International Conference on Electrical Engineering, Informatics, and Its Education 2015

Differential Evolution Algorithm For Multi-Objective Flow Shop Scheduling Problem

Rudi Nurdiansyah
Jurusan Teknologi Industri
Universitas Negeri Malang
Malang, Indonesia
rudy_nurdian@yahoo.co.id

***Abstract*— This paper considers the flow shop scheduling problem with respect to the both objectives of makespan and total flow time. Several algorithms have been proposed to solve this problem. This paper presents Differential Evolution algorithm to solve this scheduling problem. The proposed algorithm is tested with well-known problems in literature. Its solution performance was compared with HAMC Algorithms and Genetic Algorithm. The computational results show that proposed algorithm is better than other methods campared.**

***Keywords*— flow shop scheduling; multi-objective; makespan; total flow time; differential evolution**

I. Introduction

The flow shop scheduling has been a very widely studied for the past 60 years since it was introduced by Johnson in 1954. The problems of flow shop scheduling have the following characteristics, the jobs are processed in the same order at least in one machine and one machine can process maximumly one job at one point in time. The objective of this problem mostly focuses to minimize the total completion time, that is makespan. Additionally, objectives such as total flow time, tardiness, idle time are also considered [1].

The flow shop scheduling problem is a combinatorial optimization problem due to the increasing of job's number and the number of machines [2]. Combinatorial optimization problem is NP-hard and near optimum solution techniques are preferred to solve [3]. In recent years, metaheuristic approaches such as Simulated Annealing (SA), Tabu Search (TS), Ant Colony Optimization (ACO), Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Differential Evolution (DE), and Artificial Immune Systems (AIS) are very desirable to solve combinatorial optimization problems regarding to their computational performance [1]. Metaheuristics solution is a good solution to schedule the flow shop which already proven by [1] using Ant Colony Optimization; [4] using Simulated Annealing; [5] with Tabu Search, [6] with Particle Swarm Optimization; [7] and [8] with Genetic Algorithm, [9] with Artificial Immune Systems.

Most studies on flow shop scheduling focus on the minimizing of makespan. In fact, many other purposes besides makespan can be considered as the objective similar to the total flow time which is also a measure of essential work in minimizing total scheduling cost. While makespan minimization leads to total production run utilization, flow time minimization results in stable consumption of resources, rapid turn-around of jobs and work-in-process inventory minimization [3]. Therefore, this research will consider 2 objectives, the minimize of makespan and the total flow time. In order to reduce the production cost, those two objectives will be considered simultaneously. Some studies using those two objectives are [10] using Ant Colony Optimization, [11] with Particle Swarm Optimization, [12] with Simulated Annealing and [13] with Genetic Algorithm.

One of the metaheuristic algorithm which become the effective global optimization method is Differential Evolution (DE). DE is an effective evolutionary algorithm for continuous complex optimization problems [14]. DE firtsly used [14] to solve Chebychev polynomial fitting problem and it is effective. Furthermore, DE has been widely used to solve various cases, conducted by [15], [16], and [17]. The special quality of DE is the simplicity of the concept, the easy implementation, and the fast converging [14]. The use of DE algorithm to solve the combinatorial optimization problems is still limited [18]. Along with the success of [19] in applying DE to combinatorial optimization problems in flow shop scheduling, it becomes a proof that DE is able to solve the problems of scheduling [18]. [20] and [21] also successfully completed the combinatorial optimization problems using DE.

This research uses Differential Evolution Algorithm to solve the multi objective flow shop scheduling problems with m-machines by considering two objectives, makespan and total flow time. The performance of algorithm proposed in this research will be compared to the algorithm conducted by [22] and GA in paper [3]. Computational experiments are conducted on the benchmark problems from [2] as the test problem in order to verify the algorithm's performance.

II. Problem Formulation

The flow shop scheduling is characterized by the presence of work flow in one and regular direction, also few machines arranged in series. In the process, every job must go through all machines with the same order. The description of scheduling model according to [23] is as follows: a set of M machines M = {1,2,...,m} which is used to process a bunch of jobs N, N = {1,2,...,n}. At one point of time, each machines is able to process only one job and ech job is only processed by one machine. Every job in the operation phase i, processed only once in one machine.

