IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 2, Issue 7, February 2014)

A Novel Approach for Job Shop Scheduling using Particle Swarm Optimization

Shailendra Patidar
shailupatidar13@gmail.com

Narendra Kushwah
narendra_jit_mech07@yahoo.co.in

Abhishek Yadav
ydvabhishek@gmail.com

Abstract—**The Job shop Scheduling problem (JSP) is a well-known huge combinatorial problem from the field of Deterministic scheduling. It is considered the one of the hardest in the class of NP-hard problems. The objective of this paper is to optimize the schedule to a finite number of operations using PSO.**

Keywords-**Job shop Scheduling problem, PSO.**

I. INTRODUCTION

The job shop scheduling problem (French 1982) can be briefly described as follows. There are a set of jobs and a set of machines. Each job consists of a sequence of operations, each of which uses one of the machines for a fixed duration. Once started, the operation cannot be interrupted. Each machine can process at most one operation at a time. A schedule is an assignment of operations to time intervals on the machines. The problem is to find a schedule of minimal time to complete all jobs.

Job shop scheduling problem

Scheduling is the allocation of shared resources over time to competing activities. It has been the subject of a significant amount of literature in the operations research field. Emphasis has been on investigating machine scheduling problems where jobs represent activities and machines represent resources; each machine can process at most one job at a time.

Table.1: A 3 × 3 problem

Job	Operations routing (processing time)		
1	1(3)	2(3)	3(3)
2	1(2)	3(3)	2(4)
3	2(3)	1(2)	3(1)

The $n \times m$ minimum-makespan general job-shop scheduling problem, hereafterreferred to as the JSSP, can be described by a set of n jobs $\{J_i\}_{i\leq j\leq n}$which is to be processed on a set of m machines $\{M_r\}_{i\leq r\leq m}$. Each job has a technologicalsequence of machines to be processed. The processing of job J_i on machine $M_r$is called the operation O_{jr}. Operation O_{jr} requires the exclusive use of M_r for anuninterrupted duration p_{jr}, its processing time. A schedule is a set of completiontimes for each operation $\{c_{jr}\}_{i\leq j\leq n,1\leq r\leq m}$ that satisfies those constraints. The timerequired to complete all the jobs is called the makespan L. The objective whensolving or optimizing this general problem is to determine the schedule which minimizes L. An example of a 3 × 3 JSSP is given in Table 1. The data includes the routing of each job through each machine and the processing time for each operation (in parentheses).

The Gantt-Chart is a convenient way of visually representing a solution of the JSSP. An example of a solution for the 3 × 3 problem in Table 1 is given in Figure 1.

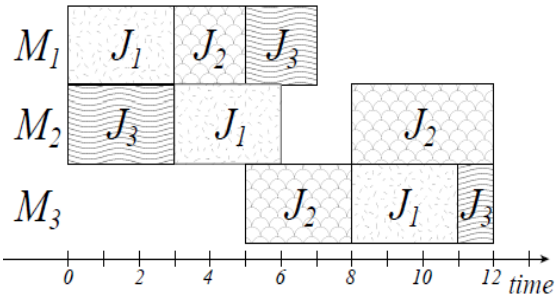

Figure.1: A Gantt-Chart representation of a solution for a 3 × 3problem

The JSSP is not only NP-hard, but it is one of the worst members in the class. An indication of this is given by the fact that one 10 × 10 problem formulated byMuth and Thompson [1] remained unsolved for over 20 years.Besides exhaustive search algorithms based on branch and bound methods, several approximation algorithms have been developed. The most popular ones in practice

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 2, Issue 7, February 2014)

are based on priority rules and active schedule generation [2]. A more sophisticated method called shifting bottleneck (SB) has been shown to be very successful [3]. Additionally, stochastic approaches such as simulated annealing (SA), tabu search [4, 5] and genetic algorithms (GAs) have been recently applied with good success.

The job shop scheduling problem is considered as a particularly hard combinatorial optimization problem (Lawler et al. 1982). Since it has practical applications, the problem has been studied by many authors, and several optimization algorithms and approximation algorithms have been proposed.

