Science Arena Publications
Specialty Journal of Engineering and Applied Science
ISSN: 2520-5943
Available online at www.sciarena.com
2018, Vol, 3 (4): 1-13

Designing A New Model and Solving the Workshop Scheduling Problem with Batch Processors and Incompatible Jobs

Saeed Akbarzadeh*, Mohammad Saidi-Mehrabad

Department of Industrial Engineering, Faculty of Industrial and Mechanical Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran.

***Corresponding Author**

Abstract: *This research studies the problem of the scheduling machines with the incompatible jobs, taking into account transition-related sequencing, limitation of access, and early and late penalties. The work is assumed to be available at zero time and also classified into several groups. In the issue examined, interruption is not allowed. In this research, minimizing the total amount of early and late penalties is considered to be related to the time. A mathematical model was proposed for the question first and then a genetic algorithm was used to simultaneously optimize early and late penalties and finally, to test the performance of the proposed algorithm, a number of sample questions are produced and solved in a variety of sizes. The results of the meta-heuristic algorithm solution are compared to the results obtained from solving the mathematical model, which compares the performance of the proposed algorithm.*

Keywords: *Workshop Scheduling, Parallel Batching Processing Machines, Incompatible Jobs, Genetic Algorithm*

INTRODUCTION

Planning the production is, in fact, the timing and determining the order of the priorities of doing things optimally. Clearly, for a production unit, cost minimization and productivity increase are important, so scheduling is needed in order to minimize costs and increase productivity. Considering the fact that the workshop system has a very high flexibility, Production planning is used in production systems based on customer orders and assembly systems with high flexibility in production, the use of this type of programming has increased in this system. Having an appropriate production scheduler has a major impact on improving efficiency and access to the organization's goals. The production scheduling model varies from one organization to another according to the goals and priorities of access. Therefore, in order to determine the appropriate timetable model in the organization, the priorities, priorities and resources of the resource must first be investigated. Most large manufacturing industries, such as automotive and assembly industries, carry out part processing and completion work linearly and step by step so that the workflow scheduling issues cover a wide range of production and assembly models. On the other hand, this issue attempts to minimize the use of limited resources by minimizing the maximum completion time.

Considering the development of batch processing in recent studies and the high utilization of the industry, the studied workshop has a batch processing feature in which there are different families of jobs. Machines can handle multiple jobs simultaneously with respect to capacity constraints. In this case, the goal is to determine the order of processing the various categories of jobs so that the predetermined target functions are optimized. In the case of a blockbuster flow problem, as in the case of the workshop's workflow, there must be a chain of operations on different machines, with the difference that in this environment, machines can simultaneously perform multiple jobs. It has already been proved by the researchers that the objective function of minimizing the time it takes to complete all jobs in a workshop environment with two machines can be solved by innovative methods, but this problem turns into hard-NP for more than two machines. In this research, the work is considered to be incompatible. Considering that batch processor machines are included in the scheduling of jobs, it is necessary to provide a new model for solving the workflow scheduling problem and solve it using meta-heuristic methods.

The production of cells by placing parallel machines as one of the main subsets of the family is very important in scheduling, and the timing of this model has always been based on various performance criteria. Considering that the public solving methods are based on more complex models, such as a sequential flexible machine model, based on simpler model solutions, including cellulose production. In some working environments, the scheduling of cellular production may mean that all machines cant process all the work, in which case each work can be done by a set of machines, which is called the limit of a collection of processes. One of the characteristics of cellular production is the technological dependence on human resources. In many human environments, it is considered as the main element. In the activities that human beings have in common, the learning problem is very important, and so far in most articles, it is commonly believed that the processing time of jobs is constant and independent of the sequence. While in many practical cases, repeating similar or (different) jobs, the operator's abilities and skills will increase, and as a result, the processing time will be reduced. This leads to a continuous improvement in the performance of production facilities, especially human resources, which are referred to as learning effects. One of these activities can be anything to do with the manual system, such as setting up machines, cleaning machines, and preparing time.

The present study aims to reduce the gap between theoretical developments and industrial applications in the field of science of scheduling. In this regard, a new model for the cellular production problem is presented with sequence-specific preparation time limitations, processing set limits and learning effects, and the criterion for optimizing early and late times. In this research, we try to divide the jobs into different categories by modeling the problem first, then obtain the optimum order of processing of these categories in the production line, in order to minimize the maximum completion time. Also, for an actual question, a meta-heuristic solution method will be presented. Due to the fact that the scheduling question is the case study of the workshop, and this type of manufacturing system in the workshops is very common, the results of this research can be obtained in production and service systems.

