ISSN : 1978-774X

PARTICLE SWARM OPTIMIZATION BASED ON BOTTLENECK MACHINE FOR JOBSHOP SCHEDULING

Rahmi Maulidya

Industrial Engineering Department, Industrial Engineering Faculty, Trisakti University
rmauli@yahoo.com

ABSTRACT

In job shop production system, bottleneck machine is treated as a center of production planning so it is important to maximize the production capacity. Bottleneck scheduling problem divided solution into forward and backward scheduling. The completion time of the job until bottleneck machine is a due date for all job and the completion time of the job after bottleneck machine is the real completion time for all job. In most scheduling problem, solution using Particle Swarm Optimization is only to minimize makespan in forward scheduling or minimize the number of tardy job in backward scheduling. This research present the methodology to solve job shop scheduling problem based on bottleneck machine using Particle Swarm Optimization.

Keywords *: scheduling, job shop, bottleneck, Particle Swarm Optimization, due date, makespan*

1. INTRODUCTION

Job shop scheduling is characterized by the routing of each products. n job are processed to completion on m unrelated machines. The routing and processing time is known in advance. Processing time are fixed. Machine is available from time zero and operations are processed without preemption. Job shop scheduling is strongly NP-hard and solve using metaheuristic algorithm. Genetic algorithm, simulated annealing algorithm, ant colony algorithm and particle swarm optimization is clasified into metaheuristic.

Particle swarm optimization (PSO) is one of the latest evolutionary optimization methods inspired by nature that includes evolutionary strategy (ES), evolutionary programming (EP), genetic algorithm (GA), and genetic programming (GP). PSO is based on the metaphor of social interaction and communication such as bird flocking and fish schooling. PSO is distinctly different from other evolutionary-type methods in a way that it does not use the filtering operation (such as crossover and/or mutation), and the members of the entire population are maintained through the search procedure so that information is socially shared among individuals to direct the search towards the best position in the search space (Tasgetiren, 2007).

Job shop scheduling problem can also be solved by using bottleneck scheduling. Bottleneck machine is defined using the longest processing time of machine. In botlleneck scheduling, there is forward scheduling and backward scheduling, backward scheduling is used until bottleneck machine and forward scheduling is used after bottleneck machine to minimize makespan. PSO mostly use to minimize makepan in forward scheduling. Although there is PSO use to minimize due date, but there is no PSO use in both side.

This paper present the methodology to solve job shop scheduling problem using Particle swarm optimization based on bottleneck machine.

2. LITERATURE

2.1. Particle Swarm Optimization

Particle Swarm Optimization (PSO) in introduced as a method for nonlinear optimization. This method uses a model of flight of birds (or motion of particles) to solve the optimization problem in which any potential solution in search space is considered as a potential position for particles. Swarm of particles move through search space under a defined dynamics of flight and find the best solution as the optimum solution. The position of i-th particle

ISSN : 1978-774X

is presented by an n-dimensional vector for an n-dimensional search space (Rezaie, 2010).

Algorithm of PSO method has steps below (Rezaie, 2010):

1) Problem definition: This step includes the definition of cost function and determining the range of search space. The minimum and maximum values which the particles could take as coordinates of their position should be determined.
2) Initialization: Initialize positions and velocities of particles randomly. Also, best position of each particle is set to be its initial position and global best position is set to be the best initial position associated with cost function.
3) Particles motion: Find new position and velocities of particles using equations 3 and 4.
4) Evaluation: Evaluate cost function for each particle position and update best positions of particles and best global position.
5) Repeat: Go to step 3 again and repeat till a stopping criterion is satisfied. The stopping criterion could be considered as taking a predefined number of iterations.