The following notations :

$t(i, j)$ processing time for job i on machine j ($i = 1,2,...,n$), ($j = 1,2,...,m$)

n total number of jobs to be schedule

m total number of machines in the process

π_i the job sequenced in ith position of a schedule

$C(\pi_i, j)$ the completion time of job π_i on machine j an arbitrary sequence of n jobs

S an arbitrary sequence of n jobs

w_1 weight associated with makespan ($0 \le w_1 \le 1$)

w_2 weight associated with flow time ($0 \le w_2 \le 1$)

$M(S)$ the makespan criterion of S

$F(S)$ the flow time criterion of S

$TMF(S)$ objective function (the weighted combination of makespan and flow time of S)

$$C(\pi_i, j) = max\{C(\pi_{i-1}, j), C(\pi_i, j-1)\} + t(\pi_i, j)$$

$$i = 2, \ldots, n; \; j = 2, \ldots, m$$

$$M(S) = C(\pi_n, m)$$

$$F(S) = \sum_{i=1}^{n} C(\pi i, m)$$

$$TMF(S) = w_1 . M(S) + w_2 . F(S)$$

III. Proposed Algorithm

The algorithm used in this research is the Differential Evolution (DE). Differential Evolution is one of the latest metaheuristic method introduced by [14]. The use of Differential Evolution in engineering is quite broad. Since it was proposed in 1995, DE get a good reputation as the effective global optimizer [24].

DE is an algorithm based on the generation of the population. Target population is mutated by a factor mutants to form a population of mutants. In forming the mutant population, there is an F parameter that controls the rate of mutant growing population. Furthermore, the crossover operator combines the population of mutant with the population of target to generate the experimental population. The crossover operator in combining the mutant population with the target population depend on a parameter Cr. The selection operator compare the value of the fitness function between the experimental population with the population of target. At the end, the best individuals will become the member of the next generation. This process is repeated until the convergence is achieved.

The DE algorithm steps are as follows:

A. *Initializaztion*

Before performing the initialization to the population point, it is necessary to determine the upper limit (ub) and lower limits (lb). For the initial value generation g=0, variable –j and the –i vector can be represented by the follwowing notation:

$$x_{j,i,0} = lb_j + rand_j(0, 1)(ub_j - lb_j) \quad (1)$$

The random number is generated by the rand function with the result number is between (0,1). The result obtained from the initialization is the continuous numbers, while the flow shop scheduling problem is a discrete problem. So to change the continuous values into discrete, the result of the initialization is given with an index of the job being resolved. To get the initial solution, the initialization population results are sorted using the procedure of SPV (Smallest Position Value).

B. *Mutation*

After the initialization, DE perform mutation and combination to the population target to generate the trial population size N vector. The mutation is done by adding the difference of two vectors (taken randomly) to a third vector by:

$$v_{i,g} = x_{r0,g} + F(x_{r1,g} - x_{r2,g}) \quad (2)$$

The equation shows that the difference of two vectors are drawn at random, first scaled before being added to the third vector, $x_{r0,g}$. The scale factor $F \in (0,1)$ worth of positive real function is to control the population growth rate. Although there is no upper limit value F, the effective value is between 0 and 1. The index basis vectors, r0 can be determined in various ways. But it is assumed that the index r0 is determined randomly. In addition to the vector indices different from each other and different from the index of basis vectors and the target vector, the index difference vector (the difference between $x_{r1,g}$ and $x_{r2,g}$) is also selected.

C. *Crossover*

To complete the strategy of differential mutation, DE perform uniform crossover. Each of the target vector xi, g, is crossovered with each mutant vectors vi, g, forming trial vector ui, g with the formulation:

$$u_{i,g} = u_{j,i,g} = \begin{cases} v_{j,i,g} & if(rand_j(0,1) \le Cr, \quad or \quad j = j_{rand}) \\ x_{j,i,g} & else \end{cases} \quad (3)$$

Crossover probability, $Cr \in [0,1]$ is the number to control the variable faction value from the mutant vector.

D. Selection

If the trial vector $u_{i,g}$,has smaller objective function than its target objective vector $x_{i,g}$, then $u_{i,g}$ will replace the position of $x_{i,g}$ in population in the next generation. Otherwise, the target will remain in its position in the population.

$$x_{i,g+1} = \begin{cases} u_{i,g} & if(f(u_{i,g}) \le f(x_{i,g}) \\ x_{i,g} & else \end{cases} \quad (4)$$

E. Stopping Criteria

The procedure is repeated until the maximum number of iterations used as a termination criterion being reached.

