vector sequences to obtain the whale individual' s multiple new spatial positions:

$$x_i' = \frac{1}{2}(b_i - a_i) \times Z_i + \frac{b_i + a_i}{2}$$

d) Calculate fitness values for multiple spatial positions and replace the whale' s position with the one corresponding to the best fitness value.

WOA' s main parameters are the spiral path constant b, probability p^*, population size N, and maximum iterations T_{max}. To study how different values of b, p^*, N, and T_{max} affect WOA' s optimization performance, we conducted orthogonal experiments on an 8$×$8 test instance [21].

a) Based on parameter values from WOA literature, we established parameter levels to create a four-factor, three-level orthogonal experiment table ($L_9(3^4)$).

b) Each parameter level combination was run independently 20 times, using the average of the 20 results as the performance indicator.

c) We calculated the range for each factor. Based on these ranges, we determined each parameter' s importance to find the optimal factor level combination. The results show that N has the largest range and highest rank, indicating population size most significantly affects algorithm performance. The spiral constant b and probability p^* have less obvious impact. The data clearly show that larger population sizes yield exponentially better feasible solutions for the same iteration count, thus increasing the probability of obtaining optimal solutions.

Based on experimental data and analysis, the recommended parameter combination is: $b = 0.5$, $p^* = 0.5$, $N = 500$, $T_{max} = 200$.

2.5 Algorithm Solution Steps

a) Initialize parameters: population size N, current iteration t, maximum iterations T_{max}, spiral constant b, random vectors r, and random numbers l, p.

b) Initialize population using the proposed initialization rules, calculate each individual' s fitness value, and obtain X_{best}.

c) If random number $p^* < 0.5$, individuals update positions via "searching for prey" ; otherwise, they use "encircling prey." If random number $p^* \geq 0.5$, individuals use "spiral predation." Calculate fitness values and update X_{best}.

d) Apply chaotic search strategy to the elite group (top 2-10 individuals by fitness) in the newly updated population, update the elite group, merge with individuals not undergoing chaotic search to obtain a new population, and update X_{best}.

e) When the algorithm reaches maximum iterations, it terminates and outputs the optimal scheduling sequence from X_{best}.

3.2 Algorithm Optimization Capability Verification

The experimental simulation environment is Windows 7 64-bit OS, 2.5GHz processor, 4GB RAM, using Matlab R2014a. Each instance runs 20 times, taking the best value. Population size N is 500, maximum iterations 200, $b = 0.5$, $p^* = 0.5$.

To verify the algorithm' s optimization capability, we selected instances from Bilge et al.' s [22] 40 test cases: 4 processing machines, loading station, unloading station, and AGVs form four layouts, considering AGV transport and idle times to find minimum processing time. Instance "Ex21" indicates Jobset2 with Layout1. MAS represents results from reference [23]' s algorithm. GAHA-2 and GAHA-3 represent results from reference [24]' s hybrid genetic algorithm with 2 and 3 AGVs considering workpiece transport to unloading station. FDE and FMAS represent results from references [25, 26]. PSO and FPSO represent results using our improved particle swarm algorithm for job-shop and flexible job-shop scheduling, respectively.

Experimental results demonstrate our improved algorithm's strong optimization capability: for job-shop scheduling, 41.67% of solutions are better than MAS (R=2) and GAHA (R=2), and 91.67% are better than GAHA (R=3). For flexible job-shop scheduling, 50% of solutions are better than FDE (R=2) and FMAS (R=2). Additionally, increasing AGVs reduces total completion time, with greater effectiveness for longer completion times.

3.3.1 AGV Marginal Utility

According to 2017-2020 China Mobile Robot (AGV) industry research reports, mid-to-high-end AGV unit prices range from 200,000 to 500,000 RMB, even up to 1,000,000 RMB. Due to high AGV costs, investment decisions on AGV quantity are crucial. Therefore, we introduce marginal utility theory from economics to measure the effect of each AGV investment on reducing workpiece processing time.

Marginal utility theory states that commodity value is determined by its marginal utility—the incremental total utility from consuming one additional unit of a good or service [27]. In this context, utility represents the total time reduction for processing all jobs, and the commodity refers to AGVs:

$$\Delta U = \frac{\Delta T}{Q}$$

where ΔU is marginal utility, ΔT is total time reduction, and Q is AGV quantity.

[Figure 8: see original paper] shows AGV marginal utility follows the law of diminishing marginal utility: as AGV quantity increases, total utility increases (total processing time decreases), but marginal utility decreases (time reduction per additional AGV diminishes). When AGV=15, marginal utility is 0, indicating equal total utility for 14 and 15 AGVs—maximum total utility is achieved at

AGV=14. At this point, processing time is primarily constrained by machines and operations, and cannot be further reduced by simply adding AGVs.