Particle Swarm Optimization

As of yet, Particle Swarm Optimization has not been applied to the traditional Job Shop Problem, with one exception. In 2004 Weijun, et al. [6] applied a hybrid Simulated Annealing/PSO to the traditional JSP. In their study, Simulated Annealing (SA) was used to fine tune solutions found by the PSO algorithm. The results of this hybrid algorithm were very promising. Many of the most widely used Job Shop test bench problems were solved to optimal or best known solutions. This means that the implementation of a pure PSO algorithm might very well be an efficient and effective way of solving these types of huge combinatorial problems without using Simulated Annealing for fine tuning.

However, there is slightly more literature published on the application of the PSO algorithm to some of the other scheduling problems, such as the Permutation Flowshop Problem (PFSP) also called the Flow-shop Scheduling Problem (FSSP), or the Single Machine Weighted Tardiness Problem (SMTWT). Tasgetiren, et al. [7] in 2004 applied PSO to the Single Machine Weighted Tardiness problem. They developed the Smallest Value Position Rule (SVP) in order to transform the continuous space of the PSO to the permutation space used to represent a solution of the SMTWT problem. Tasgetiren also published a paper on the PSO applied to the Permutation Flow shop Sequencing Problem, along with Liang, Sevkli, and Gencyilmaz [8]. The same SPV rule was used for the necessary space transformation from continuous to permutation space. This space transformation concept was used for the Traveling Salesman Problem by Pang, et al. [9] in 2004. Their method of space transformation was called the GVP rule, or Greatest Value Priority. Using PSO and local search techniques they were able to solve medium scale (50 – 75 city) TSP Problems. Particle Swarm Optimization has also been applied to the Flexible Job Shop Problem (FJSP) by Xia and Wu [10] recently in 2005.

II. METHODOLOGY

- **Particle Swarm Algorithm**
  1. Begin
  2. Factor settings and swarm initialization
  3. Evaluation
  4. g = 1
  5. While (the stopping criterion is not met) do
  6. for each particle
  7. Update velocity
  8. revise place and localized best place
  9. Evaluation
  10. End For
  11. Update leader (global best particle)
  12. g + +
  13. End While
  14. End

The PSO procedure has various phases consist of Initialization, Evaluation, Update Velocity and Update Position.

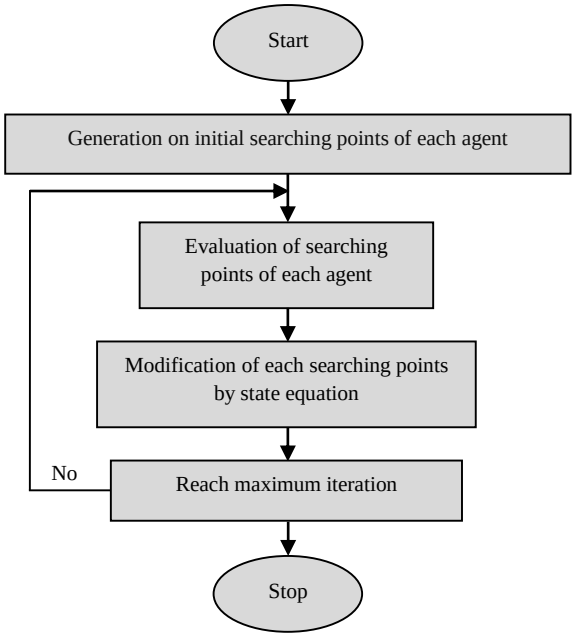

Figure.2: Flow chart of PSO

IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 2, Issue 7, February 2014)

- **Initialization**

The initialization phase is used to determine the position of the m particles. The random initialization is one of the most popular methods for this job. There is no assurance that a randomly created particle be a better answer and this will make the initialization more attractive.

A good initialization algorithm makes the optimization algorithm more efficient and reliable. For initialization, initial information or knowledge of the problem can help the algorithm to converge in less iterations.

