Int j simul model 22 (2023) 2, 315-326
Original scientific paper

OPTIMIZATION OF A SIMULATED RECONFIGURABLE HYBRID FLOW ASSEMBLY SYSTEM

Liang, R.[*,**]; Feng, Z. X.[*,**]; Dong, Y. F.[*,**]; Xin, Y.[*,**]; Yang, J. X.[*,**] & Cao, Y. P.[*,**,#]

[*] State Key Laboratory of Fluid Power and Mechatronic Systems, School of Mechanical Engineering, Zhejiang University, Hangzhou, 310027, China

[**] Key Laboratory of Advanced Manufacturing Technology of Zhejiang Province, School of Mechanical Engineering, Zhejiang University, Hangzhou, 310027, China

E-Mail: caoyp@zju.edu.cn ([#] Corresponding author)

Abstract

The existing research on hybrid flowshop scheduling (HFS) has typically neglected important factors such as workload balance and storage costs. Therefore, in this study, we simulate a reconfigurable hybrid flow assembly (RHFA) system, and propose five significant objective functions that consider the above factors, designed to attain a joint optimization of equipment composition and assembly sequence. To solve this multi-objective optimization problem, the multi-objective Harris hawks optimization (MOHHO) method is adopted to generate optimization solutions. Results demonstrate that MOHHO outperforms other alternatives in terms of generating more dominant solutions and achieving better evaluation results on several representative indicators, including mean ideal distance, maximum spread, and spacing metric. Given empirical evidence of a hybrid flow assembly workshop, the research outcomes hold significant implications for the optimization of assembly sequences and equipment layout. The findings presented herein can assist decision-makers in devising more informed and rational production plans, thereby enhancing production efficiency and reducing associated costs.

(Received in January 2023, accepted in April 2023. This paper was with the authors 1 month for 2 revisions.)

Key Words: Simulated RHFA System, Assembly Sequence, Equipment Composition, Multi-Objective Harris Hawks Optimization

1. INTRODUCTION

In traditional production and manufacturing mode, enterprises can obtain economic benefits by repeatedly producing products with fixed or single functions in large quantities. However, with the strengthening of globalization trend in manufacturing industry, manufacturing enterprises are fiercely competing with each other under diversified and unpredictable market demands. As a result, traditional mass-specific manufacturing mode dominated by products is no longer appropriate for current developing requirement [1]. Instead, it is desirable to develop a modern mass-customized manufacturing mode which is dominated by demands of customers and can well handle some important manufacturing requirements, such as small batch, fast, flexible and diversified, etc. In order to satisfy this mode, modern manufacturing systems need to have the ability to dynamically and flexibly adjust production systems according to various requirements [2]. The early flexible manufacturing system (FMS) can realize a high degree of automatic adjustment of entire manufacturing process and can adapt to production of multiple varieties and different batches to rapidly respond to changes in customer's demands for products. However, satisfying demands for multiple varieties of products will increase complexity of equipment, causing complicated and superfluous system functions and undesired waste of resources. Moreover, high flexibility of a manufacturing system will inevitability incur extra purchase and installation costs and increase investment risk of enterprises [3, 4]. With the increasing demand for individuation in process of market development and progress of modern manufacturing technology, reconfigurable manufacturing system (RMS) came into being, which combines benefits of traditional dedicated manufacturing system (DMS) and flexible manufacturing system (FMS) [5]. The concept of RMS was first proposed in 1997 by Koren et al. [1]. They argued that manufacturing companies in the 21st century must have the ability to swiftly respond to unpredicted and high-frequency market changes to adapt to challenges brought by global competition. RMS provides a brand-new significant manufacturing technology that can effectively help manufacturing enterprises survive in increasingly fierce global market competition environment.

Assembly is the last link in product manufacturing and usually takes up more than half of time consumed in the whole manufacturing process [6]. Setchi and Lagos [7] pointed out that reconfigurable assembly lines can realize assembly of multiple-varieties products, and can effectively increase productivity by means of replication and modularization. With increasing market competition and a pressing need for enterprises to improve production efficiency, more and more manufacturing firms are gradually introducing hybrid flow assembly systems, which integrate additional branches or parallel workstations to improve production efficiency and reduce production costs. It is noted that some parallel equipment exists in the hybrid flow assembly line, and the differences in assembly costs and assembly capacity caused by types of equipment will affect production scheduling results. RHFA line is further developed on the basis of conventional hybrid flow assembly line, and production capacity of the whole assembly line can be adjusted flexibly at any time since equipment can be reconfigured.

In recent years, numerous researchers have paid attention to hybrid flowshop scheduling (HFS) problems. Naderi et al. [8] adopted a novel simulated annealing (SA) algorithm to solve HFS problems, aiming at minimizing total completion time and total tardiness. Marichelvam et al. [9] took manufacturing time and average flow time as objective functions and used discrete firefly algorithm to achieve multi-objective optimization. Hosseini [10] used multi-objective genetic algorithm (MOGA) to solve HFS problems for two-stage production systems with the goal of minimizing makespan and the sum of earliness and tardiness of products. Zheng et al. [11] studied a multi-objective fuzzy distributed hybrid flowshop scheduling problem and proposed a cooperative coevolution algorithm with problem-specific strategies to simultaneously optimize fuzzy total tardiness and robustness. Xu et al. [12] provided a succinct overview of the modelling technique for the scheduling problem, developed a mathematical model aimed at minimizing the maximum completion time in a HFS scenario, and employed a genetic algorithm as a means of resolving the problem. Jemmali et al. [13] investigated a two-stage HFS problem characterized by independent setup times, and devised a genetic algorithm with tailored attributes aimed at identifying a viable schedule that minimizes the maximum completion time. In summary, HFS problems have been traditionally approached with the objective of minimizing makespan, manufacturing time, and total tardiness. However, the practical impact of storage costs for raw materials and end-products, as well as the issue of workload imbalance during production, have been overlooked. Workload imbalance can severely hamper assembly efficiency and increase assembly costs, and overemphasizing other objectives while neglecting storage costs may also lead to additional production costs. Furthermore, it is worth mentioning that limited research works have been done on HFS problem under reconfigurable setting. In the context of the RHFA problem, in addition to the aforementioned concerns, restraining the costs associated with reconfiguration and assembly are critical considerations that must be addressed in the production process. Therefore, in this paper, we direct our attention toward the pragmatic challenges of the issue on RHFA scheduling and place particular emphasis on workload balance of assembly processes, workload balance of assembly lines, reconfiguration and assembly costs, storage costs of raw materials, and warehousing costs of end-products in the context of RHFA.

