International Journal of Engineering

Journal Homepage: www.ije.ir

Distributed Production Assembly Scheduling with Hybrid Flowshop in Assembly Stage

M. Torkashvand, F. Ahmadizar*, H. Farughi

Department of Industrial Engineering, University of Kurdistan, Sanandaj, Iran

PAPER INFO

Paper history:
Received 07 January 2022
Received in revised form 15 February 2022
Accepted 18 February 2022

Keywords:
Production-assembly
Basic Scheduling
Longest Processing Time
Hybrid Genetic Algorithm
Flowshop Scheduling

ABSTRACT

A new three stage production-assembly problem is considered in this paper. To the best of our knowledge, considering parallel machines in the third stage, identical parallel factories including the three stage production-assembly system and identical parallel factories with parallel machines in the third stage of the production-assembly system, has been specifically investigated in this paper. To minimize the maximum completion time (Makespan) of all jobs in the all factories, jobs assignment to factories and their processing sequence should be done properly. A Mixed Integer Linear Programming (MILP) model is presented to solve small size problem by using cplex solver. According to the problem computational complexity, large size of problem is not possible to solve using the cplex, so to solve it and to control the computational complexity, a new improved genetic algorithm (GA) is proposed by combining GA and Longest Proseccing Time (LPT) method that is called Hybrid Genetic Algorithm Longest Proseccing Time (HGALPT). The problem parameters values are determined using one-way analysis of variance (ANOVA). Finally, in order to evaluate the efficiency and effectiveness of the proposed algorithm, and to specify each parameter impact on the objective function, sensitivity analysis is performed on the problem parameters.

***doi**: 10.5829/ije.2022.35.05.19*

1. INTRODUCTION

Study the relationship between mental ideas and reality is one of the most important issues that management should pay attention to it. Implementing any plan requires strong managerial insight. Before starting any activity, searching and gathering information about the market situation, examining the past trend of the business under study and predicting the trend of short-term and long-term changes in the future, is a necessary condition for the starting a successful business. The result of this study determines whether there is a demand for a service or a product in the market or not.

To create a plan, all of the effective parameters must be specified. Reviewing and deciding to implement the plan, requires information about the current market situation, estimating the future situation and considering related budget issues. It is clear that, if the establishment of a factory is justified to produce a product and the feasibility study is acceptable taking into account all economic, social and environmental factors; management makes the final decision to implement it. The production planning and scheduling phase occurs in the operation stage of a project, and it is at this stage that all the actions taken so far are effective with a proper planning.

A distributed production-assembly scheduling with hybrid flowshop in assembly stage is considered in this paper. In the production-assembly system, production and assembly operations are performed in two separate but consecutive stages. This system is known as the production-assembly flow shop scheduling. In this case, each job is produced in two separate stages: in the first stage, different parts of the final job are produced and in the second stage, these parts are assembled together. Often in the production stage, different operations are

*Corresponding Author Institutional Email: *f.ahmadizar@uok.ac.ir* (F. Ahmadizar)

Please cite this article as: M. Torkashvand, F. Ahmadizar, H. Farughi, Distributed Production Assembly Scheduling with Hybrid Flowshop in Assembly Stage, ***International Journal of Engineering, Transactions B: Applications***, Vol. 35, No. 05, (2022) 1037-1055

performed independently and in parallel to produce different components, which eventually become the final product in the assembly stage. In the field of production and preventive maintenance to increase the equipment life, production scheduling and preventive maintenance is used [1].

The main motivation for presenting this paper is to solve some of the problems in production and service environments through planning and scheduling of jobs and services. Delivery time of customer requests, using of equipment and tools optimalty, increase producer and consumer satisfaction, provide products in accordance with the delivery time, variety and customers requested volume, retain their customers, reduce production costs and ultimately earn more revenue, are the result of production planning and scheduling.

Achieving to the specified goals is not easily in the mentioned systems and requires appropriate treatment against of existing challenges. There are challenges in the production-assembly scheduling field, some of them are:

- Increasing the products variety by increasing the production companies of a product with a specific function.
- Trying to increase market share by customizing customer requests and produce a product according to customer requests.
- Increasing demand diversity instead of mass production.
- Forced changes in production systems of manufacturers due to the product short life cycle.
- Dynamic and uncertain global market that has caused the creation of new technologies in the production and services sectors and the manufacturer needs to update its equipment and tools daily.

In response to the mentioned challenges, by combinations of produced components in the production stage, a high variety of the final product is created in the assembly stage, therefore, production-assembly problems are used in different production systems to increase the product flexibility and variety [2]. The production-assembly scheduling problem applies to Make-To-Order environments in which production operations begin after receiving a customer order. Each order includes a combination of product types that must be delivered to the customer in a single shipment and the manufacturer schedule unscheduled and cumulative orders, periodically [3].

The three of most important reasons for presenting this paper are as follows:

- The first is the three-stage production-assembly problem.
- The second is the same parallel machines in the third stage.
- The third is to consider identical parallel factories.

According to the first reason, the identical parallel machines in the third stage in single factory and three stage production-assembly scheduling problem in multi-factory not reviewed, as it will be mentioned in the literature review section, researchers in most papers have presented two-stage production-assembly problem; while, it is more realistic that the number of stages would be more than two. According to the second reason, in order to prevent customer dissatisfaction of product delivery delay, to reduce completion time of product and accelerate the production of the final product, considering the identical parallel machines in the third stage makes the system more efficient.

According to the third reason, it is assumed that the jobs are created in a single factory in classical production-assembly flow shop problems, while in order to reduce production costs (transportation costs) and production period, many manufacturers changed a single factory production system to a distributed production system (identical parallel factories). The distributed scheduling problem involves two main decisions: assigning jobs to factories and sequencing jobs in each factory.

By considering the three importance reasons and to the best of our knowledge it seems necessary to study, presented problem in this paper; thus, the distributed production-assembly scheduling with hybrid flowshop in assembly stage is considered. In order to adjust the jobs completion times or final products and maximize the machines capacity in all three stages, we have considered minimizing the maximum completion time as a suitable criterion for evaluation.

To the best of our knowledge, two cases of problem contributions are: the mathematical model and the solution method. Two solution methods have been used, an exact method which is the problem model and an approximate method which is the Hybrid Genetic Algorithm Longest Proseccing Time (HGALPT), in order to solve the problem. To solve the proposed problem, we proposed a new Mixed Integer Linear Programming (MILP) model. This model is position based and it has been used to check the results accuracy in small sizes. This model does not have the ability to solve large size problems.

The scheduling problem for single-machine, stage and factory modes have been extensively investigated in literature. The proposed models for each mode have computational complexity and the problems in this mode have been proven to be Non-deterministic Polynomial-time hard (NP-Hard) Lee et al. [4] have been proven that for the single-factory mode, the production-assembly problem of three machines (two machine in first stage and single machine in second stage) with the makespan objective function is strongly NP-hard, while our proposed problem is more complex than doiscussed case.

Garey et al. [5] examined the complexity of flowshop and jobshop scheduling problems and showed that for flowshop scheduling problems for more than two machines and the makespan objective function, the problem is NP-complete.

Therefore, in order to manage the computational complexity, a new meta-heuristic algorithm by combining the genetic algorithm (GA) and the longest processing time method is proposed. Here we have also used the GA, with this difference that in order to improve the results. We have combined it with an longest proseccing time (LPT) heuristic method and presented a modified GA. The experiments results showed the proper performance of the HGALPT algorithm. The GA has been used in many papers to solve various problems. Deng et al. [6] proposed the GA algorithm with variable neighborhood search (VNS) to minimize the total completion time and maximum completion time simultaneously.

At the end of the paper, experiments and numerical calculations have been performed, in order to evaluate the effectiveness and efficiency of the HGALPT. To investigate and evaluate the proposed mathematical model performance, the model results in the optimal solution are checked and after ensuring of the model results accuracy, the results have been compared with meta-heuristic in small size. To evaluate the meta-heuristic algorithm, the results accuracy and the problem solving speed of the meta-heuristic algorithm is investigated compared to the proposed exact model. Due to the fact that the mathematical model is not able to solve the large size of the problem, we tried to solve the problem using the classical GA and improved GA.

After performing numerical experiments, eventually, it was found that the presented algorithm has a high ability to achieve the optimal solution in less time than MILP in small sizes and provides the optimal or near-optimal solution in large sizes. Also, in order to investigate the effect of each parameter on the objective function, sensitivity analysis was performed at the end.

According to the mentioned cases, the innovations of the problem are:

1. Three stage production-assembly problem with the identical parallel machines in the third stage.
2. Idenical parallel machines in the third stage of the distributed production-assembly problem.
3. Parallel factories including three stage production-assembly problem.
4. New position based mathematical model for the production-assembly problem.
5. Hybrid metaheuristic algorithm with a combination of GA and sorting algorithm based on the longest processing time.

The main framework of the paper is formed as follows: section 2 describes literature review related to distributed production-assembly scheduling with hybrid flowshop in assembly stage. The mathematical model of the problem is presented in section 3 completely, which this section aim is to minimize the maximum completion time the all jobs. Section 4 presents the used solution method for the problem and a complete description of the algorithm with detailes. Section 5 shows the comparisons of solution methods in small and large sizes, the computational results of each algorithm and parameters sensitivity analysis. An overview of the actions taken and suggestions for future works is provided in section 6.

2. LITERATURE REVIEW

In this section, the presented papers in the production-assembly systems were reviewed. As mentioned, the problem is n job scheduling (customer order), which are a combination of product types. Various papers have considered different production modes for this problem and have presented the applications of the real world, which are referred to these papers in the following.

Hatami et al. [7] studied the distributed unrelated parallel machines, establishing a set of parallel factories with parallel machines in production stage.The job must be assigned to factories and machines and there is a single machine assembly stage. The objective function is to minimize makespan in the assembly stage.

Framinan and Perez-Gonzalez [8] considered the order scheduling problem to minimize total tardiness so that each machine is capable of producing one (and only one) specific type of product (in fact, machines are dedicated). Xiong et al. [9] assumed the distributed two stage assembly flow shop scheduling problem. The aim is to assign job to multiple factories and schedule job in each factory so that the total completion time is minimized. N jobs are available so that they can be processed by the same factory f, and each factory has the same number of m dedicated machines in the first stage and one assembly machine in the second stage.

The minimization of the makespan in the 3-machine assembly-type flowshop scheduling problem with two machines in first and a machine in second satge was considered [4]. Lee et al. [4] presented a branch and bound algorithm and three heuristic algorithms to solve the problem. Allahverdi and Al-Anzi [10] considered the two-stage assembly scheduling problem to minimize makespan with setup times. In the first stage, there are m production machine and in the second stage, an assembly machine. The three-stage production-assembly flowshop with parallel machines in the last two stages was considered by Zhang et al. [11]. The objective function is to minimize the maximum completion time of all orders.