2.2. Notation

Zij : random number n in particle i and job *j*
U_{ij} : uniform number in particle i and job *j*
X_i^t : particle i in swarm at iteration *t*
$x_{i,j}^t$: value position for particle *i* and dimesion *j* where (*j* = 1,2,…,n) and n is the number of dimension. So, initial iteration is noted as $X_i^0 = \left[x_{i1}^0, x_{i2,\dots,}^0 x_{1n}^0\right]$
v_{ij}^t : velocity for particle i and job *j* at iteration *t*
$\pi_{i,j}^t$: the number of job j for particle I in permutation at iteration t
NP : the number of particle where the amount is twice of number of dimension (job)
f_i^t : makespan of fitness value in particle *i* and iteration t
f^{pb} : Fuction of the best makespan in personal best
f^{gb} : Function of the best makespan in global best
P_i^t : Personal Best at iteration *t* and particle i
p_{ij}^t : Personal Best at particle i, job *j* and iteration *t*
G^t : Global Best for iteration *t*
g_{ij}^t : Global Best for particle i and job *j* at iteration *t*
w^t : Inertia Weight at iteration *t*
c_1, c_2 : coefisien acceleration where c_1, c_2 = 2
r_1, r_2 : uniform random number from 0 until 1 where r_1 is uniform number for particle position and r_2 is uniform number for particle velocity. r_1 and r_2 are uniform number from U_{ij}
T_i^t : Tardy for particle *i* and iteration *t*

The steps for PSO are :

Step 1 *Initilization*

a. Generate random number r1 using LCG

$$Z_i = (a\ Z_{i-1} + b)(\text{mod } m) \quad (1)$$

$$U_i = Z_i/m \quad (2)$$

a. Set t = 0, and NP = twice number of dimension
b. Define random particle as $\{\ X_i^0, i = 1,2,\dots, NP\ \}$ where $X_i^0 = \left[x_{i1}^0, x_{i2,\dots,}^0 x_{1n}^0\right]$ using equation

$$x_{ij}^0 = x_{\min} + (x_{\max} - x_{\min}) * r_1 \quad (3)$$

where x_{min} = 0.0 , x_{max} = 4.0 and r_1 is uniform random number between 0 and 1
c. Define initial velocity for each particle i randomly

$$v_{ij}^0 = v_{\min} + (v_{\max} - v_{\min}) * r_2 \quad (4)$$

where v_{min} = -4 and v_{max} = 4, and r_2 is define from uniform random number between 1 and 0.
d. Generate permutation using Smallest Position Value (SPV) for particle X_1^0

$$\pi_1^0 = \left[\pi_{11}^0, \pi_{12,\dots,}^0 x_{1n}^0\right] \quad (5)$$

e. Evaluate every particle i in swarm to look for makespan of every particle notated as function f_1^0 where i = 1,2,...,NP
f. For every particle in swarm, set

$$P_i^0 = X_i^0; P_i^0 = \left\lfloor p_{i1}^0, x_{i1}^0 = x_{i2,\dots,}^0 p_{in}^0 = x_{in}^0 \right\rfloor \quad (6)$$

g. Define the best fitness value which means the best makespan from every particle in swarm. Makespan is notated with:

$$f_i = \min \; f_i^{\,0}, i = 1,2,..., NP \quad (7)$$

h. Set global best or Gbest

$G_i^0 = X_i^0; G^0 = [g_{i,1} = x_{i,1} = x_{i,2}, \ldots, g_{i,n} = x_{i,n}]$ (8)

i is particle. G_i^0 is value of Global Best at initial iteration untill Global Best equal to the value of particle position.

Step 2 Update the iteration

$$t = t+1 \quad (9)$$

Step 3 Update Inersia weight

Update inersia weight with

$$w^t = w^{t-1} * \beta \quad (10)$$

where β is decreament factor with value 0.975 and w^0=0.9

Step 4 Update Velocity

$$v_{ij}^t = w^{t-1} v_{ij}^{t-1} + c_1 r_1 (p_{ij}^{t-1} - x_{ij}^{t-1}) + c_2 r_2 (g_i^{t-1} - x_{ij}^{t-1}) \quad (11)$$

Where c is coeffisien corelation. For p_{ij}^{t-1} and g_{ij}^{t-1} suite to previous iteration.

