Journal of Quality Engineering and Production Optimization
Vol. 1, No. 1, PP. 21-32, 2015

Bi-objective Optimization for Just in Time Scheduling: Application to the Two-Stage Assembly Flow Shop Problem

Sahar Tadayoni Rad[1], Saiedeh Gholami[1], Rasoul Shafaei[1], and Hany Seidgar[2]
[1] *Department of Industrial Engineering, K.N. Toosi University of Technology, Tehran, Iran,*
[2] *Department of Industrial Engineering, Mazandaran University of Science and Technology, Babol, Iran*

***Corresponding Author:** Sahar Tadayoni Rad (E-mail: s.tadayonirad@yahoo.com)

Abstract- *This paper considers a two-stage assembly flow shop problem (TAFSP) where m machines are in the first stage and an assembly machine is in the second stage. The objective is to minimize a weighted sum of earliness and tardiness time for n available jobs. JIT seeks to identify and eliminate waste components including over production, waiting time, transportation, inventory, movement and defective products.Two-stage assembly flow shop is a combinational production system in which different parts are manufactured on parallel machines independently. This system can be used as a method to produce a variety of products through assembling and combining different set of parts. We apply ε-constraint method as an exact approach to validate the proposed model and to obtain fronts of the solutions in the solution spaceThe goal of the proposed problem is trade off between two objectives, minimization makespan and total weighted tardiness and earliness. To analyze effects of n and m factors on the efficiency and performance of the proposed algorithm, we calculate the complexity of sub problems based on factors n and m and the computational results demonstrate that the computational time increases with increasing in n and m, in other words, complexity of the problem increases.*

Keywords: *Two-Stage Assembly flow shop problem, Just in time scheduling, ε-constraint method.*

I. INTRODUCTION

International competition and the ability to respond to the variable demand in the markets are some of the key attributes in designing effective production systems. Two-stage assembly flow shop is a combinational production system in which different parts are manufactured on parallel machines independently. This system can be used as a method to produce a variety of products by assembling and combining different set of parts. Assembly flow shop scheduling problem (AFSP) was introduced by Lee et al. (1993) and Potts et al. (1995). AFSP is a type of flow shop problem that at first each of n jobs has to be processed at the first stage consisting of m parallel and independent machines then they are assembled at the second stage including only one assembly machine (Elmasri & Navathe). Assembly type production systems are as a reply to the market pressure and change for larger product variety.

In an assembly flowshop scheduling problem there are n jobs where each job has $m + 1$ operations and there are $m + 1$ different machines to perform each of these operations. Each machine can process only one job at a time. For each job, the first m operations are conducted at the first stage in parallel and a final operation in the second stage. Each of m operations at the first stage is performed by a different machine and the last operation at the second stage may start only after all m operations at the first stage are completed.

The two-stage assembly scheduling problem has many applications in industry. Potts et al. (1995) described an application in personal computer manufacturing where central processing units, hard disks, monitors, keyboards, etc. are manufactured at the first stage, and all the required components are assembled to customer specification at a packaging station (the second stage). Lee et al. (1993) described another application in a fire engine assembly plant. The body and chassis of fire engines are produced in parallel in two different departments. When the body and chassis are completed and the engine has been delivered (purchased from outside), they are fed to an assembly line where the fire

Manuscript Received: 8 June 2014 and revised: 17 Nov 2014 *ISSN: 2423-3781*
Accepted: 28 Dec 2014

engine is assembled. There are many other problems that can be modeled as an assembly flowshop scheduling problem, including queries scheduling on distributed database systems.

In recent years, there has been a rapid trend toward the distribution of computer systems over multiple sites that are interconnected via a communication network [9]. This new architecture has raised many new challenges and problems in the field of database systems. For example, it is common with current technology to develop forms or reports that require tens of embedded queries that retrieve information from different sites on the networks (Ceri &Pelagatti).

Just in Time (JIT) seeks to identify and eliminate waste components as over production, waiting time, transportation, inventory, movement and defective products. In JIT environment, jobs which are completed earlier than its due date may cause such opportunity costs, deterioration of product and inventory holding cost. Also, tardiness may cause missing customer, contract penalties, loses of sale and loss of reputation . An important special case in the family of earliness and tardiness problems involves minimizing the sum of absolute deviations of job completion times from a due. It is clear that an ideal schedule is the one in which all jobs are completed exactly on their due dates. Hence, the criterion which involves both earliness and tardiness costs has received considerable attention in these decades due to their practical importance and relevance.

Due to the importance of the JIT scheduling in industrial environment to overcome the variations of the demand and to provide customer satisfaction, in this research a comprehensive model is developed elaborated by several aspects of scheduling.

JIT scheduling are leading the environment to a very elegant model being very substantial for nowadays competitive industry. Recently, factories are equipped with automation and automatic unmanned manufacturing machines and tools demands for specific and subtle planning. Also, markets and customers ask for on time delivery to evaluate and classify the industrial departments.

