

Comparative Analysis of Metaheuristic Approaches for m-Machine No Wait Flow Shop Scheduling for minimizing Total Flow Time with Stochastic Input

Laxmi A.Bewoor^{#1} V.Chandra Prakash ^{#2} Sagar U.Sapkal ^{*3}

^{#1}Research Scholar, Computer Sci. and Engg. Dept.

^{#2} Professor, Computer Sci. and Engg. Dept.

K.L.University, Guntur, Andhra Pradesh, India.

¹laxmiabewoor@gmail.com

²vchandrap@kluniversity.in

^{*3} Associate Professor, Mechanical Engg. Dept.

Walchand College of Engineering, Sangali, Maharashtra, India.

³sagarus1201@gmail.com

Abstract- This paper presents comparative analysis of metaheuristic algorithms for m-machine No wait flow shop scheduling (NWFSS) for finding optimal scheduling sequence considering minimization of total flow time (TFT). Such type of analysis for NWFSS is not available in the literature published till date. Because of combinatorial optimization behavior, NWFSS is NP Hard problem. In order to find the approximate solution, population based technique, evolutionary technique and population based evolutionary technique are implemented by considering stochastic input. The computational experimentations show that, PSO finds better solution compared with other techniques but the quality of solution can be improved by hybridizing PSO with guided random search technique.

Keywords: No wait flow shop scheduling, Combinatorial optimization, NP-Hard, Total flow time.

I. INTRODUCTION

Optimization problems are widely encountered in various fields of science and technology, which can be very complex due to the actual and practical nature of the objective function or the model constraints. Flow-shop scheduling problem (FSSP) is a combinatorial optimization problem and it is an NP-hard problem as it needs to select the optimal sequence of jobs from $n!$ feasible alternative sequences to meet the objective criterion. One of variant of FSSP is NWFSS, which considers optimal sequence of jobs on given number of machine by considering fixed processing sequence, which are intrinsically difficult. NWFSS problems has application found in many industries such as steel production [1], food processing [2], chemical industry [3], pharmaceutical industry [4], production of concrete wares [5] etc.

For a no-wait flow-shop problem, once a job is started on the first machine it has to be continuously processed until its completion at the last machine without interruptions and preemptions. Additionally, each machine can handle one job at a time and each job has to be processed by each machine exactly once without preemption. In order to meet this constraint it may be necessary to postpone the start of a job on given machine so that the completion of the operation coincides the starting of the operation on the next machine. This means the difference between the completion time of a job's last operation and the starting time of its first operation is equal to the sum of its operation times on all machines. During the past few years most researchers have proposed various heuristic methods to solve no wait flow shop scheduling problems. Swarm Intelligence nowadays getting more attention for solving various optimization problems. It is a computational intelligence technique involving the study of collective behavior in decentralized systems. Examples of such systems can be found in nature, including ant colonies, animal herding, bird flocking, bee swarms, and many more. PSO is a population-based evolutionary technique which can be used to solve the combinatorial optimization problems such as the traveling salesman problem, the job shop scheduling problem and transportation problem in network design etc [6].

This paper presents comparative analysis of metaheuristic algorithms for m-machine NWFSS for finding optimal scheduling sequence considering minimization of total flow time (TFT). Such type of analysis for NWFSS is not available in the literature published till date. In order to find the approximate solution, population based technique – Tabu Search(TS), evolutionary technique - Genetic Algorithm(GA) and population based evolutionary technique - Particle Swarm Optimization(PSO) are implemented by considering stochastic input.

II. RELATED WORK

The no wait flow shop scheduling problems with more than two machines belong to class of NP-Hard and can be solved with constructive heuristic approach and metaheuristic approach. Rajendran and Chaudhari[7], Betolissi[8], Aldowaisan and Allahverdi [9], Sapkal and Laha[10] developed an efficient heuristic algorithm by considering Total flow time(TFT) and makespan as objective function under consideration. Yuhui Shi and Russell Eberhart[11], B. Liu, L. Wang et al. [12], Q-K. Pan, M. Fatih Tasgetirenc et al. [13],I.-H.Kuo, S.-J. Horng et al.[14],A.G.Rao et al.[15] developed solutions with metaheuristic by using PSO as basic global optimization method.

Yuhui Shi and Russell Eberhart [11] found PSO may suffer in global search at the end of run due to utilization of linearly decreasing inertia weight. So, they proposed self adaptive strategy to maintain inertia weight and minimize computation time.

