Knowledge-Based Systems 243 (2022) 108315

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

A multiobjective evolutionary algorithm for achieving energy efficiency in production environments integrated with multiple automated guided vehicles

Lijun He [a,b], Raymond Chiong [b], Wenfeng Li [a,*], Gregorius Satia Budhi [b,c], Yu Zhang [a]

[a] School of Transportation and Logistics Engineering, Wuhan University of Technology, Wuhan, Hubei 430063, PR China
[b] School of Information and Physical Sciences, The University of Newcastle, Callaghan, NSW 2308, Australia
[c] Informatics Department, Petra Christian University, Surabaya 60236, Indonesia

A R T I C L E   I N F O

Article history:
Received 5 September 2020
Received in revised form 24 January 2022
Accepted 25 January 2022
Available online 2 February 2022

Keywords:
Energy efficiency
Automated guided vehicles
Job-shop scheduling
Evolutionary algorithms
Multiobjective optimization
Opposition-based learning

A B S T R A C T

Increasing energy shortages and environmental pollution have made energy efficiency an urgent concern in manufacturing plants. Most studies looking into sustainable production in general and energy-efficient production scheduling in particular, however, have not paid much attention to logistical factors (e.g., transport and setup). This study integrates multiple automated guided vehicles (AGVs) into a job-shop environment. We propose a multiobjective scheduling model that considers machine processing, sequence-dependent setup and AGV transport, aiming to simultaneously minimize the makespan, total idle time of machines and total energy consumption of both machines and AGVs. To solve this problem, an effective multiobjective evolutionary algorithm (EMOEA) is developed. Within the EMOEA, an efficient encoding/decoding method is designed to represent and decode each solution. A new crossover operator is proposed for AGV assignment and AGV speed sequences. To balance the exploration and exploitation ability of the EMOEA, an opposition-based learning strategy is incorporated. A total of 75 benchmark instances and a real-world case are used for our experimental study. Taguchi analysis is applied to determine the best combination of key parameters for the EMOEA. Extensive computational experiments show that properly increasing the number of AGVs can shorten the waiting time of machines and achieve a balance between economic and environmental objectives for production systems. The experimental results confirm that the proposed EMOEA is significantly better at solving the problem than three other well-known algorithms. Our findings here have significant managerial implications for real-world manufacturing environments integrated with AGVs.

1. Introduction

Production scheduling is an important decision-making process in manufacturing. It aims to arrive at a scheduling solution that can effectively help manufacturers improve production efficiency and reduce production costs [1]. Global energy shortages and environmental pollution mean that energy efficiency issues are receiving increased attention in various research fields. According to a survey by the International Energy Agency [2], the total energy demand worldwide will increase by 37% by 2040. Energy consumed by the manufacturing sector accounts for a large proportion of this demand (e.g., energy consumption from the industrial sector is approximately 50% of total consumption [3]). Excessive energy consumption will lead to a sharp deterioration of the environment through means such as the rapid exhaustion of fuel resources and global warming [4]. Therefore, manufacturing enterprises should make efforts to reduce energy consumption from both economic and environmental perspectives. Researchers have realized that production scheduling can reduce energy consumption in manufacturing processes with less financial cost than expensive energy-saving hardware [5–7]. Consequently, there is a growing interest in energy-efficient production scheduling-related research [5–10]. To the best of our knowledge, however, existing studies mainly focus on the energy consumption of processing machines. Almost all of them have overlooked the influence of logistical factors (e.g., job transport and machine setup) on production efficiency and energy consumption.

Since the industrial revolution that initiated the proposal of Industry 4.0 strategic plans and other advanced information technologies [11,12], manufacturing operations have been changing

* Corresponding author.
E-mail address: liwf@whut.edu.cn (W. Li).

https://doi.org/10.1016/j.knosys.2022.108315
0950-7051/© 2022 Elsevier B.V. All rights reserved.

from mass standardized production to mass customized production. This means that a workshop in a manufacturing environment should have a flexible production capacity to meet customer requirements for multiple types of products. Different types of products have different processing paths, which means that many logistics conversion activities, such as transport, loading and unloading, are needed to maintain the consistency of operations. Nevertheless, these conversion activities generate a large conversion time, which greatly extends the production cycle. In some long-distance and discrete production workshops, the conversion time, especially the logistic transport time, accounts for a large proportion of the total production time [13]. Therefore, it is necessary to consider the logistic transport time of jobs when conducting production scheduling research. However, most of the relevant studies have overlooked the transport time or the effect of transport time on scheduling solutions. This means that the scheduling solutions cannot effectively match the actual production situation—resulting in a huge gap between theoretical production scheduling research and real-world production scheduling.

In a real-world production environment, various types of vehicles (e.g., trucks, conveyors and cranes) are commonly used to convey jobs with selected speeds between different machines. In recent years, automated guided vehicles (AGVs) [14,15], among various other vehicles, have been gradually introduced in advanced intelligent manufacturing systems to shorten the transport time of jobs and improve productivity. AGVs differ from traditional transport equipment in that they are integrated with emerging technologies (e.g., high-precision sensors and the Internet of Things), can acquire real-time information data, and make possible the interconnections between different production equipment [16,17]. These intelligent AGVs can avoid collisions through installed sensors and travel on guided paths [18]. These features allow them to greatly improve the robustness and flexibility of manufacturing systems. In complex, customization-oriented manufacturing environments involving products of multiple varieties and small batches, AGVs have become key equipment for achieving intelligent production [19]. However, when integrated in production systems, AGVs not only affect production efficiency, but also consume considerable energy. Therefore, studying energy-efficient production scheduling integrated with AGVs has both theoretical and practical significance. Some researchers have begun to integrate AGVs in production scheduling to solve scheduling problems associated with machines and AGVs [13,18–20]. However, the relevant studies only focus on single-objective scheduling that aims to minimize the makespan. In addition, the effect of AGV configuration on the performance improvement of production systems has not been studied.

In other words, no energy-efficient production scheduling studies that consider the transport of AGVs have been reported. Many new logistics constraints directly related to AGVs will have to be inserted into the production scheduling systems when they are integrated with AGVs—in addition to constraints related to production. We need to conduct a cooperative scheduling of operations, machines and AGVs. The problem is much more complex than common energy-efficient production scheduling. Hence, an effective optimization approach is necessary to handle this problem.

The job-shop scheduling problem (JSP) is one of the most general classical production scheduling problems [1,21–25]. Many real-world manufacturing problems can be formulated as JSP-based models. Therefore, it is of great practical significance to integrate AGVs into a JSP environment and conduct energy-efficient scheduling research. In recent years, various evolutionary algorithms have been used to solve the JSP. For example, Wu and Wu [26] developed an elitist quantum-inspired evolutionary algorithm for flexible job-shop scheduling, aiming to minimize the makespan. Chiang and Lin [27] proposed a multiobjective evolutionary algorithm (MOEA) to address a flexible JSP, with the makespan, total workload and maximum workload in mind. Shen and Yao [28] studied a dynamic multiobjective flexible JSP and developed an MOEA-based proactive-reactive method. The novelty of their method lies in its ability to handle multiple objectives simultaneously, including efficiency and stability, and adapt to a new environment quickly by incorporating heuristic-based dynamic optimization strategies. Sarker et al. [29] presented a hybrid evolutionary algorithm for solving the JSP, taking the effect of machine maintenance into account. Abedi et al. [30] designed an MOEA incorporated with disjunctive graph-based local search for energy-efficient job-shop scheduling with deteriorating machines. These existing studies showed that MOEAs are very promising in solving production scheduling problems.

In this paper, we focus on a specific JSP environment integrated with multiple AGVs. The main innovations and contributions of this work are summarized as follows:

(1) This is the first study in which an energy-efficient JSP considering multiple AGVs (EJSP-AGVs) is addressed. We formulate a new multiobjective model to minimize the makespan, total idle time of the machines and total energy consumption of both the machines and AGVs.

(2) An effective MOEA (EMOEA) is then proposed to tackle the problem. An efficient problem-based encoding/decoding operator is proposed. Novel crossover and mutation operators are adopted for the problem. An opposition-based learning (OBL) strategy is utilized to improve the local search ability.

(3) The model is validated with a CPLEX solver. The effect of AGV configuration on production system performance is discussed. The comparison results on benchmark instances and a real-world case show that the proposed EMOEA outperforms three well-known algorithms in solving the problem.

The remainder of this paper is organized as follows. In Section 2, we review the related work. In Section 3, we describe and formulate the problem at hand in detail. In Section 4, we present the proposed algorithm. In Section 5, we study the impact of AGV configuration on the overall production performance and evaluate the performance of the proposed algorithm. Finally, we conclude the paper and outline several future research directions in Section 6.

2. Related work

2.1. Energy-efficient production scheduling

There has been a growing interest in studies related to energy-efficient production scheduling, with He et al. [31] and Subai et al. [32] among the first to do so. A review of energy-efficient production scheduling was provided by Gahm et al. [33]. Mouzon et al. [8] observed that overall production energy consumption can be reduced by a machine turn-on/off mechanism. Many energy efficiency production scheduling studies with turn-on/off mechanisms, inspired by the work of Mouzon et al. [8], have emerged in recent years. Che et al. [9] addressed single-machine scheduling with a power-down mechanism to simultaneously minimize maximum tardiness and energy consumption. Shrouf et al. [10] discussed an energy efficiency single-machine scheduling problem with a turn-on/off mechanism and proposed a genetic algorithm (GA) to minimize the energy costs. Using a turn-on/off mechanism, Dai et al. [5] proposed an energy-efficient model for flexible flow shop scheduling. To address the multiobjective model, an improved genetic-simulated annealing algorithm was adopted to minimize the makespan and the total energy consumption. Wu and Sun [34] presented a nondominated sorting GA (NSGA) to solve flexible JSPs, with the aim of simultaneously minimizing the makespan, energy consumption and number of times that machines were turned on and off. Gong et al. [35] designed four rules to set the machine on/off criteria, maintenance periods and predefined maintenance windows, and three heuristics to insert maintenance activities into their solutions and move the maintenance-operation blocks to optimize the objectives.

Contrary to the aforementioned studies, Fang et al. [36] noticed that the turn-on/off mechanism was not always applicable in real-world manufacturing systems because frequently restarting the machines required a considerable amount of additional energy and could potentially cause damage to the machines. Therefore, they proposed an energy-efficient mechanism called dynamic speed scaling, which meant that machines could process jobs at different speeds. A higher processing speed led to less processing time and more energy consumption [4,36]. The study demonstrated that selecting reasonable operation assignments and machine speeds can lead to large reductions in energy use. The speed-scaling mechanism has been widely adopted in recent energy-efficient scheduling studies. For example, in a speed-scaling job-shop scheduling environment, Zhang and Chiong [7] minimized the total weighted tardiness and total energy consumption by a multiobjective GA with local search. Mansouri et al. [37] discussed a two-machine flow shop scheduling problem with a speed-scaling mechanism to minimize the makespan and total energy consumption. Ding et al. [38] minimized total carbon emissions and the makespan in a speed-scaling permutation flow shop scheduling problem. They proposed a multiobjective NEH algorithm and a modified multiobjective iterated greedy algorithm to solve this multiobjective problem. Yin et al. [39] studied an energy efficiency JSP with flexible spindle speed and proposed a novel GA to minimize the makespan, energy consumption and noise emissions.

In addition to the two aforementioned types of energy efficiency scheduling studies, a third type, based on the policy of time-of-use (TOU) electricity price, is also a hot topic. In TOU, power suppliers offer electricity with a dynamic price scheme, in which higher electricity prices are generated during peak hours and lower prices during off-peak hours. It is therefore feasible to reduce production costs by shortening the processing time in peak periods and prolonging the processing time in off-peak periods. Using the TOU scheme, Ding et al. [40] studied the unrelated parallel machine scheduling problem, with the aim of minimizing the total electricity cost by appropriately scheduling jobs and the overall completion time. Che et al. [41] proposed a new continuous-time mixed-integer linear programming model for energy-conscious single-machine scheduling, and a greedy insertion heuristic was developed to minimize electricity costs. Luo et al. [42] studied a hybrid flow shop scheduling problem with machine electricity consumption costs. To reduce electricity consumption during peak and off-peak hours, they recommended combining fast and slow processing machines to obtain higher energy efficiency using constant power/speed ratios. Zeng et al. [43] constructed a multiobjective optimization model in a flexible flow shop system, where a hybrid NSGA-II was employed to minimize the makespan, electricity consumption and material wastage. Zeng et al. [44] formulated a biobjective mixed-integer linear programming model for a uniform parallel machine scheduling problem, considering electricity cost under time-dependent or TOU electricity tariffs.

2.2. Production scheduling integrated with AGVs

Few studies on production scheduling integrated with AGVs have been reported. In the work of Chaudhry et al. [45], a problem concerning the scheduling of machines and two AGVs in a flexible manufacturing system (FMS) was solved using a GA. They compared the results of the proposed GA with four GA variants from the existing literature. In their experiments, the AGVs followed the rule of finding the shortest path to other machines. Nageswararao et al. [46] employed a sheep flock optimization algorithm to schedule both the machines and AGVs' processes with makespan and mean tardiness minimization. They tested their algorithm on four basic layouts and ten job sets, but did not present the mathematical model of the problem. Saidi-Mehrabad et al. [13] studied a basic JSP concerning the transport processes of AGVs. In this problem, a conflict-free routing problem to avoid the collision of AGVs was considered. A two-stage ant colony algorithm was proposed to minimize the makespan. Lin et al. [18] focused on the dispatching of AGVs in an FMS. They modeled the AGV system by using a network structure and postulated that collisions for AGVs were avoided by hardware. A priority-based GA was proposed for solving this problem, in which the objective was to minimize the makespan without considering the mathematical model. Liu et al. [19] proposed an improved flower pollination algorithm to handle the integrated scheduling problem of machines and AGVs in a basic JSP environment to minimize the makespan and utilization of AGVs. They assumed that AGVs could navigate autonomously to avoid obstacles and replan their path. Lacomme et al. [20] addressed the JSP of simultaneous scheduling of machines and identical AGVs to minimize the makespan. They introduced a framework-based on a disjunctive graph to model the joint scheduling problem and a memetic algorithm for machine and AGV scheduling. Caumond et al. [47] considered a scheduling problem in a small and medium-sized FMS with only one AGV. A heuristic approach based on a branch-and-bound procedure was used to handle this problem. Zheng et al. [48] studied a scheduling problem of machines and AGVs in an FMS environment. A heuristic algorithm based on tabu search was proposed to minimize the makespan. Abdelmaguid et al. [49] proposed a hybrid GA to handle a production scheduling problem integrated with only two AGVs with the objective of makespan minimization. They also did not detail the mathematical model of the scheduling problem. Babu et al. [50] developed a differential evolution algorithm for the simultaneous scheduling of machines and two identical AGVs in an FMS. Reddy and Rao [51] addressed a multiobjective scheduling problem in an FMS with two AGVs, in which both machine and AGV scheduling were considered. The problem was handled by an NSGA in the absence of a mathematical model. Udhayakumar and Kumanan [52] investigated an integrated scheduling problem in an FMS with only one AGV. Without considering the mathematical model of the problem, they minimized the total travel distance of AGVs and the total number of backtracking movements.

2.3. Research gaps

Our literature review reveals that both energy-efficient production scheduling and scheduling integrated with AGVs have received increased attention. However, there are still some research gaps to be filled. The existing research gaps can be summarized as follows.

- The existing mathematical models for energy-efficient production scheduling problems, such as models developed by Dai et al. [5], Fang et al. [36] and Ding et al. [38], only focus on processing time and overlook logistics time, such as AGV transport time and sequence-dependent setup time (SDST). Many logistics activities are related to machines or jobs and cannot be simply overlooked or simplified. It would not be conducive to the optimization of production efficiency to do so and could not effectively guide actual production.

- Almost all existing studies on energy-efficient production scheduling considered only energy consumption in the machine processing stage and the machine idle stage. No studies on energy efficiency production scheduling integrated with AGVs have been reported. This means that the existing energy-efficient models cannot accurately describe energy consumption in the actual production process.
- Most previous studies on production scheduling integrated with AGVs, such as Chaudhry et al. [45] and Abdelmaguid et al. [49], focused on a single-objective problem – aiming to minimize the makespan – and did not present a corresponding mathematical model. No relevant studies have taken the environmental objective into account. Most studies only considered one or two AGVs and did not study how many AGVs should be configured to achieve the best production system performance, which is an important aspect to balance production costs and efficiency.

Based on the above gaps identified, this work focuses on energy-efficient production scheduling by taking AGV transport and sequence-dependent setup activities into account, which better meets the requirements of practical production. We took the classic JSP as the scheduling environment and integrated multiple AGVs into the JSP environment to simultaneously improve economic and environmental objectives.

3. Problem definition and mathematical modeling

The EJSP-AGVs can be described as follows.

1. There are n different jobs to be processed on m machines. Each job consists of o_i $(i = 1, 2, \dots, n)$ operations and should be processed through the machines in a predefined order. There are ag AGVs in the workshop. Each AGV has a discrete and finite multilevel transport speed. An AGV first transports the material of a job from a material warehouse (MW) to one machine for the first process of the job, then transports the job to other machines for the other operations and, finally, transports the finished job to a product warehouse (PW). Each AGV has three statuses: nonloading, loading and idle waiting. Each job i has $o_i + 1$ transport tasks. Each transport task in a job includes a 'no-load' and a 'load' transport process. In the no-load transport process, an AGV runs in the nonloading status from its current position to the machine position of the current job to prepare for loading the job. In the load transport process, an AGV loads the job and transports it from the current machine to another machine for the next process. To simplify the transport process, we defined the pickup position of MW as 0, the pickup/delivery (P/D) positions from machine 1 to machine m as $1, 2, \dots, m$, and the D position of the PW as $m + 1$. For all transport tasks in a job, the selected AGVs and their speeds are different. The transport time of an AGV is related to the transport distance and the AGV's speed. A higher AGV speed leads to more energy consumption but contributes to shortening the transport time. AGVs and machines have sensing abilities, and the interconnection between AGVs and machines is achieved through a wireless sensor network arranged in the workshop. AGVs can navigate autonomously and choose an appropriate transport route.
2. When the job is transported to a machine position, the AGV immediately unloads the job to perform the next transport task. Before the job is processed on the current machine, setup operations such as clamping, cleaning and tool-changing will be executed. These setup operations are related to the machine and the adjacent jobs processed on the same machine [53–55]. They will generate SDSTs and consume a certain amount of energy. Only when the job is transported to the machine, the corresponding setup operations can be started, which means that there is no overlap between transport and setup operations.
3. After the AGV's transport task and setup operations are complete, job processing will start on the machine. Each machine uses the speed-scaling mechanism and has a discrete and finite multilevel processing speed to handle an operation. A higher machine speed will shorten the processing time of jobs but increase energy consumption. Each machine cannot be turned off completely unless all of the operations on it have been completed. The basic processing time of each job on each machine is known in advance. There is a certain amount of stand-by energy consumption during the idle period of each machine.