The remainder of this study is organized as follows: Section 2 gives the literature review of scheduling assembly flow shop problem. Section 3 describes the problem and introduces notations and proposed mathematical model. Section 4 consists of the proposed algorithm for finding the pareto front. Section 5 presents numerical examples to explain the behavior of the proposed model and the efficiency of the proposed algorithms for solving the problem. Finally, Section 6 presents the conclusions and directions for future works.

II. LITERATURE REVIEW

AFSP was introduced by Lee et al. (1993) and Potts et al. (1995). Lee et al. (1993) studied AFSP with considering two machines at the first stage, while Potts et al.(1995) considered the same problem with an arbitrary machine at the first stage. Both studies proved that AFSP for two machines at the first stage with objective of minimizing the makespan is Np-hard. Potts et al. (1995) showed that the search for an optimal solution may be limited to permutation schedules.The assembly flowshop scheduling problem has many applications in the industry. Most productions which have assembly as part of their production procedure can be categorized in such problems. Computer manufacturing (Hariri & Potts,1997) and fire engine production (Haouari & Daouas) are two examples. Lee et al. (1993) introduced the two-stage assembly flowshop problems with two machines in the first stage. Potts et al. (1997) presented this problem with an arbitrary number of machines in the first stage; the objective in this problem was to minimize makespan. They proved that these problems are NP-hard. Lee et al. (1993) solved it with a branch and bound algorithm on state m =2, and surveyed three heuristics for this problem. Hariri & Potts (1997) proposed a branch and bound algorithm that incorporated dominance relations with the lower bound. Haouari & Daouas (1999) also proposed another branch and bound algorithm. Moreover, Sun et al. (2003) introduced several heuristics for these problems. Tozkapan et al. (2003) considered the two-stage assembly scheduling problem, also Al-Anzi & Allahverdi (2006) investigated several heuristics for these problems with completion time criterion. Moreover, the problem is investigated based on makespan criterion (Al-Anzi & Allahverdi,2006)), maximum lateness criterion (Al-Anzi & Allahverdi,2009; Shokrollahtabar et al., 2010) and total completion time (Torabzadeh & Zandieh, 2010).

Shokrollahpour et al. (2010) studied TAFSP and proposed an imperialist competitive algorithm (ICA). They compared the performance of the proposed ICA with the SA method by Allahverdi & Al-Anzi (2006) . The results demonstrate that ICA reaches to better solutions than SA but it takes more time. Torabzadeh & Zandieh (2010) considered the two-stage assembly flow-shop problem with aim to minimize a weighted sum of makespan and mean completion time as the objective for *n* available jobs. They proposed the cloud theory-based simulated annealing algorithm (CSA) to solve it and compared CSA and SA in their study. The computational results revealed that CSA performs better. In addition, computational time has been decreased for the CSA algorithm towards the SA. Khorshidian et al. (2011) studied the single machine scheduling problem with preemption in JIT environment and proposed a genetic algorithm to solve problem. Kayvanfar et al. (2013) studied the single machine scheduling with controllable processing times to minimize total tardiness and earliness. They proposed three heuristics in the small problem and two meta-heuristics in medium to large problem as effective local search methods. Birgin & Ronconi (2012) addressed the single machine scheduling problem with a common due date and non-identical ready times for each job. The objective was to minimize the weighted sum of earliness and tardiness penalties of the jobs. Since this problem is NP-hard, the application of constructive heuristics that exploit specific characteristics of the problem to improve their performance was investigated. Mahnam et al. (2013) survived the single machine scheduling problem with unequal release times and idle insert for minimizing the sum of maximum earliness and tardiness. They proposed two dispatching rules and meta-heuristics including genetic algorithm and particle swarm optimization to solve the problem. Mozdgir et al. (2013) addressed the two-stage assembly flow shop problem with multiple non-identical assembly machines in the second stage to minimize the weighted sum of makespan and mean completion time. Also, sequence dependent set up times are considered at the first stage. They used a variable neighborhood search (VNS) algorithm and developed a novel heuristic method and compared the corresponding results with the results obtained by GAMS. Computational experiments revealed the hybrid VNS heuristic performs much better than GAMS with respect to the percentage errors and computational time. Seidgar et al. (2014) considered the two-stage assembly flow shop problem with aim of minimizing a weighted sum of makespan and mean completion time for *n* available jobs. They employed an imperialist competitive algorithm (ICA) as solution approach. To justify the claim for ICA capability, they compared their proposed ICA with CSA introduced by Torabzadeh & Zandieh (2010). They applied a new parameters tuning tool, neural network, for ICA. The computational results clarified that ICA performs better than CSA in quality of solutions. [12] investigated TASFP to minimize a weighted sum of makespan and earliness and tardiness. They applied a hybrid neighbourhood search- electromagnetism-like mechanism (VNS-EM) to solve the problem. The computational results demonstrated the proposed hybrid VNS-EM algorithm outperforms the EM and VNS algorithms in terms of both average value and standard deviation. [24] focused on the distributed two-stage assembly flow-shop scheduling problem to minimize a weighted sum of makespan and mean completion time. They proposed a mathematical model and solved the small sized instances of the proposed problem. Due to the NP-hardness of the problem, they presented a variable neighborhood search (VNS) algorithm and a hybrid genetic algorithm combined with reduced VNS (GA-RVNS) to solve the problem. Computational results have been conducted to compare the performances of the model and the proposed algorithms. For the small-sized instances, both the model and the proposed algorithms were effective. The proposed algorithms were further evaluated on a set of large-sized examples. The computational experimental statistically demonstrated GA-RVNS and VNS reach much better performances than the GA without RVNS-based local search step (GA-NOV).

