Hindawi
Complexity
Volume 2020, Article ID 8853735, 14 pages
https://doi.org/10.1155/2020/8853735

Research Article

Multisystem Optimization for an Integrated Production Scheduling with Resource Saving Problem in Textile Printing and Dyeing

Haiping Ma,[1] **Chao Sun**,[1] **Jinglin Wang**,[2] **Zhile Yang**,[3] **and Huiyu Zhou**[4]

[1]*Department of Electrical Engineering, Shaoxing University, Shaoxing, Zhejiang 312000, China*
[2]*Department of Electrical and Electronic Engineering, University of Nottingham Ningbo China, Ningbo, Zhejiang 315000, China*
[3]*Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, Guangdong 518055, China*
[4]*Department of Informatics, University of Leicester, Leicester, LE1 7RH, UK*

Correspondence should be addressed to Haiping Ma; mhping1981@126.com

Received 15 September 2020; Revised 15 October 2020; Accepted 26 October 2020; Published 18 November 2020

Academic Editor: Shi Cheng

Resource saving has become an integral aspect of manufacturing in industry 4.0. This paper proposes a multisystem optimization (MSO) algorithm, inspired by implicit parallelism of heuristic methods, to solve an integrated production scheduling with resource saving problem in textile printing and dyeing. First, a real-world integrated production scheduling with resource saving is formulated as a multisystem optimization problem. Then, the MSO algorithm is proposed to solve multisystem optimization problems that consist of several coupled subsystems, and each of the subsystems may contain multiple objectives and multiple constraints. The proposed MSO algorithm is composed of within-subsystem evolution and cross-subsystem migration operators, and the former is to optimize each subsystem by excellent evolution operators and the later is to complete information sharing between multiple subsystems, to accelerate the global optimization of the whole system. Performance is tested on a set of multisystem benchmark functions and compared with improved NSGA-II and multiobjective multifactorial evolutionary algorithm (MO-MFEA). Simulation results show that the MSO algorithm is better than compared algorithms for the benchmark functions studied in this paper. Finally, the MSO algorithm is successfully applied to the proposed integrated production scheduling with resource saving problem, and the results show that MSO is a promising algorithm for the studied problem.

1. Introduction

Industry 4.0, a new wave of industrialization, driven by smart information and communication technology, sparks a transformative view of framing process manufacturing and factory management practice [1–3]. With the recent technical advances in industry, such as sensor network, cloud computing, and artificial intelligence, production manufacturing is becoming hyperconnected because of the communication and control between more components and parts than those in the past [4–6]. So, the optimization and decision for manufacturing is generally impossible with analytical mathematical tools or traditional optimization methods [7, 8]. Meanwhile, manufacturing involves multiple subsystems, with multiple inputs, multiple outputs, multiple objectives, and multiple constraints. Its optimization and decision cannot be simply treated as a classic multiobjective optimization problem. We treat such manufacturing as a complex multisystem because it is not merely a single optimization problem but is rather a combination of several multiobjective optimization problems [9, 10]. That is, manufacturing is composed of multiple subprocesses, and each subprocess can be considered as an independent subsystem. These subsystems are coupled with the others, and they share some objectives or constraints with each other, and others are unique to a particular subsystem but must still be satisfied. Therefore, the optimization and decision for manufacturing in industry 4.0 is a more challenging problem than traditional multiobjective optimization in a single system.

In recent years, there are a few studies on the optimization of complex multisystem problems. Martins and

Lambe [11] presented the multidisciplinary design optimization (MDO) method, which was a framework of optimization methods for solving complex systems that involved multiple disciplines or subsystems. The basic principle of MDO was that the solution of a complex system depended not only on the solution of individual subsystems but also on their interactions, and its disadvantage was that it was only a framework that provided conceptual structures, but it did not specify the details of the underlying algorithms within that framework. That is, depending on the specific complex problem or the designer's preference, MDO required the designer to select particular optimization algorithms as the kernel components of MDO, which itself was a difficult problem. For the optimization of the complex multisystem, many researchers preferred to translate it as a multiobjective optimization problem and then adopted classical heuristic methods to seek optimal solutions. Zhang et al. [12, 13] proposed an efficient evolutionary optimization framework based on decomposition, which decomposed a complex problem into a series of single-objective optimization subproblems and then used the interactive information between adjacent subproblems to accelerate the optimization of each subproblem to obtain the optimal solution. Antonio and Coello [14] proposed a coevolutionary optimization (CO) method, inspired by biological coevolution and divided-and-autonomous strategy, and it established multiple competing or cooperating populations for evolving multiple subsystems to achieve global optimization. Ong and Gupta [15, 16] proposed a multiobjective multifactorial evolutionary algorithm (MO-MFEA) for solving multiple tasks in a complex system. Its basic idea was to unify the solution space of different tasks into an extended solution space and then performed implicit genetic transfer operation, which could exploit the transferable knowledge between multiple optimization tasks. The simulation results showed that MO-MFEA was competitive on multitask benchmarks. Recently, Du and Simon [17, 18] proposed a new optimization framework called BBO/Complex for solving the complex system using biogeography-based optimization (BBO). Its advantage was that it combined multipopulation architecture with the low-level optimization strategy to obtain a single method. The experimental results showed that the proposed BBO/Complex successfully solved four real-world optimization problems.

The aim of this paper is to propose and study a multisystem optimization (MSO) algorithm, combining the multipopulation architecture of BBO/Complex with the classical NSGA-II to establish within-subsystem evolution and cross-subsystem migration operators and then apply the new MSO algorithm to solving the production scheduling with resource saving problem in textile printing and dyeing plants. This paper shows how the multipopulation architecture can be integrated with popular heuristic methods to extend the application scenario of optimization algorithms and then presents a comparative study on multisystem benchmark functions and a textile printing and dyeing scheduling problem. The method in this paper could also serve as a template for the extension of any other heuristic methods to multisystem optimization.

The motivation of proposing MSO algorithm in this research is twofold. First, we have observed that there widely exist a large number of complex coupled systems in manufacturing, as mentioned above, and their optimization is more challenging and difficult than traditional multiobjective optimization in a single system. Second, we have observed that two main foundations of the MSO algorithm, multipopulation and NSGA-II, have proven themselves to be effective optimization architecture and method. Combining these two observations leads us to propose multisystem optimization, as a high-performing optimization method.

The original contributions of this paper include the following. (a) A new real-world-based textile printing and dyeing production scheduling problem considering resource saving is formulated as a multisystem optimization problem. (b) The idea of multisystem optimization is proposed in the heuristic computing field to establish the new optimization paradigm. Results show that the MSO algorithm outperforms compared algorithms for most of benchmark functions that we study. (c) The MSO algorithm solves the textile printing and dyeing production scheduling problem considering resource saving.

The remainder of this paper is organized as follows. Section 2 builds a mathematical model of textile printing and dyeing production scheduling problem considering resource saving. Section 3 realizes the MSO algorithm integrating multipopulation and NSGA-II, and Section 4 verifies its superiority with simulation results. Section 5 applies the MSO algorithm to solve the proposed production scheduling problem in textile printing and dyeing. Some concluding remarks and directions for future work are provided in Section 6.

2. Problem Descriptions and Modeling

The multisystem optimization problem studied in this paper is derived from a textile printing and dyeing plant. Its main processes include dyeing, printing, and finishing, and each process consists of multiple procedures. For example, the dyeing process includes singeing, bleaching, mercerizing, setting, and so on. According to raw materials and fabric structures, the process flow is not a completely fixed model for specific printing and dyeing requirements. Meanwhile, with the rapid development of manufacturing, resource saving and environmental protection have been increasingly concerned. Textile printing and dyeing is an energy- and resource-intensive manufacturing process and uses physical and chemical methods to make products out of the raw textile, with extensive electricity and water as the treatment mediums. Therefore, it is urgent for manufacturers to improve production efficiency while reducing electricity consumption and water wastage.

In this section, we propose an integrated optimization model for textile printing and dyeing by considering both production efficiency and resource saving. Figure 1 shows a workflow of textile printing and dyeing processes, where the model includes a production scheduling subsystem based on production orders and a resource saving subsystem based on

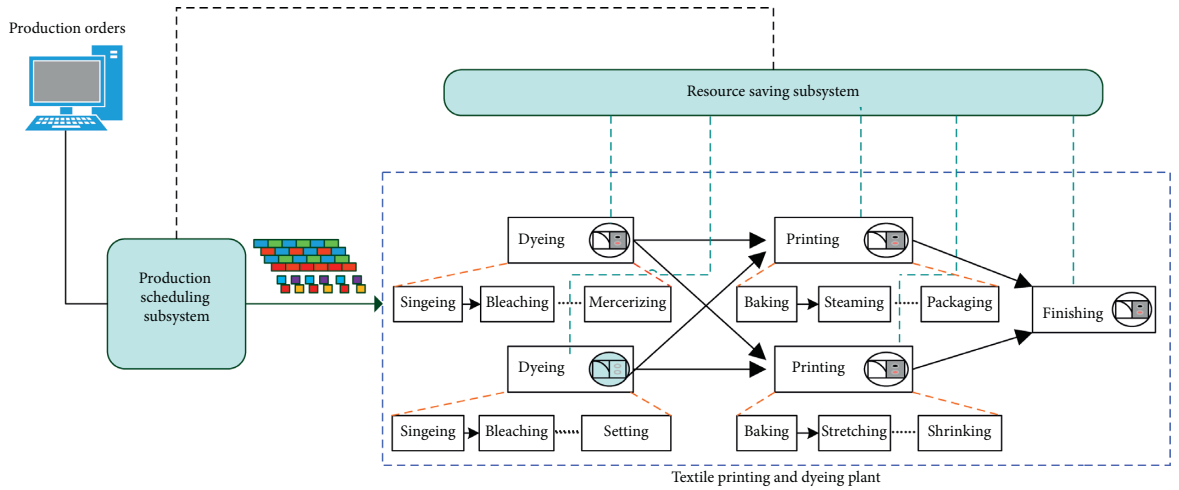

Figure 1: Workflow of textile printing and dyeing.

environment protection standard. These two subsystems are coupled with the other and share partial variables and constraints. The whole model is a hybrid flow-shop scheduling problem with unrelated parallel machines, and it is a nondeterministic polynomial hard (NP-hard) problem. In the following, we will focus on a mathematical formulation of this integrated optimization model. For convenience, the symbols and notations used in the model are shown in Table 1.