The aim of the problem was to find a reasonable schedule that simultaneously optimized economic and environmental objectives. To reach this goal, four subproblems needed to be addressed: (1) sequencing all operations on the machines; (2) assigning a machine processing speed to each operation; (3) assigning an AGV to each transport task of jobs and (4) assigning an AGV speed to each transport task of jobs. Fig. 1 presents an example of the EJSP-AGVs. The assumptions are as follows.

1. All the machines, jobs and AGVs are simultaneously available at time zero.
2. All the jobs and AGVs are initially at the material warehouse.
3. Each machine can perform only one operation at a time and each operation can be processed only once on one machine.
4. There are no precedence constraints among the operations of different jobs.
5. Once an operation has begun on a machine, it cannot be interrupted.
6. An operation of any job cannot be performed until all preceding operations of the relevant job have been completed.
7. The speed of a machine cannot be changed once it begins to process an operation.
8. The buffer of each machine is infinite.
9. Each job can be transported by only one AGV at a time and each AGV can transport only one job at a time.
10. Nonloading and loading status will not affect the speed of an AGV in a transport task.
11. The transport of each AGV cannot be interrupted once it has begun. Preemption is not considered.
12. AGVs can eliminate collision automatically through the sensor network. The time required for collision elimination is ignored.
13. Charging and failure of AGVs are not considered.

Notations

n: the total number of jobs
m: the total number of machines
ag: the total number of AGVs
O_{ij}: the jth operation of job i
o_i: the total number of operations of job i
t_{ijk}: the processing time of operation O_{ij} on machine k
S_{ij}: the starting time of operation O_{ij}

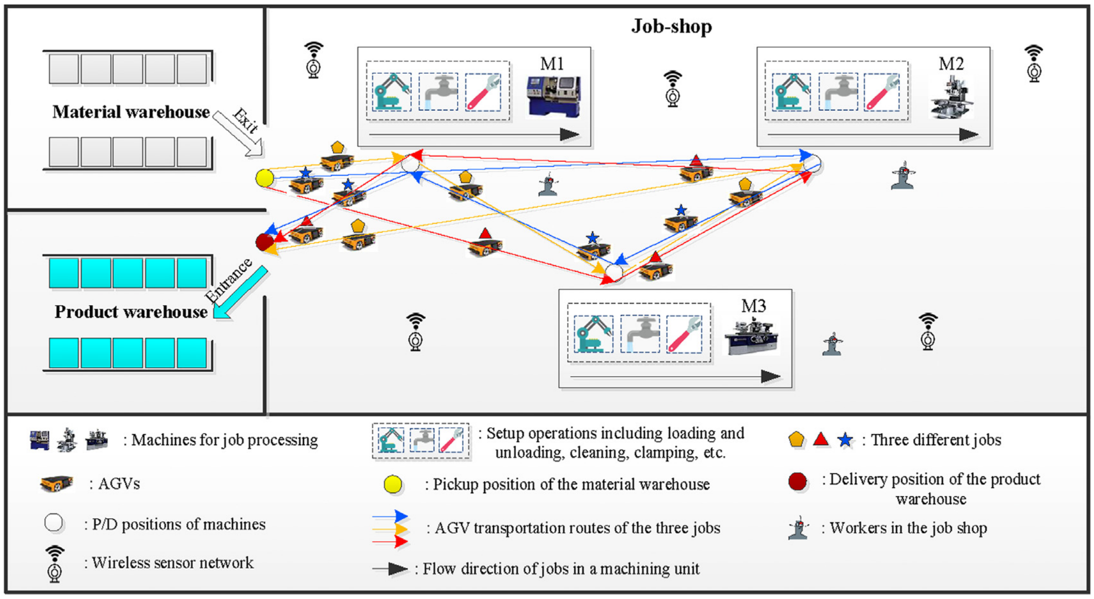

Job-shop
Material warehouse
Exit
Product warehouse
Entrance
M1
M2
M3
: Machines for job processing
: AGVs
: P/D positions of machines
: Wireless sensor network
: Setup operations including loading and unloading, cleaning, clamping, etc.
: Pickup position of the material warehouse
: AGV transportation routes of the three jobs
: Flow direction of jobs in a machining unit
: Three different jobs
: Delivery position of the product warehouse
: Workers in the job shop

Fig. 1. An example of the EJSP-AGVs scenario.

C_{ij}: the completion time of operation O_{ij}
C_i: the completion time of job i
C^k: the completion time of machine k
T_{if}: the fth transport task of job $i, f = 1, 2, \ldots, o_i, o_i + 1$
P': starting position of the no-load transport task T_{if}
P: ending position of the no-load transport task T_{if}
Q': starting position of the load transport task T_{if}
Q: ending position of the load transport task T_{if}
NST_{if}: starting time of the no-load transport task T_{if}
NET_{if}: ending time of the no-load transport task T_{if}
LST_{if}: starting time of the load transport task T_{if}
LET_{if}: ending time of the load transport task T_{if}
D_{pq}: the transport distance between two positions p and q; $D_{pq} = 0$, if $p = q$.
EA^h: the end time of AGV h
L: the total number of speed levels of each machine
R: the total number of speed levels of each AGV
v_l: the lth speed of a machine, $l = 1, 2, \ldots, L$
V_r: the rth speed of an AGV, $r = 1, 2, \ldots, R$
$S_{i'ik}$: the SDST of machine k for processing job i immediately after job i'
S_{0ik}: the setup time of machine k when job i is the first job processed on machine k
P_{kl}: the processing power of machine k when its processing speed is set to v_l
PS_k: the stand-by power of machine k
PST_k: the setup power of machine k
PA_{hr}: the transport power of AGV h when its transport speed is set to V_r
PAS_h: the stand-by power of AGV h
LP: a very large positive integer

Variables:

$$x_{ijk} = \begin{cases} 1, & \text{if operation } O_{ij} \text{ is processed on machine } k \\ 0, & \text{otherwise} \end{cases}$$

$$y_{i'j'ijk} = \begin{cases} 1, & \text{if operation } O_{i'j'} \text{ is processed before operation } O_{ij} \\ & \text{on the same machine } k \\ 0, & \text{otherwise} \end{cases}$$

$$z_{ik'k} = \begin{cases} 1, & \text{if machine } k' \text{ processes job } i \text{ before machine } k \\ 0, & \text{otherwise} \end{cases}$$

$$u_{i'ik} = \begin{cases} 1, & \text{if job } i \text{ is processed immediately after job } i' \\ & \text{on machine } k \\ 0, & \text{otherwise} \end{cases}$$

$$\alpha_{ifh} = \begin{cases} 1, & \text{if transport task } T_{if} \text{ is processed on AGV } h \\ 0, & \text{otherwise} \end{cases}$$

$$\omega_{P'P} = \begin{cases} 1, & \text{if the nonloading starting and ending positions} \\ & \text{of } T_{if} \text{ are } P' \text{ and } P, \text{ respectively} \\ 0, & \text{otherwise} \end{cases}$$

$$\varphi_{Q'Q} = \begin{cases} 1, & \text{if the loading starting and ending positions for} \\ & \text{loading of } T_{if} \text{ are } Q' \text{ and } Q, \text{ respectively} \\ 0, & \text{otherwise} \end{cases}$$

$$\beta_{i'f'ifh} = \begin{cases} 1, & \text{if transport task } T_{i'f'} \text{ is processed before} \\ & \text{transport task } T_{if} \text{ on the same AGV } h \\ 0, & \text{otherwise} \end{cases}$$

$$\gamma_{ijkl} = \begin{cases} 1, & \text{if the speed of machine } k \text{ for} \\ & \text{operation } O_{ij} \text{ is set to } v_l \\ 0, & \text{otherwise} \end{cases}$$

$$\sigma_{ifhr} = \begin{cases} 1, & \text{if the transport speed of AGV } h \text{ for transport task} \\ & T_{if} \text{ is set to } V_r \\ 0, & \text{otherwise} \end{cases}$$

Objective functions:

1. Maximum ending time of AGVs:

$$f_1 = \max\{EA^h | h = 1, 2, \cdots, ag\} \tag{1}$$

f_1 is the makespan. In this work, makespan is defined as the maximum ending time of all AGVs for transporting all jobs to the PW.

2. Total idle time of all machines:

$$f_2 = \sum_{k=1}^{m} [C^k - \sum_{i=1}^{n} \sum_{j=1}^{o_i} \sum_{l=1}^{L} x_{ijk}\gamma_{ijkl} \frac{t_{ijk}}{v_l} - \sum_{i'=0, i=1}^{n} u_{i'ik} S_{i'ik}] \tag{2}$$

3. Total energy consumption of all machines and AGVs:

$$f_3 = E1 + E2 + E3 + E4 + E5 \tag{3}$$

$$E1=\sum_{k=1}^{m}\sum_{i=1}^{n}\sum_{j=1}^{o_i}\sum_{l=1}^{L}x_{ijk}\gamma_{ijkl}\frac{t_{ijk}}{v_l}P_{kl} \tag{4}$$

$$E2=\sum_{k=1}^{m}\sum_{i'=0,i=1}^{n}u_{i'ik}S_{i'ik}PST_k \tag{5}$$

$$E3=\sum_{k=1}^{m}PS_k[C^k-\sum_{i=1}^{n}\sum_{j=1}^{o_i}\sum_{l=1}^{L}x_{ijk}\gamma_{ijkl}\frac{t_{ijk}}{v_l}-\sum_{i'=0,i=1}^{n}u_{i'ik}S_{i'ik}] \tag{6}$$

$$E4=\sum_{h=1}^{ag}\sum_{r=1}^{R}\sum_{i=1}^{n}\sum_{f=1}^{o_i+1}(\sum_{P'=0}^{m+1}\sum_{P=0}^{m+1}\omega_{P'P}\frac{D_{P'P}}{V_r}+\sum_{Q'=0}^{m+1}\sum_{Q=0}^{m+1}\varphi_{Q'Q}\frac{D_{Q'Q}}{V_r})\alpha_{ifh}\sigma_{ifhr}PA_{hr} \tag{7}$$

$$E5=\sum_{h=1}^{ag}PAS_h[EA^h-\sum_{r=1}^{R}\sum_{i=1}^{n}\sum_{f=1}^{o_i+1}(\sum_{P'=0}^{m+1}\sum_{P=0}^{m+1}\omega_{P'P}\frac{D_{P'P}}{V_r}+\sum_{Q'=0}^{m+1}\sum_{Q=0}^{m+1}\varphi_{Q'Q}\frac{D_{Q'Q}}{V_r})\alpha_{ifh}\sigma_{ifhr}] \tag{8}$$

In Eq. (3), f_3 is the total energy consumption of all machines and AGVs. Eqs. (4)–(6) refer to the processing, setup and stand-by energy consumption of machines, respectively. Eqs. (7)–(8) are the transport and stand-by energy consumption of AGVs, respectively.

Constraints:

$$C_{i'j'}+x_{ijk}\gamma_{ijkl}\frac{t_{ijk}}{v_l}+u_{i'ik}S_{i'ik}\leq C_{ij}+(1-y_{i'j'ijk})LP \tag{9}$$

where $\forall i,i'=1,2,\ldots,n;\ j=1,2,\ldots,o_i;\ j'=1,2,\ldots,o_{i'};\ k=1,2,\ldots,m;\ l=1,2,\ldots,L$

$$C_{i(j-1)}+x_{ijk}\gamma_{ijkl}\frac{t_{ijk}}{v_l}+u_{i'ik}S_{i'ik}\leq C_{ij}+(1-z_{ik'k})LP \tag{10}$$

where $\forall i,i'=1,2,\ldots,n;\ j=2,\ldots,o_i;\ k,k'=1,2,\ldots,m;\ l=1,2,\ldots,L$

$$NST_{if}+(1-\beta_{i'f'ifh})LP\geq LET_{i'f'} \tag{11}$$

where $\forall i,i'=1,2,\ldots,n;\ f=1,2,\ldots,o_i;\ f'=1,2,\ldots,o_{i'};\ h=1,2,\ldots,ag$

$$NET_{if}\geq NST_{if}+\sum_{P'=0}^{m+1}\sum_{P=0}^{m+1}\omega_{P'P}\frac{D_{P'P}}{V_r}\alpha_{ifh}\sigma_{ifhr} \tag{12}$$

where $\forall i=1,2,\ldots,n;\ f=1,2,\ldots,o_i+1;\ h=1,2,\ldots,ag;\ r=1,2,\ldots,R$

$$LET_{if}\geq LST_{if}+\sum_{Q'=0}^{m+1}\sum_{Q=0}^{m+1}\omega_{Q'Q}\frac{D_{Q'Q}}{V_r}\alpha_{ifh}\sigma_{ifhr} \tag{13}$$

where $\forall i=1,2,\ldots,n;\ f=1,2,\ldots,o_i+1;\ h=1,2,\ldots,ag;\ r=1,2,\ldots,R$

$$LST_{if}\geq \max(NET_{if},C_{i(j-1)}) \tag{14}$$

where $\forall i=1,2,\ldots,n;\ f=2,\ldots,o_i+1;\ j=2,\ldots,o_i$

$$S_{ij}\geq \max\{\max\left(LET_{if},C_{i'j'}\right)+S_{i'ik}\} \tag{15}$$

where $\forall i,i'=1,2,\ldots,n;\ i\neq i';\ f=1,2,\ldots,o_i+1;\ j'=1,2,\ldots,o_{i'};\ k=1,2,\ldots,m$

$$\sum_{k=1}^{m}x_{ijk}=1,\forall i=1,2,\ldots,n;j=1,2,\ldots,o_i \tag{16}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{o_i}x_{ijk}=1,\forall k=1,2,\ldots,m \tag{17}$$

$$\sum_{i'=0}^{n}u_{i'ik}=1,\forall i=1,2,\ldots,n;\ i\neq i';\ k=1,2,\ldots,m \tag{18}$$

$$\sum_{i=1}^{n}u_{i'ik}=1,\forall i'=0,1,2,\ldots,n;\ i\neq i';\ k=1,2,\ldots,m \tag{19}$$

$$\sum_{h=1}^{ag}\alpha_{ifh}=1,\forall i=1,2,\ldots,n;f=1,2,\ldots,o_i+1 \tag{20}$$

$$\sum_{i=1}^{n}\sum_{f=1}^{o_i+1}\alpha_{ifh}=1,\forall h=1,2,\ldots,ag \tag{21}$$

$$\sum_{l=1}^{L}\gamma_{ijkl}=1,\forall i=1,2,\ldots,n;j=1,2,\ldots,o_i;k=1,2,\ldots,m \tag{22}$$

$$\sum_{r=1}^{R}\sigma_{ifhr}=1,\forall i=1,2,\ldots,n;f=1,2,\ldots,o_i+1;h=1,2,\ldots,ag \tag{23}$$

$$\sum_{P'=0}^{m+1}\omega_{P'P}=1,\forall P=0,1,2,\ldots,m+1 \tag{24}$$

$$\sum_{P=0}^{m+1}\omega_{P'P}=1,\forall P'=0,1,2,\ldots,m+1 \tag{25}$$

$$\sum_{Q'=0}^{m+1}\varphi_{Q'Q}=1,\forall Q=0,1,2,\ldots,m+1 \tag{26}$$

$$\sum_{Q=0}^{m+1}\varphi_{Q'Q}=1,\forall Q'=0,1,2,\ldots,m+1 \tag{27}$$

Eqs. (9)–(10) are the operation precedence process constraints. Eq. (11) means that the AGV can start the next transport task only after it completes the previous transport task. Eq. (12) means that the nonloading ending time of one transport task must be greater than or equal to its nonloading starting time plus the nonloading duration time. Eq. (13) implies that the ending time of loading a transport task must be greater than or equal to its loading starting time plus the loading duration time. Eqs. (14)–(15) are the constraints between the production operation and transport task. Eq. (14) ensures that a job can be transported to the next machine for processing the next operation only after the current operation is completed and the AGV arrives can the job be transported to the next machine for processing the next operation. Eq. (15) ensures that the operation of the current job can be started only after job is transported to the corresponding machine position and the previous operation and setup operation on the same machine are completed. Eq. (16) ensures that each operation can only be processed on one machine at a time, and Eq. (17) ensures that each machine can only process one operation at a time. Eq. (18) indicates that a job must follow exactly one predecessor except when it is the first job of the machine. Eq. (19) means that when a job has finished the processing on one machine, exactly one different job can be selected for processing afterward, except when the job is the last job of the machine. Eq. (20) implies that each transport task can be performed by only one AGV at a time, and Eq. (21) means that each AGV can perform only one transport task at a time.

Algorithm 1: Pseudocode of the EMOEA

```
1: Execute the encoding and create an initial population $P$ with $N$ feasible solutions
2: $gen = 1$    %$gen$ is the current generation
3: while $gen < G_{max}$ do   %$G_{max}$ is the maximum number of generations
4:     Decode the solutions in the population and calculate their three objective function values
5:     Evaluate the population by nondominated sorting
6:     if $gen = 1$ then
7:         Construct the initial external archive
8:     end if
9:     Conduct three kinds of crossover operators on the current population
10:    Conduct the mutation operator on the current population
11:    Execute the OBL strategy on the current population
12:    if $gen > 1$ then
13:        Update the external archive
14:    end if
15:    $gen = gen + 1$
16: end while
17: Output the external archive
```

Eq. (22) ensures that only one machine speed can be selected for each operation. Eq. (23) ensures that only one AGV speed can be selected for each transport task. Eqs. (24)–(25) mean that each transport task has only one no-load starting position and only one no-load ending position, respectively. Eqs. (26)–(27) mean that each transport task has only one load starting position and only one load ending position, respectively.

4. Proposed methods

In the EJSP-AGVs, logistics activities, including the AGV transport and sequence-dependent setup, are considered, which means we need to cooperatively schedule jobs, machines and AGVs. In this section, we describe the proposed EMOEA and its components, which include an efficient encoding approach to encode the jobs, machines and AGVs; three different crossover operators; and a mutation operator. An OBL strategy is also introduced to achieve the balance between exploration (global search) and exploitation (local search). Furthermore, an external archive technique is utilized to store elite solutions found by the EMOEA. **Algorithm 1** below summarizes each process of the proposed methods.

