ISSN 2319 – 7757 EISSN 2319 –7765

Indian Journal of Engineering

Review on scheduling using particle swarm optimization

Shivasankaran N[1]☼, Senthil Kumar P[2]

1. Associate Professor, Department of Mechanical Engineering, K. S. R. College of Engineering, Tiruchengode – 637 215, Tamilnadu, India
2. Professor, Department of Mechanical Engineering, K. S. R. College of Engineering, Tiruchengode – 637 215, Tamilnadu, India

☼***Corresponding author:*** Associate Professor, Department of Mechanical Engineering, K. S. R. College of Engineering, Tiruchengode – 637 215, Tamilnadu, India. Mail:Shivasankaran.n@gmail.com

Received 20 August; accepted 23 October; published online 29 October; printed 29 October 2013

ABSTRACT

Particle swarm optimization is an evolutionary based computational technique used to solve the scheduling problems. Particle swarm optimization has few parameters to adjust, is easy to implement and has special characteristic of memory. This paper describes how the current literature uses the Particle swarm optimization approach to solve scheduling problems. An analysis of the literature allows one to conclude that Particle swarm optimization is one of the viable approaches to solve scheduling problems. On the basis of the literature review, we were not only able to derive certain guidelines for the implementation of Particle swarm optimization algorithms but also to determine possible directions for future research.

Keywords: Particle swarm optimization, Job shop scheduling.

Abbreviation: PSO- Particle Swarm Optimization

How to Cite This Article
Shivasankaran N, Senthil Kumar P. Review on scheduling using particle swarm optimization. *Indian Journal of Engineering*, 2013, 5(14), 75-80

1. INTRODUCTION

Particle Swarm Optimization (PSO) is an evolutionary computation technique, developed for optimization of continuous nonlinear, constrained and unconstrained, non-differentiable multi-modal functions (Kennedy et al. 1995). PSO is inspired firstly by general artificial life, the same as bird flocking, fish schooling and social interaction behaviour of human and secondly by random search methods of evolutionary algorithm (Jie et al. 2008). Animals, especially birds, fishes etc. always travel in a group without colliding, each member follows its group, adjust its position and velocity using the group information, because it reduces individual's effort for search of food, shelter etc. Particle swarm optimization is evolutionary technique similar to genetic algorithm because both are population based and are equally affective.

Particle swarm optimization has better computational efficiency, i.e. it requires less memory space and lesser speed of Central Processing Unit, it has less number of parameters to adjust. Genetic algorithm and other similar techniques (e.g. simulated annealing), work for discrete design variables, whereas particle swarm optimization works for discrete as well as analogue systems, because it is inherently continuous, does not need D/A or A/D conversion. Although for handling discrete design variables, Particle swarm optimization needs some modification to be done.

In this paper PSO applied in the large field of computing, including the development of the algorithm is described in Section 2. Section 3 briefly reviews the standard equations of PSO which are usually related with engineering problems. The parameter tuning and its insights are dealt in section 4. Section 5 presents the categories of scheduling problems. Section 6 explores the PSO algorithm tailed by over view of literature in section 7.

2. CONCEPT OF PARTICLE SWARM OPTIMIZATION

Human being have their own previous experiences, set beliefs and set rules of doing some work, based on which they take their actions (individual best position) also human follow the path set by society/group, this path is supposed to be the best according to the whole group (global best position) (Kennedy et al 1995). The essence for the development of particle swarm optimization was the assumption that potential solution to an optimization problem is treated as a point flying like bird in multi-dimensional space, adjusting its position in search space according to its own previous experience and that of its neighbour (Wei et al 2008). This point has got no mass and volume and is called a particle as it has velocity vector. As per Shi and Eberhart (Y. Shi et al 1998), to bring the particle to best position, initial velocity is incremented in either positive or negative direction depending on current value, if present position is less than best position then increase of velocity and vice versa is done (Kennedy et al. 1995).

$$V_x = w \times V_x + C_1 \times rand_1 \times (Pbest_x - Present_x) + C_2 \times rand_2 \times (Gbest_x - Present_x) \quad (1)$$

Shivasankaran et al.
Review on scheduling using particle swarm optimization,
Indian journal of engineering, 2013, 5(14), 75-80,

V_x and present value have two matrices to show number of particles and dimensions. In the absence of previous velocity, particles will have no momentum and will get trapped in still position. C_1 and C_2 represent weighing of stochastic acceleration terms that pull particle towards Pbest and Gbest positions. Above equation has both cognizant parts (its own previous best value, derived from human behaviour of memory and experience) and social part (derived from human as well as animal behaviour to follow the path set by society or group). Population size is generally problem dependent and kept in between 20 and 50 (Eberhart, 1995).

