HYBRID PARTICLE SWARM ALGORITHM FOR SCHEDULING IN CELLULAR MANUFACTURING SYSTEM- A CASE STUDY

Qazi Salman Khalid[1], Muhammad Abas[1], Mudassar Rauf[2], Mirza Jahanzaib[2], Shahid Maqsood[1]

ABSTRACT

Cellular Manufacturing System (CMS) lies in the heart of lean manufacturing with goal of producing the wide variety of products as efficiently as possible. Increase in customer demand for more customized products had forced industries to shift to CMS. Once CMS has been established scheduling becomes one of the challenging task. So, in present work, a real case study based on scheduling problem in CMS is presented and a hybrid particle swarm optimization (PSO) algorithm is proposed to achieve an optimize sequence. The PSO is integrated with NEH algorithm to achieve an optimal sequence faster. A mathematical model is presented to evaluate two conflicting performance measures; minimization of work in process (WIP) and maximization of average machine cell utilization. Implementation of proposed algorithm had increased the utilization from 65% to 82 % while minimized the WIP to 6 parts from 25parts.

KEYWORDS*: Cellular Manufacturing, Work in process, Machine utilization, Particle swarm optimization, Hybrid algorithm*

INTRODUCTION

Scheduling is allocation of resources over a time to accomplish a pool of tasks. Scheduling denotes towards determination of sequences in which jobs should process over the production cycle, including indication of jobs start and finish times (Conway *et al.*, 2012). Its importance has been amplified in recent decades, owing to the increasing trend in diversity of demand, variety and altering markets with global competition as well as emerging developments of modern technologies. In addition, every plan/ schedule is evaluated under certain set of objectives which measures overall performance of the schedule. Scheduling in job shop is one of the most common and challenging optimization problems. Complexity of job shop scheduling in a manufacturing cell falls under the umbrella of non-deterministic polynomial-time (NP) hard problem.

In real-world manufacturing scenarios, scheduling is carried out to obtain multiple objectives optimization simultaneously. In manufacturing cell, intra-cell scheduling is addressed by Gholipour (2011) obtaining sequence of jobs. Objective functions included minimization of make-span, tardiness, cost of intra-cell movement and lastly sequence dependent setup. Savsar (2010), Mahdavi (2009), Tsourveloudis (2006), Vidalis (2005), Amar (2010), Braglia (2011), Rafiei (2015), Altarazi (2011), Pramanik, Karim and Kissani had optimized machine utilization, throughput, flow time, make-span and work in process inventroy. However, machine utilization and work in process inventory are the important performance measures, as they are directly related to the production cost and their analysis is important because it directly relates to profit of the firm and there is rarely found any work considering both factors at same interval.

Various novel techniques used to optimize different performance measures are discussed in the following lines. Gholipour (2011) optimized the various objective function with application of meta-heuristic based on scatter search. Mahdavi (2009) used genetic algorithm to optimize the performance measures. Lian (2006) and Tasgetiren (2007) presented a technique of conversion to apply PSO algorithm in FSSP and they compared the PSO with traditional GA. As concluded by authors, PSO algorithm performed better than the traditional GA. Pan (2008), Lian (2006), Tasgetiren (2007), Damodaran (2012), Tasgetiren (2004), Liao (2007), Zhang (2010), Tseng (2008) and Sankaran (2009) had used PSO algorithm for the flow shop scheduling problems.

In this paper, a novel algorithm based on integartion of NEH algorithm and PSO. The integration allows the algorithm to optimize the performance measures (WIP and machine utilization) more qickly as compared to traditional algorithm.

1 Department of Industrial Engineering, UET Peshawar
2 Department of Industrial Engineering, UET Taxila

METHODOLOGY AND MATHEMATICAL MODELING

In this section, mathematical modeling, objective function and constraints, is presented. An overview of methodology followed in this research is given in Fig. 1.

