Discrete particle swarm optimization to solve multi-objective limited-wait hybrid flow shop scheduling problem

B Santosa[1*] , N Siswanto[1] and Fiqihesa[1]
[1] Industrial Engineering Department, Institut Teknologi Sepuluh Nopember Kampus ITS Sukolilo-Surabaya 60111, Indonesia

*budi_s@ie.its.ac.id

Abstract. This paper proposes a discrete Particle Swam Optimization (PSO) to solve limited-wait hybrid flowshop scheduing problem with multi objectives. Flow shop schedulimg represents the condition when several machines are arranged in series and each job must be processed at each machine with same sequence. The objective functions are minimizing completion time (makespan), total tardiness time, and total machine idle time. Flow shop scheduling model always grows to cope with the real production system accurately. Since flow shop scheduling is a NP-Hard problem then the most suitable method to solve is metaheuristics. One of metaheuristics algorithm is Particle Swarm Optimization (PSO), an algorithm which is based on the behavior of a swarm. Originally, PSO was intended to solve continuous optimization problems. Since flow shop scheduling is a discrete optimization problem, then, we need to modify PSO to fit the problem. The modification is done by using probability transition matrix mechanism. While to handle multi objectives problem, we use Pareto Optimal (MPSO). The results of MPSO is better than the PSO because the MPSO solution set produced higher probability to find the optimal solution. Besides the MPSO solution set is closer to the optimal solution

1. Introduction

Flow shop is one of scheduling problems where the sequence is the same for all jobs. The original flow shop scheduling problem is a scheduling problem at a single production line (each process only has a machine). While the problem that occurs frequently in the real world is scheduling problem at a parallel production line (there is a process with more than one machine). That problem is one of flow shop scheduling problem development that is called hybrid flow shop scheduling problem.

Flow shop scheduling model is not only grouped based on the number of production line, but also based on the characteristic of the product. There are some type of products such as a product that can wait for being processed at the next machine and a product that cannot wait. The basic model of flow shop scheduling problem does not consider the characteristic of the product (can wait or not). Therefore, that basic model is appropriate for the product that can wait for a long time (unlimited waiting time). While, the model for product that cannot wait is called no-wait flow shop scheduling problem [1]. Besides, there is a product that can wait for being processed at the next operation with limited waiting time, such as casting process that must be done before the material getting hardened. When using no-wait flow shop scheduling model, the process are forced without idle time. The production sequence can be different if there is a little idle time in the process. The model that is suitable for this characteristic is called limited wait flow shop scheduling problem. There are not many research carried out in this area.

The main purpose of scheduling model development is finding the optimal production sequence. The optimal production sequence can be seen from some performance criteria, such as production time, tardiness time, and machine utility. Production time is minimized based on total time needed, without considering the job deadline. Tardiness time is minimized based on the positive difference between the job deadline and the job finishing time, without consider the total production time needed. Then, machine utility is maximized based on the minimum total idle time, without considering the total production time and tardiness time. There is a possibility for an objective function is not optimal when the other objective functions are optimized. Therefore, there is a trade-off when there are three objective functions. When we have trade-off among the criteria then we will have a multi-objective problem.

Scheduling problem for n-job m-process at parallel line production is a combinatorial problem that classified into Non Polynomial-hard (NP-Hard) problem. The best method used to solve NP-Hard Problem is heuristic [2] . There were many research that using metaheuristic method to solve many type of flow shop scheduling problems, single objective and multi-objective problems. For example, hybrid flow shop scheduling problem solved by exact method, heuristic, and metaheuristics [3]. In this paper, it does not consider limited wait constraint. There is a few research that consider limited wait constraint, i.e. [4] , [5] and , Su (2003) and [6].

Particle Swarm Optimization (PSO) is easy to use and also efficient [7]. Originally, PSO was developed to solve continuous optimization problems. Since scheduling is a discrete problem, we need to modify PSO to fit the problem. The modifications are done by using transition probability matrix. As for handling the multiobjective, we use pareto optimal. The rest of the paper is organized as follows. Section 2 presents the problem definition, and Section 3 describes the literature review. Section 4 discusses the mathematical model of the problem. In section 5, we explain the proposed algorithm, MPSO. In section 6, we describe the experiments, results and analysis. Section 7 concludes the paper.

