ISSN 1330-3651
UDC/UDK 658.515.011:004.421.2

HYBRID GENETIC ALGORITHM FOR ASSEMBLY FLOW-SHOP SCHEDULING PROBLEM WITH SEQUENCE-DEPENDENT SETUP AND TRANSPORTATION TIMES

Vahid Majazi Dalfard, Allahyar Ardakani, Tak Nazalsadat Banihashemi

Original scientific paper

This paper presents a hybrid genetic algorithm for assembly flow-shop scheduling problem with sequence-dependent setup and transportation times. The used objective function in this research consists of minimizing of the sum of total weighted squared tardiness, makespan, total weighted squared earliness and number of tardy job. Since the problem is NP-hard, we solved this problem by hybrid genetic algorithm. To validate the proposed model, the Lingo 8.0 software was used. Comparison between the results of the Lingo 8.0 and hybrid genetic algorithm shows that in larger problems (if n >10, where n is the number of jobs) the results obtained by Lingo do not have adequate efficiency and cannot be compared with the proposed hybrid genetic algorithm in terms of computational time and deviation from the minimum objective function. Test results are provided for a wide range of problem instances.

Keywords: *sequence-dependent setup; Hybrid genetic algorithm, flow-shop scheduling*

Hibridni genetski algoritam za planiranje poslova montaže na tekućoj traci s vremenima za montiranje i transport ovisnima o redoslijedu odvijanja poslova

Izvorni znanstveni članak

Ovaj rad prikazuje hibridni genetski algoritam za planiranje poslova montaže na tekućoj traci s vremenima za montiranje i transport ovisnima o redoslijedu odvijanja poslova. Objektivna funkcija upotrijebljena u ovom istraživanju sastoji se od smanjenja zbroja ukupno procijenjenih zakašnjenja na kvadrat, vremena potrebnog za izradu (makespan), ukupno procijenjenih ranije obavljenih poslova na kvadrat i broja zakašnjelih poslova. Da bi se potvrdio predloženi model, korišten je program Lingo 8.0. Usporedba rezultata dobivenih pomoću Lingo 8.0 i hibridnog genetskog algoritma pokazuje da kod većih problema (ako je n >10, gdje je n broj poslova) Lingo ne daje odgovarajuću efikasnost i ne može se usporediti s predloženim hibridnim genetskim algoritmom u odnosu na vrijeme izračuna i devijaciju od minimalne objektivne funkcije. Rezultati ispitivanja daju se za veliki broj slučajeva.

Ključne riječi: *vrijeme montiranja ovisno o slijedu poslova, hibridni genetski algoritam, planiranje poslova montaže na tekućoj traci*

1 Introduction

Flow shop assembly production system is a system that combines different manufacturing operations independently and simultaneously and then the manufacturing parts are collected and transmitted to assembly line. In this problem, a set of parts is produced by independent production line and finally converted to the product by assembly stage. This type of assembly manufacturing system can be a solution to respond to market pressure to manufacture different products. Global competition and the need to control production costs, lead companies to manufacture products, so that by combining parts and sub-assemblies they will be able to manufacture more various products. This is one of the greatest benefits of these structures. In scheduling perspective, this type of manufacturing systems can be modeled such as two-stage assembly flow shop. Considering the time of collection and transportation of parts from the production stage to the assembling stage, we will face a more realistic model of the two-stage assembly flow-shop problem. It is possible to put this assumption as a discrete phase between the production and assembly stages. Thus the model can be closer to reality and we will have a three-stage model.

Three-stage flow shop assembly problem is an extended mode of the two-stage flow shop assembly problem. In the two-stage flow shop assembly problem, it is assumed that each job is made of m pieces, which are produced on m parallel machines. The required time for collecting and transporting m completed parts from the first stage to the second is very short. This assumption is particularly inappropriate especially when we use the two-stage assembly flow shop problem for the simulation of manufacturing systems with several industrial units (Factory) and a unit assembly. In these systems the main sub-assemblies are collected from numerous factories and are transported to the assembly unit for the final product to be manufactured. Thus, an operation must be done, which includes the collecting and transporting of pieces. The used time for this operation cannot be ignored. The need for intermediate operations is resolved by using carriers in the flexible manufacturing system. The role of the carriers is to collect m finished pieces, which are related to each work in the first stage. Then they transport pieces to the assembly machines. In this paper, the three-stage assembly flow shop problem is formulated with respect to the collection and transportation times, which are mentioned in the second stage. The setup time dependent on the sequence for parallel machines and independent of the first stage is modeled. So the problem consists of three stages. In the first stage m parallel and independent machines are producing pieces and doing their operations. The second is the stage of transporting the produced pieces and the third stage is pieces assembling stage.

1.1 Literature review

Arthanary and Ramaswamy [1] introduced the Hybrid flowshop (HFS) and studied a branch and bound algorithm for two stages. Salvador [2] presented one of the earliest works with m stages, where a dynamic programming algorithm for the no-wait flowshop with multiple processors was proposed. Laguna [3] considered a facility that produces supplies to photocopiers and laser printers. He pointed out that changing production from one toner to another results in large setup times (generally of the order of days).

