Plagiarism Checker X Originality Report

Similarity Found: 11%

1 Car painting process scheduling with harmony search algorithm M F Syahputra*1, A Maiyasya2, S Purnamawati3, D Abdullah4, W Albra5, M Heikal6, A Abdurrahman7 , M Khaddafi8 1,2,3 Department of Information Technology, Faculty of Computer Science and Information Technology, University of Sumatera Utara, Medan, Indonesia 4Department of Informatics, Universitas Malikussaleh, Aceh, Indonesia 5,6,7 Department of Management, Universitas Malikussaleh, Aceh, Indonesia 8Department of Accounting, Universitas Malikussaleh, Aceh, Indonesia *E-mail: nca.fadly@usu.ac.id Abstract.

Automotive painting program in the process of painting the car body by using robot power, making efficiency in the production system. Production system will be more efficient if pay attention to scheduling of car order which will be done by considering painting body shape of car. Flow shop scheduling is a scheduling model in which the job-job to be processed entirely flows in the same product direction / path.

Scheduling problems often arise if there are n jobs to be processed on the machine, which must be specified which must be done first and how to allocate jobs on the machine to obtain a scheduled production process. Harmony Search Algorithm is a metaheuristic optimization algorithm based on music. The algorithm is inspired by observations that lead to music in search of perfect harmony. This musical harmony is in line to find optimal in the optimization process.

Based on the tests that have been done, obtained the optimal car sequence with minimum makespan value. 1. Introduction Developments in the industrial world increasingly experienced an increase in terms of quality and quantity of production. Starting from using skilled human labor to use the machine in conducting production activities so that the possibility of errors caused by human error (human error).

In developed countries almost all industries do production activities using the power of the engine, so that humans only operate the machine in its production activities. Although already using the power of machinery in conducting its production activities, every industrial company keep trying to make its production activities produce an optimum result. Scheduling is one of the efforts of industrial companies to produce optimum results.

Production scheduling is attempted to obtain an effective job assignment at each work station, so that no job stacking occurs so as to reduce idle time or waiting time for subsequent processing [1]. Car industry factories in developed countries already use the machine in the process of painting the car body. In the painting process has not done scheduling optimally, where in this case there is still waiting time in the process of painting the next stage, if the painting process is done more than one variation of the form of the car in the same painting machine.

Flow shop scheduling is a scheduling model where the job to be processed entirely flows in the same product direction / path. In other words, jobs have routing work together. Scheduling problems 2 often arise if there are n jobs that will be processed on the machine m, which must be specified which must be done first and how to allocate jobs on the machine so that a scheduled production process is obtained [2]. 2.

Problem identification In the process of painting cars with different car shape variations in the same painting machine, there are still a problem in waiting time. Therefore, an optimal solution is required to achieve the effective and efficient scheduling of car sequences in the painting process. How to get effective and efficient car order in the painting process, thus obtaining minimum makepan value in solving flow shop scheduling problems. 3.

Previous Research Scheduling flow shop in the search for the best optimization, has been a lot to solve various kinds of problems in everyday life. From small problems to complex problems with various methods in the settlement. In solving the flow shop scheduling problem about the time of movement of a job that sometimes still has a free time when moving from one machine to another machine [3]. In this case [3] uses ant colony algorithm to solve the problem in finding the smallest makespan time.

Soukhal [4] in his research solve the problem of flow shop scheduling by using polynomial-time algorithm. Problems examined by Soukhal is about the truck carrying goods that will be delivered to the customer by considering the capacity of truck and transport time. Boukef [5] conducts research on the flow shop scheduling problem in

terms of minimizing the amount of prices for the production process and the delivery process undertaken on pharmaceutical drugs and the food industry. Boukef uses the genetic algorithm method to solve the problem.

Aulia [6] researches the problem of flow shop scheduling with permutation constraints on harmony search algorithm. The result of this algorithm can solve the problem well compared with the lower bound. 4. Methodology HSA is a metaheuristic optimization algorithm based on music. The algorithm is inspired by observations that lead to music in search of perfect harmony. This musical harmony is in line to find opitmal in the optimization process.

The search optimization process can be compared to the process of improvising jazz music. On the one hand, the perfection of harmony is determined by aesthetic sound standards. A musician always expects to produce a song with perfect harmony. On the other hand, an optimal solution for an optimization problem will be the best solution for objective and limited issues.