4.1. Encoding

For the EJSP-AGVs, a feasible candidate solution π should consist of four parts, namely, the operation sequence (O), the machine speed sequence (v), the AGV sequence (A) and the AGV speed sequence (V). That is, $\pi = [O, v, A, V]$. To obtain the feasible solution, a four-layer encoding method was designed.

For O and v, the encoding method is described as follows.

Step 1: Randomly generate a series of real numbers composed of $n \times m$ dimensions. The real number in each dimension is in the range (0, 10).

Step 2: Rank the $n \times m$ real numbers in descending order and obtain an integer series from 1 to $n \times m$. The integer series is denoted as $X = [X(1), X(2), \ldots, X(s), \ldots, X(n \times m)]$.

Step 3: For integer $X(s)$ on each dimension, use $1 + X(s)\ mod\ n$ to obtain a job index. Scanning all the job indices from left to right, each job index has m occurrences, which correspond to the number of operations for a job. Each job index represents an operation O_{ik}, $i = 1, 2, \ldots, n$; $k = 1, 2, \ldots, m$. With this method, a feasible O is always obtained.

Step 4: For each operation O_{ik} in O, a corresponding integer l in the range $[1, L]$ is obtained randomly. A machine speed v_l is assigned to operation O_{ik}. These integers in the range $[1, L]$ construct a feasible v.

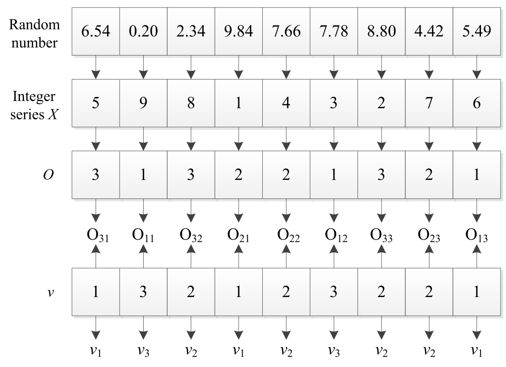


Fig. 2. An encoding example for operation and machine speed sequences.

For A and V, the encoding method is described as follows.

Step 1: Randomly generate a series of integers consisting of $n \times (m + 1)$ dimensions. In each dimension, the integer is in the range $[1, ag]$, and ag is the number of AGVs.

Step 2: This string of integers is divided into n segments from left to right. Each segment represents the assignment of AGVs to one job's transport tasks. Each segment consists of $m + 1$ dimensions since there are $m + 1$ transport tasks for each job. The first integer in a segment refers to the AGV index corresponding to the transport tasks from the material warehouse to the machine for the first operation of the job and so on. The last integer in the segment is the AGV index corresponding to the last transport task from the machine for the last operation of the job to the PW. A feasible A can always be obtained via this encoding process.

Step 3: For each AGV index $A(g)$ in A, $g = 1, 2, \ldots, n \times (m+1)$, randomly generate a corresponding integer r from the interval $[1, R]$ and assign a speed V_r to the corresponding AGV. Finally, a feasible V is always obtained.

An example case of the EJSP-AGVs case with three jobs, three machines and two AGVs is used to demonstrate the encoding process in Figs. 2–3. In this example, the machine speed is set as $v = \{v_1, v_2, v_3\}$ and AGV's speed is set as $V = \{V_1, V_2, V_3\}$.

4.2. Decoding

An active decoding method was used to decode each encoded individual. Details of this decoding method are presented as follows.

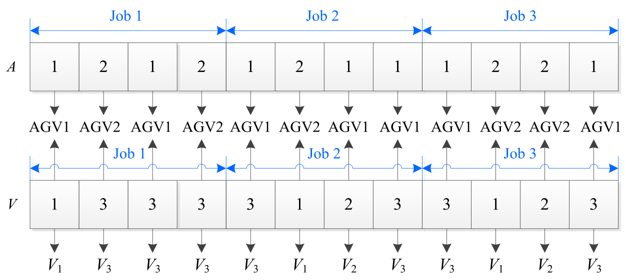

Job 1
Job 2
Job 3
A
1 2 1 2 1 2 1 1 1 2 2 1
AGV1 AGV2 AGV1 AGV2 AGV1 AGV2 AGV1 AGV1 AGV1 AGV2 AGV2 AGV1
V
1 3 3 3 3 1 2 3 3 1 2 3
V1 V3 V3 V3 V3 V1 V2 V3 V3 V1 V2 V3

Fig. 3. An encoding example for AGV and AGV speed sequences.

Step 1: Identify the set of operations processed on each machine and the set of machines used to process each job. Determine the set of machine speeds for the operations of each job.

Step 2: Determine the set of AGV indices and the set of AGV speeds for each transport task of each job. Identify the starting and ending position sets of no-load/load processes of each transport task.

Step 3: Calculate the ending time of each transport task T_{if} that transports job i to one machine position for processing operation O_{ij}. Let $LET_{if} = \max[C_{i(j-1)}, NST_{if} + D_{P'P}/V_r] + D_{Q'Q}/V_r$.

Step 4: Determine the completion time of operation $O_{i'j'}$ before operation O_{ij} on machine k. Let $C_{i'j'}$ be the completion time of operation $O_{i'j'}$ on machine k.

Step 5: Determine the starting time of operation O_{ij} on machine k. If $C_{i'j'} \leq LET_{if}$, the starting time of operation O_{ij} is $S_{ij} = LET_{if} + S_{i'ik}$; otherwise, $S_{ij} = C_{i'j'} + S_{i'ik}$.

Step 6: The completion time of operation O_{ij} on machine k can be $C_{ij} = S_{ij} + t_{ijk}/\upsilon_l$, in which t_{ijk} is the basic processing time of job i on machine k and υ_l is the processing speed of machine k for operation O_{ij}.

Step 7: Calculate the ending time of the last transport task of job i. Let $LET_{i(O_i+1)} = \max[C_{iO_i}, NST_{i(O_i+1)} + D_{P'P}/V_r] + D_{Q'Q}/V_r$, in which C_{iO_i} is the completion time of the last operation of job i and $LET_{i(O_i+1)}$ and $NST_{i(O_i+1)}$ are the load ending time and no-load time of the last transport task $T_{i(O_i+1)}$ of job i, respectively.

Fig. 4 shows a Gantt chart obtained by the active decoding method based on the example in Section 4.1. In Fig. 4, with regards to the AGV area, the white box indicates the no-load transport task of AGVs, while the magenta box indicates the load transport task of AGVs. '$T_{if} - V_r$' indicates the transport task T_{if} with AGV speed V_r. For the machine area, the yellow box indicates the sequence-dependent setup and '$S_{i'ik}$' indicates the SDST of machine k for processing two adjacent jobs i' and i. '$O_{ij} - \upsilon_l$' implies the operation O_{ij} with machine processing speed υ_l.

4.3. Crossover

The crossover operation started with randomly selecting two individuals from current population P. The crossover operation was repeated $N/2$ times, and its crossover probability was set as CR. Since the solution for EJSP-AGVs has four different parts, only one kind of crossover strategy may not be effective for the four parts or may even inevitably generate infeasible offspring solutions. In this case, a repair operation should be designed to repair the infeasible solutions, which will greatly increase the computation time. In this work, three types of crossover operators were performed on the four different parts of the selected individuals. The order crossover (OX) [56] was used for the operation sequence, the partial-mapped crossover (PMX) [57] was adopted for the machine speed sequence, and a novel crossover proposed in this paper was used for the AGV and AGV speed sequences. With these three types of crossover operators, we can always obtain feasible offspring solutions. Some preliminary experiments also showed that the performance of the algorithm was better for the EJSP-AGVs by combining the three types of crossover operators. On the other hand, it is very important to maintain the population diversity for evolutionary algorithms. Therefore, in evolutionary algorithms, the positions of crossover are generally determined in a random way to obtain a variety of offspring individuals. With this in mind, we also randomly determined the positions of crossover for the three types of crossover operators to guarantee population diversity in this work.

For operation sequence O, the OX crossover was conducted, and an example is presented in Fig. 5. The details of the OX crossover for O are as follows.

Step 1: Randomly select two operation sequence parents *PO1* and *PO2*. To maintain population diversity, randomly generate two integers $s1$ and $s2$ that meet the inequality $0 < s1 < s2 < LS$ to determine the positions of order crossover. LS is the length of the operation sequence O.

Step 2: For parent *PO1*, find the operations that fall between $s1$ and $s2$ to obtain a partial operation set *PP1*. Likewise, find the operations that fall between $s1$ and $s2$ in *PO2* to obtain a partial operation set *PP2*. Assuming that $s1 = 3$ and $s2 = 6$, as shown in Fig. 5, $PP1 = [O_{21}, O_{12}, O_{32}, O_{22}]$ and $PP2 = [O_{12}, O_{13}, O_{21}, O_{22}]$.

Step 3: Identify the *PP1* operations in *PO2* and determine their positions in *PO2* to obtain a position matrix *PM2*. Likewise, identify the *PP2* operations in *PO1* and determine their positions in *PO1* to obtain a position matrix *PM1*. Based on the example in Fig. 5, $PM1 = [3, 5, 6, 7]$ and $PM2 = [3, 4, 6, 7]$.

Step 4: The genes between $s1$ and $s2$ in *PO1* ([2, 1, 3, 2]) are inserted into offspring *O1* at the same positions. All genes in *PO2* are inserted into *O1* except those in the corresponding positions of $PM2$.

Step 5: The genes between $s1$ and $s2$ in *PO2* ([1, 1, 2, 2]) are inserted into offspring *O2* at the same positions. All genes in *PO1* are inserted into *O2* except those in the corresponding positions of $PM1$.

For machine speed sequence v, the PMX crossover is performed as shown in Fig. 6. Details of PMX crossover for v are as follows.

Step 1: Randomly generate two integers $l1$ and $l2$ that meet the inequality $0 < l1 < l2 < LS$, where LS is the length of the machine speed sequence v. Based on the example in Fig. 6, $l1 = 3$ and $l2 = 6$, respectively.

Step 2: Insert the genes between $l1$ and $l2$ in parent *Pv2* ([1, 3, 1, 2]) into offspring *v1* at the same positions. All genes in *Pv1* are copied to *v1* except those that fall between $l1$ and $l2$.

Step 3: Insert the genes between $l1$ and $l2$ in parent *Pv1* ([2, 1, 2, 3]) into offspring *v2* at the same positions. All genes in *Pv2* are copied to *v2* except those that fall between $l1$ and $l2$.

For AGV sequence A and AGV speed sequence V, we propose a crossover operator. The new crossover operator is shown in Fig. 7. Detailed steps of the proposed crossover are as follows.

Step 1: Divide both the AGV sequence and the AGV speed sequence into n (equal to the number of jobs) segments. Randomly generate two integers $h1$ and $h2$ that meet the inequality $0 < h1 < h2 \leq n$.

Step 2: Insert segment gene $h1$ in parent *PA1* into offspring *A2* at the position of segment $h2$. Insert segment gene $h2$ in parent *PA1* into offspring *A2* at the position of segment $h1$. All genes in *PA2* are copied to *A2* at the same positions except those in segments $h1$ and $h2$. Likewise, insert segment gene $h1$ in parent *PV1* into the offspring *V2* at the position of $h2$. Insert segment gene $h2$ in parent *PV1* into the offspring *V2* at the position of $h1$. All genes in *PV2* are copied to *V2* at the same positions except for those in segments $h1$ and $h2$.

Step 3: Insert segment gene $h1$ in parent *PA2* into offspring *A1* at the position of segment $h2$. Insert the segment gene $h2$ in

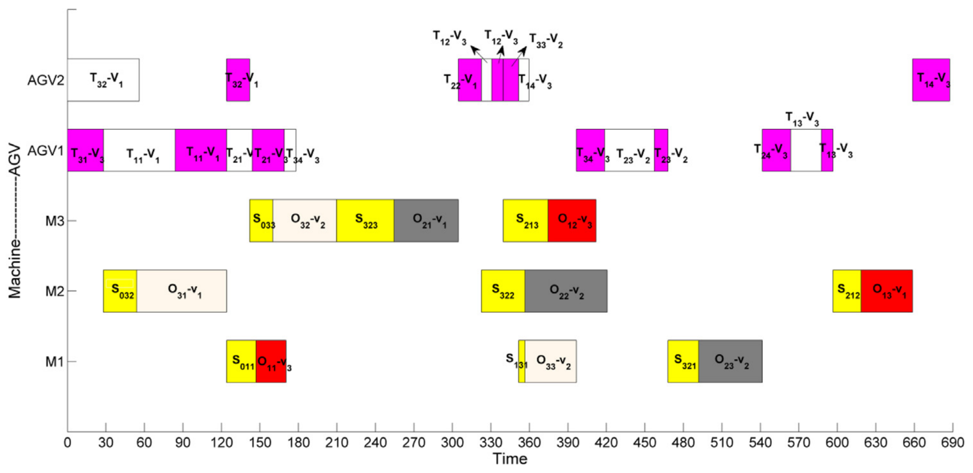


Fig. 4. A Gantt chart example obtained with the active decoding method.

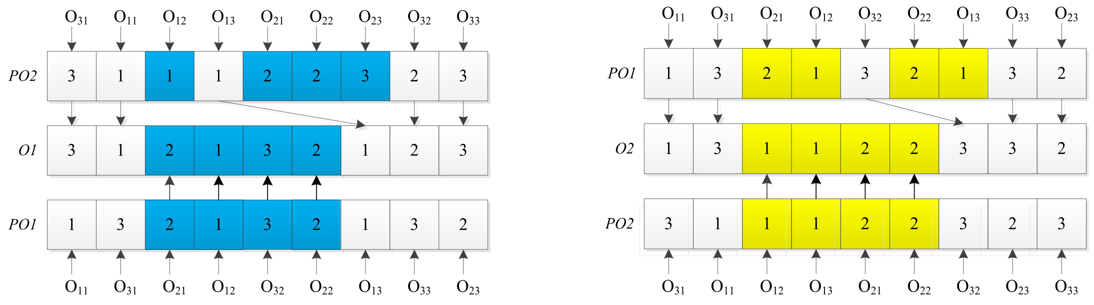


Fig. 5. A crossover example for the operation sequence.

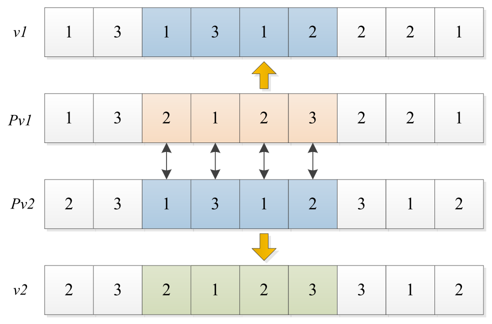


Fig. 6. A crossover example for the machine speed sequence.

parent *PA2* into the offspring *A1* at the position of segment $h1$. All genes in *PA1* are copied to *A1* at the same positions except those in segments $h1$ and $h2$. Likewise, insert the segment genes $h1$ in parent *PV2* into offspring *V1* at the position of segment $h2$. Insert the segment gene $h2$ in parent *PV2* into offspring *V1* at the position of segment $h1$. All genes in *PV1* are copied to *V1* at the same positions except for those in segments $h1$ and $h2$.

4.4. Mutation

For each individual obtained via the crossover operators, mutation is executed for the four parts with a pre-defined probability *MR*. There are many kinds of mutation operators in the literature, such as uniform and nonuniform mutation [58], Gaussian mutation [59], and polynomial mutation [60]. Some infeasible offspring solutions may be generated when these mutation strategies are applied to integer-encoded scheduling problems. Repair operations are therefore needed to make these offspring solutions feasible. As a result, these repair operations inevitably increase the computation time.

In contrast, point mutation strategies [61], such as single-point mutation, two-point mutation and multipoint mutation, can be applied directly to integer-encoded scheduling problems. When these point mutation strategies are applied in our EJSP-AGVs, the offspring solutions are always feasible. Through some preliminary experiments, we found that the performance of EMOEA with two-point mutation was better than that of EMOEA with other point mutation strategies. Therefore, considering both calculation time and performance, we chose two-point mutation in this work. The mutation points were generated randomly to maintain the diversity of the offspring population. Detailed steps of the mutation operator are as follows.

Step 1: Randomly generate two integers $g1$ and $g2$ that meet the inequality $0 < g1 < g2 < LS$, where LS is the length of the operation sequence or machine speed sequence. Randomly generate two integers $j1$ and $j2$ that meet the inequality $0 < j1 < j2 \leq n$, where n is the number of jobs.

Step 2: For a parent operation sequence, swap the genes at positions $g1$ and $g2$. Likewise, in a parent machine speed sequence, swap the genes at positions $g1$ and $g2$.

Step 3: For a parent AGV sequence, swap the segment gene $j1$ and the segment gene $j2$. Likewise, for a parent AGV speed sequence, swap the segment gene $j1$ and the segment gene $j2$. Fig. 8 shows an example of mutation for AGV and AGV speed sequences.

After the crossover and mutation operators, a new population Q was obtained. To obtain an elitist population, population Q was merged with population P and evaluated by nondominated sorting. N best solutions were selected from the evaluated population to generate a population H.

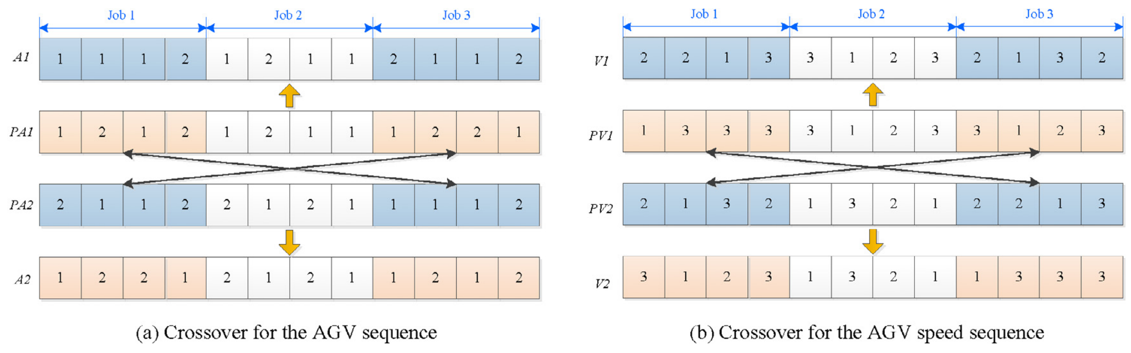

Job 1
Job 2
Job 3
A1
PA1
PA2
A2
V1
PV1
PV2
V2
(a) Crossover for the AGV sequence
(b) Crossover for the AGV speed sequence

Fig. 7. A crossover example for the AGV and AGV speed sequences.