IV. Computational Experiments

In this section, the results of computational experiments performed are presented to evaluate performance of DE. The computer programs of the proposed algorithm were developed in Matlab 7.8 and implemented on Intel Core Duo 1.66 GHz system with 1024 MB DDR-2 RAM.

DE was tested on 28 benchmark problems with 20 jobs and the number of machines varying from 5 to 20 given by [2]. The test problems can be downloaded from OR-library web site (URL:http://people.brunel.ac.uk/~mastjjb/jeb/info.html; Accessed 2 Sept. 2015). Performance of solutions to yield using test problem is compared with HAMC algorithms (HAMC1, HAMC2, HAMC3) proposed by [22] and GA [3]. The parameters of proposed algorithm in this study defined as follows: number of populations = 100 ; F = 0,5 ; Cr = 0,9 and number of iterations = 1000. Every test was repeated with 10 runs for each instance and the best solution was selected. Hence, there were 280 runs in total. Equal relative weightings chosen to the makespan and total flow time for total objective value were respectively $w_1 = w_2 = 0,5$. The relative percentage increase in total objective value for schedule S generated by any algorithm are given as follows [3]:

$$RE(S) = w_1 . \left(\frac{M(S) - \min(M(S))}{\min(M(S))} \right) + w_2 . \left(\frac{F(S) - \min(F(S))}{\min(F(S))} \right) \quad (5)$$

The relative percentage of objective makespan, total flow time and multi objective both in the case of 20 jobs can be seen in Table 1-3. The table shows that the DE algorithm used in this study has a better performance than the algorithms HAMC1, HAMC2, HAMC3 and GA. Figure 1-3 shows the average results of algorithm performance.

TABLE I. PERFORMANCE SOLUTIONS OF ALGORITHMS FOR MAKESPAN OBJECTIVE

Problem Number	*n x m*	HAMC1	HAMC2	HAMC3	GA	DE
ta001	20 x 5	1,49	3,60	2,27	0,00	0,00
ta002	20 x 5	0,00	2,62	2,62	0,00	0,00
ta003	20 x 5	5,79	6,14	6,14	0,00	1,57
ta004	20 x 5	4,71	6,27	5,90	0,00	0,85
ta005	20 x 5	2,07	6,12	6,12	1,61	0,65
ta006	20 x 5	0,00	3,47	3,47	2,99	1,26
ta007	20 x 5	3,28	6,17	4,06	0,00	0,97
ta008	20 x 5	1,74	10,99	10,99	4,35	0,00
ta009	20 x 5	0,31	6,06	6,06	2,30	0,00
ta010	20 x 5	1,36	10,09	3,56	1,95	0,00
ta011	20 x 10	6,16	8,44	6,94	0,00	1,39
ta012	20 x 10	2,34	3,83	4,69	4,17	0,96
ta013	20 x 10	5,59	14,65	14,59	3,15	1,20
ta014	20 x 10	2,68	6,98	6,24	0,20	1,60
ta015	20 x 10	15,57	4,50	6,44	1,54	0,63
ta016	20 x 10	4,61	8,63	8,18	0,00	1,65
ta017	20 x 10	0,95	2,85	3,10	0,00	1,08
ta018	20 x 10	2,19	3,95	3,95	2,82	2,47
ta019	20 x 10	5,74	10,10	8,97	0,00	1,38
ta020	20 x 10	1,45	6,52	4,30	1,75	1,51
ta021	20 x 20	2,51	4,49	4,77	0,45	2,48
ta022	20 x 20	11,96	11,96	16,22	0,00	0,67
ta023	20 x 20	0,46	0,91	3,94	7,13	1,20
ta024	20 x 20	11,41	16,88	18,14	0,00	1,53
ta025	20 x 20	0,00	1,36	3,02	0,21	0,70
ta026	20 x 20	9,04	12,62	13,56	0,17	1,39
ta027	20 x 20	12,60	16,28	15,69	10,40	1,85
ta028	20 x 20	5,12	5,50	5,46	0,00	0,95