In general, production/assembly sequence of products and layout of equipment facilities are significant factors affecting overall costs. However, these factors are discrete and can only be selected from specific sets, posing a challenge for traditional optimization methods like gradient descent and Gaussian-Newton algorithms that require computation of partial derivatives, making them unsuitable for discrete optimization problems. To address multi-objective optimization problems, we have employed the multi-objective Harris hawks optimization (MOHHO) method, a meta-heuristic algorithm that combines the benefits of multi-objective particle swarm optimization (MOPSO) and Harris hawks optimization (HHO) method, in generating solutions which dominate other alternatives.

In this paper, we simulate a RHFA system oriented toward multi-variety and small-batch production, which is conducive to shortening product production cycles, lowering production costs, and embodying flexible manufacturing characteristics. Given practical RHFA scenarios, our motivation is to attain a joint optimization of assembly equipment composition and product assembly sequence by minimizing the aforementioned five objectives. In order to solve multi-objective optimization problems, a MOHHO algorithm is adopted to generate optimization solutions for the problem presented in this paper. The results demonstrate that MOHHO algorithm can obtain more competitive results than the alternatives.

The structure of this paper is arranged as follows. In the first section, recent research advancements pertaining to RMS and HFS problems are reviewed. Subsequently, problem formation and methodology are presented in Section 2. Section 3 outlines the simulation for a RHFA system, and provides a comparison with the alternatives to highlight the superiority of MOHHO algorithm for solving the problem presented in this paper. Finally, Section 4 summarizes full text and outlines directions for future research.

2. PROBLEM FORMULATION AND METHODOLOGY

2.1 Problem formulation

Throughout the paper, some significant variables and notations are defined as follows:

i – type index of products to be assembled ($i \in I$)
j – assembly process index ($j \in J$)
k – index of hybrid flow assembly line ($k \in K$)
b – assembly batch index ($b \in B$)
m – machine's position index in each assembly process ($m \in M$)
s – machine's index in each assembly process
AM_{jm} – machine's position number with index m in assembly process j
MA_{js} – machine's number with index s in assembly process j

In this paper, we consider a RHFA line with $|J|$ assembly processes. Assembly processes are arranged from AP_1 to $AP_{|J|}$. Here we consider that each assembly process has multiple machines to select. These machines are functionally similar, but differ in price, assembly capacity, and assembly costs. Given that assembly process may be limited by size/layout of the assembly venue and some other factors (e.g., logistics complexity and manpower resources), it is practically difficult to accommodate all the above machines with similar functionalities and assemble all products simultaneously. In this paper, we consider that $|M|$ alternative machines are deployed in each assembly process for reconfigurable selection. As shown in Fig. 1, we use P to represent products to be assembled (the subscript i distinguishes the type of different products), and different types of products are divided into $|B|$ batches for assembly tasks. Given that our assembly system has the ability to simultaneously assemble $|M|$ different types of products in each batch ($|M|$ machines are deployed in each assembly process), then products in different batches will be assembled in sequence through $|J|$ individual assembly processes (AP_1 to $AP_{|J|}$) and products in the same batch will be arranged for assembly on $|K|$ (in this paper $|K|=|M|$) hybrid flow assembly lines. This process involves some important optimization questions including how to arrange different types of products into multiple batches, which machines should be selected in each assembly process for a certain batch, how machines in a certain assembly process are reconfigured for assembling different batches, etc. The questions mentioned above are optimization goals in this paper.

Note that our RHFA model is based on the following assumptions or premises.

(1) Assembly process flow is pre-designed, and machines available in each assembly process as well as their assembly capacity and purchase costs are known.

(2) All products to be assembled are of similar attributes and thus go through the same assembly processes.

(3) All products to be assembled need to go through all $|J|$ assembly processes, and after completion of assembly work in current process, products flow to the next process in sequence.

(4) $|M|$ machines in individual assembly processes work independently and they assemble $|M|$ different types of products in each batch respectively.

(5) Raw materials required for assembly have been pre-prepared and logistics costs between different assembly processes is negligible compared with assembly costs.

(6) Assembly tasks for the next batch of products only starts after assembly for the previous ones is completed, and there is no interruption in assembly processes of all products.

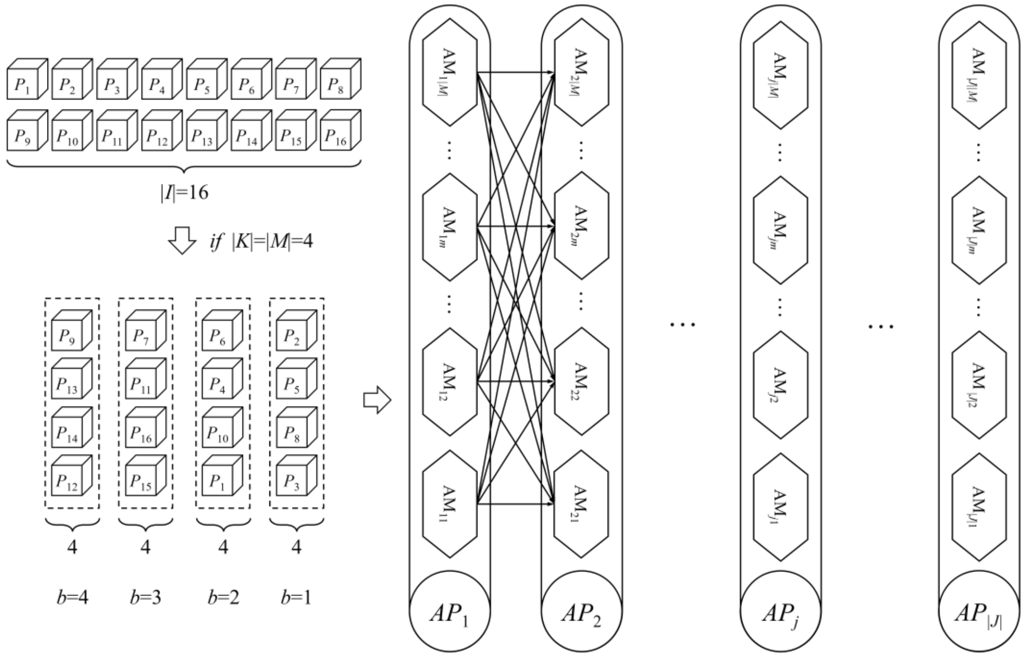

Figure 1: Diagram of the proposed RHFA system.