Given the lack of statistical analyzes, there are no hypotheses in this study and the assumptions are:

- There are K different families of jobs
- The processing time of the jobs is different and varies on the machines.
- The processing time of each group is equal to the processing time that is the maximum processing time.
- The transportation time of different categories is negligible and is considered zero.
- The capacity of each job is less than the capacity of the machine.
- When the batch processing starts, the jobs can't be deleted or added from the batch.
- All machines are batch processors.

From related work, Kim et al. (2012) Pointed out that the question of the scheduling of unrelated parallel machines, with the assumption of sequencing installation time independent of the type of machine, with the aim of minimizing the total latency, was investigated, and the algorithm Refrigeration simulation to solve their question. Wang et al. (2014) examined the timing of unrelated parallel machines, with the assumption of sequencing installation time, with the goal of minimizing the average weight of completion time of jobs. After comparing the results of the seven methods, they chose the best method in terms of computational method. The process of this method is such that the smallest processing time plus the installation time in relation to the weight of the work in the target function is allocated to the machines. And solves the question. Zhu and hady (2010) presented the question of the timing of unrelated parallel machines with a time-limited preparation time-dependent sequence of jobs and a total function of total latency and early maturity. They proposed an integer programming model for this question and showed that the model optimized for a question of nine works and three machines at a reasonable computational time.

Research Methodology

Mathematical model

1. **Indices and collections**

I: Collection of all categories of incompatible parts
i & i': Indicator of incompatible parts
J: Collection of all stages of production
j & *j'*: Indicator of production steps
K : Collection of all devices(machinery)
k & *k'*: The index of devices
*J*i : The set of production steps that the handle of the incompatible piece i traverses with them
*A*i : The first step in the production of an incompatible piece of i
*z*i: The last step in the production of an incompatible piece of i
parameters
*PT*i*jk*: The time required to process the incompatible handle of (i) in the (j) stage by the (k) device(machine)
*ST*i*jk*: Setup time of device(machine) (k) at stage (j) for incompatible piece (i) processing
Variables
*C*i*jk*: The end time of processing the incompatible piece (i) in the (j) stage by the (k) device(machine)
*Y*i*jk*: The variable is zero and one if: The handle of the incompatible piece (i) is processed in step (j) on (k) device (machine), otherwise it is zero
*X*ii'jk: The variable is zero and one if: An incompatible piece (i) after (i') in step (j) is on (k) device (machine), otherwise it is zero

2. **Limitations**

Limit (1) states that if there is a sequence of operations on a machine in one step, then only one of the components can take service earlier than the next segment and the production process of both pieces does not occur simultaneously.

$$\sum_{\substack{i\epsilon I \\ i\neq i1}} \sum_{i'\epsilon I} X_{ii'jk} \leq 1 \qquad \forall\, j\epsilon J, \forall\, k \,\epsilon\, K \qquad (1)$$

Limit (2) states that the process of producing each piece at each stage is performed on a device (It is worth noting according to its production process chart).

$$\sum_{k\epsilon K_{ij}} Y_{ijk} = 1 \qquad \forall\, i\epsilon I, \quad j \,\epsilon\, J_i \qquad (2)$$

Limit (3) states that there must be a connection between the production and the binary variables of assigning work to the communication machine.

$$C_{ijk} \leq MY_{ijk} \qquad \forall\, i\epsilon I, \quad j\,\epsilon\, J_i\,, \quad k\epsilon K_{ij} \qquad (3)$$

Limit (4) states that the completion of all steps (j) is completed when:

1. The handle of the previous piece has been completely processed by that device(machine). (Step j')
2. The device(machine) should be ready for processing (setup time limit).
3. The time required to process the handle must be expired.

$$C_{ijk} \geq \sum_{k'} C_{ij'k'} + ST_{ijk} + PT_{ijk} - M\left(1 - Y_{ijk}\right), \qquad \forall i\epsilon I,\ \ j, j'\epsilon J_i,\ \ j \neq A_i,\ \ j' = j-1,\ \ k\epsilon K_{ij} \qquad (4)$$

Limitation (5) shows that if any piece is processed on a machine, other machines it will be Y_{ijk}=0 and C_{ijk}=0 . setup of devices can be done prior to the arrival of the batches to the different stages of the production line (j). To start the process (j), it must begin at the start of the production line and the time of preparing it should finish. This connection implies the preparation of a (y) model in the first stage of the (x) string start of a piece.

