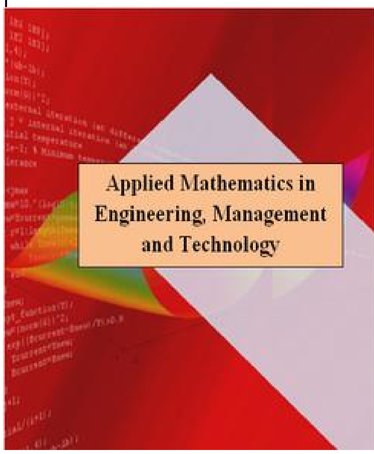
Applied mathematics in Engineering, Management and Technology 1 (3) 2013:92-102
www.amiemt-journal.com

Modeling and solving three stage assembly flow shop scheduling problem with availability machines constraint

Hamid Mahdavi, Mohammad Mohammadi * and Alireza Arshadi khamseh
*Department of Industrial Engineering, Faculty of Engineering, Kharazmi University, Karaj, Iran
Corresponding author email: mohammadi@khu.ac.ir



Abstract:

Assembly flow shop scheduling problem is one of the important problems in industry. In the most research in this case imagine the machines are available all of the time. In this paper we considered the machines are not available and for maintenance are out of available. In the otherwise the most assembly flow shop problem have one object, however in this paper we present a multi-objective problem. The objectives are: Minimize Total weighted flow time and sum of weighted earliness and tardiness. We provide 2 meta-heuristic algorithm, Genetic Algorithm (GA) & Simulated Annealing (SA), for solve.
The results represent for small problem (number of job≤30), SA is better algorithm and for large example, GA has better performance.

Keywords: Assembly Flow shop, Non-dominated Sorting Genetic Algorithm, Simulated Annealing, Multi-objective optimization

1.Introduction

Considering the intense competition in manufacturing environments and quick changes in these environments, researchers have always been trying to enhance the efficiency of manufacturing systems.(Pinedo, 2008) defines the scheduling issue as the issue of allocation of limited resources to current activities in sequence. Therefore, as long as we have not come to the optimal scheduling, it is possible to reduce costs and improve scheduling.
There are a variety of scheduling issues, some of which include: single machine problem, parallel machines, and shop scheduling system. Besides, a variety of performance criteria have been considered including: completion time of last job, minimizing the maximum delay, weighted sum of completion times, etc. (Hatami et al. 2010) Considered a three-stage assembly shop flow scheduling problem with the two purposes of mean time of task completion and maximum delay. They also added sequence-dependent preparation times and the transfer time to their model as two important assumptions to bring the problem closer to reality. Then they offer a mathematical model and a new lower limit for solving this problem. Since finding an optimal solution in feasible time for this complicated manufacturing problem is very difficult, they offered two super-innovative algorithms to solve it: simulated annealing and tabu search. Afterwards, in order to determine the efficiency of the proposed models, some examples, manufactured randomly, were solve by using these two algorithms. (Ibrahimnejad et al. 2010) Investigated the three-stage assembly shop flow scheduling problem with the two purposes of mean time of job completion and maximize delay. By considering sequence-dependent preparation times and transportation time of the parts, they attempted to bring the problem closer to reality. Afterwards, they offered a lower limit for the completion time of all jobs. Then they proceed to deploy the Longo 8 software program and the full counting method in MATLAB program in order to determine the validity of their model. (MalekiDaroonKalayi et al.,2012) studied the three-stage assembly shop flow scheduling problem dependent on sequence and blocking. They considered the weighted total of completion times and the completion time of the last job as objective function and proposed a mathematical model to solve it. Since the problem put forth by them is NP-hard, solving the mathematical model for problems with real proportions is impossible; therefore, they developed an algorithm for simulating annealing. The results of calculations show efficiency of the proposed model. The only research considering limitation was done by (Hadda et al. 2007). They studied the three-stage assembly shop flow scheduling problem by considering the accessibility limitation of the machine and attempted to minimize the maximum completion time of the jobs. They also assumed that the impaired jobs could be continued in a non-additive way. In order to solve this problem, they offered two innovative