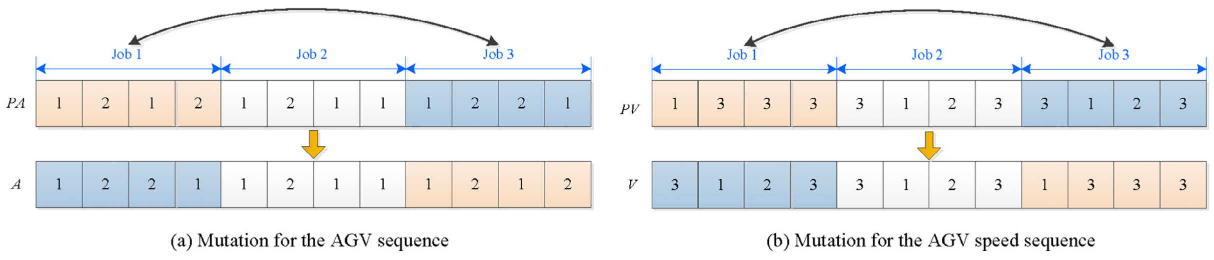

Job 1
Job 2
Job 3
PA
A
PV
V
(a) Mutation for the AGV sequence
(b) Mutation for the AGV speed sequence

Fig. 8. An example of mutation for the AGV and AGV speed sequences.

4.5. *Opposition-based learning*

We adopted OBL in the EMOEA to improve its search ability. This was helpful because OBL can learn from the opposite direction of the current search space and obtain opposite solutions in an unknown search space [62]. Many studies have reported that OBL helps algorithms achieve a balance between exploration (global search) and exploitation (local search) [63–65]. The basic definition of OBL-based optimization is as follows.

Definition 1 (*Opposite Solution*)**.** Assuming that there is an *Le*-dimensional solution $\pi = [\pi(1), \pi(2), \dots, \pi(s), \dots, \pi(Le)]$ in search space $\mathbf{\Omega}$, each dimension variable of π is updated by Eq. (28) to obtain the opposite solution π'.

$$\pi'(s) = \theta(l_s + u_s) - \pi(s) \tag{28}$$

where θ is a generalization coefficient and obeys the uniform distribution of [0,1]; l_s and u_s are the lower and upper limits of the *s*th dimension variable, respectively.

Definition 2 (*OBL-based Optimization*)**.** Solution π and its opposite solution π' are evaluated by a specified evaluation approach. The following three cases should be considered to obtain a better solution:

- If π is better than π', π will be retained, and π' will be discarded from the current population.
- If π' is worse than π, π' will be retained, and π will be discarded from the current population.
- If π and π' have the same evaluation performance, one of π and one of π' will be retained.

A new population, which contains several better solutions, can be obtained via the OBL strategy. In this work, each solution contained four different parts. The four parts in one solution had different lower and upper limits for each variable. Therefore, we needed to conduct OBL on the four parts separately. Since OBL would increase computing time, we only performed OBL on the best $r \times N$ individuals, where r is the rate of solutions executing OBL. Details of the OBL strategy for the EJSP-AGVs are shown in **Algorithm 2**.

4.6. *External archive technique*

External archiving is a common technique in many multiobjective algorithms [55]. This mechanism stores the elite solutions found in each generation of population *NP*. It involves the construction of the initial external archive (E^1) and the update of the external archive (E).

4.6.1. *Construction of the initial external archive*

In the first generation, the initial external archive was constructed as follows.

Step 1: Create an empty external archive $E^1 = \varnothing$ and set E_{max} as the maximum number of solutions it can accommodate.

Step 2: Evaluate and rank all the solutions in the current population *NP* by the nondominated sorting method. If $E_{max} \geq N$, all the solutions in population *NP* are added to the empty external archive and construct an initial external archive E^1; otherwise, the first E_{max} solutions in population *NP* are added to the empty external archive and form a full number of initial external archive E^1.

4.6.2. *Update of external archive*

For the subsequent generation, it was necessary to update and maintain E. Nondominated sorting and crowding distance [66] were used to maintain E. Each solution $\pi_i (i = 1, 2, \dots, N)$ in the current population *NP* was compared with each solution e_h ($h = 1, 2, \dots, E_{max}$) in current E to determine whether it could be added to E. To update E, we considered the following cases:

1. If π_i cannot dominate any e_h, π_i is not allowed to be added into E.
2. If π_i can dominate several e_h, π_i will be added to E and these solutions e_h dominated by π_i are discarded from E.
3. If the number of solutions in E is larger than E_{max}, the most crowded solutions are removed. This operation is repeated until the number of solutions in E is equal to E_{max}. Then, a new E is generated.

Algorithm 2: OBL for the EJSP-AGVs

```
 1: Select the best $r$ solutions from the current population $H$
 2: for $i$=1 to $r×N$ do
 3:     for $s$=1 to $LS$ do     % $LS$ is the length of $O$ or $v$
 4:         $O_i(s)= θ (O_{min}+O_{max})-O_i(s)$   % $O_{max}$ and $O_{min}$ are the upper and lower bounds of $O$
 5:         $v_i(s)= θ (v_{min}+v_{max})-v_i(s)$      % $v_{max}$ and $v_{min}$ are the upper and lower bounds of $v$
 6:     end for
 7:     for $s$=1 to $LA$ do   % $LA$ is the length of $A$ or $V$
 8:         $A_i(s)= θ (A_{min}+A_{max})-A_i(s)$    % $A_{max}$ and $A_{min}$ are the upper and lower bounds of $A$
 9:         $V_i(s)= θ (V_{min}+V_{max})-V_i(s)$    % $V_{max}$ and $V_{min}$ are the upper and lower bounds of $V$
10:     end for
11: Repair $O_i$, $v_i$, $A_i$ and $V_i$, respectively, and obtain a feasible solution
12: end for
13: Obtain a new population $R$ consisting of $r×N$ new solutions
14: Unite $R$ and $H$ to obtain a combined population $W$
15: Evaluate $W$ by nondominated sorting and select $N$ better individuals to obtain a new
    population $NP$
```

4.7. Framework of the EMOEA

Based on the above main processes, the overall framework of the EMOEA for the EJSP-AGVs can be summarized as follows. Fig. 9 shows the flowchart of the EMOEA.

Step 1: Initialize and encode the population P. Set relevant algorithm parameters and termination conditions. Set an empty initial external archive $E^1 = \varnothing$.

Step 2: Decode the individuals in population P. Calculate the three objective function values of each solution individual by Formulas (2)–(4). Evaluate and rank all current individuals by the nondominated sorting method.

Step 3: Randomly select two individuals from the current population P and conduct the three types of crossover operators with probability CR. Repeat the crossover operation $N/2$ times (N is the size of population P) and obtain a crossover population.

Step 4: Execute the mutation operator for each individual in the obtained crossover population with probability MR. A new population Q is obtained after the application of this mutation operator.

Step 5: Merge P and Q to obtain a combined population H. Evaluate and rank all individuals in H by nondominated sorting. Select the best $r \times N$ individuals from the evaluated H and then perform the OBL strategy on them to obtain a population R.

Step 6: Merge H and R to obtain a combined population W. Evaluate and rank all individuals in W and select N best individuals to obtain a new population NP.

Step 7: If it is the first generation, construct the initial external archive E^1; otherwise, perform the update the external archive E.

Step 8: If the termination condition (maximum iteration generation or maximum computation time) is met, output the solution individuals in an external archive; otherwise, return the new population NP to Step 2.

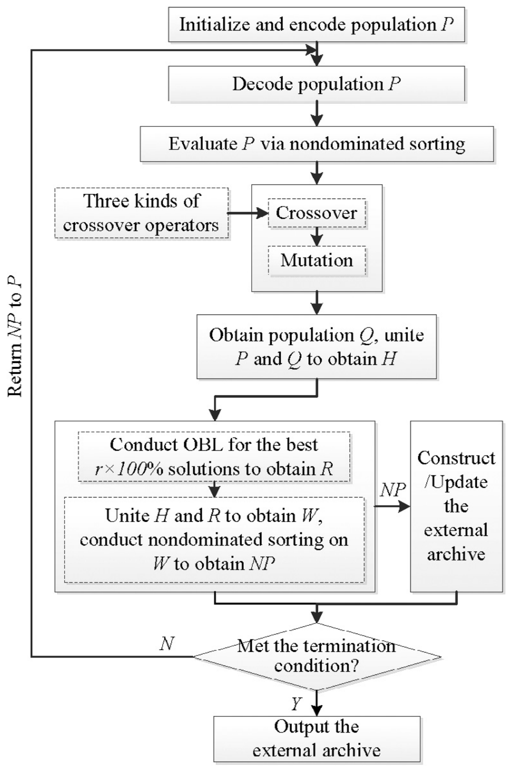

Initialize and encode population P
Decode population P
Evaluate P via nondominated sorting
Three kinds of crossover operators
Crossover
Mutation
Obtain population Q, unite P and Q to obtain H
Conduct OBL for the best r×100% solutions to obtain R
Unite H and R to obtain W, conduct nondominated sorting on W to obtain NP
NP
Construct /Update the external archive
Return NP to P
N
Met the termination condition?
Y
Output the external archive

Fig. 9. The framework of the EMOEA.

5. Experiments and result analysis

We first constructed 75 benchmark instances for the experiments in this study. In this section, the details of how these instances were constructed are first provided, following which we introduce the two performance metrics used in our study. Then, we discuss the six sets of experiments that were conducted to perform experimental evaluation. The first sets of experiments was used to obtain the best combination of key parameters for the EMOEA based on 6 selected instances with different scales. The second sets of experiments was conducted to validate our proposed model via 9 small-scale instances and 3 large-scale instances. The third sets of experiments was used to discuss the effect of AGV quantity configuration on production system performance based on all 75 instances. The fourth sets of experiments was conducted to discuss the effect of the OBL strategy based on 30 selected instances with different scales. The fifth sets of experiments was executed on all 75 instances to assess the performance of EMOEA in comparison with three other well-known algorithms. Finally, the sixth sets of experiments was carried out to test the proposed model and algorithm on a real-world case. All the relevant algorithms were coded in MATLAB R2014a and conducted on a computer with an Intel Core i7 CPU of 4.00 GHz and 16.0 GB RAM. All algorithms were independently repeated 30 times for each instance because of the stochastic

Table 1
The maximum number of AGVs for each type of instance.

$n{\times}m$	3×3	6×6	10×10	20×10	30×10
Maximum number of AGVs	10	10	10	20	25

nature of the algorithms. The mean of 30 runs was deemed the final result.

5.1. Benchmark generation

We randomly constructed five types of benchmark instances based on different scales of jobs and machines. We used $n{\times}m$ to represent the scale of each type of instance, where n is the number of jobs and m is the number of machines. In this work, the five types of instances included 3×3, 6×6, 10×10, 20×10 and 30×10. We configured each type of instance with a different number of AGVs. Table 1 shows the maximum number of AGV configurations for each type of instance. In total, there were 75 instances, including instances 01–10 based on 3×3, instances 11–20 based on 6×6, instances 21–30 based on 10×10, instances 31–50 based on 20×10 and instances 50–75 based on 30×10. We used $n \times m \times a$ to denote each instance, in which a is the number of AGVs.

For each instance, the basic processing time of each operation was generated from the uniform distribution $U[10, 100]$. The machine processing speed for each operation was set to five levels, $v = \{v_1, \dots, v_5\} = \{1, 1.25, 1.5, 1.75, 2.0\}$. The processing power consumption of machine k was obtained by $P_{kl} = \varphi_k \times v_l^2$, in which $l = 1, 2, \dots, 5$ and φ_k, i.e., the power consumption coefficient of machine k, was generated from the uniform distribution $U[5, 10]$. The stand-by power consumption for machine k was set as $PS_k = \varphi_k/4$. The SDST of each machine for processing two adjacent jobs was generated from the uniform distribution $U[10, 50]$. The setup power consumption for machine k was $PST_k = \varphi_k/2$. For each AGV, the transport speed was selected from $V = \{V_1, V_2, V_3\} = \{0.5, 0.75, 1\}$. The transport power consumption of AGV h was calculated as $PA_{hr} = \epsilon_h \times V_r^2$, where $r = 1, 2, 3$ and ϵ_h, i.e., the power consumption coefficient of AGV h was generated from the uniform distribution $U[3, 5]$. The stand-by power consumption for AGV h was set as $PAS_h = \epsilon_h/2$. The machines in the intelligent job shop were arranged in a 'U' shape, in a clockwise direction, as shown in Fig. 10, where the machines are regarded as particles. We assumed that AGVs would transport the jobs according to the shortest straight-line distance between two positions. All the relevant distances can be calculated by the layout in Fig. 10.

All benchmark instances constructed in this paper are available from https://github.com/hlj290612/EJSP-AGVs.

5.2. Performance metrics

To effectively evaluate the performance of different algorithms, we adopted the Hypervolume (HV) [1,53,55] and C-metric [1,55] as performance metrics.

1. HV. The HV is a comprehensive indicator that calculates the cumulative normalized volume covered by a solution set in comparison with a given reference point. A greater HV value represents a better convergence and diversity of a solution set. The HV is defined as follows:

$$HV\left(A, q\right) = volume(U_{X \in A}[f_1\left(X\right), q_1] \times \cdots \times [f_M\left(X\right), q_M]) \tag{29}$$

where A is an obtained solution set and $q = (q_1, q_2, \dots, q_M)$ is the HV reference point. M is the number of objectives.

2. C-metric. The C-metric is an indicator that reflects the dominance relationship between two solution sets: A and B. $C(A, B)$ represents the percentage of the solutions in B that are dominated by at last one in A. $C(A, B)$ is calculated as follows:

$$C\left(A, B\right) = \frac{|\{X_2 \in B | \exists X_1 \in A, X_1 \textit{ dominates } X_2\}|}{|B|} \tag{30}$$

where $C(A, B) = 0$ means that no solution in B is dominated by any solutions in A, while $C(A, B) = 1$ means that all solutions in B are dominated by solutions in A.

5.3. Parameter tuning

The proposed EMOEA had four key parameters: the population size N, the crossover rate CR, the mutation rate MR and the rate of population individuals executing OBL r. Taguchi analysis [67] was used to obtain the best combination of these parameters based on six selected instances, i.e., instances 12, 15, 22, 25, 52 and 55. We set each parameter to five levels, i.e., $N = [30, 60, 90, 120, 150]$, $CR = [0.1, 0.3, 0.5, 0.7, 0.9]$, $MR = [0.1, 0.3, 0.5, 0.7, 0.9]$ and $r = [0.2, 0.4, 0.6, 0.8, 1]$. Given that the HV is a compressive indicator, the mean values of the best HV results in ten independent runs were used to choose the parameter combination. Fig. 11 presents the trend of each factor level.

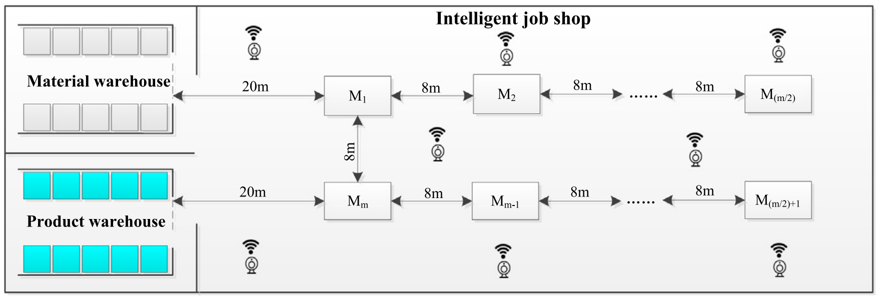
As shown in Fig. 11, the EMOEA obtained the best HV results when N was set to 60 or 90. Considering the computational time, we selected $N = 60$ for the EMOEA. For CR, we observed that the EMOEA achieved the best HV results on all instances except instances 22 and 25, when $CR = 0.8$. We therefore chose $CR = 0.8$ for the EMOEA. For MR, the EMOEA obtained the best HV results on all instances (except instance 55) when $MR = 0.5$. Therefore, we set MR to 0.5 for the EMOEA. For r, we can observe that when $r = 0.2$, the EMOEA obtained the best HV results on all selected instances. Therefore, we chose $r = 0.2$ for the EMOEA.

5.4. Validation of the proposed model

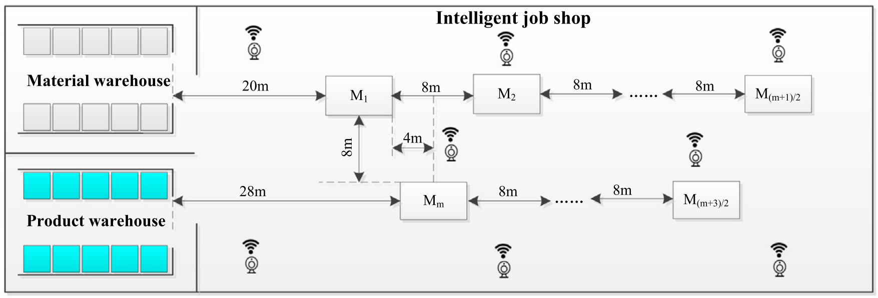
To validate the proposed model, we used the IBM ILOG CPLEX 12.6.3 solver to obtain the optimal (three) objective values on some selected instances. We did so using the method in [68]. We first optimized $f1$ to obtain an optimal solution. Then, we calculated the objective values for $f2$ and $f3$ based on the optimal solution for $f1$ from CPLEX using Eqs. (2)–(4). These three objective values were compared to those obtained by our proposed EMOEA.

Table 2 reports the solution results and computation time (CT) of CPLEX and EMOEA for these instances. We can observe from Table 2 that CPLEX solved instances 01, 03, 05, 11, 13, 15, 21, 23 and 25 within 1 h. However, when the problem size increased (e.g., instances 31, 33 and 35), CPLEX took much more time (more than 2 h) to obtain the optimal results. Our EMOEA was able to solve all of the selected instances. Furthermore, while CPLEX outperformed the EMOEA in terms of $f1$ on all 12 selected instances, the latter was able to outperform CPLEX in regard to $f2$ (except instance 03) and $f3$. It should be noted that the solutions of CPLEX and EMOEA did not dominate each other.

Additionally, we input the solutions obtained by CPLEX on six small-sized instances into the EMOEA's mathematical model and compared the three objective function values obtained by both the EMOEA and CPLEX. As shown in Table 3, the objective function values obtained by the EMOEA were exactly the same as those of CPLEX. Hence, the objective functions and constraints in our mathematical model were validated.

