Proceedings of the 2015 International Conference on Industrial Engineering and Operations Management
Dubai, United Arab Emirates (UAE), March 3 – 5, 2015

A mixed optimization model for a multi-product advanced planning and scheduling system by considering learning effect

Ali Ghahramany Baranghar
Department of Engineering
Islamic Azad University, Tehran South Branch
Tehran, Iran
st_a_ghahremany@azad.ac.ir, baranghar@gmail.com

Ahmad Allahyari
Department of Engineering
Islamic Azad University, Tehran South Branch
Tehran, Iran
st_a_allahyari@azad.ac.ir

Mehrdad Javadi
Department of Engineering
Islamic Azad University, Tehran South Branch
Tehran, Iran
mjavadi@azad.ac.ir

***Abstract*— An advanced planning and scheduling systems is defined as any computer program that uses advanced mathematical algorithms to perform optimization on finite capacity scheduling, sourcing, resource planning and others. The expectations of APS systems have been high, both from academia and industry in the subject area of manufacturing planning and control. A common assumption in APS problems is that the processing time of a given product is constant and independent of its position in the production sequence. However, the real processing time of each job on a machine depends on the position of that job in the sequence and its operator's skill that could be boosted during working time, which is known as learning effect. In order to get closer to the actual conditions of the APS problems, in this paper, a mixed optimization model for a multi-product APS is proposed. The main novelty of the paper is proposing a more efficient mathematical model for the problem of integrating planning and scheduling with learning effect. As this model classified as a NP-Hard problem, a meta-heuristic method, multi-stage genetic algorithm solution, is presented. Finally, the computational results are provided for evaluating the performance and effectiveness of the proposed solution methods.**

Keywords*— *Optimization; Advanced Planning and Scheduling; Learning Effect; Multi-Stage Genetic Algorithm

1 INTRODUCTION

In today's competitive market, organization's main concern is optimization of their operations and boosting their efficiency; therefore, they have to consider different factors in order to optimize their scheduling and be just-in-time towards the customers' demands. Such a situation has lead companies from separate process planning systems to integrated ones. By applying integrated process planning systems, they would be more flexible to answer market's rapid and changing demands by providing better quality as well as lower costs. [1-3]

In this way, companies developed integrated production systems to overcome their competitors by providing excellent service level. Advanced planning and scheduling systems (APS) include the series of scheduling techniques from Job Shop floor to upper integrated production systems and is in new concept to integrate planning models. [4] APS mostly can be applied in one or more cases if, 1. The production is done to answer demands and not to warehouse, 2. Products include many parts or operations, 3. The limitations of production are money and other resources, 4. Products compete together to use resources (multi-product environment), 5. The conditions are not constant in a day that resources cannot be scheduled and 6. Using the flexible production approach is unavoidable. In most researches about APS and workshop production, this assumption has been considered that activities' time is constant and independent from their sequences, while definitely and naturally, by repeating similar activities and even different ones, operators' capability and skills are increased and as a result, the activities' process become decreased. In the topic's, this phenomenon is called learning effect. Therefore, in this research, the learning effect is considered in an APS model for the first time. This model is not only used in flexible Job Shop scheduling models, but is applicable in parallel and assembly lines and changeable process time. In this paper, to solve this model which is a kind of NP-Hard problem, a multi-stage genetic algorithm has been proposed. In this paper, in the second section the literature review of APS and learning effect has been discussed. The third section includes an applicable mathematical model and in the fourth section, the genetic algorithm is developed to solve it. The fifth section includes calculated results and finally, the sixth section provides conclusion and recommendations for future researches.

