ROBUST-HYBRID GENETIC ALGORITHM FOR A FLOW-SHOP SCHEDULING PROBLEM (A Case Study at PT FSCM Manufacturing Indonesia)

Tanti Octavia, Iwan Halim Sahputra
Faculty of Industrial Technology, Department of Industrial Engineering
Petra Christian University, Surabaya
Email: tanti, iwanh@petra.ac.id

Johan Soewanda
Alumni of Faculty of Industrial Technology, Department of Industrial Engineering
Petra Christian University, Surabaya

ABSTRACT

This paper discusses the application of Robust Hybrid Genetic Algorithm to solve a flow-shop scheduling problem. The proposed algorithm attempted to reach minimum makespan. PT. FSCM Manufacturing Indonesia Plant 4's case was used as a test case to evaluate the performance of the proposed algorithm. The proposed algorithm was compared to Ant Colony, Genetic-Tabu, Hybrid Genetic Algorithm, and the company's algorithm. We found that Robust Hybrid Genetic produces statistically better result than the company's, but the same as Ant Colony, Genetic-Tabu, and Hybrid Genetic. In addition, Robust Hybrid Genetic Algorithm required less computational time than Hybrid Genetic Algorithm.

Keywords: robust, hybrid, genetic, algorithm, flow-shop.

1. INTRODUCTION

Income profit and debilitating loss can be significantly affected by job scheduling, machine and man arrangements. Many scheduling algorithms have been proposed and analyzed for finding optimal solutions. Generally, scheduling problem is a NP-hard problem i.e. there are no known algorithms for finding optimal solutions in polynomial time. Algorithms for solving exactly some forms of the problem are available, but they typically take time too long (i.e. more than polynomial time) especially when the problem size grows or when additional constraints are added. Consequently, most research devoted to either simplify the scheduling problem to the point where some algorithms can find solutions, or devise efficient heuristics for finding good solutions.

Yuanita (2006) compared Ant Colony and combination of Genetic Algorithm and Tabu Search for solving flow-shop scheduling problem. El-Bouri (2007) proposed a Hybrid Genetic Algorithm, which combines Genetic Algorithm and NEH algorithm to solve similar problem. Garcia (2007) proposed Robust Genetic Algorithm that considers some new genetic operators, population initialisation, and generation on new population. Robust-Hybrid Genetic Algorithm is a proposed combination of El-Bouri and Garcia.

This paper discusses the comparison between Ant Colony, Genetic-Tabu, Hybrid Genetic and Robust-Hybrid Genetic Algorithm. The real flow-shop scheduling case taken from PT FSCM Manufacturing Indonesia Plant 4 (Pranata, 2006) was used as the case study for finding minimum makespan.

The assumptions considered are 1) working hours are the same, 2) each machine are ready, 3) raw material are always available, 4) ready time for every job is zero, and 4) there is no pre-emption.

2. LITERATURE REVIEW

2.1 Hybrid Genetic Algorithm

Hybrid Genetic Algorithm (El-Bouri, 2007) is a modification of Genetic Algorithm. In this algorithm, two-point crossover operator is combined with NEH (Nawaz, Enscore, and Ham) algorithm to improve the performance of the original Genetic Algorithm. NEH is one of the heuristic algorithms widely used to solve flow shop scheduling problem (Taillard, 1990).

The combination of two-point crossover and NEH algorithm can be seen in Figure 1. Parent 1 chromosome is chosen and then cut on two points. These two points are determined randomly. Then NEH algorithm is applied to arrange the remaining jobs in between of the two points to generate child chromosome based on parent 2 chromosome sequence.