To the bset of our knowledge, most studies that addressed the assembly flow shop problem are single objective. There is no related study to consider the multi-objective optimization environment. Therefore, this paper investigates the bi-objective optimization in the two stage assembly flow shop problem in the just in time environment and uses ε-constraint method to solve the problem and reach pareto solutions for minimization of the makespan and total weighted earliness and tardiness.

III. PROBLEM DEFINITION

In this paper, we consider two stage assembly flow shop problem (TAFSP) in which *m* parallel independent machines are at the first stage and one assembly machine ($m+1th$ machine) is in the second stage. In this problem *n* jobs should be processed in the both stages and each job is composed of $m+1$ operations. The objectives of this problem are minimization of the makespan and total the weighted sum of the tardiness and earliness simultaneously.

This problem is used in the electricity market such as: Television market. Fig.(1), illustrates the concept of TAFSP graphically.

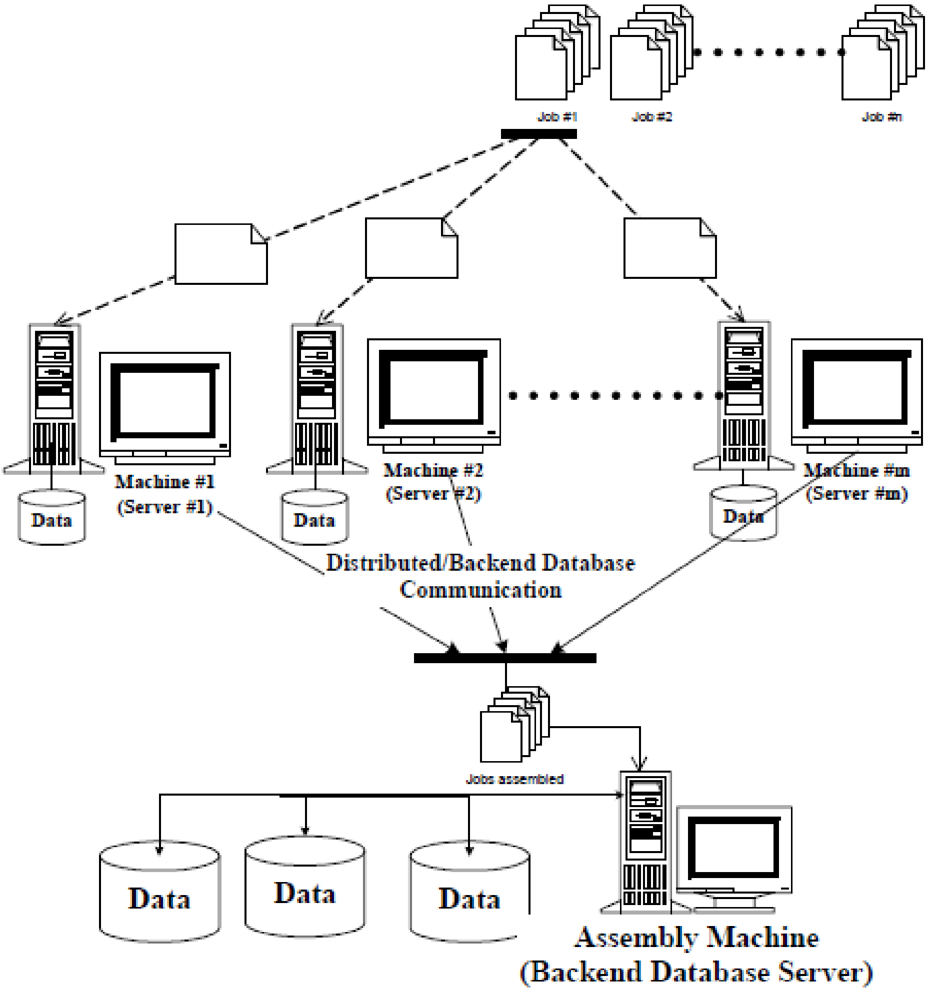

Fig. 1: Two stage assembly flow shop problem (TAFSP) [13]