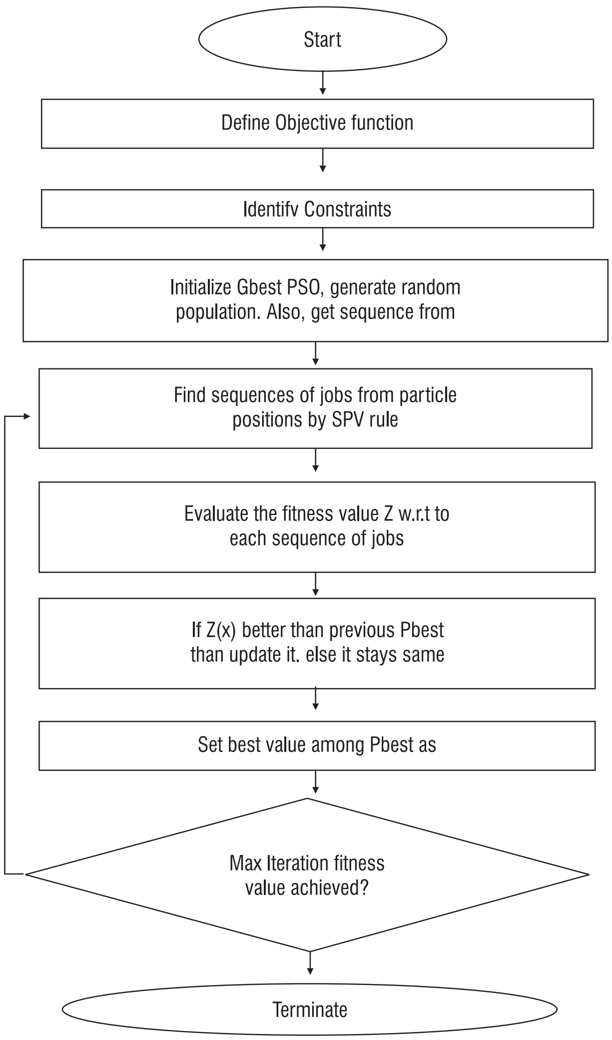

Fig. 1: Methodology overview

I is the index used to represent a part, i=1,2,3, …, n

J is the index used for a machine, j=1,2,3, …, M

AT_{ij} is the arrival time of part "i" on machine "j"

Pij is the process time of part "i" on machine "j"

$T_s(i,j)$ is the setup time of part "i" on machine "j"

ST_{ij} is the start time of part "i" on machine "j"

C_{ij} is the completion time of part "i" on machine "j"

W_{ij} is the waiting time in queue of part "i" on machine "j"

The objective functions and constraints formulated to minimize WIP and average utilization of machines are given in equations below;

$$Z_1 = \sum_{i=1}^{n} \frac{\sum_{j=1}^{m} P_{ij} + \sum_{j=1}^{m} W_{ij}}{T_c} \tag{1}$$

This equation was important for the minimization of work in process inventory.

$$Z_2 = \frac{\sum_{j=1}^{m} 1 - U_j}{m} \tag{2}$$

It is used for the utilization of machine in cellar manufacturing.

Subject to,

$$\sum_{j=1}^{m} Z_{ij} = 1 \tag{3}$$

First constraint in above equation illustrated that at a time one part will be processed by every machine.

$$\left(t_{o(ij)} + t_{s(ij)}\right) - TT_{ij} \leq 0 \tag{4}$$

Second constraint shows that summation of operation time and setup time for part "*i*" on machine "*j*" is always greater equal to or less than throughput time of that part and machine.

$$ST_{ij} = Max\left(C_{ij} + t_{rj(j-1)}, C_{(i-1)j}\right) \tag{5}$$

Third constraint stipulates that for a part "*i*" to start processing on machine "*j*"; maximum completion time is to be selected between: completion time plus traveling time of part "*i*" from previous machine and completion time of previous part "*i-1*" that is processed on machine "*j*".

$$C_{ij} - C_{i(j-1)} > 0 \tag{6}$$

Fourth constraint ensures that completion time for same part on machine "*j*" is always greater than completion time at machine "*j-1*".

Hybrid Particle Swarm Optimization Algorithm

The proposed hybrid PSO algorithm is a combination of two distinct algorithms; NEH and PSO. Motivation for integrating both these algorithms is due to need of better initial population for the search based algorithm. To imply the hybrid algorithm, a case study composing of a typical job shop scenario had been studied which consists of "m" machine and "n" parts having distinct routings with variable machine distances. Table 1 and Table 2 depicts process times and machine distances respectively.