Xiong et al. [12] considered a flexible assembly-differentiation flow shop scheduling problem to minimize total completion time with three stages of component production, assembly, and separation. All the components of a job are processed by different machines in the first stage, then in the second stage these components are assembled together by one machine. In the third stage, each job of a specific type is processed by a dedicated machine. They propose a mixed integer programming (MIP) model, two heuristic algorithms and three meta-heuristic algorithms to solve the problem.

Liu et al. [13] proposed the distributed assembly permutation flowshop scheduling problem. The objective function is to minimize the maximum completion time. The problem is formed by two stages, the processing stage and the assembly stage. Sung and Kim [14] considered minimizing the total completion times at the multi-machine production-assembly scheduling problem. The first stage consists of two independent machines and the second stage consists of two identical machines that are located in parallel.

Maboudian and Shafaei [15] proposed the two stage assembly flow shop scheduling problem with sequence-dependent setup times and minimizing the maximum completion time and maximum tardiness objective functions. In this problem, n products must be produced, each product having m inequal parts, which in the first stage are processed simultaneously by m machines, and each part has a dedicated machine. In the second stage, the final product is created by assembling m parts using a machine. Fattahi et al. [16] presented a hybrid flow shop scheduling problem with setup and assembly operations. The parts are produced in a two-stage flexible flow shop (including one machine in the first stage and m machine in the second stage) and then convert to the final product in the assembly stage.

Xiong et al. [17] considered a distributed production-assembly flow shop scheduling problem to minimize the total weight of the maximum completion time and the average completion time. The factories are located in parallel and in each factory the jobs are first processed by m dedicated parallel machine and then sent to the assembly stage, which performs on an assembly machine. Mahdavi et al. [18] also considered hybrid flowshop scheduling with assembly operations. The parts are processed in the flexible flow shop stage and then sent to the assembly stage to produce the final product. The aim is to find a schedule that minimizes the completion time of the last product. They presented an integer programming model and two heuristic algorithms and simulated annealing algorithm to solve the problem.

Pan et al. [19] demonstrated a distributed assembly permutation flowshop scheduling problem. There are some of identical factories that a flowshop for part-processing and assembly line for product-processing there is in each factory is presented. The objective function is makespan that to be minimize. Huang and Gu [20] presented a biogeography-based optimization (NBBO) algorithm to solve the distributed assembly permutation flow-shop scheduling problem with sequence-dependent set-up times. The objective function of this problem is minimizing the makespan. Each factory consists of M machines {1, 2, … , M}. The second stage is the assembly stage, which has only one assembly machine MA in the factory.

Lei et al. [21] distributed unrelated parallel machine scheduling with minimizing makespan and total tardiness simultaneously so that n jobs distributed among F factories located in different sites is considered. Each factory f is composed of m_f unrelated parallel machines. Fathollahi-Fard et al. [22] presented the new concept of production scheduling at sustainable Distributed Permutation Flow Shop Scheduling Problem (DPFSP). The aim is minimize the total energy consumption related to production and maximize, the social factors linked to job opportunities and lost working days. To solve the problem, they proposed novel multi-objective learning-based heuristic is established, as an extension of the Social Engineering Optimizer (SEO).

Wang and Fathollahi-Fard [23] proposed a multi-objective low-carbon hybrid flow shop scheduling problem (MLHFSP) with the consideration of machines with varied energy usage ratios. The objective function is minimizing total carbon emission (TCE) and makespan (C_{max}). An improved multi-objective teaching-learning-based optimization (ITLBO) algorithm proposed to solve their problem and to avoid local optimum, sequential neighbourhood search (SNS) method also adopted.

Hosseini [24] presented a distributed assembly permutation flow-shop scheduling problem. The first stage of the considered production system is composed of several non-identical factories with different technology levels and so the factories' performance is different in terms of processing time and cost. The second stage is an assembly stage where in there are some parallel jon stations to assemble the ready parts into the products. The objective function is to minimize the makespan.

Jabbari et al. [25] proposed the scheduling problem for a customized production system consisting of a flow shop production line with a parallel assembly stage that produces various products in two stages. In the first stage of the production line, parts are produced using a flow shop production line, and in the second stage, products are assembled on one of the parallel assembly lines. The objective is to minimize makespan.

In the literature, the most complex studied mood in the single-factory production-assembly system is the three stage production-assembly flow shop presented by Xiong et al. [12], which is related to the case where the dedicated parallel machines in the production stage and a two-stage flexible flow shop (one machine for assembly operations and dedicated parallel machines for post-assembly operations) are established in the assembly stage. Each job in the third stage can only be processed by one dedicated machine. For the multi-factory mode.

Xiong and Xing [17] investigated two stage production-assembly flow shop scheduling problem, that the first stage is dedicated parallel machines and the second stage is single machine.

The summary of the mentioned papers in production-assembly problem is presented in Table 1, that are separated to one factory and multi factory. In order to compare the here presented problem with the papers in the literature, four criteria: objective function,

TABLE 1. Summary of the literature review for production-assembly problem

Author	Solution algorithm/ Objective	Machines Position	Factory Number
[8]	Constructive Heuristic and Matheuristic/total tardiness	**first stage:** dedicated parallel machines	
[25]	GA Particle Swarm Optimization (PSO) hybrid algorithm(GA & PSO)/Makespan	**first stage:** Flowshop **second stage:** parallel machine	
[18]	simulated annealing (SA)/Makespan	**first stage:** Hybrid Flowshop **second stage:** parallel machines	
[4]	branch and bound and three heuristics/Makespan	**first stage:** two unrelated parallel machines **second stage:** single machine	
[10]	Particle Swarm Optimization and Tabu search/Makespan	**first stage:** unrelated parallel machines **second stage:** single machine	
[15]	mathematical model/Makespan and maximum tardiness	**first stage:** unrelated parallel machines **second stage:** single machine	Single factory
[9]	heuristics and three hybrid meta-heuristics(HVNS, HGA-RVNS, andHDDE-RVNS)/total completion time	**first stage:** unrelated parallel machines **second stage:** single machine	
[14]	branch and bound/sum of completion times	**first stage:** two unrelated parallel machines **second stage:** two parallel machines	
[11]	hybrid geneticalgorithm(HGA) /Makespan	**first stage:** unrelated parallel machines **second stage:** parallel machines	
[16]	GA, simulated annealing (SA), NEH and Johnson's algorithm /Makespan	**first stage:** single machine **second stage:** unrelated parallel machines **third stage:** single machine	
[12]	SPT-based heuristic, NEH-based heuristic, HGA-VNS, HDDE-VNS and HEDA-VNS/total flow time (TFT).	**first stage:** unrelated parallel machines **second stage:** single machine **third stage:** Differentiation parallel machines	
[7]	heuristic methods (Heuristics PJ1 and PJ2)/Makespan	**first stage:** Distributed unrelated parallel machines **second stage:** single machine	
[26]	Variable Neighborhood based Memetic Algorithm/Makespan	**first stage:** Distributed unrelated parallel machines **second stage:** single machine	Distributed factories in first stage
[20]	Biogeography-based optimization (BBO)/Makespan	**first stage:** Distributed Flowshop **second stage:** single machine	
[24]	Two-level self-adaptive variable neighborhood search (TLSAVNS) algorithm/Makespan	**first stage:** Distributed Flowshop **second stage:** parallel machine	
[22]	Social Engineering Optimizer (SEO)/Makespan,total energy, social factors linked to job opportunities and lost working days	**first stage:** flow-shop	
[21]	artificial bee colony/makespan and total tardiness	**first stage:** unrelated parallel machines	Distributed factories
[19]	heuristics, variable neighborhood search algorithms, and meta-heuristics/Makespan	**first stage:** Flowshop **second stage:** single machines	
[17]	GA-RVNS and VNS/makespan and mean completion time	**first stage:** unrelated parallel machines **second stage:** single machine	
current paper	Hybrid Genetic Algorithm Longest Proseccing Time/Makespan	**first stage:** unrelated parallel machines **second stage:** single machine **third stage:** parallel machines	Distributed factories

solution algorithm, position of machines and number of factories have been used which is shown in Table 1.

The papers are divided into three categories: single-factory, first stage distributed factories and distributed factories. In the one-factory mode, papers are divided in to three modes: one-stage (one paper), two-stage (eight papers) and three-stage (two papers). In the three-stage mode, the first paper considers a prerequisite stage before the production stage and the second paper considers an additional stage after the assembly stage with different parallel machines.

In the first stage distributed factories, the papers are reviewed in three modes: parallel machines in the first stage and one machine in the second stage (two papers), flowshop in the first stage and one machine in the second stage (one paper) and flowshop in the first stage and parallel machines in the second stage (one paper). For distributed factories mode, papers are reviewed in two mode: one-stage (two papers) and two-stage (two papers).

According to the papers classification in Table 1 as well as real world problems, the following research gaps can be mentioned:

- Only one paper in the single-factory mode addresses the three-stage production-assembly problem where there are no identical parallel machines in stage three.
- In the distributed factories mode, no paper considers three stage while this happens in the real world.
- In the distributed factories mode, no three-stage study with parallel machines in the third stage is presented.
- None of the papers have used a hybrid GA with LPT.

To the best of our knowledge, according to the presented cases and in order to cover the four mentioned gaps, three stages production-assembly flow shop scheduling problem with parallel machines in the assembly stage and parallel factories is presented. Regarding the first gap: the three stages production-assembly problem with parallel machines in the third stage is presented in parallel factories, which includes single factory too. For the second gap: the three stage production-assembly problem in several factories is considered. For the third gap: for parallel machines in the third stage of production-assembly is considered. For the fourth gap: we present a new improved GA in this paper. Regarding the objective function, the objective function has been used in other papers but has not been used with other criteria of this paper.

3. MATHEMATICAL MODEL

In this section the mathematical model of the problem is presented. In flowshop scheduling problem a set of N products are produced on a set of m machines [27]. As mentioned earlier, the distributed production-assembly scheduling problem with the hybrid flow shop in the assembly stages can be defined as following: There is the set of n jobs that are performed in three stages. Each job is processed on m + 2 machines: on m_1 dedicated parallel machine in the first stage, one machine in the second stage and one machine in the third stage. In the third stage, there are the number of m_3 same parallel machines, where each job is processed on one machine without interruption. In the second and third stages there is a hybrid flow shop. hybrid flow shop environment is similar to flow shop, but at least in one stage the number of machines is more than one.

The aim is assignment of jobs to factories and determining the processing sequence of jobs in each factory so that the maximum completion time of jobs is minimized. The assumptions of the problem are:

- Machines are available constantly.
- Each machine processes only one job at a time.
- The first stage machines are dedicated and the processing times for each part on the machine can be different.
- All jobs components are available in zero time and their processing times are specified.
- jobs pre-emption it is not permissible.
- Each job can be processed by one machine at a time.
- The assembly of a job begins when all its components have been completed in the first stage.