Luo and Chu [4] proposed a branch-and-bound algorithm of the single machine schedule with sequence-dependent setup times for minimizing maximum tardiness.

They used a modified NEH algorithm to generate the initial solution for the algorithm. Nearchou [5] proposes a hybrid SA-based algorithm which incorporates features from GA based and local search heuristics. The author has shown that the results obtained using this algorithm on Taillard's benchmark problems are comparable to those obtained using Ogbu and Smith [6, 7] Osman and Potts [8].

Solimanpur et al. [9] proposed a tabu search (EXTS) algorithm based on neural networks for permutation flow-shop scheduling problem. The authors used the modified NEH algorithm proposed by Taillard [10] to generate the initial solution. The model considered in this paper is an extension of the model proposed by Koulamas and Kyparisis [11]. They introduced a three-stage assembly flowshop, in which the middle stage is dedicated to components collection and transformation. However in our paper, we add the sequence-dependent setup times (SDST) to their model in order to make the model real.

Koulamas and Kyparisis [11] considered a three-stage assembly flowshop scheduling problem with the objective of minimizing the makespan and analyzed the worst-case ratio bound for several heuristics.

Lee et al. [12] studied a two-stage AFSP with considering two machines at the first stage. Tozkapan et.al [13] considered a two-stage AFSP by minimizing the total weighted flow time. They developed a lower bound, dominance criterion, and incorporated answers into a branch and bound algorithm. They also used a heuristic algorithm to derive an initial upper bound. Allahverdi and Al-Anzi [14] considered distributed database systems and computer manufacturing as an assembly flowshop scheduling problem by minimizing the maximum lateness.

Ravindran et al. [15] used a multi-criterion approach to flow shop scheduling problems by considering makespan time and total flow time as objectives to be minimized.

Rahimi Vahed and Mirghorbani [16] developed multi-objective particle swarm optimization to minimize the weighted mean completion time and weighted mean tardiness simultaneously in flow shop scheduling environment.

Biswal [17] found superiority of hybrid genetic algorithm in which initial solutions have been searched by particle swarm optimization for multi-objective scheduling of flexible manufacturing system. Performance of the algorithm has been tested on three instances only, which has been one of the main limitations of the work.

Al-Anzi and Allahverdi [18] considered a two-stage AFSP with the objective of minimizing a weighted sum of makespan and maximum lateness and proposed three heuristics, TS, PSO, and SDE to solve the NP-hard model. Naderi et al [19] considered SDST hybrid flow shop scheduling to minimize makespan and maximum tardiness. They hybridized the SA (HSA) with a simple local search to promote the quality of final solution.

In this paper, we consider a three-stage AFSP with sequence-dependent setup times at the first stage, where each machine requires a setup time before starting the operation. In this problem, the minimizing of the weighted, sum of total weighted squared tardiness, makespan, total weighted squared earliness and a number of tardy jobs is considered as four criteria. To our knowledge, this paper is the first work that considers the problem with sequence-dependent setup times and the four foregoing criteria.

2 Problem of three-step assembly flow-shop scheduling

2.1 Problem definition

Pinedo [20] demonstrated the information flow in a manufacturing system as shown in Fig. 1. In a manufacturing environment, the scheduling function has to interact with other decision making functions. One popular system that is widely used is the Material Requirements Planning (MRP) system.

Each work j_j, j=1,2,…,n includes a chain of action sets (($O_{l,j}$,…$O_{m,j}$), $O_{T,j}$, $O_{A,j}$)). Operation $O_{l,j}$ should be done on the machine M_i, i=1,2,…,m and requires $p_{i,j}$ time unit**.** The machine M_i per unit time can be only one to do. Collection and transfer operation $O_{T,j}$ on the machine M_T is done and PT_j unit takes time. Assembly operation $O_{A,j}$ on the machine M_A