Step 5 Update Position

$$x_{ij}^t = x_{ij}^{t-1} + v_{ij}^t \quad (12)$$

Step 6 Define Permutation

Using SPV to define new permutation:

$$\pi_{i1}^t = [\pi_{i1}^t, \pi_{i2,....,}^t x_{in}^t] \quad (13)$$

Step 7 Update Personal Best

Each particle is evaluated using permutation. Permutation can be renew if

$$f_i^{\,t} \le f_i^{\,pb}, i = 1,2,..., p \quad (14)$$

so personal best will change into $P_i^t = X_i^t$ and $f_i^{\,pb} = f_i^{\,t}$, which means if makespan in the next iteration is smaller atu equal to makespan at iteration t_{-1}, so f^{gb} is changed into the value in particle at the best iteration, so that for personal best.

Step 8 Update Global Best

Find minimum value of Global Best

$$f_l^{\,t} = \min\{f_i^{\,pb}\} \quad (15)$$

i = 1,2,...,NP

it means the best makespan of all particle, so global best is renewed into $G^t = X_l^t$ and $f^{\,gb} = f_l^{\,t}$.

Step 9 Stopping Criterion

If iteration exceed maximum number of iteration, or exceed maximum time, the iteration will be stop.

3. METHODOLOGY

Particle Swarm Optimization start with generating random number to define the number of particle, the potition of particle, and particle velocity. The number of particle is set two times from the number of job. Permutation ranking is set using SPV. Job is schedule using permutation ranking.

In this research, job is scheduled based on the position of bottleneck machine. For position before bottleneck machine, define particle due date and generate sequence using backward scheduling. For position after bottleneck machine, sequence is defined using forward scheduling. Both result can define personal best position with set of $P^0_i = X^0_i$ for each particle. Next step is the same for PSO algorithm which define Global Best $G^0_i = X^0_i$.

The sequence is set based on global best and renew at every iteration untill smalest or constant global best. In every renew iteration, it is followed by the renew of weight.

4. NUMERICAL EXAMPLE

In this paper, the data is set in table 1. The processing time and the routing is fixed and is known in advance.

Table 1. Data set

JOB	Processing time			Routing		
	Operation			Operation		
	1	2	3	1	2	3
J1	5	10	12	A	B	C
J2	4	3	8	A	C	B
J3	9	6	7	C	B	A
J4	7	5	11	B	C	A

The total processing time of each machines show that machine B is the longest. Machine B is also a bottleneck machine. Machine B will be scheduled for the first time.

Table 2. Total processing time of each machine

JOB	machine			Total
	A	B	C	
1	5	10	12	27
2	4	8	3	15
3	7	6	9	22
4	11	7	5	23
Total	27	31	29	

Now, focus is on machine B as bottleneck machine,

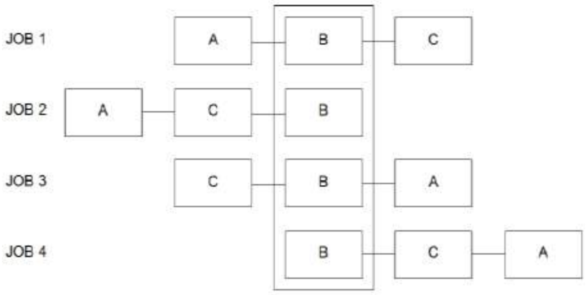

Figure 1. routing product

Random number is generate using this data:
a : multiplier = 23
b : increment factor = 0
m : modulus = 100000001
Z_0 : 47594118

Random number will result r_1 dan r_2, then it is used to define permutation in table 3.

