A Cross Entropy-Genetic Algorithm Approach for Multi Objective Job Shop Scheduling Problem

Budi Santosa and Liliek Nurkhalida
Department of Industrial Engineering
Surabaya Institute of Technology
Indonesia

Abstract

Multi Objective Job Shop Scheduling Problem (MOJSP) is a problem for finding optimal operation sequences of some jobs according to more than one goal to achieve. The problem gets harder as its complexity increases. The development of optimization method has led many new methods to solve this problem. This paper offers Cross Entropy-Genetic Algorithm (CEGA) method to solve job shop scheduling problem with multi objectives. CEGA was cnstructed from combination of Cross Entropy method with Genetic Algorithm. CEGA has been successfully applied on single objective job shop scheduling. The weighted objective approach was proposed to accommodate multi objective computation. The experiment results show that generally, CEGA produced competitive solutions compared to Simulated Annealing.

Keywords
Multi Objective, Job Shop Scheduling, Cross Entropy, Genetic Algorithm

1. Introduction
Job Shop Scheduling Problem (JSP) is one of NP hard problem. The bigger the problem size, the longer time required to solve the problem. It is very time consuming for exact approach to solve this kind of problem. While, there are numerous jobs in real world problem, make this problem need long time to find optimal solution. Besides, some objectives are commonly considered in real condition. In this paper, JSP takes some objectives to be optimized (Multi Objective Job Shop Scheduling Problem). Since the problem gets more complex, it needs particular method in order to get satisfied solution in reasonable amount of time.

In this paper, the problem has two objectives commonly found in manufacturing company, minimize makespan and TWT [6]. Minimizing total waiting time (TWT) will make production system meet JIT condition. While high efficiency on resource utility gained through minimizing the makespan [8,9]. According to [6], makespan and TWT are in contradiction. It means that getting better value in one objective will make other objective value get worse.

Cross Entropy is merely a new metaheuristics [4]. This algorithm has sample elite mechanism that used in generating sample population on the new iteration. This mechanism will keep new population in converge track to its optimal solution. Cross Entropy was successfully applied in some others NP hard problem such as Generalized Orienting Problem [17], reliable network design [6] and discrete-continuous scheduling with continuous resource discretization [10].

This method was combined with Genetic Algorithm, which have cross over and mutation as its mechanism. This mechanism was prior to diversify some solutions obtained in each iteration. This will avoid the algorithm from local optimal trap. Besides, it has elitism mechanism not found in Cross Entropy. With elitism, the best sample on population will be restored and used in the next new sample population.

This hybrid of Cross Entropy-Genetic Algorithm (CEGA) was successfully applied in Job Shop Scheduling Problem with single objective [18]. CEGA tends to have good performance on solving Multi Objective Job Shop Scheduling Problem. The performance of CEGA was compared to the reference that uses Simulated

Annealing as its solving method [6]. This comparative reference was chosen because it uses the same objectives as used in this paper, that is minimize makespan and Total Weighted Tardiness.

2. Problem Description

2.1 Job Shop Scheduling Problem

Scheduling problem usually becomes main aspect on economic point of view. Job shop scheduling was one of those problems which founded in production floor environment. This kind of problem known as difficult problem because has many constraints. It is difficult because of complexity structure of unit produced and so, makes many variation of production flow [10]. There are mono objective and multi objectives mode on its case. Job Shop Scheduling Problem was described on some points below [19]:

- The problem consists of *n* jobs and each job has some *o* operations.
- There are *m* machines, each of them can only operate one operation in a time.
- Each operation will be proceeding on certain machine and certain operation time.
- No breakdown, lost material, etc. that will stop production process.
- The Job Shop Scheduling Problem goal is to get effective production scheduling, that minimize completion time of all jobs.

There are some objectives often used in Multi Objective Job Shop Scheduling Problem such as minimize makespan, minimize machine workload, minimize total machine workload, minimize total weighted tardiness, minimize tardiness, minimize cost, minimize total weighted completion time.

