Journal of Information Technology and Computer Science
Volume 4, Number 1, June 2019, pp. 26-32
Journal Homepage: www.jitecs.ub.ac.id

Using Guided Initial Chromosome of Genetic Algorithm for Scheduling Production-Distribution System

Rafiuddin Rody[1], Wayan Firdaus Mahmudy[2], Ishardita Pambudi Tama[3],
[1,2] Faculty of Computer Science, Computer Science, Brawijaya University, Malang, Indonesia
[3]Industrial Engineering, Faculty of Engineering, Brawijaya University, Malang, Indonesia
*corresponding author, [1]rafiuddinrody3rd@gmail.com, {[2]wayanfm, [3]kangdith}@ub.ac.id

Received 12 February 2019; accepted 06 May 2019

Abstract. Production and distribution system in a company should be managed carefully. Delay in product delivery not only results in a late penalty due to customer dissatisfaction or breach of contract, but also causes a supply chain failure. Of course, all these impacts will also reduce the reputation of a company. Scheduling integrated production-distribution is classified as NP-Hard problem. Genetic algorithm can be used to solve complex problem. In this paper, genetic algorithm is used for scheduling production-distribution in make to order system where each job has a different deadline and volume (size). This problem is represented on mixed integer programing model. We verify the genetic algorithm's performance by comparing the results with the total cost calculated by lower bounds of the problems. Experiments show that the traditional initial random cannot produce good result with more than 15 job size problem. We proposed guided initial chromosome to tackle this problem. From further experiments shows that the proposed method approach can increase the performances of genetic algorithm in more than 15 job size problem. In general, proposed genetic algorithm with guided initial chromosome shows better solution quality and better time efficiency compared to previous related research.
Keywords: Production and distribution; Genetic algorithm; make to order, combinatorial problem, guided initiate chromosome;

1. Introduction

More companies today are adopting a make-to-order business model where products are made by order and then sent to customers in a very short idle time [1]. Delivering product immediately after the completion of the production process is very important, especially for time sensitive products such as ready-mix concrete [2][3].

Many researchers have studied the problem of integrated production-distribution from various aspects. One aspect related to this problem is the scheduling problem where the order has a due date [4][5]. Delay in product delivery not only results in a late penalty due to customer dissatisfaction or breach of contract, but also causes a supply chain failure. Of course all these impacts will also reduce the reputation of a company. On the other hand, finished products sent to customers before the due date can result in additional storage costs, insurance, or even result in product damage. Therefore, integrated scheduling problems that involve consideration of due dates are very important for most businesses that exist today.

In this study, we focus on the problem proposed by reference [6]. The proposed problem is the problem of scheduling production-distribution in make to order business system. The problem of scheduling job orders with different volumes and different deadlines. Firstly, the orders are produced in batching machines and then sent to customers using a transportation that have specified capacity. Each order, that has a different due date and volume, must be sent to the customer before the due date (delay is not permitted). The time required to process orders in a batch is a fixed value and does not depend on the total number of orders (volume) in it. In the

production process, set-up times and set-up costs are needed before the batch is processed. Similarly, delivering the product requires the delivery time and delivery cost for each shipment (round trip) between the factory and the customer. In addition, it is assumed that orders that arrive at customers before the due date will result in earliess penalty. The aim is to find a combinational scheme where jobs are assigned to the batches so that the total cost is minimized while ensuring the quality of service to customers.

The method proposed by reference [6] was hybrid simulated annealing and genetic algorithm that performed quite good with average error (for all cases and scenarios) no more than 16%. In this paper, we proposed more simple method to tackle this problem and aim to produce better solution quality and better time efficiency.

2. Method

2.1 Problem Description

In previous related research [6], Wang and Lou formulated production-distribution system, based on make to order production system, in mixed integer programming model. The parameters and variables used in the model are described on table 1.