Table 1: Process Times and Part Route

Parts\ Machine\ Operation	M1	1	M2	2	M3	3
1	4	M1	2	M2	3	M3
2	5	M2	6	M3	1	M1
3	3	M3	5	M1	4	M2
4	1	M2	3	M1	2	M3
5	5	M1	1	M2	2	M3
6	2	M2	3	M3	9	M1
7	1	M3	3	M1	4	M2
8	7	M2	1	M1	3	M3

Table 2: Machine Time Distance

Mch\Mch	M1	M2	M3
M1	0	1	2
M2	1	0	3
M3	2	3	0

The case problem consisted of eight parts and three machines. These parts made a single product with a demand of ten, in limited time of three hours. The average machine utilization is 65% and work in process inventory is 25 parts. So the aim of study is to find optimize sequence of parts such that WIP inventory for the cell is reduced and average machine utilization is increased.

Since it is a combinatorial optimization problem, in which various solutions are been encoded with discretized variables. Usually, in these problems, the set of solutions is converted to discrete one with a goal to determine the near to optimum solution. In PSO, constructing direct relationship among particles and problem domain for JSS problem, *n* number of dimensions are used for *n* number of jobs (*j*= 1… n); each dimension depicts a job. Every particle has continuous set of position values (PVs') that must be converted into discrete values for permutation purpose; hence, phenomenon of smallest position value (SPV) was applied. SPV arranges jobs in ascending order of their PVs' and determines its sequence. Solution representation framework for particle X_i^k is illustrated with its velocity and corresponding sequence in Table 3. It is evident that smallest position value among all is = 1.476, which relocates j = 6, giving it first value in the sequence and simultaneously other dimensions are arranged.

Steps of Simulation Integrated Hybrid PSO – NEH incorporated with case study are as following:

Step 1: Initialization

First of all, iteration k=0 and number of particles m= double the number of dimension were set. Afterwards, values of cognitive and social co-efficient C1, C2 were plugged. In search space, *m* particles were randomly generated i.e. $\{X_i^0, i = 1, n, m\}$ where, $X_i^0 = \left[x_{i1}^0, \ldots, x_{in}^0\right]$ and *n* is last job's dimension. Also, velocities were initialized

Table 3: Solution Representation

Dimension j	PV	Velocity	Ascending Order of PV	Final Sequence
	x_{ij}^0	v_{ij}^0		
1	5.16	1.073	1.476	6
2	9	-0.288	2.810	5
3	5.449	-0.348	5.16	1
4	8.916	3.184	5.449	3
5	2.810	-3.214	8.916	4
6	1.476	3.918	9	2
7	11.844	0.371	9.156	8
8	9.156	2.016	11.844	7

Table 4: First Particle of Initial Step

Dimension j	x_{ij}^0	v_{ij}^0	Ascending Order of PV	Sequence
1	5.16	1.073	1.476	6
2	9	-0.288	2.810	5
3	5.449	-0.348	5.16	1
4	8.916	3.184	5.449	3
5	2.810	-3.214	8.916	4
6	1.476	3.918	9	2
7	11.84	0.371	9.156	8
8	9.156	2.016	11.844	7

for those particles randomly, i.e., $\{V_i^0, i=1\ldots m\}$ where, = $\left[v_{i,1}^0,\ldots,v_{i,n}^0\right]$. For initializing positions and velocity in search space equation (7) and (8) were used simultaneously. Whereas X_{min} and X_{max} are minimum and maximum position values, Rn refers to random number between (0, 1).

