A Performance Analysis of Memetic Algorithm, Genetic Algorithm and Simulated Annealing in Production System Optimization

Alireza Noroziroshan, Shaghayegh Habibi

Abstract:** Researchers laid the foundation of evolutionary algorithms in the late 60s and since then, heuristic algorithms have been widely applied to several complex scheduling and sequencing problems during the recent studies. In this paper, memetic algorithm (MA), genetic algorithm (GA) and simulated annealing (SA) are applied to a complex sequencing problem. The problem under study concerns about sequencing problem in mixed-shop floor environment. The main objective is to minimize the overall make-span of multiple mixed-model assembly lines by finding the best job sequence and allocation. The superiority of MA's performance is proved by evaluating standard deviation, optimal solution and mean value of obtained solutions.

***Keywords—**Genetic Algorithm, Make-span, Memetic Algorithm, Simulated Annealing.*

I. INTRODUCTION

In the current business environment, time is the cutting-edge competitive advantage which makes companies to treat it as equivalent as money, innovation and productivity. By managing the time in production, companies are able to introduce their innovations to market as soon as possible and so cover more market segments. In particular, time-based industries have proven to be about two-fold more efficient than conventional and traditional systems. Flexible assembly lines provide special benefits for production and assembly industries by providing more degrees of flexibility in producing different types of product. In production environment, flexible production systems result in cutting down on overall costs up to 20% [24].They have played a considerable role to the development of united states industries in twentieth century [1]. They make industries more efficient to adjust the production requirements to the possible demand changes [2]. The problem under study focuses on shop-floor manufacturing that produce different product models. The shop floor consists of manual assembly stations and each workstation is able to assemble different product models. In such production systems, minimizing the operation cycle time is highly concerned by researchers as it directly addresses the effectiveness of any production activities. The longest completion time *(max {C_1... Cm})* for a set of jobs, when all jobs leave the system is called make-span [3]. Different jobs should be allocated to assembly lines and the assembly operation is performed at flexible workstations as they are moving along the assembly line. In this case, the allocated jobs are manufactured on the same line as they move along the assembly lines. Meanwhile the problem falls into flow shop sequencing problems so a decision for finding the optimum sequence of jobs should be made to increase the efficiency of lines by minimizing overall make-span [4]. In such problems, set of feasible solution is discrete and they can be characterized by a finite number of feasible solutions. Due to massive required computations, there is a tendency to use heuristics rather than exact methods to solve the problem in a reasonable amount of time [5] .

II. LITERATURE REVIEW

With the emergence of meta-heuristic algorithms, several algorithms have been applied to overcome the complexity of sequencing problems in assembly lines problems. Evolutionary computing is a research area within computer science that used for solving combinatorial optimization and complex problems. Though, there is no way to prove that the solution obtained by evolutionary algorithms(EA) are global or local optimum, usually EA reach to good solutions in the reasonable amount of time [6]. Memetic algorithm developed in the late 80th which aims to incorporate the families of meta-heuristic algorithms to take advantage of each method [22]. MA is functioning based on the principle of individual improvement plus population cooperation [23].Sequence-dependent setup times become one of the most favored assumptions in many researchers in real scheduling problems [7]. Reference [8] attempted to develop new Immune Algorithm (AI) approach for scheduling of a hybrid flow shop in which there are sequence dependent setup times, commonly known as the SDST hybrid flow shops and by using immune algorithm approach. Reference [7] proposed meta-heuristic algorithm based on simulated annealing to solve hybrid flow shop. They consider sequence setup time and transportation time in their problem. The proposed algorithm was provided some means of intensification and diversification to increase the efficiency of algorithm. An extensive comparison was done to obtain precious calibration of simulated annealing by applying Taguchi method. A comparison between the performances of the algorithms illustrated the superiority of the proposed simulated annealing. Reference [9] concerned about optimizing the job sequencing problem in mixed-model assembly line with limited intermediate buffers. Three objectives concurrently considered in this problem: minimizing the total setup, total production rate variation and the total assembly cost. Since the problem classified into NP-hard, a hybrid based genetic algorithm and simulated annealing was developed to cope with

Revised Version Manuscript Received on July 13, 2015.