3.3.2 Learning and Deterioration Effects

1) Without Considering Employee and Workpiece Heterogeneity

When employee initial skills and final skills are identical, and workpiece deterioration start and end times are the same (ignoring heterogeneity), we examine how different learning rates and deterioration factors affect completion time. Results show that as the learning curve improves from 90% to 60%, completion time for the flexible job-shop scheduling problem with AGVs (AGV=2) decreases significantly—smaller learning rates yield shorter completion times, demonstrating that learning effects improve factory productivity. Additionally, as deterioration factors increase, completion time gradually increases, indicating larger deterioration factors cause greater learning effect degradation.

2) Considering Employee and Workpiece Heterogeneity

To deeply study AGV marginal utility, we selected instance 15$×$10 [28] with more machines and jobs, testing different AGV quantities with 20 runs each. Results show that when AGV quantity $\geq$ 14, optimal completion time is 170. When AGV=13, marginal utility is 4; when AGV=14, marginal utility is 6. The slight marginal utility increase is attributed to insufficient computational runs.

Using EX31 as a simulation example, we conducted small-batch production for all jobs with batch sizes of 1, 2, 3, 4, and 5. Matrices represent each employee' s initial capability, maximum capability, and learning rate for each job type. For example, 0.77 (second row, fourth column) indicates employee 2' s initial capability of 0.77 for job 2. Workpiece deterioration factor is $B = [0.2, 0.2, 0.1, 0.2, 0.15]$, with deterioration starting at $t_{min} = [50, 50, 50, 50, 50]$ and ending at $t_{max} = [100, 100, 100, 100, 100]$.

[Figure 9: see original paper] through [Figure 11: see original paper] discuss behavioral effects on unit batch completion time with 2 AGVs.

[Figure 9: see original paper] Observations: - For the same batch size: learning effect only < learning-deterioration effect < no effect < deterioration effect only. This indicates that with batch=1 and AGV=2, work start times are not delayed, employee learning hasn' t reached maximum skill, and workpieces are just beginning to deteriorate, so learning' s time-reducing effect > deterioration' s time-increasing effect (same for batch=2, AGV=2). - Under the same effect, batch=2 unit completion time < batch=1 unit completion time. This shows AGVs often wait for operations to complete before transporting to the next machine—AGV resources are not the bottleneck.

[Figure 10: see original paper] Observations: - For the same batch size: learning effect only < no effect < learning-deterioration effect < deterioration effect only. This indicates that with batch=3 and AGV=2, work start times are longer, employee learning has reached maximum skill, and workpieces are

in the deterioration phase, so deterioration' s time-increasing effect > learning' s time-reducing effect (same for batches 3, 4, 5 with AGV=2).

[Figure 11: see original paper] Observations: - For the same batch size, more AGVs yield shorter unit batch completion times. - With 1 or 2 AGVs, unit batch completion time first decreases then increases as batch size increases, with minimum times at batch=3 (AGV=1) and batch=4 (AGV=2). - With 3 AGVs, unit batch completion time decreases as batch size increases. This shows that with 1 or 2 AGVs, as batch size increases, AGVs become bottleneck resources —machines must wait for AGV transportation. AGVs become bottlenecks at smaller batch sizes with 1 AGV than with 2 AGVs. With 3 AGVs, AGVs are not yet bottleneck resources.

Learning and deterioration effects objectively exist in actual manufacturing. The above experiments demonstrate these effects significantly impact scheduling and cannot be ignored.

To eliminate effect influences and further explore the intrinsic relationship between AGV resource constraints and batch size, we conducted supplementary experiments shown in .

4 Conclusion

This paper adopts a process-machine-AGV allocation-based encoding scheme [29], using a $3L$-dimensional vector to represent a feasible production schedule. We integrate chaotic local search strategy into the whale optimization algorithm to effectively avoid falling into local optima during iterations. Taguchi' s robust design method is used to obtain optimal parameter combinations. Computational experiments demonstrate that the whale algorithm with chaotic local search strategy exhibits high optimization performance. Key findings include: learning effects improve factory productivity while deterioration effects have the opposite effect, equivalent to learning effect degradation; AGV resource constraints and various effects jointly influence unit batch completion time; as processing time progresses, learning effects become increasingly significant until employee capability reaches its ceiling; when start times reach workpiece deterioration times, deterioration effects become increasingly significant until exceeding maximum deterioration time.

Many factors affecting workpiece processing time remain unexplored in this paper, such as forgetting effects, employee leave, and employee burnout. Future research should analyze processing time influences from more comprehensive perspectives.

References

[1] Lacomme P, Larabi M, Tchernev N. Job-shop based framework for simultaneous scheduling of machines and automated guided vehicles [J]. International Journal of Production Economics, 2013, 143(1): 24-34.

[2] Cui Miaomiao. Total completion time problem for machine-constrained

scheduling with learning and deterioration effects [J]. Electronic Test, 2017(2): 28-29.

[3] Wang Jia. Single-machine scheduling problem with learning and deterioration effects and resource allocation [D]. Shenyang: Shenyang Aerospace University, 2016.

[4] Li Xiaoping, Chen Tian, Xu Haiyan, et al. A method for no-wait flow shop scheduling problem with learning and deterioration effects: China, CN 201610785498 [P]. 2017-02-01.