Crossover points

Parent 1:	1	2	3	4	5	6	7
Parent 2:	7	6	5	4	3	2	1
Child:	1	2	6	5	4	3	7

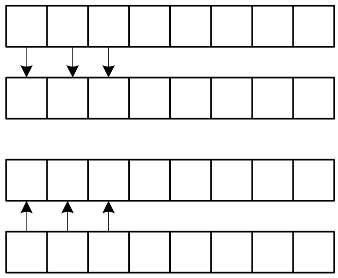
Figure 1. Two-Point Crossover combined with NEH algorithm

2.2 Robust Genetic Algorithm

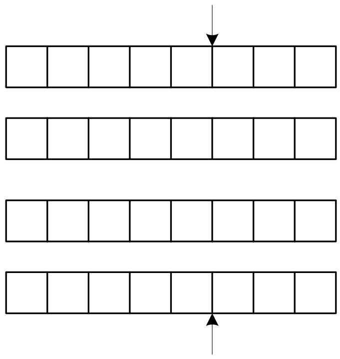
Robust Genetic Algorithm, developed by Garcia (2007), uses some operators as follows:

- Initial population setting
  Initial population is generated using NEH Algorithm and generated randomly.
- Generation selection
  Offspring will be selected for the new population if it has better make-span than the worst chromosome in the population.
- Crossover operator
  Crossover operator used is SBOX. The explanation of this crossover can be seen in Figure 2 as follows:
- Mutation using shift operator. The shift operator simply shifted a random element, a random number of places, to the right or to the left.
- Restart mechanism
  This mechanism will be performed when after several generations there is improvement in local optimum. Restart mechanism will retain best 20% member of population, and the remaining 80% member will be replaced by:
  - 50% is being mutated using shift operator
  - 25% is replaced by NEH modified solution
  - 25% is replaced by fully random chromosome

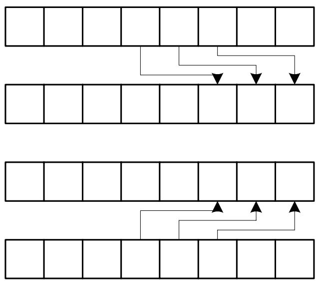
  The goal of the mechanism is to make the divergence of population therefore it can result in better local optimum
- Enhancement Probability
  Enhancement is performed by using random process after the process of selection, crossover, and mutation has been done. One job is selected randomly then it will be placed on every position in the initial job sequence. When better makespan is obtained, the processes will be repeated, start from random selection, then place the job from position 1 until position n. This process will be repeated until there is no better make-span is obtained.



(a) The same jobs of both parents are transferred into offspring.



(b) The jobs on the left side of cutting point are transferred to the offspring.



(c) The remain jobs then are transferred to the offspring.

Figure 2 Steps of SBOX cross over

The fitness value is obtained from:

$$f_i = \frac{(m_w - m_i)^2}{\sum_{i=1}^{N_{pop}} (m_w - m_i)^2} \qquad (1)$$

where:

m_i = makespan of the chromosome i

m_w = the best makespan of the chromosome in the population

3. DEVELOPMENT OF HYBRID GENETIC ALGORITHM PROGRAM

All of the tests were run using a single implementation of hybrid genetic algorithm. Each test problem consisted of 120 jobs including type of product and number of unit would be produced. The hybrid genetic was run on the following sets of parameters:

- Number of population: 25
- Number of generation: 200
- Fitness value: minimum makespan.
- Fitness function:

$$C_{max} = \max\left(C_j\right) \qquad j = 1, 2, ..., n \tag{2}$$

where C_{max} = make-span of n job.

The Initial population was set on the following sets of chromosomes:

- 1 chromosome of NEH solution.
- 1 chromosome of job sequence sorted descending.
- 1 chromosome of job sequence sorted ascending
- 22 chromosome of randomly job sequence

The program was terminated after the generation had reached 200 iterations. The Output of the program was a job sequence which has the smallest makespan. The implementation program was written in Microsoft Visual Basic 6.0.

4. DEVELOPMENT OF ROBUST-HYBRID GENETIC ALGORITHM AND THE PROGRAM

Modification had been made to improve the performance of the Hybrid Genetic Algorithm. The modification was developed based on Robust Genetic Algorithm; therefore this algorithm was called Robust-Hybrid Genetic Algorithm (RHGA). Each test problem consisted of 120 jobs including type of product and number of unit would be produced. The hybrid genetic was run on the following sets of parameters:

- Number of population: 20
- Number of generation: 2000
- Type of crossover: Similar Block Point Order Crossover (SBOX)
- Type of mutation: Shift
- Probability of mutation : 0.01
- Enhancement Probability: 0.05
- Restart: 25

The initial population in RHGA was set as follows:

- 1 chromosome of NEH solution.
- 1 chromosome of series of the jobs descending in time.
- 1 chromosome of series of the jobs ascending in time.
- 17 chromosome of series of the jobs randomly.