For a production scheduling subsystem in textile printing and dyeing, its mathematic model is formulated as a multiobjective optimization problem. Suppose the subsystem schedules a set of jobs $\{J_i\}_{i=1}^{n}$, and each job is completed orderly by m procedures. Each procedure has M_m unrelated parallel machines, and each job is worked on one machine in a procedure. A set-up time is required before a procedure of a job is processed on a machine. The production scheduling subsystem has two conflicting objectives of makespan and production cost, which should be minimized, respectively. This subsystem is defined as follows:

$$\min F_1 = (f_{11}, f_{12}), \tag{1}$$

$$f_{11} = \max_{i\in 1,2,\ldots,n} \{D_i\}, \tag{2}$$

$$f_{12} = \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{M_m} w_{ijk} C_{ijk}. \tag{3}$$

In equation (1), F_1 denotes the production scheduling subsystem, including makespan f_{11} and production cost f_{12}. In equation (2), D_i is the finishing time of the ith job through all procedures, which is iteratively calculated by

$$\begin{aligned} D_{ij} &= \max_{j\in 1,2,\ldots,m-1} \left\{D_{i(j-1)} + T_{ijk}, D_{(i-1)(j+1)}\right\}, \\ D_i &= D_{i(m-1)} + T_{imk}, \end{aligned} \tag{4}$$

where T_{ijk} is the processing time of the jth procedure of the ith job running on the kth machine and D_{ij} is the finishing time of the ith job through the jth procedure. In equation (3), C_{ijk} is production cost of the jth procedure of the ith job running on the kth machine. If the jth procedure of the ith job is processed by the kth machine, then $w_{ijk} = 1$; otherwise, $w_{ijk} = 0$.

For a resource saving subsystem in textile printing and dyeing, electricity consumption, water wastage, and processing cost caused by these components are the most important factors, which vary with the process sequences. The mathematic model of this subsystem is also formulated as a multiobjective optimization problem, including three conflicting objectives of electricity consumption, water wastage, and processing cost, which should be minimized, respectively. The subsystem is defined as follows:

$$\min F_2 = (f_{21}, f_{22}, f_{23}), \tag{5}$$

$$\begin{aligned} f_{21} = {} & \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{M_m} w_{ijk} T_{ijk} P_{ijk}^{\mathrm{PE}} + \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{M_m} w_{ijk} T_{ijk}^{\mathrm{SE}} P_{jk}^{\mathrm{SE}} \\ & + \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{M_m} w_{ijk} T_{ijk}^{\mathrm{BE}} P_{jk}^{\mathrm{BE}} \\ & + \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{M_m} w_{ijk}\left(T_{ijk} + T_{ijk}^{\mathrm{SE}} + T_{ijk}^{\mathrm{BE}}\right) P_{jk}^{\mathrm{OE}}, \end{aligned} \tag{6}$$

$$f_{22} = \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{M_m} w_{ijk} T_{ijk} Q_{ijk}^{\mathrm{PW}} + \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{M_m} w_{ijk} T_{ijk}^{\mathrm{SE}} Q_{jk}^{\mathrm{SW}}, \tag{7}$$

$$f_{23} = \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{M_m} w_{ijk} T_{ijk} \kappa_{ijk}. \tag{8}$$

In equation (5), F_2 denotes the resource saving subsystem, including electricity consumption f_{21}, water wastage f_{22}, and the corresponding processing cost f_{23}. Electricity consumption denoted by equation (6) is composed of four parts: processing electricity consumption, set-up electricity consumption, standby electricity consumption, and auxiliary electricity consumption. The first term in equation (6) represents processing

Table 1: The symbols and notations used in the proposed model.

Indices	
I	Job
J	Procedure
k	Machine
n	Number of jobs
m	Number of procedures
M_m	Number of parallel machines
Variables	
w_{ijk}	= 1 if the jth procedure of the ith job is processed on the kth machine
$\{J_i\}_{i=1}^{n}$	Job sequence $\{J_1, J_2, J_3, \ldots, J_n\}$
Parameters	
C_{ijk}	Production cost of the jth procedure of the ith job on the kth machine
κ_{ijk}	Unit processing cost of the jth procedure of the ith job on the kth machine
D_i	Finishing time of the ith job through all procedures
D_{ij}	Finishing time of the ith job through the jth procedures
T_{ijk}	Processing time of the jth procedure of the ith job on the kth machine
T_{ijk}^{SE}	Set-up time of the kth machine when the (j-1)th procedure changes to the jth procedure for the ith job
T_{ijk}^{BE}	Standby time of the kth machine when the (j-1)th procedure changes to the jth procedure for the ith job
P_{ijk}^{PE}	Unit processing power of the jth procedure of the ith job on the jth machine
P_{jk}^{SE}	Unit set-up power of the kth machine when the (j-1)th procedure changes to the jth procedure for the ith job
P_{jk}^{BE}	Unit standby power of the kth machine when the (j-1)th procedure changes to the jth procedure for the ith job
P_{jk}^{OE}	Auxiliary unit power of the kth machine during the jth procedure
Q_{ijk}^{PW}	Unit water wastage of the kth machine when processing the jth procedure of the ith job
Q_{jk}^{SW}	Unit water wastage of the kth machine in the set-up stage of the jth procedure

electricity consumption, where P_{ijk}^{PE} is the unit processing power of the kth machine when processing the jth procedure of the ith job. The second term represents set-up electricity consumption, where T_{ijk}^{SE} and P_{jk}^{SE} are, respectively, the set-up time and unit power of the kth machine when the $(j-1)$th procedure changes to the jth procedure for the ith job. The third term represents standby electricity consumption, where T_{ijk}^{BE} and P_{jk}^{BE} are, respectively, the standby time and unit power of the kth machine when the $(j-1)$th procedure changes to the jth procedure for the ith job. The fourth term represents auxiliary electricity consumption of the auxiliary equipments in the machining process, which is relevant to the production time and auxiliary power, with the former consisting of processing time, set-up time, and standby time, and P_{jk}^{OE} is the auxiliary unit power of the kth machine in the jth procedure.

Equation (7) denotes water wastage, which is different from electricity consumption. It considers two types of water wastage, including processing water wastage and set-up water wastage, and water wastage for auxiliary equipments and the standby stage of machines is neglected. In equation (7), the first term represents processing water wastage, and Q_{ijk}^{PW} is the unit water wastage of the kth machine when processing the jth procedure of the ith job. The second term represents set-up water wastage, and Q_{jk}^{SW} is the unit water wastage of the kth machine in the set-up stage of the jth procedure.

Equation (8) denotes the total processing cost, including raw material cost, electricity consumption cost, water consumption cost, and so on. κ_{ijk} is unit processing cost of the jth procedure of the ith job running on the kth machine.

Furthermore, for the proposed integrated optimization model of textile printing and dyeing, the solutions must satisfy the following constraints.

Each machine handles exactly one procedure in a job:

$$\sum_{i=1}^{n} w_{ijk} = 1, \quad \text{for } j \in \{1, 2, 3, \ldots, m\}, k \in \{1, 2, 3, \ldots, M_m\}, \tag{9}$$

and each procedure of each job is processed in only one machine sometime:

$$\sum_{k=1}^{M_m} w_{ijk} = 1, \quad \text{for } i \in \{1, 2, 3, \ldots, n\}, j \in \{1, 2, 3, \ldots, m\}. \tag{10}$$

In the above constraints, equation (9) is stand for assignment of one procedure in only one job to a machine, and equation (10) is stand for assignment of each procedure of each job to only one machine.

Now, an integrated optimization model for textile printing and dyeing has been presented, and the following section develops a multisystem optimization algorithm that will be used to solve the proposed optimization problem.

3. Multisystem Optimization

This section first presents a MSO framework, which can serve as a template for extending any other heuristic methods to the MSO algorithm. Then, it presents the implementation of the proposed MSO algorithm.

3.1. MSO Framework. A MSO problem in this paper consists of multiple subsystems, which are coupled with the others and share partial objectives or constraints. That is, each subsystem in a MSO problem has not only coupled objectives and constraints but also independent objectives and constraints. Therefore, multisystem optimization is more complicate than traditional multiobjective optimization.