Table 4 shows first particle as an outcome of initialization step for sample problem. Similarly, fifteen particles were initiated. SPV rule was applied on particles to determine sequence of parts.

$$x_{ij}^0 = x_{min} + \left(x_{max} - x_{min}\right) Rn\left(0,1\right) \quad (7)$$

$$v_{ij}^0 = v_{min} + \left(v_{max} - v_{min}\right) Rn\left(0,1\right) \quad (8)$$

Next, sub-optimal sequence by NEH heuristic for the jobs was determined with aid of simulation and it was floated as a particle in the swarm. For the sixteenth particle, we introduced sequence by solving the sequential problem with NEH algorithm with the steps:

Table 5: NEH Initial Step

Parts	Process +Move time(min)	Descending order of Parts
1	13	6
2	17	2
3	15	3
4	9	8
5	12	1
6	19	5
7	11	7
8	14	4

- All the processing times for a part *i* = 1…n on all machines were summed up on distinct routes including travelling time. These parts were arranged in descending order as illustrated in Table 5.

Select top two parts, switch their sequences and find make-span for both combinations with aid of simulation. Order of parts with minimal make-span was fixed. Next, the part with subsequent highest summed process times was selected and same procedure was repeated until all remaining parts were finished which resulted in a sequence (1 8 6 2 3 7 5 4).

All the sequences were evaluated by getting fitness values with the help of simulation. Simulation model assists to achieve throughput time for machines which leads to acquisition of WIP. The model also determines component of idle time for machines, hence calculates average underutilization of machine cell. The simulation of parts for first initialized particle resulted in throughput time of 95 mins. Illustration of Objective Function Evaluation is prescribed in which WIP = TT/ Tc; where Tc = (Available Time/ Demand) i.e. 90 minutes/ 10 = 9. Utilization of machine j = Sum of Processes time on j / (Max. of machine j completion times – Min. of machine j start times). TT = 95, Tc = 9. Calculated WIP = 10.55 ≈ 11.

Utilization of M1, M2 and M3 on calculation came out to be 0.378, 0.253 and 0.341 respectively with an overall average utilization of 0.330. Hence, average under- utilization = 1– 0.324= 0.6758 → 67.58 %. Objective function value (Z) was 39.06. This value was obtained by combining the normalized values of both performance measures; WIP and Utilization. Fitness value to be personal best (PB) for each particle in the swarm at first iteration was set. Minimum of the PB values was set as global best (GB).

Step 2: Update Iteration, inertia weight, velocity and position value.

Iteration counter was updated such as k = k+1. For position and velocity updates a sample calculation is illustrated: From initial population of first particle we attained position values and their velocities, table (4). The corresponding fitness value of the first particle for k=0 is f_1^0= 39.06 and global best fitness value from swarm is f_9^0= 25.639, these values are stored as: $f_1^0 = P_{best,1}^0 = X_1^0 and f_9^0 = G_{best}^0 = X_9^0$ as evident from table (4). Constants are set as, c1, c2 = 2, r1, r2 = 0.5, inertia weight (w) = 1.2 and α = 0.92 and plugged in equation (9) for velocity update; V_{11}^1 = 1.2*0.92*3.91896 + 2*0.5 (45.56-5.16) + 2*0.5 (25.639-5.16) = 3.310. Updated position values are derived by equation (10): x_{11}^1 = 5.16 + 3.310 = 8.470. Similar procedure is pursued for all dimensions of X_1^1 and calculated values are evident from Table 6. Inertia $w^{k-1} * a$ was modified step wise by:

$$v_{ij}^k = w^k v_{ij}^{k-1} + c_1 r_1 (pb_{ij}^{k-1} - x_{ij}^{k-1} + c_2 r_2 \left(gb_j^{k-1} - x_{ij}^{k-1}\right) \quad (9)$$

$$x_{ij}^k = x_{ij}^{k-1} + v_{ij}^k \quad (10)$$

Step 3: Sequence by SPV rule

Sequence of jobs was determined and their fitness values were evaluated.

Step 4: Update PB and GB values and Iterate

Table 6: Updating Velocity to get new PV and Sequence

Dimensio J	Previous Velocity	Previous Position Value	New Velocity	New Position Value	PV Ascending Order	Job Sequence
1	3.918	5.16	3.3101	8.4701	2.757254	3
2	-3.21	9	-4.796	4.2035	4.203567	2
3	1.073	5.449	-2.691	2.7572	5.129664	7
4	-0.34	8.916	1.1221	10.038	5.505446	8
5	3.184	2.810	3.9207	6.7311	6.731136	5
6	-0.28	1.476	8.5260	10.002	8.470132	1
7	2.016	11.84	-6.714	5.1296	10.00205	6
8	0.371	9.156	-3.650	5.5054	10.03812	4

Cycle

If the function values obtained were minimum than the previous ones, their particle positions were updated in library of personal and global best positions. For next cycle of iteration, repeat the steps from update iteration.