Table 1 Parameter and Variable of Make to Order Model

Symbol	Description	Symbol	Description
n	Total number of jobs	η_c	Delivery cost
i	Index number of jobs	η_t	Delivery time
j_i	Job i	β_i	Pinalty cost of job i
d_i	Deadline of i	u	Total number of batches
v_i	Volume of i	b	Index of *batch*es
c	Batch capacity	C_i	Completion time (arrival) of Job i
λ_c	Set-up cost	C_b^B	Completion time (arrival) of batch b
λ_t	Set-up Time	b_i	the number of batch to which the job i is assigned
p_t	Production time	Z	Total Cost

The formulated problem is descripted as follows:

$$min\, Z = (\lambda_c + \eta_c).u + \beta_i.\{\sum_{i=1}^{n}(d_i - C_i)\} \quad (1)$$

Subject to:

$$C_i \le d_i,\ \forall i \in [1,n], \quad (2)$$

$$\sum_{i=1}^{n} v_i \le c,\ \ b_i = b, b \in [1,u], \quad (3)$$

$$C_b^B = C_i, \forall i\ \in [1,n]/b_i = b, b \in [1,u], \quad (4)$$

$$C_b^B \le C_{b+1}^B - max\{\lambda_t, \eta_t\}, b\epsilon[1,u], \quad (5)$$

Constraint (2) is used to ensure that each job has to be delivered to customer before or on its deadline. Constraint (3) is used to ensure that total volume of jobs in one batch may not surpass the transporter's capacity. Constraint (4) shows that jobs that are assigned to the same batch should have the same completion time. Constraint (5) defines the rule of the completion time of two consecutive batches.

The problem now, derived from the MIP model, is to find a combinational scheme where jobs are assigned to the batches so that the total cost is minimized. To verify the performance of the proposed method, we compare the resulted solution with the lower bounds of the problems based on the lower bound (LB) proposed by c.

2.2 Genetic Algorithm

Genetic algorithm (GA) is a stochastic optimisation technique based on the evolutionary ideas of genetic and natural selection. GA is widely studied and applied in many fields in engineering worlds. This algorithm begins by encoding the possible solutions in the search space. every

possible solution in the population is called a chromosome [8]. The chromosomes are evolved through iterations (generations) using genetic operators (crossover, mutation and selection). In every generation, new chromosomes (offsprings) that are created from the current population (parents) will be produced [7].

Here is an example how genetic algorithm used in this research:

1. Create some initial chromosomes to represent the solution which length is the same as the job size. The size of initial chromosome is based on the population size used in the experiment.
2. To get new solution from the current solution, genetic operators (crossover and mutation) are used. In crossover operator, two parents are selected to randomly combine the gens of each parent to produce new individual. The mutation operator is used to randomly change the gens of a selected individual and become new individual. The size of individual produced using mutation and crossover operator depends on mutation rate and crossover rate set in the experiment.
3. Select the best individuals from both parents and offspring to be the next generation population.
4. The next generation population than become the current solution. The solutions produced by using genetic algorithm are improved through these iterations. The iterations will be stopped if a stopping condition is fulfilled.

2.3 Proposed Method 1: Guided Initial Chromosome (GIC)

Feasible solution (chromosome) for this problem is represented by a sequence of batch numbers in which the i-th entry number indicates the number of the batch to which the job i is assigned to. For example, assume that the representation of a feasible solution to a problem with 7 jobs, is [3, 2 4, 2, 1, 2, 1]. So, it shows that the 1st job is assign to batch 3, the 2nd job is assign to batch 2, the 3rd job is assign to batch 4 and so on.

There are two ways for generating the initial population: creating random chromosomes or creating better chromosomes that are located in feasible search space or even in the vicinity of the global optimum area [9]. In this research, we propose guided initial chromosome (GIC). The guided initial chromosome is to generate initial chromosome from the range [1, x]. x is the number of batch that is most likely the vicinity of the global optimum area that is the number of batch calculated using lower bound method proposed by reference [7].