Suppose that we have a complex system that consists of several subsystems. Without loss of generality, all subsystems are assumed to be minimization problems. Inspired by implicit parallelism of multipopulation heuristic approaches [19, 20], we first treat a subpopulation as a subsystem to optimize a subsystem optimization problem by excellent evolution operators. Then, we realize information sharing between multiple subsystems by migration, based on the relations of sharing variables and similarity levels between objectives and constraints, to accelerate the global optimization of the whole system. Based on the above idea, each subsystem is comprised of three sets of elements. The first set includes candidate solutions to the subsystem optimization problem. The second and third sets include the objectives and constraints of the subsystem. The MSO algorithm mainly includes two steps: evolution within subsystems and sharing information via migration across subsystems. We refer to these two types of operators as within-subsystem evolution and cross-subsystem migration. The MSO framework is illustrated in Figure 2, where a complex multisystem problem includes multiple coupled subsystems, and each subsystem includes multiple objectives and constraints. Within-subsystem evolution is used in each subsystem, and cross-subsystem migration is used between multiple subsystems.

3.2. Implementation of MSO Algorithm. In the proposed MSO architecture, we use a modified version of NSGA-II [21], initially designed for single systems with multiobjectives, as a within-subsystem evolution operator. The modified NSGA-II employs solution ranks as selection probabilities considering the relative performance of a candidate solution because each subsystem has its own set of candidate solutions, objectives, and constraints, and the ranks assigned to the candidate solutions in a subsystem denote the relative fitness of those solutions only in that particular subsystem. Then, we recombine the candidate solutions using any desired recombination method in heuristic methods. Finally, we mutate the child population and replace the parents with the children.

Cross-subsystem migration is an important operator in the MSO algorithm. For the development of heuristic methods including the proposed MSO algorithm in this paper, we must consider two challenges. One is to converge to the optimal solutions. To address this challenge in the MSO algorithm, we define similarity levels for both objectives and constraints. If two subsystems have high similarity levels, the optimization problems of those subsystems are similar to each other. This also means that the features that are important in one subsystem have a similar level of importance in the other subsystem. Migration between subsystems with similar objectives and constraints is expected to be helpful for all such subsystems. Another important challenge is to maintain population diversity as the main factor that enables the population to improve. If the population has a low diversity, most of candidate solutions are similar to each other, and the probability that a candidate solution improves after migration is low. In this case, migration may not effectively contribute to improvement in the population.

In cross-subsystem migration, we first use the similarity levels of both constraints and objectives to find pairs of subsystems that are suitable for migration. The similarity level calculation is based on the fast similarity level calculation (FSLC) [10] shown in Algorithm 1, where G and H are the sets of objective costs or constraints of two solutions in different subsystems. The pair probability P_{sub} between the subsystems is calculated as follows:

$$P_{\text{sub}} = \frac{1}{2}\left(\frac{\text{SL}_1}{\text{SL}_{1,\max}} + \frac{\text{SL}_2}{\text{SL}_{2,\max}}\right), \tag{11}$$

where SL_1 and SL_2 are, respectively, the objective and constraint similarity levels between two solutions in different subsystems and $\text{SL}_{1,\max}$ and $\text{SL}_{2,\max}$ are, respectively, the maximum objective and constraint similarity levels in the entire system.

Then, we use Euclidian distances between candidate solutions in a pair of subsystems to decide which solutions migrate between each other. In a multisystem problem, subsystems typically have different candidate solution structures, and the solution variables are at least partly different between subsystems. Table 2 illustrates an example about candidate solution structures in different subsystems. In the table, the candidate solution in subsystem 1 has four variables composed of Var-1, Var-2, Var-4, and Var-5, and the candidate solution in subsystem 2 has three variables composed of Var-1, Var-3, and Var-5. Apparently, these two candidate solutions have two variables Var-1 and Var-5 in common. To calculate their distances, we need to have identical solution structures and use an N/A value to denote a missing variable, shown as [1, 2, N/A, 0, 3] for the solution in subsystem 1 and [3, N/A, 4, N/A, 5] for the solution in subsystem 2.

Next, we use the partial distance strategy (PDS) [17, 22] to calculate Euclidian distance between candidate solutions in different subsystems, which is given as follows:

$$\sigma_{ikjl} = \begin{cases} \dfrac{N}{I_{ikjl}} \sqrt{\displaystyle\sum_{s=1}^{N} \left(y_{ik}(s) - y_{jl}(s)\right)^2 I_{ikjl}(s)}, & \text{if } I_{ikjl} > 0, 0, \text{if } I_{ikjl} = 0, \end{cases} \tag{12}$$

where

$$\begin{aligned} I_{ikjl} &= \sum_{s=1}^{N} I_{ikjl}(s), \\ I_{ikjl}(s) &= \begin{cases} 0, & \text{if } y_{ik}(s) = N/A \text{ or } y_{jl}(s) = N/A, \\ 1, & \text{if } y_{ik}(s) \neq N/A \text{ and } y_{jl}(s) \neq N/A. \end{cases} \end{aligned} \tag{13}$$

In equation (12), σ_{ikjl} denotes the partial distance between candidate solution y_{ik} in subsystem i and candidate

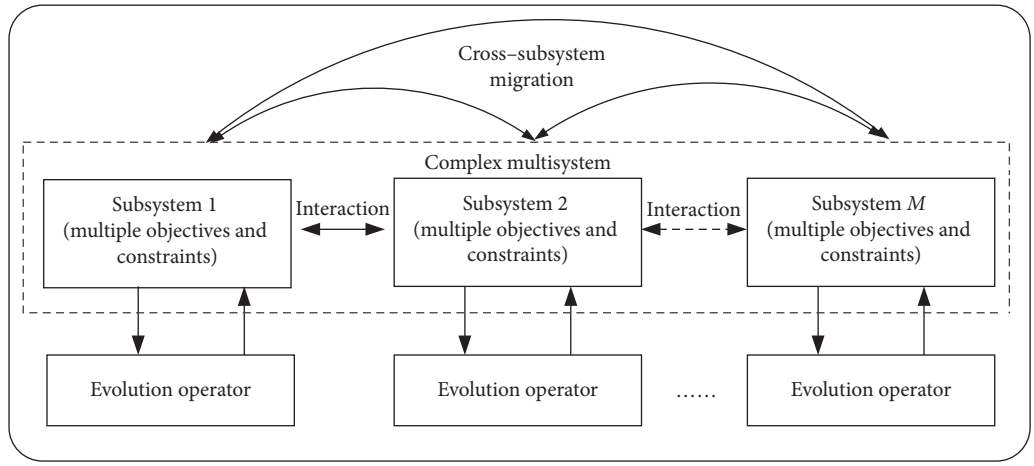

Figure 2: The MSO framework, mainly including within-subsystem evolution and cross-subsystem migration.

```
Set the objective or constraint similarity level SL = 0
For each g ∈ G, where G is the set of objectives or constraints of one solution
   For each h ∈ H, where H is the set of objectives or constraints of another solution
      If g = h, then
         SL = SL + 1
      End if
   End for
End for
```

Algorithm 1: Similarity level calculation across subsystems.

solution y_{jl} in subsystem j, which quantifies the difference level between each pair of solutions. N denotes the total number of solution variable for the entire system, and $y_{ik}(s)$ denotes the sth solution variable of y_{ik}.

Finally, we perform migration between the immigrating solution y_{ik} and the emigrating solution y_{jl} cross the subsystems. We use partial distance σ to calculate migration probabilities and then use the roulette-wheel selection method to probabilistically choose emigrating solution based on the migration probabilities. Migration is defined as

$$y_{ik}(s) \leftarrow y_{jl}(s). \tag{14}$$

Equation (14) states that a solution variable in the emigrating solution replaces one in the immigrating solution. In cross-subsystem migration, each solution variable in each immigrating solution in a subsystem has a chance to be replaced by a solution variable of an emigrating solution from another subsystem. One generation of the proposed MSO algorithm is shown in Algorithm 2.

4. Simulation Results

This section presents the optimization performance of the proposed MSO algorithm on the benchmark functions from the literature [23]. These functions are originally to evaluate the performance of MO-MFEA. Here, we take a task of a function as a subsystem, and each subsystem includes a multiobjective function. Multitask benchmark functions naturally change to multisystem functions suitable for the evaluation of the proposed MSO. These functions are built considering intersecting degrees of solution variables: complete intersection, partial intersection, and no intersection, and similarity degree of objectives: high similarity, medium similarity, and low similarity. Accordingly, there are nine combinations in total, and they are briefly summarized in Table 3, where the functions are named as follows: the first capital letters "*C*," "*P*," and "*N*" denote "complete," "partial," and "no," respectively, the second capital letter "I" denotes "intersection", the third capital letters "*H*," "*M*," and "*L*" denote "high," "medium," and "low," respectively, and the last capital letter "*S*" denotes "similarity". The aim of such categories is to have a comprehensive evaluation of the proposed method. In addition, $sim(T_1, T_2)$ denotes the similarity coefficient between subsystems T_1 and T_2 in a benchmark function.

In this experiment, we use the proposed MSO algorithm for solving simultaneously two subsystems of a benchmark function. To show the benefits of the proposed method, the results are compared with those of MO-MFEA and improved NSGA-II [24]. MO-MFEA is a multifactorial evolutionary algorithm [15], which is to simultaneously tackle multiple optimization tasks. Improved NSGA-II uses multiple populations with different crossover operators to separately solve each subsystem in a benchmark function. To ensure a fair comparison, the parameter configurations of

Table 2: Candidate solution structures in different subsystems.