Step 5: Termination

Iterations were stopped when minimum function value did not decrease for 10 iteration. Minimum function value decreased to 17.021 from 25.8 till 13^{th} iteration and then remain constant. Hence, the termination condition reached. To be on save side, iterations were run till 35^{th} iteration but the minimum function value did not change. So, the proposed algorithm had stopped.

RESULTS AND DISCUSSION

The proposed PSO-NEH algorithm was applied on job-shop problem. Convergence of particles toward optimum search point is shown in Fig. 2.

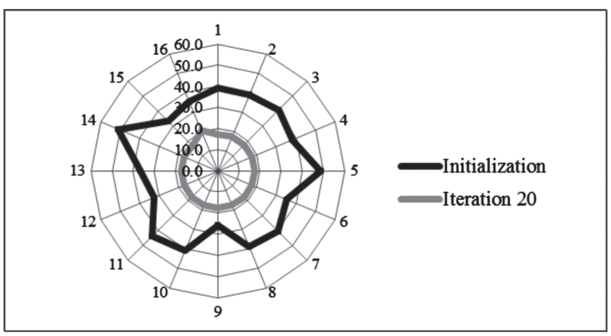

Fig. 2: Convergence of Particles from Initial and 20th

It can be seen that for initial population, minimum particle value was 25.63 while for 20^{th} iteration it was 17.021. Comparison of initial and 20^{th} iteration was given in Fig. 3.

The optimal sequence of jobs achieved is [3 7 1 5 4 8 2 5] with a normalized objective function value of 17.021 which is 82 % machine utilization and 6 part of WIP. So the work in process inventory is reduced from 25 parts to 6 parts while machine utilization is increased from 65% to 82%.

As population size consisted of sixteen particles with

Fig. 3: Comparison normalized function value of Initial and 20th iteration

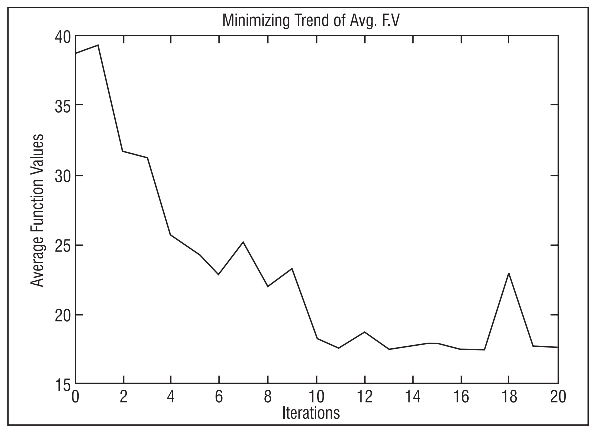

Fig. 4: Trend of Avg. function values against iterations

eight dimensions wandering in search space, their average objective function value was plotted against respective iterations and a minimization trend is evident. Particles are eager to converge and trying to overcome local optima as depicted from Fig. 4.