algorithms. They showed that SA is the best algorithm for solving the mentioned problem. (Allahverdi& Al-Anzi, 2008) investigated the three-stage assembly shop flow scheduling problem by considering weighted total criteria of the maximum completion time of jobs and the mean completion time of jobs. They employed the weighted total method of two objective functions in their model and proposed three innovative algorithms for solving the problem: simulated annealing algorithm, ant colony algorithm, and differential evolution algorithm.
In the literature, the problem of flow shop scheduling, no study has been conducted to obtain Pareto optimal solutions for a multi-purpose problem.
By investigating and reviewing the previous research done in the area related to the research problem, we come to realize that the problem defined in this paper is different from other investigated problems in two ways. First, in this problem, the completion time dependent on the sequence of tasks, which is a practical assumption in most industries, has been considered. Second, the two objective functions of minimizing the mean delay time and minimizing the mean time of job performance are used by adopting the approach dependent on finding Pareto solutions simultaneously and the approach dependent on blending objectives non-simultaneously.
In this study, a new model is proposed for the assembly flow shop scheduling, and is solved with the two meta-heuristic algorithms and is compared results. Therefore, the principle aim, definition, formulize and solve this new problem.

2.Problem analyses and mathematical model

In this paper, the model under investigation comprises three stages:
The first stage comprises several non-identical parallel machines which perform a job related to them (manufacture).
The second stage is the stage of gathering and transferring parts.
The third stage is the stage of assembling the manufactured parts.

Each job j=1,2…n comprises some of ($\{O_{1,j}\ldots,O_{M,j}\}$, $O_{T,j}$, $O_{A,j}$) operations. The O_{ij}operation must be done on the M_i, j = 1,…m machine and requires $P_{i,j}$ time units. The M_i can perform one job at a same time. Collection and transfer of $O_{T,j}$ is done on the MT machine and lasts $P_{T,j}$ time units. The assembly of $O_{A,j}$ is done on the MA machine and lasts $P_{A,j}$ time units. For each operation of I and k where k=1,…,m and i=1,…,m, operations $O_{i,j}$, $O_{k,j,}$ and j=1,…,n could be done simultaneously. The $O_{T,j}$ operation begins only when all $O_{1,j}$,…, $O_{M,j}$ simultaneous operations are completed. Also, the $O_{A,j}$ operation begins after the completion of $O_{T,j}$. The MT collection and transfer device could collect all parts of a product simultaneously. Likewise, the MA assembly machine could assemble all parts of a product simultaneously.
This study simultaneously follows the two objectives of minimizing the weighted total of the presence of jobs in the system, and the weighted total of the tardiness and the performance of jobs before deadline(earliness).

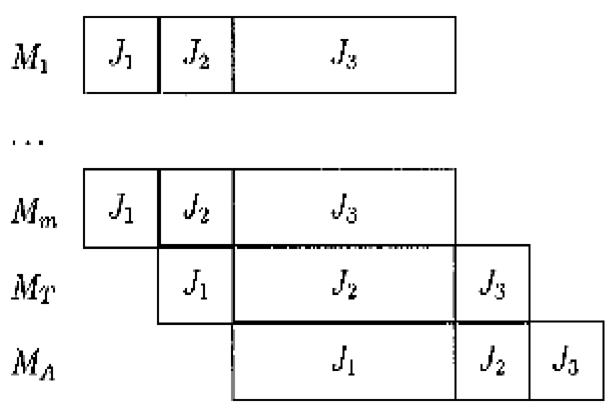


Figure1: figure of manufacturing process

3.Parameter and variable:

I: sequence position index
J,q: job number index
K: machine number index in stage1
$x_{[i]j}$: Equal 1 if assign job j to position i
$S_{k,q,j}$: The prepare time depended to sequence
$S_{0,j,k}$: The first prepare time
$f1_j$: finish time of job j in stage1
$s2_j$: start time of job j in stage2
$f2_j$: finish time of job j in stage2
$s3_j$: start time of job j in stage3
C_j : completed time of job j
$t_{j,k}$: Process time of job j on machine k in stage1
PT_j : Gathering and transferring time of job j
PA_j : Assembly time of job j
$SOH_{l,k}$: start time of l'st over hall on machine k
$FOH_{l,k}$: finish time of l'st over hall on machine k
SD_{iq} : If job q perform after job j immediately
St_{jk} : start time of operation k on job j
Ft_{jk} : finish time of operation k on job j
JST_j : Start time of job j
$u1_{jkl}$: Equal 1 if l'st over hall on machine k started after op.k of job j start
$u2_{jkl}$: Equal 1 if l'st over hall on machine k started before op.k of job j start
$v1_{jkl}$: Equal 1 if l'st over hall on machine k started before op.k of job j finish
$v2_{jkl}$: Equal 1 if l'st over hall on machine k finished after op.k of job j start
u_{jkl} : Equal 1 if l'st over hall on machine k is overlap with op.k of job j