Subsystem 1	Var-1	Var-2	Var-3	Var-4	Var-5
Solution	1	2	N/A	0	3
Subsystem 2	Var-1	Var-2	Var-3	Var-4	Var-5
Solution	3	N/A	4	N/A	5

these algorithms are adjusted to be optimal through many experiments to obtain the appropriate performance. The population size of NSGA-II is set to 100 for solving a single subsystem, while the population size of the proposed MSO algorithm and MO-MFEA is set to 200 for solving a benchmark function with two subsystems. The maximal number of function evaluations on a subsystem is set to 100,000 for NSGA-II, while the proposed MSO algorithm and MO-MFEA use 200,000 for a benchmark function since they solve two subsystems in a function together at a time. That is, for each subsystem, their maximal number of function evaluations is still 100,000. For all algorithms, we use the same simulated binary crossover with a crossover probability $p_c = 1$ and use the same polynomial mutation with a mutation probability $p_m = 1$. To compare with existing results, we use the inverted generational distance (IGD) in the literature [23] to evaluate the performance on each subsystem of the considered benchmark function. A small IGD value means both good convergence and good diversity for a minimization problem. Table 4 shows the average and standard deviation of IGD values for the proposed MSO algorithm, MO-MFEA, and NSGA-II, and all results are computed from 30 independent simulations.

The results shown in Table 4 lead to the following discussion. First, considering a similarity degree of objectives, we take CIHS, PIHS, and NIHS as a group of high similarity, CIMS, PIMS, and PIHS as a group of medium similarity, and CILS, PILS, and NILS as a group of low similarity. From Table 4, we observe that for high and medium similarity groups, the proposed MSO algorithm performs better than MO-MFEA and NSGA-II for all benchmark functions. For the low similarity group, the proposed MSO algorithm performs better than MO-MFEA and NSGA-II but worse than MO-MFEA for two of the benchmark functions (CILS and NILS). It indicates that objective similarity has a great influence on optimization performance of the proposed MSO algorithm, and it is an important factor composing cross-subsystem migration in the MSO algorithm.

Second, considering intersection degree of solution variables, we take CIHS, CIMS, and CILS as a group of complete intersection, PIHS, PIMS, and PILS as a group of partial intersection, and NIHS, NIMS, and NILS as a group of no intersection. From Table 4, we observe that for partial intersection, the proposed MSO algorithm is better than MO-MFEA and NSGA-II for all benchmark functions. For the complete intersection group and no intersection group, the proposed MSO algorithm is better than NSGA-II but worse than MO-MFEA for CILS and NILS. It indicates that solution variable intersection has a certain effect on optimization performance of the proposed MSO algorithm, and it is another important factor composing cross-subsystem migration in the MSO algorithm.

Furthermore, we also use symbol "†" to show statistically significant differences between the proposed MSO algorithm and compared algorithms based on the p value, which is smaller than 0.05 regarded as the significance level. In Table 4, out of 18 groups of data, there are 14 statistically significant differences between the proposed MSO algorithm and MO-MFEA and there are 16 statistically significant differences between the proposed MSO algorithm and NSGA-II. Based on this result, the probability that MSO and compared algorithms are from the same distribution is low. It indicates that the proposed MSO is an independent algorithm.

Combining with the above observations, we find that as a whole, the proposed MSO algorithm performs better than the other algorithms used in this paper for the most of benchmark functions. It implies that the MSO algorithm can fully consider the inheritance of evolution information and relationship between optimization environment and performance, and migration between subsystems can effectively utilize these factors to accelerate global optimization.

5. Application of MSO to Manufacturing Optimization

In this section, we use the proposed MSO algorithm to solve the integrated production scheduling with resource saving in textile printing and dyeing described in Section 2. In the traditional textile printing and dyeing industry, production scheduling is conducted by workers. Manual scheduling is very labor-consuming, and it usually leads to unsatisfactory schemes. Moreover, resource consumption is rarely considered when workers schedule jobs. In this experiment, 10 instances with different procedures and number of machines are investigated, which are shown in Table 5. These instances are generated from a real textile dyeing and printing plant located in Eastern China, and we use uniform distribution for the range of model parameters based on manual scheduling experience of workers because their real values are difficult to obtain. The job processing time in each machine is uniformly distributed in [50 100] minutes, and the set-up time and standby time of a machine are uniformly distributed in [5 10] minutes. Production cost and unit processing cost are uniformly distributed in [50 120] and [20 80], respectively, and various unit electricity consumption and unit water wastage are uniformly distributed in [20 50] and [2 10], respectively. The tuning parameters of the proposed MSO algorithm and the compared algorithms are the same as those used in the benchmark functions. In addition, the proposed integrated optimization model is a combinatorial optimization problem. We use permutation coding for each procedure, and the decoding includes the sorting of jobs and the allocation of machines. For job

Calculate the similarity levels of objectives and constraints between each pair of subsystems
Calculate the rank of each solution y_{ik} in subsystem i
Calculate migration probability λ_{ik}, which is linearly proportional to the rank of y_{ik}
For each subsystem i
  Perform within-subsystem evolution using the modified NSGA-II
    Find a suitable subsystem j to pair with subsystem i based on similarity levels of objectives and constraints
  For each solution y_{ik}
      Calculate partial distances $\{\sigma_{ikjl}\}$ between y_{ik} and each solution in subsystem j
    For each solution variable s
      Use λik to decide whether or not to immigrate to y_{ik}
      If migrating, then
          Select emigrating solution y_{jl} using roulette wheel selection, and its probability is proportional to partial distances $\{\sigma_{ikjl}\}$
        y_{ik} (s) $\longleftarrow$ y_{jl} (s)
      End if
    End for
  End for
End for

Algorithm 2: One generation of the proposed MSO algorithm.

Table 3: Benchmark function sets.

Function	$Sim\ (T_1, T_2)$	Subsystem no.	Pareto set	Pareto front	Properties
CIHS	0.97	T_1	$x_1 \in [0,1]$, $x_i = 0, i = 2{:}\ 50$	$f_1^2 + f_2^2 = 1$, $f_1 \geq 0, f_2 \geq 0$	Concave, unimodal Separable
		T_2	$x_1 \in [0,1]$, $x_i = 0, i = 2{:}\ 50$	$f_2 = 1 - f_1^2$, $0 \leq f_1 \leq 1$	Concave, unimodal Separable
CIMS	0.52	T_1	$x_1 \in [0,1]$, $x_i = 1, i = 2{:}\ 10$	$f_2 = 1 - f_1^2$, $0 \leq f_1 \leq 1$	Concave, multimodal Nonseparable
		T_2	$x_1 \in [0,1]$, $(x_2, \ldots, x_{10})^T = s_{cm2}$	$f_1^2 + f_2^2 = 1$, $f_1 \geq 0, f_2 \geq 0$	Concave, unimodal Nonseparable
CILS	0.07	T_1	$x_1 \in [0,1]$, $x_i = 0, i = 2{:}\ 50$	$f_1^2 + f_2^2 = 1$, $f_1 \geq 0, f_2 \geq 0$	Concave, multimodal Separable
		T_2	$x_1 \in [0,1]$, $x_i = 0, i = 2{:}\ 50$	$f_2 = 1 - \sqrt{f_1}$, $0 \leq f_1 \leq 1$	Convex, multimodal Nonseparable
PIHS	0.99	T_1	$x_1 \in [0,1]$, $x_i = 0, i = 2{:}\ 50$	$f_2 = 1 - \sqrt{f_1}$, $0 \leq f_1 \leq 1$	Convex, unimodal Separable
		T_2	$x_1 \in [0,1]$, $(x_2, \ldots, x_{50})^T = s_{ph2}$	$f_2 = 1 - \sqrt{f_1}$ $0 \leq f_1 \leq 1$	Convex, multimodal Separable
PIMS	0.55	T_1	$x_1 \in [0,1]$, $(x_2, \ldots, x_{50})^T = s_{pm1}$	$f_1^2 + f_2^2 = 1$ $f_1 \geq 0, f_2 \geq 0$	Concave, unimodal Nonseparable
		T_2	$x_1 \in [0,1]$, $x_i = 0, i = 2{:}\ 50$	$f_2 = 1 - f_1^2$ $0 \leq f_1 \leq 1$	Concave, multimodal Nonseparable
PILS	0.002	T_1	$x_1 \in [0,1]$, $x_i = 0, i = 2{:}\ 50$	$f_1^2 + f_2^2 = 1$ $f_1 \geq 0, f_2 \geq 0$	Concave, multimodal Nonseparable
		T_2	$x_1 \in [0,1]$, $(x_2, \ldots, x_{50})^T = s_{pl2}$	$f_1^2 + f_2^2 = 1$ $f_1 \geq 0, f_2 \geq 0$	Concave, multimodal Nonseparable
NIHS	0.94	T_1	$x_1 \in [0,1]$, $x_i = 0, i = 2{:}\ 50$	$f_1^2 + f_2^2 = 1$ $f_1 \geq 0, f_2 \geq 0$	Concave, multimodal Nonseparable
		T_2	$x_1 \in [0,1]$, $x_i = 0, i = 2{:}\ 50$	$f_2 = 1 - \sqrt{f_1}$ $0 \leq f_1 \leq 1$	Convex, unimodal Separable
NIMS	0.51	T_1	$x_1 \in [0,1], x_2 \in [0,1]$, $\sum_{i=1}^{3} f_1^2 = 1$	$x_i = 1, i = 3{:}\ 20$ $f_i \geq 0, i = 1, 2, 3$	Concave, multimodal Nonseparable
		T_2	$x_1 \in [0,1], x_2 \in [0,1]$, $x_i = 0, i = 3{:}\ 20$	$f_2 = 1 - f_1^2$ $0 \leq f_1 \leq 1$	Concave, unimodal Nonseparable
NILS	0.001	T_1	$x_1 \in [0,1], x_2 \in [0,1]$, $(x_3, \ldots, x_{25})^T = s_{nl1}$	$\sum_{i=1}^{3} f_1^2 = 1$ $f_i \geq 0, i = 1, 2, 3$	Concave, multimodal Nonseparable
		T_2	$x_1 \in [0,1], x_2 \in [0,1]$, $x_i = 0, i = 3{:}\ 50$	$f_2 = 1 - f_1^2$ $0 \leq f_1 \leq 1$	Concave, multimodal Nonseparable

Table 4: The average and standard deviation of IGD values obtained by the proposed MSO algorithm, MO-MFEA, and NSGA-II. The best result in each row is shown in bold font.