The mathematical model of the problem is defined as following.

Parameters and Indices	
n	The number of jobs
F	The number of factories
m_1	The number of machines in the first stage
m_3	The number of machines in the third stage
k	The machines indice in the first stage $\{1,2,\dots,m_1\}$
i, j, r	The jobs indices $\{1,2,\dots,n\}$
l, s	The machines indice in the third stage $\{1,2,\dots,m_3\}$
f, q	The factories indice $\{1,2,\dots,F\}$
$p_{j,k}$	the processing time of job j on machine k at the first stage
tt_j	the processing time of job j at the second stage
pt_j	the processing time of job j at the third stage
M	A large positive number

Decision varibales	
$X_{i,w,f}$	1 If job i is processed in position w in the production stage and the first assembly stage in factory f, 0 otherwise
$Y_{i,w,l,f}$	1 If job i is processed in position w on machine l in the second assembly stage at factory f, 0 otherwise
$C_{w,k,f}$	Completion time of job in position w on machine k at production stage in factory f
$CA_{w,f}$	Completion time of the job in position w in the first assembly stage in factory f
$CT_{w,l,f}$	Completion time of the job in position w on the machine l in the second assembly stage in te factory f

Model

Minimise C_{max} (1)

$\sum_{f=1}^{F}\sum_{w=1}^{n} X_{i,w,f} = 1 \quad \forall\, i = 1,2,\dots,n$ (2)

$\sum_{i=1}^{n} X_{i,w,f} \leq 1 \quad \forall\, f = 1,\dots,F;\ w = 1,\dots,n$ (3)

$\sum_{i=1}^{n} X_{i,w-1,f} \geq \sum_{j=1}^{n} X_{j,w,f} \quad \forall\, f = 1,\dots,F;\ w = 2,\dots,n$ (4)

$\sum_{f=1}^{F}\sum_{w=1}^{n}\sum_{l=1}^{m_3} Y_{i,w,l,f} = 1 \quad \forall\, i = 1,2,\dots,n$ (5)

$\sum_{i=1}^{n} Y_{i,w,l,f} \leq 1 \quad \forall\, f = 1,\dots,F;\ w = 1,\dots,n;\ l = 1,2,\dots,m_3$ (6)

$\sum_{i=1}^{n} Y_{i,w-1,l,f} \geq \sum_{j=1}^{n} Y_{j,w,l,f} \quad \forall\, w = 2,\dots,n;\ l = 1,2,\dots,m_3;\ f = 1,\dots,F$ (7)

$\sum_{r=1}^{n}\sum_{l=1}^{m_3} Y_{i,r,l,f} = \sum_{w=1}^{n} X_{i,w,f} \quad \forall\, i = 1,2,\dots,n;\ f = 1,\dots,F$ (8)

$C_{w,k,f} \geq C_{w-1,k,f} + \sum_{i=1}^{n} p_{i,k} * X_{i,w,f} \quad \forall\, w = 2,\dots,n,;\ k = 1,2,\dots,m_1;\ f = 1,\dots,F$ (9)

$C_{1,k,f} \geq \sum_{i=1}^{n} p_{i,k} * X_{i,1,f} \quad \forall\, k = 1,2,\dots,m_1;\ f = 1,\dots,F$ (10)

$CA_{1,f} \geq \sum_{i=1}^{n} t_i * X_{i,1,f} \quad \forall\, f = 1,\dots,F$ (11)

$CA_{w,f} \geq CA_{w-1,f} + \sum_{i=1}^{n} t_i * X_{i,w,f} \quad \forall\, w = 2,\dots,n;\ f = 1,\dots,F$ (12)

$CA_{w,f} \geq C_{w,k,f} + \sum_{i=1}^{n} t_i * X_{i,w,f} \quad \forall\, w = 1,\dots,n;\ k = 1,2,\dots,m_1;\ f = 1,2,\dots,F$ (13)

$CT_{w,l,f} \geq CT_{w-1,l,f} + \sum_{i=1}^{n} pt_i * Y_{i,w,l,f} - M * \left(1 - \sum_{i=1}^{n} Y_{i,w,l,f}\right) \quad \forall\, l = 1,2,\dots,m_3;\ w = 2,\dots,n;\ f = 1,\dots,F$ (14)

$CT_{w,l,f} \geq CA_{r,f} + pt_i * Y_{i,w,l,f} - M * \left(2 - Y_{i,w,l,f} - X_{i,r,f}\right) \quad \forall\, i = 1,2,\dots,n;\ w,r = 1,2,\dots,n;\ l = 1,2,\dots,m_3;\ f = 1,\dots,F$ (15)

$C_{max} \geq CT_{w,l,f} \quad \forall\, w = 1,2,\dots,n;\ l = 1,2,\dots,m_3;\ f = 1,\dots,F$ (16)

$X_{i,w,f} \in \{0,1\}\,,\ Y_{i,w,l,f} \in \{0,1\} \quad \forall\, i = 1 \dots. n;\ w = 1,\dots. n;\ l = 1,\dots,m_3;\ f = 1,\dots,F$ (17)

$C_{w,k,f}, CA_{w,f}, CT_{w,l,f} \geq 0 \quad \forall\, w = 1,\dots. n;\ k = 1,2,\dots,m_1;\ f = 1,\dots,F$ (18)

Constraint (1) represents the objective function of the problem, ie the maximum completion time.

Set of the production stage and first assembly stage constraints: Constraint (2) means that each job must be assigned to one position. Constraint (3) indicates that one job is assigned to each position in each factory extremely. Constraint (4) indicates that a position is filled if the previous position be filled.

Set of second assembly stage constraints: Constraint (5) specifies that each job is assigned to only one position of a machine. Constraint (6) specifies that one job is assigned to each position of each machine, extremely. Constraint (7) indicates that a position of a machine is filled if the previous position be filled.

Common constraints: Constraint (8) forces that if a job is assigned to a factory for the first stages of production and assembly, it must be assigned to the same factory for the third stage. Constraint (9) specifies that completion time a job in one position at the production stage cannot be less than completion time another job in the previous position. Constraint (10) indicates that if a job is placed in the first position in the production stage, its completion time will not be less than its processing time. Constraint (11) indicates that if a job is placed in the first position in the first assembly stage, its completion time will not be less than its processing time. Constraint (12) specifies that the completion time of any job in a certain position in the first assembly stage can not be less than the its completion time in the previous position of same stage.

Constraint (13) specifies that the completion time of each job in a certain position in the first assembly stage can not be less than completion time of the same job in the production stage. Constraint (14) shows that completion time of each job in a specific position in the second assembly stage can not be less than the completion time of job in the previous position in the same stage. Constraint (15) specifies that the completion time of each job in a specific position in the second assembly stage can not be less than the completion time of the same job in the first assembly stage. Constraint (16) specifies that the maximum completion time of jobs is greater than the completion time each job. Constraints (17) and (18) show the range of decision variables values.

4. SOLUTION METHOD

As previously proven in the introduction section, the distributed production-assembly scheduling problem with hybrid flowshop in assembly stage is NP-Hard and solving large sizes is not possible using the model. Abtahi and Sahraeian [28] presented two-machine flow shop scheduling problem that is NP-Hard, too. Different methods are proposed to solve the distributed scheduling problem.

To solve the distributed permutation flow-shop scheduling problem, Fathollahi-Fard et al. [22] presented a meta-heuristic algorithm called Social Engineering Optimizer (SEO). Fathollahi-Fard et al. [29] gave a full explanation of how this algorithm works. A simple, intelligent and new single-solution algorithm that has just four main steps and three simple parameters to tune. Social Engineering Optimizer starts with two initial solutions divided into attacker and defender. The attacker obtains the rules of Social Engineering techniques to reach its desired goals [29].

Garey et al. [5] investigated the behavior of Scottish red deer in order to develop a new nature-inspired algorithm. The main inspiration of this meta-heuristic algorithm is to originate from an unusual mating behavior of Scottish red deer in a breading season. the red deer algorithm (RDA) is a population-based meta-heuristics, that starts with an initial population called red deers (RDs). Individuals in this population are separated into two types: hinds and male RDs. Besides, a harem is a group of female RDs. The general steps of this evolutionary algorithm are considered by the competition of male RDs to get the harem with more hinds via roaring and fighting behaviors.

In order to solve the mentioned scheduling problem, a GA is presented in this section. The GA is one of the most well-known evolutionary algorithms. Many papers have used this algorithm to solve the problem. Gholizadeh et al. [30] proposed a novel scenario-based GA for flexible flowshop scheduling. Li et al. [31] developed a GA for the flow shop scheduling problem. Noroozi and Mokhtari [32], Jia et al. [33, 34], Chang et al. [35], Tavakoli and Mahdizadeh [36] have used GA to solve scheduling problems. Maghzi et al. [37] used GA for multi objective scheduling problem. The GA algorithm has many applications in other fields. Abbasi and Rafiee [38] presented a parallel GA on the traveling salesman problem with Multi-core and Many-core Systems.

Eiben and Smith [39] summarized the main framework of the GA in 5 sections:

1. Representation
2. Recombination
3. Mutation
4. Parent selection
5. Survival selection

In this paper, we combine a classical GA with a local search algorithm to present a hybrid algorithm that is called a HGALPT. An heuristic algorithm that is called the Longest Processing Time (LPT) is used in order to improve the results in each iteration of the algorithm. The main steps of the improved GA in this paper are: providing the structure of the solution representation and creating the initial population, improving the solution using local search, parent selection, cross over, mutation, generation selection.

4. 1. Chromosome Representation and Initial Population

Each problem solution is called a chromosome in the GA. Each chromosome contains components that are as the problem inputs. An solution consists of two parts. The first part is related to jobs processing order in the first and second stages of each factory, which is shown in Figure 1, and the second part is related to determining the jobs processing sequence on third stage machines of each factory that is done by the decoding process. In Figure 1, the processing sequence of 7 jobs in 3 factories on the first and second stage machines are shown. The number zero is known as the factory separator. The number of zeros in the solution representation is equal to f-1 because the first factory does not need a separator, jobs 1 and 2 are processed in the same sequence on the first and second stage machines of the first factory, jobs 5, 4 and 6 in the second factory, and jobs 3 and 7 in the third factory.