4.Mathematical model

$$\min\{\sum_{j=1}^{n} w1_j \times (C_j), \sum_{j=1}^{n} w2_j \times Tn_j + \sum_{j=1}^{n} w3_j \times En_j\} \quad 1$$

$$\sum_{i=1}^{n} x_{[i],j} = 1 \qquad \forall j \quad 2$$

$$\sum_{j=1}^{n} x_{[i],j} = 1 \qquad \forall i \quad 3$$

$$JST_j \le ST_{j,k} \quad 4$$

$$1 + SD_{jq} \ge x_{[i]j} + x_{[i+1]q} \qquad \forall i, j, q \quad 5$$

$$St_{jk} + M \times (1 - SD_{qj}) \ge Ft_{qk} \quad \forall j, q, k \quad 6$$

$$Ft_{j,k} \ge St_{j,k} + t_{j,k} + \sum_{q=1, q\ne j}^{n} SD_{qj} \times S_{k,q,j} + x_{[1],j} \times S0_{j,k} \quad \forall j, k \quad 7$$

$$St_{jk} + M \times u1_{jkl} \ge SOH_{l,k} \quad \forall j, k, l \quad 8$$

$$SOH_{l,k} + M \times v1_{jkl} \ge St_{j,k} \quad \forall j, k, l \quad 9$$

$$St_{j,k} + M \times v2_{jkl} \ge FOH_{l,k} \quad \forall j, k, l \quad 10$$

$$v1_{j,k,l} + v2_{j,k,l} \le 1 \quad \forall j, k, l \quad 11$$

$$SOH_{l,k} + M \times u2_{j,k,l} \geq St_{j,k} + t_{j,k} + \sum_{q=1,q\neq j}^{n} SD_{qj} \times S_{kqj} + x_{[1],j} \times S0_{j,k} \quad \forall j,k,l \qquad 12$$

$$u_{jkl} + 1 \geq u1_{jkl} + u2_{jkl} \quad \forall j,k,l \qquad 13$$

$$u_{jkl} + 1 \geq 2 - u1_{jkl} + v1_{jkl} \quad \forall j,k,l \qquad 14$$

$$Ft_{jk} + M \times (1 - u_{jkl}) \geq FOH_{l,k} + St_{jk} + t_{j,k} + \sum_{q=1,q\neq j}^{n} SD_{qj} \times S_{k,q,j} + x_{[1],j} \times S0_{j,k} + \alpha \times (SOH_{l,k} - St_{jk}) \qquad 15$$