2. Problem Definition

This paper focuses on solving the limited wait hybrid flow shop scheduling (LWHFS) by using Modified Particle Swarm Optimization. The objectives used in this paper are minimizing total production time (makespan), minimizing tardiness time (total tardiness), and maximizing machine utility (minimizing total machine idle time). Since the original PSO is to solve continuous optimizaton problem, here we modified PSO to fit the discrete problem and also develop it to handle the multi-objectve case.

3. Literature Review

3.1 Hybrid Flow Shop

Hybrid Flow Shop Scheduling (HFS) problem is one of Flow Shop Scheduling problem extension. There are many types of HFS problems based on the characteristic of the problem. Generally HFS problems have following characteristics:

- The number of process equals 2 or more.
- The number of machine at each process equals 1 or more and the number of process that have more than 1 machine is minimum one.
- Each job processed through the same sequence, but there is a possibility that some jobs are not processed at some operations.

Research on flow shop scheduling problem usually imposes an assumption that each job is processed at all of operations. In some real production problem, sometime there is a missing operation. For example is the assembly process at stainless steel factory [8]. At that assembly process, some joint types do not need pre-processing, such that there are some jobs that do not need pre-processing. In this case, we refer to as hybrid flow shop scheduling problem because the job skip some stage without changing the others production sequence.

Limited wait is a constraint that restricts the waiting time between the operations. Waiting time means a reasonably long time when the product waits to be processed at the next operation. For example, the fabrication process of semiconductor wafer. The waiting time after processed at furnace tubes are limited to avoid particle absorption from the air [5]. That waiting time is counted from the product finished at a process until that product start being processed at the next operation. Because of this limitation, the limited wait constraint (equation 1) is adding to the main HFS model.

$$S_{(i+1,j)} - C_{ij} \leq U \qquad (1)$$

where S_{ij} is a start time of job j at operation i, C_{ij} is completion time of job j at operation i, and U is a limitation of waiting time. The characteristics of HFS problem we solved in this paper are :

- The number of machine at each process is more than or equal to one.
- The waiting time for each job between the process are limited (limited wait constraints).
- Some job may not be processed at some operations, but the sequence of the other process are same (missing operations).

3.2. Particle Swarm Optimization

Particle Swarm Optimization (PSO) is developed based on the behavior of the swarm (bird or fish). The individual movement is influenced by its individual (cognitif) and social (swarm) movements [9]. Each individual (particle) is charaterized by velocity and position. Updating the velocity (v) and the position (x) of each particle based on its initial position, initial velocity, Gbest, and Pbest is done aording the following formula:

$$V_t = V_{t-1} + c_1r_1(Pbest - X_t) + c_2r_2(Gbest - X_t) \qquad (2)$$

$$X_{t+1} = X_t + V_t \qquad (3)$$

where Pbest is the best position of each particle among iterations, Gbest is the best of Pbest. PSO is suitable used to solve continuous problem, because the value of position and velocity of all of particles are continuous numbers. The disadvantage of PSO is the possibility to be trapped at local optimal. The solution can be trapped at local optimal because of the updating solution only based on the best solution from previous iteration and the algorithm does not have a mechanism to jump out from the local optimal. The modification by changing the updating mechanism (based on some best solution) can reduce the possibility to be trapped at local optimal. The approach that can save some best solutions is pareto optimal.

3.3. Multi Objectives

Pareto optimal is an approach based on the dominance concept. This approach used to find a solution set that closest to the optimal solution (pareto front). The number of the solution set's member depend on the number of non-dominated solution that can be found. The most suitable solution for the real condition can be chosen from the solution set. Pareto optimal insure that each criteria or objective function get a same treatment, because there is no priority from each criteria [7][. This approach have been proven effective for solving multi-objective problem which each criteria have the same weight.

Non-dominated solution is solutions that minimum have a better criteria than the others solution. When all of criterias are better than the others solution, the solution is a dominance solution. When all of criterias are worst than the others solution, the solution is a dominated solution. Non-dominated solutions will be added to the solution set, while the dominated solutions eliminated from the solution set and replace by the dominance solution. When there are some non-dominated solution at the solution set, a mechanism is needed to choose which is the best solution for generate new solutions. That mechanism is crowding distance.