3. MODIFIED EQUATION WITH ADDITION OF NEW OPERATORS (Inertia weight and Constriction factor)

To reach the target in a group of population, random numbers of particles are generated with random positions and random velocities. Their velocity vectors and positions are updated in number of iterations and then their fitness is calculated according to objective function (Hassan et al. 2004). Simulation is processed through simple equations (2) and (3) which are used as equations of standard particle swarm optimization (SPSO). The particle moves towards an optimum solution through its present velocity and its individual best solution obtained by itself in each iteration and global best solution is obtained by all particles. In a physical dimensional search space, updated position and velocity of the 'ith' particle are represented as

$$V_{ik+1} = K(V_{k\,i} * \omega + C_1 * R_1 * (P\ best(i) - X_{ki}) + C_2 * R_2 * (G\ best - X_{ki})) \quad (2)$$

$$X_{ik+1} = X_{ik} + V_{ik+1} \quad (3)$$

Where $X_i = [X_{i1}, X_{i2}, \ldots, X_{id}]$
$V_i = [V_{i1}, V_{i2}, \ldots, V_{id}]$,
Here 1, 2. . . d shows possible dimensions for i = 1, 2. . . i particles with position X and velocity V.
Pbest (i) = [$X_{i1\ Pbest}$, $X_{i2\ P\ best}$. . . $X_{idP\ best}$] represent individual best positions of particle i.
G best = [X_{1Gbest}, $X_{2G\ best}$. . . $X_{nG\ best}$] represent the global best positions. k represents the iteration number for total n iterations.

ω is inertia weight, K is constriction factor, C_1 and C_2 are non-negative coefficients called acceleration factors, R_1 and R_2 are two random numbers different from each other and generally distributed in the range [0, 1].

In a simplified model taking into account only one dimension of particle P best and G best are combined into one single term, making convergence and divergence of a particle easy for some of the benchmark functions (Yasada et al. 2004).

4. EFFECT OF PSO PARAMETERS IN PROBLEM SOLVING

Searching for reaching success or optimization depends on selection of appropriate parameters and their tuning through the search process. A brief discussion of adjustment of parameters of particle swarm optimization is given in this section.

4.1. Average and maximum velocity

To converge to optimum point craziness or neighbour matching velocity concept was initially proposed, and then it is suggested to make an increment in desired direction by some random number. Average velocity gradually decreases, as a good solution tends to be obtained. Average velocity of large scale problem is larger than that of a small scale problem under the condition of same parameters. Success or failure of search is related to the average of absolute value of velocity (Yasada et al. 2004). Particle's velocity in each direction is clamped to some maximum value V_{max}, so that particles do not try to overpass the search space. V_{max} is an important parameter to determine resolution; too high value may make the particle fly past the good solution and too low value will cause particles to be trapped in local minima, not allowing them to travel far places in search of good solution in the search space (R. C. Eberhart et al 2001). In early experiments V_{max} was set at 10 – 20% of dynamic range of variable in each dimension.

4.2. Inertia Weight

Motivated by the desire to better control the scope of the search, reduce the importance of *V*max, and perhaps eliminate it altogether, the following modification of the PSO's update equations was proposed (Shi and Eberhart 1998):

$$\vec{v_i} \leftarrow \vec{\omega}\_\vec{v_i} + U(0,\varphi_1) * (\vec{p_i} - \vec{x_i}) + U(0,\varphi_2) * (\vec{p_g} - \vec{x_i}), \quad (4)$$