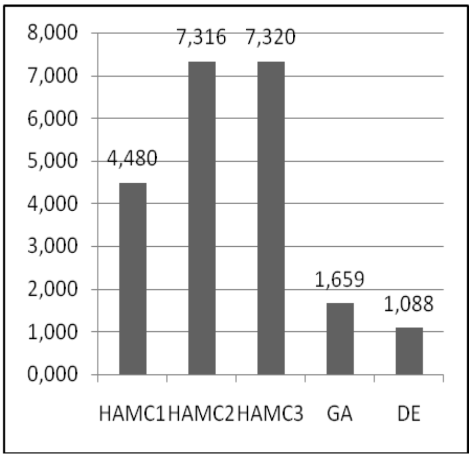

Fig. 1. Average relative error in makespan

TABLE II. PERFORMANCE SOLUTIONS OF ALGORITHMS FOR TOTAL FLOW TIME OBJECTIVE

Problem Number	*n x m*	HAMC1	HAMC2	HAMC3	GA	DE
ta001	20 x 5	1,16	0,28	0,59	0,00	0,00
ta002	20 x 5	7,42	0,27	0,27	1,40	0,00
ta003	20 x 5	2,94	2,50	2,50	1,17	0,00
ta004	20 x 5	3,31	1,28	1,30	0,53	0,00
ta005	20 x 5	8,78	0,55	0,61	0,00	0,00
ta006	20 x 5	3,47	0,00	0,18	2,32	0,00
ta007	20 x 5	2,67	0,79	0,82	1,11	0,00
ta008	20 x 5	2,03	1,06	1,13	0,73	2,11
ta009	20 x 5	7,91	1,17	1,17	1,87	0,00
ta010	20 x 5	3,57	0,00	0,21	1,55	1,11
ta011	20 x 10	5,29	4,23	4,43	0,80	0,11
ta012	20 x 10	2,31	0,31	0,50	0,00	3,35
ta013	20 x 10	8,22	2,07	2,22	0,00	0,00
ta014	20 x 10	3,51	1,81	2,10	2,87	0,34
ta015	20 x 10	4,09	8,15	2,63	0,00	2,84
ta016	20 x 10	2,31	0,71	1,20	0,00	0,86
ta017	20 x 10	2,94	0,00	1,46	1,78	0,00
ta018	20 x 10	4,81	1,71	2,15	1,45	0,39
ta019	20 x 10	4,46	0,73	0,86	0,48	2,08
ta020	20 x 10	1,29	0,00	0,15	1,36	2,38
ta021	20 x 20	5,59	2,31	3,05	0,96	0,00
ta022	20 x 20	4,46	4,46	7,65	0,31	0,00
ta023	20 x 20	1,99	0,00	1,55	5,51	3,20
ta024	20 x 20	5,04	10,28	11,33	0,00	0,00
ta025	20 x 20	5,52	0,00	1,46	6,43	0,00
ta026	20 x 20	11,51	5,53	6,58	1,72	0,00
ta027	20 x 20	3,91	0,93	1,18	1,50	0,00
ta028	20 x 20	1,48	1,22	1,28	0,54	0,00