Crowding distance used to measure the distance of a solution from other solutions. The bigger crowding distance will produce a better diversity [10]. The more far distance between the solution means that the solution set has better diversity and dispersion. When the solution dispersion is increasing, the probability to find the optimal solution is increasing too. Because of that, the solution with biggest crowding distance will be chosen. The steps to calculate crowding distance are shown below this [10]:

1. Sorting the solution based on each criteria value.
2. Based on the ranking at each criteria, calculate the distance for each criteria and each solution. The distance calculate based on this below equation.
$$d_{k,m} = \frac{f_{k+1,m} - f_{k-1,m}}{f_m^{max} - f_m^{min}} \tag{4}$$
where *k* is the solution index, *n* is the number of solution set member, and *m* is the criteria index. The distance value of the first and the last solution are infinite so that solution always chosen [11]. The first and the last solution are the limitation of the solution range. So, when one of them isn't chosen, the range of solution become smaller and the dispersion will be reduced.
3. Calculate the crowding distance by summarizing the distance at each criteria.

4. Mathematical Model

The model used in this paper is the modification of those proposed by [12]. The development were done by adding and changing constraints to suit the missing operations and limited wait characteristics. The objective functions are minimizing completion time (makespan), total tardiness time, and total machine idle time. Those three objective functions have some weight and to be minimized simultaneously.

i	: operation index
j	: job index
k	: machine index
S_{ij}	: starting time of job *j* at operation *i*
C_{ij}	: completion time of job *j* at operation *i*
p_{ij}	: processing time of job *j* at operation *i*
d_j	: due date of job *j*
M	: a very big number (big M)
x_{ijk}	: binary variable, 1 if job *j* processed by machine *k* at operation *i* and 0 if not
y_{ihj}	: binary variable, 1 if job *h* processed before job *j* at operation *i* dan 0 if not
a_{ij}	: binary variable, 1 if job *j* processed at operation *i* dan 0 if not

$$MIN\ f_p(Z_1 \colon Z_2 \colon Z_3) \tag{5}$$

$$Z_1 = max_{ij}\left\{C_{ij}\right\} \tag{6}$$

$$Z_2 = \sum_j \left(max\left\{max_i\{C_{ij}\} - d_j, 0\right\}\right) \tag{7}$$

$$Z_3 = \sum_k \left(max_{ij}\left\{C_{ij}\right\} - min_{ij}\left\{S_{ij} + M(1 - x_{ijk})\right\} - \sum_i \sum_j p_{ij} x_{ijk} \right) \tag{8}$$

$$C_{ij} = \left(S_{ij} + \sum_k p_{ij} x_{ijk}\right) a_{ij}\ , \forall i \in I, j \in J \tag{9}$$

$$\sum_k x_{ijk} = a_{ij}\ , \forall i \in I, j \in J \tag{10}$$

$$S_{(i+1)j} \geq C_{ij}, \forall i \in I, j \in J \tag{11}$$

$$S_{(i+1)j} - C_{ij} \leq w_{ij}, \forall i \in I, j \in J \tag{12}$$

$$C_{ij} - S_{ih} \leq y_{ihj}M, \forall i \in I, h \in J, j \in J, h \neq j \tag{13}$$

$$C_{ih} - S_{ij} \leq (1 - y_{ihj})M, \forall i \in I, h \in J, j \in J, h \neq j \tag{14}$$

$$x_{ijk} \in \{0,1\}, \forall i \in I, j \in J, k \in K \tag{15}$$

$$y_{ihj} \in \{0,1\}, \forall i \in I, h \in J, j \in J \tag{16}$$

$$a_{ij} \in \{0,1\}, \forall i \in I, j \in J \tag{17}$$

Equation (5) is the multi-objective function to minimize Z_1, Z_2, and Z_3 simultaneously. Equation (6) is the first objective function (Z_1) to calculate makespan. Equation (7) is the second objective function (Z_2) to count total tardiness. Equation (8) is the third objective function (Z_3) to count total machine idle time. Equation (9) is to count the completion time of the job at each process. Equation (10) is to assure that each job at each operation is processed once by one machine. Equation (11) is to assure that the processing of the job at an operation is started after the predecessor process of that job finished. Equation (12) is to make sure that the waiting time between the process do not exceed the limitations. Equation (13) and (14) are to assure that each machine can process only one job at a time. Equation (15)-(17) are to make sure that the variable values are binary.