2.2 Cross Entropy

Cross Entropy works based on information on the sample distribution [4]. This information helps the algorithm to capture the distribution of the good sample. The new sample generated based on this distribution. This distribution will be continually updated through each iteration until it finds the optimal solution. Cross Entropy has to complete two main steps on its algorithm, they are:

1. Generate sample randomly
2. Update its parameter based on data of the best sample (sample elite). So the next iteration will have a good sample produced base on this parameter.

2.3 Genetic Algorithm

Genetic Algorithm is one of the oldest algorithms known in metaheuristics [5]. This algorithm mechanism is inspired by biological reproduction process. On this algorithm, the initial sample population generated randomly. For each sample generation, there will be fitness value evaluation. The output of fitness evaluation is that the sample with the best fitness value will be chosen as sample parent. Two sample parent then processed in cross over and mutation mechanism. The spring samples obtained from this process are going to be listed in new population sample on the next iteration.

2.4 Multi Objective Problem

Utility function is a technique that can convert some objective function in multi objective problem into single one. It will mark the preference of an objective to the others with a weighted preference [14]. Hence, those objectives can be treated as single objective which is easier to solve. This research has two objectives with different unit, makespan with time unit and Total Weighted Tardiness with currency. Therefore, it needs advance treatment as well as objective function normalization. The formula used in this research based on equation (1) [11], is the one which can do both converting multi objective function into single one also unit normalization.

$$\text{distance} = \sum_i w_i \left[\frac{f_i(x) - f_i^*}{f_i^- - f_i^*}\right] \qquad (1)$$

with w_i as ith objective's weight and , f_i+ as the best value of *i*th objective solution and f_i^- as the worst value of *i*th objective solution.

3. Proposed Algorithm

CEGA to solve Multi Objective Job Shop Scheduling Problem can be explained through a simple example on some steps below:

1. Initial ization

Some parameters used are number of samples (N) = 6, number of operations in one sample (n) = 9, rho (ρ) = 0.02, alpha (α) = 0.8, Initial crossover parameter (Pps) = 1, Stopping criteria (β) = 0.001, Weighted of makespan (bMS) = 0.5, Weighted of Total Weighted Tardiness (bTWT) = 0.5.
A simple case to explan how the alorithm works is given. Three jobs with its due date and weighted can be seen in Table 1, including machine used to process and time duration for processing each operation.

Table 1. Data of 3 *Jobs*-3 Operations-3 Machines Problem

job	due date	weighted	1st operation		notation	2nd operation		notation	3rd operation		notation
			machine	op. time		machine	op. time		machine	op. time	
1	17	1	1	4	1	3	2	2	2	1	3
2	15	1	1	2	4	2	5	5	3	3	6
3	11	1	3	1	7	1	3	8	2	2	9

2. Generating samples

The sample is sequence notation form that shows priority of each operation. The notation represents the identity of the number, which contains information about its particular operations and job, according to Table 1. For example, number 4 according to Table 1, represents 1st operation of job 2, while number 8 represent 2nd operation of job 3. N samples are generated randomly. For instances, we generated 6 samples as follows:

X1 : 4-5-7-6-1-2-3-8-9, X2 : 4-7-3-9-8-1-5-6-2, X3 : 1-9-7-8-4-6-5-2-3, X4 : 4-5-6-1-7-8-9-2-3
X5 : 6-3-7-8-5-1-2-4-9, X6 : 5-7-8-3-6-1-2-4-9

3. Get objective value for all samples

For instance, to get objective value of a sample, we use 1st sample (sample X1).

1. The first number in notation sequence of sample X1 is 4. Then we check on Table 1, which machine is needed to process the operation. This information is important to check the position of the operation through others operation. Also check the position on Gantt chart:
   a) If the operation has neither predecessor operation (P_o) nor earlier operation (E_o), then start time of this operation should be 0. (P_o was indicate by smaller number in its job row on Table 1 and Eo was indicate by operation exist in front of it in Gantt chart on the same machine)
   b) If there is E_o but no P_o of the operation, then the operation start time is equal to finish time of E_o
   c) If there is P_o but no E_o of the operation, then the operation start time is equal to finish time of P_o
   d) If there are both E_o and P_o, then the operation start time is equal to the longer finish time of E_o or P_o.

