Received: September 28, 2020. Revised: December 21, 2020.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Optimization of Multi-Stage Distribution Process Using Improved Genetic Algorithm

Wayan Firdaus Mahmudy[1]* **Mohammad Zoqi Sarwani[2]** **Asyrofa Rahmi[1]**
Agus Wahyu Widodo[1]

[1]*Universitas Brawijaya, Indonesia*
[2]*Universitas Merdeka Pasuruan, Indonesia*
* Corresponding author's Email: wayanfm@ub.ac.id

Abstract: A supply chain process in manufacturing industries involves a distribution process that ensures finished goods distribute to their customers properly. The process may involve several parties such as manufacturing plans, distributions centres, and retailers. The companies should optimize the process by minimizing related costs to achieve the maximum profit. The multi-stage distribution problem contains several constrains so obtaining optimum solution using exact methods may require excessive processing time. Genetic algorithm is proposed the solve the complex problem. The genetic algorithm is improved by modifying chromosome representation and related genetic reproduction operators. The new chromosome is designed to address different number of distribution stages in manufacturing industries and could be applied for various distribution problems. Several genetic operators are tested to obtain the most suitable reproduction operator for the multi-stage distribution process. The numerical experiments prove that the improved genetic algorithm is suitable for the optimization of the multi-stage distribution process and produce better result compared to those achieved by original genetic algorithm, simulated annealing, and random search as baseline method. The improved genetic algorithm produces solution with cost of Rp 20,167.8, lower than cost of Rp 21,860.4, Rp 23,354.30 and Rp 34,328.0 obtained by original genetic algorithm, simulated annealing, and random search, respectively. For a bigger size problem, the improved genetic algorithm also produces solution with lower cost. The result proves the robustness of the improved GA for solving big size problems. The improved GA also produces solutions that has the lowest value of coefficient of variation. It means that the improved GA is more stable and produces less variation if it is run several times.

Keywords: Genetic algorithm, Supply chain, Manufacturing, Cost optimization, Distribution process.

1. Introduction

Distribution is common activity in manufacturing companies to deliver finished products to their customers [1]. The product may be delivered through several stages such as manufacturing plants, distribution centres, and retailers before it gets to the customer [2, 3]. Thus, the process can be defined as multi-stage distribution process. Distribution cost together with production and promotion cost have high percentage to the total cost of product [4]. Thus, minimizing the distribution cost is an important task in manufacturing industries.

A higher number of stages is used to cover the wide area of customers' locations. Thus, a longer distribution network and also a higher distribution cost are required. The cost could be reduced by choosing the right distribution routes and appropriate product quantity for each route. This effort will help the company to stay competitive with other companies [2]. Since minimizing cost is also resulting in minimizing distribution time, a good distribution process may also increase company's service level [3].

The decision related to the distribution process should be frequently made as there is an increase of fluctuations in consumer demands. The demands should be fulfilled on time while considering

DOI: 10.22266/ijies2021.0430.19

manufacturer's inventory level [5]. Therefore, an efficient approach is required to produce good solutions in reasonable amount of time [6].

In term number of stages, several types of the distribution network have been addressed in the previous studies. The related researches includes solving the problem by using two-stage distribution [7, 8]. In the studies, there was central distributor as intermediary. In this scenario, the products were sent to the central distributor in the first stage. In the second stage, the central distributor distributed the products into customers. A higher level of distribution network was solved in [9] that addressed a three stages distribution who involved four entities namely producers, distributors, wholesalers and retailers.

In this study, rather than using a fixed number of distribution stages, we develop a flexible solution representation so it could be used to solve a problem that involves any level of distribution stages in manufacturing industries. The solution representation is then implemented as chromosome representation used by genetic algorithm to produce feasible solutions. The proposed approach in this study is developed based on our previous research [6]. The GA is chosen as it has been successfully applied for various complex combinatorial problems [10]. Several genetic operators are modified and tested to obtain the most suitable reproduction operator for the multi-stage distribution process. The numerical experiments are carried out to assess the effectiveness of the proposes approach. An original genetic algorithm, simulated annealing, and random search are used as baseline methods.

This paper consists of five sections. An introduction that contains the background problem and the need for a robust algorithm and flexible solutions are discussed Section 1. Section 2, literature review, explains related previous research. Section 3, methodology, explains the development of mathematical formulation and operators of the genetic algorithm. Presentation of computational experiment result is discussed in Section 4. Conclusions are presented at the last section.