$$C_{ijk} \geq Y_{ijk} * \left(ST_{ijk} + PT_{ijk}\right), \qquad \forall\ i\epsilon I,\ \ \forall\ j\epsilon A_i,\ \ \forall\ k\epsilon K_{ij} \qquad (5)$$

3. **Operational sequence limits**

In x, the order of processing the batch of parts should be done with care. These constraints indicate the sequence of processing the batch of components and the connection between the model variables.

$$\forall\ i, i'\epsilon\ I, \qquad i < i',\ \ j\epsilon J_i \cap J_{i'}, \qquad k\epsilon K_{ij} \cap K_{i'j}$$
$$X_{iik} + X_{ji\prime ik} \leq 1 \qquad (6)$$

$$\forall\ i, i'\epsilon\ I, \qquad i \neq i',\ \ j\epsilon J_i \cap J_{i'}, \qquad k\epsilon K_{ij} \cap K_{i'j}$$
$$2X_{iik} \leq Y_{jik} + Y_{j\prime i\prime k} \qquad (7)$$

$$\forall\ i, i'\epsilon\ I, \qquad i < i',\ \ j\epsilon J_i \cap J_{i'}, \qquad k\epsilon K_{ij} \cap K_{i'j}$$
$$Y_{ijk} + Y_{i\prime jk} \leq X_{ik} + X_{j'iik} + 1 \qquad (8)$$

Explanation: In our x factories we do not have a limit immediately, as no wait. And the processing time of a piece is independent of the other piece and is not related to each other. According to the limitation (6), or the piece (i), before the (i') is processed on the k device in (j) step, or vice versa and or none of them.

But in limitation (7): the piece (i) is processed before (i') That means, if X_{iik} wants to take a value, then Y_{ijk} and $Y_{i'jk}$ get a value of one. This means both of (i) and (i') are processed on the same device. And this state is corrected with equation (8) and and with the help of relation 6, when two pieces (i) and (i') are processed in the (j) stage, either the piece (i) before (i') is placed on the corresponding machine or, on the contrary.

4. **Non-interference limit**

The processing of batches in the (x) factories should not interfere and and the piece (i') ends when it runs on (k) machine, which completes the processing of the piece (i) on the (k) machine and the devices necessary settings for the (i') handle are also performed and the time required to process the batch of (i') has also been spent. Of course, these connections are for devices that can process more than one piece.

$$C_{ijk} \geq C_{i'jk} + ST_{ijk} + PT_{ijk} - M * X_{i'ijk} - 2M + MY_{ijk} + MY_{i'jk}$$
$$\forall\ k\epsilon\ K: N_k > 1,\ \ j = J_k, \quad i, i' = I_k:\ \ i < i' \qquad (9)$$

$$C_{i\prime jk} \geq C_{ijk} + ST_{ijk} + PT_{ijk} - M * X_{ii\prime jk} - 2M + MY_{ijk} + MY_{i'jk}$$
$$\forall\ k\epsilon\ K: N_k > 1,\ \ j = J_k, \quad i, i' = I_k:\ \ i < i' \qquad (10)$$

5. The objective function

The objective function is to minimize the total cost of delaying component production and the cost of moving parts. It is obvious that the batches have different production and completion times due to the production process and the total time x of the components is equal to the production time of a piece that is processed and completed later than the other parts.(It has a higher completion time) and we want to minimize the total production time of these bundles, which are represented by $\underset{i}{\mathrm{Max}} \sum_{k\epsilon K_{iz_i}} C_{iz_ik}$

$$MIN\ C = \left(MAX\ \textstyle\sum_i C_{ijk}\right) \tag{11}$$

$$\underset{i}{\mathrm{Max}} \sum_{k\epsilon K_{iz_i}} C_{iz_ik} = T$$

and instead of it, in the target function, it becomes a form of T, which is a limitation in itself.

$$t_i \geq MAX\ \textstyle\sum_{k\epsilon K_{iz_i}} C_{ikz_i} \quad \forall i\epsilon I \tag{12}$$

$$C_{ijk} \geq 0 \tag{13}$$

$$T \geq 0 \tag{14}$$

$$X_{iik}: Binary \tag{15}$$

$$Y_{iik}: Binary \tag{16}$$

According to the mathematical model presented, as well as incompatible work, the application of this model is used in various manufacturing companies such as tiling, automotive component parts for vehicles that are in the workshop environment and with project-based incompatible jobs .

In the case of incompatible work, this issue is further increased in project-based companies, which must be manufactured in several production workshops by different units, This case is also very popular in these industries, for example, on a milling machine in one stage of production, the need for machining and roughing should immediately be made to prepare the hole for the bodywork cylinder. The operational nature of the work on the device is subject to fundamental changes and even changes to the instrument.