   Since operation 4 has no smaller number in row job 2 and also the Gantt chart still blank (no P_o and E_o), so it satisfied point (a) and should be placed in row "machine 1" on Gantt chart and fill some column equal to its operation time.

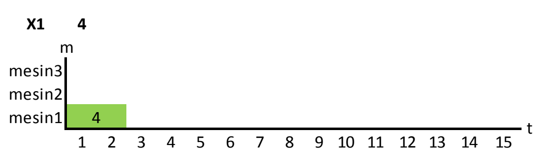

Figure 1. First Step of Operation Scheduling of Sample X1

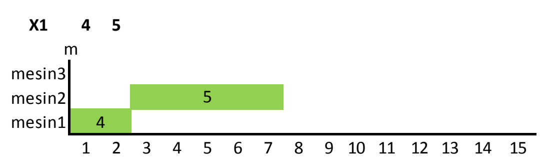

Figure 2. Second Step of Operation Scheduling of Sample X1

2. The next priority operation on sample X1 is operation notated by number 5. This operation has 4 as Po but has no Eo in row "machine 2" of the last updated Gantt chart. Thus, it satisfied point (c) and has start time equal to finish time of operation 4. The operation will update the Gantt chart as seen on Figure 2.

3. Do the same procedure to the rest of operations. Noted that for operation which its Po have not yet scheduled in Gantt chart, then the operation can not be executed until its Po is scheduled. When finally all operations scheduled, the Gantt chart will be seen like Figure 3.

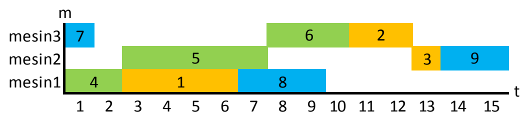

Figure 3. The Last Step of Operation Scheduling of Sample X1

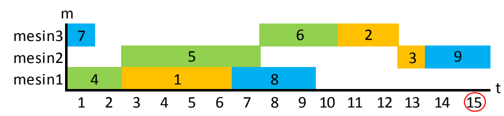

Figure 4. Determining Makespan Value

4. When the Gantt chart already complete, we can decide the makespan value. The value is equal to the latest operation finish time among all operations. For sample X1, the last operation scheduled is operation 9, and it is finish on 15. So the makespan of sample X1 is 15 time unit.

5. Next step is counting the Total Weighted Tardiness. Count on the lateness job due to its due date, multiply with its weighted to get weighted tardiness. Then sum all weighted tardiness of all jobs, that's so called Total Weighted Tardiness. For Sample X1, it has Total Weighted Tardiness = 4 with detail for its counting as in Table 2.

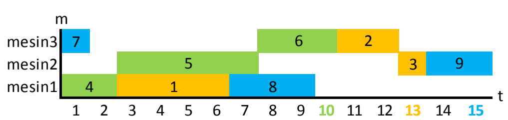

Figure 5. Determining Completion Time of Each Job

Table 2. Counting TWT of Sample X1

	due date	completion time	weighted	total weighted
job 1	17	13	1	max(0,(13-17)*1)
job 2	15	10	1	max(0,(10-15)*1)
job 3	11	15	1	max(0,(15-11)*1)
TWT				**4**

6. Do step 1 – 5 for sample 2 to 6.
7. After computing the makespan and Total Weighted Tardiness of each sample, the next step is to obtain final objective value (z) of each sample. It uses equation (2) to get z.

$$z = \left(\left(\frac{makespan_i - makespan_{min}}{makespan_{max} - makespan_{min}}\right) * bobot\_makespan\right) + \left(\left(\frac{TWT_i - TWT_{min}}{TWT_{max} - TWT_{min}}\right) * bobot\_TWT\right) \quad (2)$$

The makespan, Total Weighted Tardiness, and z for each sample are in Table 3.