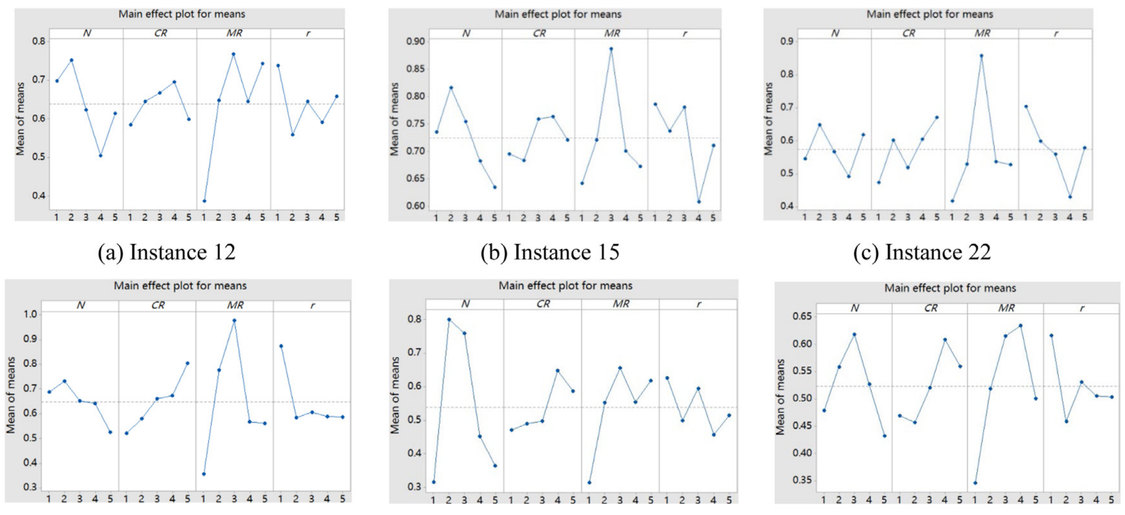


(a)The layout of an intelligent job shop when the number of machines is even



(b)The layout of an intelligent job shop when the number of machines is odd

Fig. 10. The layout of an intelligent job shop.



(a) Instance 12 (b) Instance 15 (c) Instance 22

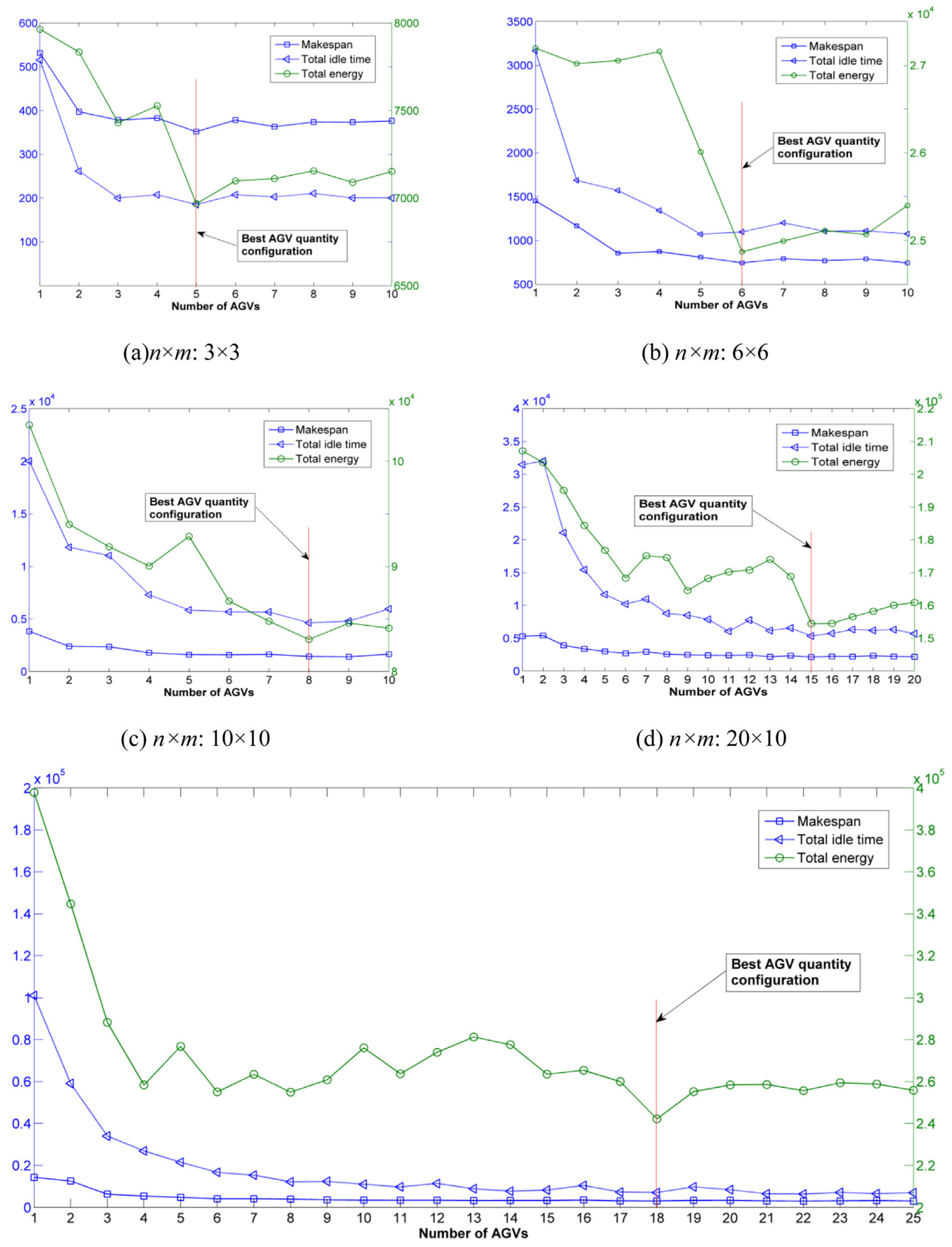
(d) Instance 25 (e) Instance 52 (f) Instance 55

Fig. 11. The trend of factor levels for key parameters.

Table 2
Solution results obtained by the CPLEX solver and EMOEA.

Inst.	Size ($n \times m \times a$)	CPLEX				EMOEA			
		$f1$	$f2$	$f3$	CT (s)	$f1$	$f2$	$f3$	CT (s)
01	3 × 3 × 1	**531**	515	7986	**43**	562	**506**	**7666**	110
03	3 × 3 × 3	**380**	**208**	7520	**40**	402	224	**7485**	124
05	3 × 3 × 5	**358**	198	7123	**52**	375	**185**	**6968**	118
11	6 × 6 × 1	**1453**	3137	27 182	**234**	1542	**3043**	**26 195**	440
13	6 × 6 × 3	**856**	1670	24 565	**257**	902	**1569**	**23 454**	398
15	6 × 6 × 5	**812**	1185	27 892	**235**	845	**1072**	**27 009**	462
21	10 × 10 × 1	**3838**	21 182	112 375	1705	4042	**20 023**	**103 480**	**1688**
23	10 × 10 × 3	**2362**	13 772	82 745	**1824**	2450	**12 034**	**81 884**	2015
25	10 × 10 × 5	**1610**	6012	94 775	2076	1722	**5851**	**92 869**	**1942**
31	20 × 10 × 1	**5290**	33 355	221 890	9565	5320	**31 461**	**207 129**	**3458**
33	20 × 10 × 3	**3872**	25 467	217 854	9762	3913	**21 088**	**195 086**	**3604**
35	20 × 10 × 5	**3003**	13 245	202 078	11 016	3096	**11 674**	**176 826**	**3887**

Note: best results are marked in bold for each instance.

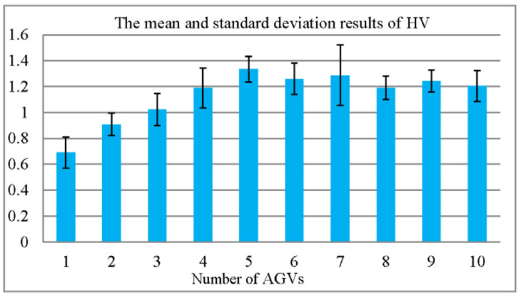


(a)n×m: 3×3 (b) n×m: 6×6 (c) n×m: 10×10 (d) n×m: 20×10 (e) n×m: 30×10

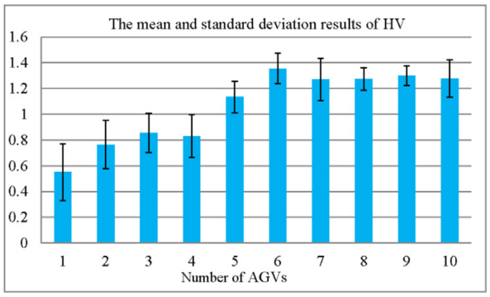
Fig. 12. Curves of the mean objective values on five types of instances with different numbers of AGVs.

Table 3
Instances for which the CPLEX solver and EMOEA generated the same solutions.

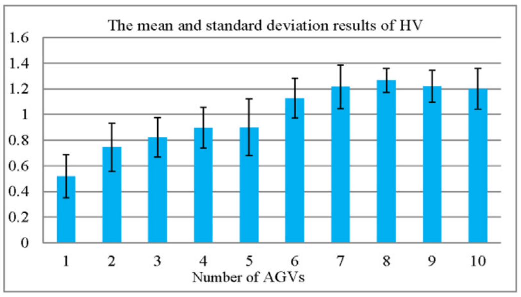
Inst.	Size ($n \times m \times a$)	CPLEX			EMOEA		
		$f1$	$f2$	$f3$	$f1$	$f2$	$f3$
01	3 × 3 × 1	531	515	7986	531	515	7986
03	3 × 3 × 3	380	208	7520	380	208	7520
05	3 × 3 × 5	358	198	7123	358	198	7123
11	6 × 6 × 1	1453	3137	27 182	1453	3137	27 182
13	6 × 6 × 3	856	1670	24 565	856	1670	24 565
15	6 × 6 × 5	812	1185	27 892	812	1185	27 892



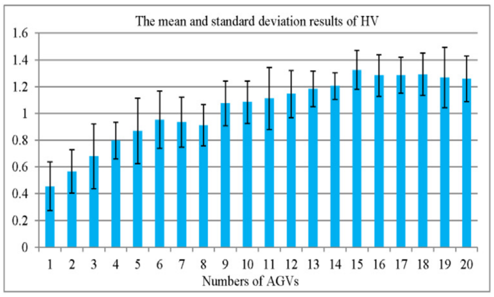
(a) $n\times m$: 3×3



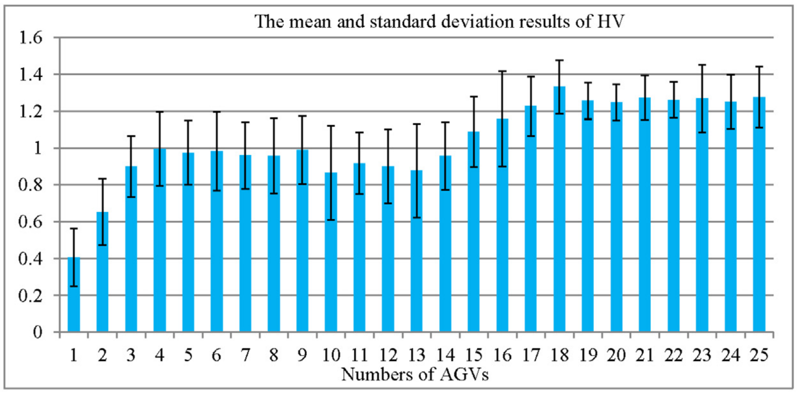
(b) $n\times m$: 6×6



(c) $n\times m$: 10×10



(d) $n\times m$: 20×10



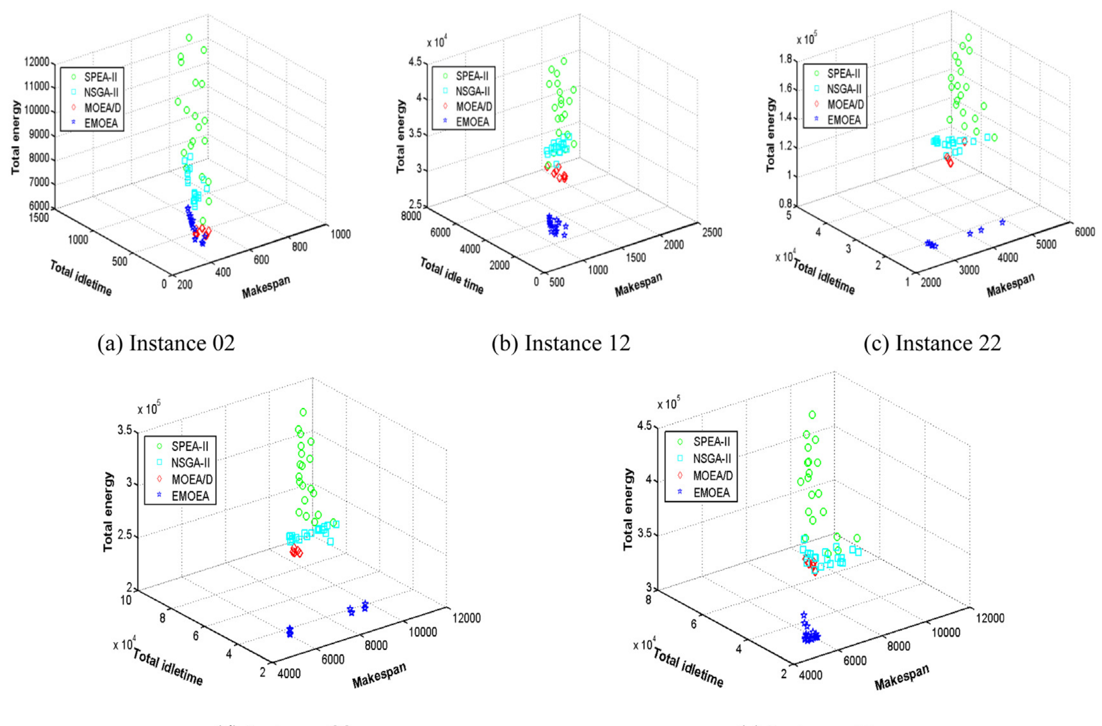
(e) $n\times m$: 30×10

Fig. 13. Mean and standard deviation results of HV on five types of instances with different numbers of AGVs.

5.5. *Effect of AGV quantity configuration on system performance*

To study how many AGVs should be configured to achieve the best performance of the production system, we used the EMOEA to solve each type of instance with different numbers of AGVs. A total of 30 independent runs were conducted for each instance. In each run, we calculated the mean values of the three objectives for the obtained solution set. Then, the mean of the mean values of the three objectives for the 30 independent runs for each instance was obtained. Additionally, the mean and standard deviation of the HV results were calculated for each type of instance with different numbers of AGVs. By analyzing the change in trends of the mean objective values and the means and standard deviations of HV, we wanted to determine how many AGVs can be configured to maximize the performance of the production system. Fig. 12 shows the curves of the mean objective values on five types of instances with different numbers of AGVs. In Fig. 12, the y-axis on the left represents the makespan and total idle time, and the y-axis on the right represents the total energy consumption. Fig. 13 depicts the mean and standard deviation results of HV on five types of instances with different numbers of AGVs.

Fig. 12(a)–(f) show that, for each type of instance, the makespan and total idle time generally decreased as the number of AGVs increased, and finally remained at a relatively stable value. However, there were some small fluctuations. With the increase in the number of AGVs, the total energy consumption was first reduced to an extreme point and then increased slightly. The reasons behind this result may be as follows. First, more AGVs can transport more jobs, which can reduce the waiting time of jobs and machines. Therefore, the makespan and total idle time are significantly reduced with an appropriate number of AGVs. Second, when the number of AGVs increases beyond a certain level, the redundant AGVs cannot transport more jobs, which means that the makespan and total idle time cannot be further reduced. Third, redundant AGVs may result in some idle waiting time for AGVs, which will consume a certain amount of energy. Fig. 12(a)–(f) reveal that, to achieve better performance, more AGVs need to be configured in the system as the size of instances increases. In particular, the production system can achieve better and more balanced performance when 5, 6, 8, 15 and 18 AGVs are configured for the corresponding five types of instances. Likewise, for each type of instance, we can observe from Fig. 13



(a) Instance 02 (b) Instance 12 (c) Instance 22

(d) Instance 32 (e) Instance 55

Fig. 14. Pareto fronts of the four algorithms on five selected instances.

that the mean HV results increased with the increase in the number of AGVs. Then, there was a slight decrease after the HV reached the maximum mean value. The results in Fig. 13 imply that properly increasing the number of AGVs can improve the performance of the production system, but after the number of AGVs increases beyond a certain extent, the system performance is not significantly improved. As expected, the conclusion drawn from Fig. 13 was consistent with that from Fig. 12.

5.6. Effect of the OBL strategy

To investigate the effect of the OBL strategy, we conducted a performance comparison between the proposed EMOEA with and without OBL. We denoted the proposed EMOEA without OBL as EMOEA^{no}. The only difference between the EMOEA and EMOEA^{no} is the OBL strategy. Due to space constraints, for each type of instance, six subinstances with a different number of AGVs were selected to conduct this comparison experiment. For the EMOEA and EMOEA^{no}, the average of 30 independent runs on each instance was used as the final result. The statistical metric results of HV and C-metric for the EMOEA and EMOEA^{no} can be found in Table 4. Due to the stochastic nature of these MOEAs, statistical tests should be carried out to ascertain the significance of the results. To this end, a nonparameter statistical significance test method, i.e., the Wilcoxon sign rank test at a significance level of 0.05 was used to analyze the statistical significance of the results obtained by the EMOEA and EMOEA^{no}. The test results are shown as "+", "−", or "=" in Table 4 to denote when the EMOEA is significantly better than, significantly worse than, or statistically equivalent to EMOEA^{no}, respectively.