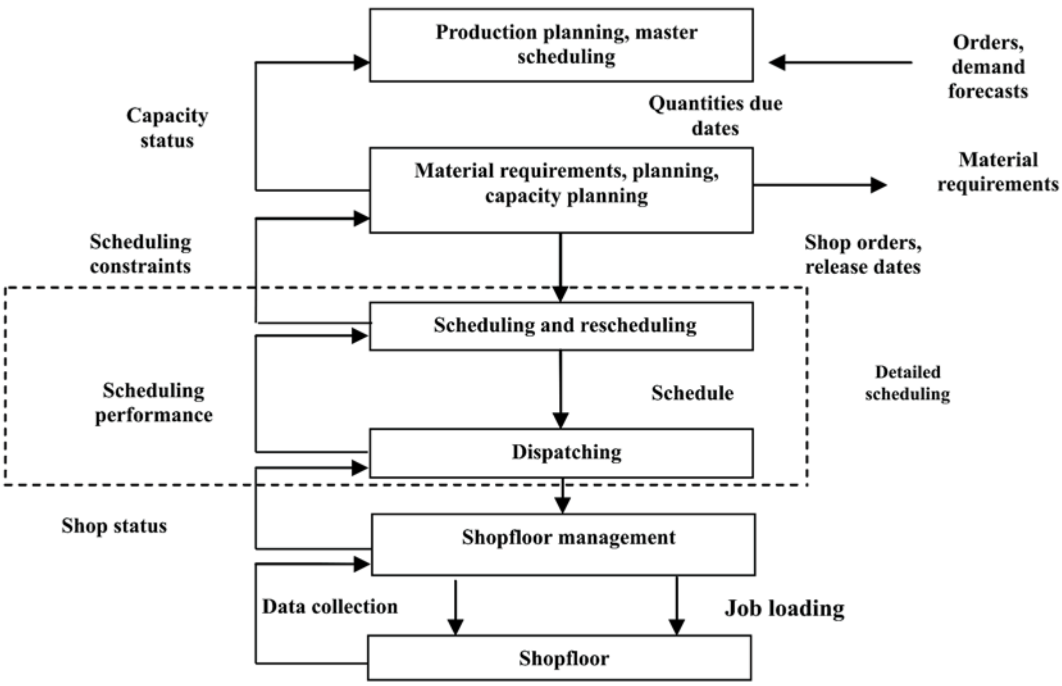

Figure 1 Information flow diagram in a manufacturing system

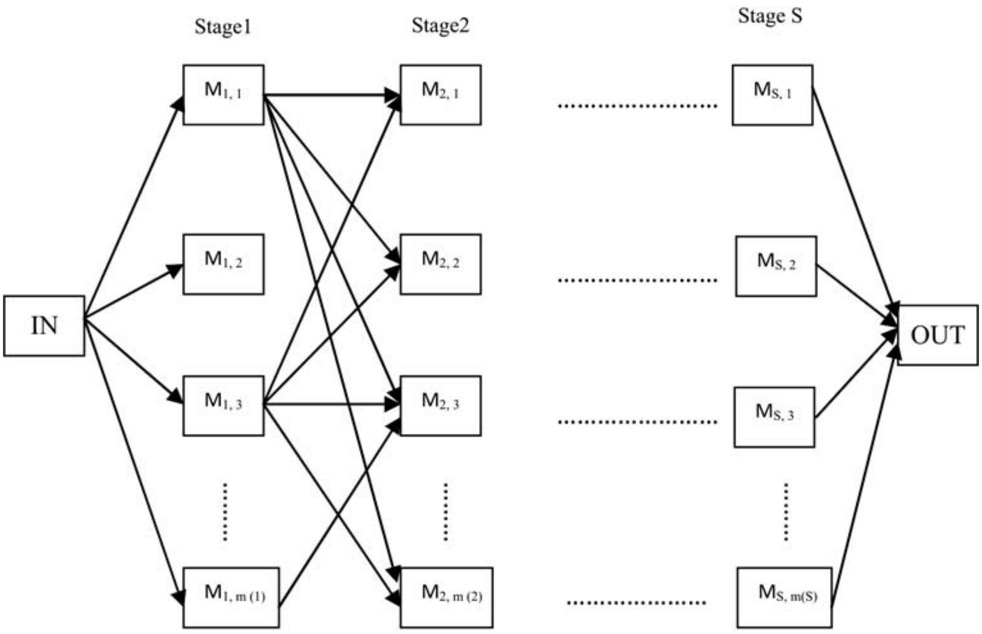

Figure 2 Schematic view of the production process

is done and PA_j units take time. Job for i and k, i=1,2,…,m, k=1,2,…,m, $k \neq i$. Operation $O_{i,j}$ and $O_{k,j}$, j=1,2,...,n can be done simultaneously. Operation $O_{T,j}$ begins only when all operations simultaneously $O_{l,j}$,…,$O_{m,j}$ reach completion and operation $O_{A,j}$ after operation $O_{T,j}$ can begin. Collected by M_T could eventually transfer all components of a task. Similarly, M_T assembly machine can assemble all parts at one time. A schematic representation of a flowshop environment is given in Fig. 2.

2.2 Goals

Problem objectives include: minimizing of the weighted, sum of total weighted squared tardiness, makespan, total weighted squared earliness and a number of tardy jobs.