Table 3. Makespan, TWT, and *z* Value for Each Sample

sampel	makespan	TWT	Z
X1	15	4	0.150
X2	18	1	0.300
X3	20	5	0.700
X4	17	5	0.400
X5	19	11	0.900
X6	19	11	0.900

Table 4 Ascending order of the Sample Based on *z* Value

sampel	makespan	TWT	Z
X1	15	4	0.150
X2	18	1	0.300
X3	17	5	0.400
X4	20	5	0.700
X5	19	11	0.900
X6	19	11	0.900

4. **Determine sample elite**

Sort all z values in ascending mode. With rho of 0.02, there is N*rho = 6*0.02 = 0.12 ~ 1 sample elite. Thus, X1 with the smallest z value becomes the only one sample elite among individu in the sample.

5. Update crossover parameter

Crossover parameter continually updated in each iteration. Thus, the more iteration goes, the smaller the parameter value get. This parameter is going to be used to determine stopping criterion.

$u = \frac{\overline{Ze}}{2*Z_{best}} = \frac{0.143}{2*0.143} = 0.5$

Pps(1) = (1-)*u + (Pps(0)*)

= (1-0.8)*0.5 + (1*0.8)
= 0.9

6. Sample elite weighting

The weight obtained from evaluating the best value sample in the previous iteration. If makespan value of the sample is better than the best sample in the previous iteration, then give it weight as big as number of sample elite. Conversely, give one as it weight. In this case, the process is still in first iteration and just has one sample elite. So, the weighted of sample elite is 1.

7. Linear Fitness Ranking (LFR)

Linear Fitness Ranking (LFR) value is necessary as parameter in choosing parent sample for crossover process. The formula of Linear Fitness Ranking is:

LFR(I(N-i+1)) = Fmax-(Fmax-Fmin)*((i-1)/(N-1)) (3)

Using Fmin = 1/Z(1) = 6.67 and Fmax = 1/Z(N) = 1/Z(6) = 1.11, we can count LFR for all samples.

X1 : LFR = 6.67-((6.67-1.11)*((1-1)/(6-1))) = 6.67 , X2 : LFR = 6.67-((6.67-1.11)*((2-1)/(6-1))) = 5.56
X3 : LFR = 6.67-((6.67-1.11)*((3-1)/(6-1))) = 3.33 , X4 : LFR = 6.67-((6.67-1.11)*((4-1)/(6-1))) = 4.44
X5 : LFR = 6.67-((6.67-1.11)*((5-1)/(6-1))) = 2.22 , X6 : LFR = 6.67-((6.67-1.11)*((6-1)/(6-1))) = 1.11

8. Elitism

Elitism needs to be held at each iteration to save the sample with the best value. This best sample will occur in the next new sample population. The procedure will have only one best sample to be elitism. In this case, sample X1 was chosen to be elitism sample.

9. Parent sample selection

In applying roulette wheel mechanism, parent sample 1 chosen from population of sample elite and parent sample 2 chosen from the whole population sample.

a) Parent sample 1 selection : the evaluation done for each sample. Cumulative weighted of sample divided by total weighted. The result then compared with random number. If division result value higher than random number, then the sample can be considered as parent sample 1. Because there is only one sample elite, sample X1 automatically become parent sample 1.

b) Parent sample 2 selection : the selection procedure mostly same as parent sample 1, except it use LFR than weighted value. For instance, sample X3 decided as parent sample 2.

Total LFR = 6.67+5.56 +3.33 +4.44 + 2.22 +1.11 = 23.33
for X1=6.67/23.33 = 0.29 (< 0.6227)
X2=(6.67+5.56)/23.33=0.52(< 0.6227)
X3=(6.67+5.56+3.33)/23.33=0.71(> 0.6227)

10. Crossover

From the previous step, it got X1 and X3 as parent sample. To decide whether the sample have to cross over or not, generate a random number. If this random number has smaller value than cross over parameter, then do the cross over. If doesn't, then do not do the cross over and go to the next process. For example, random number generated is 0.015, which smaller than cross over parameter (0.9). So the sample parents have to be in cross over. The cross over mechanism will do as follows: generate two random numbers, for example we get 0.6948 and 0.8491. Convert these numbers into integer number. The numbers will become border of genes in sample parent that will be exchange between two parents.

ri = ceil (random*n) (4)
r1 = ceil(0.4387*9) = 4 -> p1 = 4
r2 = ceil(0.8491*9) = 8 -> p2 = 8

sample parent 1 = 4 5 7 6 |**1 2 3 8**| 9

sample parent 2 = 1 9 7 8 |**4 6 5 2**| 3

The numbers inside the brackets will be exchange each other between sample parent 1 and 2.