2 THE LITERATURE REVIEW

Generally, researches about APS problems can be categorized into qualitative and quantitative. Qualitative researches talk about analytical, critical and descriptive aspects and interested people to such problems, see the references [5, 6]. In quantitative researches, usually the creativity has been necessary to provide mathematical models or improve the present solution methods. In the scheduling problems, many researchers have been attracted by outsourcing. These problems usually are applicable when the producer wants deliver the demands in a precise deadline. Lee et al proposed a general model in outsourcing mode and considered the variety of processing paths in APS problems for the first time; they used an effective heuristic method on the genetic algorithm. [7] Chen and Ji provided a MIP model with capacity, operation sequence, process time and delivery deadline constrains in order to integrate operation scheduling with production planning problem in a multi-product environment. This model's objective function is minimizing the costs by considering system idle time (nonvoting time) as well as delay cost factor. [8] One of the most important points in papers that have used genetic algorithm is the proper method results' coding and encoding. [8] Yang and Tang [9] provided an adaptive genetic algorithm to solve the APS multi-objective problem and used numerical samples of Zhang and Gen [10] paper to illustrate the efficiency and speed of this algorithm. [9] Usually, machines can produce different kind of parts, but this point has not been considered in modeling of APS problems. Ornak et al developed mother of Chen and Ji [8] by using disjunctive graph which considers machines' capabilities to produce different kind of parts and, parts could be produced on different machines. [11] Chen et al did a case study for APS problem in a factory in China by using Chen and Ji model and solve this model by a heuristic intelligence method on basis of genetic algorithm and random keys. [12] in all previous resource about APS problems, divert process time is constant and independent from activity sequence which doesn't change during the time, but in current years, researchers have cast doubt on these assumptions by conducting researches on topics like learning effect, over wearying effect as well as variable and controllable process times.

Therefore, in this paper, model of reference [11] was developed innovative variable process time due to factors like learning effect was considered and made the model more applicable in production units. In order to understand the effects of learning scheduling models, a brief history about it is provided below:

A common assumption in classic scheduling problems is that the production process time is constant and independent from sequence of activities [11] these methods are inefficient and evolved restricted multi-product conditions. They developed to new models such as scheduling by considering controllable process time, scheduling the processes that depend on passing the time and scheduling the multipurpose machines, since 1990 decade. [14] Many researchers by evaluating theoretical and practical scheduled problems have concluded that process time can be reduced [15] and learning concept is one of the approaches to decrease the process time. [16] By repeating activity, an operator's capabilities and this gives are increased; so, the process time is reduced and performance of production facilities, specially human resources, continuously would be improve which this phenomenon is known as learning effect.

Wright conducted a research on factors affecting the cost of airplanes included learning effect. [16] He provided the famous 80% assumption that by using scientific experiments in airplane industry, the process time of each production unit is reduced to 20%, if the production output becomes twice. As the role of human resource in production scheduling is important and the number of operations which can be potentially affected by learning is considerable and rational, Biskup introduced learning effect on scheduling problems [15, 17] and then, many researchers conducted their research on this topic.

There are two main approaches about learning effects on scheduling problems: 1. Position-based learning approach in which learning is affected by the number of work which has been produced till now. Usually, this approach is applied in processes which are mainly machine-based and not influenced by human. 2. The sum of processing time approach in which learning is affected by the sum of activity time which has been processed till now. In this approach those processes can be considered that are so complicated and error causing. Researchers the device that if human has impressive impact on these processes, it's better that learning effect is considered dependent on the time. [19]

In recent years, learning effect, due to its undeniable role in new managerial concepts, industry and commercials sectors, has absorbed so much attention [20] which excessive released papers to demonstrate it.

Dejong proposed in new applicable learning model to estimate their reduced processing time of work and in the next reputations by starting tiny study researches' results. [14, 15] Okolowski and Gawiejnowicz considered Dejong's model in the scheduling problems for the first time and provided the model to minimize activity domain for parallel machines scheduling. [15] Ji Wang and Jian Wang considered both position-based and the sum of processing time approaches and provided a mixed learning model. [18] Zhang et al, focused on the logarithmic some of processing time and position-based processing approaches and proposed an equation for logarithmic some of processing time. [19] Amini et al, provided a multi-objective model for identical and parallel machine scheduling with setup and removal times with deteriorating and Learning effects. [21] Hamta et al, proposed a multi-objective optimization model for assembly line balancing with bounded processing Times, Learning Effect, and Sequence-dependent Setup Time [22] and then provided a hybrid PSO algorithm For the same problem. [23]

3 THE MATHEMATICAL MODEL AND ITS COMPLEXITY

Usually, models provided in APS literature are so similar or deliver a similar concept. The model presented by Ornek et al, is one of the newest and the most complete models which is applicable in flexible job shop environments, but assumes that process time is constant independent from activity sequence. [11]

Therefore, in this research, the model provided in [11] was developed in a way that the learning effect is considered in the APS model and the process time is not constant. It should be mentioned that by considering learning models' prerequisites, in this paper, the structure of model as well as its decision variables in the basic paper has been changed.