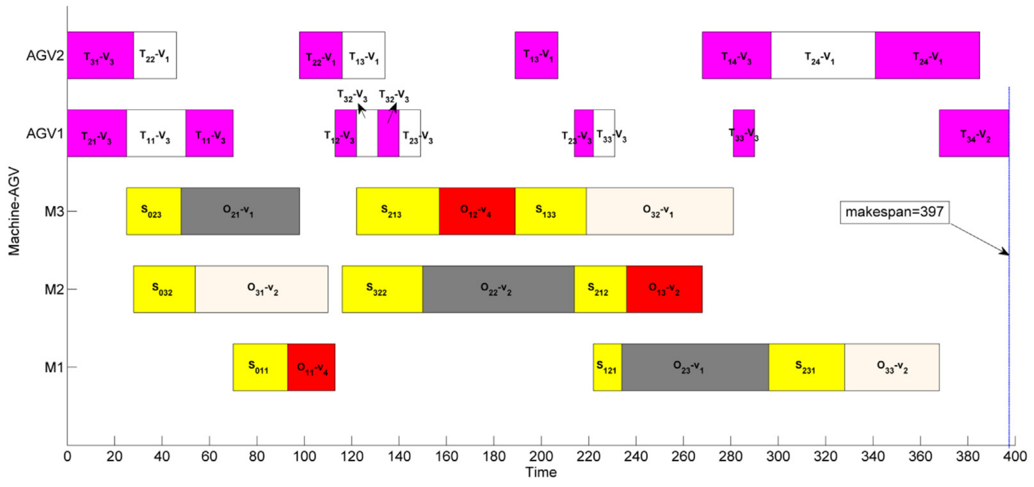
In terms of HV, we can observe from Table 4 that the EMOEA performed significantly better than EMOEA^{no} in 24 instances, worse than EMOEA^{no} in 2 instances, and equivalent to EMOEA^{no} in 2 instances. For the C-metric, we can see that the EMOEA performed better than EMOEA^{no} on all 28 instances. The results in Table 4 clearly demonstrate that the OBL strategy was very useful in improving the convergence and diversity of the proposed EMOEA.

5.7. Comparison with existing algorithms

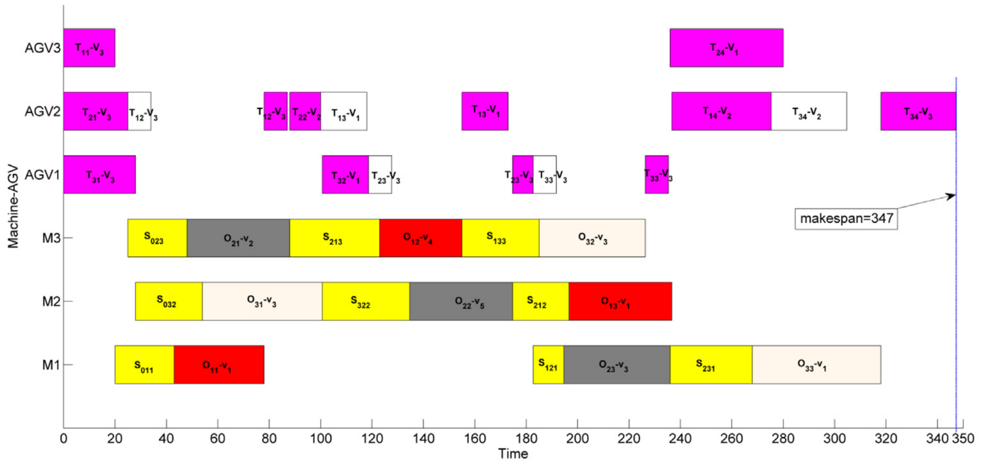
To evaluate the performance of the EMOEA, we compared it with three well-known algorithms, namely, the SPEA-II [69], NSGA-II [66] and MOEA/D [70]. The reasons for selecting these three algorithms for comparisons were as follows. First, all four algorithms have a similar structure, since they are based on a GA framework. Second, the SPEA-II, NSGA-II and MOEA/D are popular algorithms that are often used as baseline algorithms to verify the performance of new algorithms. For each algorithm, the average of 30 independent runs on each instance was used as the final result.

The SPEA-II, NSGA-II and MOEA/D utilized the simulated binary crossover and polynomial mutation, and the population size of MOEA/D was dependent on the number of reference points. In our experiments, the Tchebycheff approach [70] was selected as the scalarizing function for the MOEA/D. The parameters of the three algorithms were calibrated using the method described in Section 5.3. Due to space constraints, we report only the parameter values of these three algorithms in Table 5. The same encoding, decoding, computational time limit and HV reference point that were applied to the EMOEA were applied to all three algorithms. Furthermore, the OBL strategy and the external archive were integrated in all four algorithms to output the final solution set that was used for computing the performance measures.

Tables 6 and 7 show the mean and standard deviation values for the HV and C-metric obtained by the four algorithms. Wilcoxon sign rank tests at a significance level of 0.05 were used to analyze the statistical significance of the results obtained by these algorithms. In Table 6, we observe that the EMOEA performed significantly better than the other three algorithms on all 75 instances in terms of HV. In Table 7, with respect to the C-metric, we see that the EMOEA was significantly better than the SPEA-II, NSGA-II and MOEA/D. On most large-sized instances, the C-metric values for the EMOEA were equal to 1, which means that the EMOEA had at least one solution that dominated all of the



(a) Instance 02



(b) Instance 03

Fig. 15. Gantt charts obtained by the EMOEA on two selected instances.

solutions obtained by the other three algorithms (SPEA-II, NSGA-II and MOEA/D).

To visualize the Pareto fronts of the four algorithms, we chose five instances under different scales (i.e., instances 02, 12, 22, 32 and 55) from the tested instances. Fig. 14 presents the Pareto fronts of the four algorithms, in which we observe that the solutions obtained by the EMOEA were closer to the coordinate origin. This means that the EMOEA had better convergence performance and that the solution set of the EMOEA had a higher probability of dominating those of SPEA-II, NSGA-II and MOEA/D.

Fig. 15 presents two Gantt charts obtained by the EMOEA on two selected instances (i.e., instances 02 and 03) with the same number of jobs and the same number of machines. We can observe that the makespan of instance 03 was shorter than that of instance 02—implying that properly increasing the number of AGVs can shorten the waiting time of machines and improve the production efficiency.

5.8. A real-world case study

5.8.1. Case introduction

In this final series of experiments, a real-world intelligent manufacturing workshop integrated with AGVs for producing the components of security monitoring robots at a Chinese company was used to test the proposed mathematical model and algorithm. The layout of the intelligent manufacturing workshop is shown in Fig. 16. The functional areas are divided into five parts, namely, the central control room, manufacturing area, AGV transportation area, AGV charging area and automatic warehouse area. MW and PW are in the automatic warehouse. The dotted line in Fig. 16 represents the AGV transport track. The area of the intelligent manufacturing workshop is 33*14 (m^2). In this workshop, there are 6 manufacturing cells (MCs), each of which contains a CNC machining machine and several setup devices, such as robot arms for loading and unloading jobs. Several AGVs are used to transport jobs in this workshop. Each CNC machine has five processing speeds, and each AGV has three kinds of transportation speeds. Some important parts of the security monitoring robots, such as their shell, chassis and wheel, can be processed in this intelligent manufacturing workshop.

Fig. 17 shows a real-world security monitoring robot and its wheels. In a certain period, five robot wheels with different sizes need to be processed. Each robot wheel has 6 operations, i.e., (1) cutting $\rightarrow$ (2) turning $\rightarrow$ (3) coarse grinding $\rightarrow$ (4) drilling $\rightarrow$ (5) tapping $\rightarrow$ (6) fine grinding. Due to the differences in size and weight, the processing time for the operations of different wheels is different. There is SDST when a machine processes two different robot wheels. Two AGVs are used to transport the raw materials of the robot wheels. The data related to this real-world case can be found at https://github.com/hlj290612/EJSP-AGVs.

Table 4
Performance comparison between the EMOEA and EMOEA^{no}.

Inst.	Size ($n \times m \times a$)	HV		C-metric	
		EMOEA	EMOEA^{no}	EMOEA	EMOEA^{no}
02	3 × 3 × 2	**1.1293 (0.2077)**	0.9174 (0.2557) +	0.4780 (0.0949)	0.1300 (0.0567) +
03	3 × 3 × 3	1.1109 (0.1618)	**1.1214 (0.2612)** −	0.3050 (0.0657)	**0.3250 (0.0140)** −
04	3 × 3 × 4	**1.1927 (0.0666)**	0.9977 (0.0844) +	0.5050 (0.0358)	0.0886 (0.0400) +
05	3 × 3 × 5	1.1190 (0.1215)	**1.1401 (0.1465)** −	0.1254 (0.0486)	**0.2200 (0.0632)** −
06	3 × 3 × 6	**1.2003 (0.0468)**	1.1960 (0.1233) =	**0.3545 (0.0680)**	0.1125 (0.0316) +
08	3 × 3 × 8	**1.1818 (0.0684)**	1.1844 (0.1247) =	**0.2234 (0.0532)**	0.1333 (0.0675) +
12	6 × 6 × 2	**1.1582 (0.1065)**	0.7572 (0.1530) +	**0.5950 (0.0411)**	0.1228 (0.0315) +
13	6 × 6 × 3	**1.2229 (0.2346)**	0.9873 (0.1984) +	**0.6982 (0.0822)**	0.0835 (0.0264) +
14	6 × 6 × 4	**1.3013 (0.1988)**	1.0236 (0.2225) +	**0.7230 (0.0775)**	0.1456 (0.0660) +
15	6 × 6 × 5	**1.3128 (0.0985)**	1.1075 (0.1452) +	**0.7658 (0.0840)**	0.1675 (0.0802) +
16	6 × 6 × 6	**1.2007 (0.1234)**	0.9877 (0.0980) +	**0.6873 (0.0805)**	0.0879 (0.0633) +
18	6 × 6 × 8	**1.1973 (0.1560)**	0.9552 (0.0766) +	**0.7055 (0.0772)**	0.1238 (0.0555) +
22	10 × 10 × 2	**1.3337 (0.0850)**	1.1286 (0.0965) +	**0.5776 (0.6262)**	0.1007 (0.0478) +
23	10 × 10 × 3	**1.2095 (0.0777)**	1.2100 (0.0734) =	**0.3090 (0.0578)**	0.2796 (0.0600) +
24	10 × 10 × 4	**1.2456 (0.0986)**	1.1875 (0.0589) +	**0.4899 (0.0457)**	0.1005 (0.0384) +
25	10 × 10 × 5	**1.1987 (0.0685)**	1.1942 (0.0600) =	**0.3630 (0.0450)**	0.2264 (0.0382) +
26	10 × 10 × 6	**1.2021 (0.0778)**	1.1456 (0.0686) +	**0.5678 (0.0560)**	0.0923 (0.0278) +
28	10 × 10 × 8	**1.3168 (0.2437)**	1.0133 (0.1686) +	**0.7754 (0.0820)**	0.2000 (0.1204) +
32	20 × 10 × 2	**1.2871 (0.1345)**	0.9890 (0.2003) +	**0.7658 (0.0923)**	0.0939 (0.0677) +
35	20 × 10 × 5	**1.3332 (0.0768)**	1.1248 (0.1034) +	**0.6895 (0.0720)**	0.1156 (0.0823) +
36	20 × 10 × 6	**1.2896 (0.1132)**	1.0989 (0.1236) +	**0.7347 (0.0688)**	0.0899 (0.0674) +
38	20 × 10 × 8	**1.3478 (0.1364)**	1.1057 (0.0987) +	**0.7256 (0.0934)**	0.1237 (0.0845) +
42	20 × 10 × 12	**1.2780 (0.1562)**	1.1345 (0.1256) +	**0.6917 (0.0856)**	0.1073 (0.0766) +
45	20 × 10 × 15	**1.3567 (0.0978)**	1.0785 (0.1116) +	**0.7451 (0.0923)**	0.1366 (0.0688) +
53	30 × 10 × 3	**1.2980 (0.0895)**	1.1037 (0.0989) +	**0.7377 (0.1010)**	0.0944 (0.0457) +
55	30 × 10 × 5	**1.3620 (0.1234)**	1.1567 (0.1569) +	**0.7878 (0.0976)**	0.0867 (0.0440) +
58	30 × 10 × 8	**1.2477 (0.1117)**	1.1010 (0.0987) +	**0.7670 (0.0854)**	0.0888 (0.0388) +
60	30 × 10 × 10	**1.3619 (0.1566)**	1.1198 (0.1045) +	**0.8700 (0.0678)**	0.1023 (0.0582) +
62	30 × 10 × 12	**1.3684 (0.1405)**	1.0987 (0.1720) +	**0.8234 (0.0701)**	0.0912 (0.0455) +
65	30 × 10 × 15	**1.3572 (0.1380)**	1.1672 (0.1556) +	**0.7893 (0.0688)**	0.0845 (0.0387) +

Note: best results are marked in bold for each instance.

Table 5
Parameter values for the three competing algorithms.

Algorithm	Parameter value
SPEA-II	$N = 60$, $CR = 0.8$, $MR = 0.3$, $\eta_c = 20$, $\eta_m = 20$
NSGA-II	$N = 50$, $CR = 0.9$, $MR = 0.2$, $\eta_c = 20$, $\eta_m = 10$
MOEA/D	$N = 91$, $CR = 0.7$, $MR = 0.3$, $\eta_c = 20$, $\eta_m = 20$, $T = 15$

η_c: distribution index in simulated binary crossover; η_m: distribution index in polynomial mutation; T: neighborhood size.

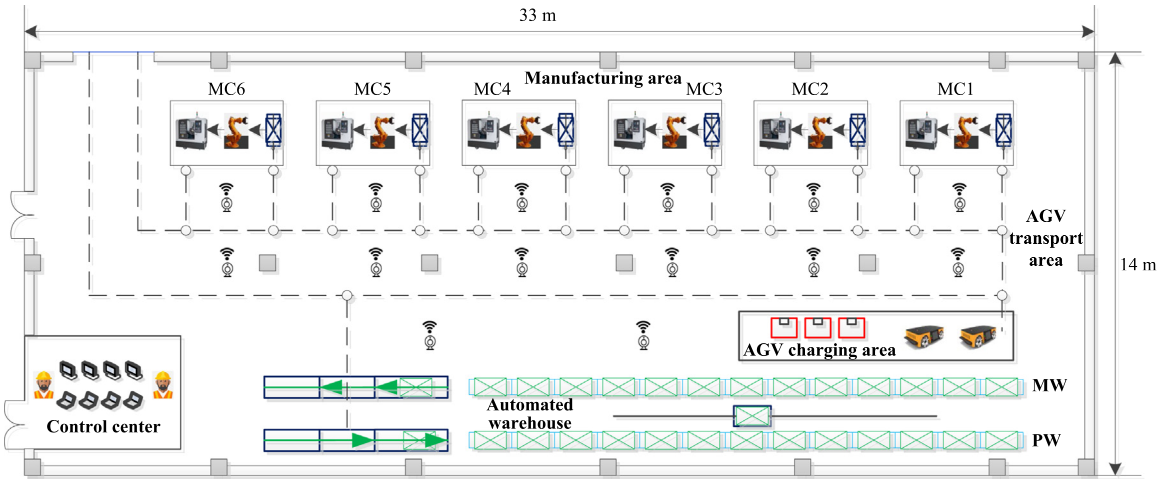


Fig. 16. The layout of an intelligent manufacturing workshop integrated with AGVs.

5.8.2. Result discussion

We used the SPEA-II, NSGA-II, MOEA/D and EMOEA to solve this real-world case. Tables 8 and 9 present the mean and standard deviation values of HV and the C-metric for the four algorithms on this real-world case. Wilcoxon sign rank tests at a significance level of 0.05 were used to analyze the statistical significance of the results obtained by these algorithms. We can observe from the tables that the EMOEA performed significantly better than the SPEA-II, NSGA-II and MOEA/D. From Fig. 18, it can be seen that the solution set of the EMOEA was able to dominate those of SPEA-II, NSGA-II and MOEA/D.

Table 6
Mean and standard deviation values of HV with the SPEA-II, NSGA-II, MOEA/D and EMOEA.