Indices, parameters and decision variables are presented as follows:

Indices	
j	Index of job $j=1,\ldots,n$
i	Index of position in the sequence of jobs $i=1,\ldots,n$
k	Index of machine $k,=1,\ldots,m+1$
Parameters	
n	The number of jobs
m	The number of parallel and independent machines at the first stage
P_j	The processing time of job j at the assembly stage
t_{jk}	The processing time of job j on the k th parallel machine at the first stage
α_j	Earliness cost of job j per unit time
β_j	Tardiness cost of job j per unit time

Decision variables

x_{ji}	$\begin{cases} 1 & \text{if } j\text{th of job is assigned in position } i \text{ in the sequenc} \\ 0 & \text{otherwise} \end{cases}$
C_{ik}	Completion time of the job in position *i* on the *k th* machine
C_{max}	Maximum completion time (Makespan)
E_i	Earliness of job *i*
T_i	Tardiness of job *i*

Mathematical model

According to the above mentioned parameters and decision variables, the bi-objective mathematical model is as follows:

$$OB1 = C_{max} \quad (1)$$

$$OB2 = \sum_{j=1}^{n} \alpha_j E_j + \beta_j T_j \quad (2)$$

$$\sum_{j=1}^{n} x_{ji} = 1 \qquad \forall i = 1, \dots, n \quad (3)$$

$$\sum_{i=1}^{n} x_{ji} = 1 \qquad \forall j = 1, \dots, n \quad (4)$$

$$C_{0k} = 0 \quad (5)$$

$$C_{ik} \geq C_{i-1\,k} + \sum_{j=1}^{n} t_{jk}.x_{ji} \qquad \forall i = 1, \dots, n, \ \forall k = 1, \dots, m \quad (6)$$

$$C_{im+1} \geq C_{i-1\,m+1} + \sum_{j=1}^{n} P_j.x_{ji} \qquad \forall i = 2, \dots, n \quad (7)$$

$$C_{im+1} \geq C_{i\,k} + \sum_{j=1}^{n} P_j.x_{ji} \qquad \forall i = 1, \dots, n, \ \forall k = 1, \dots, m \quad (8)$$

$$C_{max} = C_{n\,m+1} \quad (9)$$

$$T_j \geq C_{im+1} - D_j - M(1 - x_{ji}) \qquad \forall i, j \quad (10)$$

$$E_j \geq D_j - C_{im+1} - M(1 - x_{ji}) \qquad \forall i, j \quad (11)$$

$$x_{ji} \in \{0,1\} \quad (12)$$

$$C_{ik}, C_{max}, T_j, E_j \geq 0 \quad (13)$$

Objective functions makespan and the weighted sum of earliness and tardiness are shown in Eqs. (1) and (2), respectively. Note that when in Eq.(2), the weight of each criterion is equal to 0 or 1, the problem is reduced to the single criterion of E_j or T_j, respectively. Eq. (3) ensures that each position in the sequence of jobs is only assigned to one job. Eq. (4) ensures that each job is assigned only to one position in the sequence of jobs. In other words, Eqs. (4)

and (5) determine the sequence of jobs. Eqs.(6) and(7) determine the completion time of jobs at the first stage and Eqs.(7) and(8) calculate the completion time of jobs at the assembly machine. Eq. (9) returns the value of makespan. Eqs. (10) and (11) return the value of tardiness and earliness, respectively. Eqs. (12) and (13) provide limits on the decision variables.

IV. SOLUTION METHOD

In this paper, we propose the use of a mathematical programming technique called the ε-constraint method. The ε-constraint method uses the cultured differential evolution to produce one point of the Pareto front of a multi-objective optimization problem at each iteration. This approach is able to solve difficult multi-objective problems, relying on the efficiency of the single-objective optimizer.

The ε-constraint method is probably the best known technique for solving multi-objective discrete optimization problems. It ensures the exact set of the efficient solutions in the solution space of the problems.

The ε-constraint is a repetitive methodwhich starts its work with optimizing one of the selected objectives of the problem by considering all constraints of the problem. The selected objective function is referred to as the base objective function (Arkat et al., 2010).

An exact ε-constraint method tackles multi-objective optimization problems by solving a series of single objective sub-problems, where all but one objective are transferred to constraints. We show that the Pareto front of bi-objective problems can be efficiently generated with the ε-constraint method [1], [8].

V. NUMERICAL EXAMPLES

In this section, we present two examples to elaborate the behavior of the new bi-objective mathematical model and the efficiency of the proposed method for solving the proposed problem. Minimization the sum of earliness and tardiness of jobs is considered as the base objective for the ε-constraint.