Alireza Noroziroshan, Department of Industrial, Manufacturing & System Engineering, The University of Texas at Arlington, UTA Box 19017, Arlington, Texas 76019, USA.

Shaghayegh Habibi, Faculty of Entrepreneurship, The University of Tehran, Tehran, Iran.

problem complexity. The proposed algorithm examined on different size of sequencing problems with different number of machine and different production plan. The MA algorithm illustrates better performance compare to simple genetic algorithm as it converges to higher quality of solution in smaller number of generation. Reference [10] research focused on mixed-model assembly line-balancing problem with sequencing approach in which most of the assembly operations were performed manually. The jobs are cyclically sent to the assembly line based on a sequence. Precedence relations among different operations of different products are also considered. Several experiments carried out and the results were discussed in detail. Reference [11] presented simulated annealing based heuristic for job sequencing problem in mixed-model assembly line just in time environment. Two specific sequencing objectives were considered in their research, number of required setups and usage rate. In the proposed simulated annealing, an initial solution is elected from randomly generated population of 10000 solutions which has the best composite percentile rank for both number of usage rate and required setups. Through the entire search, algorithm is guided by the temperature level, cooling rate and the acceptability of new solutions. The efficiency of proposed simulated annealing heuristic examined by several test problems and the superiority of proposed algorithm is proved through comparing the results to the Tabu Search approach. It's concluded that simulated annealing reach to the near-optimal solution in small size problems.

III. PROBLEM DEFINITION

The problem is to organize the execution on N jobs on L mixed-model assembly lines (L_1, L2,...,L_l). All workstations in each line are able process each job. Each assembly station is capable of serving any task of any model. In order to serve new product models, assembly lines should be set up for new material requirement. Initial setup time is required for the first job of sequence in each line. Change over time is necessary to change the settings from one job to another in the same line. The assembly operation can be performed independently in all assembly lines. Jobs are not allowed to switch to other lines once assigned. All the parameters are fixed and all times are deterministic. Due to massive permutations for job allocation problem, three meta-heuristic algorithms are applied to find the near optimal solutions. The solutions obtained are compared in terms of algorithm's capability for probing the potential solution space. It should be noted that the total permutations for this problem is obtained by summing the permutation for all possible configurations of job allocation. Thus, solving such problems by classical mathematical techniques cannot be solved within a reasonable amount of time.

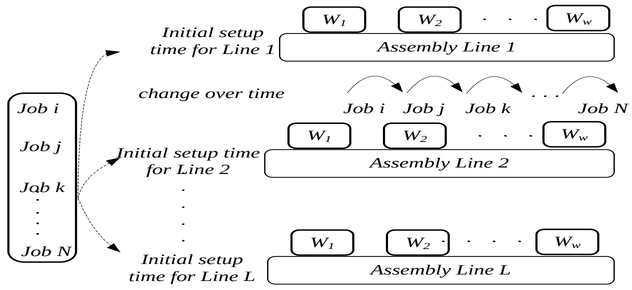

Fig. 1. Model diagram of problem under study

Meanwhile, meta-heuristic methods are applied to solve the problem and find the solutions in efficient way. The best configuration of job allocation is mainly affectedby jobs' process time so all configurations should be checked. The total configurations of job allocation are obtained through solving (1).

$$y_1 + y_2 + \dots + y_l = n \qquad (1)$$

$$y_1 \geq y_2 \geq \cdots \geq y_l\,,\ y_l \in Integer$$

Where $y_l$represents the number of jobs assigns to the lth assembly line and n shows the total number of jobs in the system.

IV. MATERIAL AND METHODS

Evaluating the performance of MA, GA and SA is the main objective of this research through challenging algorithm for finding minimum make-span of multiple mixed-model assembly lines. Also, the paper attempts to check the steadiness and reliability of the heuristic methods in finding the optimal solutions in five runs on each problem. For this means, three meta-heuristic algorithms are applied and the performance of each is evaluated.