- **Update velocity and position**

In each iteration, each particle updates its velocity and position according to its heretofore best position, its current velocity and some information of its neighbour. Equation below is used for updating the velocity:

$$v_1(t) = \underbrace{w\overline{v_l(t-1)}}_{inertia} + \underbrace{c_1 r_1\left(x_1^{\#}(t-1) - \overline{x_l(t-1)}\right)}_{Personalinfluence} + \underbrace{c_2 r_2\left(x^{*}(t-1) - \overline{x_l(t-1)}\right)}_{Socialinfluence}$$

Where

$\overline{x_l(t)}$ = The position-vector in iteration t

i = The index of the particle

$\overline{v_l(t)}$ = The velocity- vector in iteration t

$x_1^{\#}(t)$ = The position so for of particle i in iteration t and its j[th] dimensional value is $x_{ij}^{\#}(t)$.

The best position vector among the swarm here to force is then stored in a vector $x^{*}(t)$ and its j[th] dimensional value is $x_{ij}^{*}(t)$.

r_1 , r_2 = random numbers in the interval [0, 1].

c_1, c_2 = positive constants and

w is called the inertia factor.

$$\overline{x_l(t)} = \overline{x_l(t-1)} + \overline{v_l(t)}$$

Each of the three terms of the velocity update equation have different roles in the PSO algorithm. This process is repeated until some stopping condition is met. Some common stopping conditions include: a pre-set number of iterations of the PSO algorithm, a number of iterations since the last update of the global best candidate solution, or a predefined target fitness value.

III. SIMULATION ANDRESULTS

Table.1: Job shop scheduling optimization using CPSO problem description: 6 machine/ 6 jobs order of operation:

J1	3	1	2	4	6	5
J2	2	3	5	6	1	4
J3	3	4	6	1	2	5
J4	2	1	3	4	5	6
J5	3	2	5	6	1	4
J6	2	4	6	1	5	3

Optimized Total Time: 59 Mins

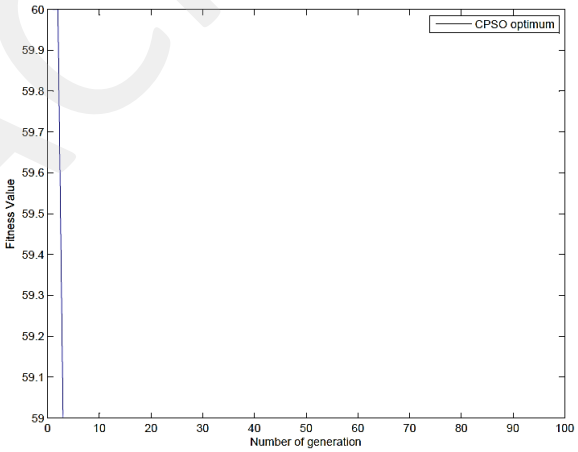

Figure.3: Convergence graph of PSO for 6 machine/ 6 jobs

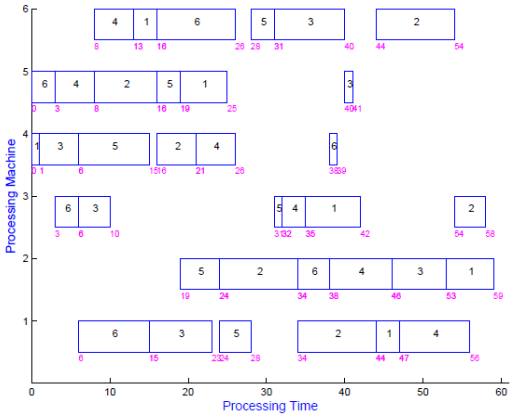

Figure.4: Gantt graph of PSO for 6 machine/ 6 jobs

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary research
Website: www.ijdacr.com (Volume 2, Issue 7, February 2014)

Table.2: Job shop scheduling optimization using CPSO Problem description: 10 machine/ 10 jobs Order of operation:

J1	1	2	3	4	5	6	7	8	9	10
J2	1	3	5	10	4	2	7	6	8	9
J3	2	1	4	3	9	6	8	7	10	5
J4	2	3	1	5	7	9	8	4	10	6
J5	3	1	2	6	4	5	9	8	10	7
J6	3	2	6	4	9	10	1	7	5	8
J7	2	1	4	3	7	6	10	9	8	5
J8	3	1	2	6	5	7	9	10	8	4
J9	1	2	4	6	3	10	7	8	5	9
J10	2	1	3	7	9	10	6	4	5	8