$$f1_j \geq Ft_{j,k} \qquad 16$$

$$s2_j + M \times (1 - SD_{qj}) \geq f2_q \quad \forall j,q \qquad 17$$

$$s2_j \geq f1_j \quad \forall j \qquad 18$$

$$f2_j \geq s2_j + Pt_j \quad \forall j \qquad 19$$

$$s3_j + M \times (1 - SD_{qj}) \geq C_q \quad \forall j,q \qquad 20$$

$$s3_j \geq f2_j \quad \forall j \qquad 21$$

$$C_j = s3_j + PA_j \quad \forall j \qquad 22$$

$$C_j - D_j = Tn_j - En_j \qquad 23$$

$$x_{[i],j}, u_{jkl}, SD_{jq} \in \{0,1\} \qquad 24$$

In this model number “1” shows the objective function. The limitations “2” and “3” respectively show that each place in the sequence receives only one job, and each job is assigned only to one place. The “4” limitation calculates the start time which is the start time of first operation for each job. The “5” limitation ensures that if task j is assigned the i^{th} place and q is assigned to the $i+1^{th}$ place, then SD_{jq} equals 1; that is, job q is done immediately after job i. The “6’ ensures that in case job j is scheduled after job q, then the start of operation k of task j is after the completion of operation k of job q. the “7” limitation calculates the completion time of operation k of job j if this does not interfere with the planned overhaul process. The “8” limitation makes sure $u1_{jkl}$ equals 1 if the start time of the k^{th} operation of job j is earlier than the start time of the l^{th} planned overhaul, . Likewise, the “9” limitation makes sure $u1_{jkl}$ equals 1 if the start time of the k^{th} operation of job j is later than the start time of the l^{th} planned overhaul. The “10” limitation indicates that if the start time of operation k of job j is earlier than the completion time of the l^{th}overhaul, then $v2_{jkl}$ equals 1. The “11” limitation makes sure that no operation of no job is done in during the planned overhaul. The “12” limitation makes sure u2jkl equals 1 if the l^{th}overhaul on machine k is started before the completion of the k^{th} operation of job j. The “13” indicates that $u1_{jkl}$ and $u2_{jkl}$ equal 1 simultaneously; that is, if the l^{th} overhaul on machine k starts after the start and before the completion of the k^{th}operation of job j, u_{jkl} equals 1, meaning that the l^{th} planned overhaul interferes with operation k of job j. the “14” shows that if the start time of operation k of job j precisely corresponds to the l^{th} planned overhaul, u_{jkl} equals 1. The “15” calculates the completion time of operation k of job j if this job interferes with the overhaul. The “16” calculates the completion time of the first stage of the task – that is, the start time of the transportation stage of job j- which is the latest completion time of the operations of the first stage. The “17” limitation indicates that is job j is scheduled immediately after job q, the start time of the transportation of this job will be after the completion of the transportation operation of job q. The “18” constraint shows that for each job j, the start time is after the completion of the first stage. The “19” limitation calculates completion time of stage2, gathering and transportation. This time equals the start time of the second stage plus the time needed for transportation. The “20” indicates that if job j is scheduled immediately after job q, the start time of the assembly of this job is after the completion of the assembly of job q. The “21” shows that for each job j, the start time of the third stage is after the completion time of the second stage. The “22” limitation calculates the completion time of job j. The “23” calculates tardiness or earliness completion. The “24” limitation indicates that SD_{jq}, u_{jkl}, and $x_{[i],j}$ are binary variable.

5.Proposed algorithms:

In this section we explain 2 algorithms for solution the mathematical problem. These algorithms are: Non-dominated Sorting Genetic Algorithm (NSGA), Multi-objective Simulated Annealing (MSA) Algorithm.

5.1.Non-dominated Sorting Genetic Algorithm (NSGA)

For all non-dominated sorting genetic algorithms, the binary tournament selection operator is used. In this operator, from the current population two chromosomes are randomly selected, and are compared to choose the better one. Selection criteria in NSGA-II algorithm are first the solution's rank and then the cumulative distance related to the solution. Lower solution ranks and longer cumulative distances are more appropriate.
The operator takes the crossover of two parents as the input and combines their characteristics to make an offspring. In the algorithm offered in this study, two operators are considered for the crossover among which the appropriate operator is selected. The two operators are the partially mapped crossover (PMX) and the modified order crossover (OX).
In the PMX operator, two spots on the string related to the parent chromosomes are randomly selected. The values between these two spots are substituted with their corresponding values in the other parent. The OX operator finds a random point on the sting related to the parent. This point divides the parents into the two right and left parts. The first offspring inherits its left side from the left side of the first parent and his right side from the right side of the first parent. However, the sequence of these operations is determined by the second parent. Likewise, the second offspring is produced, too. After operating crossover on the chromosomes, the mutation operator is applied to them. In the algorithm proposed in this article, two mutation operators –substitution and shift- were considered and the appropriate operator of the following sections were determined by the experiment design method. In the substitution mutation, two values of the selected string are substituted with each other. In the shift mutation, two values one the string are randomly selected, the first value replaces the second, and all values between these two values take one move toward the first shift value. In the algorithm proposed by this study, a maximum operation time is considered as the termination condition.

5.2.Multi-objective Simulated Annealing algorithm

The proposed multi-objective simulated annealing algorithm attempts to produce optimum Pareto solutions. In this algorithm, the initial solution is accidentally produced. The produced initial solution is added to the set of optimal Pareto solutions, which is initially a null one, and is considered as the current solution. Two moves – substitution move and shift move -are considered and the appropriate move is determined in the following sections by using experiment design method. In the substitution move, two points on the string related to the solutions are randomly selected and the value related to there are substituted with each other. In the shift move, two points on the string related to the solutions are randomly selected and the first value replaces the second one and the values between the two values take one move toward the first value. In the proposed simulated annealing algorithm, a short-term memory is employed for storing solutions which have recently been considered so that we do not return to the previously investigated solutions.
After obtaining one new solution, the Pareto solutions are updated. In so doing, the new solution is compared with the solutions existing in the Pareto set. If the new solution beats some of the solutions in the set, it is added to the Pareto set.