Inst.	Size ($n \times m \times a$)	HV			
		SPEA-II	NSGA-II	MOEA/D	EMOEA
		Mean (std)	Mean (std)	Mean (std)	Mean (std)
01	3 × 3 × 1	0.6814 (0.2020) +	0.7713 (0.1987) +	1.0855 (0.1579) +	**1.6831 (0.0298)**
02	3 × 3 × 2	1.1764 (0.0782) +	1.0757 (0.0934) +	1.4296 (0.0654) +	**1.6452 (0.0370)**
03	3 × 3 × 3	1.1548 (0.1122) +	1.0429 (0.0879) +	1.2357 (0.0920) +	**1.6814 (0.0300)**
04	3 × 3 × 4	1.2978 (0.1904) +	1.2681 (0.2021) +	1.2151 (0.1897) +	**1.6525 (0.0425)**
05	3 × 3 × 5	1.0762 (0.2345) +	1.1922 (0.1872) +	1.2368 (0.2675) +	**1.4177 (0.1236)**
06	3 × 3 × 6	1.2560 (0.2252) +	1.3148 (0.1655) +	1.4238 (0.1600) +	**1.6410 (0.0660)**
07	3 × 3 × 7	1.3328 (0.1277) +	1.3770 (0.1808) +	1.3815 (0.2004) +	**1.6573 (0.0484)**
08	3 × 3 × 8	1.4044 (0.1435) +	1.3975 (0.1560) +	1.4124 (0.1878) +	**1.6086 (0.0782)**
09	3 × 3 × 9	1.2821 (0.1876) +	1.3325 (0.1764) +	1.2442 (0.1872) +	**1.4592 (0.1635)**
10	3 × 3 × 10	1.1953 (0.2022) +	1.2023 (0.1846) +	1.2217 (0.2350) +	**1.6250 (0.0765)**
11	6 × 6 × 1	0.5148 (0.2567) +	0.4616 (0.3027) +	0.4468 (0.2650) +	**1.6844 (0.0307)**
12	6 × 6 × 2	0.5681 (0.0920) +	0.5426 (0.1034) +	0.6699 (0.0877) +	**1.6200 (0.0771)**
13	6 × 6 × 3	0.7475 (0.3023) +	0.7124 (0.2855) +	0.8796 (0.1843) +	**1.6684 (0.0562)**
14	6 × 6 × 4	0.6697 (0.3425) +	0.7984 (0.2873) +	0.8994 (0.2034) +	**1.6806 (0.0387)**
15	6 × 6 × 5	0.6790 (0.2568) +	0.6100 (0.3333) +	0.9073 (0.2452) +	**1.6552 (0.0472)**
16	6 × 6 × 6	0.9467 (0.1935) +	0.9186 (0.2456) +	0.8512 (0.3012) +	**1.6235 (0.0733)**
17	6 × 6 × 7	0.9173 (0.2436) +	0.8368 (0.2785) +	1.0194 (0.2020) +	**1.6811 (0.0320)**
18	6 × 6 × 8	0.9840 (0.2612) +	1.0233 (0.1763) +	1.1058 (0.1580) +	**1.6662 (0.0455)**
19	6 × 6 × 9	0.9519 (0.2318) +	0.8065 (0.2276) +	1.0961 (0.1670) +	**1.6374 (0.0623)**
20	6 × 6 × 10	0.9793 (0.1923) +	0.8842 (0.2450) +	1.2664 (0.1781) +	**1.6646 (0.0500)**
21	10 × 10 × 1	0.2766 (0.1555) +	0.3585 (0.1880) +	0.2480 (0.1345) +	**1.7000 (0.0202)**
22	10 × 10 × 2	0.5088 (0.1345) +	0.4897 (0.2002) +	0.5966 (0.2430) +	**1.6553 (0.0844)**
23	10 × 10 × 3	0.5640 (0.2020) +	0.6623 (0.1654) +	0.5095 (0.2114) +	**1.6647 (0.0357)**
24	10 × 10 × 4	0.6721 (0.1890) +	0.7345 (0.2300) +	0.7118 (0.1964) +	**1.6271 (0.0723)**
25	10 × 10 × 5	0.6884 (0.2345) +	0.9532 (0.1893) +	0.7930 (0.2416) +	**1.6845 (0.0377)**
26	10 × 10 × 6	0.8105 (0.2404) +	0.6497 (0.2020) +	0.7511 (0.3030) +	**1.6211 (0.0416)**
27	10 × 10 × 7	0.7560 (0.2121) +	0.6767 (0.1734) +	0.7272 (0.2842) +	**1.6472 (0.0626)**
28	10 × 10 × 8	0.6945 (0.3456) +	0.6820 (0.2034) +	0.7878 (0.1985) +	**1.6565 (0.0512)**
29	10 × 10 × 9	0.7546 (0.2567) +	0.6735 (0.1456) +	0.8856 (0.2455) +	**1.6630 (0.0481)**
30	10 × 10 × 10	0.8130 (0.2045) +	0.8278 (0.1987) +	0.7266 (0.1883) +	**1.6845 (0.0233)**
31	20 × 10 × 1	0.3619 (0.2278) +	0.4856 (0.1780) +	0.3759 (0.2455) +	**1.7000 (0.0155)**
32	20 × 10 × 2	0.3563 (0.2500) +	0.5074 (0.1578) +	0.3188 (0.2876) +	**1.6480 (0.0622)**
33	20 × 10 × 3	0.8294 (0.3205) +	0.8192 (0.2260) +	0.6785 (0.3651) +	**1.6774 (0.0405)**
34	20 × 10 × 4	0.6445 (0.2670) +	0.6170 (0.3345) +	0.5008 (0.2711) +	**1.6822 (0.0277)**
35	20 × 10 × 5	0.5267 (0.3033) +	0.6338 (0.2761) +	0.4293 (0.3730) +	**1.6511 (0.0333)**
36	20 × 10 × 6	0.5836 (0.2876) +	0.6124 (0.2520) +	0.5864 (0.3025) +	**1.6464 (0.0555)**
37	20 × 10 × 7	0.6730 (0.2871) +	0.6872 (0.1911) +	0.5587 (0.2238) +	**1.6078 (0.0832)**
38	20 × 10 × 8	0.5641 (0.2235) +	0.6552 (0.2048) +	0.5433 (0.3124) +	**1.6712 (0.0357)**
39	20 × 10 × 9	0.6875 (0.2416) +	0.7777 (0.2519) +	0.7218 (0.2760) +	**1.6231 (0.0578)**
40	20 × 10 × 10	0.6043 (0.2546) +	0.7245 (0.1897) +	0.6980 (0.3125) +	**1.6574 (0.0613)**
41	20 × 10 × 11	0.6672 (0.2603) +	0.7904 (0.2134) +	0.6502 (0.2561) +	**1.6316 (0.0537)**
42	20 × 10 × 12	0.7236 (0.2560) +	0.7022 (0.2367) +	0.6893 (0.3232) +	**1.6467 (0.0623)**
43	20 × 10 × 13	0.6794 (0.2325) +	0.6880 (0.1907) +	0.6023 (0.2768) +	**1.6565 (0.0443)**
44	20 × 10 × 14	0.7231 (0.2111) +	0.7015 (0.2236) +	0.6820 (0.3214) +	**1.6613 (0.0382)**
45	20 × 10 × 15	0.6554 (0.1893) +	0.6907 (0.1900) +	0.5823 (0.2630) +	**1.6432 (0.0457)**
46	20 × 10 × 16	0.5974 (0.2675) +	0.7680 (0.2572) +	0.6682 (0.3032) +	**1.6017 (0.0725)**
47	20 × 10 × 17	0.7688 (0.2013) +	0.6896 (0.3030) +	0.7231 (0.2317) +	**1.6345 (0.0613)**
48	20 × 10 × 18	0.6567 (0.1983) +	0.7562 (0.2784) +	0.5785 (0.3126) +	**1.6234 (0.0576)**
49	20 × 10 × 19	0.7643 (0.2134) +	0.6785 (0.1985) +	0.6876 (0.2025) +	**1.6437 (0.0673)**
50	20 × 10 × 20	0.6675 (0.1982) +	0.6680 (0.2345) +	0.5893 (0.1658) +	**1.6195 (0.0344)**
51	30 × 10 × 1	0.3856 (0.1030) +	0.3987 (0.0935) +	0.2905 (0.1004) +	**1.6726 (0.0370)**
52	30 × 10 × 2	0.5034 (0.2034) +	0.5567 (0.1456) +	0.3780 (0.1605) +	**1.6308 (0.0465)**
53	30 × 10 × 3	0.5560 (0.2564) +	0.6054 (0.2552) +	0.4832 (0.2030) +	**1.6540 (0.0400)**
54	30 × 10 × 4	0.5663 (0.2315) +	0.4975 (0.2316) +	0.4981 (0.1974) +	**1.6433 (0.0565)**
55	30 × 10 × 5	0.5785 (0.1980) +	0.6767 (0.3210) +	0.7822 (0.2375) +	**1.6380 (0.0487)**
56	30 × 10 × 6	0.4906 (0.2450) +	0.5785 (0.1985) +	0.6637 (0.2315) +	**1.6565 (0.0642)**
57	30 × 10 × 7	0.6065 (0.1945) +	0.5983 (0.2023) +	0.4748 (0.2122) +	**1.6423 (0.0333)**
58	30 × 10 × 8	0.5979 (0.1873) +	0.6545 (0.1783) +	0.7023 (0.1893) +	**1.5968 (0.0456)**
59	30 × 10 × 9	0.6767 (0.1920) +	0.7056 (0.2121) +	0.5985 (0.1356) +	**1.6036 (0.0892)**
60	30 × 10 × 10	0.5556 (0.1563) +	0.6784 (0.2035) +	0.5786 (0.2225) +	**1.6162 (0.0685)**
61	30 × 10 × 11	0.5875 (0.2013) +	0.6845 (0.1874) +	0.4476 (0.1893) +	**1.6234 (0.0584)**
62	30 × 10 × 12	0.6544 (0.1834) +	0.6756 (0.1935) +	0.8236 (0.2025) +	**1.6346 (0.0678)**
63	30 × 10 × 13	0.6756 (0.2014) +	0.5879 (0.1734) +	0.6798 (0.2365) +	**1.5897 (0.0825)**
64	30 × 10 × 14	0.5878 (0.1770) +	0.6568 (0.2121) +	0.5989 (0.2455) +	**1.5943 (0.0793)**
65	30 × 10 × 15	0.6565 (0.1897) +	0.5560 (0.2020) +	0.4894 (0.1972) +	**1.6215 (0.0812)**
66	30 × 10 × 16	0.6787 (0.2345) +	0.7834 (0.1876) +	0.5674 (0.2457) +	**1.6543 (0.0480)**
67	30 × 10 × 17	0.6463 (0.1794) +	0.6907 (0.2561) +	0.7829 (0.1926) +	**1.6049 (0.0782)**
68	30 × 10 × 18	0.5975 (0.2781) +	0.6819 (0.1189) +	0.6992 (0.3035) +	**1.6237 (0.0781)**
69	30 × 10 × 19	0.6982 (0.1987) +	0.7793 (0.1846) +	0.7250 (0.2634) +	**1.5895 (0.1345)**
70	30 × 10 × 20	0.7892 (0.2246) +	0.6892 (0.1937) +	0.6893 (0.2934) +	**1.6238 (0.0785)**
71	30 × 10 × 21	0.7504 (0.3402) +	0.7763 (0.2562) +	0.6987 (0.2825) +	**1.6311 (0.0872)**

(*continued on next page*)

Table 6 (*continued*).

Inst.	Size ($n \times m \times a$)	HV			
		SPEA-II Mean (std)	NSGA-II Mean (std)	MOEA/D Mean (std)	EMOEA Mean (std)
72	30 × 10 × 22	0.6769 (0.2655) +	0.7500 (0.2432) +	0.7076 (0.1895) +	**1.6464 (0.0716)**
73	30 × 10 × 23	0.6980 (0.3132) +	0.7872 (0.2675) +	0.6879 (0.2435) +	**1.5912 (0.0820)**
74	30 × 10 × 24	0.5976 (0.2020) +	0.8345 (0.1936) +	0.7456 (0.3451) +	**1.6065 (0.0725)**
75	30 × 10 × 25	0.6583 (0.1745) +	0.7256 (0.2567) +	0.6868 (0.2120) +	**1.5864 (0.0567)**

Note: best results are marked in bold for each instance.

Table 7
Mean and standard deviation values of C-metric with the SPEA-II, NSGA-II, MOEA/D and EMOEA.

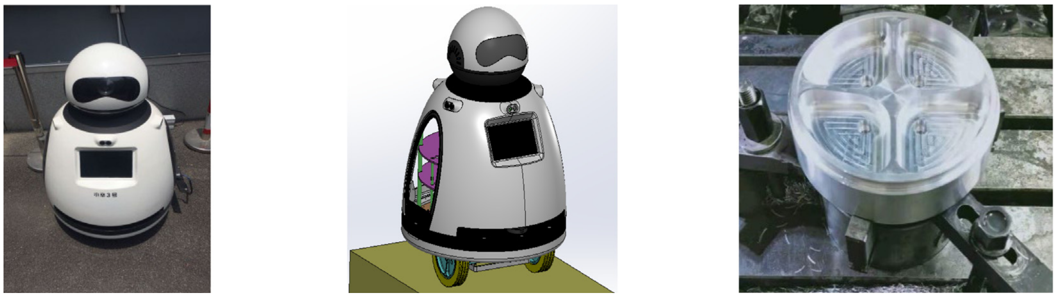
Inst.	Size ($n \times m \times a$)	C-metric					
		SPEA-II (S) v. EMOEA (E)		NSGA-II (N) v. EMOEA (E)		MOEA/D (M) v. EMOEA (E)	
		C(S, E) Mean (std)	C(E, S) Mean (std)	C(N, E) Mean (std)	C(E, N) Mean (std)	C(M, E) Mean (std)	C(E, M) Mean (std)
01	3 × 3 × 1	0.0000 (0.0000) +	**0.9565 (0.0370)**	0.0000 (0.0000) +	**0.7688 (0.1212)**	0.0982 (0.0234) +	**0.5896 (0.1290)**
02	3 × 3 × 2	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**0.8236 (0.1544)**	0.0888 (0.0456) +	**0.7344 (0.1606)**
03	3 × 3 × 3	0.0000 (0.0000) +	**0.9344 (0.0355)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**0.8760 (0.1212)**
04	3 × 3 × 4	0.0000 (0.0000) +	**0.9022 (0.0425)**	0.0000 (0.0000) +	**0.9467 (0.0502)**	0.1326 (0.0298) +	**0.7878 (0.1400)**
05	3 × 3 × 5	0.0000 (0.0000) +	**0.7898 (0.1034)**	0.0000 (0.0000) +	**0.8345 (0.0987)**	0.0000 (0.0000) +	**0.8008 (0.1683)**
06	3 × 3 × 6	0.0000 (0.0000) +	**0.9480 (0.0467)**	0.0000 (0.0000) +	**0.8236 (0.1008)**	0.1004 (0.0382) +	**0.7553 (0.1456)**
07	3 × 3 × 7	0.0000 (0.0000) +	**0.8085 (0.1323)**	0.0988 (0.0123) +	**0.6834 (0.1978)**	0.1230 (0.0456) +	**0.6540 (0.1532)**
08	3 × 3 × 8	0.0000 (0.0000) +	**0.8652 (0.1034)**	0.0000 (0.0000) +	**0.8056 (0.1378)**	0.1360 (0.1011) +	**0.6825 (0.1755)**
09	3 × 3 × 9	0.0000 (0.0000) +	**0.8122 (0.1234)**	0.0867 (0.0234) +	**0.6012 (0.2222)**	0.0900 (0.0183) +	**0.6578 (0.1856)**
10	3 × 3 × 10	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.1180 (0.0346) +	**0.7875 (0.1647)**
11	6 × 6 × 1	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
12	6 × 6 × 2	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
13	6 × 6 × 3	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
14	6 × 6 × 4	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
15	6 × 6 × 5	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
16	6 × 6 × 6	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
17	6 × 6 × 7	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
18	6 × 6 × 8	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
19	6 × 6 × 9	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
20	6 × 6 × 10	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
21	10 × 10 × 1	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
22	10 × 10 × 2	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
23	10 × 10 × 3	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
24	10 × 10 × 4	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
25	10 × 10 × 5	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
26	10 × 10 × 6	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
27	10 × 10 × 7	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
28	10 × 10 × 8	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
29	10 × 10 × 9	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
30	10 × 10 × 10	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
31	20 × 10 × 1	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
32	20 × 10 × 2	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
33	20 × 10 × 3	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
34	20 × 10 × 4	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
35	20 × 10 × 5	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
36	20 × 10 × 6	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
37	20 × 10 × 7	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
38	20 × 10 × 8	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
39	20 × 10 × 9	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
40	20 × 10 × 10	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
41	20 × 10 × 11	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
42	20 × 10 × 12	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
43	20 × 10 × 13	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
44	20 × 10 × 14	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
45	20 × 10 × 15	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
46	20 × 10 × 16	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
47	20 × 10 × 17	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
48	20 × 10 × 18	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
49	20 × 10 × 19	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
50	20 × 10 × 20	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
51	30 × 10 × 1	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
52	30 × 10 × 2	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
53	30 × 10 × 3	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
54	30 × 10 × 4	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
55	30 × 10 × 5	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
56	30 × 10 × 6	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
57	30 × 10 × 7	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
58	30 × 10 × 8	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**

(*continued on next page*)

Table 7 (*continued*).

Inst.	Size ($n \times m \times a$)	C-metric					
		SPEA-II (S) v. EMOEA (E)		NSGA-II (N) v. EMOEA (E)		MOEA/D (M) v. EMOEA (E)	
		C(S, E) Mean (std)	C(E, S) Mean (std)	C(N, E) Mean (std)	C(E, N) Mean (std)	C(M, E) Mean (std)	C(E, M) Mean (std)
59	30 × 10 × 9	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
60	30 × 10 × 10	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
61	30 × 10 × 11	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
62	30 × 10 × 12	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
63	30 × 10 × 13	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
64	30 × 10 × 14	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
65	30 × 10 × 15	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
66	30 × 10 × 16	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
67	30 × 10 × 17	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
68	30 × 10 × 18	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
69	30 × 10 × 19	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
70	30 × 10 × 20	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
71	30 × 10 × 21	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
72	30 × 10 × 22	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
73	30 × 10 × 23	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
74	30 × 10 × 24	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**
75	30 × 10 × 25	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**

Note: best results are marked in bold for each instance.



(a) Real-world monitoring robot (b) 3D view of the monitoring robot (c) Robot wheels being processed

Fig. 17. A security monitoring robot and its wheels.

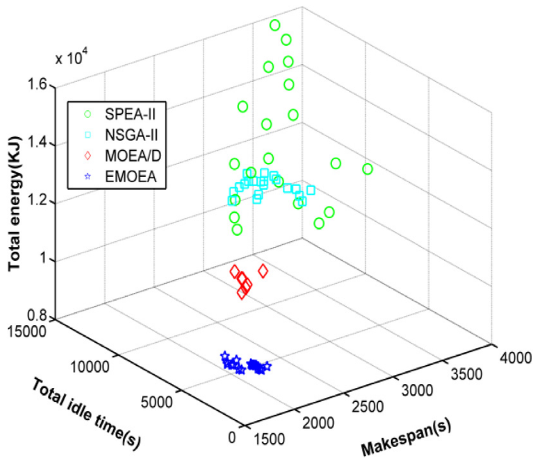


Fig. 18. Fronts of the four algorithms.

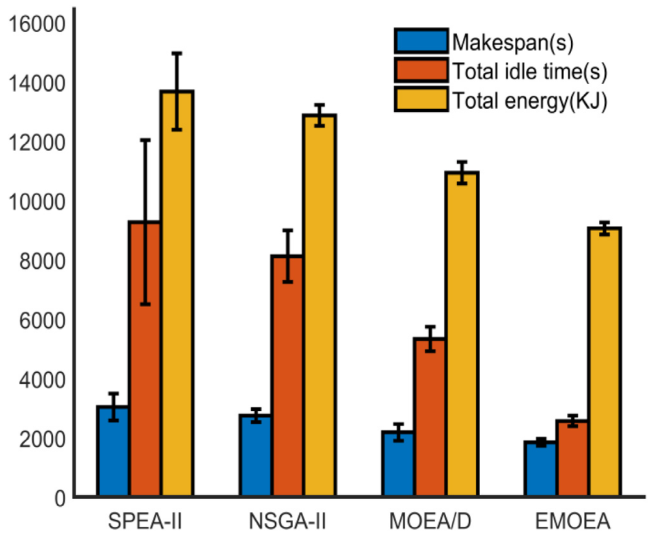


Fig. 19. Mean and standard deviations of the three objective values for the final solution sets obtained by the four algorithms.

Fig. 19 presents the mean and standard deviations of the three objective values for the final solution sets obtained by the four algorithms. Fig. 19 shows that the EMOEA was able to obtain better mean values in terms of the three studied objectives. For standard deviation, the EMOEA achieved the smallest values on the three studied objectives. Fig. 19 indicates that the EMOEA can obtain high-quality and relatively reliable candidate solutions for the real-world case.

Fig. 20 shows the Gantt chart of a candidate solution obtained by the EMOEA for the real-world case. From Fig. 20, we can clearly

Table 8
Mean and standard deviation values of HV with the SPEA-II, NSGA-II, MOEA/D and EMOEA on the real-world case.

SPEA-II	NSGA-II	MOEA/D	EMOEA
Mean (std)	Mean (std)	Mean (std)	Mean (std)
0.6413 (0.2507) +	0.4820 (0.1982) +	0.9222 (0.2134) +	**1.6334 (0.0872)**

Note: best results are marked in bold for the real-world case.