A. Simulated Annealing

Simulated annealing algorithm is known as Mont Carlo annealing works based on Monte Carlo approach that could be used for simulating the behavior of a set of atoms which is taken from thermodynamic. Simulated annealing is able to deal with noisy search space. The thermal equilibrium at given temperature is obtain through applying small random perturbation to the atomic structure. If this perturbation results in the lower energy sate, the algorithm is repeated by using new energy state. But if higher energy state is achieved, the new state is accepted with certain probability which is depends on the history of the search [12]. The uphill probability varies during the annealing process. The initial temperature should be high enough to provide appropriate degree of exploration towards its "freezing point" as the search progresses. Neighborhood search generates(NSS): in order to probe the solution space, a set of allowable moves are required. These moves are visiting form solutions to solutions as the annealing proceeds. The applied NSS is SWAP operator in which randomly exchange position of two elements. Meanwhile, two jobs are randomly swapped by generating two random keys [7]. In order to avoid algorithm to stick in a local optimum, worse moves might be accepted based on current temperature. The temperature is decreasing to minimize the probability of accepting non-improving moves[13]. The exponential cooling scheduling is applied as cooling schedule for the SA because of the ability to compromise between fast schedule and also the ability to reach lower energy state [14]. Initial temperature must be high enough to provide opportunity for all potential solutions in the search space to be visited .it also should not be too high to do a lot of unnecessary searches which might increase the algorithm's process time [15].The exponential cooling schedule is given by $T_k = \gamma * T_{k-1}$ where $\gamma \epsilon\ (0,1)$ is temperature decrease rate. SA algorithm will stop when the current temperature reaches to 1. Meanwhile total number of iteration can be calculated as follows:

$$T_o(1-\gamma)^n \leq 1, \quad n \leq \frac{Ln(\frac{1}{T_o})}{Ln(1-\gamma)} \tag{2}$$

In order to check the performance of algorithms, total number of iterations for both GA and SA should be the same. The initial temperature is set to 10 with 0.01 cooling rate. According to (2) it makes 340 iterations for simulated annealing. The same numbers of iterations are set for GA to provide equal condition for performance comparison. The pseudo code for SA is shown in Fig. 2.

```
Set initial temperature T0, cooling rate δ
Generate initial solution x0
Tcur = T0
δ = ε * T0
Count =1 to generate new solution xcount.
If Z (xcount) < Z (x0)    then   x0 = xcount
  If Z (xcount) > Z (x0)  then
    ΔC = Z (x0) − Z (xcount)
    Uphill = e^(ΔC/Tcur)
    Rnd =Generate uniform random number
       If pup hill > Rnd then
        x0 = xcount
       End
  End
   Tcur = Tcur − δ
   If Tcur > Stopping temprature then
   Count = count + 1
   Else
   Print {xcur, Z (xcur)}
   End
 End
```

Fig. 2. Simulated annealing

B. Genetic Algorithm

The nature ability to learn and adapt as the generation proceed is represented by Holland who focused on the natural genetic selection mechanism and adaptive processes of natural systems [16]. New individuals are produced by means of selection, cross over and mutation operators. Good attributes are transmitted from parent to offspring, so the average quality of solutions improving from generation to generation. The algorithm supposes to stop when some predetermined criteria are met [17]. The population in GA is fixed during all generations and set to 30 in each generation. Total number of generation is used as a stopping criterion for both GA and MA and programs terminates at 340 generations. Tournament selection is a mechanism of running tournament among population to choose few individuals. Tournament aims to imitate natural competition of specious [18]. Two individuals are selected from mating pool. The individual with the highest fitness value is selected as the winner of the tournament. Selection continues by selecting a new tournament group until the required number of individuals is collected. Finally the winner of each competition is copied to worst chromosomes [6]. Crossover is considered as the most important genetic operator. It aims to combine two different chromosomes and generate new offspring which captures both parents information. Partially Mapped Crossover (PMX) is employed as crossover operator in GA. Crossover rate is set to 80%which is able to find good solution in a reasonable amount of time [19].Mutation operator prevents algorithm from rapid convergence or premature convergence by perturbing solution and preserving diversity of generated solution. Swap mutation is selected as mutation operator and mutation probability is set to 0.02 which is a typical value for genetic algorithm [20].The pseudo code for GA is presented in Fig. 3.

```
Initialize pm, pc, Pi ∈ [0,1],i=1, maxGen
Generate population P0
Evaluate P0 and find the best solution π
πElite = π*
 While Not stop criterion do
 For j:=1 to PopSize/2 do
  Select two parents p1 and p2 from Pi-1
  With probability pc, crossover (P1, P2)
  With probability pm, mutate offspring
  Evaluate off spring and add it to Pi
End
Add Pi-1 to Pi
Sort Pi
Keep the PopSize best solution in Pi
Find the best solution π* in Pi
  If π* is better than πElite then
  πElite = π*
  End
Update(i)
End
```