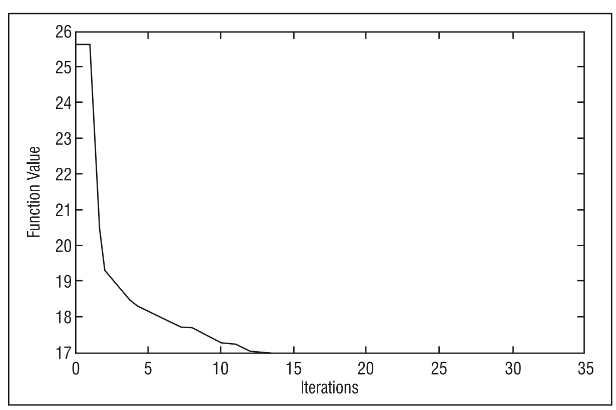

Fig. 5: G-best values from Proposed DPSO-NEH Algorithm

It showed the average normalized function value against various iteration. It can be seen from the Fig. (5) that convergence had occurred rapidly. Minimum objective function value of 17.021 had achieved in 12^{th} iteration and remained constant for 35 of iterations. So, algorithm was terminated as no reasonable change was noticed. Fast convergence and better performance of the proposed algorithm can also be verified from the literature. As in literature, Sha (2006), Lin (2010), Zhao (2006) and Dousthaghi (2013) had proved that hybrid PSO algorithm perform better for job shop scheduling as compared to traditional PSO algorithm. While Mirabi (2011), Marichelvam (2014), Sun (2010) and Nouha (2015) had confirmed that integration of NEH algorithm with other meta-heuristic algorithms resulted in faster convergence. Above mentioned literature had verified our findings that hybrid PSO algorithm with the integration of NEH algorithm performed better as compared to traditional PSO algorithm.

REFERENCES

1. *Conway, R. W., Maxwell, W. L., Miller, L. W., (2012), "Theory of scheduling. Courier Corporation.*

2. *Gholipour-Kanani, Y., Tavakkoli-Moghaddam R., Khorrami A., (2011), "Solving a multi-criteria group scheduling problem for a cellular manufacturing system by scatter search", Journal of the Chinese Institute of Industrial Engineers, Vol 28 (3): pp. 192-205.*

3. *Savsar, M., (2010), " Reliability Modeling of a Manufacturing Cell Operated under Degraded Mode".*

4. *Mahdavi, I., Paydar, M. M., Solimanpur, M., Heidarzade, A., (2009), "Genetic algorithm approach for solving a cell formation problem in cellular manufacturing", Expert Systems with Applications, Vol 36 (3): pp. 6598-6604.*

5. *Tsourveloudis, N. C., Doitsidis, L., Ioannidis, S., (2006), "Work-in-process scheduling by evolutionary tuned distributed fuzzy controllers", In: Robotics and Automation, ICRA 2006. Proceedings 2006 IEEE International Conference on, 2006. IEEE, pp 1420-1425*

6. *Vidalis, M., Papadopoulos, C. T., Heavey, C, (2005), "On the workload and 'phaseload'allocation problems of short reliable production lines with finite buffers", Computers & Industrial Engineering, Vol 48 (4), pp. 825-837.*

7. *Amar MAB, Camus H, Korbaa O., (2010), "A mathematical model for cyclic scheduling with assembly tasks and Work-In-Process minimization. In: Frontiers of Assembly and Manufacturing. Springer, pp 279-292*

8. *Braglia, M., Frosolini, M., Gabbrielli, R., Zammori, F, (2011), "CONWIP card setting in a flow-shop system with a batch production machine", International Journal of Industrial Engineering Computations, Vol 2 (1), pp. 1-18.*

9. *Rafiei, H., Rabbani, M., Nazaridoust, B., Ramiyani, SS. (2015), "Multi-objective cell formation problem considering work-in-process minimization", Int J Adv Manuf Technol, Vol 76 (9-12), pp. 1947-1955.*

10. *Altar1azi, S. A., (2011), "An optimization tool for operational tolerances allocation, work in process inventory minimization, and machines assignment in a discrete part manufacturing environment. Int J Adv Manuf Technol, Vol 55, pp. 1069-1078.*

11. *Pramanik, N., Process Flow Simulation to Reduce WIP Built-up and to Maximize Throughput in a Multi-part Multi-operation Process–A Case Study.*

12. *Karim, A., Rahman, M. A., Yusop, N. H., "Mitigation of WIP-related Problems at an IC Production Line through a Suitable Inspection Sampling Plan".*

13. *Kissani, I., Bouya, W., "Analysis of WIP Inventory Control and Simulation of KANBAN System within Wiring Harness Company".*

14. *Lian, Z., Gu, X., Jiao, B., (2006), "A similar particle swarm optimization algorithm for permutation flowshop scheduling to minimize makespan". Applied Mathematics and Computation, Vol 175 (1), pp. 773-785*

15. *Tasgetiren, M. F., Liang, Y. C., Sevkli, M., Gencyilmaz,*

G., (2007), "A particle swarm optimization algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing problem", European Journal of Operational Research, Vol 177 (3), pp. 1930-1947.