5. Proposed Algorithm

The proposed algorithm is an algorithm that combines PSO with probability transition matrix and pareto optimal. The algorithm is called Modified Particle Swarm Optimization (MPSO). The MPSO algorithm can be described as follows:

1. Generate initial position and initial velocity for particles. Each initial position is in the form of a probability transition matrix . The dimension of initial position and initial velocity respectively is N by p, where N is the number of particles and P is the number of jobs.
2. Generate solution sequence based on the probability value of the transition matrix. Normalize first the probability transition matrix such that the row sum is equal to 0.
3. Calculate the fitness function
4. Find non-dominated solution (NDS) and update the pareto archive. The updating process based on some conditions as below :
   - If the new NDS be dominated by minimum one of pareto archive member, the pareto archive does not need an updating.
   - If the new NDS dominate one of pareto archive member or more, the new NDS are adding into the pareto archive and pareto archive member that dominated by the new NDS eliminated from the pareto archive.
   - If the new NDS does not dominate and is not dominated, the new NDS are adding to the pareto archive without eliminated pareto archive member.
   - If the size of the pareto archive exceeds the limitation, the pareto archive member are sorted by its crowding distance value, from the biggest value to smallest value. Pareto archive member which has smallest crowding distance value is eliminated until the size of the pareto archive does not exceed the limitation.
5. Find Gbest and Pbest
   - Gbest are chosen from pareto archive member randomly. In the MPSO algorithm, Gbest for each particle may be different, it is not like in original PSO that each particle has same Gbest.

- Pbest is chosen by comparing the old Pbest and the new solution for each particle. When the old Pbest and the new solution are not dominated each other, the Pbest is chosen from one of them that has higher order ranking

6. Updating the velocity use equation (18).

$$v_t = v_{t-1} + c_1(Pbest - random_t) + c_2(Gbest - random_t) \tag{18}$$

7. Updating the probability value of transition matrix use equation (19).

$$random_{t+1} = random_t + v_t \tag{19}$$

8. Checking the stopping criteria. If the stopping criteria is not reached, so the iteration repeated from the step 3. If the stopping criteria is reached, so the iteration stopped.

6. Experiment

Data sets used in this experiment are from OR Library. The data sets hel1 (Heller, 1960) were modified such that we have three different data sets

1. Data set *flow shop* (FS), dimension 10x10 (10 *job* 10 *stage*) with 1 machine at each stage.
2. Data set *hybrid flow shop* 2 (HFS2), dimension 20x10 (20 *job* 10 *stage*) with 3 machines at each stage.
3. Data set *hybrid flow shop* 3 (HFS3), dimension 30x10 (30 *job* 10 *stage*) with 3 machines at each stage.

The experiments are done to evaluate performance of the algorithm. The proposed algorithm is used to solve some cases where characterized by the number of jobs and machines. Paramater used in this paper based on the results of the parameter testing in the preliminary experiments, i.e. population number (n) = 1000, maximum iterations (itmax) = 200, c_1 = 0.3, c_2 = 0.7, and the pareto archive size limitation is 10 percent of population number. There are many kind of performance metrics that can be used to know the quality of the solution set. The performance criteria that previously used in the research is shown below [13]:

- General Distance Metric (GD) or convergence metric is distance between solution set and pareto front.

$$GD = \frac{\sqrt{\sum_{i=1}^{n} d_i^2}}{n} \tag{20}$$

- Spacing Metric (S) atau divergence metric is a dispersion measure of solution set

$$S = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(d - d_i)^2} \tag{21}$$

- Number of Non-Dominated Solution (NNDS) indicates how many non-dominated can be found.

The performance criteria used in this paper are the number of non-dominated, convergence metric, divergence metric, and computation time. The results of the experiment can be seen in Tables 1, 2, 3 and 4.

Table 1. Computation time for different cases.

Data Sets	**Average Computation Time (second)**		**Minimum Computation Time (second)**		**Maximum Computation Time (second)**	
	PSO	**MPSO**	**PSO**	**MPSO**	**PSO**	**MPSO**
FS1	410.6196	587.6464	396.5077	578.4517	439.0492	599.0126