It is worth to note that the proposed genetic algorithm has repair mechanism. In the initial generate population or a newly generated population, each chromosome has to satisfy the capacity constraint. If a chromosome breaks the capacity constraint, it would be corrected to be possible solution. This repair mechanism is aimed to transfer jobs from the over-capacity batches to other possible batches with sufficient capacity.

GIC utilizes the repair mechanism of the proposed genetic algorithm to exploit the solution in the vicinity of the global optimum area, which is the lower bound of the problem. Using this method, we expect better solution quality and computing time.

2.4 Proposed Method 2: Two-Types of Crossovers and Mutations Operation

The number of new chromosomes produced is determined by the crossover-rate (cr) and mutation-rate (mr) parameters [10]. The main purpose of reproduction operations is to recombine the features of two randomly selected parents or change the value of a chromosome to produce better offsprings. To solve this combinational problem, firstly we propose GIC for initial chromosome and so we need to find the way to improve the initial solution. Here, we propose two-type of crossovers and mutations operations. For the crossover operators, we use two-types of crossover with equal possibility. Crossover A: two-cut point crossover, swap a

part of solution between two points randomly selected from a parent with a part of solution randomly selected between two points from another parent. Crossover B: Select two batch numbers from two parents, and swap their number.

For the mutation operators, we use two-types of mutation with equal possibility. Mutation A: Select two gen of a parent and swap their position. Mutation B: Select a batch index and modify it randomly to a number in the range [1, *x*]. This method also utilizes the repair mechanism of the proposed genetic algorithm to exploit the solution in the vicinity of the global optimum. The two-types of crossover and mutation are aimed to speed up the exploitation of solution by giving more access to change the value of chromosome (changing the gen location or changing the batch number). This method is proposed to utilize the advantage given by GIC which gives initial chromosomes better random solution.

2.5 Proposed Genetic Algorithm and Random Instance Guide

By implementing two proposed methods to the genetic algorithm, now we obtain the following genetic algorithm.

Step 1: Initialize parameters: PopSize, generation, crossover rate (cr), Mutation rate (mr).

Step 2: Find the lower bound of the instance problem.

Step 3: Generate a number of guided initial random chromosome as an initial population according to PopSize (generation n=0).

Step 4: Repair the chromosomes.

Step 5: Produce new solutions according to cr and mr with reproduction operations.

Step 6: Repair the chromosomes.

Step 7: Calculate the fitness value in the population.

Step 8: Select a number of the chromosomes that have the best fitness value to next generation (generation n+1).

Step 9: Repeat Step 5 to Step 8 until the last generation.

We used random instance guides proposed by reference [6] for this experiment. We conducted an experiment where c is generate randomly from range [150, 200]. For job volume, it is used three cases with small, middle and large job volumes, that were randomly generated from the range [30, 50], [30, 100] and [30, 150]. In every case, the number of jobs are set as 30, 50, 70, 100 and 150. Every parameter were generated randomly as follow: The $\boldsymbol{p_t}$ generated from the range [1,5], $\boldsymbol{\lambda_t}$ from [10, 40] and $\boldsymbol{\lambda_c}$ from [200, 400], $\boldsymbol{\eta_t}$ from [50, 150] and $\boldsymbol{\eta_c}$ from [300, 600], $\boldsymbol{\beta_i}$ from the range [0.001, 0.005]. The deadline $\boldsymbol{d_i}$ was generated from the range [0,2000]. For every case, we randomly generated 50 problem instances. The error ratio is defined as follows:

$$Err = (GA - LB)/LB \quad (6)$$

In equation 6, GA represents cost resulted from GA solution and LB is cost calculated by lower bound metode proposed by reference [7]. Average error ratio ($A.Err$) and average computing time ($A.CpuT$) are calculated using these equations.

$$A.Err = (\sum Err)/50. \quad (7)$$

$$A.CpuT = \sum(CpuT\ of\ each\ instance)/50. \quad (8)$$

3. Result and Discussion

In this section, we explained three testing results from our experiments. Firstly, we tested the proposed genetic algorithm using GIC compared to genetic algorithm without using GIC. Secondly, we tested the proposed method using GIC and two-type operators compared to genetic algorithm using GIC and one-type operators. Lastly, we compared the proposed genetic algorithm with the result from previous research conducted by reference [7].