Consider a multi-product production environment which produces goods by using a series of machines. Each product has a tree structure and is composed of several assembly and sub-assembly parts, and different machines are available to produce these parts. Machines can be available in any time except then zero, but all demands are accessible in base time, zero. The model's target is to provide a scheduling solution for all the demands fully integrated in a way that total work accomplished time becomes minimized. In this program, it is precisely define that which part of each product at what time and in which machine should be produced. Here the model's structure is provided:

3.1 Indexes and sets:

i, j	indexes of products (customers' demands) = 1, 2, …, n
n	total products (customers' demands)
$A(P_i)$	sets of different parts required to produce the ith product
$N(A(P_i))$	number of different parts required to produce the ith product
b	total required parts required to fulfill all customers' demands obtained from equation 2:

$$b = \sum_{i=1}^{n} N(A(P_i)) \qquad (2)$$

q, p	index of parts =1, 2, …, b
N_{ip}	number of p parts required per the ith product
l, k	index of machines = 1, 2, …, m
m	total number of machines
r	index of work position of its maximum value is the total parts number =1, 2, …, b

3.2 Parameters:

wh	daily work hours
P_i	final parts of the ith product, the uppermost part in product tree
Q_i	number of the ith product's demands
t_{ipk}	the normal time of producing the pth part in i^{th} product's on the kth machine done it all human interaction
β	learning factor in the same activities
$\bar{t}_{ipk}$	Main normal time of the pth activity for the ith demand obtained from equation 3 using normal process time, demand value as well as learning rate for the same activities. In other words, $\bar{t}_{ipk}$ is t_{ipk} in which the learning effect has been considered

$$\bar{t}_{ipk} = t_{ipk} \times \frac{\sum_{a=1}^{N_{ip}Q_i} (M_{ipk} + (1 - M_{ipk}) a^{\beta_{ipk}})}{N_{ip}Q_i} \quad \forall i, p, k \qquad (3)$$

a	the total learning factor total learning factor which is logarithm of learning's rate to base 2
$t_{ipk[r]}$	the actual work time required to produce the pth parts of the ith product in the rth position on the kth machine obtained from equation 4

$$t_{ipk[r]} = \bar{t}_{ipk}\left(M_{ipk} + \left(1 - M_{ipk}\right) r^{a}\right) \qquad \forall i, p, k, r \qquad (4)$$

RT_k	the time in which the kth machine is accessible
d_i	delivery time of the ith product
I	system's idle cost per hour
M	positive and enough big number
R_i	set of (p, q) pair products for the it^{th} product's demand so that product q is the direct prerequisite of product p
F_p	machine set which can produce the product p

3.3 decision variables:

C_{max}	accomplishment time of all work
$Cr_{[kr]}$	accomplishment time of the kth machine's r^{th} position
$Z_{ipk[r]}$	binary variable which is 1 if product p for the ith production's demand is scheduled on the kth machine in the rth position
$fin_{ipk[r]}$	positive variable used to linearize the model

There are numerous target functions about scheduling problems and one of the most famous and useful of them is minimizing the work domain, accomplishment time of all work. Minimizing the operations means to minimize the machine usage and the work in process. The declaration 5 is the target function.

$$Min\ Z = C_{max} \qquad (5)$$

3.4 Constrains:

$$Cr_{k1} \geq \sum_{i=1}^{n} \sum_{p \in A(p_i)} t_{ipk[1]} \times Z_{ipk[1]} \times Q_i \times N_{ip} + RT_k$$
$$\forall i, p \in A(p_i) \qquad (6)$$

$$Cr_{k[r]} \geq Cr_{k[r-1]} + \sum_{i=1}^{n} \sum_{p \in A(p_i)} t_{ipk[r]} \times Z_{ipk[r]} \times Q_i \times N_{ip}$$
$$\forall i, p \in A(p_i), r = 2, \dots, b-1 \qquad (7)$$