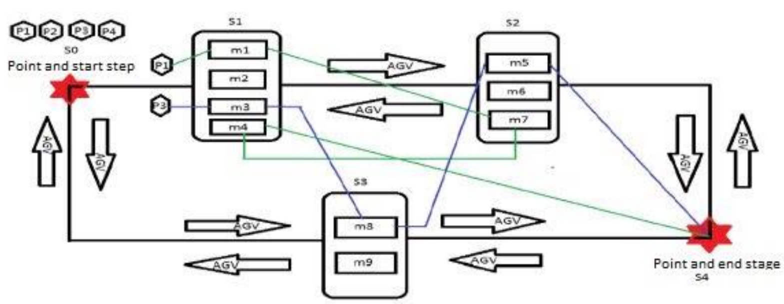

Figure 1: the symbolic image of the mathematical model drawn to examine the question(Source: Based on Computational Math Model)

Data Analysis

- **Genetic Algorithm**

Genetic algorithms are one of the most robust methods derived from the field, which searches for randomly driven question space, which is a search in the form of trying to create better answers in each generation than the previous generation's responses.

The basic feature of the genetic algorithm is its simplicity. The classical solution method is depicted in Figure 1 and this is how we first define the answer in the form of a chromosomal structure. By introducing the fitness function, we describe the quality of the responses presented in each chromosome as a number. Then, we produce a certain number of chromosomes randomly or quasi-randomly, which are known as primitive populations of chromosomes.

At this point, we have a number of answers to the questions that are mostly of low quality. Determine the quality of each chromosome from the population based on the fitness function. Now, using the appropriate method (a method in which the chances of selecting a chromosome with a better fit are greater than the chromosome with less fitting), we select two chromosomes for reproduction. Then, with the release of these two chromosomes, we create a new chromosome.

With a certain probability, we change the number of genes of some of the chromosomes. Selection, intersection, and mutation processes create a new population (new generation) of chromosomes. In the case of the convergence of chromosomes, the optimal response to the production of the generation stops. Otherwise, the generation of each generation from the previous generation to the satisfactory solution or the completion of the condition of the end of the algorithm continues. We use the genetic algorithm to solve the problem.

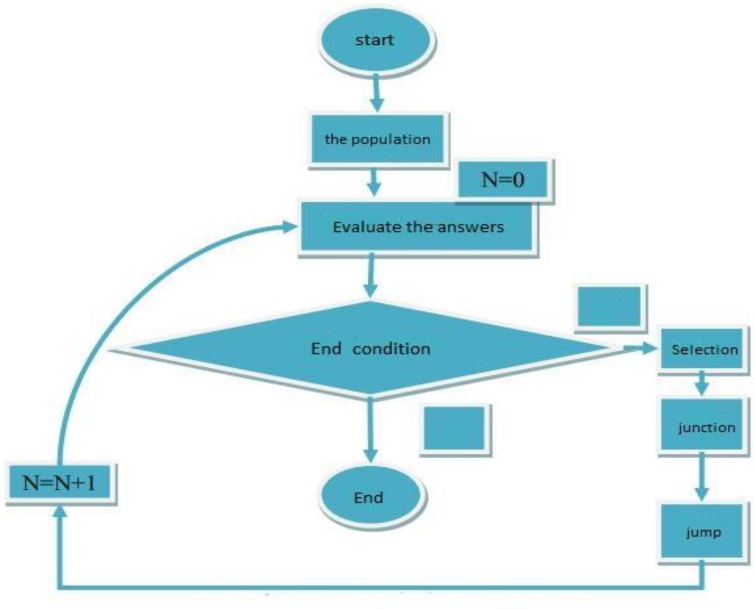

Figure 2: Overview of the classic genetic algorithm

- **Introducing Taguchi Method**

The Taguchi experiments were designed in 1960 by Professor Taguchi. This method can determine the optimal conditions with the least number of experiments and will significantly reduce the time and cost of performing the required tests. In the Taguchi method, different orthogonal arrays are used as matrices of experimentation based on the number of selected parameters and related levels. In this method, changes are introduced by a factor called (signal to noise ratio) and the test conditions with the highest signal to noise ratio are selected as optimal conditions. In the following, for better explanation of the design of the experiments, the procedures of the Taguchi method are summarized in Figure 3:

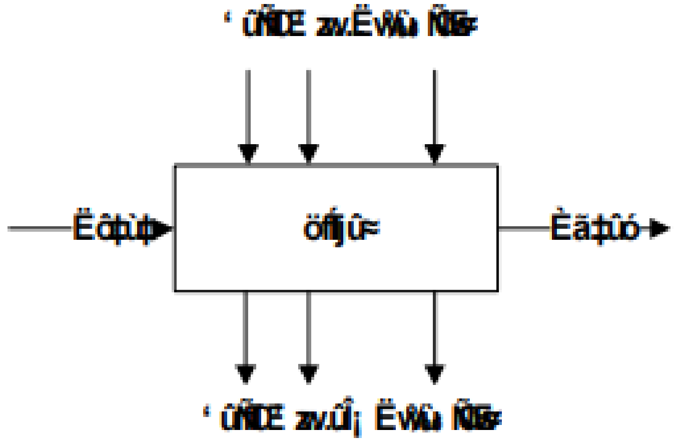

Figure 3: Test design process

1. Selection of control and non-controlling factors along with related levels.

Control factors are in fact the input parameters of the questions that change in order to achieve optimal conditions during the experiments in terms of selective levels and matrix of experiments. Non-controlling factors also refer to all factors that cause changes but they are assumed to be constants during constant tests.

2. Calculation of the loss function, to determine the variation between test results and desired values.

This function is calculated according to the terms of the question and with the help of relations 17-19

First mode: The smaller value represents the optimal state:

$$SB = \frac{1}{n}\sum_{i} y_i^2 \qquad (17)$$

Second mode: The larger value represents the optimal state:

$$LB = \frac{1}{n}\sum_{i} \frac{1}{v_i^2} \qquad (18)$$

Third mode: Nominal size is desirable:

$$NB = \frac{1}{n}\sum_{i} (y_i - y_0)^2 \qquad (19)$$

Where (n) is the number of iterations, *Yi* is the measured output and *Y0* is Desirable nominal size.

3. Calculation of signal to noise values for each output is calculated according to equation 20:

$$SN = -10 \log_{i}(L_i) \qquad (20)$$

4. Calculate the amount of signal to noise for each level of the parameters: In the Taguchi method, the surfaces of the parameters that have the highest signal-to-noise values (regardless of the type of loss function) are introduced as optimal levels.
5. Determining the importance of each of the parameters using the statistical analysis tool of variance.
6. Data analysis and optimal prediction of output.

7. Performing a confirmation test to verify the results.

To determine the optimal parameters for the proposed model, we use the Taguchi method in the following.

- **Generate sample question form**

The real world questions are usually large, and accurate resolution software is usually not able to solve such questions in large dimensions. For this reason, the meta-innovative algorithm is used to solve such questions in large dimensions. In the following, we solve the problem solved in the previous section with the fragmentary algorithm and also the GAMS software, and compare the results with each other.

- **Taguchi test on parameters of the genetic algorithm**

We analyze the parameters of the genetic algorithm in three levels, Regarding the existence of six variables and three levels, we use the Minitab software to apply Taguchi tests to determine the optimal parameters of the parameters. This method proposes 27 tests in the form of Table 1. The result is calculated by the genetic algorithm based on the specified level for each parameter, and putting in Table 1:

Table 1: Taguchi's proposed experiments for the colonial competition algorithm

↓	C1	C2	C3	C4	C5	C6	C7
	Max decade	N_imp	N_countries	Beta	P_revolution	Zeta	Objective Function
1	1	1	1	1	1	1	10821
2	1	1	1	1	2	2	11098
3	1	1	1	1	3	3	10623
4	1	2	2	2	1	1	10188
5	1	2	2	2	2	2	9140
6	1	2	2	2	3	3	9219
7	1	3	3	3	1	1	9179
8	1	3	3	3	2	2	9496
9	1	3	3	3	3	3	9397
10	2	1	2	3	1	2	9872
11	2	1	2	3	2	3	10524
12	2	1	2	3	3	1	10069
13	2	2	3	1	1	2	9238
14	2	2	3	1	2	3	9278
15	2	2	3	1	3	1	9575
16	2	3	1	2	1	2	9179
17	2	3	1	2	2	3	9278
18	2	3	1	2	3	1	9159
19	3	1	3	2	1	3	10603
20	3	1	3	2	2	1	10030
21	3	1	3	2	3	2	10702
22	3	2	1	3	1	3	9278
23	3	2	1	3	2	1	9792
24	3	2	1	3	3	2	9337
25	3	3	2	1	1	3	9258
26	3	3	2	1	2	1	9377
27	3	3	2	1	3	2	9238

Also, the result of the analysis of Taguchi experiments is shown in Figures 2 and 5:

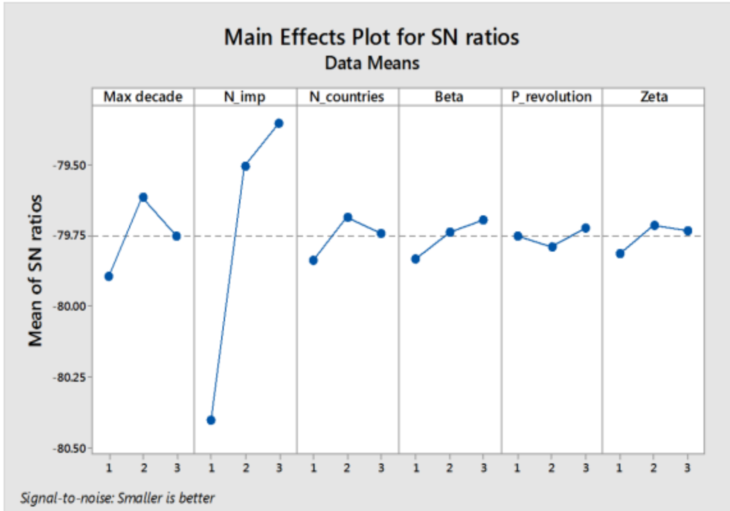

Figure 4: Signal to Noise Rate Graph as a result of the analysis of the results of Taguchi experiments

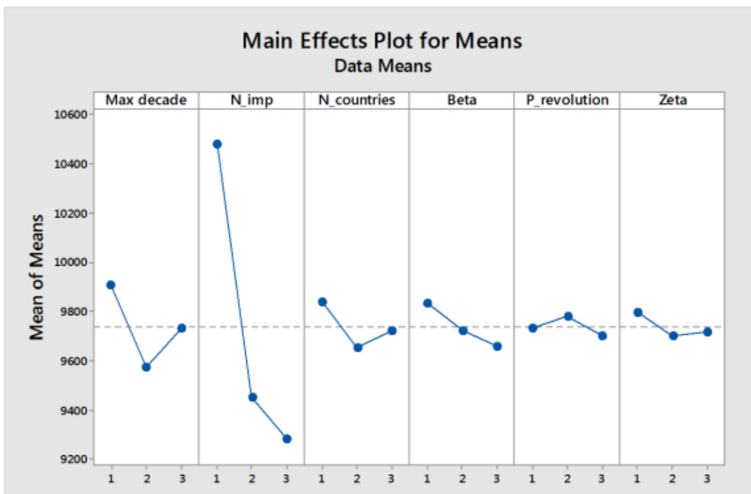

Figure 5: Average graph as a result of analyzing the results of Taguchi experiments

As we know, the signal-to-noise ratio is higher and, if the average value is lower, we will have a better result. For formulas 4 and 5, for the first, third, and sixth parameters of the level two, and for the other parameters, the level three creates the optimal state.

- **Implementing Genetic Algorithms**

After generating the initial society, it is time to run successive generations of solutions using the genetic algorithm. This process is presented in the form of a pseudo program:

1. Determine the initial parameters: Number of generations (itermax), Intersection rate (pc), Mutation rate (pm), Population size (ps).
2. Estimation of the random society created by the fit function.
3. t=1
4. Repeat the steps below to resolve the (t >iter_max) statement
   4.1 Intersection operation to produce (pc*popsize) offspring.
      4.1.1. Selection operation by roulette cycle selection method.
      4.1.2. Intersection operator for offspring production.
   4.2 Mutation operations to produce (pm*popsize) offspring.

4.2.1. Selection operation, by random selection method.
4.2.2. Moving mutation operation is based on the chosen parent and offspring production.
4.3 Population ranking and selection of popsize*(1- pm-pc) first work as elite.
4.4 Replacing the offspring produced as a new population.
4.5 Evaluation of newly created society.
5. t=t+1

In this pseudo-program, after generating the parameters of the genetic algorithm controller, the random-created community chromosomes are evaluated by fit function and in each replication, the algorithm is performed using roulette cycle selection, parent selection, and intersection operations. And by using a random selection method, a chromosome is selected and the mutation is performed on it. In addition to the two cases, a number of chromosomes that are more suitable for the purpose of the function than other chromosomes are introduced as part of a new generation population. Then the new society will be valued. The algorithm ends with generating a certain number of generations and the last produced society is introduced as the ultimate community of algorithms.