$$\vec{x_i} \leftarrow \vec{x_i} + v_i, \quad (5)$$

Where ω - inertia weight

Note that if we interpret $U(0,\varphi_1) * (p_i - x_i) + U(0,\varphi_2) * (p_g - x_i)$ as the external force f_i, acting on a particle, then the change in a particle's velocity (i.e., the particle's acceleration) can be written as $v_i = f_i - (1 - \omega)v_i$. That is, the constant $1 - \omega$ acts effectively as a friction coefficient, and so ω can be interpreted as the fluidity of the medium in which a particle moves. This perhaps explains why researchers have found that the best performance could be obtained by initially setting ω to some relatively high value, which corresponds to a system where particles moves in a low viscosity medium and perform extensive exploration, and gradually reducing ω to a much lower value, where the system would be more dissipative and exploitative and would be better at homing into local optima. It is even possible to start from values of ω>1, which would make the swarm unstable, provided that the value is reduced sufficiently to bring the swarm in a stable region (the precise value of ω that guarantees stability depends on the values of the acceleration coefficients. Naturally, other strategies can be adopted to adjust the inertia weight. For example, in (Eberhart and Shi 2000) the adaptation of ω using a fuzzy system was reported to significantly improve PSO performance. Another effective strategy is to use an inertia weight with a random component, rather than time-decreasing. For example, (Eberhart and Shi, 2001) successfully used ω = U (0.5, 1).

With equation 5 and an appropriate choice of ω and of the acceleration coefficients, φ_1 and φ_2, the PSO can be made much more stable,so much so that one can either do without V_{max} or can set V_{max} to a much higher value, such as the value of the dynamic range of each variable (on each dimension). In this case, V_{max} may improve performance, though with use of inertia or constriction techniques, it is no longer necessary for damping the swarm's dynamics. In the velocity updating process, the value of the parameters such as V, C_1, C_2, K should be determined in advance. The inertia weight ω is linearly decreasing as the iteration proceeds and obtained as

Shivasankaran et al.
Review on scheduling using particle swarm optimization,
Indian journal of engineering, 2013, 5(14), 75-80,

$$\omega = \omega_{max} - \frac{(\omega_{max} - \omega_{min})iteration}{iteration_{max}} \quad (6)$$

Where ω_{max}: final inertia weight; ω_{min}: initial inertia weight; iteration: current iteration number; $iteration_{max}$: maximum iteration number.

5. SCHEDULING PROBLEMS

Scheduling is broadly defined as the process of assigning a set of tasks to resources over a period of time or it may be defined as ‘‘the allocation of resources over time to perform a collection of tasks’’. Scheduling problems in their simple static and deterministic forms are extremely simple to describe and formulate, but are difficult to solve because they involve complex combinatorial optimization. For example, if n jobs are to be performed on *m* machines, there are potentially (*n*!) *m* sequences, although many of these may be infeasible due to various constraints.

5.1. Open Shop Scheduling

Open shop scheduling problem (OSSP) is a scheduling problem where, given n jobs and m work stations, each job has to be processed on workstations at least once. However, some of these processing times may be zero. The order in which this happens is not relevant (in contrast to the job-shop problem, where the order of job does not matter).

5.2. Flow-Shop Scheduling

A flow-shop is a shop design in which machines are arranged in series Jobs begin processing on an initial machine, proceed through several intermediary machines, and conclude on a final machine. In each job exactly one operation for every machine, all jobs go through all the machines in the same order.

5.3. Job-Shop Scheduling

A job-shop does not have the same restriction on workflow as a flow-shop. In a job-shop, jobs can be processed on machines in any order. The usual job shop, from a research standpoint, is one in which there are *m* machines and *n* jobs to be processed. Each job requires *m* operations, one on each machine, in a specific order, but the order can be different for each job. Real job-shops are more complicated. Jobs may not require all machines and yet they may have to visit some machines more than once. Clearly, workflow is not unidirectional in a job-shop. Any given machine may observe new jobs arriving from outside the shop (as new inputs), and from other machines within the shop (as WIP), the same machine may be the last machine for a particular job, or it may be an intermediate processing step.

6. PARTICLE SWARM OPTIMIZATION ALGORITHM

Each individual in the particle swarm is composed of three D-dimensional vectors, where D is the dimensionality of the search space. These are the current position X_i, the previous best position P_i, and the velocity V_i. The current position $X_i$can be considered as a set of coordinates describing a point in space. On each iteration of the algorithm, the current position is evaluated as a problem solution. If that position is better than any that has been found so far, then the coordinates are stored in the second vector, P_i. The value of the best function result so far is stored in a variable that can be called $Pbest_i$ (for “previous best”), for comparison on later iterations. The objective is to keep finding better positions and updating $P_i$and $Pbest_i$. New points are chosen by adding $V_i$coordinates to X_i, and the algorithm operates by adjusting V_i, which can effectively be seen as a step size.