Sample spring 1 = … … … … 4 6 5 2 … → Sample spring 1 = 7 1 3 8 4 6 5 2 9
Sample spring 2 = … … … … 1 2 3 8 … Sample spring 2 = 9 7 4 6 1 2 3 8 5

These sample springs will substitute the old Sample X2 and Sample X3 in a series. Evaluate the next sample, the 5th and the 6th sample. Do the same procedure to get two sample parents. Do the cross over mechanism if its meet the requirement. If do not, then use both sample parents to substitute 5th and 6th sample. The new population when cross over complete as follows:

X1 : 4-5-7-6-1-2-3-8-9 X4 : 4-5-6-1-7-8-2-3-9 X3 : 9-7-4-6-1-2-3-8-5
X2 : 7-1-3-8-4-6-5-2-9 X5 : 5-7-8-4-6-1-2-3-9 X6 : 7-3-8-6-1-4-5-2-9

11. Mutation

Mutation parameter (Pm) is a half value of cross over parameter, 0.9/2 = 0.45. The example for mutation mechanism can be explained as follow: for sample X3, where i = 1 and r = 0.1174 (<Pm), so do the mutation on gen a and gen b, where the position of gen a is ceil (r*n) = ceil(0.1174*9)   2, and position for gen b is X3(1,i) = X3(1,1) = 7.

Old sample X3 = 9-7-4-6-1-2-3-8-5 → mutated sample X3 = 9-3-4-6-1-2-7-8-5
For i = 2, the generated random is r = 0.7757 (> Pm), so there is no need to do the mutation on this level. Do the same procedure to the entire operation of all samples, except elitism sample. The complete evaluation will disorder the original sequence of gen in some sample like below:

X2 : 7-8-5-6-4-2-3-1-9, X3 : 9-3-6-8-1-2-4-7-5, X4 : 1-7-6-3-4-8-2-5-9, X5 : 3-9-5-4-6-1-2-8-7, X6 : 2-9-8-6-1-3-5-7-4

12. Compute objective value from updated sample

In this step, we already have a new population consist of samples from cross over, mutation, and elitism process. These samples then evaluated through z values like the previous one. It is hoped that the best sample of the higher iteration will have better value (smaller, in minimizing case) than the best one in the previous iteration. This indicates the successful of the algorithm in purpose of converging solution into optimality approach.

Table 5. *Makespan*, *TWT*, and z From The First Iteration

i th sample	makespan	TWT	z
1	15	4	0.1
2	19	8	0.9
3	20	6	0.8
4	15	3	0
5	18	7	0.7
6	20	6	0.8

4. Numerical Experiments

To test the algorithm performance, it uses data which have been tested before in the previous same problem research. The research was held by Fattahi [6] using Simulated Annealing to solve Multi Objective Job Shop Scheduling with same objective as used in this research.

4.1 Data

There are four type of problem with different size for each problem [6] .

Table 6. Problem Description of MOJSP

Type of problem	Number of *jobs*	Number of operations	Number of machines
MOJ 1	4	3	3
MOJ 2	6	4	4
MOJ 3	10	4	6
MOJ 4	15	4	7

Table 6 shows jobs, machine needed particularly by each operation, time duration to process each operation, and due date for each job.