3.1 Testing of Guided Initial Chromosome vs Non Guided Initial Chromosome

In this test, we used parameters as follow: the number of jobs were set as 10, 15, 20, 25, 30; capacity c was generate randomly from range [150,200]; job volumes from [30,50]; $\boldsymbol{\lambda_c}$[200, 400] and $\boldsymbol{\lambda_t}$ from [10, 40]; $\boldsymbol{\eta_t}$ from [50, 150] and $\boldsymbol{\eta_c}$ from [300, 600]; $\boldsymbol{d_i}$ was generated from the range [0,2000]; $\boldsymbol{\beta_i}$ from the range [0.001, 0.005]. For each method, we randomly generated 20 problem instances. Non GIC is a traditional initial random chromosome that generates initial batch numbers from all possible solution. For *n* job-size problem, the chromosome is randomly generated from range [1,*n*]

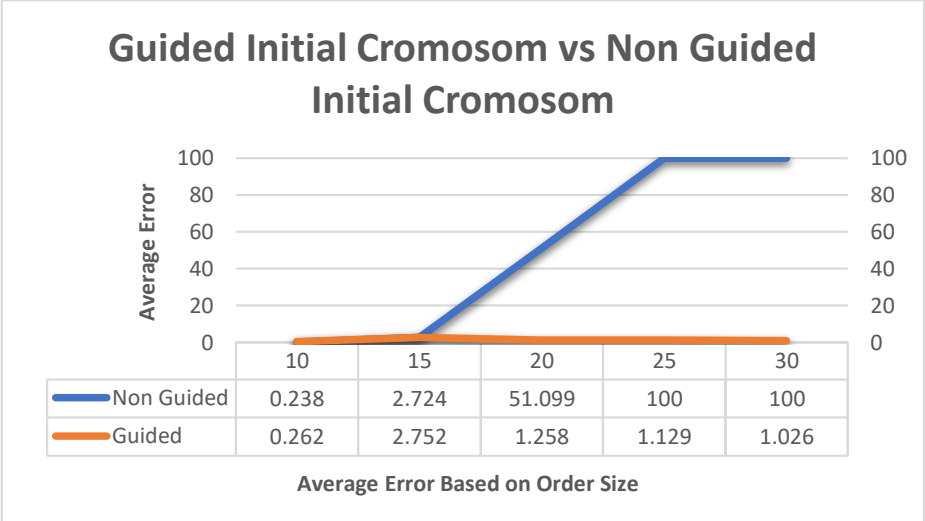

Fig. 1 Testing of Guided Initial Chromosome VS Non Guided

Based on Fig. 1, non GIC cannot produce good result when the problem has more than 15 job size. The increasing number of sizes causes an increase in searching area for solutions. By using GIC, the searching area for solution is narrowed down and because of that the performance of the genetic algorithm improves.

3.2 Testing of A. Error Using One-Type Operators VS Two-Type Operator

In this test, we used random instance guide from section 2.5. For every scenario (one-type or two-type operator), we randomly generated 50 problem instances and compare the results.