6.Calculation results

In this section, calculation results for 2 algorithms are proposed. In the first, for each algorithm we selected suitable parameters by design of experiment (DOE) (Taguchi, 1986). After selection parameters, solve some examples by 2 algorithm and compare results. After design of experiment and recognize factors, we coding algorithms by C++ program and analyzing results with MINITAB software.

The parameters obtained such that:

Table1: optimization surface of GA

GA Factors	Proposal surface	Optimization surface for large problems	Optimization surface for medium problems	Optimization surface for small problems
pop_size	{20,40 }	{40}	{40}	{40}
mutation_prob	{0.05,0.1,0.15}	{0.05}	{0.1}	{0.05}
mutation_type	{swap,shift}	{swap }	{swap }	{Shift }
Crossover_type	{PMX,OX}	{OX }	{OX }	{OX }

Table2: optimization surface of SA

SA Factors	Proposal surface	Optimization surface for large problems	Optimization surface for medium problems	Optimization surface for small problems
initial_temp	{100,150,350}	{100}	{150}	{150}
Cooling coefficient(q)	{0.7,0.8,0.9}	{0.9 }	{0.8}	{0.7}
method of manufacturing a neighborhood	{swap,shift}	{shift}	{swap }	{swap }
Iteration in each temp	{50,100,150}	{150}	{50}	{100}
Duringthe bannedlist (TL)	{10,15,20}	{10}	{15}	{10}

7.Results analyses:

In this section we solved some small example by enumeration method and so proposal algorithms, then compared obtained results. The results show the effectiveness of the proposed algorithm. Then solved some medium and large example by GA&SA and compared results.
In this paper, we used 3 measure, where proposed by (Kumar & Singh, 2007), for compare proposal algorithms. These measures are: convergence to pareto optimization answer, the spread between pareto optimization answer and number of pareto solutions.

7.1.Convergence measure:

The convergence measure evaluates the convergence of the obtained non-dominated solution set towards a reference set, i.e., the difference between the obtained solution set and reference set:

$$\mathrm{convergence}(A) = \frac{\sum_{i=1}^{|A^*|} dt_i}{|A^*|}$$

$$dt_i = \min_{j=1}^{|A^*|} \sqrt{\sum_{m=1}^{M} \left[\frac{f_k(i) - f_k(j)}{f_m^{max} - f_m^{min}}\right]^2}$$

Here, f_m^{max} و f_m^{min} are the minimum and maximum function values, respectively, of k'th objective function in P∗. For this measure too, lower value indicates superiority of solution set and an ideal value 0 indicates that the obtained solution set has got all the solutions in the reference set.

7.2.Spread measure:

The spread measure computes distribution of the solutions in the obtained non-dominated solution set by calculating a relative distance between consecutive solutions:

$$\mathrm{spread}(A) = \frac{\sum_{m=1}^{M} d_m^e + \sum_{i=1}^{|A|} |d_i - \bar{d}|}{\sum_{m=1}^{M} d_m^e + |A|\bar{d}}$$

Here, M is the number of objectives, d_i is the distance, and the parameter d_m^e is the distance between the extreme solutions of P and A corresponding to m'th objective function. The value of this measure is zero for an ideal distribution (when obtained solution set consists of extreme solutions and the distribution of intermediate solutions are uniform) but it can be more than 1 as well for a bad distribution (when solutions get more and more closer from the ideal distribution) of solutions in the set.

7.3.Non-dominated solutions ratio measure:

This measure compares two sets directly and indicates coverage of one set over another:

$$R_{nds}(A_j) = \frac{|A_j - \{(x \in A_j | \exists y \in A : y \prec x\}|}{|A_j|}$$

In this formula $y \prec x$indicated the x solution dominated by y solution. For this measure too, large value of the measure indicates superiority of the solution set.