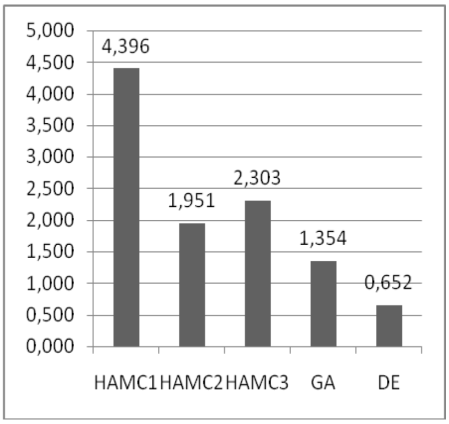

Fig. 2. Average relative error in total flow time

TABLE III. PERFORMANCE SOLUTIONS OF ALGORITHMS FOR MULTIPLE OBJECTIVES

Problem Number	*n x m*	HAMC1	HAMC2	HAMC3	GA	DE
ta001	20 x 5	1,19	0,56	0,73	0,00	0,00
ta002	20 x 5	6,72	0,38	0,38	1,34	1,18
ta003	20 x 5	1,95	1,57	1,57	0,00	1,02
ta004	20 x 5	3,10	1,36	1,35	0,40	1,10
ta005	20 x 5	7,99	0,85	0,90	0,00	0,00
ta006	20 x 5	2,87	0,00	0,16	2,12	1,10
ta007	20 x 5	2,43	0,96	0,81	0,90	1,63
ta008	20 x 5	1,76	1,64	1,70	1,15	1,51
ta009	20 x 5	6,79	1,10	1,10	1,50	0,05
ta010	20 x 5	3,04	0,49	0,15	1,30	1,91
ta011	20 x 10	5,18	4,36	4,44	0,86	1,00
ta012	20 x 10	1,74	0,00	0,23	0,00	2,21
ta013	20 x 10	7,45	2,41	2,55	0,16	0,00
ta014	20 x 10	3,07	1,80	2,02	2,40	0,97
ta015	20 x 10	4,66	7,61	2,65	0,00	1,89
ta016	20 x 10	2,35	1,17	1,58	0,00	1,93
ta017	20 x 10	2,56	0,00	1,37	1,62	0,00
ta018	20 x 10	4,15	1,41	1,82	1,37	1,59
ta019	20 x 10	4,11	1,00	1,04	0,16	1,73
ta020	20 x 10	0,84	0,03	0,00	1,16	2,26
ta021	20 x 20	5,31	2,38	3,08	1,02	2,08
ta022	20 x 20	4,64	4,64	7,88	0,34	0,00
ta023	20 x 20	1,83	0,00	1,65	5,67	2,42
ta024	20 x 20	5,27	10,52	11,58	0,00	0,76
ta025	20 x 20	5,04	0,00	1,47	6,47	0,00
ta026	20 x 20	11,30	5,96	7,00	1,67	0,09
ta027	20 x 20	3,73	1,18	1,37	1,51	1,67
ta028	20 x 20	1,50	1,29	1,34	0,55	0,29

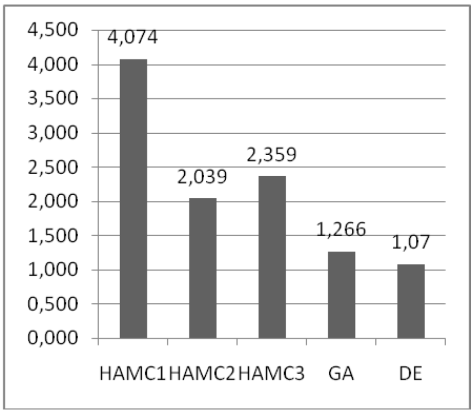

Fig. 3. Average relative error in multi-objective

V. Conclusion

Most studies of the flow shop scheduling focus only on a single criterion. In fact, an approach considering multiple criteria are needed. Therefore, this research completed the flow shop scheduling problem with multi objective of makespan and total flow time.

In this paper, the proposed algorithm is Differential Evolution algorithm. To see the performance of the algorithm, a testing of benchmark problems is conducted. The results of this research indicate that the DE algorithm proposed in this study has a better performance than the algorithms HAMC1, HAMC2, HAMC3 and Genetic Algorithm for objective makespan, total flow time and multi objective.

Algorithms DE in this research can also be tested on a single objective and multi-objective by considering other objective such as mean flow time, total tardiness and maximum tardiness. For the next, DE can also be applied to scheduling problems in several other manufacturing systems such as job shop, cellular manufacturing and flexible manufacturing.

References

[1] Yagmahan, Betul, and M.M. Yenisey, "Ant colony optimization for multi-objective flow shop scheduling problem," Computers & Industrial Engineering 54.3 (2008): 411-420.

[2] Taillard and Eric, "Benchmarks for basic scheduling problems," european journal of operational research 64.2 (1993): 278-285.

[3] Yagmahan, Betul, and M.M. Yenisey, "A multi-objective ant colony system algorithm for flow shop scheduling problem," Expert Systems with Applications 37.2 (2010): 1361-1368.

[4] Ishibuchi, Hisao, S. Misaki, and H. Tanaka, "Modified simulated annealing algorithms for the flow shop sequencing problem," European Journal of Operational Research 81.2 (1995): 388-398.

[5] Nowicki, Eugeniusz, and C. Smutnicki, "Some aspects of scatter search in the flow-shop problem," European Journal of Operational Research 169.2 (2006): 654-666.