In RHGA, enhancement process and restart mechanism were applied. This program was terminated when number of generation had reached 2000 and the output was a sequence of the job which has lowest makespan.

5. RESULTS AND DISCUSSION

Before running the program to solve all the case problems, makespan from manual calculation was compared with the output of the program to validate the program. We found that the results were the same; therefore we could conclude that the program can produce a valid output.

5.1 Output of Hybrid Genetic Algorithm

We run each algorithm program ten times then we compared the results of all algorithms. The comparison can be seen in the Table 1. Data of makespan of Genetic & Tabu was produced by Yuanita (2006) and data of makespan of Ant Colony and company's algorithm were taken from Pranata (2006)

Table 1. Comparison of Hybrid Genetic Algorithm with Other Algorithms

Replication	Makespan (hours)			
	Hybrid Genetic Algorithm (HGA)	Genetic & Tabu	Ant Colony	Company's Algorithm
1	211.888	211.020	212.965	226.341
2	195.248	203.126	208.134	222.133
3	211.406	210.940	213.602	225.678
4	203.446	204.152	209.399	241.858
5	211.692	212.259	214.054	223.264
6	219.253	216.988	217.482	237.874
7	207.692	205.547	209.925	233.786
8	212.393	211.159	214.271	226.021
9	214.537	212.475	213.607	223.772
10	207.645	207.958	211.303	230.427
Average	209.520	209.562	212.474	229.115

Based on makespan of four algorithms, statistic tests were applied by using Minitab 14. Variance testing with $\alpha = 5\%$ was performed in order to compare variance of two populations. The example of variance hypothesis testing was:

H_o : σ^2 Hybrid Genetic Algorithm = σ^2 Genetic & Tabu Search

H_1 : σ^2 Hybrid Genetic Algorithm $\neq$ σ^2 Genetic & Tabu Search

The result shows that there was no difference between variance of two populations. Subsequently, the mean of four algorithms were compared for investigating the mean difference. The Paired t test with 95% significance level and with $\alpha = 5\%$ was applied and the hypothesis was in the following:

H_o : μ Hybrid Genetic Algorithm $\geq$ μ Genetic & Tabu Search

H_1 : μ Hybrid Genetic Algorithm $<$ μ Genetic & Tabu Search

The result shows that there was no significantly difference between make span of HGA and Genetic & Tabu Search Algorithm. The make span of Genetic & Tabu Search was more efficient than hybrid as 0.02%. The mean of HGA and Genetic & Tabu Search Algorithm can be seen in Figure 3.