Fig. 3. Genetic algorithm

C. Memetic Algorithm

Genetic algorithm and simulated annealing are able to deal with complexity of large problems in finding near optimal solutions. The evolution process for SA is performed based on iterative changes of the current solution's neighborhood, while GA handles a population of possible solutions. The evolution process is executing through transferring useful information during generations [14]. The hybrid approach combines the advantage of both GA and SA for global and local search respectively[21]. A memetic algorithm is applied to solve the sequencing problem and the results are compared with SA and GA to evaluate the performance of hybrid method.

```
Initialize pm, pc, Pi ∈ [0, 1], i=1
Initial temperature T0, cooling rate δ maxGen
Generate population P0
Evaluate P0 and find the best solution π
πElite = π*
While Not stop criterion do
  For j:=1to PopSize/2 do
    Select two parents p1 and p2 from Pi-1
    Perform crossover with probability of Pc
    Offspring= Crossover (P1,P2)
    With probability pm, mutate offspring
    Improve offspring by using SA
    Evaluate off spring and add it to Pi
  End
  Add Pi-1 to Pi
  Sort Pi
  Keep the PopSize  best solution in Pi
  Find the best solution π* in Pi
  If π* is better than πElite then
    πElite = π*
  End
Update(i)
End
```

Fig. 4. Memetic Algorithm

As presented in MA algorithm pseudo code, it starts by simple genetic algorithm and at each iteration; simulated annealing is applying to improve the quality of solution. The best created solution by GA is elected to be improved by SA algorithm. Two termination conditions are designated for SA.

1) New off-springs improve the fitness function
2) Simulated annealing reaches to freezing point

The pseudo code for MA is shown in Fig.4.

D. Fitness Function

In order to find the minimum make-span, each configuration of job allocation is considered as a new problem that algorithms attempts to find the best job sequence and allocation. After exploring all configurations of job allocation, all the obtained solutions from different configurations of job allocation are compared and minimum value of objective function. By this means, the quality of obtained solutions by all three algorithms are compared to evaluate the algorithm performance. In order to minimize the overall make-span of assembly lines, a cost function is developed to compute the completion time of each line and the longest completion time is selected as overall make-span. Meanwhile minimizing the longest completion time is seeking by proposed function.Fig.5 presents a chromosome of jobs. In this case different configurations of job allocation also exist in job allocation problem which only one of them gives us the minimum make-span time of considered problem.

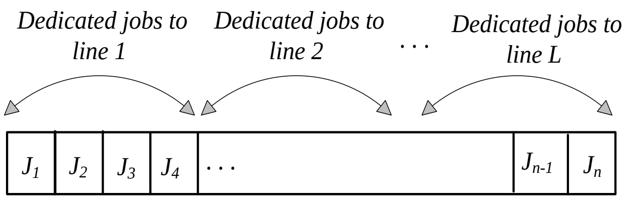

Fig. 5. The sequence of allocated job to every single line

The cost function for is given by:

$$E = \frac{1}{(\mathrm{Max}\,\{\mathrm{Cmax}_l\})};\ l = 1, \dots, L \qquad (3)$$

Where

$$\mathrm{Cmax}_l = (S_i) + (C_{ij}) + (F_l) \qquad (4)$$

S_i :Initial setup time

C_{ij} : Change over time from job i to j

F_l : Flow time line on line L

V. NUMERICAL EXAMPLE

Algorithms are coded in MATLAB 12.0 and run on a PC with 1.7 GHz Intel Core 2 Duo and 2 GB of RAM memory. Three assembly lines are selected to process 13 jobs and each line equipped with two workstations. Fourteen different configurations of job allocation is available in which each are consider as a new sequencing problem.Table I contains required process time for every single job at each workstations.