$$Cr_{k[r]} \geq C_{max} \quad k = 1, \dots, m \ , \ r = b \qquad (8)$$

$$fin_{ipk[r]} = Z_{ipk[r]} \times Cr_{k[r]} \qquad \forall i, k, p, r \qquad (9)$$

$$fin_{ipk[r]} \geq -M\left(1 - Z_{ipk[r]}\right) + Cr_{k[r]}$$
$$\forall i, p \in A(p_i), k \in F_p, r = 1, \dots, b-1 \qquad (10)$$

$$fin_{ipk[r]} \leq M\left(1 - Z_{ipk[r]}\right) + Cr_{k[r]}$$
$$\forall i, p \in A(p_i), k \in F_p, r = 1, \dots, b-1 \qquad (11)$$

$$fin_{ipk[r]} \leq MZ_{ipk[r]}$$
$$\forall i, p \in A(p_i), k \in F_p, r = 1, \dots, b-1 \qquad (12)$$

$$\sum_{k=1}^{m}\sum_{r=1}^{b}(fin_{ipk[r]} - t_{ipk[r]} \times Z_{ipk[r]} \times Q_i \times N_{ip})$$
$$\geq \sum_{k=1}^{m}\sum_{r=1}^{b} fin_{ipk[r]} \qquad \forall i, (q,p) \in R_i \qquad (13)$$

$$\sum_{k=1}^{m}\sum_{r=1}^{b} Z_{ipk[r]} = 1 \qquad \forall i, p \in A(P_i) \qquad (14)$$

$$\sum_{k=1}^{m}\sum_{r=1}^{b} Z_{ipk[r]} \leq 1 \qquad \forall k, r \qquad (15)$$

$$Z_{ipk[r+1]} \leq \sum_{i=1}^{n}\sum_{\substack{q \in A(q_i),\\ q \neq p}} Z_{iqk[r]}$$
$$\forall i, k, r, p \in A(p_i) \qquad (16)$$

$$\sum_{k \in F_p}\sum_{r=1}^{b} Z_{iqk[r]} = 0$$
$$\forall i, p, N_{ip} = 0 \qquad (17)$$

$$\sum_{r=1}^{b} Z_{ipk[r]} = 0 \qquad \forall i, p, k \notin F_p \qquad (18)$$

$$C_{max} \geq 0 \qquad (19)$$

$$C_{k[r]} \geq 0 \qquad \forall\, k, r \qquad (20)$$

$$Z_{ipk[r]} \in \{0, 1\} \qquad \forall\, i, p, k, r \qquad (21)$$

The 6th and 7th constraints are in order used to calculate the compilation time of first position for its machine and other positions. The 8th equation specifies the operations domain, all work's accomplishment time. In complex integer models, usually, as variables increase, the problems dimension would increase exponentially and this causes to increase the solution time. In this study, therefore, it has been tried to combine the variables to minimize the number of variables. So, the 9th constrain is used to compute the parts' compilation time by regarding the position's compilation. As 9th equation makes the model nonlinear, the 10th and 11th equations are used to linearize it. It should be mentioned that after this linearization, the 9th constrain can be eliminated. The 13th constrain indicates that the start time of a typical part (p), is equal or bigger than compellation time of prerequisite parts (q). The 14th constrain insures the part is processed only in one machine and in one position. The15th equation shows that in each position, just one or no machine has to be placed. The 16th constrain insures that the continuity of operation sequences. It means that the position is not completed until its previous positions are not completed. If parts p does not belong to product i, z values on all machines have to be zero. (Constrain 17) The 18th equation indicates that activities are scheduled on machines which can be processed by them.

Many studies have illustrated that APS is a kind of NP-Hard problems. [2, 8] also, solving such problems need much more calculations and attempts to be solved by considering learning effect [16] because in this case, a new index is added to the problem's decision variables. It can be proved that this paper's proposed model is a kind of NP-Hard problems and if α, β=0, the APS models in the literature are obtained which all are NP-Hard. As in NP-Hard problems the exact methods cannot be applied to get appropriate answer in reasonable computational times, in the next section and multi-stage genetic algorithm is proposed to solve the model.