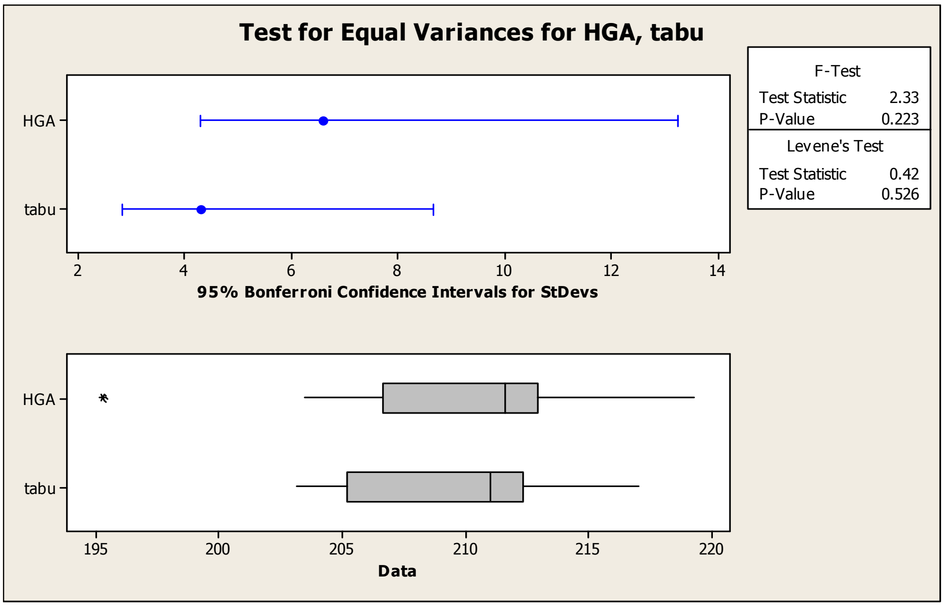


Figure 3. Makespan of Hybrid Genetic Algorithm and Genetic & Tabu Search Algorithm

Testing hypothesis was applied for evaluating the performance of HGA. The result shows that the capability of HGA was not good enough compare to the other Algorithms in term of makespan. In term of the computational time, HGA gave the worst result. The computational time of Hybrid Genetic Algorithm can be seen in Figure 4.

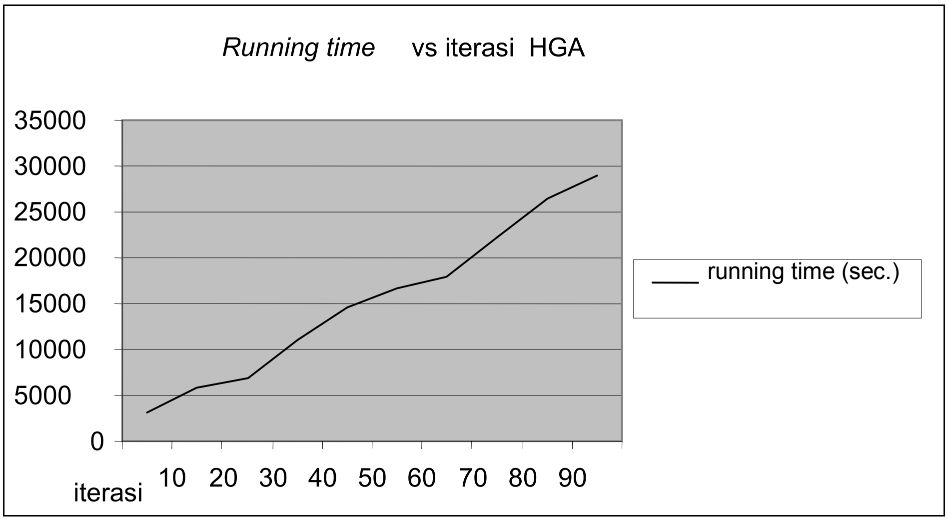


Figure 4. Plot between Running Time and Iteration of Hybrid Genetic Algorithm

5.2 Output of Robust-Hybrid Genetic Algorithm

We run each algorithm program ten times then we compared the results of those algorithms. The comparison can be seen in the Table 2. Data of makespan of Genetic & Tabu was produced by Yuanita (2006) and data of makespan of Ant Colony and company's algorithm were taken from Pranata (2006).

Table 2. Comparison of Robust Hybrid Genetic Algorithm with Other Algorithms

Replication	Makespan			
	Robust-HGA	**Genetic- Tabu**	**Ant Colony**	**Company's Algorithms**
1	211.888	211.020	212.965	226.341
2	195.248	203.126	208.134	222.133
3	211.406	210.940	213.602	225.678
4	203.445	204.152	209.399	241.858
5	211.692	212.259	214.054	223.264
6	219.253	216.988	217.482	237.874
7	209.918	205.547	209.925	233.786
8	212.392	211.159	214.271	226.021
9	214.536	212.475	213.607	223.772
10	207.645	207.958	211.303	230.427
Average	209.742	209.562	212.474	229.115