After assigning the jobs to the factories and sequencing them in the first and second stages, according to Figure 2, the jobs are assigned to the third stage machines using decoding. Each job is assigned to a

1	2	0	5	4	6	0	3	7

Figure 1. Chromosome representation

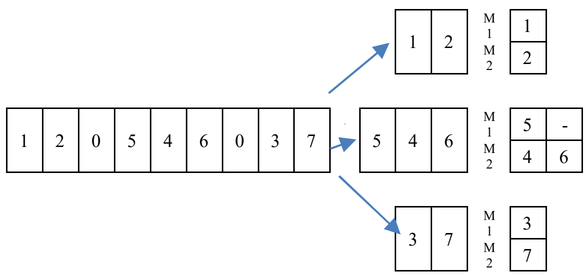

Figure 2. Decoding process

machine that has the least completion time. Figure 2 assumes that there are 2 machines in the third stage. In factory 2, job 5 is assigned to the first machine, job 4 to the second machine, and job 6 after job 4 to the second machine, which is supposed to have the shortest completion time. In order to produce the initial population, as a population size, jobs sequence with f-1 zeros is generated randomly.

4. 2. Local Search Based on the Longest Processing Time In order to improve the solutions, we have used of a heuristic method based on longest processing time. In this method, a local search is performed on some of the best solutions with P_{lpt} rate of the entire population. In the LPT method, jobs are arranged in descending order of processing times. For our problem, there are three stages of operations, different modes are considered for processing time. Each job is processed simultaneously on m_1 machine in the first stage, so in order to use the LPT algorithm, we considered the maximum processing time of each job as its processing time in the first stage. In the second and third stages, there is only one processing time for each job in each step. There are 6 modes for selection and arranging processing times as following:

1) $JS_1 = \max_K\{p_{j,k}\}$
2) $JS_2 = tt_j$
3) $JS_3 = pt_j$
4) $JS_4 = \max_K\{p_{j,k}\} + tt_j$
5) $JS_5 = tt_j + pt_j$
6) $JS_6 = \max_K\{p_{j,k}\} + tt_j + pt_j$

After determining the solutions at the end of each iteration of the algorithm, to perform the LPT, the assigned jobs to each factory are processed in descending of mode 1 (JS_1). This operations is performed in 6 modes JS_1 to JS_6 on a solution separately and the best solution is selected as an alternative to the current solution. The LPT heuristic algorithm accelerate reaching of better solutions in the main algorithm.

4. 3. SELECTION In the GA, the selection operator is the parents selection to perform the crossover, mutation, and create next generation. Some of solutions are transferred to the next generation with P_e rate, unchanged. In order to select the parents, the rank-based roulette wheel selection has been used [40]. In this mechanism, for each population solution, a rank based on its fitness value (the objective function value of each solution) is assigned, firstly. If PN indicates population size, the best solution rank is PN and the worst solution is 1. In order to better selection, the new rank of the parents is determined based on the linear relation (19):

$$NewRank\ (or) = 2 - SP + \left(2 * (SP-1) * \frac{(or-1)}{(PN-1)}\right) \qquad (19)$$

In the above statement, "or" is old rank, "NewRank (or)" is the new rank and SP is the selection pressure, the SP value is in the range [1.0, 2.0]. The parent selection probability is determined by rank or according to Equation (20).

$$Pr\ (or) = \frac{NewRank\ (or)}{\sum_{i=1}^{PN} NewRank\ (i)} \qquad (20)$$

"Pr (or)" is the probability of selecting a individual with the rank of "or".

4. 4. Crossover There are different types of crossovers to use in the algorithm according to the problem type. Various methods are presented for implementation of crossover in sequence-based representation. Xiong et al. [9] have used different crossover types. Deng et al. [6] used of sequential crossover to combine two parents and create new individual in the production-assembly problem, also, we have used this crossover type with P_c rate. In this method, two points are randomly selected on the parents. The contents between the two points in the first parent are passed directly to the first child. In order to fill blanks in the first child, start from the second point of the second parent and select the job that do not exist in the first child, respectively, and completing start from the second point of the first child. The same steps are done to produce second child. An example of an crossover is shown in Figure 3.

4. 5. Mutation After generating offspring by performing the crossover operator, a mutation operation with the P_m possibility is performed on each child. There are different types of mutation operator that one of them is swap. In this mutation type, two genes are selected and mutated. Figure 4 shows an example of this mutation type.

4. 6. Generation Selection The previous population and the new population are combined and the

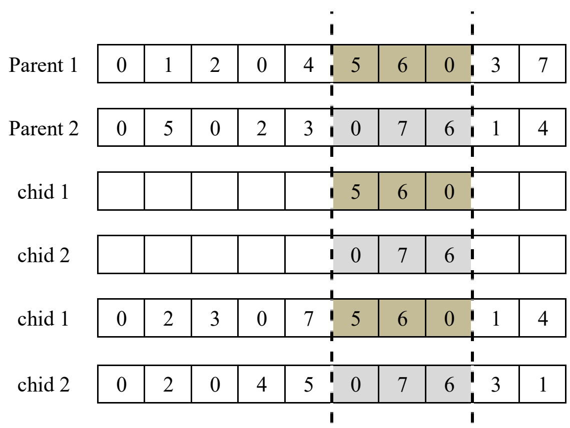

Figure 3. Crossover operator

Parent	0	2	0	4	5	0	7	6	3	1
Child	0	2	0	3	5	0	7	6	4	1

Figure 4. Mutation operator

next generation is selected among of this population individuals. To do this, the number of individuals who have the best fitness will be passed on to the next generation unchanged and with P_{ef} rate. For the rest of the next generation population, the roulette wheel mechanism is used. Here, in order to calculate the individual selection probability, the fitness function of each solution is used [40]. Equation (21) shows how to calculate each individual selection probability.

$$PrS(i) = \frac{Fitness\,(i)}{\sum_{j=1}^{PN} Fitness\,(j)} \qquad \forall\, i = 1,2,\dots,PN \tag{21}$$

where PrS(i) and Fitness (i) indicate the selection probability and fitness function value of individual i, respectively.

5. COMPARISONS AND CALCULATION RESULTS

The calculations results are presented in different cases, in this section. Given that there are no similar papers to our problem and the presented papers are different in the literature, we used of provided values by Xiong et al. [12, 17] to determine the problem parameter ranges, which are simpler but closer to the our problem. Table 2 summarizes the problem parameters values. The parameters p, tt and pt show the processing times of jobs in all stages and have a discrete uniform distribution in the interval [1.0, 100.0].

In order to evaluate the efficiency of the proposed algorithm, the results of the algorithm in small size have been compared with results of MILP model and the results have been extended to large size. The proposed GA is coded in Java IntelliJ IDEA 2020.1.2 software and IBM ILOG CPLEX solver Concert technology and problem model in GAMS 28.2.0 software and CPLEX solver. All calculations are performed on the computer with Intel (R) Core (Tm) i5-323M CPU @ 2.60 GHz 6.0 GB specifications.

Potts et al. [26] showed that two-stage assembly scheduling problem is NP-Hard and Single machine scheduling problem with considering the sequence dependent set-up times are classified in NP-Hard problems [41-43], then given that the our problem is three-stage and multi-factory mode, so, is more complex than [26]. Therefore, according to our problem complexity, it proves to be NP-Hard.

Optimal solution for small sizes can be achieved using the Mixed Integer Linear Programming (MILP) model, but for large sizes, optimal or near-optimal solutions can be achieved by a HGALPT. Due to the mentioned cases and high complexity of the problem, solving the problem using the Cplex has a memory error for large sizes, but this issue does not exist in the proposed GA. For small sizes the time limit is 3600 seconds in Cplex and 0.5*n*f seconds limit in HGALPT. For large sizes the 1200 seconds limit is considered as algorithm stop criterion.

In order to evaluate the performance of the algorithm, three parts are presented: the first part is related to parameters setting for small and large sizes of the problem, the second part is related to comparing the results of mathematical model and improved GA and shows the superiority of the proposed algorithm. After proving the superiority of the improved GA over the mathematical model, The third part compares the results of the GA and the HGALPT that the HGALPT results are better.

TABLE 2. Parameter values to create problem instance

Parameter	Parameter values ranges	
	Small size	Large size
f	{2,3,4}	{4,6,8}
n	{5,6,7,8,10}	{20,30,40,60,80,100}
m_1	{2,3,4}	{2,4,6,8}
m_3	{2,3}	{3,4,5}
p	U(1,100)	U(1,100)
tt	U(1,100)	U(1,100)
pt	U(1,100)	U(1,100)

5. 1. Parameter Setting

In order to determine the parameters of the problem, more accurately, Analysis Of Variance (ANOVA) of Taguchi method for small and large sizes has been used. Seven parameters affect on the algorithm results that three levels are defined for each parameter. Problem parameters include initial population (PS), next generation elite rate (P_e), local search rate (P_{lpt}), crossover rate (P_c), mutation rate (P_m), next generation transmission rate (P_{ef}), selection pressure (SP). Their levels valeues are shown in Table 3. The value of crossover rate parameter is considered equal to the complement of the mutation rate parameter, so it is not in the parameter setting calculations.

We have used Minitab software, in order to determine each parameter level. an instance of small size with values of n = 8, f = 3, $m_1 = 4$, $m_3 = 2$ and an instance of large size with values of n = 40, f = 4, $m_1 = 4$, $m_3 = 3$ is examined. For small size, using orthogonal matrix $L_{27} = (3^6)$, 27 different combinations of parameter levels are specified in Table 4. Relative Error (RE) and Average Relative Error (ARE) have been used to compare the results of the algorithms, that are presented

TABLE 3. Parameter levels

Parameter	Level	Value	Parameter	Level	Value
	1	30		1	0.07
PS	2	50	P_m	2	0.09
	3	80		3	0.11
	1	0.02		1	0.01
P_e	2	0.04	P_{ef}	2	0.03
	3	0.06		3	0.06
	1	0.02		1	1.3
P_{lpt}	2	0.05			
	3	0.07	SP	2	1.5
	1	0.93			
P_c	2	0.91		3	1.7
	3	0.89			

TABLE 4. ARE value according to the orthogonal array $L_{27} = (3^6)$ for small size

Experiment number	Parameters level						ARE
	PS	P_e	P_{lpt}	P_m	P_{ef}	SP	
1	1	1	1	1	1	1	0.0159
2	1	1	1	1	2	2	0.0019
3	1	1	1	1	3	3	0.0070
4	1	2	2	2	1	1	0.0165
5	1	2	2	2	2	2	0.0037
6	1	2	2	2	3	3	0.0014
7	1	3	3	3	1	1	0.0122
8	1	3	3	3	2	2	0.0072
9	1	3	3	3	3	3	0.0000
10	2	1	2	3	1	2	0.0113
11	2	1	2	3	2	3	0.0019
12	2	1	2	3	3	1	0.0043
13	2	2	3	1	1	2	0.0000
14	2	2	3	1	2	3	0.0000
15	2	2	3	1	3	1	0.0041
16	2	3	1	2	1	2	0.0027
17	2	3	1	2	2	3	0.0000
18	2	3	1	2	3	1	0.0014
19	3	1	3	2	1	3	0.0000
20	3	1	3	2	2	1	0.0029
21	3	1	3	2	3	2	0.0019
22	3	2	1	3	1	3	0.0048
23	3	2	1	3	2	1	0.0027
24	3	2	1	3	3	2	0.0023
25	3	3	2	1	1	3	0.0048
26	3	3	2	1	2	1	0.0000
27	3	3	2	1	3	2	0.0033

by Xiong and Xing [17]. The value of RE is calculated using Equation (22).

$$RE = \frac{(Z-BEST)}{BEST} * 100 \tag{22}$$

where BEST is the best value obtained from each of the algorithms and Z is the value obtained from the execution of an instance of the problem. Each parameters combination is executed 20 times independently, the obtained ARE is given in Table 4. Table 5 summarizes the mean and standard deviation (StDev) results based on the 95% confidence level for different parameters levels in small size.