Both of these processes will produce the best or optimum solution [7]. Figure 1. Improvisasion of Music Analogy The analogy between musical improvisation and optimization techniques can be illustrated in figure 1: 3 Each music player (saxophonist, double bassist and guitarist) can be analogous to the variables (x1, x2, x3) and the tone levels of each musical instrument (saxophone = {Do, Re, Mi}; double bass = {Mi, Fa, Sol}; And guitar = {Sol, La, Si}) are analogous to the variables (x1 = {100, 200, 300}; x2 = {300, 400, 500}; and x3 = {500, 600, 700}).

If the saxophonist produces a Re sound, the double bassist emits a sound of the Mi and the guitarist emits a Si sound, so all three will simultaneously create a new harmony (Re, Mi, Si). If this new harmony is better than the previous harmony, then the new harmony will be used. Likewise a new vector solution (200mm, 300mm, 700mm) will be used as well if it is better than the previous one for objective function value conditions [8].

In accordance with the above concept, HSA consists of five stages, namely: A) Initialization of optimization problems and algorithm parameters. B) Initialize harmony memory. C) Improvisation of new harmony. D) Harmony memory update. E) Repeat steps c and d until the stop criteria are met. implementation of harmony search algorithm in flow shop scheduling process can be described as below: A.

Initialization of Optimization Problems and Algorithm Parameters In the first stage, the optimization problem is determined as follows: Minimize (or Maximize) f (x) With ? i,i 1,., f(xan bivfio , Xi: decision variable i, Xi: the set of decision variables, and N: the number of

decision variables Next, determine the parameters needed in the HSA, namely: a) Harmony Memory Size (HMS) HMS is the number of vector solutions in harmony memory.

b) HMMM Memory Considering Rate (HMCR) HMCR is a continuous value used as a harmonized improvisation parameter in HSA. The c) Pitch Adjustment Rate (PAR) PAR is a continuous value used as a harmonized improvisation parameter after the HMCR d) Stop Criteria Stop Criteria is the value used to stop the repetition of the new harmony improvisation. B.

Harmony Memory Initialization In the initialization phase of Harmony Memory, solution vectors will fill the memory harmony according to the number of HMS. The solution vector is generated from randomly generated decision variables. The solution vector that is randomly generated is a sequence of jobs in the flow shop. Then each vector solution will be calculated its mask value. So it will produce the solution vector with the value makespannya respectively.

An example of data that will be solved based on flow shop problems is shown in table 1. 4 Table 1 Example of Time Flow Shop Scheduling Machine/Car 1 2 3 4 5 6 7 1 1 3 5 2 4 5 3 2 2 4 1 3 5 4 2 From the data in table 1, the number of machines is 2 and the number of cars is 7.

So the decision variable that is formed is Xi = {1,2,3,4,5,6,7} After the Xi decision variable is generated, then determine the number of HMS to be formed to accommodate the number of solution vectors in HM. Then the solution vector generation is generated randomly and calculated makespan from each solution vector. C. Improvisation of New Harmony At this stage, the new harmony vector x '= (x'1, x'2, x'3, x'4, ...,

x'N) has three basic parts, namely: A) Memory Consideration The use of memory consideration is very important, this is the same as choosing the best individual in the genetic algorithms. Memory consideration will ensure the best harmony that will be brought to the last new harmony memory. B) Pitch Adjustment The pitch adjustment is fixed by a pitch band-width brange and a pitch adjusting rate rpa.

Although in music, pitch adjustment is a tool to change the frequency, it is suitable for connecting a slight difference solution inside the HSA. In theory, patterns can be arranged linearly or nonlinear, but in practice linear adjustment is used. So it can be concluded xnew = o+ rang* we o the estle r lutioo omoa new is a new pattern after pitch adjusting action.

C) Random Selection Random selection is useful for expanding the diversity of solutions. Although pitch adjustment has the same role, but pitch adjustment is limited to certain local pitch adjustments and so is local search. Random selection can run on a more advanced system to examine a variety of solutions to find the optimal global.

In this research, the improvisation process is described as follows: Iteration 1 The random number generated is a1 = 0,432. Then the random number is adjusted to HMCR = 0.9. Because a1 <HMCR, then the decision variable is chosen randomly. Suppose the selected x1 '= 2. Next generate a random number to be adjusted with PAR = 0.3. Suppose that the random number generated is a2 = 0.512.