Table 3. Sequence Partikel $X_1^0 – X_8^0$

Particle	Sequence			
X_1^0	J2	J3	J4	J1
X_2^0	J1	J3	J2	J4
X_3^0	J4	J1	J2	J3
X_4^0	J4	J2	J3	J1
X_5^0	J3	J2	J4	J1
X_6^0	J1	J2	J4	J3
X_7^0	J1	J3	J4	J2
X_8^0	J3	J1	J2	J4

Each permutation in t=0 is observed to get makespan for initial. Due date is set until machine bottleneck. Due date is the longest time untill machine bottleneck. Particle which give the best result is J4-J1-J3-J2. Gantt chart for the sequence is in figure 2.

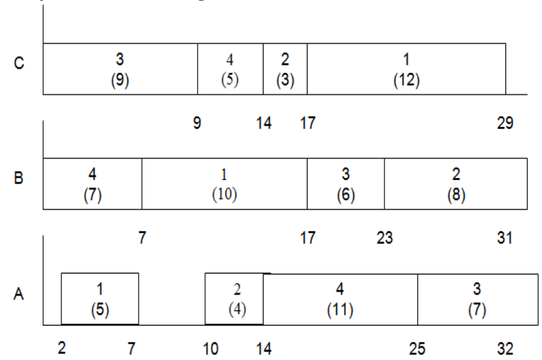

Figure 2. Gantt chart

Table 4. Result

	Makespan	Flowtime	Idle Time
PSO proposed	32	19,5	A = 5 B = 1 C = 3
Bottleneck scheduling	32	19,5	A = 5 B = 1 C = 3
Aktive schedule	38	19,5	A = 11 B = 7 C = 9

5. CONCLUSION

The PSO based on bottleneck machine is effective for large number of job. Testing the PSO into flowhop scheduling is the next research and define the bottleneck machine using theory of constraint.

6. REFERENCES

(a) Baker, Kenneth R. (1974). *Introduction to Sequencing and Scheduling.* John Wiley & Sons, Inc.Canada
(b) Brucker, Peter. (2006). *Scheduling Algorithms*: Springer 5th edition. German
(c) Clerc, M. (2005). *Particle Swarm Optimization*. Lavosier Iste. London
(d) Divani, Vania. (2012). Usulan Penjadwalan Job Dengan Metode *Particle Swarm Optimization* Untuk Meminimasi *Number Of Tardy* Jobs Di PT. Faco Global Engineering. Universitas Trisakti. Jakarta
(e) Law, A.M dan Kelton, A.W. (2000). *Simulation Modeling and Analysis*: MGH 3rd Edition. New Jersey
(f) Morton, T. and Pentico, D. (1993). *Heuristic scheduling system*. John Wiley Interscience
(g) Pinedo, Michael. (2002). *Scheduling: Theory, Algorithms, and Systems*. New Jersey: Prentice Hall Inc. 2nd Edition
(h) Rezaie, K., Nazari, S. S., Nazari-Shirkouhi Salman, dan Ghodsi Reza. (2010). "Theory of Constraints and Particle Swarm Optimization Approaches for Product Mix". Australian Journal of Basic and Applied Sciences, ISSN 1991-8178.
(i) Tasgetiren, M. F., Liang, Y.-C., Sevkli, M., dan Gencyilmaz, G. (2007). "A Particle Swarm Optimization Algorithm for Makespan and Total Flowtime Minimization in Permutation Flowshop Sequencing Problem". *European Journal*

ISSN : 1978-774X

of Operational Research, Vol 177, 1930 – 1947.

AUTHOR BIOGRAPHIES

Rahmi Maulidya is a lecturer in Department of Industrial Engineering, Faculty of Industrial Technology, Trisakti University, Jakarta. She received her Master of Industrial Engineering from Institut Technology Bandung in 2005. Her research interests are in the area of Production Planning & Control, Robotic and Design Product. She is a member of the Production System Laboratory. Her email address is <rmauli@yahoo.com>