2.3 Hypotheses and limitations

Hypotheses and limitations of the model considered below are:

- All the jobs and machines are available at time Zero.
- No break in operations is allowed, it means that each operation, which is started, should be completed without interruption.
- There is no priority to do jobs.
- Problem is static, definitive and fixed.
- Each machine can only work on one job in each moment.
- The order of works on all machines (both parallel machines of the first stage and machines of the second and third stages) must be the same. They must also be as permutation scheduling.
- All parallel machines of the first stage have setup time dependent on the sequence. In other words, despite the running time of activities and works on a machine, the setup time must be considered depending on the previous and current job.
- Parallel machines in the production phase (first) are non-uniform (preparation and processing times for different machines.)
- When all productions are completed in the first stage, the pieces will be collected and transported into a second stage. The minimum start time specified in the second stage is equal to the maximum completion time of all its components on parallel machines (first).
- The assembly operations will be started in the third stage only when the transportation in the second stage is ended.
- Pre-emption is not allowed. Once an operation is started on the machine it must be completed before another operation can begin on that machine.
- All processing time on the machine is known, deterministic, finite and independent of the sequence of the jobs to be processed.
- The first machine is assumed to be ready whichever and whatever job is to be processed on it first [21].

2.4 Parameters and decision variables

Parameters:

n – Number of jobs that must be planned
m – Number of simultaneous independent machines at the first stage
j – Index for jobs
i – Index for machines
α – Weight for total weighted squared tardiness
β – Weight for makespan
γ – Weight for total weighted squared earliness
δ – Weight for no tardy jobs
C_j – Completion time of job j
d_i – Due date of job j
E_j – Earliness of job j
T_j – Tardiness of job j
N_t – Number of tardy jobs
W_j – Weight for job j
$S_{[i-l,i,k]}$ – Preparation time on machine k, work situations i

$S_{[0,l,k]}$ – Initial preparation work for the first sequence position on k machine in the first stage.
$T_{[i,k]}$ – The working process in the i position on machine k in the first stage.
$PT_{[i]}$ – Time required for collecting and transferring the work on i position of the first stage of the assembly stage (second stage transfer).
$PA_{[i]}$ – Assembly work time (third phase) in the position i.
$D_{[i]}$ – Job in the position i of lead time delivery.
$L_{[i]}$ – Deviation from the position of the delivery job.

Variables:
$CT_{[i]}$ – Job complication time in i_{th} position in transportation stage (stage 2)
$C_{[i]}$ – Assembly stage (stage 3) finishing time in the position i.
$\Delta_{2[i]}$ – Idle time to start the first job in position i in the second stage.
$\Delta_{3[i]}$ – Idle time to start job in the position i in the third stage.
$T_{[i]}$ – Amount of tardy job in position i.

3 Multi -Objective Fitness Function

Multi-objective fitness function considered here is the minimizing of the weighted, sum of total weighted squared tardiness, makespan, and total weighted squared earliness and number of tardy jobs.

Total weighted squared tardiness is given as:

$$\sum_{j=1}^{n} W_j T_j^2.$$

Where T_j

$$T_j = (C_j - d_j) \text{ if} \qquad (1)$$
$$= 0 \text{ otherwise}$$

Second performance measures for scheduling is makespan ($C_{\max}$) which

$$C_{\max} = \max(C_1, ..., C_n). \qquad (2)$$

3.1 One method estimated completion time of tasks

From where $CT_{[0]} = 0$ with regard to the eq. (3) finishing time can be allocated in the position i phase sequence at the end of the following transportation acquired:

$$CT_{[i]} = \{\max_{k=1,...,m} \{\sum_{l=1}^{i} s_{[l-1,i,k]} + t_{[l,k]}\} CT_{i-1}\} + PT_{[i]}. \qquad (3)$$
$$i = 1,2,3,..., n$$

By considering $C_{[0]} = 0$, and the eq. (3) we can obtain the time of job completion, which is assigned in sequence i_{th} position in assembly stage as follows:

$$C_{[i]} = \max\{CT_{[i]}, C_{[i-1]}\} + PA_{[i]}. \qquad (4)$$
$$i = 1,2,3,..., n$$

The third criterion is total weighted squared earliness which has been given as:

$$\sum_{j=1}^{n} W_j E_j^2.$$

Where $T_j = (d_j - C_j)$ if $d_j - C_j \geq 0$ (5)
$= 0$ otherwise.

The fourth and last criterion for scheduling is to minimize the number of tardy jobs. Associated with each job j is a due date $d_j \succ 0$. Let $U_j = 1$ if due date for job j is smaller than the completion time C_j of job j, otherwise $U_j = 0$. The total number of tardy jobs N_t is defined as:

$$N_t = \sum_{j=1}^{n} U_j. \qquad (6)$$

Therefore multi-objective fitness function is obtained by combining all the above four objectives into a single scalar function so as to minimize the weighted sum of total weighted squared tardiness, makespan, total weighted squared earliness and number of tardy jobs which has been framed as:

$$\min\ [\alpha \sum_{j=1}^{n} W_j T_j^2 + \beta C_{\max} + \gamma \sum_{j=1}^{n} W_j E_j^2 + \delta (N_t)]. \qquad (7)$$

Where α, β, γ and δ, are the weight values for the considered objective functions having constraints:

$$\alpha \geq 0,\ \beta \geq 0,\ \gamma \geq 0,\ \delta \geq 0$$
and
$$\alpha + \beta + \gamma + \delta = 1.$$