FS2	453.5219	994.7531	448.7813	975.1563	458.375	1004.938
FS3	578.5492	1314.6	576.4705	1306.4	580.3237	1325.3
HFS1	371.4384	561.0381	365.7599	552.0875	377.5224	569.232
HFS2	483.7687	907.4625	481.8281	903.6406	485.7031	911.6719
HFS3	547.7273	1283.3	543.7259	1280.4	552.0719	1286
LWFS1	464.9298	687.9551	456.1313	682.9724	474.243	694.7349
LWFS2	537.7281	1041.422	534.2656	1015.328	541.0781	1055.672
LWFS3	657.825	1458.575	651.6786	1440.3	662.2866	1469.2
LWHFS1	437.9291	659.3694	434.9308	651.2886	440.656	666.1555
LWHFS2	520.5719	1042.494	516.9063	1030.813	528.2188	1055.438
LWHFS3	646.7372	1455.05	630.946	1448.8	655.7034	1465.2

The NNDS shows the number of the optimal solution that can be found. The NNDS of MPSO's solution sets are more than the NNDS of PSO's solution sets. If the number of the NNDS is increasing, the probability to find the optimal solution is increasing too. It means that MPSO algorithm has bigger probability to find the optimal solution than PSO algorithm. The convergence metric shows the distance between solution set and optimal solution. The convergence metric of MPSO's solutions sets are fewer than the convergence metric of PSO's solution sets. If the convergence metric is decreasing, the solution sets is closer to the optimal solution. It means that MPSO algorithm can produce solution sets closer to the optimal solution than those of PSO algorithm.

Table 2. The result of number of non-dominated solution (NNDS).

Data Sets	Average NNDS		Maximum NNDS		Minimum NNDS	
	PSO	MPSO	PSO	MPSO	PSO	MPSO
FS1	46.6	44.6	51	54	43	34
FS2	70.6	56.4	89	68	39	38
FS3	7.25	24.25	13	39	4	10
HFS1	36.2	39	39	45	35	32
HFS2	50	53	66	79	37	31
HFS3	47.75	50.75	66	76	38	23
LWFS1	39.4	42	41	47	37	39
LWFS2	71.2	62.6	81	93	46	43
LWFS3	5	26	6	32	3	20
LWHFS1	9.6	19.8	15	32	5	14
LWHFS2	17.2	28.2	29	35	4	11
LWHFS3	5.25	33	8	42	3	29

Table 3. The result of convergence metric.

Data set	Average		Minimum convergence		Maximum convergence	
	PSO	MPSO	PSO	MPSO	PSO	MPSO
FS1	18.69064	19.74676	17.9344	18.0728	19.1383	22.2714
FS2	15.83864	19.36146	13.9196	16.9493	20.3738	23.899
FS3	40.2946	24.75198	28.7164	19.6508	49.5226	33.0869

HFS1	20.12258	19.72883	19.4348	18.7562	20.4303	21.2668
HFS2	16.81366	18.78364	14.482	15.1776	18.5806	24.0503
HFS3	14.32188	15.58568	11.6963	11.7155	16.7184	21.7906
LWFS1	20.89856	20.2554	20.3979	18.9415	21.7414	20.9456
LWFS2	16.01526	19.08602	15.0727	15.0646	19.0826	22.2889
LWFS3	46.77843	21.79778	40.7414	19.6466	59.0604	24.5089
LWHFS1	43.23386	29.95048	32.7506	22.6481	56.9076	33.5503
LWHFS2	38.25176	29.83808	22.7312	25.5861	62.7081	41.4912
LWHFS3	45.38035	18.1212	34.1204	15.6371	54.7496	19.0556

The divergence metric shows the dispersion of solution sets. The divergence metric of MPSO's solution sets are more than those of PSO's solution sets. If the divergence metric is decreasing, the dispersion of solution sets is better. It means that MPSO algorithm cannot produce solutions sets better than the solution sets produced by PSO algorithm. The modification by adding the transition matrix to PSO make the MPSO algorithm needed much time to find the optimal solution. This is disadvantage of MPSO algorithm compared to PSO algorithm. But, MPSO algorithm also has advantage that is the quality of the solution is better than the solution produced by PSO algorithm. The solution of MPSO algorithm is better, because it has bigger probability to find optimal solution and produce solution sets that closer to the optimal solution.

Table 4. The results of divergence metric.