Function	$Sim\ (T_1, T_2)$	Subsystem no.	IGD		
			MSO	MO-MFEA	NSGA-II
CIHS	0.97	T_1	**7.51*E*-05 ± 8.63*E*-06**	4.16*E*-04 ± 2.67*E*-05†	2.17*E*-03 ± 4.32*E*-04†
		T_2	**9.25*E*-05 ± 1.31*E*-05**	5.64*E*-04 ± 3.28*E*-05†	5.31*E*-03 ± 1.66*E*-04†
CIMS	0.52	T_1	**7.36*E*-03 ± 5.14*E*-04**	1.19*E*-02 ± 4.32*E*-02†	8.01*E*-01 ± 3.42*E*-02†
		T_2	**3.42*E*-03 ± 7.10*E*-03**	7.48*E*-03 ± 3.26*E*-02†	7.90*E*-02 ± 3.26*E*-02†
CILS	0.07	T_1	9.58*E*-03 ± 7.51*E*-02	**7.84*E*-04 ± 9.50*E*-05**†	8.06*E*-01 ± 2.35*E*-01†
		T_2	3.47*E*-04 ± 3.85*E*-05	**7.84*E*-04 ± 2.56*E*-06**	7.65*E*-04 ± 2.29*E*-06
PIHS	0.99	T_1	**7.12*E*-04 ± 2.74*E*-05**	4.16*E*-03 ± 3.24*E*-03†	7.35*E*-03 ± 4.26*E*-04†
		T_2	**5.17*E*-03 ± 6.30*E*-04**	5.37*E*-02 ± 7.71*E*-02†	5.17*E*-02 ± 8.92*E*-02†
PIMS	0.55	T_1	**3.28*E*-03 ± 3.52*E*-04**	5.19*E*-03 ± 7.42*E*-03	6.45*E*-03 ± 4.33*E*-03†
		T_2	**2.24*E*-02 ± 4.16*E*-03**	9.63*E*+01 ± 2.47*E*-00†	1.58*E*-01 ± 3.65*E*-00†
PILS	0.002	T_1	**1.85*E*-04 ± 1.10*E*-05**	5.32*E*-04 ± 1.15*E*-05	2.83*E*-04 ± 4.02*E*-04
		T_2	**3.32*E*-02 ± 1.07*E*-04**	8.54*E*-02 ± 2.33*E*-03	6.62*E*-01 ± 1.24*E*-04†
NIHS	0.94	T_1	**4.25*E*-02 ± 1.18*E*-02**	8.45*E*-00 ± 3.74*E*-02†	7.90*E*-01 ± 5.88*E*-01†
		T_2	**5.32*E*-05 ± 6.58*E*-06**	6.70*E*-04 ± 5.39*E*-04†	3.51*E*-04 ± 4.27*E*-04†
NIMS	0.51	T_1	**6.37*E*-02 ± 5.89*E*-03**	1.85*E*-01 ± 4.34*E*-01†	8.02*E*-01 ± 1.48*E*-01†
		T_2	**3.24*E*-03 ± 6.35*E*-04**	2.67*E*-02 ± 5.06*E*-02†	7.70*E*-02 ± 3.18*E*-02†
NILS	0.001	T_1	7.17*E*-03 ± 8.24*E*-04	**8.15*E*-04 ± 2.32*E*-05**†	9.06*E*-04 ± 3.11*E*-05†
		T_2	4.46*E*-00 ± 3.29*E*-01	**6.14*E*-01 ± 4.16*E*-04**†	6.54*E*-01 ± 4.23*E*-04†

sorting, it firstly develops job orders for the first procedure based on the list scheduling principle and then determines job sequences for the remaining procedures by a FIFO manner. For machine allocation, we use the rule of the first idle machine. The performance metric is based on the values of makespan, production cost, electricity consumption, water wastage, and processing cost, and the goal is to achieve the smallest values. The optimization results and manual scheduling results are summarized in Table 6.

It is seen from Table 6 that all optimization algorithms including the proposed MSO, MO-MFEA, and NSGA-II perform better than manual scheduling for all instances based on the smallest makespan value, the smallest production cost, the smallest electricity consumption value, the smallest water wastage value, and the smallest processing cost. The proposed MSO algorithm performs best for all of the instances except J4 and J8, for which MO-MFEA is the best because the approach itself is an excellent multitask evolutionary algorithm. Furthermore, out of 50 groups of data in Table 6, there are 49, 34, and 43 statistically significant differences between the proposed MSO and manual scheduling, the proposed MSO and MO-MFEA, and the proposed MSO and NSGA-II. Based on this result, the probability that the proposed MSO and compared algorithms are from the same distribution is low. The overall optimization results indicate that these optimization algorithms are high-efficiency ways, and the proposed MSO is the most promising approach for studied production scheduling with resource saving in textile printing and dyeing. The reason for the superior performance of the proposed MSO algorithm is that it increases interaction between subsystems and solution diversity by effectively using cross-subsystem migration.

To further investigate the potential of the proposed method in production scheduling with resource saving, saving ratio of each objective in each instance is shown in Table 7, which is got by, respectively, calculating the ratio of the values obtained by each optimization algorithm to the values obtained by manual scheduling. It is found from Table 7 that using optimization algorithms, the effect of production scheduling with resource saving including makespan, production cost, electricity consumption, water wastage, and processing cost is very significant for all instances. Furthermore, Figure 3 provides the visual presentation of mean saving ratio of all objectives of each instance. It is seen that for the proposed MSO algorithm, MO-MFEA, and NSGA-II, mean saving ratio is over 15%, 10%, and 6%, respectively, for the most of instances. Figure 4 provides the visual presentation of mean saving ratio of each objective for all instances. It is seen that for makespan, mean saving ratio is about 15%, 12%, and 6%, respectively, for the proposed MSO algorithm, MO-MFEA, and NSGA-II. For production cost, mean saving ratio is about 19%, 14%, and 7%, respectively, for the proposed MSO algorithm, MO-MFEA, and NSGA-II. For electricity consumption, mean saving ratio is about 15%, 13%, and 6%, respectively, for the proposed MSO algorithm, MO-MFEA, and NSGA-II. For water wastage, mean saving ratio is about 17%, 15%, and 8%, respectively, for the proposed MSO algorithm, MO-MFEA, and NSGA-II. For processing cost, mean saving ratio is about 17%, 15%, and 7%, respectively, for the proposed MSO algorithm, MO-MFEA, and NSGA-II. It indicates that for each performance metric for studied production scheduling with resource saving in textile printing and dyeing, the proposed MSO algorithm is better than other two compared algorithms. That is, the proposed MSO algorithm can not

Table 5: The number of jobs, procedures, and machines in each procedure, and the corresponding procedures of each instance in textile printing and dyeing.

Instance	Number of jobs	Number of procedures	Number of machines in each procedure	Procedures
J1	10	10	{2, 3, 4, 4, 3, 2, 3, 4, 2, 3}	Preparing–singeing–bleaching–setting–mercerizing–hot-melting–padding–stretching–shrinking–packaging
J2	10	7	{2, 4, 4, 3, 2, 3, 2}	Preparing–bleaching–dyeing–cleaning–dewatering–drying–packaging
J3	10	12	{2, 3, 3, 3, 2, 2, 3, 3, 2, 3, 2, 3}	Preparing–singeing–bleaching–setting–mercerizing–underprinting–printing–baking–steaming–stretching–shrinking–packaging
J4	20	10	{2, 4, 4, 3, 3, 2, 3, 3, 2, 2}	Preparing–bleaching–setting–mercerizing–dyeing–drying–singeing–stretching–shrinking–packaging
J5	20	12	{3, 2, 2, 3, 4, 4, 2, 3, 2, 3, 2, 4}	Preparing–singeing–bleaching–setting–mercerizing–underprinting–printing–baking–steaming–stretching–shrinking–packaging
J6	20	7	{2, 3, 4, 3, 2, 4, 2}	Preparing–bleaching–dyeing–cleaning–dewatering–drying–packaging
J7	30	10	{2, 3, 4, 2, 2, 4, 3, 3, 2, 3}	Preparing–singeing–bleaching–setting–mercerizing–hot-melting–padding–stretching–shrinking–packaging
J8	30	7	{2, 3, 3, 4, 4, 2, 4}	Preparing–bleaching–dyeing–cleaning–dewatering–drying–packaging
J9	30	12	{3, 2, 2, 4, 2, 4, 3, 4, 2, 3, 2, 2}	Preparing–singeing–bleaching–setting–mercerizing–underprinting–printing–baking–steaming–stretching–shrinking–packaging
J10	30	8	{2, 4, 4, 3, 2, 3, 4, 3}	Preparing–bleaching–dyeing–cleaning–dewatering–drying–shrinking–packaging

Table 6: Result comparisons between manual scheduling, the proposed MSO algorithm, MO-MFEA, and NSGA-II for the integrated production scheduling with resource saving.