Optimized Total Time: 967 Mins

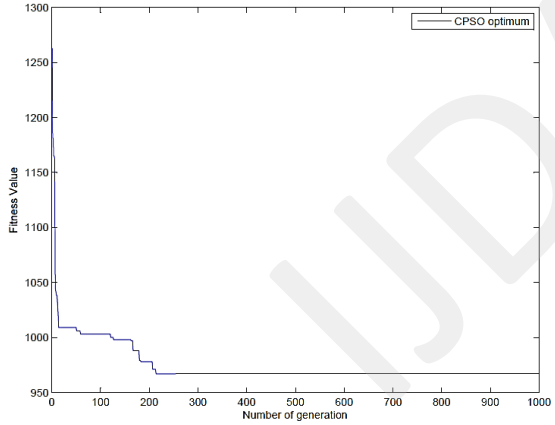

Figure.5: Convergence graph of PSO for 10 machine/ 10 jobs

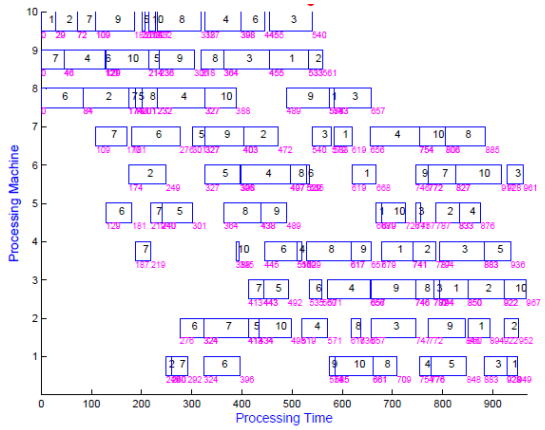

Figure.6: Gantt graph of PSO for 10 machine/ 10 jobs

IV. CONCLUSION

This paper has presented job shop scheduling problem solved using PSO algorithm. In this paper 6 job 6 machine and 10 job 10 machine problems are presented, PSO gives the optimum time for scheduling which is better than GA as given in literature. Further improvement can be done using hybrid optimization algorithms.

V. REFERENCES

[1]. J.F. Muth and G.L. Thompson. Industrial Scheduling. Prentice-Hall, Englewood Cliffs, N.J., 1963.

[2]. S. S. Panwalkar and Wafix Iskander. A survey of scheduling rules. Oper. Res., 25(1):45–61, 1977.

[3]. J. Adams, E. Balas, and D. Zawack. The shifting bottleneck procedure for job shop scheduling. Mgmt. Sci., 34(3):391–401, 1988.

[4]. M. Dell'Amico and M. Trubian. Applying tabu search to the job-shop scheduling problem. Annals of Operations Research, 41:231–252, 1993.

[5]. T. Yamada and R. Nakano. Job-Shop Scheduling by Simulated Annealing Combined with Deterministic Local Search. Kluwer academic publishers, MA, USA, 1996.

[6]. Weijun X., Zhiminig W., Wei Z. and Genke Y., "A New Hybrid Optimization algorithm for the job shop scheduling problem" proceeding of the 204 american control conference. Boston, MA, 2004.

[7]. Tasgetiren M. F., Sevkli M., Yun-chia Liang, "Particle Swarm Optimization Algorithm for t Single machine Total Weighted Tardiness Problem", Congress on Evolutionary Computation, Portland, 2004.

[8]. Tasgetiren M. F., Sevkli M., Yun-chia Liang, Gencyilmaz G., "Particle Swarm Optimization Algorithm for Makespan and Maximum Lateness Minimization in permutation Flowshop problem" 4-th International Symposium on Intelligent manufacture systems Sakarya, Turkey, 2004.

[9]. Pang W., Wang K., Zhou C., Dong L., Liu M., Zhang H. "Modified Particle Swarm Optimization based on Space Transformation for Solving Traveling Salesman Problem", Shanghai, 2004.

[10]. Xia W., Wu Z., "An Effective Hybrid Optimization Approach for Multi objective Flexible job shop scheduling problems", computer and industrial Engineering 2005.