[6] Lian, Zhigang, X. Gu, and B. Jiao, "A similar particle swarm optimization algorithm for permutation flowshop scheduling to minimize makespan," Applied Mathematics and Computation 175.1 (2006): 773-785.

[7] Ruiz, Rubén, C. Maroto, and J. Alcaraz, "Two new robust genetic algorithms for the flowshop scheduling problem," Omega 34.5 (2006): 461-476.

[8] Kordoghli and Bessem, "Multi-objective Scheduling of Flow-Shop Problems in Finishing Factory Using Genetic Algorithms," Journal of Textile and Apparel, Technology and Management 6.3 (2010).

[9] Gao, Haichang, and X. Liu, "Improved artificial immune algorithm and its application on the permutation flow shop sequencing problems," Information Technology Journal 6.6 (2007): 929-933.

[10] Rajendran, Chandrasekharan, and H. Ziegler, "Ant-colony algorithms for permutation flowshop scheduling to minimize makespan/total flowtime of jobs," European Journal of Operational Research 155.2 (2004): 426-438.

[11] M.F. Tasgetiren, Y.C. Liang, M. Sevkli, and G. Gencyilmaz, "A particle swarm optimization algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing problem," European Journal of Operational Research 177.3 (2007): 1930-1947.

[12] T.K. Varadharajan and C. Rajendran, "A multi-objective simulated-annealing algorithm for scheduling in flowshops to minimize the makespan and total flowtime of jobs," European Journal of Operational Research 167.3 (2005): 772-795.

[13] T. Pasupathy, C. Rajendran, and R.K. Suresh, "A multi-objective genetic algorithm for scheduling in flow shops to minimize the makespan and total flow time of jobs," The International Journal of Advanced Manufacturing Technology 27.7-8 (2006): 804-815.

[14] Storn, Rainer, and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," Journal of global optimization 11.4 (1997): 341-359.

[15] Ilonen, Jarmo, J.K. Kamarainen, and J. Lampinen, "Differential evolution training algorithm for feed-forward neural networks," Neural Processing Letters 17.1 (2003): 93-105.

[16] Chang, F. Pi, and C. Hwang, "Design of digital PID controllers for continuous time plants with integral performance criteria," Journal of the Chinese Institute of Chemical Engineers 35.6 (2004): 683-696.

[17] Chang, Y. Pin, and C.J. Wu, "Optimal multiobjective planning of large-scale passive harmonic filters using hybrid differential evolution method considering parameter and loading uncertainty," Power Delivery, IEEE Transactions on 20.1 (2005): 408-416.

[18] Pan, Q. Ke, M.F. Tasgetiren, and Y.C. Liang, "A discrete differential evolution algorithm for the permutation flowshop scheduling problem," Computers & Industrial Engineering 55.4 (2008): 795-816.

[19] M.F. Tasgetiren, Y.C. Liang, M. Sevkli., and G. Gencyilmaz, "Differential evolution algorithm for permutation flowshop sequencing problem with makespan criterion," Proceedings of the 4th international symposium on intelligent manufacturing systems (IMS 2004), Sakarya, Turkey. 2004.

[20] B. Qian, L. Wang, R. Hu, W.L. Wang, D.X. Huang, and X. Wang, "A hybrid differential evolution method for permutation flow-shop scheduling," The International Journal of Advanced Manufacturing Technology 38.7-8 (2008): 757-777.

[21] Mingyong, Lai, and C. Erbao, "An improved differential evolution algorithm for vehicle routing problem with simultaneous pickups and deliveries and time windows," Engineering Applications of Artificial Intelligence 23.2 (2010): 188-195.

[22] D. Ravindran, S.J. Selvakumar, R. Sivaraman, and A.N. Haq, "Flow shop scheduling with multiple objective of minimizing makespan and total flow time," The international journal of advanced manufacturing technology 25.9-10 (2005): 1007-1012.

[23] S.R. Hejazi and S. Saghafian, "Flowshop-scheduling problems with makespan criterion: a review." International Journal of Production Research 43.14 (2005): 2895-2929.

[24] Santosa, Budi, and P. Willy, "Metoda Metaheuristik Konsep dan Implementasi." Surabaya: Guna Widya (2011).