According to Table 5, PS for small size, the mean of ARE value decreases with increasing population size. The the mean of ARE difference between levels 2 and 3 is not significant, but due to the fact that the scatter in level 3 is less, this level with a value of PS = 80 is selected as the best level. For the P_{ef}, the scattering around the mean is significant also according to Table 5 with increasing the elite rate value, the response value improves to level two but at level three, the results are not better than at level two and intensification increases, therefore, P_{ef} = 0.03 is considered.

According to the results in Table 5 for the SP, the results of the algorithm are improved by changing and increasing the parameter value. In fact, increasing the selection pressure is appropriate for small sizes and

TABLE 5. Individual 95% CIs For Mean Based on Pooled StDev for small size

Parameter	Level	N	Mean	StDev
	1	9	0.0073	0.0063
PS	2	9	0.0029	0.0036
	3	9	0.0025	0.0017
	1	9	0.0052	0.0053
$\mathbf{P_e}$	2	9	0.0039	0.0050
	3	9	0.0035	0.0041
	1	9	0.0043	0.0048
$\mathbf{P_{lpt}}$	2	9	0.0052	0.0053
	3	9	0.0031	0.0042
	1	9	0.0041	0.0051
$\mathbf{P_m}$	2	9	0.0034	0.0051
	3	9	0.0052	0.0042
	1	9	0.0076	0.0065
$\mathbf{P_{ef}}$	2	9	0.0023	0.0023
	3	9	0.0029	0.0021
	1	9	0.0067	0.0064
SP	2	9	0.0038	0.0034
	3	9	0.0022	0.0027

improves the algorithm performance. The appropriate value of this parameter is SP = 1.7. For P_e, P_{lpt} and P_m, the mean of ARE results at Table 5 show no significant effect on the algorithm results and the mean of ARE value by changing the parameter level does not change significantly. According to the mean and standard deviation in Table 5, For small sizes, the parameters values in response to the best values of the levels are: PS=80, P_e=0.06, P_{lpt}=0.07, P_m=0.09, P_c=0.91, P_{ef}= 0.03 and SP=1.7.

For large size, An instance of the general problem with the values n = 40, f = 4, $m_1 = 4$, $m_3 = 3$ is used to set the parameters. Like small size, the different combinations of parameters is adjusted according to Table 6 in 27 rows by using the orthogonal matrix of the Taguchi method. Each row is run 15 times and the obtained results ARE is calculated. According to the mean of ARE in Table 7, different levels for SP, P_e, P_{ef} and PS parameters have a significant effect on the algorithm results. With increasing the PS parameter value, the mean of ARE values have improved. Due to the increase in the size of the problem, the increase in population has led to the investigate more solution areas and the results are improved, so the value of the parameter PS is considered equal to PS = 80.

According to Table 7, for the P_e parameter and transferring some of the best solutions without crossover and mutation to the next generation, the best value of this parameter is equal to P_e = 0.02. For the P_{ef} parameter, which indicates the elite rate for unchanged transmission to the next generation, as shown in Table 7, the mean of ARE of the obtained results has improved with increasing rate value. This rate is related to the intensification in the problem, directly, which means that by increasing this parameter value, the amount of algorithm intensification also increases. The appropriate value of this parameter according to the Table 7 is equal to P_{ef}= 0.06.

The next parameter that its different values affects to the problem results is SP, which its low or high value indicates an increase in diversification or intensification. In this parameter, by increasing of the parameter value, the mean of ARE of the obtained results increases according to Table 7 and the results become worse. Therefore, its appropriate value is equal to SP = 1.3. Finally, according to the mean and standard deviation ARE Table 7, The parameters values based on the best levels value are: PS=80, P_e=0.02, P_{lpt}=0.05, P_m=0.11, P_c=0.89, P_{ef}=0.06 and SP=1.3.

TABLE 6. Average response value according to the orthogonal array $L_{27} = (3^6)$ for large size

Experiment number	Parameters level						ARE
	PS	P_e	P_{lpt}	P_m	P_{ef}	SP	
1	1	1	1	1	1	1	0.0631
2	1	1	1	1	2	2	0.0463
3	1	1	1	1	3	3	0.0395
4	1	2	2	2	1	1	0.0561
5	1	2	2	2	2	2	0.0672
6	1	2	2	2	3	3	0.0484
7	1	3	3	3	1	1	0.0753
8	1	3	3	3	2	2	0.0391
9	1	3	3	3	3	3	0.0622
10	2	1	2	3	1	2	0.0405
11	2	1	2	3	2	3	0.0341
12	2	1	2	3	3	1	0.0288
13	2	2	3	1	1	2	0.0519
14	2	2	3	1	2	3	0.0656
15	2	2	3	1	3	1	0.0403
16	2	3	1	2	1	2	0.0631
17	2	3	1	2	2	3	0.0563
18	2	3	1	2	3	1	0.0362
19	3	1	3	2	1	3	0.0647
20	3	1	3	2	2	1	0.0299
21	3	1	3	2	3	2	0.0407
22	3	2	1	3	1	3	0.0350
23	3	2	1	3	2	1	0.0294
24	3	2	1	3	3	2	0.0524
25	3	3	2	1	1	3	0.0451
26	3	3	2	1	2	1	0.0532
27	3	3	2	1	3	2	0.0344

5. 2. CPLEX and HGALPT Results Comparison for Small Size

In order to evaluate the algorithm efficiency, its results are compared with the Cplex exact solver in 20 different sizes. The results of the calculations are shown in Table 8. where the solution quality is equal to the difference percentage from the best obtained solution. Cplex calculations are performed with 3600 seconds time limit. If the value in the column "Time" (column 7) is 3600, the value in the column "Quality" is the deviation percentage of best feasible solution that obtain from Cplex, Otherwise, the column "Quality" value is the deviation percentage of optimal solution.

For HGALPT algorithm, each instance is run 20 times. The column "Min" shows the minimum relative error, the column "ARE" shows the average relative error, the column "Max" shows the maximum relative error for the each instance, the column "STD" shows the standard deviation for the relative errors and the column "Time" shows the time limit of 0.5*n*f seconds.

From 20 instances, in 14 instance optimal solution are calculated by CPLEX, 3 instances are out of memory

TABLE 7. Individual 95% CIs For Mean Based on Pooled StDev for large size

Parameter	Level	N	Mean	StDev
	1	9	0.0553	0.0127
PS	2	9	0.0463	0.0133
	3	9	0.0428	0.0121
	1	9	0.0431	0.0130
Pe	2	9	0.0496	0.0129
	3	9	0.0517	0.0140
	1	9	0.0468	0.0126
Plpt	2	9	0.0453	0.0123
	3	9	0.0522	0.0155
	1	9	0.0488	0.0106
Pm	2	9	0.0514	0.0134
	3	9	0.0441	0.0160
	1	9	0.0550	0.0130
Pef	2	9	0.0468	0.0146
	3	9	0.0426	0.0102
	1	9	0.0458	0.0168
SP	2	9	0.0484	0.0112
	3	9	0.0501	0.0126

error (as indicated by OM) and 3 instances provided the best solution according to time limit of 3600 seconds, according to Table 8. from the 14 optimal solution instance obtained by CPLEX, 12 instance of the HGALPT algorithm have reached the optimal solution in a much shorter time which indicates the efficiency of the algorithm. Also, since the optimal solution has been reached, the effectiveness of the algorithm is also guaranteed.

The average of columns values for Table 8 are calculated for the total instances at the Table bottom. For comparing the solution time of the two algorithms, the average of CPLEX solution time is 1419.3005 seconds while the average of HGALPT solution time is much less and is equal to 11.225. As the instance size increases, the solution time increases and the CPLEX needs more time to reach the optimal solution or CPLEX can not provide

TABLE 8. Comparison of CPLEX results and HGALPT algorithm for small sizes

Instance	n	f	m_1	m_3	CPLEX		HGALPT				
					Quality	Time (s)	Min	ARE	Max	STD	Time (s)
1	5	2	2	3	0.000	4.87	0.000	0.000	0.000	0.000	5
2	5	3	4	2	0.000	5.84	0.000	0.000	0.000	0.000	7.5
3	6	2	2	3	0.000	31	0.000	0.000	0.000	0.000	6
4	6	3	3	2	0.000	18	0.000	0.000	0.000	0.000	9
5	6	4	4	2	0.000	44	0.000	0.000	0.000	0.000	12
6	7	2	2	3	0.000	381	0.000	0.000	0.000	0.000	7
7	7	2	3	3	0.000	398	0.000	0.001	0.011	0.002	7
8	7	3	3	2	0.000	235	0.000	0.000	0.000	0.000	10.5
9	7	3	4	2	0.000	149	0.000	0.000	0.000	0.000	10.5
10	7	4	4	2	0.000	509	0.000	0.000	0.000	0.000	14
11	8	2	2	3	0.000	2379	0.000	0.007	0.025	0.008	8
12	8	2	4	3	0.000	1340	0.000	0.010	0.043	0.012	8
13	8	3	3	2	0.000	1667	0.000	0.000	0.000	0.000	12
14	8	3	4	2	0.000	3487	0.000	0.003	0.029	0.009	12
15	8	4	2	2	0.000	3600	0.000	0.000	0.000	0.000	16
16	10	2	3	3	0.012 (OM)	3107	0.000	0.012	0.031	0.011	10
17	10	3	2	3	0.000	3600	0.000	0.010	0.027	0.008	15
18	10	3	4	2	0.000	3600	0.000	0.014	0.057	0.015	15
19	10	4	2	2	0.016 (OM)	2411	0.000	0.001	0.016	0.004	20
20	10	4	3	2	0.037 (OM)	1593	0.000	0.017	0.066	0.020	20
Average					0.0033	1427.9855	0.0000	0.0037	0.0152	0.0044	11.2250

the optimal solution and faces a time limit or out of memory error while the HGALPT algorithm is reached to solution in much less time. In view of the above, the HGALPT algorithm is more efficient than CPLEX.