4 THE PROPOSED ALGORITHM

Genetic algorithm is one of the most famous and widely used methods to solve different problems and specially APS problems. This concept was applied for the first time by Jan Holland University of Michigan, Ann Arbor. Then he and his students developed this algorithm which is categorized us stochastic search techniques and used to optimize complex problems with unknown search space models. [22] This algorithm includes a series of steps and important components which if they would not be defined properly, it will be unable to find problems' proper solutions. They would display the possible answers, create the first answer, and calculate the fitness of function, operators and stop the calculation, which are described individually here:

4.1 Displaying the possible solutions

APS has a serious of prerequisite constrains as well as alternative machines to produce the products. In this case, to create a possible answer, displaying way of answers has to consider all possible solutions. In this paper, a multi-stage method, the best and the most common method in literature of APS, is applied to display the solutions.

Generally, coding process of solutions includes two stages: in the first stage, the sequence of activities has been defined. By fixing this sequence, in the second stage, one machine (between the allowed machines) is selected for each part; therefore, a two-dimensional chromosome has been specified, one dimension includes border of sequences and the second one contains border of machines. For example, a product has four parts which has to be produced by three machines. If one part is parent and the others are children, one of the possible solutions can be coded into form of Table I. In this table, sequence of work accomplishment is 2, 4, 3 and 1 which produce them to machines 3, 1, 2 and 2 have been used.

TABLE I. DISPLAYING THE POSSIBLE SOLUTIONS

Activity sequence	1	3	4	2
Machine number	2	2	1	3

4.2 Creating the initial solution

After specifying the coding method of possible solutions, several answers are randomly generated to create the first generation and this stage has two steps:

1. Generating the possible sequences in which the random keys concept has been used. In this method or parts have been firstly put in the unscheduled item list (UIL) and for each one, a unique random number in interval [0, 1] is generated. Then a part with the minimum random number is chosen as a candidate to locate in the first scheduling program. Two modes occur:
   - The part has a prerequisite in which is returned to UIL and another part is chosen the same mechanic.
   - The part has no prerequisite in which the part is scheduled and by its elimination the next parts become free

Determination of the second and next activities is performed by this same mechanism. When the first step is completed, by holding the obtained sequences, the second step begins.

2. In the second step, a machine is randomly chosen between allowed machines to process the part.

4.3 Fitness function

Assume that the current answer of the target function is f. As this problem is a kind of minimizing one is inverse function is used. As target function's value in large dimensions is big and its inverse does not show the differences between solutions well, the ranking method has been used in this study.

The next stage is creation of the second generation based on the present generation by using genetic operators. These operators are choosing, regenerating, intersection and mutation.

4.4 Crossover operator

This operator is used to produce children from parent chromosomes in the mating pool. The most famous crossover operators are one-point, two-point and order-based crossovers. Due to special structure of APS which has prerequisite relationships and alternative machines, these operators usually produce infeasible children in the first dimension of chromosomes (order of activities). As order-based crossover insures that the produced children are feasible, it is widely applied in APS literature. [3, 10]

4.5 Mutation operator

In this operator which is used in this study, one part is randomly chosen and its machine is replaced by one of the other feasible machines.

4.6 Stopping criteria

Achieving to a fixed number of generations or worsening the target function in specified number of algorithm interactions are the stopping criteria in proposed algorithm.

5 EVALUATION AND VALIDATION OF RESULTS

As this research's innovation are in problem's modeling and solving, results' computational analysis is performed in two sections, model validation and its efficiency.

5.1 Model's validation

In this section model's validation is illustrated by providing a small-size example and comparing its exact solution with the proposed model's results. Consider a three-part product in which second and third parts are prerequisite of the first part. Process normal time and production cost of each part on allowed machines have shown in table II.

TABLE II. PARTS' NORMAL PROCESS TIME AND VARIATION PRODUCTION COST

Part number	**Machines**	
	Machine 1	***Machine 2***
1	-	(5, 50)
2	(15, 80)	(10, 100)
3	(10, 70)	(5, 100)

Idle cost of system is 50 monetary units per an hour and each working day is 8 hours. Machines' access times are 0 and 3 in order. Learning index is -0.322 and incompressibility factor is 1.5. Product's lead time is 2 days from the customer's demand time. The target is the optimum scheduling so that total accomplishment time of parts becomes minimum. This goal is C_{max}=17.5 by using the exact solution (full counting method). This numerical example was coded and run in the GAMS 22.2 which its report including the target function and the mail variables have been illustrated in Table III. According to the tables III and 4, the solution is the same in both methods which indicates that the proposed model has the ability to model the problem and find its right solution.