[5] Tayebi Araghi M E, Jolai F, Rabiee M. Incorporating learning effect and deterioration for solving a SDST flexible job-shop scheduling problem with a hybrid meta-heuristic approach [J]. International Journal of Computer Integrated Manufacturing, 2014, 27(8): 733-746.

[6] Azzouz A, Ennigrou M, Said L B. A self-adaptive evolutionary algorithm for solving flexible job-shop problem with sequence dependent setup time and learning effects [C]// Proc of IEEE Congress on Evolutionary Computation. 2017: 1827-1834.

[7] Mir S, Saber M, Rezaeian J. Two meta-heuristic algorithms for parallel machines scheduling problem with past-sequence-dependent setup times and effects of deterioration and learning [J]. International Journal of Industrial Engineering & Production Research, 2016, 27(1): 69-88.

[8] Araghi M E T, Jolai F, Rabiee M. Incorporating learning effect and deterioration for solving a SDST flexible job-shop scheduling problem with a hybrid meta-heuristic approach [J]. International Journal of Computer Integrated Manufacturing, 2014, 27(8): 733-746.

[9] Zhao Chuanli, Fang Ji, Cheng T C E, et al. A note on the time complexity of machine scheduling with DeJong' s learning effect [J]. Computers & Industrial Engineering, 2017, 447-449.

[10] Biskup D. Single-machine scheduling with learning considerations [J]. European Journal of Operational Research, 1999, 115(1): 173-178.

[11] Yu Xiuli. Modeling and staff organization optimization of manual operation systems (MOS) [D]. Guangzhou: Guangdong University of Technology, 2013.

[12] Cao Lei, Ye Chunming, Huang Xia. Multi-objective flexible job-shop scheduling based on employee learning behavior [J/OL]. Computer Integrated Manufacturing Systems, 2018: 1-21 [2018-05-20]. http://kns.cnki.net/kcms/detail/11.5946.TP.20170818.1120.0

[13] Browne S, Yechiali U. Scheduling Deteriorating Jobs on a Single Processor [J]. Operations Research, 1990, 38(3): 495-498.

[14] Wang X, Cheng T C E. Single-machine scheduling with deteriorating jobs and learning effects to minimize the makespan [J]. European Journal of Operational Research, 2007, 178(1): 57-70.

[15] Long Chuanze, Yang Yujun. Research on flexible robot manufacturing cell scheduling problem based on genetic algorithm [J]. Combined Machining Equipment and Machining Technology, 2015, 148(11): 141-144.

[16] Mirjalili S, Lewis A. The whale optimization algorithm [J]. Advances in Engineering Software, 2016, 95: 51-67.

[17] Zhong Minghui, Long Wen. A whale optimization algorithm with random adjustment of control parameters [J]. Science Technology and Engineering, 2017(12): 68-73.

[18] Liu Zhusong, Li Sheng. Sine-cosine chaotic double-string whale optimization algorithm [J]. Computer Engineering and Applications, 2018(7).

[19] Jia Zhaohong, Chen Huaping, Sun Yaohui. Application of multi-objective particle swarm optimization algorithm in flexible job-shop scheduling [J]. Mini-Micro Systems, 2008, 29(5): 885-889.

[20] Liu Changping, Ye Chunming. Variable scale chaotic particle swarm optimization algorithm based on logical self-mapping [J]. Application Research of Computers, 2011, 28(8): 2825-282.

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

[22] Bilge U, Ulusoy G. A Time window approach to simultaneous scheduling of machines and material handling system in FMS [J]. Operations Research, 1995, 43(6): 1058-1070.

[23] Erol R, Sahin C, Baykasoglu A, et al. A multi-agent based approach to dynamic scheduling of machines and automated guided vehicles in manufacturing systems [J]. Applied Soft Computing, 2012, 12(6): 1720-1732.

[24] Long Chuanze. Research on scheduling algorithms for flexible multi-robot manufacturing cells [D]. Guangzhou: Guangdong University of Technology, 2015.

[25] Kumar M V S, Janardhana R, Rao C S P. Simultaneous scheduling of machines and vehicles in an FMS environment with alternative routing [J]. International Journal of Advanced Manufacturing Technology, 2011, 53(1-4): 339-351.

[26] Sahin C, Demirtas M, Erol R, et al. A multi-agent based approach to dynamic scheduling with flexible processing capabilities [J]. Journal of Intelligent Manufacturing, 2017, 28(8): 1827-1845.

[27] Li Yang, Wang Jue. Research on optimal human resource management model based on marginal utility function [J]. Systems Engineering - Theory & Practice, 2016, 36(1): 106-112.

[28] Mousavi M, Yap H J, Musa S N, et al. Multi-objective AGV scheduling in an FMS using a hybrid of genetic algorithm and particle swarm optimization [J]. PLoS One, 2017, 12(3): 1-24.

[29] Xu Yunqin, Ye Chunming, Cao Lei. Research on optimization of flexible job-shop scheduling with AGV [J]. Application Research of Computers, 2018, 35(11).

Note: Figure translations are in progress. See original paper for figures.