4 Proposed mathematical model and its formulation

In this section a mathematical model of integer non-linear programming (INLP) for the mentioned problem is presented. First, parameters and mathematical models of decision variables are introduced:
i – Sequence position index
j – Work number index
k – Index numbers of machines in the first stage
$X_{[i]j}$ – Will be 1 if i_{th} job is assigned to j_{th} position. Otherwise it will be 0.
$S_{q,j,k}$ – Set up time dependent sequences of job q to job j on machine k to the first stage
$S_{0,j,k}$ – Initial preparation time on machine k in the first stage, if the job j is placed in the first sequence.
$y_{[i]}$ – Maximum time to complete all operations m jobs in the position i in the sequence of the first stage.
$A_{[i],j,k}$ – Total operation time on machine k, the first position to job in positions i, if the job j is located in the position i.
$Z_{[i]}$ – Since transportation to work in positions i start.
$W_{[i]}$ – Since the assembly operation to work in positions i.
$t_{j,k}$ – Process time job j on machine k in the first stage.
$PT_{j,k}$ – Time needed to collect and transfer the job to the second stage j.
PA_j – Assembly time, i job in the third stage.

$$\min\ [\alpha \sum_{j=1}^{n} W_j T_j^2 + \beta C_{\max} + \gamma \sum_{j=1}^{n} W_j E_j^2 + \delta (N_t)] \qquad (9)$$

s. t.

$$\sum_{i=1}^{n} x_{[i]j} = 1 \quad \forall j \tag{10}$$

$$\sum_{j=1}^{n} x_{[i]j} = 1 \quad \forall j \tag{11}$$

$$A_{[1],j,k} = x_{l.j} \times (t_{j,k} + s_{0,j,k}) \quad \forall j,k \tag{12}$$

$$A_{[i],j,k} = x_{i.j} \times (t_{i,k} + \sum_{q=1,q\neq i}^{n} (x_{[i-1]q} \times S_{q,j,k})) + \sum_{l=1,l\neq j}^{n} A_{[i-1],l,k}$$

$$i = 1,2,3,...,n \tag{13}$$

$$A_{[i],j,k} \leq y_{[i]} \quad \forall i,j,k \tag{14}$$

$$y_{[i]} \leq Z_{[i]} \quad \forall i \tag{15}$$

$$CT_{[i]} = Z_{[i]} + \sum_{j=1}^{n} (x_{[i]j} \times PT_j) \quad \forall i \tag{16}$$

$$CT_{[i-l]} \leq Z_{[i]} \quad \forall i \tag{17}$$

$$CT_{[i]} \leq w_{[i]} \quad \forall i \tag{18}$$

$$C_{[i]} = w_{[i]} + \sum_{j=1}^{n} (x_{[i]j} \times PA_j) \quad \forall i \tag{19}$$

$$C_{[i-1]} \prec w_{[i]} \quad \forall i \tag{20}$$

$$T_{[i]} = C_{[i]} - \sum_{j=1}^{n} (x_{[i]j} \times d_{[i]}) \quad \forall i \tag{21}$$

$$T_{[i]} \geq 0 \quad \forall i \tag{22}$$

$$T_{[i]} \leq T_{\max} \tag{23}$$

$$x_{ij} \in \{0,1\} \quad \forall i,j \tag{24}$$

$$S_{i,j,k} = 0 \tag{25}$$

$$C_{[0]} = CT_{[0]} = 0 \tag{26}$$

Eq. (9) represents the objective function of the scalar problem combining the four objective functions that consist of minimizing of the weighted, sum of total weighted squared tardiness, makespan, total weighted squared earliness and number of tardy job.

Constraints (10) and (11), respectively, show that each job can be located in only one position of the sequence and can only be allocated to a task. Eqs (12) and (13) respectively show the way of calculating introduced variables $A_{[i],j,k}$ and $A_{[i],j,k}$ according to $A_{[i],j,k}$.

Constraint (14) shows the completion time of m operation in the first stage for the job in sequence i_{th} position is before the start of the second stage (of transportation). Constraint (15) is about the requirement of the completion of all m operations in sequence i_{th} position in the first stage before starting the second stage (transportation). Constraint (16) shows how to calculate job completion time in sequence i_{th} position of transportation stage. Constraint (17) is about requirement of carrier freedom for transportation of sequence $(i–l)_{th}$ position before starting the transportation in sequence i_{th} position. Constraint (18) shows the requirement of finishing the transportation for the job in sequence i_{th} position before starting its assembly operation in the last stage. Constraint (19) shows the way of calculating job completion time in sequence i_{th} position in assembly stage. Constraint (20) is about the requirement of finishing the assembly work in sequence $(i–l)_{th}$ before starting assembly operation for job in i_{th} position. Constraints (21) and (22) show how to calculate the tardiness, related to the assigned job in sequence i_{th} position. Constraint (23) calculates the maximum tardiness of jobs. Eq. (24) represents the allocation of zero values and the decision variable is $x_{i,j}$.