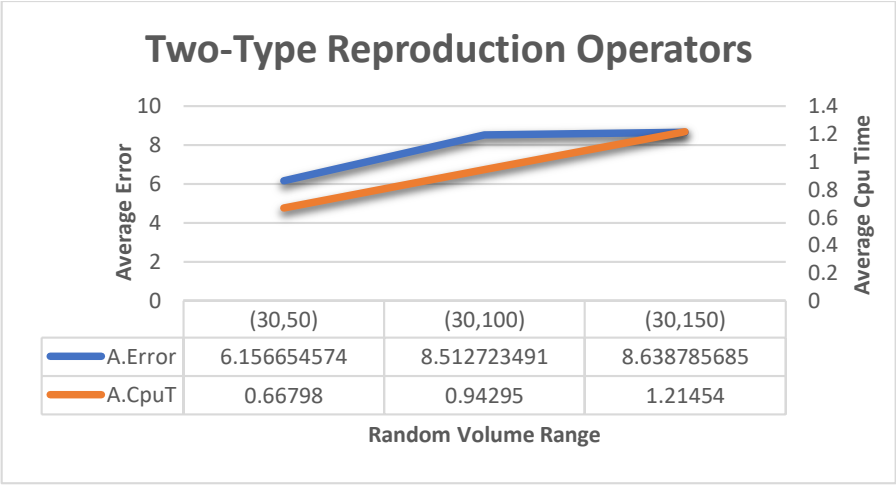

Fig. 2 Testing of Average Error using Two-Type Operators

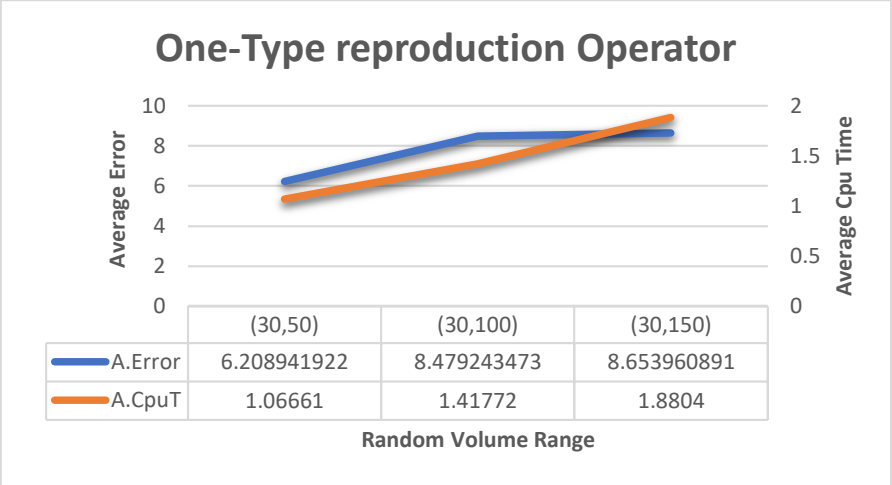

Fig. 3 Testing of Average Error One-Type Operator

Fig. 2 and Fig. 3 show the average results for all cases from generated random instance based on rules in section 2.5. From these results, we know that using two-type of operators reduces more than 30% computing time compared to one-type operator and slightly improves the solution quality.

3.3 Comparing proposed genetic algorithm with previous related research

Fig. 4 is average error and computing time summary from previous research done by reference [6].