16. *Pan Q. K., Tasgetiren, M. F., Liang, Y. C., (2008), "A discrete particle swarm optimization algorithm for the no-wait flowshop scheduling problem", Computers & Operations Research, Vol 35 (9), pp. 2807-2839*

17. *Damodaran, P., Diyadawagamage, D. A., Ghrayeb, O., Vélez-Gallego, M.C., (2012), "A particle swarm optimization algorithm for minimizing makespan of nonidentical parallel batch processing machines", Int J Adv Manuf Technol. Vol 58 (9-12), pp. 1131-1140.*

18. *Tasgetiren, M. F., Sevkli, M., Liang, Y. C., Gencyilmaz, G., (2004), "Particle swarm optimization algorithm for single machine total weighted tardiness problem", In: Evolutionary Computation, CEC2004. Congress on, 2004. IEEE, pp. 1412-1419.*

19. *Liao, C. J, Tseng, C. T, Luarn, P., (2007), "A discrete version of particle swarm optimization for flowshop scheduling problems, Computers & Operations Research, Vol 34 (10), pp. 3099-3111.*

20. *Zhang, J., Zhang, C., Liang, S., (2010), "The circular discrete particle swarm optimization algorithm for flow shop scheduling problem", Expert Systems with Applications, Vol 37 (8), pp. 5827-5834.*

21. *Tseng, C. T., Liao, C. J., (2008), "A particle swarm optimization algorithm for hybrid flow-shop scheduling with multiprocessor tasks" International Journal of Production Research, Vol. 46 (17), pp. 4655-4670.*

22. *Sankaran, V., (2009), "A Particle Swarm Optimization Using Random Keys For Flexible Flow Shop Scheduling Problem With Sequence Dependent Setup Times", Clemson University.*

23. *Sha, D., Hsu, C. Y., (2006), "A hybrid particle swarm optimization for job shop scheduling problem", Computers & Industrial Engineering, Vol 51 (4), pp. 791-808.*

24. *Lin, T. L., Horng, S. J., Kao, T. W., Chen, Y. H., Run, R. S., Chen, R. J., Lai, J. L., Kuo, I. H., (2010),"An efficient job-shop scheduling algorithm based on particle swarm optimization", Expert Systems with Applications, Vol 37(3), pp. 2629-2636.*

25. *Zhao, F., Zhu, A., Yu, D., Yang, Y., (2006), "A hybrid Particle Swarm Optimization (PSO) algorithm schemes for integrated process planning and production scheduling. In: Intelligent Control and Automation", WCICA 2006. The Sixth World Congress IEEE, pp. 6772-6776.*

26. *Dousthaghi S, Tavakkoli-Moghaddam R, Makui A., (2013), "Solving the economic lot and delivery scheduling problem in a flexible job shop with unrelated parallel machines and a shelf life by a proposed hybrid PSO", The International Journal of Advanced Manufacturing Technology, 68 (5-8), pp. 1401-1416.*

27. *Mirabi M., (2011), "Ant colony optimization technique for the sequence-dependent flowshop scheduling problem", The International Journal of Advanced Manufacturing Technology, Vol 55 (1-4), pp. 317-326.*

28. *Marichelvam M., Prabaharan T., Yang X. S., (2014), "Improved cuckoo search algorithm for hybrid flow shop scheduling problems to minimize makespan", Applied Soft Computing, Vol 19, pp. 93-101.*

29. *Sun Y., Liu M., Zhang C., Gao L., Lian K., (2010), "New high performing hybrid particle swarm optimization for permutation flow shop scheduling problem with minimization of makespan", In: Industrial Engineering and Engineering Management (IEEM), 2010 IEEE International Conference IEEE, pp. 1706-1710.*

30. *Nouha N., Talel L., (2015), "A Particle Swarm Optimization Metaheuristic for the Blocking Flow Shop Scheduling Problem: Total Tardiness Minimization", In: European Conference on Multi-Agent Systems, 2015. Springer, pp. 145-153.*