5. 3. GA and HGALPT Results Comparison for Large Size Given the efficiency and effectiveness for small size problems, to prove the effectiveness of the HGALPT algorithm, the results for large size are calculated and the GA and HGALPT algorithms are compared. To prove the algorithm effectiveness, comparison of GA algorithm without LPT and using it has been done. 30 instances of problem parameters combining have been created to compare algorithms. Each algorithm is executed 5 times with a time limit of 1200 seconds and the results are recorded. The minimum relative error (Min), average relative error (ARE), maximum relative error (Max) and standard deviation for the relative error (STD) for each instance are listed in Table 9.

TABLE 9. Comparison of GA and HGALPT algorithms for large sizes

Instance	n	f	m_1	m_3	GA				HGALPT			
					Min	ARE	Max	STD	Min	ARE	Max	STD
1	20	4	2	3	0.067	0.110	0.147	0.035	0	0.031	0.086	0.034
2	20	4	4	4	0.015	0.064	0.121	0.038	0	0.033	0.051	0.021
3	20	4	6	5	0.011	0.060	0.091	0.032	0	0.024	0.057	0.023
4	20	6	2	4	0.000	0.041	0.068	0.029	0.019	0.035	0.080	0.025
5	20	6	4	3	0.076	0.104	0.132	0.027	0	0.017	0.052	0.020
6	20	6	8	3	0.054	0.088	0.120	0.030	0	0.033	0.094	0.037
7	20	8	8	3	0.026	0.125	0.168	0.057	0	0.011	0.033	0.013
8	30	4	2	3	0.033	0.053	0.077	0.016	0	0.007	0.018	0.007
9	30	4	4	4	0.04	0.079	0.130	0.044	0	0.015	0.036	0.017
10	30	4	6	5	0.043	0.100	0.161	0.043	0	0.020	0.039	0.019
11	30	6	8	3	0.032	0.089	0.174	0.059	0	0.025	0.051	0.021
12	30	8	6	3	0.056	0.111	0.147	0.037	0	0.022	0.038	0.015
13	40	4	2	3	0.052	0.086	0.115	0.024	0	0.015	0.026	0.011
14	40	4	4	4	0.099	0.110	0.120	0.010	0	0.013	0.035	0.014
15	40	4	6	5	0.076	0.106	0.129	0.020	0	0.013	0.024	0.009
16	40	6	8	3	0.115	0.185	0.241	0.046	0	0.017	0.036	0.014
17	40	8	6	3	0.074	0.113	0.143	0.027	0	0.008	0.021	0.009
18	60	4	2	3	0.111	0.141	0.185	0.027	0	0.020	0.042	0.019
19	60	4	4	4	0.087	0.122	0.142	0.022	0	0.010	0.025	0.009
20	60	6	2	4	0.128	0.147	0.183	0.024	0	0.013	0.032	0.012
21	60	6	8	3	0.130	0.157	0.175	0.019	0	0.037	0.110	0.044
22	60	8	4	3	0.076	0.144	0.187	0.046	0	0.017	0.040	0.016
23	80	4	2	3	0.027	0.074	0.104	0.028	0	0.012	0.028	0.010
24	80	4	6	5	0.079	0.088	0.109	0.012	0	0.012	0.030	0.012
25	80	6	8	3	0.093	0.154	0.212	0.044	0	0.010	0.019	0.008
26	80	6	4	5	0.111	0.133	0.167	0.022	0	0.011	0.017	0.007
27	80	8	6	3	0.137	0.187	0.226	0.040	0	0.013	0.024	0.009
28	100	4	2	3	0.066	0.094	0.134	0.027	0	0.011	0.017	0.007
29	100	4	4	4	0.048	0.079	0.114	0.031	0	0.013	0.035	0.015
30	100	6	2	5	0.073	0.164	0.308	0.102	0	0.028	0.052	0.021
Average					0.068	0.110	0.151	0.034	0.001	0.018	0.042	0.017

According to Table 9, from 30 instances, 29 instances of the minimum relative error, 30 instances of the ARE value, 29 instances of the maximum relative error and 27 instances of the standard deviation relative error, HGALPT values are less than GA, therefore, the effectiveness of HGALPT algorithm is determined against GA. The overall average is calculated for all columns in Table 9. The avrage of minimum relative error, ARE, maximum relative error and standard deviation for all instances for GA and HGALPT is {0.068, 0.110, 0.151, 0.034} and {0.001, 0.018, 0.042, 0.017}, respectively.

Therefore, for all four modes of minimum, average, maximum and standard deviation, the HGALPT algorithm has lower values than GA, indicating that the scatter of solutions around the best solution in HGALPT is less, and given that in most instances the results of HGALPT are better than those of GA, as a result, the HGALPT algorithm is more effectiveness.

5. 3. Sensitivity Analyses Sensitivity analysis is performed in two ways:

1- Investigating the effect of different values of the parameters on the makespan objective function.
2- Performance analysis of HGALPT and GA for different values of problem parameters.

The effective parameters in the makespan objective function are: n, f, m_1 and m_3, which, by keeping constant the other parameters, their effect on the objective function value can be examined. For the first sensitivity analysis, to examine the effect of each parameter, four different scenarios are considered. S (n), S (f), S(m_1) and S(m_3) are the scenarios related to the parameters n, f, m_1 and m_3, respectively, which are summarized in Table 10. In this table, for each scenario, assuming the other parameters are constant, the effect of different values of a parameter on the makespan objective function is specified.

As it can be seen in Figure 5, for the parameter of n, for different scenarios, by increasing the value of n, the value of the makespan increases in the second scenario, but in the third and fourth scenarios remains constant. The objective function is fixed for the values 6, 7 and 8 and can take any of n diffrent values. In fact, with fixed equipment and costs, more jobs can be processed. For parameter f, as shown in Figure 6, by increasing the value of this parameter, the value of the objective function is decreases in 2 and 3 scenarios then fixed for the fourth scenarios (makespan value 242). This means that in order to reduce costs, the number of factories can be reduced, assuming that other parameters are constant.

For parameter m_1, As be seen in Figure 7, the first and second scenarios have the same objective function values, and decision-makers can choose any of these scenarios as needed. For the third scenario, the value of

TABLE 10. Sensitivity Analyses on the parameters

Parameter scenario	Parameters				Objective
	n	f	m_1	m_3	Cmax
S(n)	5	2	2	2	260
	6	2	2	2	278
	7	2	2	2	278
	8	2	2	2	278
S(f)	8	1	2	2	455
	8	2	2	2	278
	8	3	2	2	242
	8	4	2	2	242
S(m_1)	8	2	1	2	278
	8	2	2	2	278
	8	2	3	2	281
	8	2	4	2	281
S(m_3)	8	2	2	1	307
	8	2	2	2	278
	8	2	2	3	278
	8	2	2	4	278

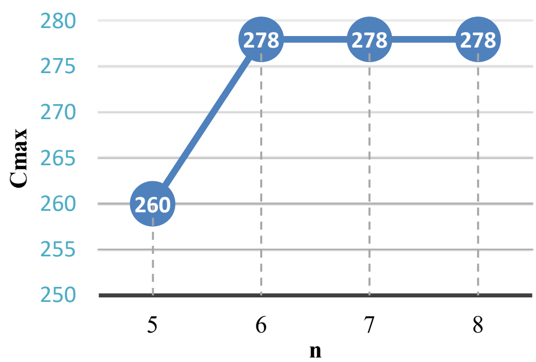

Figure 5. Cmax overlap vs. n

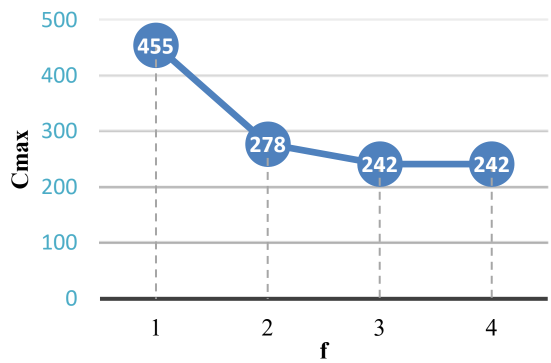

Figure 6. Cmax overlap vs. f

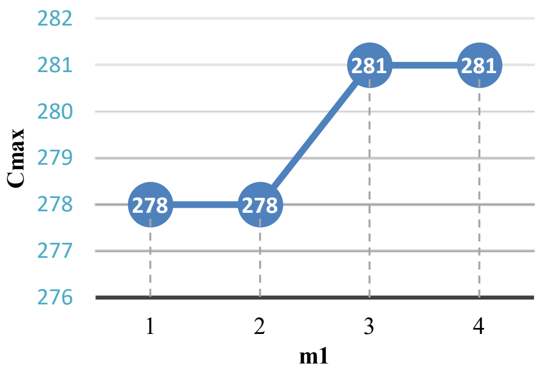

Figure 7. Cmax overlap vs. m_1

the objective function changes and its value is equal to the fourth scenario, so either of these two scenarios can be used. The last parameter is m_3, According to Figure 8, in which the value of the objective function remains constant from the second scenario onwards, ie increasing the number of machines in the third stage has no effect on the value of the objective function, and fewer machines are needed for the constant values of other parameters.

For the second sensitivity analysis, in order to evaluate the HGALPT algorithm and compare the degree of deviation of the solutions of the two algorithms GA and HGALPT, some possible sensitivity analyzes have been performed in this section. For this purpose, for each parameter, the percentage of relative deviation is calculated for different values of that parameter. Figure 9 shows the performance of the two algorithms based on the ARE for different values of jobs. In all sizes, the ARE value in HGALPT algorithm is less than GA and the superiority of HGALPT algorithm is obvious. Figure 10 shows a performance comparison of the two algorithms based on the ARE for factories different values. As the number of factories increases, the ARE in the GA increases.