Instance	Objective functions	Manual scheduling	Optimization algorithms		
			MSO	MO-MFEA	NSGA-II
J1	Makespan	$1.45E+03 \pm 2.12E+01^{\dagger}$	$\mathbf{1.22E+03 \pm 5.90E+01}$	$1.31E+03 \pm 2.86E+01$	$1.36E+03 \pm 6.74E+01$
	Production cost	$1.49E+06 \pm 5.34E+04^{\dagger}$	$\mathbf{9.98E+05 \pm 2.14E+03}$	$1.05E+06 \pm 5.71E+04^{\dagger}$	$1.40E+06 \pm 7.58E+04^{\dagger}$
	Electricity consumption	$6.11E+04 \pm 8.22E+02^{\dagger}$	$\mathbf{5.01E+04 \pm 3.27E+02}$	$5.35E+04 \pm 3.05E+02^{\dagger}$	$5.78E+04 \pm 3.65E+02^{\dagger}$
	Water wastage	$9.84E+03 \pm 5.37E+02^{\dagger}$	$\mathbf{8.15E+03 \pm 2.36E+01}$	$8.61E+03 \pm 4.49E+01^{\dagger}$	$9.25E+03 \pm 1.18E+01^{\dagger}$
	Processing cost	$1.41E+06 \pm 5.36E+04^{\dagger}$	$\mathbf{1.12E+06 \pm 1.75E+04}$	$1.24E+06 \pm 2.35E+04$	$1.29E+06 \pm 2.36E+04$
J2	Makespan	$7.85E+02 \pm 6.34E+01^{\dagger}$	$\mathbf{6.91E+02 \pm 4.32E+01}$	$7.05E+02 \pm 1.18E+01^{\dagger}$	$7.10E+02 \pm 4.16E+01^{\dagger}$
	Production cost	$3.31E+05 \pm 7.48E+03^{\dagger}$	$\mathbf{2.75E+05 \pm 7.78E+03}$	$3.03E+05 \pm 5.21E+03^{\dagger}$	$3.12E+05 \pm 5.42E+03^{\dagger}$
	Electricity consumption	$2.74E+04 \pm 4.02E+02^{\dagger}$	$\mathbf{2.29E+04 \pm 6.63E+02}$	$2.48E+04 \pm 3.16E+02^{\dagger}$	$2.59E+04 \pm 3.60E+02^{\dagger}$
	Water wastage	$4.05E+03 \pm 2.30E+01^{\dagger}$	$\mathbf{3.36E+03 \pm 9.17E+01}$	$3.56E+03 \pm 7.06E+01^{\dagger}$	$3.80E+03 \pm 8.82E+01^{\dagger}$
	Processing cost	$3.05E+05 \pm 6.32E+03^{\dagger}$	$\mathbf{2.51E+05 \pm 2.64E+03}$	$2.79E+05 \pm 4.23E+03^{\dagger}$	$2.91E+05 \pm 1.16E+03^{\dagger}$
J3	Makespan	$7.08E+02 \pm 5.31E+01^{\dagger}$	$\mathbf{5.87E+02 \pm 3.62E+01}$	$6.04E+02 \pm 7.18E+01^{\dagger}$	$6.50E+02 \pm 5.42E+01^{\dagger}$
	Production cost	$6.31E+05 \pm 8.55E+03^{\dagger}$	$\mathbf{5.04E+05 \pm 1.31E+03}$	$5.51E+05 \pm 5.56E+03^{\dagger}$	$5.71E+05 \pm 3.18E+03^{\dagger}$
	Electricity consumption	$7.54E+04 \pm 4.60E+02^{\dagger}$	$\mathbf{6.28E+04 \pm 3.47E+02}$	$6.32E+04 \pm 3.17E+02$	$6.95E+04 \pm 4.29E+02^{\dagger}$
	Water wastage	$9.31E+03 \pm 8.66E+01^{\dagger}$	$\mathbf{7.93E+03 \pm 6.36E+01}$	$8.19E+03 \pm 2.36E+01^{\dagger}$	$8.67E+03 \pm 8.15E+01^{\dagger}$
	Processing cost	$5.25E+05 \pm 4.14E+03^{\dagger}$	$\mathbf{4.35E+05 \pm 7.89E+03}$	$4.61E+05 \pm 4.42E+03^{\dagger}$	$4.73E+05 \pm 6.63E+03^{\dagger}$
J4	Makespan	$1.76E+03 \pm 7.12E+01^{\dagger}$	$1.53E+03 \pm 4.35E+01$	$\mathbf{1.51E+03 \pm 1.15E+01}$	$1.60E+03 \pm 7.24E+01$
	Production cost	$8.12E+05 \pm 8.56E+03^{\dagger}$	$7.05E+05 \pm 6.69E+03$	$\mathbf{7.01E+05 \pm 3.27E+03}$	$7.35E+05 \pm 3.28E+03^{\dagger}$
	Electricity consumption	$4.51E+04 \pm 6.10E+02^{\dagger}$	$3.94E+04 \pm 3.78E+02$	$\mathbf{3.85E+04 \pm 8.13E+02}$	$4.00E+04 \pm 6.12E+02$
	Water wastage	$8.05E+03 \pm 5.23E+01^{\dagger}$	$7.00E+03 \pm 1.47E+01$	$\mathbf{6.90E+03 \pm 9.74E+01}$	$7.41E+03 \pm 6.69E+01^{\dagger}$
	Processing cost	$5.37E+05 \pm 4.16E+03^{\dagger}$	$4.89E+05 \pm 4.58E+03$	$\mathbf{4.78E+05 \pm 6.32E+03}$	$4.91E+05 \pm 4.42E+03^{\dagger}$
J5	Makespan	$1.72E+03 \pm 1.59E+01^{\dagger}$	$\mathbf{1.44E+03 \pm 6.61E+01}$	$1.56E+03 \pm 1.18E+01$	$1.58E+03 \pm 8.15E+01^{\dagger}$
	Production cost	$9.96E+05 \pm 8.44E+03^{\dagger}$	$\mathbf{8.21E+05 \pm 3.64E+03}$	$8.76E+05 \pm 2.35E+03^{\dagger}$	$9.25E+05 \pm 6.64E+03^{\dagger}$
	Electricity consumption	$7.84E+04 \pm 3.24E+02^{\dagger}$	$\mathbf{6.70E+04 \pm 8.96E+02}$	$6.92E+04 \pm 4.43E+02^{\dagger}$	$7.30E+04 \pm 2.01E+02^{\dagger}$
	Water wastage	$1.22E+04 \pm 1.10E+02^{\dagger}$	$\mathbf{9.81E+03 \pm 7.78E+01}$	$9.99E+03 \pm 1.06E+01^{\dagger}$	$1.13E+04 \pm 3.68E+02^{\dagger}$
	Processing cost	$8.94E+05 \pm 2.36E+03^{\dagger}$	$\mathbf{7.44E+05 \pm 5.93E+03}$	$7.63E+05 \pm 4.45E+03^{\dagger}$	$8.32E+05 \pm 1.17E+03^{\dagger}$
J6	Makespan	$1.07E+03 \pm 4.41E+01$	$\mathbf{1.01E+03 \pm 1.74E+01}$	$1.04E+03 \pm 7.78E+01$	$1.06E+03 \pm 5.24E+01$
	Production cost	$2.96E+05 \pm 8.39E+03^{\dagger}$	$\mathbf{2.49E+05 \pm 2.36E+03}$	$2.65E+05 \pm 5.36E+03^{\dagger}$	$2.75E+05 \pm 3.32E+03^{\dagger}$
	Electricity consumption	$2.62E+04 \pm 7.45E+02^{\dagger}$	$\mathbf{2.15E+04 \pm 4.47E+02}$	$2.32E+04 \pm 1.13E+02^{\dagger}$	$2.48E+04 \pm 4.40E+02^{\dagger}$
	Water wastage	$4.21E+03 \pm 8.36E+01^{\dagger}$	$\mathbf{3.29E+03 \pm 3.62E+01}$	$3.66E+03 \pm 7.58E+01^{\dagger}$	$4.01E+03 \pm 3.65E+01^{\dagger}$
	Processing cost	$1.56E+05 \pm 9.60E+03^{\dagger}$	$\mathbf{1.29E+05 \pm 1.18E+03}$	$1.42E+05 \pm 6.31E+03^{\dagger}$	$1.47E+05 \pm 2.47E+03^{\dagger}$
J7	Makespan	$1.71E+03 \pm 2.36E+01^{\dagger}$	$\mathbf{1.45E+03 \pm 6.35E+01}$	$1.51E+03 \pm 6.35E+01$	$1.62E+03 \pm 1.29E+01^{\dagger}$
	Production cost	$1.11E+06 \pm 6.03E+04^{\dagger}$	$\mathbf{9.31E+05 \pm 9.68E+03}$	$9.79E+05 \pm 4.47E+03^{\dagger}$	$1.02E+06 \pm 4.33E+04^{\dagger}$
	Electricity consumption	$3.67E+04 \pm 5.60E+02^{\dagger}$	$\mathbf{3.10E+04 \pm 7.41E+02}$	$3.23E+04 \pm 1.18E+02^{\dagger}$	$3.45E+04 \pm 9.54E+02^{\dagger}$
	Water wastage	$1.18E+04 \pm 8.15E+02^{\dagger}$	$\mathbf{9.73E+03 \pm 3.26E+01}$	$9.98E+03 \pm 5.69E+01^{\dagger}$	$1.03E+04 \pm 6.37E+02^{\dagger}$
	Processing cost	$1.16E+06 \pm 7.34E+04^{\dagger}$	$\mathbf{9.60E+05 \pm 5.59E+03}$	$9.98E+05 \pm 6.37E+03^{\dagger}$	$1.07E+06 \pm 2.50E+04^{\dagger}$
J8	Makespan	$1.53E+03 \pm 8.56E+01^{\dagger}$	$1.32E+03 \pm 5.23E+01$	$\mathbf{1.30E+03 \pm 2.28E+01}$	$1.42E+03 \pm 4.31E+01$
	Production cost	$4.16E+05 \pm 2.33E+03^{\dagger}$	$3.58E+05 \pm 1.82E+03$	$\mathbf{3.51E+05 \pm 1.25E+03}$	$3.82E+05 \pm 7.89E+03^{\dagger}$
	Electricity consumption	$5.53E+04 \pm 6.34E+02^{\dagger}$	$4.87E+04 \pm 4.34E+02$	$\mathbf{4.68E+04 \pm 4.13E+02}^{\dagger}$	$5.35E+04 \pm 1.12E+02^{\dagger}$
	Water wastage	$7.62E+03 \pm 7.79E+01^{\dagger}$	$6.71E+03 \pm 8.29E+01$	$\mathbf{6.45E+03 \pm 8.75E+01}^{\dagger}$	$7.02E+03 \pm 5.36E+01^{\dagger}$
	Processing cost	$3.91E+05 \pm 4.18E+03^{\dagger}$	$3.37E+05 \pm 1.44E+03$	$\mathbf{3.30E+05 \pm 5.36E+03}$	$3.63E+05 \pm 8.92E+03^{\dagger}$
J9	Makespan	$1.56E+03 \pm 6.40E+01^{\dagger}$	$\mathbf{1.33E+03 \pm 2.54E+01}$	$1.36E+03 \pm 1.23E+01$	$1.40E+03 \pm 1.23E+01$
	Production cost	$2.28E+06 \pm 8.23E+04^{\dagger}$	$\mathbf{1.78E+06 \pm 1.98E+04}$	$1.92E+06 \pm 7.55E+04^{\dagger}$	$2.02E+06 \pm 5.51E+04^{\dagger}$
	Electricity consumption	$4.92E+04 \pm 5.26E+02^{\dagger}$	$\mathbf{4.30E+04 \pm 6.65E+02}$	$4.37E+04 \pm 1.34E+02$	$4.46E+04 \pm 4.33E+02^{\dagger}$
	Water wastage	$1.98E+04 \pm 7.41E+02^{\dagger}$	$\mathbf{1.66E+04 \pm 1.27E+02}$	$1.77E+04 \pm 9.25E+02^{\dagger}$	$1.75E+04 \pm 7.75E+02^{\dagger}$
	Processing cost	$1.89E+06 \pm 2.35E+04^{\dagger}$	$\mathbf{1.52E+06 \pm 3.65E+04}$	$1.58E+06 \pm 3.11E+04$	$1.72E+06 \pm 8.24E+04^{\dagger}$
J10	Makespan	$2.21E+03 \pm 5.01E+01^{\dagger}$	$\mathbf{1.83E+03 \pm 4.42E+01}$	$2.03E+03 \pm 7.42E+01^{\dagger}$	$2.19E+03 \pm 5.61E+01^{\dagger}$
	Production cost	$3.30E+05 \pm 8.42E+03^{\dagger}$	$\mathbf{2.74E+05 \pm 1.12E+03}$	$3.02E+05 \pm 6.35E+03^{\dagger}$	$3.29E+05 \pm 8.76E+03^{\dagger}$
	Electricity consumption	$4.77E+04 \pm 7.41E+02^{\dagger}$	$\mathbf{3.95E+04 \pm 7.14E+02}$	$4.19E+04 \pm 8.17E+02^{\dagger}$	$4.67E+04 \pm 5.32E+02^{\dagger}$
	Water wastage	$9.21E+03 \pm 8.56E+01^{\dagger}$	$\mathbf{7.66E+03 \pm 8.23E+01}$	$8.11E+03 \pm 3.62E+01^{\dagger}$	$8.26E+03 \pm 9.01E+01^{\dagger}$
	Processing cost	$2.67E+05 \pm 9.62E+03^{\dagger}$	$\mathbf{2.23E+05 \pm 3.31E+03}$	$2.44E+05 \pm 4.28E+03^{\dagger}$	$2.66E+05 \pm 5.27E+03^{\dagger}$