TABLE I. PROCESS TIME AT WORKSTATIONS

Job	Work load at W1	Workload at W 2
1	64.91	64.23
2	36.55	36.55
3	124.56	124.57
4	30.45	30.47
5	124.14	122.86
6	136.5	138.6
7	132.3	130.2
8	67.15	67.15
9	43.34	44.16
10	189.95	187.76
11	103.87	105.13
12	103.7	104.72
13	103.95	102.6

Initial setup time and change over times are illustrated in Table II. Each algorithm is run for 5 times (S_i) and the minimum value of objective function, average and standard deviation (STDEV) are computed for each problem. Number of jobs assign to each line is specified by L_i (Second, third and fourth column)and algorithms attempts to find which jobs are assigned to which line in what sequence. Column five to nine illustrates make-span achieved from five runs. The tenth column refers to the best make-span achieved by algorithms from five runs of SA. Eleventh and twelfth columns symbolize the mean and standard deviation of five runs results. The problems are solved by SA, GA and MA and results have been indicated in Table III, Table IV and Table V respectively. STDEV is considered as a powerful criterion which makes stronger ground for evaluating the steadiness and accuracy of algorithms in reaching near optimal solution in facing with different problems.

TABLE II. INITIAL SETUP TIME & CHANGE OVER TIME

J/J	1	2	3	4	5	6	7	8	9	10	11	12	13	Initial setup time
1	0	10	9	8	10	11	12	11	9	7	13	5	14	25
2	10	0	12	16	17	8	6	15	13	7	10	9	16	30
3	9	12	0	19	7	16	12	14	13	18	19	20	12	32
4	8	16	19	0	13	18	11	8	19	16	11	7	5	22
5	10	17	7	13	0	17	15	20	12	19	13	16	8	35
6	11	8	16	18	17	0	9	10	8	6	10	11	17	33
7	12	6	12	11	15	9	0	6	15	13	12	10	19	35
8	11	15	14	8	20	10	6	0	5	16	11	18	10	39
9	9	13	13	19	12	8	15	5	0	14	14	5	7	33
10	7	7	18	16	19	6	13	16	14	0	11	13	9	29
11	13	10	19	11	13	10	12	11	14	11	0	6	14	37
12	5	9	20	7	16	11	10	18	5	13	6	0	6	28
13	14	16	12	5	8	17	19	10	7	9	14	6	0	36

TABLE III. RESULTS OBTAINED BY SIMULATED ANNEALING

Problem	L1		L2	L3	S1	S2	S3	S4	S5	MIN	AVG	STDEV
1	11		1	1	1158.40	1161.80	1158.10	1154.40	1153.30	1153.30	1157.20	3.40
2	10		2	1	1015.90	1016.40	1024.7	1018.40	1020.40	1015.90	1019.16	3.57
3	9		2	2	886.34	882.92	884.92	888.57	884.92	882.92	885.534	2.08
4	9		3	1	884.34	888.92	887.24	890.02	894.23	884.34	888.95	3.64
5	8		3	2	732.22	730.39	733.04	730.39	734.22	730.39	732.052	1.67
6	8		4	1	738.86	730.39	730.39	737.86	730.90	730.39	733.68	4.29
7	7		3	3	649.17	649.17	649.17	649.17	649.17	649.17	649.17	0
8	7		4	2	659.68	656.07	657.99	657.99	665.57	656.07	659.46	3.64
9	7		5	1	741.44	747.80	742.22	751.23	737.46	737.46	744.03	5.45
10	6		4	3	619.22	619.22	630.28	622.30	634.01	619.22	625.006	6.76
11	6		5	2	651.37	656.39	661.38	660.97	651.37	651.37	656.296	4.90
12	6		6	1	730.24	733.29	739.09	737.28	736.99	730.24	735.378	3.56
13	5		4	4	631.08	629.93	629.93	630.84	626.69	626.69	629.694	1.75
14	5		5	3	624.18	624.12	622.12	627.67	628.46	622.12	625.31	2.66
AVG										763.54	767.20	3.38
MIN										619.22		