Then based on the aforementioned assembly model and combined with the practical production demands, five objective functions are emphatically considered.

(1) Workload balance of assembly processes

$$F_1 = \min \sum_{b\in B}\sum_{k\in K}\sum_{i\in I} \frac{\beta_{bik} N_{bik}}{N} (T_{bik})_{\max} \sum_{j\in J} \left| \frac{T_{bijk} - (T_{bik})_{\text{avg}}}{(T_{bik})_{\text{avg}}} \right| \tag{1}$$

Eq. (1) is used to ensure the workload balance on assembly line as much as possible. T_{bijk} denotes the assembly time of each machine. $(T_{bik})_{\text{avg}}$ and $(T_{bik})_{\max}$ represent the average assembly time and the bottleneck time of machines in one assembly line of the system, respectively. β_{bik} determines whether products i are assembled in batch b using assembly line k (if yes, $\beta_{bik} = 1$, otherwise $\beta_{bik} = 0$). N_{bik} and N describe the quality of product i assembled in batch b using assembly line k and the total quantity of products to assemble, respectively.

(2) Workload balance of assembly lines

$$F_2 = \min \sum_{b\in B} \left| (TS_b)_{\max} - (TS_b)_{\min} \right| \tag{2}$$

Eq. (2) calculates the difference between the longest time $(TS_b)_{\max}$ consumed by an assembly line and the shortest time $(TS_b)_{\min}$ consumed by another one in the same batch b to optimize the working time balance of assembly lines.

(3) Reconfiguration and assembly costs

$$F_3 = \min\left(\sum_{b\in B\setminus\{|B|\}} \sum_{m\in M} \sum_{j\in J} \gamma_{bjm} \left(\varphi_d C_{bjm} + \varphi_i C_{(b+1)jm} \right) + \sum_{b\in B} \sum_{k\in K} \sum_{i\in I} \beta_{bik} \sum_{j\in J} A_{bijk} N_{bik} \right) \tag{3}$$

Eq. (3) describes the sum of reconfiguration costs and assembly costs. $\gamma_{bjm} = 0$ means no reconfiguration is required, while $\gamma_{bjm} = 1$ means reconfiguration is necessary. φ_d and φ_i represent the dismantling coefficient and installation coefficient, respectively. C_{bjm} denotes purchase cost of machine with index m in process j in batch b, and A_{bijk} denotes the unit cost for assembling product i in process j in batch b using assembly line k.

(4) Storage costs of raw materials

$$F_4 = \min \sum_{b\in B\setminus\{|B|\}} \left((TS_b)_{\max} \sum_{u\in\{b+1,\ldots,|B|\}} \sum_{k\in K} \sum_{i\in I} \beta_{uik} \delta_i \right) \tag{4}$$

Eq. (4) calculates storage costs of raw materials of all batches except the first one. The storage factor δ_i is defined to distinguish the storage conditions of raw materials required for different types of products.

(5) Warehousing costs of end-products

$$F_5 = \min \sum_{b\in B\setminus\{1\}} \left((TS_b)_{\max} \sum_{v\in\{1,\ldots,b-1\}} \sum_{k\in K} \sum_{i\in I} \beta_{vik} \lambda_i \right) \tag{5}$$

Eq. (5) describes warehousing costs of end-products of all batches except the last one. The end-product storage factor λ_i is defined to distinguish the storage conditions of different types of end-products.

2.2 Optimization method

We aim to optimize batch sequence of products to assemble and equipment composition of production line for assembling different types of products, which represents a complicated multi-objective optimization problem in which five individual objective functions are required to be optimized simultaneously. In this paper, we employ a MOHHO algorithm to effectively address the above multi-objective optimization problem.

Harris hawks optimization (HHO) algorithm, a population-based gradientless optimization method, was first proposed by Heidari et al. [14]. It is expected to be widely applied to many optimization problems and provide relatively competitive performance with fewer/simpler parameter settings. HHO algorithm was originally inspired by cooperative and chasing patterns of Harris hawks. Specifically, the method divides Harris hawks' predation into three individual phases, including exploration, transition from exploration to exploitation, and exploitation. In the exploration phase, Harris hawks may search for prey based on the location of other family members and a random perching location in the range of their group. Depending on the escaping energy of prey, transition between different exploitation behaviours can be implemented during the transition phase from exploration to exploitation, and in this phase, the energy of prey decreases with escaping. In the exploitation phase, Harris hawks pounce on the target prey they found in the previous phase. According to [14], four strategies are employed to simulate behaviours in this phase based on the escaping pattern of the prey and the chasing strategies of Harris hawks.

Confronted with complicated multi-objective optimization problems, it is promising to combine other evolutionary schemes with HHO algorithm to give full play to advantages of both. Here we adopt MOHHO algorithm to solve the proposed multi-objective optimization problem. The overall steps of MOHHO algorithm are described in detail as follows:

Step 1. Randomly generate an initial population, initialize individual location and calculate its fitness value.

Step 2. Archive all non-dominated solutions after sifting through population non-dominated solutions.

Repeat *steps 3-5* until the number of algorithm iterations reaches the maximum value set in advance.

Step 3. Update the location in iteration process and repeat *substeps a-d* until the maximum population is satisfied.

a. Set optimal individual location as current prey location.

b. Calculate random value of the initial state of prey's energy E_0, and use the formula of transition phase to update the escaping energy E.

c. Determine relationship between the escaping energy $|E|$ and 1. If $|E| \geq 1$, use the formula of exploration phase to update the location, otherwise, execute *substep d*.

d. Depending on whether the prey is capable of successfully escaping ($r < 0.5$ corresponding to successfully escaping while $r \geq 0.5$ corresponding to the opposite case) before surprise pounce, four subdivided phases are further considered, which are $r \geq 0.5$ and $|E| \geq 0.5$ (the soft besiege phase), $r \geq 0.5$ and $|E| < 0.5$ (the hard besiege phase), $r < 0.5$ and $|E| \geq 0.5$ (the phase of soft besiege with progressive rapid dives), $r < 0.5$ and $|E| < 0.5$, (the phase of hard besiege with progressive rapid dives), respectively. Determine which phase it belongs to and then update location with the formula of the corresponding phase.

Step 4. Calculate fitness value of the prey and add non-dominant solution to the archive.

Step 5. Check whether the number of location vectors exceeds the setting value about the size of repository and truncate the archive if it does.