4.1
Solving the OR model in lingo software

In the lingo software, according to the fast growth of the space dimensions of the answer, it takes a long time to reach a definitive answer and if n >10, it is not possible to reach the answer. Thus, this issue is NP-hard [11]. In Tab. 1 the amount of objective functions obtained by lingo solution with their process time is mentioned. The Lingo solutions have a large deviance of optimal solutions (above 30 %) in large problems (n >10, for example, Instance vmd12) or lingo cannot obtain optimal solutions in a reasonable time (20 hours).

5
Outline of Hybrid Genetic Algorithm (HGA)

The Hybrid Genetic Algorithm (HGA) acts as a global search technique which is similar to simple genetic algorithm with generation deviation of initial solution. In HGA, initial feasible solution is generated with the help of some heuristics and then this initial sequence is used along with the population according to the population size for executing the procedure of simple genetic algorithm. The proposed HGA is described as:

Step 1: Initialization and evaluation

a) The algorithm begins with the generation of initial sequence with special heuristics (SH) called as one of the chromosome of population

b) Generation of (Ps-1) sequences randomly as per population size (Ps).

c) Combining of initial sequence obtained by special heuristics with randomly generated sequence to form the number of sequences equal to population size (Ps).

Step 2: Reproduction

The algorithm then creates a set of new populations. At each generation, the algorithm uses the individuals in the current generation to generate the next population. To generate the new population, the algorithm performs the following steps:

a) Scores each member of the current population by computing fitness.

b) Selects parents based on the fitness function.

c) Some of the individuals in the current population that have best fitness are chosen as elite and these elite individuals are utilized in the next population.

d) Production of offspring from the parents by crossover from the pair of parents or by making random changes to a single parent (mutation).

e) Replaces the current population with the children to form the next generation.

Step 3: Stopping limit

Stopping condition is used to terminate the algorithm for certain numbers of generations [22].

Table 1 The amount of objective functions obtained by lingo solution with their process time are mentioned

Prob name	n	m	–α=0,4, β=0,3, γ=0,2, δ=0,1			α=0,3, β=0,2, γ=0,1, δ=0,4			α=0,1, β=0,4, γ=0,3, δ=0,2			α=0,2, β=0,1, γ=0,4, δ=0,3			α=0,25, β=0,25, γ=0,25, δ=0,25		
			Best solution	Optimal solution	Computational time	Best solution	Optimal solution	Computational time	Best solution	Optimal solution	Computational time	Best solution	Optimal solution	Computational time	Best solution	Optimal solution	Computational time
vmd1	6	2	171	171	182	169	169	169	148	148	124	182.5	182.5	111	176	176	153
vmd2	6	4	219	219	822	212	212	798	198	198	741	234	234	738	226	226	802
Vmd3	6	6	415	415	2921	401	401	2856	391	391	2814	455	455	2802	406	406	2919
Vmd4	6	8	966	966	4855	958	958	4848	952	952	4835	974	974	4832	961	951	4842
Vmd5	8	2	521	521	2002	518	518	1998	511	511	1986	539	539	2001	519	519	2000
Vmd6	8	4	1339	1339	4245	1324	1324	4218	1317	1317	4211	1428	1428	4215	1327	1327	4214
Vmd7	8	6	1886	1886	7562	1871	1871	7533	1869	1869	7529	1915	1915	7521	1873	1873	7533
Vmd8	8	8	2282	2282	15825	2279	2279	15818	2275	2275	15802	2324	2324	15559	2281	2281	15749
Vmd9	10	2	1711	1711	8429	1709	1709	8421	1702	1702	8411	1731	1731	8415	1710	1710	8421
Vmd10	10	4	3015	3015	21151	2998	2998	21141	2295	2295	21139	3021	3021	21135	3002	3002	21139
Vmd11	10	6	5188	5188	68741	5181	5181	68709	5175	5175	68691	5216	5216	68690	5184	5184	68659
Vmd12	10	8	**9846**	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Vmd13	25	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Vmd14	25	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Vmd15	25	6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Vmd16	25	8	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Vmd17	30	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Vmd18	30	4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Vmd19	30	6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Vmd20	30	8	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 2 The amounts of objective functions obtained by HGA with their process time