TABLE III. GAMS SOLUTION FOR THE NUMERICAL EXAMPLE

	LOWER	LEVEL	UPPER	MARGINAL
---- *VAR Z_obj*	-INF	17.500	+INF	.
---- *VAR cmax*	.	17.500	+INF	.

5.2 Algorithm validation and results' analysis

In order to evaluate the proposed algorithm's efficiency, 27 large-size, medium-size and small-size problems were created with their relevant data have

been shown in table IV. These data have been derived from previous researches [8, 12] and were randomly generated in the required cases.

TABLE IV. APPLIED DATA RELATED TO THE 27 PROBLEMS

Parameters	Values			Modes
	Small-sized	*Medium-sized*	*Large-sized*	
n	5, 8, 12	15, 20, 30	50, 70, 100	27
m	2, 5, 10	5, 10, 15	10, 15, 20	
t_{ipk}	U[0, 1]	U[0, 1]	U[0, 1]	1
RT_k	U[0, 30]	U[0, 30]	U[0, 30]	1
α, β	-0.322	-0.322	-0.322	1
Total				27

Due to randomly nature of genetic algorithm, it was run 10 times for each problem and the best answer was chosen as the final solution of problem. The most appropriate values of parameters were gained from numerical experiments to increase the quality of multi-stage genetic algorithm (MSGA) which have shown in table V. In order to evaluate the proposed algorithm, quality of solutions (target function's value) and the run-time was used. By considering these two criteria, the exact solution (GAMS) as well as the proposed algorithm was compared in small, medium and large scale which are discussed in the next section.

TABLE V. APPROPRIATE VALUES FOR ALGORITHM PARAMETERS

Parameters	Values		
	Small-sized	*Small-sized*	*Small-sized*
crossover rate	0.6	0.6	0.65
mutation rate	0.2	0.2	0.65
population size	10	10	15
Generation's Repeat number	500	800	1000
regeneration rate	0.5	0.5	0.5

5.2.1 MSGA and GAMS solutions' quality

The obtained solutions for the problems in different scales have been shown in table 4 and figure 5. Propose of comparison of these methods in small scale is to evaluate the MSGA ability to reach the optimized or near optimized solutions. In problems 1, 2, 3 and 7, both solutions are the same and in the problem 4, MSGA's solution has 4 % deviation from the optimized solution which demonstrates the proposed algorithm's ability to achieve appropriate solutions. Also, GAMS doesn't have the ability to achieve the optimized solutions for problems 5, 6, 8 and 9 in the specified time (4000 seconds) and in their related value in the table VI the symbol (-) has been put.

The propose of MSGA and GAMS algorithms' comparison in medium and large scale is demonstration of MSGA's ability to achieve optimized or near optimized solution and exact solution's inability to solve them.

According to table VI and figure I, in medium and large scale size problems (10 to 27) GAMS software does not have even the ability to achieve a feasible solution in the defined computation time (4000 seconds); therefore, there is no diagram for problems 10 to 27 obtained from exact solution. Regarding to results of different size problems, it can be safely figured out that MSGA is efficient to solve different size problems.

5.2.2 Comparison of MSGA and GAMS solution's run-time

One of the most important criteria to evaluate the performance of an algorithm is its run-time which in some papers is called execution speed. This factor becomes more important when the complexity and size of the problem in increased. Run-time data of both algorithms have been shown in table VI and figure I. By considering the significant difference of run-time between exact and proposed solution, their run-time comparison does not seem appropriate; therefore, analysis if the proposed algorithm in different scales in only performed. Generally, according to the figure I, MSGA's run-time is increased exponentially as the size of the problem, number of activities, especially after problem 17 is increased because it depends on n rather than m ($O(n^2m)$). So, beyond the problem 17, each three consecutive points which have equal n and different m, have approximately the same run-time and when n varies, like variation between 21 and 22 points, have had a significant and sudden increase in their run-time.