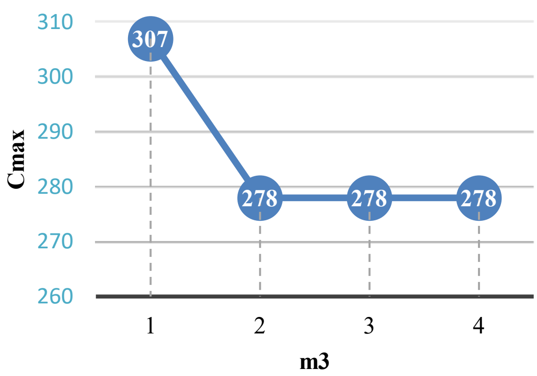

Figure 8. Cmax overlap vs. m_3

Figures 11 and 12 show change rate of the ARE per number of machines in the first and third stages. In these figures, the relative error rate of HGALPT algorithm is less than GA. Assuming the values of the parameters f, m_1 and m_3 are constant, the ARE for different values of n in two proposed algorithms is as shown in Figure 13. Four different scenarios SC_1 to SC_4 are as follows:

- SC_1: f=4 , $m_1 = 2$, $m_3 = 3$
- SC_2: f=4 , $m_1 = 4$, $m_3 = 4$
- SC_3: f=4 , $m_1 = 6$, $m_3 = 5$
- SC_4: f=6 , $m_1 = 8$, $m_3 = 3$

According to Figure 13, the HGALPT algorithm performs better than the GA for different values of n in

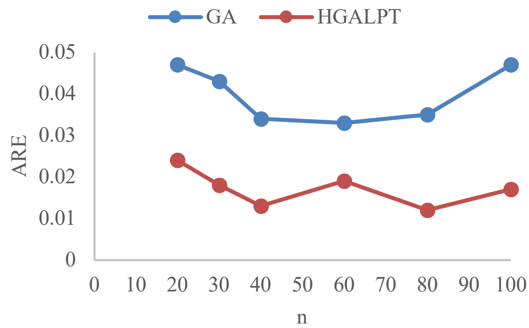

Figure 9. ARE overlap vs. n

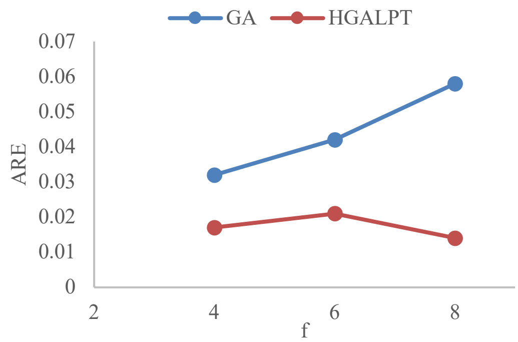

Figure 10. ARE overlap vs. f

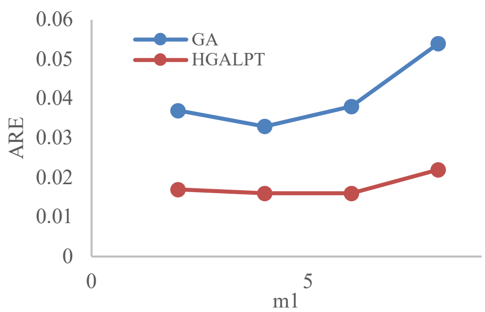

Figure 11. ARE overlap vs. m_1

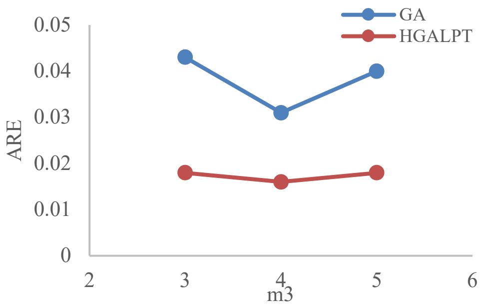

Figure 12. ARE overlap vs. m_3

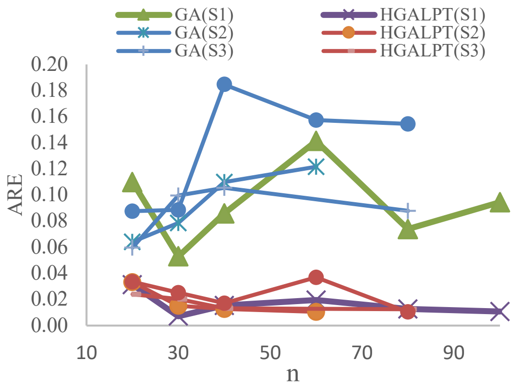

Figure 13. ARE overlap vs. S_1, S_2, S_3, S_4

the four defined scenarios. According to the mentioned cases, the superiority of HGALPT algorithm over GA is determined. Also, the superiority of HGALPT algorithm over Cplex for small sizes was investigated due to less solution time. Therefore, the efficiency of the algorithm in order to solve the problem is proved in this paper.

6. CONCLUSION

In the classical production-assembly problems, one step is often considered for production and one step for assembly, while in the real world, the number of steps can be more (transportation, packaging, etc.). Here we have three steps. However, in order to take advantage of cheap labor or available space, it is possible to build factories in several geographical locations, so the issue is considered as a multi-factory. In order to reduce the downtime in the third stage, parallel machines are assumed in this stage.

In order to fill the gaps raised, this research is the first study in the distributed field, that investigates distributed production-assembly scheduling with hybrid flowshop in assembly stage. In order to solve the problem, a mathematical model is presented with minimizing the maximum completion time. Due to the high complexity of the proposed mathematical model, it was proved that this problem is Non-deterministic Polynomial-time hard (NP-hard). Therefore, the mathematical model can not to solve it in large scale.

Due to the problem is NP-hard, a GA has been proposed. GAs have been used in many papers. An improved mode of this algorithm is called Hybrid Genetic Algorithm Longest Proseccing Time (HGALPT) algorithm is used to solve the problem in large scale. Using analysis of variance (ANOVA), the values of all parameters for small and large sizes is determined. The results of HGALPT algorithm compared with the results of mixed integer linear programming model (MILP) in small size and the results of genetic algorithm (GA) is compared with its improved algorithm HGALPT for large size which shows the efficiency of the proposed algorithm.

This research findings are: examining the possibility of several machines in the third stage of production-assembly problem, using the three stage in production-assembly problem, considering parallel factories in the three stage production-assembly problem with parallel machines in the third stage, new mathematical model to solve the problem, adjust the parameters and select the best values to run the algorithm, provide a suitable algorithm to solve the problem in large sizes and check the accuracy of the results by comparing the results of the mathematical model and finally, provide sensitivity analyses Table and graphs of the relative error percentage for different values of the problem parameters and sensitivity analyses of objective function.

Some of the limitations that we encountered in this paper, as follows: the lack of easy access to some of the papers in the problem literature, the impossibility of referring to some production environments due to the distance and the limitations of the disease outbreak. Based on the results of the research, some recommendations include the following: The first issue is to provide market research reports to management about presentation a service or product to prevent of failure. The next issue in the production is the possibility of factories building, purchasing equipment and budget control by management.

Given the above, the next issue is to consider the cost items in the proposed model and costs control. Better results are obtained if the results of the presented problem are combined with cost items and managerial insights.
Finally, although the presented problem in this study has not been studied so far and is highly complex, some suggestions for future work include:

- Improving the proposed solution method by combining it with other crossover and mutation methods.
- Applying other objective functions including jobs tardiness or earliness as used.
- considering different parallel machines in the second assembly stage.

- Using other meta-heuristic algorithms such as Social Engineering Optimizer (SEO), Red Deer Algorithm (RDA) and Biogeography-Based Optimization. (BBO) to solve the problem and compare the results with the algorithm in this paper.

6. REFERENCES

1. Babaeimorad. S, Fatthi. P, Fazlollahtabar. H, "A Joint Optimization Model for Production Scheduling and Preventive Maintenance Interval." ***International Journal of Engineering, Transactions B: Applications,*** Vol. 34, No. 11 (2021), 2508-2516, DOI: 10.5829/IJE.2021.34.11B.13.
2. Shoaardebili, N., Fattahi, P. "Multi-objective meta-heuristics to solve three-stage assembly flow shop scheduling problem with machine availability constraints." ***International Journal of Production Research***, Vol. 53, No. 3, (2015), 944-968. DOI: https://doi.org/10.1080/00207543.2014.948575.
3. Ahmadi, R., Bagchi, U., Roemer, T. A. "Coordinated scheduling of customer orders for quick response." ***Naval Research Logistics (NRL),*** Vol. 52, No. 6 (2005), 493-512. DOI: https://doi.org/10.1002/nav.20092.
4. Lee, C. Y., Cheng, T. C. E., Lin, B. M. T. "Minimizing the makespan in the 3-machine assembly-type flowshop scheduling problem." ***Management Science***, Vol. 39, No. 5, (1993), 616-625. DOI: https://doi.org/10.1287/mnsc.39.5.616.
5. Garey, M. R., Johnson, D. S., Sethi, R. "The complexity of flowshop and jobshop scheduling." ***Mathematics of Operations Research,*** Vol. 1, No. 2 (1976), 117-129. DOI: https://doi.org/10.1287/moor.1.2.117
6. Deng, J., Wang, L., Wang, S. Y., Zheng, X. L. "A competitive memetic algorithm for the distributed two-stage assembly flow-shop scheduling problem." ***International Journal of Production Research***, Vol. 54, No. 12 (2016), 3561-3577. DOI: https://doi.org/10.1080/00207543.2015.1084063.
7. Hatami, S., García, R. R., Romano, C. A. "The Distributed Assembly Parallel Machine Scheduling Problem with eligibility constraints." ***International Journal of Production Management and Engineering***, Vol. 3, No. 1, (2015), 13-23. DOI: https://doi.org/10.4995/ijpme.2015.3345
8. Framinan, J. M., Perez-Gonzalez, P. "Order scheduling with tardiness objective: Improved approximate solutions." ***European Journal of Operational Research***, Vol. 66, No. 3 (2018). 840-850, DOI: https://doi.org/10.1016/j.ejor.2017.10.064.
9. Xiong, F., Xing, K., Wang, F., Lei, H., Han, L. "Minimizing the total completion time in a distributed two stage assembly system with setup times." ***Computers & Operations Research***, Vol. 47, (2014), 92-105. DOI: https://doi.org/10.1016/j.cor.2014.02.005.
10. Allahverdi, A., Al-Anzi, F. S. "Evolutionary heuristics and an algorithm for the two-stage assembly scheduling problem to minimize makespan with setup times." ***International Journal of Production Research,*** Vol. 44, No. 22, (2006), 4713-4735. DOI: https://doi.org/10.1080/00207540600621029.
11. Zhang, Y., Zhou, Z., Liu, J. "The production scheduling problem in a multi-page invoice printing system." ***Computers & Operations Research***, Vol, 37, No. 10, (2010), 1814-1821. DOI: https://doi.org/10.1016/j.cor.2010.01.014.
12. Xiong, F., Xing, K., Wang, F. "Scheduling a hybrid assembly-differentiation flowshop to minimize total flow time". ***European Journal of Operational Research,*** Vol, 240, No. 2, (2015), 338-354. DOI: https://doi.org/10.1016/j.ejor.2014.07.004.
13. Liu, B., Wang, K., Zhang, R. (2016, July). "Variable neighborhood based memetic algorithm for distributed assembly permutation flowshop." In Congress on Evolutionary Computation**,** (2016), 1682-1686. DOI: 10.1109/CEC.2016.7743990.
14. Sung, C. S., Kim, H. A. "A two-stage multiple-machine assembly scheduling problem for minimizing sum of completion times." ***International Journal of Production Economics***, Vol, 113, No. 2, (2008), 1038-1048. DOI: https://doi.org/10.1016/j.ijpe.2007.12.007.
15. Maboudian, Y., Shafaei, R. "Modeling a bi-criteria two stage assembly flow shop scheduling problem with sequence dependent setup times." In International Conference on Industrial Engineering and Engineering Management*,* (2009), 1748-1752, DOI**:** 10.1109/IEEM.2009.5373156.
16. Fattahi, P., Hosseini, S., Jolai, F. "Some heuristics for the hybrid flow shop scheduling problem with setup and assembly operations." ***International Journal of Industrial Engineering Computations***, Vol. 4, No. 3 (2013), 393-416. DOI: 10.5267/j.ijiec.2013.03.004.
17. Xiong, F., Xing, K. "Meta-heuristics for the distributed two-stage assembly scheduling problem with bi-criteria of makespan and mean completion time." ***International Journal of Production Research,*** Vol. 52, No. 9 (2014), 2743-2766. DOI: https://doi.org/10.1080/00207543.2014.884290.
18. Mahdavi, I., Komaki, G. M., Kayvanfar, V. "Aggregate hybrid flowshop scheduling with assembly operations." In ***18th International Conference on Industrial Engineering and Engineering Management,*** (2011), 663-667. DOI**:** 10.1109/ICIEEM.2011.6035244.
19. Pan, Q. K., Gao, L., Xin-Yu, L., Jose, F. M. "Effective constructive heuristics and meta-heuristics for the distributed assembly permutation flowshop scheduling problem." ***Applied Soft Computing***, Vol. 81, (2019), 105492. DOI: https://doi.org/10.1016/j.asoc.2019.105492.
20. Huang, J., Gu, X. "Distributed assembly permutation flow-shop scheduling problem with sequence-dependent set-up times using a novel biogeography-based optimization algorithm." ***Engineering Optimization***, (2021), 1-21. DOI: https://doi.org/10.1080/0305215X.2021.1886289.
21. Lei, D., Yuan, Y., Cai, J. "An improved artificial bee colony for multi-objective distributed unrelated parallel machine scheduling." ***International Journal of Production Research***, Vol. 59, No. 17, (2021), 5259-5271. DOI: https://doi.org/10.1080/00207543.2020.1775911.
22. Fathollahi-Fard, A. M., Woodward, L., Akhrif, O. Sustainable distributed permutation flow-shop scheduling model based on a triple bottom line concept. ***Journal of Industrial Information Integration,*** Vol 24, 100233. (2021). https://doi.org/10.1016/j.jii.2021.100233
23. Wang, W., Fathollahi-Fard, A. M. "Multi-objective low-carbon hybrid flow shop scheduling via an improved teaching-learning-based optimization algorithm." ***Scientia Iranica,*** (2022), DOI: 10.24200/SCI.2022.58317.5665
24. Hosseini, S. M. H. "Distributed assembly permutation flow-shop scheduling problem with non-identical factories and considering budget constraints." ***Kybernetes***, (2022). DOI: https://doi.org/10.1108/K-11-2021-1112
25. Jabbari, M., Tavana, M., Fattahi, P., Daneshamooz, F. "A parameter tuned hybrid algorithm for solving flow shop scheduling problems with parallel assembly stages." ***Sustainable Operations and Computers,*** Vol. 3, (2022). 22-32. DOI: https://doi.org/10.1016/j.susoc.2021.09.002
26. Potts, C. N., Sevast'Janov, S. V., Strusevich, V. A., Van Wassenhove, L. N., Zwaneveld, C. M. "The two-stage assembly scheduling problem: complexity and approximation." ***Operations***