B. Liu, L. Wang et al. [12] proposed hybrid PSO algorithm by using ranked-order-value (ROV) rule based on random key representation to convert continuous position of particles to job permutations. Nawaz-Enscore-Ham (NEH) heuristic was used to initialize the particle position and velocity. They proposed NEH_1 insertion technique to improve the quality of a job permutation. SA is incorporated with adaptive meta-Lamarckian learning strategy to avoid premature convergence for local search.

Q-K. Pan, M. Fatih Tasgetirenc et al. [13] developed DPSO as a variant of PSO. DPSO considers local search with variable neighborhood descent (VND) algorithm based on variable neighborhood search (VNS) for makespan criterion.

I.-H.Kuo,S.-J.Horng et al. [14] developed individual enhancement(IE) scheme with PSO for improving local search quality and random encoding scheme to improve global search quality of PSO for minimizing makespan.

P. Damodaran, A. G. Rao et al. [15] proposed heuristics to modify particle position. Heuristic is improved with constructive Knapsack heuristic and batch formation NEH heuristic for particle generation; they achieved minimization of makespan.

III. PROBLEM DEFINITION

No Wait Flowshop scheduling has a set of n jobs and m machines. Every job is processed on each machine for processing time $p(i,j)$ without preemption. Sequence σ given with $\{\sigma_1, \sigma_2, \dots, \sigma_n\}$ is n jobs to be processed on m machines. The problem is to determine a sequence of n jobs which gives minimum Total Flow Time (TFT). Given matrix of size $n \times m$ with processing time $p[i,j]$ generates $(n!)$ number of feasible solution from which optimal sequence is to be opted. Delay between two machines is calculated by δ time. Where, $\delta(i,k)$ is minimum delay time between start of job i and start of job k where job k follows job i . ($1 \leq i \leq n, 1 \leq k \leq n, i \neq k$). Delay matrix represents delay time between current job and next job to be submitted. $\delta(\sigma_i-1, \sigma_i)$ denote minimum delay on the first machine between the start of two consecutive jobs i and $i-1$. The formula for total flow time (TFT) is given as:

$$TFT = \sum_{i=2}^n (n+1-i) \delta(\sigma_{i-1}, \sigma_i) + \sum_{i=1}^n \sum_{j=1}^m p(i,j) \quad (1)$$

$$\delta(i,k) = p(i,1) + \max((\sum_{i=1}^n p(i,h) - \sum_{k=i+1}^n p(k,h)), 0) \quad (2)$$

IV. METAHEURISTIC APPROACHES

The above problem is solved using two classical algorithmic approaches namely, population based technique (Tabu Search) and evolutionary based technique (Genetic Algorithm). The results are compared with population based evolutionary technique (PSO). The approaches are briefed below:

1. *Tabu Search (TS)*: TS works with basic three principles :

- Forbidding Strategy: Controls what enters in Tabu List.
- Freeing Strategy: Controls what exits from tabu list and when to exit
- Short Term Strategy: Manage between forbidding and freeing strategy to select trial solution.

Algorithm TabuSearch()

Inputs: Local Search, Neighbourhood, Aspiration condition, Tabu list and moves, stopping condition

1. Randomly select initial solution i in S_0 . Set $i^* = i$ and $k=0$

2. set $k=K+1$ and generate a subset V^* of solution in $N(i,k)$ such that one of the Tabu condition is violated.

3. Choose a best j in V^* and set $i=j$.

4. if $f(i) < f(i^*)$ then $i^* = i$

5. Update Tabu list

6. If not termination condition go to step 2

End

Though TS improves the solution to escape from a local optimum, but global optimum may not be found as it depends on parameter setting. The algorithm also suffers from finding too many parameters and requires large number of iteration affects overall computation time.

2. *Genetic Algorithm (GA)*: GA works with basic principle of survival of fittest. The algorithm maintains a population of candidates and evolution is possible with crossover, mutation, fitness evaluation stochastic operators.

Algorithm GA()

1. *Produce an initial population of individuals.*

2. *Evaluate fitness of all individuals*

3. *While not termination*

3.1. *find fit individual for reproduction*

3.2. *Recombine two individual*

3.3. *mutate individual*

3.4. *Evaluate fitness of new candidates*

3.5. *Generate new candidate population*

End

End

GA is very useful to provide the solutions where the traditional methods failed. But still GA suffers from finding global maxima. Time taken for convergence depends majorly on decent sized population and a lot of generations. Fine tuning all the parameters for the GA, like mutation rate, elitism percentage, crossover parameters, fitness normalization/selection parameters, etc, is often just trial and error.