The particle swarm is more than just a collection of particles. A particle by itself has almost no power to solve any problem; progress occurs only when the particles interact. Problem solving is a population-wide phenomenon, emerging from the individual behaviors of the particles through their interactions. In any case, populations are organized according to some sort of communication structure or topology, often thought of as a social network. The topology typically consists of bidirectional edges connecting pairs of particles, so that if j is in i’s neighborhood, i is also in j’s. Each particle communicates with some other particles and is affected by the best point found by any member of its topological neighborhood. This is just the vector $p_i$for that best neighbor, which we will denote with p_g. The potential kinds of population “social networks” are hugely varied, but in practice certain types have been used more frequently. In the particle swarm optimization process, the velocity of each particle is iteratively adjusted so that the particle stochastically oscillates around $P_i$and $P_g$locations. The outline of the standard PSO process is as follows:

Step 1: Initialize a population of particles with randompositions and velocities in a D-dimensional searchspace.
Step 2: Update the velocity of each particle using Eq. (2).
Step 3: Update the position of each particle using Eq. (3).
Step 4: Map the position of each particle into the solutionspace and evaluate its fitness value according to thedesired optimization fitness function.Simultaneously update the pbest and gbestpositions if necessary.
Step 5: Loop to Step 2 until the termination criterion is met, usually after a sufficient good fitness or a maximum number of iterations.

7. OVERVIEW OF LITERATURE

A hybrid particle swarm optimization (PSO) for the job shop problem (JSP) is proposed by D.Y. Sha, Cheng-Yu Hsu, 2006. Since the solution space of the JSP is discrete, they modified the particle position representation, particle movement, and particle velocity to better suit PSO for the JSP. They also modified the particle position based on preference list-based representation, the particle movement based on swap operator, and the particle velocity based on the concept of tabu list.

Bo Liu, Ling Wang, Yi-Hui Jin (2007), proposed an effective hybrid algorithm based on particle swarm optimization (HPSO) for permutation flow shop scheduling problem (PFSSP) with the limited buffers between consecutive machines to minimize the maximum completion time (i.e., makespan). First, a novel encoding scheme based on random key representation is developed, which converts the continuous position values of particles in PSO to job permutations. Second, an efficient population initialization based on the famous Nawaz Enscore–Ham (NEH) heuristic is proposed to generate an initial population with certain quality and diversity. Third, a local search strategy based on the generalization of the block elimination properties, named block-based local search, is probabilistically applied to some good particles. Moreover, simulated annealing (SA) with multi-neighborhood guided by an adaptive meta-Lamarckian learning strategy is designed to

Shivasankaran et al.
Review on scheduling using particle swarm optimization,
Indian journal of engineering, 2013, 5(14), 75-80,

prevent the premature convergence and concentrate computing effort on promising solutions. Simulation results and comparisons demonstrate the effectiveness of the proposed HPSO.

D.Y. Sha, Cheng-Yu Hsu (2007), presented a new particle swarm optimization (PSO) for the open shop scheduling problem. Compared with the original PSO, they modified the particle position representation using priorities, and the particle movement using an insert operator. They also implemented a modified parameterized active schedule generation algorithm (mP-ASG) to decode a particle position into a schedule. In mP-ASG, we can reduce or increase the search area between non-delay schedules and active schedules by controlling the maximum delay time allowed.

Deming Lei (2008) presented a particle swarm optimization algorithm for solving multi-objective job shop scheduling problem. The objective is to simultaneously minimize makespan and total tardiness of jobs.I-Hong Kuo et al. (2009) used hybrid particle swarm optimization model named HPSO that combines random-key (RK) encoding scheme, individual enhancement (IE) scheme, and particle swarm optimization (PSO) is presented and used to solve the flow-shop scheduling problem (FSSP). The objective of FSSP is to find an appropriate sequence of jobs in order to minimize makespan. Makespan means the maximum completion time of a sequence of jobs running on the same machines in flow-shops.Changsheng Zhang et al. (2010) proposed a hybrid alternate two phases particle swarm optimization (PSO) algorithm called ATPPSO is proposed to solve the flow shop scheduling problem (FSSP) with the objective of minimizing makespan which combines the PSO with genetic operators and annealing strategy.