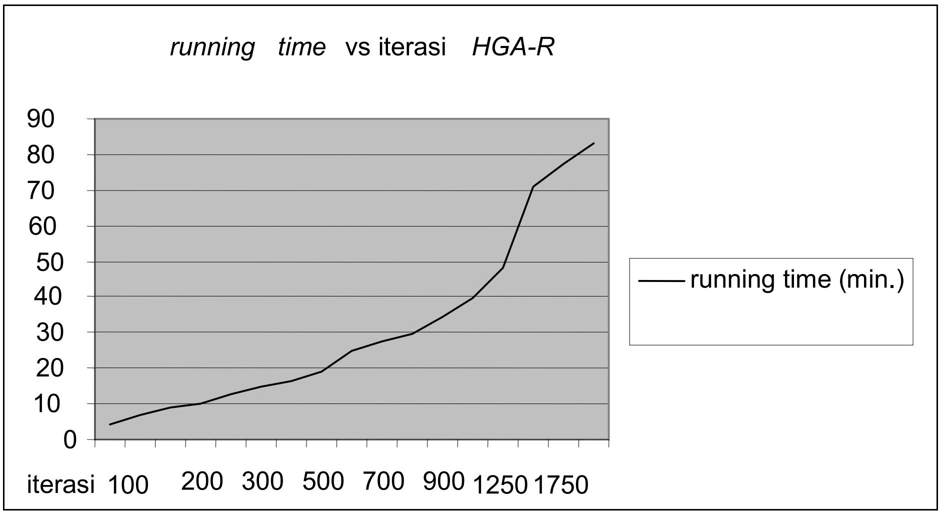


Figure 5. Plot between Running Time and Iteration of Robust Hybrid Genetic Algorithm

From the result of mean test with α = 5%, Robust HGA was only significantly better than company's algorithm. It seems that mutation operator, restart scheme, and enhancement did not significantly affect the performance of Robust HGA. In term of computational time, Robust HGA produced significantly better result compare to HGA. It can be seen in figure 4 and 5, for example, to perform 50 iterations; Robust HGA needed only less than 4.25 minutes of computational time and for HGA needed 15,000 minutes (about 4 hours) of computational time.

6. CONCLUSION AND FURTHER WORK

After running and evaluating the algorithms, we can conclude that

- Hybrid Genetic Algorithm produced statistically better makespan than the company's algorithm. The Hybrid Genetic Algorithm did not produce a better makespan statistically than other three algorithms.
- The result of Robust-Hybrid Genetic Algorithm was not different statistically with the Hybrid Genetic Algorithm. For this case study, operator mutation, restart scheme, and enhancement did not have significant impact on the improvement result.
- Hybrid Genetic Algorithm required longer computational time than Robust-Hybrid Genetic Algorithm to perform the same iteration.

For the further work, Hybrid Genetic Algorithm can be modified using other crossover operator and combined with other algorithm.

REFERENCES

El-Bouri, Ahmed. "A Hybrid Genetic Algorithm for Flowshop Scheduling." Retrieved 2007 from *http://www.Umoncton.ca/cie/conferences/29thconf/29thICCIE/Papers/paper043.PDF*

Garcia, Ruben Ruiz. "Two New Robust Genetic Algorithms for the Flowshop Scheduling Problem." Retrieved 2007 from *http://www.upv.es/gio/rruiz/files/GARM.pdf*

Nawaz, M.; Enscore, E.; and Ham, I. 1983, "A heuristic algorithm for the m-machine, n-job ow-shop sequencing problem", OMEGA, *The International Journal of Management Science* 11/1:91-95.

Pranata, Ong Lina, 2006. *Penjadwalan Produksi dengan Algoritma Ant Colony di PT FSCM Manufacturing Indonesia Plant 4*, Tugas Akhir Universitas Kristen Petra.

Taillard, E. 1990, "Some Eficient Heuristic Methods for the Flow Shop Sequencing Problem", *European Journal of Operations Research* 47:65-74.

Yuanita, Fena, 2006. *Analisa Perbandingan Algoritma Kombinasi Genetic & Tabu Search dengan Metode Ant Colony pada Permasalahan Penjadwalan Flowshop*, Tugas Akhir Universitas Kristen Petra.