only obtain better scheduling strategy but also reduce more electricity consumption and water wastage.

These results show that the proposed integrated optimization model and the corresponding solutions have a great potential in production scheduling with resource savings for textile printing and dyeing industry. Many manufacturing industries, such as papermaking manufacturing and glass manufacturing, are both energy-intensive and material-intensive. Production in these manufacturing often wastes a lot of energy and material, so

Table 7: Saving ratio comparisons between manual scheduling, the proposed MSO algorithm, MO-MFEA, and NSGA-II for the integrated production scheduling with resource saving.

Instance	Objective functions	Saving ratio (%)		
		MSO	MO-MFEA	NSGA-II
J1	Makespan	15.86	9.66	6.21
	Production cost	33.02	29.53	6.04
	Electricity consumption	18.00	12.44	5.40
	Water wastage	17.17	12.50	6.00
	Processing cost	20.57	12.06	8.51
J2	Makespan	11.97	11.08	9.55
	Production cost	16.92	8.46	5.74
	Electricity consumption	16.42	12.04	5.47
	Water wastage	17.04	12.10	6.17
	Processing cost	17.70	11.80	4.59
J3	Makespan	17.09	14.69	8.19
	Production cost	20.13	12.68	9.51
	Electricity consumption	16.71	16.18	7.82
	Water wastage	14.82	12.03	6.87
	Processing cost	17.14	12.19	9.90
J4	Makespan	13.07	14.20	9.09
	Production cost	13.18	13.67	9.48
	Electricity consumption	12.64	14.63	11.31
	Water wastage	13.04	14.29	7.95
	Processing cost	8.94	10.99	8.58
J5	Makespan	16.28	11.63	8.14
	Production cost	17.57	12.05	7.13
	Electricity consumption	14.54	11.73	6.89
	Water wastage	19.59	19.36	7.38
	Processing cost	16.78	14.65	6.94
J6	Makespan	5.61	2.80	0.93
	Production cost	15.88	10.47	7.09
	Electricity consumption	17.94	11.45	5.34
	Water wastage	21.85	13.06	4.75
	Processing cost	17.31	15.38	5.77
J7	Makespan	15.20	11.70	5.26
	Production cost	16.13	11.80	8.11
	Electricity consumption	15.53	11.99	5.99
	Water wastage	17.54	15.42	12.71
	Processing cost	17.24	13.97	7.76
J8	Makespan	13.73	15.03	7.19
	Production cost	13.94	15.63	8.17
	Electricity consumption	13.74	15.37	3.25
	Water wastage	14.57	15.35	7.87
	Processing cost	13.81	15.60	7.16
J9	Makespan	14.74	12.82	10.26
	Production cost	21.93	15.79	11.40
	Electricity consumption	12.60	11.18	9.35
	Water wastage	16.16	10.61	11.62
	Processing cost	19.58	16.40	8.99
J10	Makespan	17.19	12.67	0.90
	Production cost	16.67	8.48	0.30
	Electricity consumption	17.19	12.16	2.10
	Water wastage	16.83	11.94	10.31
	Processing cost	16.48	12.36	0.37

the decision makers tend to reduce them during the production process. This study can serve as a reference for these manufacturers who are interested in improving energy and material savings through production scheduling, and the proposed MSO algorithm can be an attractive method for solving these complex multisystem problems.

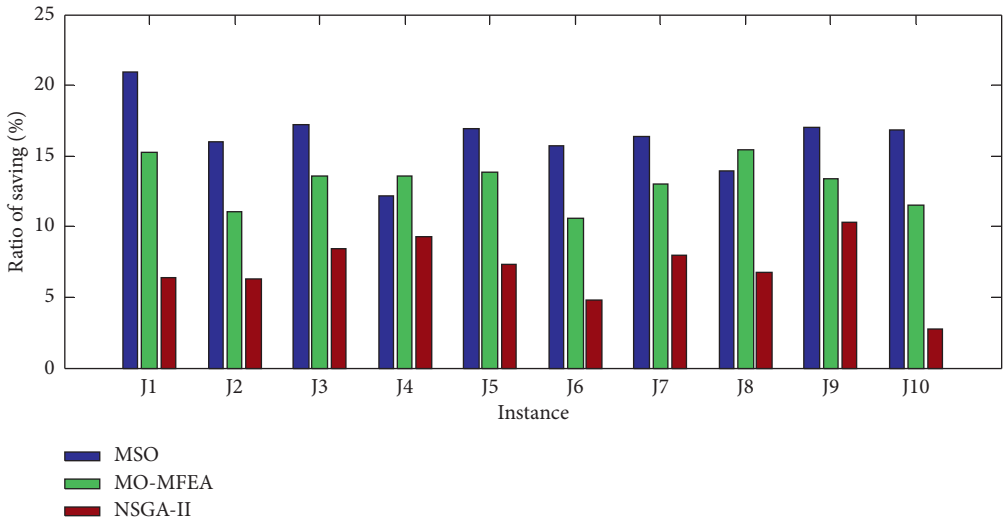

Figure 3: The mean saving ratio of all objectives of each instance obtained by the proposed MSO algorithm, MO-MFEA, and NSGA-II for the integrated production scheduling with resource saving.