Algorithm 1: ε-constraint method fo bi-objective minimization problem

Step 1: Select one of the objectives randomly. (Consider $f_1(X)$ for example)

Step 2: Set *i*=1

Step 3: Solve the following model to find the global optimum solution, namely X_1 :

$$min\ f_1(X)$$

Subject to: $X \in (S_1 \cap S_2)$

If we yield a feasible solution for the above model, go to Step 5 else, go to Step 8

Step 4: Solve the below model to find the global optimum solution, namely X_1

$$min\ f_1(X)$$

Subject to: $X \in (S_1 \cap S_2)$

$$f_2(X) \leq f_2(X_{i-1}) - \varepsilon$$

If we yield a feasible solution for the above model, go to Step 5 else, go to Step 8.

Step 5: Calculate the objective function values for both objectives namely $f_1(X_i)$ and $f_2(X_i)$

Step 6: Add the below constraint to the original constraints:

$$f_2(X) \leq f_2(X_i) - \varepsilon$$

Step 7: Set *i*=*i*+1 and go to Step 4.

Step 8: The exact set of the efficient solutions is X_i.

Stop

Note: $f_i(X)$ is the *i*th objective; X_i is the *i*th efficient solution; S_i is the solution space for *i*th objective; and $f_1(X_j)$ is *i*th objective value in X_j

TABLE I. DATA FOR (5JOBS, 2 MACHINES) PROBLEM, INSTANCE FOR NUMERICAL EXAMPLE 1

j	1	2	3	4	5
t_{j1}	8	8	6	3	4
t_{j2}	6	5	4	4	4
P_j	4	3	2	2	5
D_j	12	18	6	8	22
α_j	2	3	2	1	3
β_j	5	6	7	5	6

These values are obtained in the first round for the numerical example by the ε-constraint method:
$Cmax_1$=33
TET_1=170

The constraint (14) should be added to the model:

$$Cmax_1=33-\varepsilon \quad (14)$$

The ε is assumed equal to 1 in these calculations. The following objectives values are found in the second round of the ε-constraint method:
$Cmax_1$=32
TET_1=181

This procedure is iterated until no feasible solution is found. The results for the numerical example are represented in Table II.

Proposed bi-objective methodology reached pareto solutions with three non-dominated solutions in seven seconds that these solutions are reported in Table II.

In Fig (2), the front of the solutions for the first numerical example is illustrated graphically.

For more analyzing, we present the second numerical example in Table III.

TABLE II: RESULTS OF THE ε-CONSTRAINT METHOD IN THE FIRST NUMERICAL EXAMPLE

Round	Number of *Cmax*	Number of *TET*	Efficiency	Time(Minute: Second)
1	33	170a	Efficient	00:03
2	32	181a	Efficient	00:02
3	31	231a	Efficient	00:02
4	-	Infeasible		
Total run time				00:07

[a] Global optimum

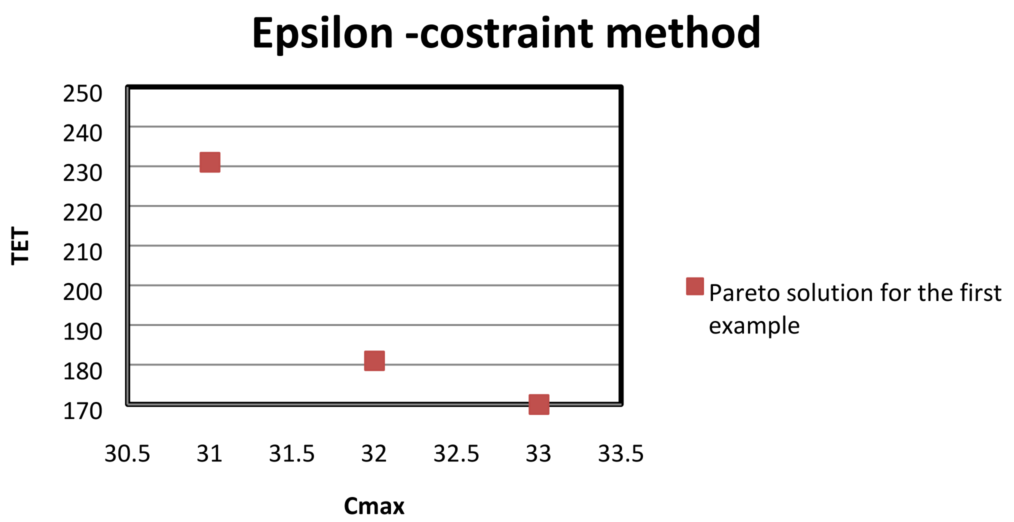

Fig. 2: Front of the solutions (Job; 5, Machine at the first stage; 2)