A hybrid Alternate Two Phases Particle Swarm Optimization (ATPPSO) algorithm is proposed by Changsheng Zhang et al. (2010) to solve the flow shop scheduling problem (FSSP) with the objective of minimizing makespan which combines the PSO with genetic operators and annealing strategy. In the ATPPSO algorithm, each particle contains two states, the attractive state and the repulsive state. In order to refrain from the shortcoming of premature convergence, a two point reversal crossover operator is defined and in the repulsive process, each particle is repelled away from some inferior solution in the current tabu list to fly towards some promising areas which can introduce some new information to guide the swarm searching process. To preserve the swarm diversity, an annealing criterion is used to update the personal best of each particle. Moreover an easy understanding makespan computation method based on matrix is designed. Finally, the proposed algorithm is tested on different scale benchmarks and compared with the recently proposed efficient algorithms. The result shows that, both the solution quality and the convergence speed of the ATPPSO algorithm precede the other two recently proposed algorithms. It can be used to solve large scale flow shop scheduling problem effectively.

The previous research on the job-shop scheduling problem has concentrated on finding a single objective optimization (e.g., makespan), even though the actual requirement of most production systems requires multi-objective optimization. The aim of D.Y. Sha et al. (2010) is to construct a particle swarm optimization (PSO) for an elaborate multi-objective job-shop scheduling problem. The original PSO was used to solve continuous optimization problems. Due to the discrete solution spaces of scheduling optimization problems, the authors modified the particle position representation, movement and its velocity in this study. The modified PSO was used to solve various benchmark problems. Test results demonstrated that the modified PSO performed better in search quality and efficiency than traditional evolutionary heuristics.

Rui Zhang et al. (2012) proposed a two-stage particle swarm optimization (PSO) algorithm for SJSSP with the objective of minimizing the expected total weighted tardiness. In the first-stage PSO, a performance estimate is used for quick evaluation of the solutions, and a local search procedure is embedded for accelerating the convergence to promising regions in the solution space. The second-stage PSO continues the search process, but applies a more accurate solution evaluation policy, i.e. the Monte Carlo simulation. In order to reduce the computational burden, the optimal computing budget allocation (OCBA) method is used in this stage.

8. CHARACTERISTICS OF PARTICLE SWARM OPTIMIZATION

An Inertia weight ω is a proportional agent that is related with the speed of last time, and the formula for the change of the speed is the following:

$$V_{id}^{k+1} = \omega\, v_{id}^{k} + C_1 \qquad (8)$$

The influence of the last speed on the current speed can be controlled by inertia weights. When ω values are higher, the PSO's searching ability for the global solutions are higher and if ω values are smaller, the searching ability remains partial. Generally, ω is equal to 1, but during several generations, there is a lack of searching ability. Experimental results show that PSO has the biggest speed of convergence when ω is between 0.8 and 1.2. While experimenting, ω is confined from 0.9 to 0.4 according to the linear decrease, which makes PSO search for the bigger space at the beginning and locate the position quickly where there is the most optimist solution. As ω is decreasing, the speed of the particle will also slow down to search for the delicate partial. The method quickens the speed of the convergence, and the function of the PSO is improved. When the problem that is to be solved is very complex, this method makes PSO's searching ability for the whole at the later period after several generation is not adequate, the most optimist solution cannot be found, so the inertia weights can be used to work out the problem. A PSO algorithm with convergence agents is introduced by Clerc M (1999) and the particle position and speed changes are

$$V_{id} = \chi \{V_{id} + C_1 \text{ rand } ( )*(P_{id}- X_{id})+C_2 \text{ rand}( )*(P_{gd}- X_{id})\} \qquad (9)$$

$$\text{Convergence factor } \phi = C_1+C_2>4 \qquad (10)$$

Generally, $\phi$is equal to 4.1, so χ is equal to 0.729. The experimental result shows compared with the particle swarm optimization algorithm with inertia weights, the convergence speed in the particle swarm optimization algorithm with the convergence agent is much quicker. In fact, when the proper ω, C_1 and C_2 is decided, the two calculation methods are identical. So, the particle swarm optimization algorithm with convergence agent can be regarded as a special example of the particle swarm optimization algorithm with inertia weights. PSO can be used for both the single objective problems and the multi objective problems.