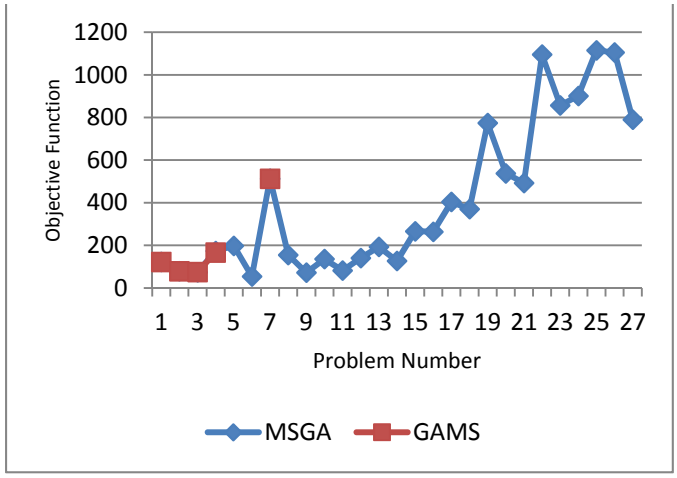

Fig. 1. MSGA and GAMS solutions

6 CONCLUSION

Unlike the current assumption in APS problems, work processing time is not constant and independent from production sequence and one of phenomenon which could be seen by considering learning effect.

Due to this problem's complexity, a multi-stage genetic algorithm was proposed and its efficiency was evaluated.

Achieved results demonstrated that the proposed algorithm has only 4 % deviation in one example out of 27 one and in other examples which have the exact solution, has the ability to find the optimized solutions.

By considering the applicable nature of the proposed model, it is recommended that in the future researches, this problem would be optimized by combining this paper's algorithm with other meta-heuristic algorithms to boost its efficiency.

TABLE VI. GAMS AND MSGA DATA AND SOLUTIONS

Dimension	Problem No	n×m	GAMS		MSGA		
			Objective function	*run-time (seconds)*	*Objective function*	*standard deviation*	*run-time (seconds)*
Small	1	5×2	122.5	162	122.5	0	0.3
	2	5×5	79	3.9	79	0	0.3
	3	5×10	74.6	23.2	74.6	0	0.6
	4	8×2	167	15.4	174.7	0	0.5
	5	8×5	-	-	198.1	0.03	0.7
	6	8×10	-	-	54.2	17.28	0.7
	7	12×2	513	56.5	513.3	0	0.8
	8	12×5	-	-	154.6	19.40	1.2
	9	12×10	-	-	77.2	17.37	1.5
Medium	10	15×5	-	-	136	9.90	1.1
	11	15×10	-	-	82.9	17.70	2.2
	12	15×15	-	-	141	7.94	1.6
	13	20×5	-	-	193.2	11.47	3.2
	14	20×10	-	-	126.5	17.52	3.0
	15	20×15	-	-	265.8	27.04	3.0
	16	30×5	-	-	563.6	58.48	4.5
	17	30×10	-	-	404.4	46.49	4.3
	18	30×15	-	-	369.6	30.25	3.0
Large	19	50×10	-	-	774.1	57.54	13.1
	20	50×15	-	-	537.5	70.97	15.2
	21	50×20	-	-	493.6	65.13	11.8
	22	70×10	-	-	1095	91.11	21.8
	23	70×15	-	-	856.8	74.98	22.1
	24	70×20	-	-	901	80.08	17.6
	25	100×10	-	-	1115.1	121.12	24.4
	26	100×15	-	-	1105.5	136.53	24.2
	27	100×20	-	-	790.3	96.45	26.6

7 ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments and feedbacks to enhance the quality of the paper.

8 REFRENCES

[1] Gen, M., Cheng, R. and Lin, L. (2008), Advanced Planning and Scheduling Models, in Network Models and Optimization Decision Engineering, p.p. 297-417.

[2] Moon, C. and Seo, Y. (2005), Evolutionary algorithm for advanced process planning and scheduling in a multi-plant, Computers & Industrial Engineering, vol. 48, p.p.311-325.

[3] Moon, C., Seo, Y. and Yun, Y. (2006) "Adaptive genetic algorithm for advanced planning in manufacturing supply chain," J Intell Manuf, vol. 17: 509–522.