3. *Particle Swarm Optimization (PSO)*: PSO is evolutionary population based algorithm incorporates five basic principles of swarm intelligence (1) quality principle; (2) proximity principle; (3) principle of diverse response; (4) principle of stability; and (5) principle of adaptability.

Initialize population particles **p** in search space with random positions and velocities

Loop

Update position of p according to optimization;

Update velocity of p according to optimization;

$$V_i(k+1) = \omega * V_i(k) + c1 * \text{rand}() * (P_i(k) - X_i(k)) + c2 * \text{rand}() * (g(k) - X_i(k))$$

$$X_i(k+1) = X_i(k) + V_i(k+1)$$

Where,

$V_i(k)$ is velocity of particle i at iteration k.

$X_i(k)$ is the position of particle i at iteration k.

$V_i(k+1)$ is velocity of particle i at iteration k+1.

ω is inertia weight

$X_i(k+1)$ is the position of particle i at iteration k+1.

$\text{rand}()$ is random number between (0,1).

c1 cognitive acceleration coefficient.

c2 social acceleration coefficient.

Map obtained position in solution space;

Evaluate fitness function;

Update local best, global best;

End loop

PSO is less sensitive to the nature of objective function and can handle objective functions with stochastic nature. It has the flexibility to be integrated with other optimization techniques to form hybrid tools. It has less parameter to adjust unlike many other competing evolutionary techniques. It has the ability to escape local minima. It does not require a good initial solution to start its iteration Process unlike population based techniques.

V. COMPUTATIONAL EXPERIENCE

The Metaheuristics under consideration were coded in Java and run on Intel Core i5, 8 GB RAM, 2.20 GHz PC. The experimentation is carried out in two phases. In first phase, small problem sizes with number of jobs(n)=6,8,10,12 and number of machines(m)=5,10,15,20,25 were considered. Second phase comprises of large problem sizes with n=20, 40, 60, 80,100 and m=5, 10,15,20,25. Each problem size in both phases has thirty independent problem instances. Each problem instance corresponds to a new processing time matrix generated randomly using uniform distribution $u(1,99)$ used more often in research community[10]. The experimentation measures the performance by average relative percentage deviation (ARPD) (for small problem size) and percent of best heuristic solution (for large problem size)

ARPD for small number of jobs problems is given by

$$ARPD = (100 / k) \sum_{i=1}^k \frac{Metaheuristic-Optimal}{Optimal} \quad (3)$$

$$ARPD = (100 / k) \sum_{i=1}^k \frac{Metaheuristic-Best}{Best} \quad (4)$$

Where, $Metaheuristic_i$ denotes the objective function value obtained for i^{th} instance by a Metaheuristics. $Optimal$ is optimal solution value obtained for that instance. $Best_i$ is optimal solution value obtained for that instance (PSO in this case) and k is the number of problem instances for a problem size. Table I displays comparative evaluation of PSO with GA and TS with enumerative method based on ARPD for small problem sizes ($n=6, 8, 10, 12$). The results of table shows that PSO approximates near optimal solution compared to GA and TS. PSO performs better than GA and TS in 43 cases out of 45 cases with respect to ARPD.

TABLE I. ARPD Obtained for Small Problem Sizes

Problem Size		No. of instances	ARPD		
n(jobs)	m(machines)		PSO	TS	GA
6	5	30	4.36	12.12	4.54
	10	30	3.81	10.80	8.34
	15	30	1.02	6.82	4.78
	20	30	1.79	8.10	4.54
	25	30	2.13	6.54	5.68
8	5	30	4.47	35.34	9.35
	10	30	2.97	15.09	10.32
	15	30	3.43	50.31	5.44
	20	30	5.01	10.83	7.54
	25	30	3.46	8.57	5.44
10	5	30	13.92	19.11	17.55
	10	30	46.41	17.67	17.89
	15	30	9.94	12.81	6.74
	20	30	5.19	42.71	7.56
	25	30	3.86	10.23	4.35
12	5	30	6.05	19.46	6.45
	10	30	6.79	19.52	7.56
	15	30	6.25	15.41	8.52
	20	30	6.16	14.14	7.95
	25	30	5.48	12.67	8.92