8.1. Single objective problems

Some of the authors considers any one of the objective functions and optimized the objective function using PSO. They are

(i) **Flow time (F_j).**It is the amount of time job *j* spends in system.
(ii) **Makespan (C_{max}).**It is the total amount of time for all jobs to finish processing
(iii) **Lateness (L_j).** It is the amount of time by which the completion time differs from the due date. Positive lateness of a job means job is completed after the due date.
(iv) **Tardiness (T*j*).** It is the lateness of the job *j* if it fails to meet its due date, or zero. *i.e*T*j*= max {0, L*j*}

Shivasankaran et al.
Review on scheduling using particle swarm optimization,
Indian journal of engineering, 2013, 5(14), 75-80,

Table 1
Parameters of PSO in Scheduling Problems

Author	Application Area	Approach	Objectives	Parameters				
				Social Learning Factor C_1	Cognizant Learning Factor C_2	Inertia Weight ω	Max. No. of iterations	Swarm Particle Size
Xia, W. J et al. (2005)	Flexible Job shop scheduling	Hybrid PSO + SA	Minimizing the makespan, Critical Machine Workload and Total Workload of all Machines	2	2	0.4–1.2	50	100
Zhigang Lian et al. (2006)	Job shop scheduling	Similar PSO	Minimizing the makespan	2	2	N/A	1000	N/A
D. Y. Sha et al. (2006)	Job shop scheduling	Hybrid particle swarm optimization	Minimizing the makespan	0.5	0.3	0.5	1000	30
FAN Kun et al. (2007)	Job-Shop Scheduling	Binary particle swarm optimization (BPSO) algorithm	Shortest Completion time	2	2	0.9	20	10
Qun Niu et al. (2008)	Job shop scheduling	Particle swarm optimization combined with genetic operators (GPSO)	Minimizes the makespan and uncertainty of makespan	0.85	1.25	0, 0.5 and 1	2000	40
Deming Lei (2008)	Job shop scheduling	Pareto archive particle swarm optimization	Minimizing the makespan	2	2	0.5	2000	20
Changsheng Zhang, Jigui Sun (2009)	flow shop scheduling	alternate two phases particle swarm optimization algorithm	Minimizing the makespan	1	1	N/A	1000	60
Xinyu Shao et al. (2009)	Flexible Job shop scheduling	effective hybrid particle swarm optimization algorithm	Minimizing the makespan, Critical Machine Workload and Total Workload of all Machines	2	2	0.4–1.2	50	100
I-Hong Kuo et al. (2009)	Flow shop scheduling	Hybrid particle swarm optimization	Minimizing maximum completion time	2	0.4	0.4	10	50
D. Y. Sha et al. (2010)	Open Shop Scheduling	Modified PSO	Minimizing the makespan, Total flow time and Machine idle time	0.7	0.1	0.7	60	80
D. Y. Sha et al. (2010)	Open Shop Scheduling	Modified parameterized active schedule generation algorithm	Minimizing the makespan	0.7	0.1	0.9	20	60
Changsheng Zhang et al.(2010)	Flow shop scheduling	Hybrid alternate two phases particle swarm optimization	Minimizing the makespan	0.75	0.5	0.5	6000	60
Ghasem Moslehi et al. (2011)	Flexible Job shop scheduling	hybridization of the particle swarm and local search algorithm	Minimizing the makespan, Critical Machine Workload and Total Workload of all machines	2	2	0.4–0.9	100	20 – 50
D. Hajinejad et al. (2011)	Flow shop scheduling	Hybrid particle swarm optimization algorithm	Minimizing total flow time	0.4 – 2	0.4 – 2	0.4 – 2	100	10
Ching-Jong Liao et al. (2012)	Flow shop scheduling	Hybridizing PSO with bottleneck heuristic and SA	Minimizing the makespan	0.5	0.5	N/A	N/A	10 - 30
Rui Zhang et al. (2012)	Job shop scheduling	Two-stage hybrid particle swarm optimization algorithm	Expected total weighted tardiness	1.4944	1.4944	0.729	100	100
M. Bank et al. (2012)	Flow shop scheduling	Particle swarm optimization with & without Local Search	Minimizing total tardiness	2	2	0.4–0.9	100n	100
Ching-Jong Liao et al (2012)	Flow shop scheduling	Hybrid PSO with bottleneck heuristic	Minimizing the makespan	0.5	0.5	0.1	5, 10 and 25	10 – 30
T.C. Wong et al (2013)	Assembly job shop	Hybrid particle swarm optimization	Minimizing the makespan	0.5	0.5	0.95	20	N/A

8.2. Multi-objective problems

Few authors demonstrates the capabilities of solving multiple objective problems by considering more than one of the below mentioned objectives.