TABLE III: DATA FOR (5 JOBS, 2 MACHINES) PROBLEM, INSTANCE FOR NUMERICAL EXAMPLE2

J	1	2	3	4	5	6	7
t_{j1}	2	4	2	5	1	3	4
t_{j2}	1	3	2	4	3	2	4
t_{j2}	2	4	1	5	2	4	3
P_j	2	3	1	6	4	4	5
D_j	10	12	8	9	20	25	16
α_j	2	3	3	1	2	2	3
β_j	6	5	5	5	6	5	7

We solved the second proposed numerical example and presented the values of objective functions in the first round of the proposed methodology by the ε-constraint method:

$Cmax_1$=31

TET_1=128

The constraint (15) should be added to the model:

$Cmax_1$=31-ε (15)

The ε is assumed equal to 1 in these calculations. The following objectives values are found in the second round of the ε-constraint method:

$Cmax_1$=30

TET_1=135

This procedure is iterated until no feasible solution is found. The results for the numerical example are reported in Table IV.

For more understanding, we illustrated graphically the front of the solutions for the second numerical example in Fig.(3).

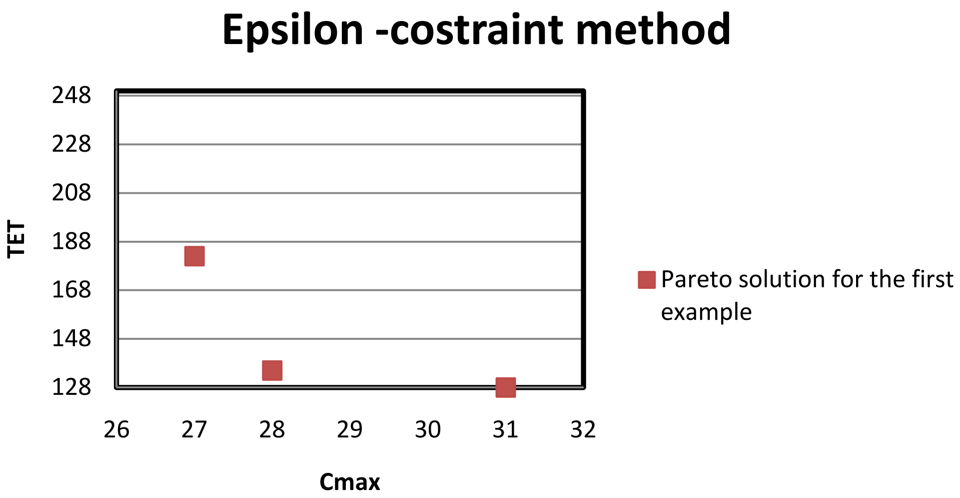

Fig. 3: Front of the solutions (Job; 7, Machine at the first stage; 3)

TABLE IV. RESULTS OF THE ε-CONSTRAINT METHOD IN THE SECOND NUMERICAL EXAMPLE

Round	Number of *Cmax*	Number of *TET*	Efficiency	Time(Minute: Second)
1	31	128a	Efficient	00:04
2	30	135a	-	00:07
3	29	135a	-	00:14
4	28	135a	Efficient	00:09
5	27	182a	Efficient	00:11
4		Infeasible		
Total run time				00:45

[a] Global optimum

To analyze effects of *n* and *m*factors on the efficiency and performance of the proposed algorithm, we calculate the complexity of sub problems based on factors *n* and *m*. For this purpose, *n*=5,6, 8,10,12 and15; *m*=2,3,4 and 5 are considered

The computational time for all combinations of number of jobs and number of machines at the first stage is demonstrated in Table V.

Fig.(4) demonstrates the computational time increases with increasing the number of jobs. In other words, complexity of the problem increases.

TABLE V. COMPUTATIONAL TIME FOR COMBINATION OF N AND M

n/m	Job:5	Job:8	Job:10	Job:12	Job:15
Machine:2	00:07	00:23	01:16	01:47	01:53
Machine:3	00:09	00:31	01:23	02:12	02:41
Machine:4	00:12	00:37	01:51	02:19	03:27
Machine:5	00:13	00:46	02:12	03:01	03:47