- **Intersection operator**

In order to produce offspring from selected parents, an intersection operator is used as follows. Figure 6 shows an example of how to use the intersection operator for a question with eight jobs and two machines for producing an offspring. Regarding the type of chromosome and to prevent the occurrence of unusual chromosomes, for each machine, a random number is selected based on the number of jobs on it. Then the first part is transferred to the offspring and the same jobs are removed from the other parent and the rest of the parent's work is transferred to the offspring.

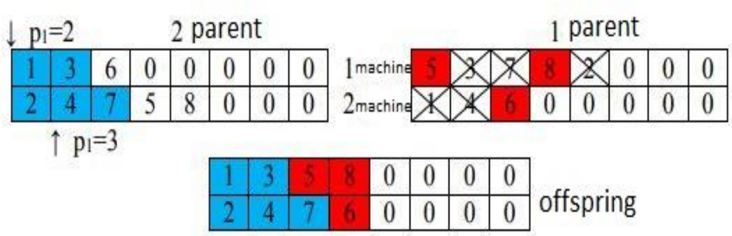

Figure 6: Intersection Operation

- **Mutation operator**

The mutation operator is used in two ways. In the first method, for each chromosome, a machine was selected randomly and then, given the number of jobs on which the machine is located, two jobs are randomly selected and their locations are replaced. What is shown in Figure 7.

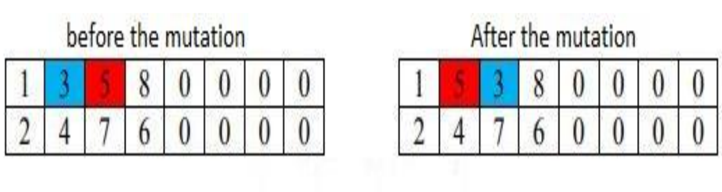

Figure 7: First type of mutation

In addition to the above, two machines are randomly selected and then a job from each machine is selected randomly and their locations are replaced by a limited set of processing. This method is shown in Figure 8.

(1) Before the mutation

1	3	5	0	0	0	0	0
2	4	7	0	0	0	0	0
8	6	0	0	0	0	0	0

(1) After the mutation

6	3	5	0	0	0	0	0
2	4	7	0	0	0	0	0
8	1	0	0	0	0	0	0

Figure 8: the second type of the first mutation

(2) Before the mutation

1	3	5	0	0	0	0	0
2	4	7	0	0	0	0	0
8	6	0	0	0	0	0	0

(2) After the mutation

3	5	0	0	0	0	0	0
2	4	7	0	0	0	0	0
8	6	1	0	0	0	0	0

Figure 9: second type of second mutation

A special type of this case occurs when a job is moved to a vacant place. In this case, the chromosome must be repaired as shown in Figure 9.

- **A summary of the data entered in the genetic algorithm**

In order to study the mathematical model in the exact algorithm in GAMS software and the approximate solution by genetic algorithm, the information in Table 2 and Figure 10 was used.

Table 2: Dimensions

Large	Average	Small	3	Question size
40 & 30	16 & 14	8 & 6	2	Number of taks
6 & 5	5 & 4	4 & 2	2	Number of machines
U ~ [1, 25]			1	Processing time
U ~ [0.2*mean(p), 0.4*mean(p)]			1	Preparation time
b_i=U~[1, m]			1	Total processing

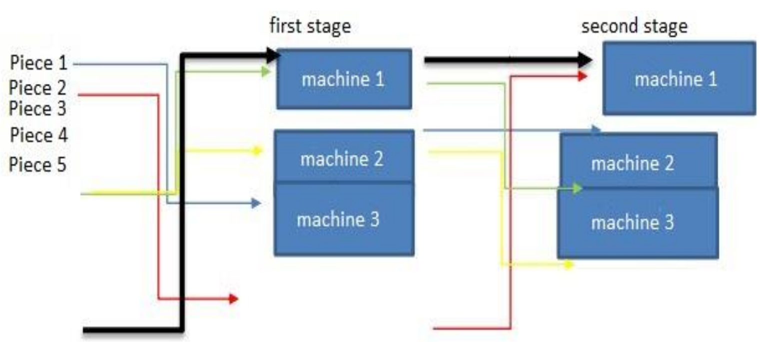

Figure 10: Schematic modeling in GAMS

- **Comparison of Genetic Algorithm and GAMS Results**

A processor running a computer with a (5300 GB RAM, 34GHZ.2) profile is under Windows 8 operating system. MATLAB software has been used to design the genetic meta-innovative method. Each question has been executed 10 times randomly. In the questions, we present the computational results of the larger dimensional questions. Since GAMS can't solve a larger dimensional model, we use the proposed algorithm to solve the model. The purpose of this test is to determine the performance of the proposed algorithm under different conditions.