TABLE II. ARPD Obtained for Large Problem Sizes

Problem Size			ARPD	
n(jobs)	m(machines)	No. of instances	TS	GA
20	5	30	6.40	0.29
	10	30	22.95	24.65
	15	30	20.68	25.87
	20	30	18.48	27.65
	25	30	18.22	24.56
40	5	30	15.86	19.46
	10	30	16.76	24.88
	15	30	15.32	28.43
	20	30	18.45	25.31
	25	30	17.81	35.56
60	5	30	13.40	30.65
	10	30	15.44	29.65
	15	30	17.43	30.65
	20	30	16.84	28.59
	25	30	16.61	27.89
80	5	30	11.89	29.39
	10	30	13.76	26.84
	15	30	12.49	26.90
	20	30	18.72	25.71
	25	30	14.28	26.78
100	5	30	11.17	25.55
	10	30	13.73	25.83
	15	30	14.57	27.89
	20	30	15.72	28.95
	25	30	14.02	40.49

It has been observed that, ARPD values obtained (as shown in Table I and II) indicates that, PSO algorithm converges to near optimal solution for small as well as for large problem size.

V. CONCLUSION

This paper has discusses comparative analysis of some metaheuristic algorithms to optimize the total flow time of NWFSS problem with stochastic input. The metaheuristic under consideration were population based; evolutionary algorithms and population based evolutionary algorithms. The performances of the said algorithms are tested with optimum results for small problem size and with best metaheuristic results for large problem size. Computational experiences conclude that, PSO has many key advantages over other optimization techniques.

As PSO has no strong mathematical foundation there is wide scope to improve solution quality by hybridizing PSO with guided random search for NWFSS. Effectiveness and efficiency of such algorithms can be validated further with statistical analysis.

REFERENCES

- [1] Wismer DA. Solution of the flowshop-scheduling with no intermediate queues. Operations Research 1972,vol.20.
- [2] Hall NG, Sriskandarayah C. A survey of machine scheduling problems with blocking and no-wait in process. Operations Research 1996,vol.44.
- [3] Rajendran C. A no-wait flowshop scheduling heuristic to minimize makespan. Journal of the Operational Research Society 1994,vol.45.
- [4] Raaymakers W, Hoogeveen J. Scheduling multipurpose batch process industries with no-wait restrictions by simulated annealing. European Journal of Operational Research 2000,vol.126.
- [5] Grabowski J, Pempera J. Sequencing of jobs in some production system. European Journal of Operational Research 2000,vol.125.

- [6] Poli, R. "An analysis of publications on particle swarm optimisation applications" (PDF). Technical Report CSM-469 (Department of Computer Science, University of UK,2007.
- [7] C. Rajendran and D.Chaudhari,"Heuristic algorithms for continuous flow shop problem", Naval Research Logistic Quarterly, Vol.37.
- [8] E.Berlollisi,"Heuristic Algorithms for scheduling in the no wait flow shop",Journal of Materials Processing Technology, 2000,vol 107.
- [9] T.Aldowiasn and A.Allahverdi,"New heuristic for m machine no wait flow shop to minimize total completion time",Omega,vol.107.
- [10] S.U.Sapkal and Dipak Laha, "An Efficient Heuristic Algorithm for m-Machine No-Wait Flow Shops ", Proceedings of the International Multiconference of Engineers and Computer scientist,2011,vol1.
- [11] Yuhui Shi and Russell Eberhart, "Empirical study of particle swarm optimization", Evolutionary Computation, 1999
- [12] B. Liu, L. Wang, and Y.-H. Jin, "An effective hybrid particle swarm optimization for no-wait flow shop scheduling," Int. J. Adv. Manuf. Technol.,2008, vol. 31.
- [13] Q.-K. Pan, M. Fatih Tasgetiren, and Y.-C. Liang, "A discrete particle swarm optimization algorithm for the no-wait flowshop scheduling problem," Comput. Oper. Res.2008,, vol. 35.
- [14] I.-H. Kuo, S.-J. Horng, T.-W. Kao, T.-L. Lin, C.-L. Lee, T. Terano, and Y. Pan, "An efficient flow-shop scheduling algorithm based on a hybrid particle swarm optimization model," Expert Syst. Appl.,2009, vol. 36.
- [15] P. Damodaran, A. G. Rao, and S. Mestry, "Particle swarm optimization for scheduling batch processing machines in a permutation flowshop," Int. J. Adv. Manuf. Technol.,2012, vol. 64.

AUTHOR PROFILE

Mrs.Laxmi A.Bewoor is research scholar at K.L.University, Guntur. Her research interest includes Algorithmic Analysis and Artificial Intelligence.

Dr.V.Chandra Prakash is working as a Professor at K.L.University, Guntur. His research interest includes Artificial Intelligence and Data Mining.

Dr.Sagar U.Sapkal is working as a Associate Professor at Walchand college of Engineering, Sangali. His research interest includes optimization and manufacturing.