TABLE IV. RESULTS OBTAINED BY GENETIC ALGORITHM`

Problem	L1		L2	L3	S1	S2	S3	S4	S5	MIN	AVG	STDEV
1	11		1	1	1157.1	1153.3	1153.3	1153.3	1157.1	1153.3	1154.82	2.08
2	10		2	1	1023.8	1018.5	1015.7	1020.2	1025.6	1015.7	1020.76	3.99
3	9		2	2	883.92	884.92	987.01	883.57	884.92	883.57	904.868	45.92
4	9		3	1	883.57	882.92	882.92	883.52	884.92	882.92	883.57	0.81
5	8		3	2	734.86	731.22	730.39	732.22	730.39	730.39	731.816	1.86
6	8		4	1	733.54	733.54	733.9	739.86	736.37	733.54	735.442	2.73
7	7		3	3	649.17	666.44	658.75	656.07	649.17	649.17	655.92	7.24
8	7		4	2	661.24	677.41	663.63	671.67	657.99	657.99	666.388	7.96
9	7		5	1	750.03	735.32	736.65	790	737.31	735.32	749.862	23.20
10	6		4	3	628.46	630.28	633.56	626.06	621.08	621.08	627.888	4.68
11	6		5	2	664.85	653.72	657.37	661.47	651.37	651.37	657.756	5.50

12	6		6	1	734.24	741.87	742.67	733.7	736.51	733.7	737.798	4.22
13	5		4	4	627.42	641.62	629.93	638.01	630.15	627.42	633.426	6.06
14	5		5	3	621.12	630.19	628.46	637.9	630.04	621.12	629.542	5.97
AVG										764.04	770.704	8.738
MIN										621.08		

TABLE V. RESULTS OBTAINED BY MA

Problem	L1		L2	L3	S1	S2	S3	S4	S5	MIN	AVG	STDEV
1	11		1	1	1153.5	1153.5	1153.3	1153.3	1153.3	1153.3	1153.38	0.1095
2	10		2	1	1014.6	1016.6	1014.9	1014.6	1016.4	1014.6	1015.42	0.9959
3	9		2	2	883.57	883.52	883.92	883.57	883.57	883.52	883.63	0.1635
4	9		3	1	883.92	883.92	882.92	884.92	884.92	882.92	884.12	0.8366
5	8		3	2	733.54	733.54	732.22	730.39	730.39	730.39	732.016	1.5791
6	8		4	1	731.22	730.39	732.22	733.5	731.9	730.39	731.846	1.1608
7	7		3	3	649.17	649.17	649.17	657.99	649.17	649.17	650.934	3.9444
8	7		4	2	656.07	660.16	657.99	657.99	656.07	656.07	657.656	1.6973
9	7		5	1	732.78	737.31	732.78	732.29	732.29	732.29	733.49	2.1494
10	6		4	3	620.09	619.22	623.06	619.08	619.22	619.08	620.134	1.6841
11	6		5	2	651.37	656.06	652.26	660.78	652.2	651.37	654.534	3.9360
12	6		6	1	729.01	742.78	734	730.02	730.02	729.01	733.166	5.7050
13	5		4	4	636.5	628.26	632.77	628.91	629.94	628.26	631.276	3.3912
14	5		5	3	621.95	630.28	630.28	633.17	633.96	621.95	629.928	4.7608
AVG										763.0	765.10	2.29
MIN										619.08		

TABLE VI. STATISTICAL COMPARISON BETWEEN SA, GA AND MA

		MIN				MEAN			STDEV	
Problem	SA	GA	MA		SA	GA	MA	SA	GA	MA
1	1153.30	1153.30	1153.30		1157.20	1154.82	1153.38	3.40	2.08	0.10
2	1015.90	1015.70	1014.60		1019.16	1020.76	1015.42	3.57	3.99	0.99
3	882.92	883.57	883.52		885.53	904.868	883.63	2.08	45.92	0.16
4	884.34	882.92	882.92		888.95	883.57	884.12	3.64	0.817	0.83
5	730.39	730.39	730.39		732.052	731.816	732.01	1.67	1.86	1.57
6	730.39	733.54	730.39		733.68	735.442	731.84	4.29	2.74	1.16
7	649.17	649.17	649.17		649.17	655.92	650.93	0	7.24	3.94
8	656.07	657.99	656.07		659.46	666.388	657.65	3.64	7.96	1.69
9	737.46	735.32	732.29		744.03	749.862	733.49	5.45	23.20	2.14
10	619.22	621.08	619.08		625.006	627.888	620.13	6.76	4.68	1.68
11	651.37	651.37	651.37		656.296	657.756	654.53	4.90	5.50	3.93
12	730.24	733.70	729.01		735.378	737.798	733.16	3.56	4.22	5.70
13	626.69	627.42	628.26		629.694	633.426	631.27	1.76	6.06	3.39
14	622.12	621.12	621.95		625.31	629.542	629.93	2.6	5.97	4.76
Percentage	57%	42%	78%		21%	14%	65%	28%	7%	64%