Because a2> PAR, then the decision variable x1 '= 2 is retained. Iteration 2 The random number generated is a1 = 0.925. Then the random number is adjusted to HMCR = 0.9. Because a1> HMCR, then the decision variable is chosen randomly from X1. Suppose the selected x1 '= 5. 5 Iteration 3 The randomized number generated from the previous iteration, e.g. a1 = 0.276, will be compared with the value of HMCR = 0.9 .

Because the condition a1 < HMCR is fulfilled, the decision variable will be selected randomly, e.g. x1' = 3. The next step is generating random value to be adjusted by comparing the value with PAR = 0.3 . For example, if the value, defined by variable a2, is 0.114, and the rule a2 < PAR is fulfilled, then the decision variable x1' = 3 will be adjusted to the next variable, which is x2' = 4.

Iteration 4 The random number generated is a1 = 0.456. Then the random number is adjusted to HMCR = 0.9. Because a1 < HMCR, then the decision variable is chosen randomly. Suppose the selected x1 '= 6. Next generate a random number to be adjusted with PAR = 0.3. Suppose the random number generated is a2 = 0.212. Since a2 <PAR, then the decision variable x1 '= 6 is adjusted to the adjacent variable, ie x2' = 7.

Iteration 5 The random number generated is a1 = 0.941. Then the random number is adjusted to HMCR = 0.9. Because a1> HMCR, then the decision variable is chosen randomly from X1. Suppose the selected x1 '= 6. Iteration 6 The random number generated is a1 = 0.672. Then the random number is adjusted to HMCR = 0.9. Because a1 <HMCR, then the decision variable is chosen randomly. Suppose the selected x1 '= 1.

Next generate a random number to be adjusted with PAR = 0.3. Suppose the random number generated is a2 = 0.322. Because a2> PAR, then the decision variable x1 '= 1 is retained. Iteration 7 The random number generated is a1 = 0,731. Then the random number is adjusted to HMCR = 0.9. Because a1 <HMCR, then the decision variable is chosen randomly. Suppose the selected x1 '= 3.

Next generate a random number which will be adjusted with PAR = 0.3. Suppose that the random number generated is a2 = 0.843. Because a2> PAR, then the decision variable x1 '= 3 is retained. From the above improvisation process obtained a new vector solution, namely: X = [2 5 4 7 6 1 3] Furthermore from the sequence of the car, it will be calculated its objective function to know the time. D. Harmony Memory Update After the improvisation process is completed it will get a new car sequence.

Then from the sequence the car is calculated makespan. Having obtained the value of makespan, then the value of makespan is compared with the value of makespan contained in HM. If the makespan obtained at the improvisation stage is smaller than the makespan contained in HM, then the sequence of car and the makespan value in HM will be replaced by the sequence of the car and the new makespan.

However, if the makespan obtained at the improvisation stage is not smaller than the makespan contained in HM, the sequence of cars and makespan in HM will be maintained. 6 E. Stop Criteria Criteria for quitting is a condition in which the improvisation process stops if it meets the stipulation of the stop criteria. However, if the determination of the stop criteria has not been met, then the improvisation process continues to be done repeatedly until it meets the stop criteria.

F. Objective Functions The objective function of this research is using makespan, which is the time period of completion of a job which is the sum of all the processing time of a machine. The smallest makespan is the best use of machine and job combinations and ensures the job from start to finish is completed.

Makespan is calculated using the following equation: The calculation method of make span value is described in table 2. Table 2. Makespan Calculation Machine Car Duration Start time End time 1 2 3 0 3 1 5 4 3 7 1 4 2 7 9 1 7 3 9 12 1 6 5 12 17 1 1 1 17 18 1 3 5 18 23 2 2 4 3 7 2 5 5 7 12 2 4 3 12 15 2 7 2 15 17 2 6 4 17 21 2 1 2 21 23 2 3 1 23 24 From table 2, the car column has been arranged in the order of cars to be processed on each of the 1st and 2nd engines. In the sequence of the 1st car (car 2), the start time starts 0. Since each car is in the order of- 1 must start from minute 0.

To know the start time on the 2nd car sequence (car 5), that is the duration of time and start time on the sequence of car 1 (car 2) summed. So the result is the amount of time starting from the order of the 2nd car (car 5). And so on until the last car. To fill in the end time, the duration of time and start time is added to each car.