7.4.After solution and compare, the results obtained this:

Table3: Calculation results for small examples

Example specification		Solution algorithm								
		enumeration			NSGA II			MSA		
Jobs number	Machines number	convergence	spread	ratio	convergence	spread	ratio	convergence	spread	ratio
5	3	0	0.452	1	0	0.452	1	0	0.452	1
7	3	0	0.417	1	0	0.417	1	0	0.417	1
7	4	0	0.496	1	0	0.496	1	0	0.496	1
10	3	0	0.531	1	0	0.531	1	0	0.531	1
10	4	0	0.441	1	0	0.441	1	0	0.493	1
10	5	0	0.525	1	0	0.525	1	0	0.525	1

It is clear from the above results for small problems GA&SA has similar performance and both algorithms are based strictly on enumeration solutions. The following figure illustrates this is:

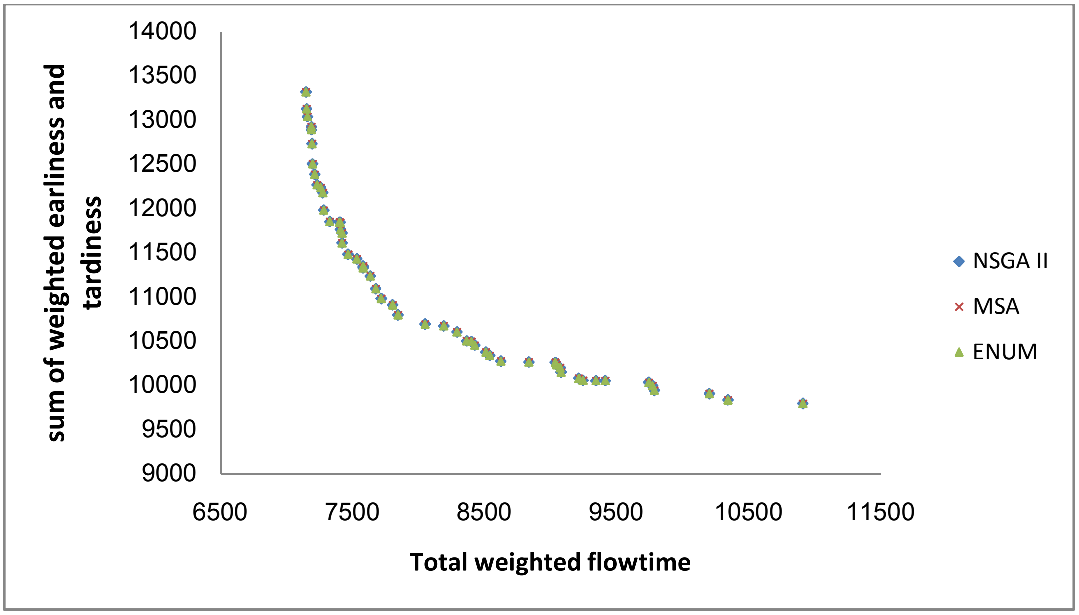


Figure2: Obtained Pareto-front by algorithms

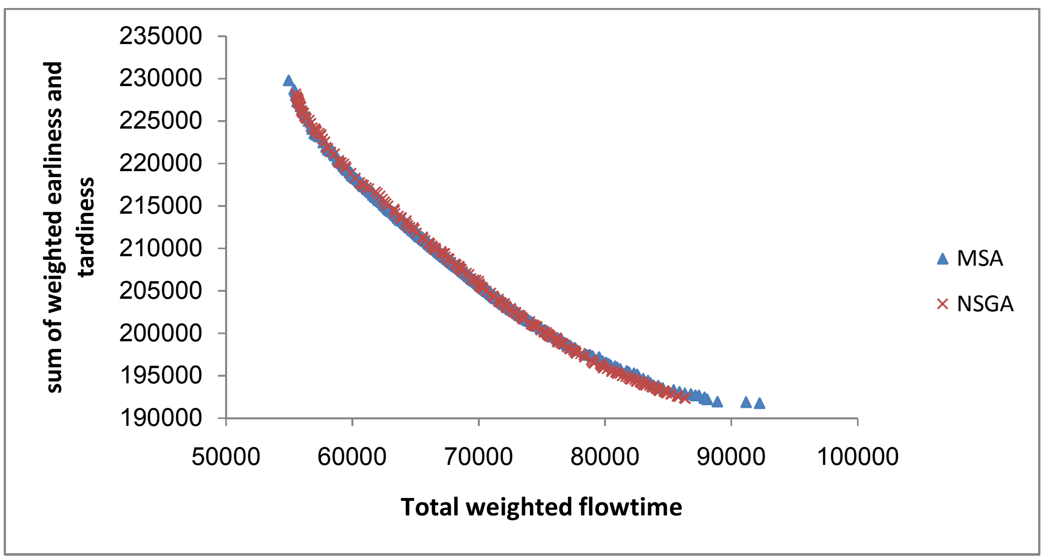


Figure3: Obtained Pareto-front by MSA&NSGA

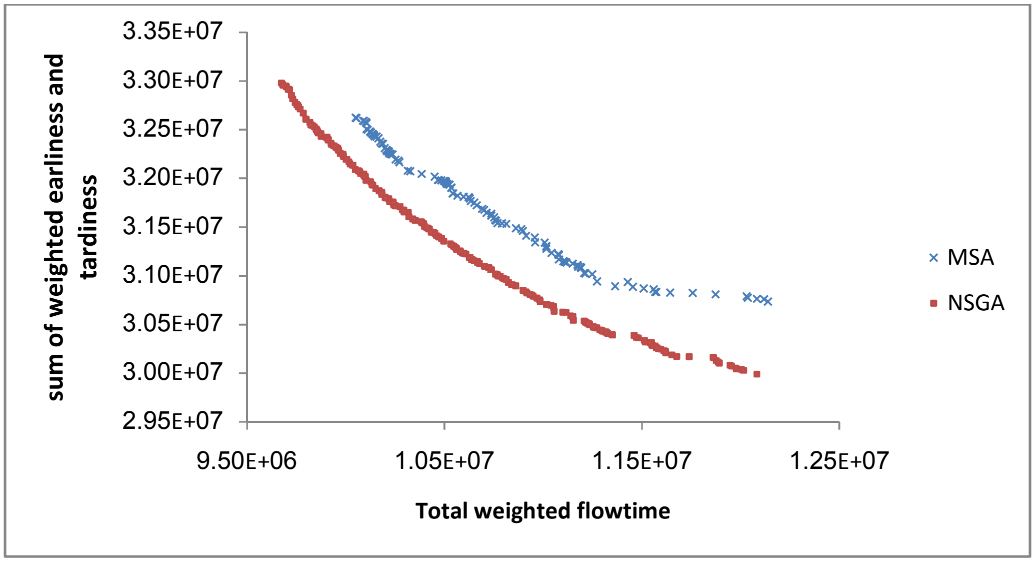


Figure4: Obtained Pareto-front by MSA&NSGA

As can be inferred from Fig, in most cases, simulation algorithm solutions dominated genetic algorithm solutions.

Table4: Calculation results for medium examples