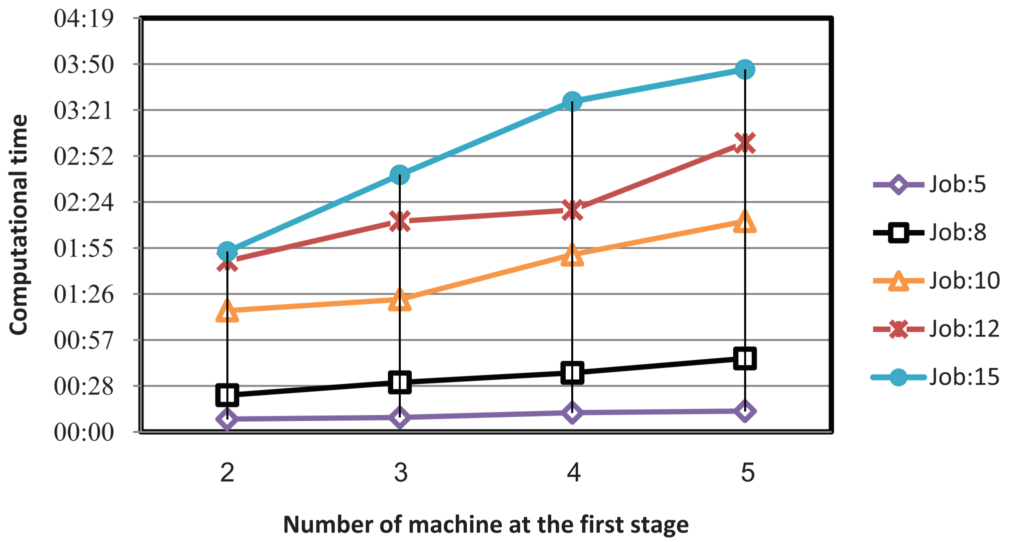

Fig. 4: Computational time under different number of machines

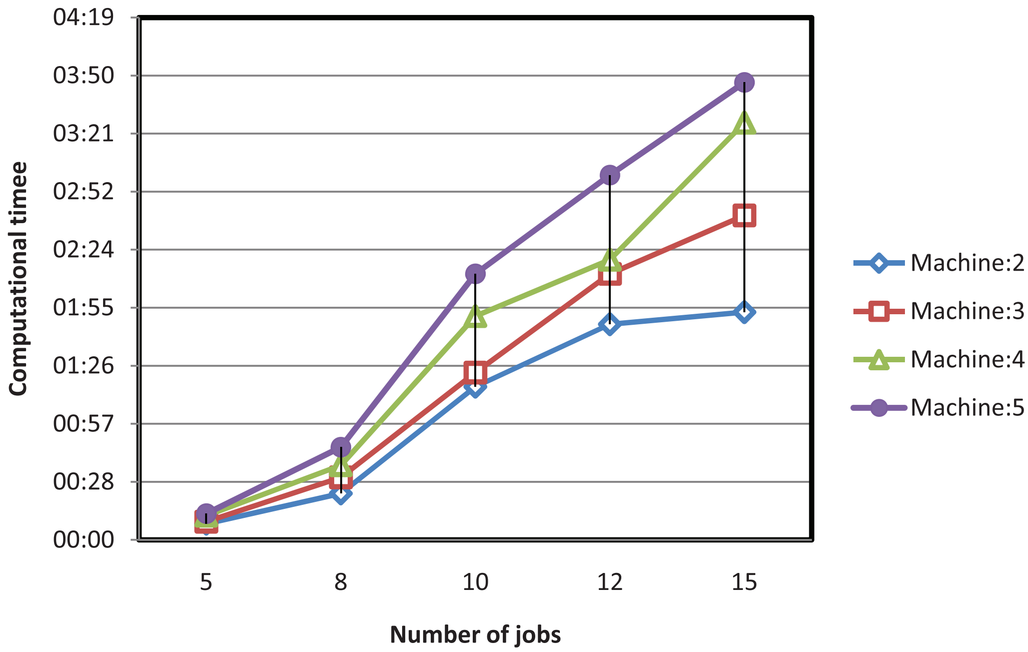

Fig. 5: Computational time under different number of jobs

Fig.(5) illustrates that the computational time increases with increasing the number of machines. In other words, complexity of the problem increases.

VI. CONCLUSION AND FUTURE RESEARCHES

This paper considered two-stage assembly flow shop problem (TAFSP) in a just in time enviroment. The goals of the problem were to minimize makespan and total tardiness and earliness (TET) simultaneously.We presented a new bi-objective mathematical model for just in time scheduling then applied ε-constraint method to obtain the front of the solutions. We analized effects of n and m factors on the efficiency and performance of the proposed algorithm and calculated the complexity of sub problems based on the proposed factors. Results demonstrated that the computational time increases by increasing in n and m, in other words, complexity of the problem was increased.

Since this problem was Np-hard, therefore it is worth to propose another solution approach and compare its performance with the current used solution approach to justify the computational results. Good lower bounds for TAFSP should be developed to evaluate the solution quality of the proposed algorithm. As a direction for future research, it would be interesting to consider the random machine breakdown on machines. Besides, other multi-objective evolutionary algorithms can be employed to find pareto-fronts for future researches. The algorithm can also be extended to a wider class of the proposed problem such as the problems with precedence constraints, learning effect and so on.