Data set	**Average**		**Minimum Divergence**		**Maximum Divergence**	
	PSO	**MPSO**	**PSO**	**MPSO**	**PSO**	**MPSO**
FS1	3.52396	5.21076	2.5495	3.4303	5.3137	7.6514
FS2	8.44268	8.91154	3.1566	4.9111	20.3184	12.3185
FS3	6.0442	6.797925	3.4641	2.7889	8.8918	16.9039
HFS1	6.8421	6.6502	4.6557	3.877	13.9267	8.4275
HFS2	3.94416	5.75744	2.4305	2.991	8.5563	7.9315
HFS3	5.891475	6.169175	3.0623	3.743	8.9818	7.841
LWFS1	4.04944	5.03074	3.2364	3.208	5.882	7.8089
LWFS2	9.04326	8.83724	4.6961	5.5712	12.9217	11.3803
LWFS3	10.028	7.69615	2.6833	5.2942	19.9575	10.7271
LWHFS1	15.12552	7.25384	8.8428	3.1557	25.026	15.495
LWHFS2	10.37906	9.31244	5.6195	5.3377	16.2384	19.3111
LWHFS3	7.6178	7.9582	5.0709	6.0333	9.815	11.5819

7. Conclusion

MPSO algorithm has been successfully developed to solve the multi-objective discrete problem. It can solve some type of flow shop scheduling problems, i.e. flow shop scheduling, hybrid flow shop scheduling, limited wait flow shop scheduling and limited wait hybrid flow shop scheduling problems. All of those are multi-objective problems with three objective functions (i.e minimizing makespan, total tardiness, and total machine idle time). Furthermore, this algorithm can be applied to solve other type of scheduling problems. The modification of PSO by adding transition matrix making the MPSO algorithm is more efficient than the original PSO. Although MPSO algorithm required longer computing time, the quality of solution sets are better than those of original PSO. Though the dispersion of solution sets are not better, but the distance of solution sets are closer to the optimal solution. The distance between solution sets and the optimal solution is more important than the dispersion of solution sets. MPSO algorithm still can be further developed so it has a new mechanism of generating initial solution, updating solution, and stopping criteria that can reduce the computation time. When the computation time reduced, MPSO algorithm perfectly can be better than the original PSO algorithm.

8.References

[1] Bagchi T P, Gupta J N D and Sriskandarajah C 2006 A review of TSP based approaches for flowshop scheduling *European Journal of Operational Research* **169** (3) 816–854

[2] Avriel M, Penn M and Shpirer N 2000 Container ship stowage problem : complexity and connection to the coloring of circle graphs *Discrete Applied Mathematics* **103** (1–3) 271–279.

[3] Ruiz R and Vazquez-Rodriguez J A 2010 The hybrid flow shop scheduling problem *European Journal of Operational Research* **205** (1) 1–18

[4] Yang D and Chern M 1995 A two-machine flowshop scheduling problem with limited waiting time constraints *Computers & Industrial Engineering* **28** (1) 63–70

[5] Su L 2003 A huybrid two-stage flowshop with limited waiting time constraints *Computers & Industrial Engineering* **44** (3) 409–424

[6] Gicquel C, Hege L, Minoux L and vanCanneyt W 2012 A discrete time exact solution approach for a complex hybrid flow-shop scheduling problem with limited-wait constraints *Computers & Operations Research* **39** (3) 629–636

[7] Jarraya B and Bouri A 2012 Metaheuristic Optimization Backgrounds : A Literature Review. *International Journal of Contemporary Business Studies* **3** (12)

[8] Tseng C, Liao C and Liao T 2008 A note on two-stage hybrid flowshop scheduling with missing operations *Computers & Industrial Engineering* **54** (3) 695–704

[9] Santosa B and Willy P 2011 *Metode Metaheuristik Konsep dan Implementasi* (Surabaya:Guna Widya)

[10] Rahman R A and Santosa B 2013 *Differential Evolution dan Bottleneck Heuristik untuk Menyelesaikan Permasalahan Penjadwalan Multi-Objective Hybrid Flowshop Unrelated Parallel Machines* (Surabaya: Sepuluh Nopember Institute of Technology thesis)

[11] Racquel C R and Naval Jr P C 2005 An effective use of crowding distance in multiobjective particle swarm optimization *Proc of the 2005 conference on Genetic and evolutionary computation* 257-264

[12] Liao C J, Tjandradjaja E and Cung T 2012 An approach using particle swarm optimization and bottleneck heuristic to solve hybrid flow shop scheduling problem *Applied Soft Computing* **12** (6) 1755–1764

[13] Qiu C, Wang C and Zuo X 2013 A novel multi-objective particle swarm optimization with K-means based global best selection strategy *International Journal of Computational Intelligence System* **6** (5) 822–835