Table 7. Data Input for Problem MOJ 1

problem MOJ 1							
		operation 1		operation 2		operation 3	
jobs	due date	machine number	process time	machine number	process time	machine number	process time
1	150	1	85	2	55	2	95
2	190	1	26	2	54	3	83
3	140	2	45	3	23	3	68
4	480	2	47	2	58	3	42

Table 8. Data Input for Problem MOJ 2

problem MOJ 2									
		operation 1		operation 2		operation 3		operation 4	
jobs	due date	machine number	process time	machine number	process time	machine number	process time	machine number	process time
1	450	1	25	2	55	3	45	4	45
2	650	1	26	2	54	3	55	4	23
3	160	2	40	3	25	2	60	3	55
4	350	1	30	2	75	4	60	4	62
5	440	2	45	2	23	3	25	3	68
6	220	1	47	3	25	3	35	4	42

Table 9. Data Input for Problem MOJ 3

problem MOJ 3									
		operation 1		operation 2		operation 3		operation 4	
jobs	due date	machine number	process time	machine number	process time	machine number	process time	machine number	process time
1	650	1	25	2	55	4	45	5	45
2	450	1	26	2	54	5	55	6	23
3	330	2	40	3	25	5	60	6	55
4	300	1	30	2	75	4	60	5	62
5	1000	2	50	3	45	5	45	6	65
6	250	2	45	3	65	4	25	5	47
7	350	3	50	3	35	5	34	6	26
8	550	2	47	4	20	4	25	5	54
9	250	2	45	2	23	6	25	5	68
10	850	1	47	3	25	5	35	6	42

Table 10. Data Input for Problem MOJ 4

problem MOJ 4									
		operation 1		operation 2		operation 3		operation 4	
jobs	due date	machine number	process time	machine number	process time	machine number	process time	machine number	process time
1	700	1	25	3	55	4	45	7	45
2	850	1	58	3	54	5	75	7	93
3	300	2	40	2	25	5	60	6	55
4	350	1	30	2	85	4	60	6	92
5	220	2	50	3	45	5	45	7	65
6	400	2	45	2	65	4	25	7	47
7	790	3	90	2	95	5	104	6	126
8	650	2	47	3	20	4	25	6	54
9	750	2	45	4	23	6	25	6	68
10	390	3	55	2	45	4	55	6	55
11	560	3	45	3	65	4	36	7	47
12	470	2	35	3	32	5	45	7	48
13	900	1	62	2	33	5	65	6	35
14	400	2	28	3	25	6	33	6	65
15	230	1	47	3	25	5	35	7	42

4.2 Experiment Results

Comparison table for experiment result of Cross Entropy-Genetic Algorithm versus Simulated Annealing in solving Multi Objective Job Shop Scheduling as follow:

Table 11. Comparison Table for SA vs CEGA Solution 1

Simulated Annealing							
MOJ 1		MOJ 2		MOJ 3		MOJ 4	
makespan	TWT	makespan	TWT	makespan	TWT	makespan	TWT
354	233	407	247	580	**572**	826	**2769**
396	175	430	138	583	**533**	831	**2385**
408	140	433	78	590	419	842	**2039**
		445	**75**	596	**167**	868	1344
		459	40			883	**1285**
		484	33				
		487	32				
		494	20				

Cross Entropy-Genetic Algortihm							
MOJ 1		MOJ 2		MOJ 3		MOJ 4	
makespan	TWT	makespan	TWT	makespan	TWT	makespan	TWT
354	233	407	247	580	**597**	826	**1901**
396	175	430	138	583	**538**	831	**2320**
408	140	433	78	590	419	842	**2030**
		445	**79**	596	**223**	868	1344
		459	40			883	**1282**
		484	33				
		487	32				
		494	20				

Table 12. Gap Solution of SA-CEGA in MOJ

MOJ 1						
i th alternative solution	makespan SA	makespan CEGA	gap	TWT SA	TWT CEGA	gap
1	354	354	0	233	233	0
2	408	408	0	140	140	0
3	396	396	0	175	175	0

For more details about comparison on each type of problem, the table below will show gap between makespan SA – makespan CEGA and TWT SA – TWT CEGA.