(i) **Flow time (F*j*).**It is the amount of time job *j* spends in system.
(ii) **Makespan(C_{max}).**It is the total amount of time for all jobs to finish processing
(iii) **Lateness (L*j*).** It is the amount of time by which the completion time differs from the due date. Positive lateness of a job means job is completed after the due date.
(iv) **Tardiness (T*j*).** It is the lateness of the job *j* if it fails to meet its due date, or sometimes it is zero. *i.e*T*j*= max {0, L*j*}
(v) **Work load (W_M).** It may be defined as the maximum load capacity that can be bear by a machine.

Shivasankaran et al.
Review on scheduling using particle swarm optimization,
Indian journal of engineering, 2013, 5(14), 75-80,

9. CONCLUSION

Like other evolutionary algorithms, PSO has become an important tool for optimization and other complex problem solving. It is an interesting and intelligent computational technique for finding global minima and maxima with high capability or multimodal functions and practical applications. Particle swarm optimization works on theory of cooperation and competition between particles. This paper explores the capabilities of PSO in handling NP hard scheduling problems which also states the characteristics of PSO Parameters and its impact on rapid convergence to the optimal solutions. PSO has only few parameters than other heuristic algorithms, where most of the literature utilizes the similar values for these parameters as shown in the Table 1. Yet PSO is a promising method working in the direction of simulation and optimization of difficult engineering and other problems. Further analysis of the comparative potency of PSO, and the problems in using a PSO based system are needed.

DISCLOSURE STATEMENT

There is no financial support for this research work from the funding agency.

ACKNOWLEDGMENT

Much thanks to my guide for his constructive criticism, and assistance towards the successful completion of this research work.