Step 6. When the number of iterations of algorithm meets the pre-defined maximum value, output location vectors of the prey and corresponding fitness value.

3. SIMULATION AND RESULTS DISCUSSION

3.1 Numerical illustration

In this section, we use the proposed basic framework of RHFA model to simulate a RHFA system and numerically illustrate application of our method. Without loss of generality, we simulate an assembly system consisting of two hybrid flow assembly lines (two machines can be deployed in each assembly process) and five assembly processes. In each assembly process, another alternative machine is available for reconfigurable selection. Assuming that there are six different types of products to assemble, they are divided into three batches for assembly in sequence. More specifically, possible assembly sequence would be $(P_2, P_5) \rightarrow (P_3, P_1) \rightarrow (P_4, P_6)$ or $(P_1, P_5) \rightarrow (P_2, P_6) \rightarrow (P_4, P_3)$ (products to be assembled in the same batch are represented in parentheses). Note that assembly of a type of products will be completed through a hybrid flow assembly line composed of selected machines. For example, considering that products P_1 and P_4 are assembled in the same batch, then diversified equipment composition such as P_1: $MA_{11} \rightarrow MA_{22} \rightarrow MA_{31} \rightarrow MA_{41} \rightarrow MA_{52}$ and P_4: $MA_{12} \rightarrow MA_{21} \rightarrow MA_{32} \rightarrow MA_{42} \rightarrow MA_{51}$ or P_1: $MA_{12} \rightarrow MA_{22} \rightarrow MA_{32} \rightarrow MA_{41} \rightarrow MA_{51}$ and P_4: $MA_{11} \rightarrow MA_{21} \rightarrow MA_{31} \rightarrow MA_{42} \rightarrow MA_{52}$ can either be selected. Table I shows the number of six types of products to be assembled and storage factors for raw materials and end-products. Table II illustrates assembly capacity of various machines (quantified by assembly time), costs of assembling various types of products (assembly costs), and purchase costs of individual machines (price).

Table I: The number of products and storage factors for raw materials (δ_i) and end-products (λ_i).

Product notation	Number	Storage factor (δ_i)	Storage factor (λ_i)
P_1	1000	0.3	0.5
P_2	2000	0.2	0.3
P_3	800	0.6	0.6
P_4	1500	0.4	0.1
P_5	3000	0.1	0.2
P_6	2400	0.5	0.4

Table II: The assembly capacity (time), assembly costs and purchase costs (price) of equipment.

Assembly process	Machine available	Price ($\times 10^4$)	Assembly time (min)						Assembly costs					
			P_1	P_2	P_3	P_4	P_5	P_6	P_1	P_2	P_3	P_4	P_5	P_6
AP_1	MA_{11}	17	8.2	6.6	7.5	7.0	7.2	6.0	6.4	7.5	7.9	6.2	7.8	8.3
	MA_{12}	24	6.3	5.4	5.6	4.6	5.8	4.9	7.2	8.0	9.0	7.5	8.6	9.2
	MA_{13}	12	9.7	7.8	8.0	8.2	8.5	7.5	5.8	6.7	6.9	5.4	7.0	7.2
AP_2	MA_{21}	36	4.4	2.2	2.8	2.5	2.0	3.0	8.5	9.8	7.5	9.0	9.4	8.0
	MA_{22}	20	6.7	4.4	4.9	4.0	4.6	5.0	6.9	7.1	5.0	6.0	7.3	5.5
	MA_{23}	18	7.1	4.6	5.2	4.2	5.0	5.5	6.5	6.7	4.4	5.8	7.0	5.1
AP_3	MA_{31}	27	4.5	5.3	5.1	3.9	6.5	5.5	7.6	6.6	7.0	7.5	6.0	5.7
	MA_{32}	30	4.0	5.0	4.6	3.3	5.8	5.2	8.1	7.0	7.5	7.8	6.6	6.4
	MA_{33}	40	2.8	3.7	3.0	2.1	3.9	3.5	9.5	8.9	9.1	9.9	8.8	9.3
AP_4	MA_{41}	42	3.3	4.3	3.7	3.0	4.0	3.5	12.2	10.2	11.9	13.0	11.7	11.1
	MA_{42}	30	4.6	6.0	5.3	4.1	5.0	4.8	9.0	7.1	7.9	10.6	8.5	8.3
	MA_{43}	50	2.5	3.5	3.0	2.2	2.8	2.6	14.5	12.6	14.0	14.9	13.6	13.8
AP_5	MA_{51}	32	3.5	4.1	5.3	4.8	4.3	4.0	11.6	10.9	9.8	8.6	9.2	9.9
	MA_{52}	27	3.8	5.0	5.9	5.6	5.5	4.4	10.3	10.1	9.3	7.8	8.5	9.1
	MA_{53}	19	5.6	6.1	7.3	6.9	7.1	6.5	8.3	7.9	7.0	6.5	6.2	7.5

3.2 Simulation results of our method

We constructed the simulated scenario with Python, and performed the MOHHO algorithm on MATLAB. The optimization results in MATLAB were fed back to the scenario in Python to achieve simulated hybrid flow assembly operation. Fig. 2 shows the simulation scenario. Note that the highlighted and unilluminated equipment function as discriminators for two discrete assembly lines. Table III shows five simulation results for the RHFA system. Fig. 3 figuratively represents one of the simulated illustrations. Apparently, the simulated results not only provide the batch sequence of products to assemble and the equipment composition of production line for assembling different types of products, but also explicitly indicate whether to reconfigure the equipment in the assembly of the next batch of products. Note that there is no optimal solution for multi-objective optimization problems. However, we can get a Pareto solution set (i.e., a group of non-dominated solutions) by applying MOHHO. These optimized non-dominant solutions stand out from numerous feasible solutions, revealing the validity of our method.

Moreover, according to requirements of enterprises or preferences of decision-makers, it is practicable to assign corresponding weights to single-objectives and sum them (note that all the results are processed by normalization). For example, assuming that the importance of optimization objectives is ranked from highest to lowest as reconfiguration and assembly costs, raw material storage costs, end-product warehousing costs, workload balance of assembly machines, and workload balance of assembly lines, and their corresponding weights are 0.32, 0.24, 0.20, 0.15, and 0.09, then the final optimization results can be determined. Evidently, based on the computed results shown in Table III, the third scheme is more likely to be selected as the preferred one.