For machine 2, the start time of the 1st car sequence (car 2) is taken from the engine

end time 1 of the 1st car sequence (car 2). Then for the start time of the next car sequence, it is obtained from the comparison between the end time of the 1st car sequence (car 2) on the 2nd (j) machine with the end time of the 2nd car sequence (car 5) (i + 1). What is taken is the greatest end time.

So that get the value makespan is the last sequence (car 3) on the order of the 7th car in machine 2, which is 24. 5. Result and Discussion In this study, the data to be used is a combination of engines and the number of cars 2 x 20 based on the time that has been calculated. Testing the data using parameters, namely HMS = 5, HMCR = 0.9 and PAR = 0.3.

The data will be tested 10 times with the number of improvisation 100, 500, 1000, 2000 and 2500. 7 Table 3. Test result Replication Iterations 100 500 1000 2000 5000 1 241.49 241.41 241.86 241.49 240.85 2 242.59 241.44 241.74 241.51 241.37 3 241.16 241.49 241.65 241.36 240.52 4 242.76 240.16 241.3 241.3 241.34 5 242.79 242.13 241.63 241.48 241.26 6 242.75 241.7 241.65 240.61 241.06 7 241.76 241.89 241.8 241.09 240.93 8 241.71 241.92 241.77 241.16 241.15 9 241.28 241.89 241.85 241.72 241 10 242.28 241.76 241.49 240.91 241.25 Makespan 241.16 240.16 241.3 240.61 240.52 From the test with the combination of engine and the number of cars 2 x 20, obtained results on each iteration with the value makespan vary.

So obtained the best makespan value is 240.16 obtained on iteration to 500, with the order of cars 17-16-12-0-5-10-9-13-8-11-4-15-7-6-1- 18-19-2-14-3. 6. Conclusion From the research that has been done can be concluded that, flow shop scheduling on car body painting at basecoat and clearcoat stage by using harmony search algorithm obtained minimum makepan value, that is 240.16 at iteration to 500 with sequence of car, that is 17-16-12-0- 5-10-9-13-8- 11-4-15-7-6-1-18-19-2-14-3.

In the next study it is suggested to add the painting process from the beginning of the painting to the top coat (basecoat and clearcoat) process. And consider the number of machines in each painting process. References [1] Masruroh N 2008 Analysis of production scheduling using ampbell dudeck smith, palmer and dannenbring method at PT.

Loka refractories surabaya Journal of Industrial Engineering and Management 158-171 [2] Ginting R 2006 Study the application of artificial immune system method in flow shop scheduling Jurnal Sistem Teknik Industri 7 pp 5-21 [3] Lamoudan T, Khoukhi F E, & Alaoui A E H 2011 Flow shop scheduling problem with transportation times, two-robots and input/output with limited capacity International Journal of Intelligent Computing Research 2 pp 244-253 [4] Soukhal A, Oulamara A, & Martineau P 2005 Complexity of

flow shop scheduling problems with transportation constraints European Journal of Operational Research 161 pp 32-41 [5] Boukef H, Benrejeb M, & Borne P 2007 A proposed genetic algorithm coding for flow-shop scheduling problems International Journal of Computers, Communications and Control 2 pp 229-240 [6] Aulia I, Nababan E B, & Muchtar M A 2012 Application of harmony search algorithm in flow shop scheduling problem Jurnal Dunia Teknologi Informasi 1 pp 1-7 [7] Zong Woo Geem (ed.)

2009 Music-Inspired Harmony Search Algorithm Theory and Applications Springer: New York [8] Chakraborty P, Roy G G, Das S, & Jain D 2009 An improved harmony serach algorithm with differential mutation operator Fundamenta Informaticae 95 pp 1-26

INTERNET SOURCES:

<1% - https://mafiadoc.com/poster-presentation-tuberculosis_59a824451723ddbbc5efbb56.html
1% - https://www.researchgate.net/topic/Harmony-Search
<1% - http://www.inderscienceonline.com/doi/full/10.1504/IJISE.2014.060133
<1% - https://www.scribd.com/document/358861665/11-Chapter-3
<1% - https://www.scribd.com/document/201047805/musiccomp-harmonysearch
<1% - https://en.wikipedia.org/wiki/Productivity
<1% - https://farolforum.wordpress.com/category/developed-countries/
<1% - https://en.wikipedia.org/wiki/Automation
<1% - https://link.springer.com/article/10.1007/s11227-015-1438-4
<1% - https://link.springer.com/article/10.1057/jos.2014.40
<1% - https://www.sciencedirect.com/science/article/pii/S095219761200067X
<1% - http://www.academia.edu/1484996/Complexity_of_flow_shop_scheduling_problems_with_transportation_constraints
<1% - http://www.unesco.org/education/tlsf/mods/theme_a/popups/mod01t05s01.html
<1% - http://dl.acm.org/citation.cfm?id=1491047
1% - http://repository.usu.ac.id/bitstream/handle/123456789/42173/Reference.pdf;sequence=1
<1% - https://link.springer.com/article/10.1007/s10462-017-9595-x
<1% - http://www.asa3.org/ASA/education/teach/music.htm
<1% - http://www.gutenberg.org/files/24727/24727-h/24727-h.htm
<1% - https://arxiv.org/pdf/1003.1599
<1% - http://www.softcomputing.net/chakraborty.pdf
<1% - https://www.sciencedirect.com/science/article/pii/S0950584912000134

<1% - http://www.academia.edu/2724470/An_improved_harmony_search_algorithm_with_differential_mutation_operator

<1% - http://www.iaeng.org/publication/WCECS2010/WCECS2010_pp460-465.pdf

<1% - https://www.researchgate.net/publication/220444674_An_Improved_Harmony_Search_Algorithm_with_Differential_Mutation_Operator

<1% - https://www.sciencedirect.com/science/article/pii/S2210650216304965

<1% - http://thescipub.com/PDF/ajassp.2005.1552.1557.pdf

<1% - http://www.tandfonline.com/doi/pdf/10.1080/00207721.2015.1086931?needAccess=true

<1% - https://www.science.gov/topicpages/h/harmony+search+hs.html

<1% - http://www.inderscienceonline.com/doi/full/10.1504/IJMMNO.2013.059192

<1% - https://www.researchgate.net/profile/Nicolaos_Theodossiou/publication/257634720_Application_of_the_Harmony_Search_optimization_algorithm_for_the_solution_of_the_multiple_dam_system_scheduling_Optimization_and_Engineering/links/5601249608aeafc8ac8c8220.pdf

<1% - https://www.researchgate.net/profile/Azadeh_Ghari_Neiat/publication/224320731_An_efficient_algorithm_for_bandwidth-delay_constrained_least_cost_multicast_routing/links/0a85e537282be176b9000000.pdf?origin=publication_detail

<1% - http://home.zcu.cz/~kovarikp/MT/freestudy/FLUID_MECHANICS_D203/FLUID_MECHANICS_D203.pdf

<1% - https://www.statista.com/statistics/299972/average-age-of-cars-on-the-road-in-the-united-kingdom/

<1% - https://www.hindawi.com/journals/tswj/2013/320489/

<1% - https://www.scribd.com/document/168055132/Optimal-Allocation-of-Shunt-Var-Compensators-in-Power-Systems-Using

<1% - https://www.slideshare.net/xinshe/10031599

<1% - http://www.wseas.org/multimedia/journals/mechanics/2012/54-743.pdf

<1% - https://link.springer.com/article/10.1007/s13042-014-0312-7

<1% - https://www.scribd.com/document/272051841/IJEST-Special-Issue-Complete

<1% - http://bioconductor.org/help/workflows/methylationArrayAnalysis/

<1% - http://www.academia.edu/17031403/Vortex_search_algorithm_for_the_analog_active_filt

er_component_selection_problem

<1% -
https://vdocuments.site/documents/a-hybrid-learning-method-for-constructing-compact-rule-based-fuzzy-models.html

<1% - http://www.autohausaz.com/html/vehicle_identification_numbers.html

<1% - https://www.sciencedirect.com/science/article/pii/S0305054816302477

<1% - http://www.pharmpress.com/files/docs/PharmStats.pdf

<1% - https://www.irs.gov/irm/part4/irm_04-010-003

<1% - https://www.sciencedirect.com/science/article/pii/S0957417415003632

<1% - http://eprints.upnjatim.ac.id/view/types/article.html

<1% - https://www.sciencedirect.com/science/article/pii/S1474667015334200

1% -
http://www.rroij.com/open-access/hybrid-flowshop-scheduling-usingdiscrete-harmony-search-and-geneticalgorithm-.php?aid=49980