REFERENCES

1. Arkat, J., Hosseini, L., & Hosseinabadi Farahani, M. (2011). Minimization of exceptional elements and voids in the cell formation problem using a multi-objective genetic algorithm. *Expert Systems with Applications*, 38(8) 9597-9602.
2. Al-Anzi, F.S., & Allahverdi, A. (2006). A hybrid tabu search heuristic for the two-stage assembly scheduling problem, *International Journal of Operation Research*, 3(2) 109–119.
3. Al-Anzi, F.S., & Allahverdi, A. (2007). A self-adaptive differential evolution heuristic for two- stage assembly scheduling problem to minimize maximum lateness with setup times, *European Journal of Operational Research*, 182(1) 80–94.
4. Allahverdi, A., & Al-Anzi, F.S. (2006). A PSO and a tabu search heuristics for assembly scheduling problem of the two-stage distributed database application. *Computer and Operation Research,* 33(4) 1056–1080.
5. Allahverdi, A., & Al-Anzi, F.S. (2009). The two-stage assembly scheduling problem to minimize total completion time with setup times. *Computer Operation Research*, 36(10) 2740–2747.
6. Birgin Ernesto G., & Ronconi Debora P. (2012). Heuristic methods for the single machine scheduling problem with different ready times and a common due date. *Engineering Optimization*, 44(10) 1197-1208.
7. Ceri, S., & Pelagatti, G. (1984). *Distributed Databases: principles and systems*, NewYork: McGraw-Hill.
8. Ehrgott, M. (2005). *Multicriteria optimization,* 2nd Edition. Berlin Heidelberg New York, Springer.
9. Elmasri, R. & Navathe, B. (1999). *Fundamentals of database systems*, 3rd Edition, NewYork: Addison-Wesley.
10. Hariri, A.M.A., & Potts, C.N.(1997).A branch and bound algorithm for the two-stage assembly scheduling problem, *European Journal of Operation Research*, 103(3) 547–556.
11. Haouari, M., & Daouas, T. (1999). Optimal scheduling of the 3-machine assembly-type flow shop, *RAIRO Operation Research*, 33(4) 439–445.
12. Hong-Sen, Y., Xiao-Qinn, W., & Fu-Li, X. (2014). A hybrid electromagnetism-like algorithm for two-stage assembly flow shop scheduling problem. *International Journal of Production Research*, In Press.
13. Keyvanfar, V., Mahdavi, Iraj., & Komaki, G.H.M. (2013). Single machine scheduling with controllable processing times to minimize total tardiness and earliness. *Computers and industrial Engineering*, 65(1) 166-175.
14. Khorshidian, H., Javadian, N., Zandieh, M., Rezaeian, J., & Rahmani, K.(2011). A genetic algorithm for JIT single machine scheduling with preemption and machine idle time. *Expert Systems with Applications*, 38(7) 7911-7918.

15. Lee, C.Y., Cheng, T.C.E., & Lin, B.M.T. (1993).Minimizing the makespan in the 3-machine assembly-type flowshop scheduling problem, *Manage Sci*ence, 39(5) 616–625.

16. Mahnam, M., Moslehi, G., & Fatemi Ghomi, S.M.T. (2013). Single machine scheduling with unequal release dates and idle insert for minimizing the sum of maximum earliness and tardiness. *Mathematical and Computer Modelling*, 63(1) 2549-2563.

17. Mozdgir, A., Fatemi Ghomi, S.M.T., Joli, F., & Navaei, J. (2013). Two-stage assembly flow-shop scheduling problem with non-identical assembly machines considering setup times. *International Journal of Production Research, 51(12)* 3625-3642.

18. Potts, C.N., Sevast'janov, S.V., Strusevich, V.A., Van Wassenhove, L.N., & Zwaneveld, C.M. (1995). The two-stage assembly scheduling problem complexity and approximation, *Operation Research,* 43(2) 346–355.

19. Seidgar, H., Kiani, M., Abedi, M., & Fazlollahtabar, H. (2014).An efficient imperialist competitive algorithm for scheduling two-stage assembly flow shop problem. *International Journal of Production Research*, 52(4) 1240-1256.

20. Torabzadeh, E., & Zandieh, M. (2010).Cloud theory-based simulated annealing approach for scheduling in the two-stage assembly flow shop. *Advances in Engineering Software*, 41(10-11) 1238-1243.

21. Tozkapan, A., Kirca, O., & Chung, C-S.(2003).A branch and bound algorithm to minimize the total weighted flow time for the two-stage assembly scheduling problem, *Computer & Operation Research,* 30(2) 309–320.

22. Sun, X., Morizawa, K., & Nagasawa, H.(2003). Powerful heuristics to minimize makespan in fixed, 3-machine, assembly-type flowshop scheduling, *European Journal of Operation Research*, 146(3) 498–516.

23. Shokrollahpour, E., Zandieh, M., & Dorri, B. (2010). A novel imperialist competitive algorithm for bi-criteria scheduling of the assembly flow shop problem. *International Journal of Production Research*, 49(11) 3087-3103.

24. Xiong, F., & Xing, K. (2014). Meta-heuristics for the distributed two-stage assembly scheduling problem with bi-criteria of makespan and mean completion time. *International Journal of Production Research*, 52(9) 2743-2766.