Table 13. Gap Solution of SA-CEGA in MOJ 2

MOJ 2						
i th alternative solution	makespan SA	makespan CEGA	gap	TWT SA	TWT CEGA	gap
1	407	407	0	247	247	0
2	430	430	0	138	138	0
3	433	433	0	78	78	0
4	445	445	0	**75**	**79**	5.33
5	459	459	0	40	40	0
6	484	484	0	33	33	0
7	487	487	0	32	32	0
8	494	494	0	20	20	0

Table 14. Gap Solution of SA-CEGA in MOJ 3

MOJ 3						
i th alternative solution	makespan SA	makespan CEGA	gap	TWT SA	TWT CEGA	gap
1	580	580	0	572	597	4.371
2	583	583	0	533	538	0.938
3	590	590	0	419	419	0
4	596	596	0	167	223	33.533

5. Analyses

On Table 11, some numbers bolded means that there is some gap between SA and CEGA solutions. But it only appears on TWT's column, which means that there is no gap between SA and CEGA in makespan value. This could happen because solution of CEGA got from so many alternative solutions. With big number of N solutions, the possibility of getting the same solution as SA is high. In job shop scheduling case, makespan from different sequence priority can have same value. Because makespan only considering total completion time. While TWT can be different in different sequence priority because it is depend on completion time of each job. For MOJ 1 problem, there is no gap between CEGA and SA, both on makespan and TWT value. Thus, CEGA could be considered success in small size problem. As well as MOJ 2, although there is a gap for 5.3%, with the other maximum solution achievement, it can be ignored. Thus, CEGA for problem with MOJ 2 size have good performance as effective as in MOJ 1 problem. There are so many solutions produced by CEGA which make it possible to get this no-gap-solution. With same weight preference between two objectives, CEGA could give optimal solution as well as non-dominated solution for MOJ 1 and MOJ 2 problem.

Table 13. Gap Solution of SA-CEGA in MOJ 2

MOJ 2						
i th alternative solution	makespan SA	makespan CEGA	gap	TWT SA	TWT CEGA	gap
1	407	407	0	247	247	0
2	430	430	0	138	138	0
3	433	433	0	78	78	0
4	445	445	0	**75**	**79**	5.33
5	459	459	0	40	40	0
6	484	484	0	33	33	0
7	487	487	0	32	32	0
8	494	494	0	20	20	0

Table 14. Gap Solution of SA-CEGA in MOJ 3

MOJ 3						
i th alternative solution	makespan SA	makespan CEGA	gap	TWT SA	TWT CEGA	gap
1	580	580	0	572	597	4.371
2	583	583	0	533	538	0.938
3	590	590	0	419	419	0
4	596	596	0	167	223	33.533

Table 15. Gap Solution of SA-CEGA in MOJ 4

MOJ 4						
i th alternative solution	makespan SA	makespan CEGA	gap	TWT SA	TWT CEGA	gap
1	826	826	0	2769	1901	-31.347
2	831	831	0	2385	2320	-2.725
3	842	842	0	2039	2030	-0.441
4	868	868	0	1344	1344	0
5	883	883	0	1285	1282	-0.233

For MOJ 3 problem, all of TWT CEGA have gap to TWT SA. 33.35% as the biggest gap founded on 4th alternative solution. This could be happen because CEGA doesn't have restore mechanism for non-dominated solution founded in every iteration. It needs more effort to deliver more alternative solutions. With more alternative solutions, make possible to get the solution close to TWT SA. There is negative gap founded on gap TWT for MOJ 4 problem. It means that solution obtained by CEGA is better that SA's. This is because CEGA could perform wider search solution with crossover and mutation mechanism. When SA stuck with local search, it limits searching area to get possible better solution. As mentioned before, SA evaluating a solution based on local search. So, the chances to gain a solution outside the searching space were too small. While CEGA continue in diversified through cross over and mutation process. This diversification process allows CEGA to search solution in the area not explored before. In this case, CEGA had the edge on SA. Despitefully, SA which suggested in the reference journal has stopping criteria in particular condition. This stopping criterion was added to algorithm by purposes of shorting computational time. It is delimitate searching solution of SA.

However, the first alternative solution of TWT CEGA is smaller than TWT CEGA on the second and third ones. Considering makespan value, the second and third alternative solutions are dominated by the first alternative solution. This is in contradiction with Pareto concept that the solution in Pareto set solutions doesn't dominating each other. This phenomenon appears because of CEGA mechanism which is count on solution selection based on single objective value, not on Pareto concept.