Example specification		Solution algorithm					
		NSGA II			MSA		
Jobs number	Machines number	Convergence	spread	ratio	convergence	spread	ratio
20	5	0.00	0.90	0.22	0.00	0.95	0.78
20	10	0.00	0.50	0.28	0.00	0.62	0.84
25	5	0.00	0.43	0.24	0.00	0.72	0.85
25	10	0.00	0.88	0.70	0.00	0.90	0.29
25	15	0.00	0.48	0.33	0.00	0.63	0.67
30	10	0.00	0.88	0.46	0.00	0.94	0.54
30	15	0.00	0.90	0.66	0.00	0.94	0.34
30	20	0.00	0.73	0.97	0.02	0.79	0.03

Table5: Calculation results for large examples

		Solution algorithm					
Example specification		NSGA II			MSA		
Jobs number	Machines number	Convergence	spread	ratio	convergence	spread	ratio
100	20	0.00	0.63	0.96	0.10	0.78	0.04
100	30	0.00	0.70	1	0.16	0.83	0
100	40	0.00	0.71	1	0.11	0.78	0
120	20	0.00	0.68	1	0.17	0.81	0
120	30	0.00	0.60	1	0.20	0.80	0
120	40	0.00	0.70	1	0.18	0.86	0
140	30	0.00	0.80	1	0.12	0.74	0
140	40	0.00	0.77	1	0.20	0.85	0
140	50	0.00	0.96	1	0.01	0.97	0

As is clear from, the results obtained by genetic algorithm have been defeated simulated Annealing algorithm answers completely. According to the results, for major example, the advantage of genetic algorithm, in all cases, it is recommended.