Table 9
Mean and standard deviation values of *C*-metric with the SPEA-II, NSGA-II, MOEA/D and EMOEA on the real-world case.

SPEA-II (S) v. EMOEA (E)		NSGA-II (N) v. EMOEA (E)		MOEA/D (M) v. EMOEA (E)	
C(S, E) Mean (std)	*C*(E, S) Mean (std)	*C*(N, E) Mean (std)	*C*(E, N) Mean (std)	*C*(M, E) Mean (std)	*C*(E, M) Mean (std)
0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**	0.0000 (0.0000) +	**1.0000 (0.0000)**

Note: better results are marked in bold for each instance.

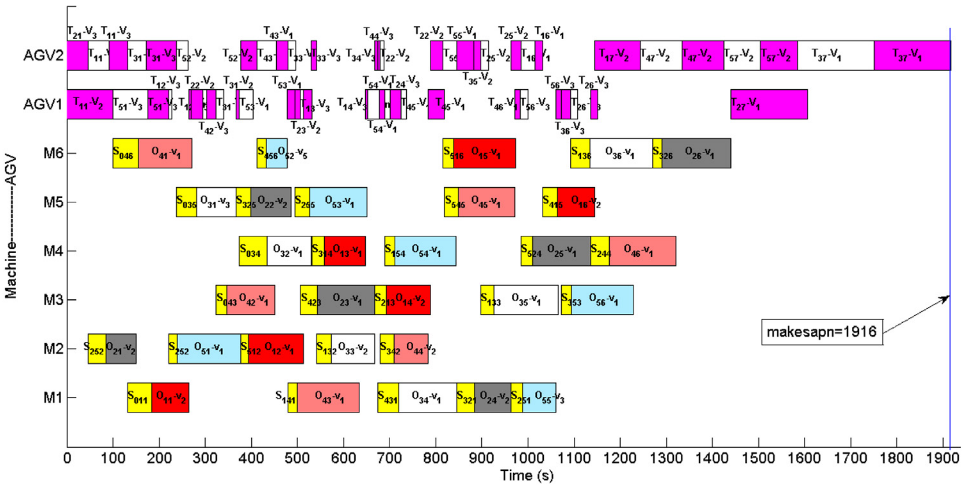


Fig. 20. The Gantt chart obtained by our proposed EMOEA on the real-world case.

see the operation sequence of the five types of robot wheels on each machine and the machine processing speed for each wheel's operation. In addition, AGVs and their transportation speeds can be effectively selected to cooperate with the processing operations of machines. With the cooperation of machines and AGVs, all robot wheel operations can be closely arranged, and finally, a shorter completion time can be obtained.

In summary, the above results clearly demonstrated that the EMOEA can provide better scheduling plans for the energy-efficient scheduling problem at hand integrated with multiple AGVs.

6. Conclusion and future work

This paper studied an energy-efficient JSP integrated with AGVs and formulated a new model for EJSP-AGVs. An EMOEA was also proposed, with effective encoding and decoding approaches, three types of crossover operators, and one mutation operator. To improve the convergence and enhance the local search ability, an effective OBL strategy was integrated into the EMOEA. In addition, a nondominated sorting-based external archive technique was used to store elitist solutions. The experimental study was conducted with generated benchmark instances and a real-world case, and Taguchi analysis was carried out to obtain the best combination of key parameters for the EMOEA. We used CPLEX to validate our proposed model. The AGV quantity configuration and system performance in a job shop were also studied and analyzed. Additionally, extensive experiments were conducted to assess the performance of the EMOEA by comparing it with three well-known algorithms. Our experimental results showed that (1) the overall performance of the production system can be enhanced by properly increasing the number of AGVs, and (2) the proposed EMOEA is able to outperform the other three algorithms in solving the EJSP-AGVs.

For future work, several issues deserve further attention. First, in this work, AGVs transported jobs by a predefined path without collision conflicts. In practice, however, both the path and collision of AGVs can affect the production performance. Therefore, it is important to consider path optimization and the collision of AGVs. Second, since dynamic events such as machine breakdowns often occur in real-world production systems, it is worthwhile to consider dynamic events in the EJSP-AGVs.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors thank the anonymous referees for their valuable suggestions that greatly improved the paper. This work is supported by the National Natural Science Foundation of China (Grants 62173263 and 72174160), and the Fundamental Research Funds for the Central Universities, China (Grant 2019-YB-033).

References

[1] G.L. Gong, Q.W. Deng, R. Chiong, et al., An effective memetic algorithm for multi-objective job-shop scheduling, Knowl.-Based Syst. 182 (2019) 104840.
[2] International Energy Agency (IEA), World Energy Investment Outlook, IEA, Paris, 2015.
[3] K. Fang, N. Uhan, F. Zhao, J.W. Sutherland, A new approach to scheduling in manufacturing for power consumption and carbon footprint reduction, J. Manuf. Syst. 30 (4) (2011) 234–240.
[4] C. Lu, L. Gao, X. Li, et al., Energy-efficient permutation flow shop scheduling problem using a hybrid multi-objective backtracking search algorithm, J. Clean. Prod. 144 (2017) 228–238.
[5] M. Dai, D. Tang, A. Giret, M.A. Salido, W.D. Li, Energy-efficient scheduling for a flexible flow shop using an improved genetic-simulated annealing algorithm, Robot. Comput.-Integr. Manuf. 29 (5) (2013) 418–429.
[6] X. Wu, A. Che, A memetic differential evolution algorithm for energy-efficient parallel machine scheduling, Omega 82 (2019) 155–165.
[7] R. Zhang, R. Chiong, Solving the energy-efficient job shop scheduling problem: a multi-objective genetic algorithm with enhanced local search for minimizing the total weighted tardiness and total energy consumption, J. Clean. Prod. 112 (2016) 3361–3375.
[8] G. Mouzon, M.B. Yildirim, A framework to minimise total energy consumption and total tardiness on a single machine, Int. J. Sustain. Eng. 1 (2) (2008) 105–116.
[9] A. Che, X. Wu, J. Peng, P. Yan, Energy-efficient bi-objective single-machine scheduling with power-down mechanism, Comput. Oper. Res. 85 (2017) 172–183.
[10] F. Shrouf, J. Ordieres-Mer_e, A. García-S_anchez, M. Ortega-Mier, Optimizing the production scheduling of a single machine to minimise total energy consumption costs, J. Clean. Prod. 67 (2014) 197–207.
[11] L.D. Xu, E.L. Xu, L. Li, Industry 4.0: state of the art and future trends, Int. J. Prod. Res. 56 (8) (2018) 2941–2962.
[12] R.Y. Zhong, X. Xu, E. Klotz, S.T. Newman, Intelligent manufacturing in the context of industry 4.0: A review, Engineering 3 (5) (2017) 616–630.
[13] M. Saidi-Mehrabad, S. Dehnavi-Arani, F. Evazabadian, V. Mahmoodian, An ant colony algorithm (ACA) for solving the new integrated model of job shop scheduling and conflict-free routing of AGVs, Comput. Ind. Eng. 86 (2015) 2–13.
[14] I.F.A. Vis, Survey of research in the design and control of automated guided vehicle systems, European J. Oper. Res. 170 (2006) 677–709.
[15] T. Le-Anh, M.B.M. De Koster, A review of design and control of automated guided vehicle systems, European J. Oper. Res. 171 (1) (2006) 1–23.
[16] X.F. Yao, Z.T. Lian, Y. Yang, et al., Wisdom manufacturing: new humans-computers-things collaborative manufacturing model, Comput. Integr. Manuf. Syst. 20 (6) (2014) 1490–1498.
[17] L. Tai, S.H. Li, M. Liu, Autonomous exploration of mobile robots through deep neural networks, Int. J. Adv. Robot. Syst. 14 (4) (2017) 1–9.
[18] L. Lin, S.W. Shinn, M. Gen, H. Hwang, Network model and effective evolutionary approach for AGV dispatching in manufacturing system, J. Int. Manuf. 17 (4) (2006) 465–477.
[19] E.H. Liu, X.F. Yao, T. Tao, H. Jin, Improved flower pollination algorithm for job shop scheduling problems integrated with AGVs, Comput. Integr. Manuf. Syst. 25 (9) (2019) 2219–2236.
[20] P. Lacomme, M. Larabi, N. Tchernev, Job-shop based framework for simultaneous scheduling of machines and automated guided vehicles, Int. J. Prod. Econ. 143 (1) (2013) 24–34.
[21] T.K. Dao, T.S. Pan, T.T. Nguyen, J.S. Pan, Parallel bat algorithm for optimizing makespan in job shop scheduling problems, J. Int. Manuf. 29 (2) (2018) 451–462.
[22] V. Kachitvichyanukul, S. Sitthitham, A two-stage genetic algorithm for multi-objective job shop scheduling problems, J. Int. Manuf. 22 (3) (2011) 355–365.
[23] W. Wisittipanich, V. Kachitvichyanukul, An efficient PSO algorithm for finding pareto-frontier in multi-objective job shop scheduling problems, Ind. Eng. Manage. Syst. 12 (2013) 151–160.
[24] W.F. Li, L.J. He, Y.L. Cao, Many-objective evolutionary algorithm with reference point-based fuzzy correlation entropy for energy-efficient job shop scheduling with limited workers, IEEE Trans. Cybern. (2022) http://dx.doi.org/10.1109/TCYB.2021.3069184, In press.
[25] L.J. He, R. Chiong, W.F. Li, et al., Multiobjective optimization of energy-efficient job-shop scheduling with dynamic reference point-based fuzzy relative entropy, IEEE Trans. Ind. Inf. 18 (1) (2022) 600–610.
[26] X.L. Wu, S.M. Wu, An elitist quantum-inspired evolutionary algorithm for the flexible job-shop scheduling problem, J. Int. Manuf. 28 (2017) 1441–1457.
[27] T.C. Chiang, H.J. Lin, A simple and effective evolutionary algorithm for multiobjective flexible job shop scheduling, Int. J. Prod. Econ. 141 (1) (2013) 87–98.
[28] X.N. Shen, X. Yao, Mathematical modeling and multi-objective evolutionary algorithms applied to dynamic flexible job shop scheduling problems, Inform. Sci. 298 (2015) 198–224.
[29] R. Sarker, M. Omar, S.M. Kamrul-Hasan, D. Essam, Hybrid evolutionary algorithm for job scheduling under machine maintenance, Appl. Soft Comput. 13 (3) (2013) 1440–1447.
[30] M. Abedi, R. Chiong, N. Noman, R. Zhang, A multi-population, multi-objective memetic algorithm for energy-efficient job-shop scheduling with deteriorating machines, Expert Syst. Appl. 157 (2020) 113348.
[31] Y. He, F. Liu, H.J. Cao, C.B. Li, A bi-objective model for job-shop scheduling problem to minimise both energy consumption and makespan, J. Cent. South Univ. Technol. 12 (2) (2005) 167–171.
[32] C. Subai, P. Baptiste, E. Niel, Scheduling issues for environmentally responsible manufacturing: the case of hoist scheduling in an electroplating line, Int. J. Prod. Econ. 99 (1–2) (2006) 74–87.
[33] C. Gahm, F. Denz, M. Dirr, A. Tuma, Energy-efficient scheduling in manufacturing companies: a review and research framework, European J. Oper. Res. 248 (3) (2016) 744–757.
[34] X.L. Wu, Y.J. Sun, A green scheduling algorithm for flexible job shop with energy-saving measures, J. Clean. Prod. 172 (2018) 3249–3264.
[35] G. Gong, R. Chiong, Q. Deng, W. Han, L. Zhang, D. Huang, Energy-efficient production scheduling through machine on/off control during preventive maintenance, Eng. Appl. Artif. Intell. 104 (2021) 104359.
[36] K.T. Fang, B.M. Lin, Parallel-machine scheduling to minimise tardiness penalty and power cost, Comput. Ind. Eng. 64 (1) (2013) 224–234.
[37] S.A. Mansouri, E. Aktas, U. Besikci, Green scheduling of a two-machine flow-shop: trade-off between makespan and energy consumption, European J. Oper. Res. 248 (3) (2016) 772–788.
[38] J.Y. Ding, S.J. Song, C. Wu, Carbon-efficient scheduling of flow shops by multi-objective optimization, European J. Oper. Res. 248 (3) (2016) 758–771.
[39] L. Yin, X. Li, L. Gao, C. Lu, Z. Zhang, Energy-efficient job shop scheduling problem with variable spindle speed using a novel multi-objective algorithm, Adv. Mech. Eng. 9 (2017) 1–21.
[40] J.Y. Ding, S. Song, R. Zhang, R. Chiong, C. Wu, Parallel machine scheduling under time-of-use electricity prices: new models and optimization approaches, IEEE Trans. Autom. Sci. Eng. 13 (2016) 1138–1154.
[41] A. Che, Y. Zeng, K. Lyu, An efficient greedy insertion heuristic for energy-conscious single machine scheduling problem under time-of-use electricity tariffs, J. Clean. Prod. 129 (2016) 565–577.
[42] H. Luo, B. Du, G.Q. Huang, H. Chen, X. Li, Hybrid flow shop scheduling considering machine electricity consumption cost, Int. J. Prod. Econ. 146 (2) (2013) 423–439.
[43] Z. Zeng, M. Hong, Y. Man, J. Li, Y. Zhang, H. Liu, Multi-object optimization of flexible flow shop scheduling with batch process-consideration total electricity consumption and material wastage, J. Clean. Prod. 183 (2018) 925–939.
[44] Y.Z. Zeng, A. Che, X.Q. Wu, Bi-objective scheduling on uniform parallel machines considering electricity cost, Eng. Optim. 50 (1) (2018) 19–36.
[45] I.A. Chaudhry, S. Mahmood, M. Shami, Simultaneous scheduling of machines and automated guided vehicles in flexible manufacturing systems using genetic algorithms, J. Cent. South Univ. Technol. 18 (2011) 1473–1486.
[46] M. Nageswararao, K.N. Rao, G. Rangajanardhana, Integration of strategic tactical and operational level planning of scheduling in FMS by metaheuristic algorithm, Int. J. Adv. Eng. Res. Stud. 1 (2) (2012) 10–20.
[47] A. Caumond, P. Lacomme, A. Moukrim, N. Tchernev, An MILP for scheduling problems in an FMS with one vehicle, European J. Oper. Res. 199 (3) (2009) 706–722.
[48] X. Zheng, Y.J. Xiao, Y. Seo, A tabu search algorithm for simultaneous machine/AGV scheduling problem, Int. J. Prod. Res. 52 (19) (2014) 5748–5763.
[49] T.F. Abdelmaguid, A.O. Nassef, B.A. Kamal, M.F. Hassan, A hybrid GA/heuristic approach to the simultaneous scheduling of machines and automated guided vehicles, Int. J. Prod. Res. 42 (2) (2004) 267–281.
[50] A.G. Babu, J. Jerald, A.N. Haq, V.M. Luxmi, T.P. Vigneswaralu, Scheduling of machines and automated guided vehicles in FMS using differential evolution, Int. J. Prod. Res. 48 (16) (2010) 4683–4699.
[51] B.S.P. Reddy, C.S.P. Rao, A hybrid multi-objective GA for simultaneous scheduling of machines and AGVs in FMS, Int. J. Adv. Manuf. Technol. 31 (2006) 602–613.

[52] P. Udhayakumar, S. Kumanan, Integrated scheduling of flexible manufacturing system using evolutionary algorithms, Int. J. Adv. Manuf. Technol. 61 (2012) 621–635.
[53] J. Xu, C.C. Wu, Y. Yin, W.C. Lin, An iterated local search for the multi-objective permutation flowshop scheduling problem with sequence-dependent setup times, Appl. Soft Comput. 52 (2017) 39–47.
[54] A. Allahverdi, The third comprehensive survey on scheduling problems with setup times/costs, European J. Oper. Res. 246 (2) (2015) 345–378.
[55] L.J. He, W.F. Li, Y. Zhang, Y.L. Cao, A discrete multi-objective fireworks algorithm for flowshop scheduling with sequence-dependent setup times, Swarm Evol. Comput. 51 (2019) 100575.
[56] L. Davis, Applying adaptive algorithms to epistatic domains, in: Proceedings of the IJCAI, 1985, pp. 162–164.
[57] D.E. Goldberg, R. Lingle, Alleles, loci, and the traveling salesman, in: J.J. Grefenstette (Ed.), Proceedings of the International Conference on Genetic Algorithms and their Applications, Lawrence Erbium, Hillsdale, New Jersey, Pittsburgh, PA, 1985.
[58] Y.L. Gao, Y.Z. Chen, Q.Y. Jiang, Multi-objective differential evolution algorithm based on the non-uniform mutation, Int. J. Model. Identif. Control 15 (4) (2012) 284–289.
[59] G.J. Sun, Y.F. Lan, R.Q. Zhao, Differential evolution with Gaussian mutation and dynamic parameter adjustment, Soft Comput. 23 (5) (2019) 1615–1642.
[60] B.H. Abed-alguni, Island-based cuckoo search with highly disruptive polynomial mutation, Int. J. Artif. Intell. 17 (1) (2019) 57–82.
[61] M. Wnętrzak, P. Błazej, P. Mackiewicz, Optimization of the standard genetic code in terms of two mutation types: point mutations and frameshifts, BioSystems 181 (2019) 44–50.
[62] H.R. Tizhoosh, Opposition-based learning: a new scheme for machine intelligence, in: Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06), IEEE, 2005.
[63] M.A. Ahandani, H. Alavi-rad, Opposition-based learning in shuffled frog leaping: an application for parameter identification, Inform. Sci. 291 (291) (2015) 19–42.
[64] M.A. Remli, S. Deris, M.S. Mohamad, S. Omatu, J.M. Corchado, An enhanced scatter search with combined opposition-based learning for parameter estimation in large-scale kinetic models of biochemical systems, Eng. Appl. Artif. Intell. 62 (C) (2017) 164–180.
[65] L.J. He, W.F. Li, R. Chiong, et al., Optimising the job-shop scheduling problem using a multi-objective Jaya algorithm, Appl. Soft Comput. 111 (2021) 107654.
[66] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.
[67] D.C. Montgomery, Design & Analysis of Experiments, John Wiley and Sons, New York, 2008.
[68] G.L. Gong, R. Chiong, Q.W. Deng, W.W. Han, L. Zhang, W.H. Lin, K.X. Li, Energy-efficient flexible flow shop scheduling with worker flexibility, Expert Syst. Appl. 141 (2020) 112902.
[69] E. Zitzler, M. Laumanns, L. Thiele, Spea2: Improving the Strength Pareto Evolutionary Algorithm, TIK-report, 2001.
[70] Q. Zhang, H. Li, MOEA/D: a multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2007) 712–731.