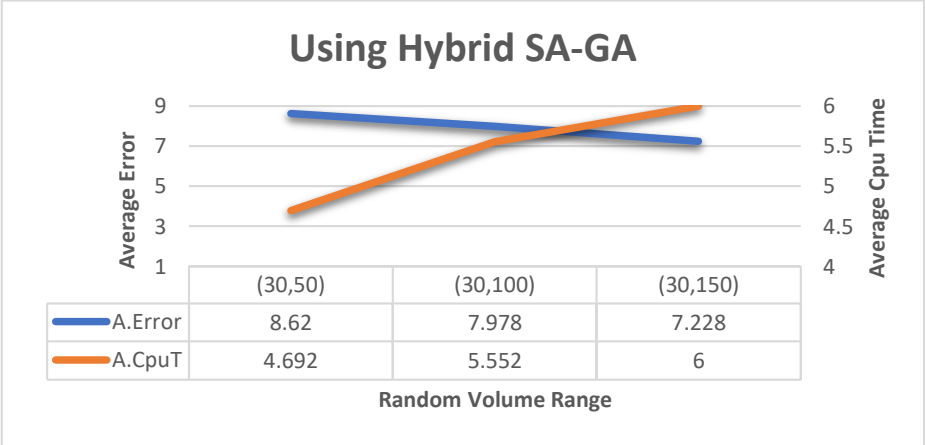

Fig. 4 Average Error from previous related research

Fig. 2 and Fig. 4 are the Average Error and Average Computing time from random instance which used the same random instance rules in section 2.5 and similar computer specification. From these two figures, we know that our proposed method performs better in term of average solution quality (A.Error = 7.77%) and in term of average computing time (A.CpuT = 0,948).

4. Conclusion

Guided initial chromosome has shown that this method can solve the problem for scheduling production-distribution system in more the 15 job-size problem. Similarly, two-type of crossover and mutation operator has proven its efficiency to reduce computing time by giving more opportunity for the chromosome to get better solution quickly. Both GIC and two-type of operator are proposed because the genetic algorithm has the repair mechanism. Without the repair mechanism, the solution produced most likely would not be good solution and the computing time would be too long.

References

1. Z.-L. Chen, "Integrated Production and Outbound Distribution Scheduling: Review and Extensions," *Oper. Res.*, vol. 58, no. 1, pp. 130–148, Feb. 2010.
2. J. M. Garcia and S. Lozano, "Production and vehicle scheduling for ready-mix operations," in *Computers and Industrial Engineering*, 2004, vol. 46, no. 4 SPEC. ISS., pp. 803–816.
3. J. M. Garcia and S. Lozano, "Production and delivery scheduling problem with time windows," *Comput. Ind. Eng.*, vol. 48, no. 4, pp. 733–742, Jun. 2005.
4. D. Wang, H. Guo, and K. Zhu, "Lot sizing and scheduling problem for production-delivery system with job volume and due date considerations to minimise the total cost while guaranteeing a certain customer service level," *Int. J. Manuf. Res.*, vol. 9, no. 3, p. 294, 2014.
5. . Suginouchi, T. Kaihara, D. Kokuryo, and S. Kuik, "A Research on Optimization Method for Integrating Component Selection and Production Scheduling under Mass Customization," *Procedia CIRP*, vol. 57, pp. 527–532, 2016.
6. D. Wang and H. Luo, "Simultaneous Lot-Sizing and Scheduling for Single-Stage Multi-product Production-Distribution System with Due Date Considerations to Minimize Total Logistics Cost," in *2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, 2016, vol. 1, pp. 200–203.
7. D. Wang, O. Grunder, and A. EL Moudni, "Using genetic algorithm for lot sizing and scheduling problem with arbitrary job volumes and distinct job due date considerations," *Int. J. Syst. Sci.*, vol. 45, no. 8, pp. 1694–1707, Aug. 2014.
8. M. L. Seisarrina, I. Cholissodin, and H. Nurwarsito, "Invigilator Examination Scheduling using Partial Random Injection and Adaptive Time Variant Genetic Algorithm," *J. Inf. Technol. Comput. Sci.*, vol. 3, no. 2, pp. 113–119, 2018.
9. [W. F. Mahmudy, "Optimisation Of Integrated Multi-Period Production Planning and Scheduling Problem In Flexible Manufacturing System (FMSs) Using Hybrid Genetic Algorithms," University of South Australia, 2014.
10. W. F. Mahmudy, R. M. Marian, and L. H. S. Luong, "Optimization of part type selection and loading problem with alternative production plans in flexible manufacturing system using hybrid genetic algorithms - part 1: Modelling and representation," in *2013 5th International Conference on Knowledge and Smart Technology (KST)*, 2013, pp. 75–80.