8.Conclusions:

In this paper the assembly flow shop with limited machine availability was studied. Also assume that there are access restrictions for machines in first step and to solve this problem, a mathematical programming model is presented. The objectives intended for this model are: Minimize Total weighted flow time and sum of weighted earliness and tardiness. Since the problem is NP-hard, two meta heuristic algorithms to solve this problem was presented. These two algorithms are non-dominated sorting genetic algorithm (NSGA II) and Simulated Annealing Algorithm. parameters for two algorithms were selected by design of experiment. Computational results show that for small problems, the number of jobs is less than 30, the simulation algorithm has a better performance and for large example genetic algorithm has better performance.
Proposed other meta-heuristic algorithms such as ant colony algorithm and Mass particle optimization algorithm, also consider other goals, such as the completion time of last job, as guidelines for future research are suggested.

Resource:

Aggoune, R., 2004. Minimizing the makespan for the flow shop scheduling problem with availability constraints.. European Journal of Operational Research, 153, p.534–543.
Aggoune, R. & Portmann, M.-C., 2006. Flow shop scheduling problem with limited machine availability: A heuristic approach. International Journal of Production Economics, 99, p.4–15.
Al-Anzi, F.S. & Allahverdi, A., 2006. A Hybrid Tabu Search Heuristic for the Two-Stage Assembly Scheduling Problem. International Journal of Operations Research, 3, pp.109-19.
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 , p. 80–94.
Allahverdi, A. & Al-Anzi, F.S., 2008. The two-stage assembly flowshop scheduling problem with bicriteria of makespan and mean completion time. Int J Adv Manuf Technol, 37, p.166–177.
Andrés, C. & Hatami, S., 2011. The three stage assembly permutation flowshop scheduling problem. In 5th International Conference on Industrial Engineering and Industrial Management., 2011.
Azadeh, A., Jeihoonian, M., Maleki Shoja, B. & Seyedmahmoudi, S.H., 2012. An integrated neural network–simulation algorithm for performance optimisation of the bi-criteria two-stage assembly flow-shop scheduling problem with stochastic activities. International Journal of Production Research, p.to appear.
Hadda, H., Dridi, N. & Hajri-Gabouj, S., 2007. The Two-Stage Assembly Flow Shop Scheduling with an Availability Constraint. In multidisciplinary international scheduling conference theory and applications (MISTA). Paris, 2007.

Hadda, H., Dridi, N. & Hajri-Gabouj, S., 2010. An improved heuristic for two-machine flow shop scheduling with an availability constraint. 4OR-Q J Oper Res, 8, pp.87-99.
Hatami, S., Ebrahimnejad, S., Tavakkoli-Moghaddam, R. & Maboudian, Y., 2010. Two meta-heuristics for three-stage assembly flowshop scheduling with sequence-dependent setup times. Int J Adv Manuf Technol, 50, p.1153–1164.
R. Kumar and P.K. Singh: Pareto Evolutionary Algorithm Hybridized with Local Search for Bi-objective TSP, Studies in Computational Intelligence (SCI) 75, 361–398 (2007).
Kirkpatrick, S., Gelatt, C.D. & Vecchi, M.P., 1983. Optimization by simulated annealing. Science, 220, p.671–680.
Lee, C.Y., 1996. Machine scheduling with an availability constraints. Journal of Global Optimization, 9, p.363–382.
Lee, C.Y., 1997. Minimizing the makespan in the two-machine flowshop scheduling problem with an availability constraint. Operations Research Letters, 20, p. 129–139.
Lee, C.Y., 1999. Two-machine flowshop scheduling with availability constraints. European Journal of Operational Research, 114, p. 420–429.
McKay, K., Safayeni, F. & Buzacott, J., 1988. Job-Shop Scheduling Theory: what is relevant? Interfaces, 18(4), pp. 84-90.
N. Metropolis, N. et al., 1953. Equation of state calculations by fast computing machines. Journal of Chemical Physics, 21, p.1087–1092.
Pinedo, M.L., 2008. Scheduling: theory, algorithms, and systems. 3rd ed. Springer.
Potts, C. et al., 1995. The two-stage assembly scheduling problem: complexity and approximation. Operations Research, 43, pp.346-55.
Taguchi, G., 1986. Introduction to quality engineering. White Plains: Asian Productivity Organization/UNIPUB.
Torabzadeh, E. & Zandieh, M., 2010. Cloud theory-based simulated annealing approach for scheduling in the two-stage assembly flowshop. Advances in Engineering Software, 41 , p.1238–1243.