Table 3: Symbols used in the model

Time required	$t\ (S)$
The value of the objective function of the GOM	f_{opt}
The best value of the objective of the genetic algorithm	f_{best}
The average value of the objective function of the genetic algoritm	f_{avr}

Table 4: Comparison of GAMS and Genetic Algorithms

Algoritm GA			GAMS		
$t\ (S)$	f_{avr}	f_{best}	$t\ (S)$	f_{opt}	
2.5	7314	7212	1	6100	1
2.6	9723	9652	2	8240	2
3.2	8423	8382	5	7560	3
3.8	39086	38097	6	32300	4
4.3	43140	42945	3	39500	5
5.6	67236	66235	15	64640	6
5.4	104645	103937	27	102288	7
6.9	212342	188937	436	192239	8
9.3	362652	335383	-	-	9
14.3	988082	854963	-	-	10
16.4	1203547	1123973	-	-	11
24.1	2359474	2140080	-	-	12
46	1511535	1423249	-	-	13
57.4	3127420	3023791	-	-	14
68	2147112	2007377	-	-	15
71.7	4271063	4128562	-	-	16
194	6855	6525810	-	-	17
186		8965668	-	-	18
417		18287658	-	-	19
647		22170352	-	-	20

The model is solved for larger dimensions using the proposed algorithm. Given the values of Table 4, the algorithm has arrived at the optimal solution within a reasonable time. The proposed algorithm in solving very small dimensional questions requires more computational time than the optimization GAMS software. While solving the questions, with increasing dimensions of the question, the computational time of the proposed algorithms is much lower than the GAMS. Therefore, in these examples, it was found that the algorithms can achieve a satisfactory answer at a much shorter time interval than the GAMS on large-scale questions.

Conclusion

In this research, a workspace environment was developed with parallel batch processors and incompatible jobs in the sequence-dependent processing time. To create the proper platform for the development of the mathematical model, it was assumed that jobs are available at zero time and also are grouped into several groups. All jobs are processed using several parallel machines in the workshop, Also the preparation times are transient sequencing and the car is available at a certain time. Work in a group must be fully prepared before processing, which is called external preparation activity. This activity is carried out by the same machine operator when another group on the machine is automatically processed. By doing so, the role of the external preparation operator is completely eliminated. Transient sequencing is referred to as the preparation of the machine itself.

The main focus of modeling was on the integration of external and internal preparation times and this means that when the machine is processing on a group, the machine operator is preparing a group or subsequent groups. Hence, according to the mathematical model presented, in order to validate the model, we codify and model this model in a small space and after fixing the faults, the model was eventually verified. Finally, with respect to model solving in GAMS software and the background of the research, we found that the proposed model was Hard-NP and it is necessary to calculate the upper limit of the model using approximate methods.
Hence, according to the research background presented, The efficiency of the genetic algorithm has been proven in terms of the timing of production machines and the approximate solution of the model was developed and Due to the design space for the model, it was designed in 20 spaces that, in exact resolution, with the GAM software in the eight design space of the model, was optimized for the time period specified and in the rest of the spaces due to the large space designed, the exact solution did not get to the answer and then, according to the coding of the genetic algorithm, the proposed model with a small error of accuracy (from 9 to 12 percent) is reached at the lower limit at the upper limit of the model in a suitable time hence, in small workshops with a low volume of work, the solution is accurate and in a larger space, it is appropriate to solve the genetic algorithm.

Suggestions

Considering the design of the questionnaire and the proposed model, the following suggestions for the development of this research are presented:

- Development of the mathematical model in the multi-objective model of the model, taking into account the income from production and the reduction of production costs
- The study of the efficiency of ultra-innovative algorithms for the proposed model with the consideration of standard deviations from the exact solution solutions (sampling algorithms: ant colony algorithm, prohibited search algorithm, colonial competition)
- The development of a math model from workshop production to cell production and the goal of minimizing intercellular transport
- Creating uncertainty in the processing time of devices in the mathematical model presented
- Fuzzy mathematical model for increasing the accuracy of the model.

Reference

1. Wang , Reha Uzsoy" A genetic algorithm to minimize maximum lateness on a batch processing machine" Computers &Operations Research 29 (2002) 1621}1640
2. Y.H. Kim C.S. Sung, Y.I. Choung, J.M. Hong," Minimizing makespan on a single burn-in oven with job families and dynamic job arrivals" Computers & Operations Research 29 (2012) 995}1007
3. Zhu, Z. & Heady, R.B. (2010). Minimizing the sum of earliness-tardiness in multi-machine scheduling: a