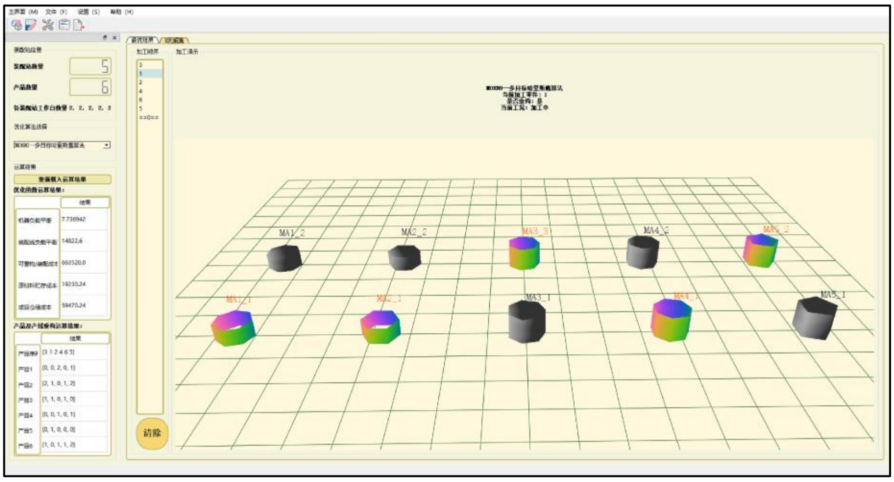

Figure 2: The simulation scenario of the RHFA system.

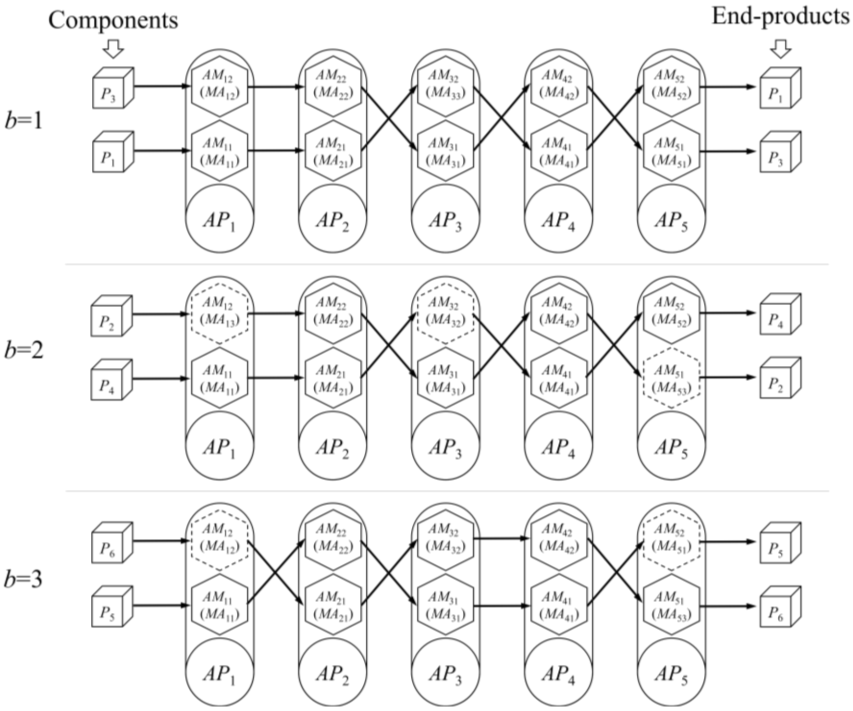

Figure 3: A simulated illustration of the RHFA system optimized via MOHHO algorithm, where dotted lines denote machines which have been reconfigured.

Table III: Five simulation results calculated by MOHHO algorithm.

Algorithm	Soln.	F_1	F_2 ($\times 10^4$)	F_3 ($\times 10^5$)	F_4 ($\times 10^4$)	F_5 ($\times 10^4$)	$n(F_1)$	$n(F_2)$	$n(F_3)$	$n(F_4)$	$n(F_5)$	F
MOHHO	1	7.74	1.48	6.64	1.92	5.95	0.59	0	1	0	1	0.61
	2	8.56	1.70	6.22	2.24	4.70	1	0.48	0.80	0.45	0.16	0.59
	3	6.57	1.94	4.49	2.63	4.46	0	1	0	1	0	0.33
	4	7.45	1.64	5.13	2.63	4.46	0.44	0.35	0.30	1	0	0.43
	5	8.38	1.66	5.77	2.50	5.38	0.91	0.39	0.60	0.82	0.62	0.68

3.3 Comparison and analysis

In this subsection, to demonstrate the preponderance of MOHHO algorithm adopted in this paper, we additionally consider two alternative algorithms MOPSO [15] and MOHPSO [16], which are widely used in multi-objective optimization problems, as comparative methods. For an impartial comparison, all the methods calculated simulation results using MATLAB R2018b on a personal computer equipped with Intel Core i5-1135G7 CPU (2.40 GHz). The simulation settings of three methods are presented in Table IV. Simulated results of assembly batch sequence and equipment composition calculated by three methods are respectively indicated in Table V and Table VI. Besides, Table V also provides optimization results of five single-objectives, and the smaller value means better performance due to the purpose of minimizing five objective functions. These optimization results correspond to the simulation results of assembly batch sequence and equipment composition. According to our simulation scenario, the assembly sequence of three batches is presented from left to right, where products in the same parentheses represent that they are assembled simultaneously on two assembly lines. Table VI shows serial numbers of equipment required by five processes for assembling products, and for different batches, the changing serial numbers of equipment in the same process indicate the equipment in this process is required to be reconfigured before assembling the next batch of products. It should be noted that our method is conducive to find more dominant solutions than others in terms of the multi-objective optimization problem investigated in this paper. For instance, solution 4 of MOHHO can dominate five solutions of MOPSO and four solutions of MOHPSO except the second one, demonstrating the superiority of our method to some extent.

Table IV: Simulation settings of three methods.

Parameters	MOPSO	MOHPSO	MOHHO
Maximum number of iterations	100	100	100
Population size	150	150	150
Repository size	15	15	15
Inertia weight	0.5	0.5	×
Inertia weight damping rate	0.99	0.99	×
Personal learning coefficient	1	1	×
Global learning coefficient	2	2	×
Number of grids per dimension	7	7	7
Inflation rate	0.1	0.1	0.1
Leader selection pressure	2	2	2
Deletion selection pressure	2	2	2
Mutation rate	0.1	0.1	0.1
Hybridization rate	×	0.8	×
Hybrid pool size ratio	×	0.1	×

Table V: Five simulation results of assembly batch sequence (products in the same parentheses are in no particular order) calculated by three methods.