Prob name	n	m	α=0,4, β=0,3, γ=0,2, δ=0,1			α=0,3, β=0,2, γ=0,1, δ=0,4			α=0,1, β=0,4, γ=0,3, δ=0,2			α=0,2, β=0,1, γ=0,4, δ=0,3			α=0,25, β=0,25, γ=0,25, δ=0,25		
			Best solution	Optimal solution	Computational time	Best solution	Optimal solution	Computational time	Best solution	Optimal solution	Computational time	Best solution	Optimal solution	Computational time	Best solution	Optimal solution	Computational time
vmd1	6	2	173	171	1,2	169	169	1,2	148	148	1,1	182,5	182,5	1,2	178	176	1,02
vmd2	6	4	219	219	1,2	212	212	1,1	199	198	1,2	234	234	1,2	226	226	1,3
Vmd3	6	6	415	415	1,2	401	401	1,3	391	391	1,3	455	455	1,3	406	406	1,3
Vmd4	6	8	967	966	1,5	958	958	1,5	952	952	1,5	974	974	1,4	961	951	1,5
Vmd5	8	2	521	521	1,3	518	518	1,3	511	511	1,3	539	539	1,2	519	519	1,3
Vmd6	8	4	1339	1339	2,1	1324	1324	2,3	1317	1317	2,1	1428	1428	2,1	1324	1327	2,1
Vmd7	8	6	1886	1886	2,4	1871	1871	2,2	1871	1869	2,2	1916	1915	2,2	1873	1873	2,2
Vmd8	8	8	2284	2282	3,01	2279	2279	3.01	2275	2275	3.1	2324	2324	3.01	2281	2281	3.01
Vmd9	10	2	1711	1711	2.9	1711	1709	2.9	1702	1702	2.9	1733	1731	2.9	1711	1710	2.9
Vmd10	10	4	3015	3015	3.6	2998	2998	3.5	2295	2295	3.7	3021	3021	3.6	3002	3002	3.6
Vmd11	10	6	5188	5188	3.9	5182	5181	3.9	5175	5175	3.9	5217	5216	3.9	5184	5184	3.9
Vmd12	10	8	**7213**	7213	5.5	7210		5.8	7202		5.7	7221		5.7	7112		5.7
Vmd13	25	2	19212		5.3	19189		5.2	19128		5,7	19218		5,3	19192		5,7
Vmd14	25	4	41519		12	41509		12	41488		12	41729		12	41512		12
Vmd15	25	6	52941		13,5	52920		14	52896		14,5	53019		14	52926		14,5
Vmd16	25	8	87219		22	87141		22,6	87111		22	87349		24	87152		22
Vmd17	30	2	45689		25	45671		25	45662		25	45786		23	45682		25
Vmd18	30	4	91863		29	91852		25	91842		29	92004		27	91889		29
Vmd19	30	6	129254		32	129006		32	128849		31	129859		29	129021		31
Vmd20	30	8	197218		37	197108		36	197039		39	497849		36	197304		37

5.1 Considering the function of Hybrid Genetic Algorithm (HGA)

In this section, 20 experimental issues of flow-shop scheduling problem in different dimensions were done by HGA. MATLAB 7.5 is used for programming the HGA and a personal computer with 3.2 PIV, 2 GB RAM is used for running the algorithm.

6 Parameter setting

In this section, the results of the computational experiments are used to evaluate the performance of the proposed algorithm for assembly flow-shop scheduling problem with sequence-dependent setup and transportation times. There are nine instances for each problem size. At this point, some information about parameter analysis would be useful. Initially, several experiments were conducted on test problems in order to determine the tendency for the values of parameters. Six test problems were used for this purpose [23].

In each step, only one of the parameters was tested. Each test was repeated four times. We considered the following values for the several parameters required by the proposed HGA:

Crossover probability (*pc*): four levels (0,90; 0,85; 0,80 and 0,75).

Mutation probability (*pm*): four levels (0,02; 0,04; 0,06 and 0,08).

Number of initial population (*np*): three levels (300, 200 and 100).

Number of generation (*ng*): one level (200).

Test results showed that these values were suitable for the problem. Later, additional tests were conducted in order to determine the best values. After completing the tests, the Taguchi analysis is applied for the different values of parameters. The best values of the computational experiments for assembly flow-shop scheduling problem with sequence-dependent setup and transportation times were obtained for pc = 0,85, pm = 0,06, np = 200 and ng = 200. These values were set as the default value of the parameters.

7 Conclusions

We presented an efficient hybrid genetic algorithm HGA that solves assembly flow-shop scheduling problem with sequence-dependent setup and the transportation times that minimize the total weighted squared tardiness, makespan, total weighted squared earliness and number of tardy job. First, the problem was formulated by using a nonlinear model that is strongly NP-hard [11], especially for large-sized problems. We proposed a HGA to solve the presented problems, and its performances and results were compared with the lingo solutions. Lingo answers have a large deviance from optimal solutions (above 30 %) in large problems (n>10, for example, instance vmd12) or they cannot obtain optimal solutions in a reasonable time (20 hours). But the answers obtained by the proposed HGA got optimal solution with a very little deviance in a short time. So the suggested algorithm is a more efficient one.

8 References

[1] Arthanary, L.; Ramaswamy, K. An extension of two machine sequencing problem. // Journal of the Operational Research Society of India. 8, 4(1971), pp.10–22.

[2] Salvador, M. A solution to a special case of flow shop scheduling problems. In S. E. Elmaghraby (Ed.), // Symposium of the theory of scheduling and applications. Springer-Verlag, New York, USA. 1973.