TABLE VI represents statistical comparison between three evolutionary algorithms. The first column indicates minimum make-span achieved by each algorithm. Hybrid approach reaches to near optimal solutions in 78% of problems. Optimal solution obtained by SA for 57% of problems while GA could lead to near optimal solution in 42% of problems. The second column contains mean value of objective function (make-span) from five runs for each algorithm. MA reaches to minimum value of 65%. Simulated annealing and genetic algorithm achieved minimum mean value of 21% and 14% respectively. The optimum solution (shortest make-span) is attained by MA with 619.08.Standard deviation reveals that how tightly all the various solutions are clustered around the mean for every problem. Hybrid approach outstandingly has lower STDEV as it attains 64% of minimal standard deviation. Genetic algorithm has higher value of STDEV as only 7% of problem has minimal STDEV.As illustrated in Fig.6, GA reveals high fluctuations in reaching to near optimal solutions over the fourteen problems. Meanwhile it shows higher standard deviation than two other methods. MA has minimum value of standard deviation for the most of problems.

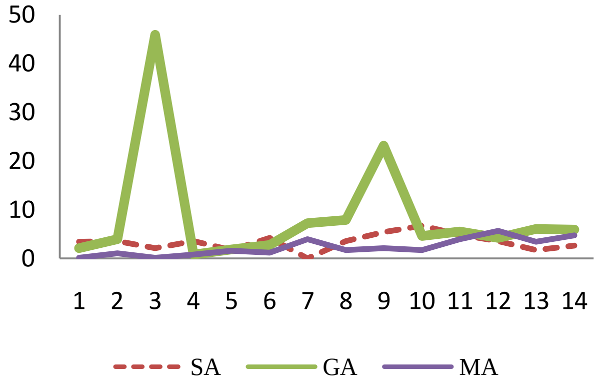

Fig. 6. STDEV comparison between GA, SA and MA

The mean STDEV is the mean value of standard deviation for all fourteen problems. MA has minimum value of 2.29. SA shows a bit increase in STDEV with value of 3.38 while GA has the highest standard deviation among other methods with 8.73. As depicted in TABLEVII, mean value of minimum of all fourteen problems in hybrid method is less than both SA and GA. MA, SA and GA reach to least average for all problems respectively. As MA has lowest standard deviation, it reveals superior performance over the two other algorithms in both finding optimal solutions and steadiness. It can be inferred from TABLEVII that the hybrid approach (MA) is statistically outperform other considered algorithms.

TABLE VII.COMPARISON BETWEEN SA, GA AND MA

	SA	GA	MA
AVG(MIN)	763.54	764.04	763.02
AVG(MEAN)	767.20	770.70	765.10
AVG (STDEV)	3.38	8.73	2.29

The results emphasizes on the importance of the neighborhood search operator embedded in simulated annealing in which works as an efficient fine tuning operator to solve the optimization problem. In contrast, permutation GA is reducing the computational effort at the beginning of the search by radically changing the search space but it might not be as efficient as SA to converge to a near optimal solution. However, MA provides a quite steady solution in several job allocation configurations by incorporating both global and local search operators.

VI. CONCLUSION

The memetic approach for sequencing has been successfully implemented. Hybrid's major advantage over other methods is an ability to prevent becoming stuck at local minima. A set of experiment is carried out to illustrate the effectiveness of MA in complex sequencing problems. Compared to GA and SA the lower make-span is achieved by MA can be attributed to the fact that the MA has higher degree of accuracy and steadiness in reaching to optimal solutions. The computational results reveal the competent performance of MA. There are potential opportunities to research in improving hybrid algorithms performance.