Algorithms	Soln.	F_1	F_2 (×10^4)	F_3 (×10^5)	F_4 (×10^4)	F_5 (×10^4)	Assembly sequence by batch		
MOPSO	1	9.33	2.21	6.91	2.90	6.88	(P_2, P_1)	(P_6, P_3)	(P_4, P_5)
	2	10.86	2.00	5.95	2.90	6.50	(P_2, P_1)	(P_6, P_3)	(P_5, P_4)
	3	9.86	1.87	6.44	3.71	5.63	(P_1, P_2)	(P_3, P_5)	(P_4, P_6)
	4	9.27	2.68	7.57	4.39	5.94	(P_4, P_5)	(P_2, P_3)	(P_1, P_6)
	5	8.64	2.68	6.41	3.54	4.85	(P_4, P_2)	(P_5, P_3)	(P_1, P_6)
MOHPSO	1	8.23	1.80	5.15	3.99	5.84	(P_5, P_4)	(P_3, P_1)	(P_6, P_2)
	2	8.17	1.23	6.27	3.36	5.05	(P_4, P_5)	(P_1, P_3)	(P_2, P_6)
	3	7.94	1.98	5.86	3.45	4.90	(P_4, P_5)	(P_6, P_3)	(P_2, P_1)
	4	8.93	1.86	5.97	4.08	5.46	(P_6, P_5)	(P_4, P_1)	(P_2, P_3)
	5	8.07	1.68	6.75	4.00	5.35	(P_5, P_6)	(P_3, P_2)	(P_4, P_1)
MOHHO	1	7.74	1.48	6.64	1.92	5.95	(P_3, P_1)	(P_4, P_2)	(P_5, P_6)
	2	8.56	1.70	6.22	2.24	4.70	(P_4, P_3)	(P_5, P_6)	(P_1, P_2)
	3	6.57	1.94	4.49	2.63	4.46	(P_4, P_2)	(P_1, P_6)	(P_3, P_5)
	4	7.45	1.64	5.13	2.63	4.46	(P_4, P_2)	(P_6, P_1)	(P_3, P_5)
	5	8.38	1.66	5.77	2.50	5.38	(P_1, P_3)	(P_5, P_4)	(P_6, P_2)

Table VI: Five simulation results of equipment composition calculated by three methods.

Algorithms	Soln.	Equipment composition									
		P_1					P_2				
MOPSO	1	MA_{13}	MA_{22}	MA_{33}	MA_{41}	MA_{51}	MA_{11}	MA_{21}	MA_{31}	MA_{43}	MA_{53}
	2	MA_{13}	MA_{23}	MA_{33}	MA_{41}	MA_{51}	MA_{11}	MA_{21}	MA_{31}	MA_{43}	MA_{53}
	3	MA_{13}	MA_{23}	MA_{33}	MA_{41}	MA_{51}	MA_{12}	MA_{21}	MA_{31}	MA_{43}	MA_{53}
	4	MA_{11}	MA_{21}	MA_{33}	MA_{42}	MA_{52}	MA_{12}	MA_{22}	MA_{31}	MA_{42}	MA_{53}
	5	MA_{11}	MA_{21}	MA_{33}	MA_{43}	MA_{52}	MA_{12}	MA_{22}	MA_{31}	MA_{41}	MA_{53}
MOHPSO	1	MA_{11}	MA_{23}	MA_{31}	MA_{41}	MA_{51}	MA_{12}	MA_{21}	MA_{33}	MA_{41}	MA_{51}
	2	MA_{11}	MA_{23}	MA_{31}	MA_{41}	MA_{51}	MA_{12}	MA_{21}	MA_{33}	MA_{41}	MA_{51}
	3	MA_{12}	MA_{23}	MA_{33}	MA_{43}	MA_{53}	MA_{11}	MA_{21}	MA_{31}	MA_{41}	MA_{51}
	4	MA_{12}	MA_{23}	MA_{31}	MA_{41}	MA_{53}	MA_{11}	MA_{22}	MA_{31}	MA_{41}	MA_{51}
	5	MA_{13}	MA_{23}	MA_{31}	MA_{41}	MA_{53}	MA_{12}	MA_{21}	MA_{31}	MA_{41}	MA_{51}
MOHHO	1	MA_{11}	MA_{21}	MA_{33}	MA_{41}	MA_{52}	MA_{13}	MA_{22}	MA_{31}	MA_{42}	MA_{53}
	2	MA_{11}	MA_{21}	MA_{33}	MA_{41}	MA_{52}	MA_{13}	MA_{22}	MA_{31}	MA_{42}	MA_{53}
	3	MA_{12}	MA_{22}	MA_{31}	MA_{41}	MA_{52}	MA_{12}	MA_{21}	MA_{32}	MA_{41}	MA_{51}
	4	MA_{13}	MA_{22}	MA_{31}	MA_{41}	MA_{52}	MA_{12}	MA_{21}	MA_{32}	MA_{41}	MA_{51}
	5	MA_{11}	MA_{21}	MA_{32}	MA_{41}	MA_{52}	MA_{12}	MA_{21}	MA_{31}	MA_{41}	MA_{52}