[3] Laguna, M. A heuristic for production scheduling and inventory control in the presence of sequence-dependent setup times. // IIE Transactions. 31, (1999), pp. 125–134.

[4] Luo, X.; Chu, C. A branch-and-bound algorithm of the single machine schedule with sequence-dependent setup times for minimizing maximum tardiness. // European Journal of Operational Research. 180, (2007), pp. 68–81.

[5] Nearchou, A.C. A novel metaheuristic approach for the flow shop scheduling problem. // Engineering Applications of Artificial Intelligence. 17, (2004), pp. 289–300.

[6] Ogbu, F. A.; Smith, D. K. The application of the simulated annealing algorithm to the solution of the n/ m / Cmax flow shop problem. // Computers and Operations Research. 17, 3(1990), pp. 243–253.

[7] Ogbu, F.A.; Smith, D.K. Simulated annealing for the permutation flowshop scheduling. // OMEGA. 19, 1(1992), pp. 64–67.

[8] Osman, I. H.; Laporte, G. Meta-heuristics: A bibliography. // Annals of Operations Research. 63, (1996), pp. 513–623.

[9] Solimanpur, M.; Vrat, P; Shankar, R. A neuro-tabu search heuristic for the flow shop scheduling problem. // Computers and Operations Research. 31, (2004), pp. 2151–2164.

[10] Taillard, E. Benchmarks for basic scheduling problems. // European Journal of Operational Research. 64, (1993), pp. 278–285.

[11] Koulamas C.; Kyparisis G. The three-stage assembly flowshop scheduling problem. // Computer and Operation Research. 28, (2001), pp. 689–704.

[12] Lee, C-Y.; Cheng, T. C. E.; Lin, B. M. T. Minimizing the makespan in the 3-machine assembly type flowshop scheduling problem. // Management Sciences. 39, (1993), pp. 616–625.

[13] Tozkapan, A.; Kirca, O.; Chung, C. S. A branch and bound algorithm to minimize the total weighted flow time for the two stage assembly scheduling problem. // Computer and Operation Research. 30, (2003), pp. 309–320.

[14] Allahverdi, A.; Al-Anzi, F. S. A PSO and a tabu search heuristics for the assembly scheduling problem of the two-stage distributed database application. // Computer and Operation Research. 33, 4(2006), pp. 1056–1080.

[15] Ravindran, D.; Noorul-Haq, A.; Selvakumar, S. J.; Sivaraman, R. Flow shop scheduling with multiple objective of minimizing makespan and total flow time. // International Journal of Advanced Manufacturing Technology. 25, (2005), pp.1007–1012.

[16] Rahimi-Vahed, R.; Mirghorbani, S. M. A multi-objective particle swarm for a flow shop scheduling problem. // Journal of Combinatorial Optimization. 13, 1(2007), pp.79–102.

[17] Biswal, B. B.; Jena, N. A Hybrid Evolutionary Algorithm for Flexible Job-shop Scheduling Problem. // IE (I) Journal–PR. 89, (2008), pp. 14-18.

[18] Al-Anzi, F. S.; Allahverdi, A. Heuristics for a two-stage assembly flowshop with bicriteria of maximum lateness and makespan. // Computer and Operation Research. 36, (2009) , pp. 2682–2689.

[19] Naderi, B.; Zandieh, M.; Balagh, K. K. G.; Roshanaei, V. Scheduling hybrid flowshops with sequence dependent setup times to minimize makespan and maximum tardiness. // International Journal of Advanced Manufacturing Technology. 41, (2009), pp.1186–1198.

[20] Pinedo, M. Scheduling theory, algorithms, and system, // (Prentice-Hall). 2005.

[21] Ebrahimnejad, S.; Tavakkoli-Moghaddam, R.; Hatami, S.; Maboudian, Y. An Assembly Flow-Shop Scheduling Problem with Sequence- Dependent Setup, // International Journal of Industrial Engineering. 21, 1(2010), pp. 15-23.

[22] Ashwani, D.; Pankaj, C. Hybrid Genetic Algorithm for Multi criteria Scheduling with Sequence Dependent Set up Time. // International journal of Engineering. 3, (2010), pp. 510-520

[23] Majazi, V. Adjustment of the primitive parameters of the simulated annealing heuristic. // Indian Journal of Science and Technology. 4, 6 (2011), pp. 627-631.

Authors' addresses:

Vahid Majazi Dalfard
Young Researchers Club
Kerman Branch
Islamic Azad University
Kerman, Iran
Email:vmd_60@yahoo.co.nz
Tel: 00989125700730

Allahyar Ardakani
Young Researchers Club
Qazvin Branch
Islamic Azad University
Qazvin, Iran
Email: ar.ardakani1989@gmail.com

Tak Nazalsadat Banihashemi
Young Researchers Club
Qazvin Branch
Islamic Azad University
Qazvin, Iran
Email:Taknaz.bh@gmail.com