REFERENCES

[1] M. P. Groover, *Automation, production systems, and computer-integrated manufacturing*, p.^pp. 212-215, Upper Saddle River, N.J.: London : Prentice Hall ; Prentice-Hall International, 2001.
[2] D. R. Sule, *Industrial scheduling*, 20 Park Plaza,Co Boston: PWS Publishing Company. , 1997.
[3] R. S. Russell, and B. W. Taylor, *Operations management: quality and competitiveness in a global environment*: Wiley, 2006.
[4] V. Sekar, "Minimizing the make-span in a high-product mix shop-floor using integer programming," M.S., State University of New York at Binghamton, United States -- New York, 2007.
[5] R. Brahim, "Assembly line design: The balancing of mixed-model hybrid assembly lines with genetic algorithms,(Series in advanced manufacturing)," 2006, 2006.
[6] A. E. Eiben, and J. E. Smith, *Introduction to evolutionary computing*, Verlag Berlin Heidelberg New York: Springer, 2003.
[7] B. Naderi, M. Zandieh, A. Khaleghi Ghoshe Balagh *et al.*, "An improved simulated annealing for hybrid flowshops with sequence-dependent setup and transportation times to minimize total completion time and total tardiness," *Expert Systems With Applications*, 2008.
[8] M. Zandieh, S. M. T. Fatemi Ghomi, and S. M. Moattar Husseini, "An immune algorithm approach to hybrid flow shops scheduling with sequence-dependent setup times," *Applied Mathematics and Computation*, vol. 180, no. 1, pp. 111-127, 2006.
[9] B. Wang, Y. Rao, X. Shao *et al.*, "Scheduling Mixed-Model Assembly Lines with Cost Objectives by a Hybrid Algorithm." p. 387.
[10] S. Karabati, and S. Sayin, "Assembly line balancing in a mixed-model sequencing environment with synchronous transfers," *European Journal of Operational Research*, vol. 149, no. 2, pp. 417-429, 2003.
[11] P. R. McMullen, and G. V. Frazier, "A simulated annealing approach to mixed-model sequencing with multiple objectives on a just-in-time line," *IIE Transactions*, vol. 32, no. 8, pp. 679-686, 2000.
[12] W. L. Winston, *Introduction to mathematical programming: applications and algorithms*, Fourth Edition ed., CA93950, USA: Duxbury Resource Center, 2003.
[13] M. Vazquez, and D. Whitley, "A hybrid genetic algorithm for the quadratic assignment problem." pp. 135–142.
[14] L. Wang, and D.-Z. Zheng, "An effective hybrid optimization strategy for job-shop scheduling problems," *Computers & Operations Research*, vol. 28, no. 6, pp. 585-596, 2001.
[15] M. Yazdani, M. Gholami, M. Zandieh *et al.*, "A simulated Annealing Algorithm for Flexible Job Shop Scheduling Problem," *Journal of Applied Sciences*, pp. 1-9, 2009.
[16] M. Negnevitsky, *Artificial intelligence: a guide to intelligent systems*: Addison-Wesley, 2005.
[17] Y. K. Kim, C. J. Hyun, and Y. Kim, "Sequencing in mixed model assembly lines: A genetic algorithm approach," *Computers & Operations Research*, vol. 23, no. 12, pp. 1131-1145, 1996.
[18] Z. Michalewicz, *Genetic algorithms+ data structures= evolution programs*, Charlotte, USA: Springer, 1996.
[19] J. J. Grefenstette, "Optimization of control parameters for genetic algorithms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 16, no. 1, pp. 122-128, 1986.
[20] Y. Y. Leu, L. A. Matheson, and L. P. Rees, "Assembly Line Balancing Using Genetic Algorithms with Heuristic-Generated Initial Populations and Multiple Evaluation Criteria*," *Decision Sciences*, vol. 25, no. 4, pp. 581-606, 1994.
[21] C. Oysu, and Z. Bingul, "Application of heuristic and hybrid-GASA algorithms to tool-path optimization problem for minimizing airtime during machining," *Engineering Applications of Artificial Intelligence*, vol. 22, no. 3, pp. 389-396, 2009.
[22] Fogel, L. J., Owens, A. J., and Walsh, M. J. Artificial Intelligence through Simulated Evolution. John Wiley & Sons, New York, 1966.
[23] Moscato, P. On genetic crossover operators for relative order preservation.C3P Report 778, California Institute of Technology, Pasadena,CA 91125, 1989.
[24] Stalk, George. "Time--the next source of competitive advantage." (1988): 41-51.