Algorithms	Soln.	Equipment composition									
		P_3					P_4				
MOPSO	1	MA_{11}	MA_{23}	MA_{33}	MA_{43}	MA_{51}	MA_{11}	MA_{23}	MA_{33}	MA_{42}	MA_{51}
	2	MA_{11}	MA_{23}	MA_{33}	MA_{43}	MA_{51}	MA_{11}	MA_{23}	MA_{33}	MA_{43}	MA_{51}
	3	MA_{11}	MA_{23}	MA_{33}	MA_{43}	MA_{51}	MA_{11}	MA_{22}	MA_{33}	MA_{42}	MA_{51}
	4	MA_{11}	MA_{23}	MA_{33}	MA_{41}	MA_{51}	MA_{11}	MA_{23}	MA_{33}	MA_{42}	MA_{51}
	5	MA_{11}	MA_{23}	MA_{33}	MA_{42}	MA_{51}	MA_{11}	MA_{23}	MA_{33}	MA_{42}	MA_{51}
MOHPSO	1	MA_{12}	MA_{21}	MA_{33}	MA_{43}	MA_{53}	MA_{11}	MA_{22}	MA_{31}	MA_{43}	MA_{51}
	2	MA_{13}	MA_{21}	MA_{33}	MA_{43}	MA_{53}	MA_{11}	MA_{21}	MA_{31}	MA_{43}	MA_{51}
	3	MA_{13}	MA_{23}	MA_{33}	MA_{43}	MA_{53}	MA_{11}	MA_{21}	MA_{31}	MA_{41}	MA_{53}
	4	MA_{13}	MA_{23}	MA_{33}	MA_{43}	MA_{53}	MA_{11}	MA_{21}	MA_{33}	MA_{43}	MA_{51}
	5	MA_{13}	MA_{23}	MA_{33}	MA_{43}	MA_{52}	MA_{12}	MA_{21}	MA_{33}	MA_{43}	MA_{51}
MOHHO	1	MA_{12}	MA_{22}	MA_{31}	MA_{42}	MA_{51}	MA_{11}	MA_{21}	MA_{32}	MA_{41}	MA_{52}
	2	MA_{12}	MA_{22}	MA_{31}	MA_{42}	MA_{51}	MA_{11}	MA_{21}	MA_{32}	MA_{41}	MA_{52}
	3	MA_{11}	MA_{22}	MA_{31}	MA_{41}	MA_{51}	MA_{11}	MA_{22}	MA_{31}	MA_{42}	MA_{52}
	4	MA_{11}	MA_{22}	MA_{31}	MA_{41}	MA_{51}	MA_{11}	MA_{22}	MA_{31}	MA_{42}	MA_{52}
	5	MA_{12}	MA_{23}	MA_{33}	MA_{42}	MA_{51}	MA_{13}	MA_{23}	MA_{31}	MA_{42}	MA_{51}

Algorithms	Soln.	Equipment composition									
		P_5					P_6				
MOPSO	1	MA_{12}	MA_{22}	MA_{31}	MA_{41}	MA_{52}	MA_{12}	MA_{21}	MA_{31}	MA_{41}	MA_{53}
	2	MA_{12}	MA_{21}	MA_{32}	MA_{41}	MA_{52}	MA_{12}	MA_{21}	MA_{31}	MA_{41}	MA_{53}
	3	MA_{12}	MA_{21}	MA_{31}	MA_{41}	MA_{52}	MA_{12}	MA_{21}	MA_{31}	MA_{41}	MA_{52}
	4	MA_{12}	MA_{21}	MA_{31}	MA_{43}	MA_{53}	MA_{13}	MA_{23}	MA_{32}	MA_{41}	MA_{53}
	5	MA_{12}	MA_{21}	MA_{31}	MA_{41}	MA_{53}	MA_{13}	MA_{23}	MA_{31}	MA_{41}	MA_{53}
MOHPSO	1	MA_{12}	MA_{23}	MA_{33}	MA_{41}	MA_{53}	MA_{11}	MA_{23}	MA_{31}	MA_{43}	MA_{53}
	2	MA_{12}	MA_{23}	MA_{33}	MA_{41}	MA_{52}	MA_{11}	MA_{23}	MA_{31}	MA_{43}	MA_{52}
	3	MA_{12}	MA_{23}	MA_{33}	MA_{42}	MA_{51}	MA_{12}	MA_{21}	MA_{31}	MA_{41}	MA_{51}
	4	MA_{11}	MA_{22}	MA_{31}	MA_{43}	MA_{53}	MA_{12}	MA_{23}	MA_{33}	MA_{41}	MA_{51}
	5	MA_{11}	MA_{22}	MA_{32}	MA_{43}	MA_{53}	MA_{12}	MA_{21}	MA_{33}	MA_{41}	MA_{51}
MOHHO	1	MA_{11}	MA_{22}	MA_{31}	MA_{41}	MA_{51}	MA_{12}	MA_{21}	MA_{32}	MA_{42}	MA_{53}
	2	MA_{11}	MA_{22}	MA_{31}	MA_{41}	MA_{51}	MA_{13}	MA_{21}	MA_{32}	MA_{42}	MA_{53}
	3	MA_{12}	MA_{21}	MA_{32}	MA_{42}	MA_{52}	MA_{11}	MA_{21}	MA_{32}	MA_{42}	MA_{51}
	4	MA_{12}	MA_{21}	MA_{32}	MA_{42}	MA_{52}	MA_{11}	MA_{21}	MA_{32}	MA_{42}	MA_{51}
	5	MA_{11}	MA_{21}	MA_{33}	MA_{41}	MA_{52}	MA_{11}	MA_{23}	MA_{33}	MA_{42}	MA_{51}

Additionally, we quantitatively evaluate the quality of solutions calculated by different methods. In this paper, three commonly used evaluation metrics including mean ideal distance (*MID*) [17], maximum spread (*MS*) [18] and spacing metric (*SM*) [19] are employed to comprehensively evaluate performance of different methods. In the realm of optimization algorithms, *MID* serves as a valuable metric for gauging the degree to which the Pareto solution set aligns with the ideal solution. The attainment of a reduced *MID* value signifies a heightened proximity between the optimized solution set and the ideal solution. *MS* is commonly utilized to evaluate the universality of solutions distribution. A higher *MS* value suggests a more widely dispersed solution set, thereby indicating a comparably superior performance. *SM* serves as a crucial index for assessing the uniformity of the solution distribution. A decrease in the *SM* value is indicative of more evenly distributed solutions, revealing a relatively better performance. We performed each algorithm five times under the condition of random initialization, and acquired average value of each index from five optimization results for evaluation. Table VII summarizes evaluation results of three methods using *MID*, *MS* and *SM* indexes. It is observed that MOHHO adopted in this paper obtains minimum value in *MID* and *SM* (the smaller the value indicates that the better the algorithm performs) and maximum value in *MS* (the larger the value indicates that the better the algorithm performs). Therefore, compared with other alternatives, MOHHO exhibits more competitive results in terms of algorithm convergence, universality, and uniformity of solution distribution in solving the multi-objective optimization problem investigated in this paper.

Table VII: Algorithms performance evaluated on *MID*, *MS* and *SM* indexes.