Research, Vol. 43, No. 2, (1995), 346-355. DOI: http://dx.doi.org/10.1287/opre.43.2.346.

27. Jabbari, M., Azizi, V. "Lot streaming in no-wait multi product flowshop considering sequence dependent setup times and position based learning factors." ***International Journal of Engineering, Transactions A: Basics,*** Vol. 28, No. 7, (2015), 1031-1039. DOI: 10.5829/idosi.ije.2015.28.07a.09

28. Abtahi, Z., Sahraeian, R. (2021). "Robust and Stable Flow Shop Scheduling Problem Under Uncertain Processing Times and Machines Disruption." ***International Journal of Engineering Transactions A: Basics,*** Vol. 34, No. 4, (2022). DOI: 10.5829/ije.2021.34.04a.20

29. Fathollahi-Fard, A. M., Hajiaghaei-Keshteli, M., Tavakkoli-Moghaddam, R. "The social engineering optimizer (SEO)." ***Engineering Applications of Artificial Intelligence,*** Vol. 72, (2018), 267-293. DOI: https://doi.org/10.1016/j.engappai.2018.04.009

30. Gholizadeh, H., Fazlollahtabar, H., Fathollahi-Fard, A. M., Dulebenets, M. A. Preventive maintenance for the flexible flowshop scheduling under uncertainty: a waste-to-energy system. ***Environmental Science and Pollution Research***,. (2021), 1-20. DOI: https://doi.org/10.1007/s11356-021-16234-x

31. Li, X., Zhang, X., Yin, M., Wang, J. "A genetic algorithm for the distributed assembly permutation flowshop scheduling problem." In Congress on Evolutionary Computation**,** (2015), 3096-3101. DOI**:** 10.1109/CEC.2015.7257275.

32. Noroozi, A., Mokhtari, H. "Scheduling of printed circuit board (PCB) assembly systems with heterogeneous processors using simulation-based intelligent optimization methods." ***Neural Computing and Applications***, Vol. 26, No. 4, (2015), 857-873, DOI: 10.1007/s00521-014-1765-z.

33. Jia, H. Z., Fuh, J. Y., Nee, A. Y., Zhang, Y. F. "Web-based multi-functional scheduling system for a distributed manufacturing environment." ***Concurrent Engineering***, Vol. 10, No. 1, (2002), 27-39. DOI: https://doi.org/10.1177/1063293X02010001054.

34. Jia, H. Z., Nee, A. Y., Fuh, J. Y., Zhang, Y. F. "A modified genetic algorithm for distributed scheduling problems." ***Journal of Intelligent Manufacturing***, Vol. 14, No. 3, (2003), 351-362, DOI: https://doi.org/10.1023/A:1024653810491.

35. Chang, P. C., Huang, W. H., Wu, J. L., Cheng, T. C. E. "A block mining and re-combination enhanced genetic algorithm for the permutation flowshop scheduling problem." ***International Journal of Production Economics***, Vol. 141, No. 1, (2013), 45-55. DOI: https://doi.org/10.1016/j.ijpe.2012.06.007.

36. Tavakoli, M. R., Mahdizadeh, E. "A new ILP model for identical parallel-machine scheduling with family setup times minimizing the total weighted flow time by a genetic algorithm." ***International Journal of Engineering, Transactions A: Basics,*** (2007), 183-194. https://www.ije.ir/article_71659_16ca2c1872353424a108cbed1308c9e3.pdf

37. Maghzi. P, Mohammadi. M, Pasandideh. S. H. R, Naderi. B." Operating Room Scheduling Optimization Based on a Fuzzy Uncertainty Approach and Metaheuristic Algorithms." ***International Journal of Engineering, Transactions B: Applications,*** Vol. 35, No. 2, (2022), 258-275, DOI: 10.5829/IJE.2022.35.02B.01

38. Abbasi, M., Rafiee, M. "Efficient Parallelization of a Genetic Algorithm Solution on the Traveling Salesman Problem with Multi-core and Many-core Systems." ***International Journal of Engineering, Transactions A: Basics,*** Vol. 33, No. 7, (2020). 1257-1265. DOI: 10.5829/IJE.2020.33.07A.12

39. Eiben, A. E., Smith, J. E. "Introduction to evolutionary computing," ***Berlin: Springer***, Vol. 53, (2003), DOI: https://link.springer.com/book/10.1007%2F978-3-662-44874-8

40. Pillai, A. S., Singh, K., Saravanan, V., Anpalagan, A., Woungang, I., Barolli, L. "A genetic algorithm-based method for optimizing the energy consumption and performance of multiprocessor systems." ***Soft* Computing,** Vol. 22, No. 10, (2018), 3271-3285. DOI: https://doi.org/10.1007/s00500-017-2789-y.

41. Bahalke, U., Yolmeh, A. M., Hajizade, A. N., Bahalke, A., "Genetic and tabu search algorithms for the single machine scheduling problem with sequence-dependent set-up times and deteriorating jobs." ***International Journal of Engineering, Transactions A: Basics,*** Vol. 23, No. 3, (2010), 227-234. https://www.ije.ir/article_71863_722b2d35a63e36bcda4a13526b9d5525.pdf

42. Yazdani, M., Kabirifar, K., Fathollahi-Fard, A. M., Mojtahedi, M. "Production scheduling of off-site prefabricated construction components considering sequence dependent due dates." ***Environmental Science and Pollution Research***, (2021), 1-17. DOI: https://doi.org/10.1007/s11356-021-16285-0

43. Fathollahi-Fard, A. M., Hajiaghaei-Keshteli, M., Tavakkoli-Moghaddam, R. Red deer algorithm (RDA): a new nature-inspired meta-heuristic. ***Soft Computing,*** Vol. 24, No. 19 (2020), 14637-14665. DOI: https://doi.org/10.1007/s00500-020-04812-z

Persian Abstract

چکیده

در این مقاله یک مسئله تولید مونتاژ سه مرحله‌ای در نظر گرفته شده است. با توجه دانش بدست آمده ما از مسئله، فرض ماشین‌های موازی در مرحله سوم، سیستم تولید مونتاژ سه مرحله‌ای با کارخانه‌های موازی و ماشین‌های موازی در مرحله سوم سیستم تولید-مونتاژ با کارخانه‌های موازی فقط در این مقاله بررسی شده است. به منظور کمینه کردن بیشینه زمان تکمیل کارها در تمام کارخانه‌ها (Makespan)، تخصیص کارها به کارخانه‌ها و تعیین توالی آنها بایستی به صورت مناسب انجام گردد. به منظور حل مسئله در ابعاد کوچک، یک مدل برنامه‌ریزی عدد صحیح آمیخته ارائه شده است که توسط حل کننده سیپلکس (CPLEX) اجرا شده است. با توجه به پیچیدگی محاسباتی، حل مسئله در ابعاد بزرگ توسط سیپلکس امکان‌پذیر نیست، بنابراین به منظور حل آن و کنترل پیچیدگی محاسباتی، یک الگوریتم ژنتیک بهبود یافته از ترکیب الگوریتم ژنتیک و روش طولانی‌ترین زمان پردازش (LPT) ارائه شده است که الگوریتم ژنتیک ترکیبی با طولانی‌ترین زمان پردازش (HGALPT) نامیده می‌شود. پارامترهای مسئله با استفاده از تحلیل واریانس یک طرفه (ANOVA) تنظیم شده‌اند. در پایان به منظور ارزیابی کارایی و اثربخشی الگوریتم ارائه‌شده و تعیین تاثیر هر پارامتر روی تابع هدف، تحلیل حساسیت روی پارامترهای مسئله انجام شده است.