6. Conclusion

Based on the experiment results and analysis above, it is proved that hybrid CEGA is an efficient method in solving MOJSP. It produces competitive solutions especially in small-scale problems. CEGA is much better in doing diversification solution than Simulated Annealing which use neighborhood search method. But, CEGA is worse in determining non-dominated solution because it's only evaluating a solution based on single objective value.

Acknowledgements

This research was supported by Higher Education Directorate of Indonesian Government.

References

[1]. Altiparmak F., Dengiz B. 2008, ‘A cross entropy approach to design of reliable networks’, European Journal of Operational Research, vol. 199, pp. 542-552.

[2]. Anonim. Pareto Optimality, Optimization in Engineering Design, System Realization Laboratory, Georgia Institute of Technology, handout lecturer.

[3]. Baker K.R., Trietsch D. 2009, Principles of sequencing and scheduling, A John Willey & Sons, Inc. Publication, New Jersey.

[4]. De Boer, Pieter-Tjerk, Kroese Dirk P., Mannor S., Rubinstein R.Y. 2003, ’A tutorial on the Cross-Entropy’, Haifa: Department of Industrial Engineering, Technion – Israel Institute of Technology.

[5]. Dreo J., Petrowski A., Siarry P., Taillard E. 2006, Metaheuristics for Hard Optimization, Springer, Heidelberg.

[6]. Fattahi P., Mehrabad M.S., Aryanezhad M.B. 2006, ‘An algorithn for multi objective job shop scheduling’, Journal of Industrial Engineering International, vol. 2, no. 3, pp. 43-53.

[7]. Fogel D.B. 2000, ‘What is evolutionary computation?’, IEEE Spectrum, pp. 26-32.

[8]. Gao J. 2010, ‘A novel artificial immune system for solving multi objective scheduling problems subject to special process contraint’, Computers & Industrial Engineering, vol. 58, pp. 602-609.

[9]. Gao J., Gen M., Sun L., and Zhao X. 2007, ‘A hybrid of genetic algorithm and bottleneck shifting for multiobjective flexible job shop scheduling problems’, Computers & Industrial Engineering, vol. 53, pp. 149-162.

[10]. Jedrzejowicz P., Skakovski A. 2009, ‘A cross-entropy-based population-learning algorithm for discrete-continuous scheduling with continuous resource discretisation’, Neurocomputing, vol. 73, pp. 655-660.

[11]. Low C., Wu T.H., Ming Hsu C. 2005, ’Mathematical modelling of multi-objective job shop scheduling with dependent setups and re-entrant operations’, International Journal Advanced Manufacturing Technology, vol. 27, pp. 181-189.

[12]. Moghaddam R.T., Azarkish M., Barkousaraie A.S. 2010, ‘Solving a multi objective job shop scheduling problem with sequence-dependent setup times by a pareto archive PSO combined with genetic operators and VNS’, International Journal Advance Manufacturing Technology.

[13]. Moslehi G., Mahnam M. 2010, ‘A pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization and local search’, International Journal Production Economics, vol. 129, pp. 14-22.

[14]. Ramesh S., Cary J. M. 1989, ‘Multicriteria job shop scheduling’, Computer Industrial Engineering, vol. 17, no. 1-4, pp. 597-602.

[15]. Rao S.S. 2009, Engineering Optimization: Theory and Practice, Fourth Edition, John Willey & Sons, Inc., New Jersey.

[16]. Santosa B., Willy P. 2011, Metoda Metaheuristik : Konsep dan Implementasi, Guna Widya, Surabaya. (Bahasa Indonesia edition) .

[17]. Santosa, B and Hardiansyah, N., 2010**,** Cross Entropy Method for Solving Generalized Orienteering Problem**,** iBusiness,, vol. 2, pp. 342-347

[18]. Santosa, B, Budiman, MA, and Wiratno, S.E, 2011, A Cross Entropy-Genetic Algorithm for m-Machines No-Wait Job-Shop Scheduling Problem, JILSA, vol. 3 no. 3.

[19]. Xu Q. 2001, ’Introduction to job shop scheduling problem’. Handout lecturer