Multi-objective optimization algorithms	MID ↓	MS ($\times 10^5$) ↑	SM ($\times 10^4$) ↓
MOPSO	1.28	2.27	1.81
MOHPSO	1.23	2.51	1.69
MOHHO	1.21	2.54	1.55

4. CONCLUSION

In this paper, we simulate a RHFA system with high flexibility oriented toward multi-variety and small-batch production. It is conducive to promoting the reduction of production cycle duration, regulation of production expenses, and enhancement of product quality, and sufficiently reflecting the characteristics of flexible manufacturing. For the RHFA scenario, we propose five objective functions to perform joint optimization for equipment composition and assembly batch sequence. To address the multi-objective optimization problem, we adopt MOHHO algorithm to construct an optimization scheme. Results indicate that MOHHO outperforms other algorithms such as MOPSO and MOHPSO. Nevertheless, our study also has some limitations, primarily characterized by the disregard of certain assembly constraints on the model, including makespan, total energy consumption expenses of machines, and tardiness. In future research, we plan to incorporate these constraints and explore more competitive multi-objective optimization methods to design better performing RHFA systems.

ACKNOWLEDGEMENT

This research was supported by the National Key Research and Development Program of China, grant number 2018YFB1701303. This work was also supported by the Key Research and Development Program of Zhejiang Province (No. 2022C01139).

REFERENCES

[1] Koren, Y.; Heisel, U.; Jovane, F.; Moriwaki, T.; Pritschow, G.; Ulsoy, G.; van Brussel, H. (2003). Reconfigurable manufacturing systems, Dashchenko, A. I. (Ed.), *Manufacturing Technologies for Machines of the Future*, Springer, Berlin, 627-665, doi:10.1007/978-3-642-55776-7_19

[2] Priego, R.; Iriondo, N.; Gangoiti, U.; Marcos, M. (2017). Agent-based middleware architecture for reconfigurable manufacturing systems, *The International Journal of Advanced Manufacturing Technology*, Vol. 92, No. 5-8, 1579-1590, doi:10.1007/s00170-017-0154-z

[3] Mehrabi, M. G.; Ulsoy, A. G.; Koren, Y.; Heytler, P. (2002). Trends and perspectives in flexible and reconfigurable manufacturing systems, *Journal of Intelligent Manufacturing*, Vol. 13, No. 2, 135-146, doi:10.1023/A:1014536330551
[4] Malhotra, V.; Raj, T.; Arora, A. (2010). Excellent techniques of manufacturing systems: RMS and FMS, *International Journal of Engineering Science and Technology*, Vol. 2, No. 3, 137-142
[5] Bhatwadekar, S. G.; Ingale, S. M.; Dandage, R. V. (2019). Reconfigurable manufacturing systems (RMS): a literature review, *International Journal for Research in Engineering Application & Management*, Vol. 4, Special Issue CTRD 2018, 35-39
[6] De Lit, P.; Delchambre, A.; Henrioud, J.-M. (2003). An integrated approach for product family and assembly system design, *IEEE Transactions on Robotics and Automation*, Vol. 19, No. 2, 324-334, doi:10.1109/TRA.2003.808853
[7] Setchi, R. M.; Lagos, N. (2004). Reconfigurability and reconfigurable manufacturing systems: state-of-the-art review, *Proceedings of the 2nd IEEE International Conference on Industrial Informatics*, 529-535, doi:10.1109/INDIN.2004.1417401
[8] Naderi, B.; Zandieh, M.; Khaleghi Ghoshe Balagh, A.; Roshanaei, V. (2009). An improved simulated annealing for hybrid flowshops with sequence-dependent setup and transportation times to minimize total completion time and total tardiness, *Expert Systems with Applications*, Vol. 36, No. 6, 9625-9633, doi:10.1016/j.eswa.2008.09.063
[9] Marichelvam, M. K.; Prabaharan, T.; Yang, X. S. (2014). A discrete firefly algorithm for the multi-objective hybrid flowshop scheduling problems, *IEEE Transactions on Evolutionary Computation*, Vol. 18, No. 2, 301-305, doi:10.1109/TEVC.2013.2240304
[10] Hosseini, S. M. H. (2017). A multi-objective genetic algorithm (MOGA) for hybrid flow shop scheduling problem with assembly operation, *Journal of Industrial and Systems Engineering*, Vol. 10, Special issue on production and inventory, 132-154
[11] Zheng, J.; Wang, L.; Wang, J.-J. (2020). A cooperative coevolution algorithm for multi-objective fuzzy distributed hybrid flow shop, *Knowledge-Based Systems*, Vol. 194, Paper 105536, 11 pages, doi:10.1016/j.knosys.2020.105536
[12] Xu, W.; Sun, H. Y.; Awaga, A. L.; Yan, Y.; Cui, Y. J. (2022). Optimization approaches for solving production scheduling problem: a brief overview and a case study for hybrid flow shop using genetic algorithms, *Advances in Production Engineering & Management*, Vol. 17, No. 1, 45-56, doi:10.14743/apem2022.1.420
[13] Jemmali, M.; Hidri, L.; Alourani, A. (2022). Two-stage hybrid flowshop scheduling problem with independent setup times, *International Journal of Simulation Modelling*, Vol. 21, No. 1, 5-16, doi:10.2507/IJSIMM21-1-577
[14] Heidari, A. A.; Mirjalili, S.; Faris, H.; Aljarah, I.; Mafarja, M.; Chen, H. (2019). Harris hawks optimization: algorithm and applications, *Future Generation Computer Systems*, Vol. 97, 849-872, doi:10.1016/j.future.2019.02.028
[15] Coello Coello, C. A.; Lechuga, M. S. (2002). MOPSO: a proposal for multiple objective particle swarm optimization, *Proceedings of the 2002 Congress on Evolutionary Computation*, 1051-1056, doi:10.1109/CEC.2002.1004388
[16] Yang, J.; Liu, F.; Dong, Y.; Cao, Y.; Cao, Y. (2022). Multiple-objective optimization of a reconfigurable assembly system via equipment selection and sequence planning, *Computers & Industrial Engineering*, Vol. 172, Part A, Paper 108519, 11 pages, doi:10.1016/j.cie.2022.108519
[17] Nemati-Lafmejani, R.; Davari-Ardakani, H.; Najafzad, H. (2019). Multi-mode resource constrained project scheduling and contractor selection: mathematical formulation and metaheuristic algorithms, *Applied Soft Computing*, Vol. 81, Paper 105533, 22 pages, doi:10.1016/j.asoc.2019.105533
[18] Zitzler, E.; Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach, *IEEE Transactions on Evolutionary Computation*, Vol. 3, No. 4, 257-271, doi:10.1109/4235.797969
[19] Schott, J. R. (1995). *Fault Tolerant Design Using Single and Multi-Criteria Genetic Algorithms*, Master Thesis, Massachusetts Institute of Technology, Cambridge