REFERENCES

1. Bank M, Fatemi Ghomi SMT, Jolai F, Behnamian J. Application of particle swarm optimization and simulated annealing algorithms in flow shop scheduling problem under linear deterioration, *Advances in Eng. Software* 2012, 47, 1–6
2. Changsheng Zhang, Jiaxu Ning, Dantong Ouyang, A hybrid alternate two phases particle swarm optimization algorithm for flow shop scheduling problem, *Comp.& Ind. Eng*. 2010, 58, 1–11
3. Changsheng Zhang, Jigui Sun. An alternate two phases particle swarm optimization algorithm for flow shop scheduling problem, *Expert Syst. with Appl*. 2009, 36, 5162–5167
4. Ching-Jong Liao, Evi Tjandradjaja, Tsui-Ping Chung, An approach using particle swarm optimization and bottleneck heuristic to solve hybrid flow shop scheduling problem, *App. Soft Comput.*12 (2012) 1755–1764
5. Clerc M. The swarm and the queen: towards a deterministic and adaptive particle swarm optimization, Proceedings of ICEC 1999, Washington, DC, 1999, pp.1951-1957
6. Deming Lei. A Pareto archive particle swarm optimization for multi-objective job shop scheduling, *Comp.& Ind. Eng*., 2008, 54, 960–971
7. Eberhart R, A new optimizer using particle swarm theory, Sixth Int. Symp. on Micro Machine and Human Science, IEEE, 1995, pp.39-44
8. Eberhart RC, Shi Y. Particle swarm optimization: developments application and resources, Proceedings of IEEE Int. Conf. on Systems, 2001, pp. 68-73
9. FAN Kun, ZHANG Ren-qian, XIA Guo-ping, Solving a Class of Job-Shop Scheduling Problem based on Improved BPSO Algorithm, Systems Eng. - Theory & Practice, 27 – 11, 2007, 111 - 117
10. Ghasem Moslehi, Mehdi Mahnam, A Pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization and local search, *Int. J. Prod. Economics* 2011, 129, 14–22
11. Guohui Zhang, Xinyu Shao, Peigen Li, Liang Gao. An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem, Comp.& Ind. Eng., 2009, 56, 1309–1318
12. Hajinejad D, Salmasi N, Mokhtari R. A fast hybrid particle swarm optimization algorithm for flow shop sequence dependent group scheduling problem, Scientia Iranica Transactions E: *Ind. Eng.* 2011, 18(3), 759–764
13. Hassan R, Cohanim B, Weck OD. A Comparison of Particle Swarm Optimization and Genetic Algorithm, Evolutionary Programming VII. Springer Berlin Heidelberg 2004
14. Hou ZX, Wiener. Model identification based on adaptive particle swarm optimization, in: IEEE Proceedings of Seventh Int. Conf. on MachineLearning and Cybernetics, Kumming 12-15th July, 2008, pp. 1041-1045
15. I-Hong Kuo, Shi-Jinn Horng, Tzong-Wann Kao, Tsung-Lieh Lin, Cheng-Ling Lee, Takao Terano, Yi Pan, An efficient flow-shop scheduling algorithm based on a hybrid particle swarm optimization model, *Expert Syst. with Appl*. 2009, 36, 7027–7032
16. Jie X, Deyun X. New Metropolis coefficients of particle swarm optimization, In Control and Decision Conference, 2008. CCDC 2008. Chinese, pp. 3518-3521. IEEE, 2008
17. Kennedy J, Eberhart R. Particle swarm optimization, Proceedings of the 1995 IEEE Int. Conf. on Neural Networks,1995, 4, 1942-1948
18. Qun Niu, Bin Jiao, XingshengGu. Particle swarm optimization combined with genetic operators for job shop scheduling problem with fuzzy processing time, *Appl. Mathematics and Comput*. 205 (2008) 148–158
19. Rui Zhang, Shiji Song, Cheng Wu, A two-stage hybrid particle swarm optimization algorithm for the stochastic job shop scheduling problem, *Knowledge-Based Syst*. 2012, 27, 393–406
20. Sha DY, Cheng-Yu Hsu. A new particle swarm optimization for the open shop scheduling problem, *Comp.& Oper. Res*. 2008, 35, 3243–3261
21. Sha DY, Hsing-Hung Lin, Hsu C-Y. A Modified Particle Swarm Optimization for Multi-objective Open Shop Scheduling, Proceedings of the Int. Multi-conf.of Engineers and Computer Scientists (IMECS 2010), March 17 -19, 2010, Hong Kong
22. Sha DY, Hsu C-Y. A hybrid particle swarm optimization for job shop scheduling problem, *Comp.& Ind. Eng*., 2006, 51, 791–808
23. Shi Y, Eberhart RC. Empirical study of particle swarm optimization. In Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on, vol. 3. IEEE, 1999
24. Shi Y, Eberhart RC. Parameter selection in particle swarm optimization, In Evolutionary Programming VII, pp. 591-600. Springer Berlin Heidelberg, 1998
25. Wei J, Guangbin L, Dong L. Elite particle swarm optimization with mutation, In System Simulation and Scientific Computing, 2008. ICSC 2008. Int. Conf. on Asia Simulation Conference-7th, pp. 800-803. IEEE, 2008
26. Wong TC, Ngan SC. A comparison of hybrid genetic algorithm and hybrid particle swarm optimization to minimize makespan for assembly job shop, App. Soft Comput. 13 (2013) 1391–1399
27. Xia WJ, Wu ZM. An effective hybrid optimization approach for Multiobjective flexible job-shop scheduling problems. *Comp.& Ind. Eng*., 2005, 48-2, 409–425
28. Yasada K, Iwasaki N. Adaptive particle swarm optimization using velocity information of swarm, IEEE Int. Conf. on Systems, Man and Cybernetics, 2004, pp. 3475-3481
29. Zhang X, Wen S, Li H. A novel particle swarm optimization with self-adaptiveinertia weight, in: Proceedings of 24th Chinese Control Conference, Guangzhou,P.R. China, 2005, pp. 1373-1376
30. Zhigang Lian, Bin Jiao, Xingsheng Gu. A similar particle swarm optimization algorithm for job-shop scheduling to minimize makespan. *Appl. Mathematics and Comput*. 2006, 183, 1008–1017

Shivasankaran et al.
Review on scheduling using particle swarm optimization,
Indian journal of engineering, 2013, 5(14), 75-80,