[4] Blazewicz, J., Ecker, K. H., Schmidt, G. and Weglarz, J., (1994), Scheduling in Computer and Manufacturing Systems, Springer.

[5] Bermudez, J., (1998), Advanced Planning and Scheduling: Is it as good as it sounds? The report on Supply Chain Management, March, p.p. 3-18.

[6] Marjolein and van Eck, (2003), Is logistics everything, a research on the use(fullness) of advanced planning and scheduling systems, in BWI paper, April.

[7] Lee, Y.H., Jeong, C.S. and Moon, C., (2002), Advanced planning and scheduling with outsourcing in manufacturing supply chain, Computers & Industrial Engineering, vol. 43, pp. 351-374,.

[8] Chen, K. and Ji, P. A, (2007), mixed integer programming model for advanced planning, European Journal of Operational Research, vol. 181, pp. 515–522.

[9] Yang, J., Tang, W., (2009), Preference-based adaptive genetic algorithm for multiobjective advanced planning and scheduling problem, IEEE International Conference, pp. 1935 - 1940.

[10] Zhang, H. and Gen, M., (2006), Effective Genetic Approach for Optimizing Advanced Planning and Scheduling in Flexible Manufacturing System, GECCO, pp. 1841-1848. pp. 1841-1848.

[11] Örnek, A., Özpeynirci, S. and Öztürk, C., (2010), A note on ''A mixed integer programming model for advanced planning, European Journal of Operational Research, vol. 203, pp. 784-785.

[12] Chen, K., JI, P. and Wang, Q., (2011), A case study for advanced planning and scheduling (APS), J Syst Sci Syst Eng, Dec, vol. 20, no. 4, pp. 460-474.

[13] Okołowski, D. and Gawiejnowicz, S., (2010), Exact and heuristic algorithms for parallel-machine scheduling with DeJong's learning effect, Computers & Industrial Engineering, vol. 59, pp. 272–279.

[14] Zegerdi, S. and Bahlouli, E., (2009), Group scheduling by considering learning effect in cellular manufacturing system. International Journal of Industrial engineering and Production Management, vol. 2

[15] Biskup, D. A., (2008), state-of-the-art review on scheduling with learning effects, European Journal of Operational Research, vol. 188, pp. 315–329.

[16] Wright, T.P., (1936) Factors affecting the cost of airplanes, Journal of Aeronautical Sciences 3, pp. 122–128.

[17] Biskup, D., (1999), Single-machine scheduling with learning considerations, European Journal of Operational Research, vol. 115, pp. 173-178.

[18] Wang, J-B. and Wang, J-J., (2012), Scheduling jobs with a general learning effect model, Appl. Math. Modelling, pp. article in press.

[19] Zhang, X., Yan, G., Huang, W. and Tang, G., (2012), A note on machine scheduling with sum-of-logarithm processing-time-based and position-based learning effects, Information Sciences, vol. 187, pp. 298–304.

[20] Janiak, A. and Rudek, R., (2008), A new approach to the learning effect: Beyond the learning curve restrictions, Computers & Operations Research, vol. 35, pp. 3727 – 3736.

[21] Amini, A., Tavakkoli Moghaddam, R. and Niakan, F. A, (2011), Multi-Objective Identical Parallel Machine Scheduling with Setup and

Removal Times with Deteriorating and Learning Effects, in Industrial Engineering and Engineering Management (IEEM), IEEE International Conference on, Singapore, pp. 1271 - 1274.

[22] Hamta, N., Fatemi Ghomi, S.M.T., Asiabar, M.H. and Hooshangi Tabrizi, P., (2011), Multi-objective Assembly Line Balancing Problem with Bounded Processing Times, Learning Effect, and Sequence-dependent Setup Times, in Industrial Engineering and Engineering Management (IEEM), 2011 IEEE International Conference on, Singapore, pp. 768-772.

[23] Hamta, N., FatemiGhomi, S.M.T., Jolai, F. and Akbarpour Shirazi, M., (2012), A hybrid PSO algorithm for a multiobjective assembly line balancing problem with flexible operation times, sequence-dependent setup times and learning effect, Int. J. Production Economics: http://dx.doi.org/10.1016/j.ijpe.2012.03.013.