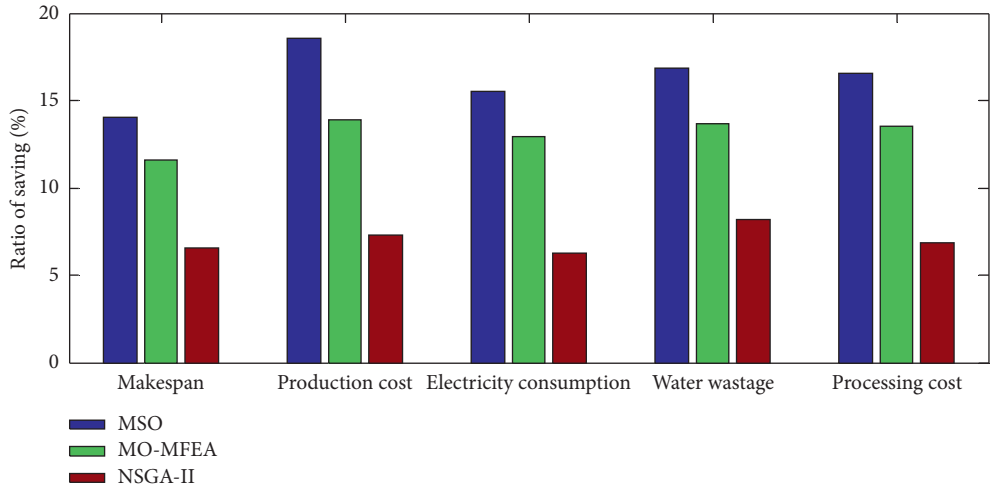

Figure 4: The mean saving ratio of each objective for all instances obtained by the proposed MSO algorithm, MO-MFEA, and NSGA-II for the integrated production scheduling with resource saving.

6. Conclusions

In this paper, an integrated optimization model for textile printing and dyeing is first built by considering both production efficiency and resource saving, which is formulated as a multisystem optimization problem. It includes a production scheduling subsystem and a resource saving subsystem, and each of the subsystems contains multiple objectives. Then, a multisystem optimization algorithm called the MSO algorithm, composed of within-subsystem evolution and cross-subsystem migration operators, is proposed to solve general multisystem optimization problem. The performance of the MSO algorithm is investigated on a set of benchmark functions, and the numerical simulations show that the proposed MSO algorithm is better than MO-MFEA and NSGA-II for the most of the benchmark functions.

Finally, the MSO algorithm is applied to the proposed integrated production scheduling with resource saving problem, and the results again show that the proposed MSO algorithm is a competitive multisystem optimization method. In particular, the MSO algorithm is better than its constituent NSGA-II on all the test instances, and it is better than MO-MFEA on 8 of 10 instances.

The proposed MSO algorithm is a heuristic method for complex multisystem optimization. The complex multisystem contains multiple subsystems, and each of the subsystems may contain multiple objectives and multiple constraints. Compared to single-objective and multiobjective optimization algorithms, the proposed MSO algorithm has a more complex structure that is well-suited for real-world problems. Based on the optimization tests in this paper, the MSO algorithm can effectively solve multisystem optimization problems, including real-world textile printing and dyeing scheduling with resource saving. The MSO framework presented here could be extended for other types of optimization algorithms. Moreover, real-world textile

printing and dyeing scheduling problem with resource saving is formulated in a way that is amenable to the multisystem optimization algorithm, which serves as a reference for other manufacturing with similar production processes to promote in resource saving.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the Zhejiang Provincial Natural Science Foundation of China under grant no. LY19F030011, National Natural Science Foundation of China under grant nos. 52077213 and 62003332, and Natural Science Foundation of Guangdong Province under grant no. 2018A030310671. H. Zhou was supported by the UK EPSRC under grant no. EP/N011074/1, Royal Society-Newton Advanced Fellowship under grant no. NA160342, and European Union's Horizon 2020 Research and Innovation Program under the Marie-Sklodowska-Curie grant agreement no. 720325.

References

[1] Y. Lu, T. Peng, and X. Xu, "Energy-efficient cyber-physical production network: Architecture and technologies," *Computers & Industrial Engineering*, vol. 129, pp. 56–66, 2019.

[2] M. Vieira, T. Pinto-Varela, and A. P. Barbosa-Póvoa, "A model-based decision support framework for the optimisation of production planning in the biopharmaceutical industry," *Computers & Industrial Engineering*, vol. 129, pp. 354–367, 2019.

[3] S. Ye, H. Ma, S. Xu et al., "An effective fireworks algorithm for warehouse-scheduling problem," *Transactions of the Institute of Measurement and Control*, vol. 39, no. 1, pp. 75–85, 2017.

[4] F. Li, L. Zhou, G. Xu et al., "An empirical study on solving an integrated production and distribution problem with a hybrid strategy," *PloS One*, vol. 13, no. 11, Article ID e0206806, 2017.

[5] H. Ma, S. Su, D. Simon, and M. Fei, "Ensemble multi-objective biogeography-based optimization with application to automated warehouse scheduling," *Engineering Applications of Artificial Intelligence*, vol. 44, pp. 79–90, 2015.

[6] Z. Zeng, M. Hong, Y. Man, J. Li, Y. Zhang, and H. Liu, "Multi-object optimization of flexible flow shop scheduling with batch process - consideration total electricity consumption and material wastage," *Journal of Cleaner Production*, vol. 183, pp. 925–939, 2018.

[7] M. Biondi, G. Sand, and I. Harjunkoski, "Optimization of multipurpose process plant operations: a multi-time-scale maintenance and production scheduling approach," *Computers & Chemical Engineering*, vol. 99, pp. 325–339, 2017.

[8] L. S. Dias, R. C. Pattison, C. Tsay, M. Baldea, and M. G. Ierapetritou, "A simulation-based optimization framework for integrating scheduling and model predictive control, and its application to air separation units," *Computers & Chemical Engineering*, vol. 113, pp. 139–151, 2018.

[9] J. Allison, *Complex System Optimization: A Review of Analytical Target Cascading, Collaborative Optimization, and Other Formulations*, M. S. Thesis, University of Michigan, Ann Arbor, MI, USA, 2004.

[10] J. Abell and D. Du, "A framework for multi-objective, biogeography-based optimization of complex system families," in *Proceedings of the AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*, pp. 1–8, Fort Worth, TX, USA, 2010.

[11] J. R. R. A. Martins and A. B. Lambe, "Multidisciplinary design optimization: a survey of architectures," *AIAA Journal*, vol. 51, no. 9, pp. 2049–2075, 2013.

[12] K. Li, K. Deb, Q. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 5, pp. 694–716, 2015.

[13] Q. Zhang and H. Li, "MOEA/D: a multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[14] L. M. Antonio and C. A. Coello, "Coevolutionary multi-objective evolutionary algorithms: a survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 6, pp. 851–865, 2018.

[15] A. Gupta, Y. S. Ong, L. Feng et al., "Multiobjective multifactorial optimization in evolutionary multitasking," *IEEE Transactions on Cybernetics*, vol. 47, no. 7, pp. 1652–1665, 2016.

[16] Y.-S. Ong and A. Gupta, "Evolutionary multitasking: a computer science view of cognitive multitasking," *Cognitive Computation*, vol. 8, no. 2, pp. 125–142, 2016.

[17] D. Du and D. Simon, "Complex system optimization using biogeography-based optimization," *Mathematical Problems in Engineering*, vol. 8, Article ID 456232, , 2013.

[18] D. Simon, "Biogeography-based optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 6, pp. 702–713, 2008.

[19] H. Ma, M. Fei, Z. Jiang, L. Li, H. Zhou, and D. Crookes, "A multipopulation-based multiobjective evolutionary algorithm," *IEEE Transactions on Cybernetics*, vol. 50, no. 2, pp. 689–702, 2020.

[20] C. Li, T. T. Nguyen, M. Yang, M. Mavrovouniotis, and S. Yang, "An adaptive multipopulation framework for locating and tracking multiple optima," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 4, pp. 590–605, 2016.

[21] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[22] R. J. Hathaway and J. C. Bezdek, "Fuzzy c-means clustering of incomplete data," *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 31, no. 5, pp. 735–744, 2001.

[23] Y. Yuan, Y. S. Ong, L. Feng et al., "Evolutionary multitasking for multiobjective continuous optimization: benchmark problems, performance metrics and baseline results," Technical Report, pp. 1–10, Cornell University, Ithaca, NY, USA, 2016.

[24] Z. Zhao, B. Liu, C. Zhang, and H. Liu, "An improved adaptive NSGA-II with multi-population algorithm," *Applied Intelligence*, vol. 49, no. 2, pp. 569–580, 2019.