2. Literature review

Some approaches to solve the distribution problems have been proposed. The approaches included mathematical modelling and fuzzy logic-based methods. To get good solutions in reasonable amount of time, several meta-heuristic algorithms such as simulated annealing, particle swarm optimization, and genetic algorithm were also implemented. Hybrid algorithms were also applied in several studies to address the complexity of the distribution problems.

Mathematical modelling-based methods were implemented in several studies. For example, min–max dynamic programming formulation was used to solve a two-echelon supply chain problem [11]. While the min–max dynamic programming was used for small size problem, they implemented a min–max matheuristic for larger problems. Similar approaches were applied in [12, 13] for optimization of inventory levels and transportation cost. The research [12] used non-integer linear optimization problems in the formulation to minimize inventory and transportation costs. To solve the formulated problem, Lagrangian approach was applied. Results of the study was the achievement of cost savings and profit improvement. Distribution issues raised in the study encouraged the non-linearity of inventory cost. The study [13] used max-min penalty approach to reducing the total cost of transportation.

Fuzzy logic based methods were also applied by researchers to solve the problems of fuzzy transportation and distribution with the function of ranking [14]. Trapezium fuzzy value represented distribution costs resulting in a ranking function. Once it is converted into crisp, which was solved easily using standard transport algorithm. The approach was still too classic to be easily understood and applied in the real world. The benefit was reduced computational complexity. However, the optimal solution produced had the possibility of the same value with standard transport algorithms without fuzzification. This shows that there was no change in the optimal solution of methods offered. While the study [15] used type-2 fuzzy parameters on the two problems fixed costs of transportation. Unit transportation cost and fixed cost in the first problem and the second problem was the unit transportation cost, fixed cost, supplies, and demands. As with previous studies [14], the study addressed the distribution problems only take two stages from multiple sources to multiple destinations (customer) without going through the distribution centre or retailer.

Artificial intelligence based methods were also implemented like in [16] which resulted in a three-stage distribution model so that it can generate a good simulation for distribution management. Meta-heuristic algorithms that belong to artificial intelligence-based methods were also implemented. For example, simulated annealing was used to solve the distribution problem in [17]. The transfer of the product to the customer through a cross-docking centre that has a short storage time. The problem was modelled using mix-integer programming. Even though the study reported promising results by minimizing costs related to the cross-docking network, they addressed only fixed level multi-stage distribution problems.

Population based meta-heuristic algorithms that have capability in exploring a wide search space were also implemented in several studies. For example, particle swarm optimization (PSO) was used to solve the distribution problem that exists in real-world [18]. The study had an objective determining the best reorder point of warehouses in supply chains.

A genetic algorithm (GA) is well known as a robust optimization tool has been successfully used for complex combinatorial problems [10]. Thus, the algorithm has been used for solving various cases related to the transportation problems. For example, the genetic algorithm in [7] was used to solve the distribution problems that involves several parameters such as transportation cost per unit of product, a fixed cost for certain distribution route, and capacity of each distributor centre (DC). As other studies, this study addressed only fixed level multi-stage flexible distribution problems.

The mentioned previous studies mostly used a fixed number of distribution stages and fixed number of vehicles used for transportation. Optimization of multi-stage distribution using genetic algorithm was also presented in [6]. The results of GA were compared with iterative local search. This research study conducted a flexible multi-stage distribution problems and flexible number of vehicles. The GA with different operators is developed to obtain better results.

3. Methodology

This study addresses a multi-stage distribution problem and use one type of product. A data set is generated according to the statistical distribution of the real data from surveys. The data have several parameters such as products stock in the plant, storage capacity of distributors in each stage, number and capacity of vehicles in each stage and other related costs [6].

Genetic algorithm (GA) is chosen to address the distribution problem. The GA has several steps including determining chromosome representation to encode solution of the problem, reproduction to generate alternative solutions, and selection to choose which solutions should be passed into the next iteration. The process is adopted from biological natural selection [19]. In each iteration, the chromosomes are evaluated its fitness value to determine the goodness of the solutions for the considered problem [20].

3.1 Problem formulation

All objectives and constraints of the problem are determined in the problem formulation.

3.1.1. Variables of the problem

We address a distribution problem that have I stages. In stage i there are J_i manufacturing plants/distributors/retailers that are involved in the distribution process. Next, we will consider the plants/distributors/retailers as an entity.

Each entity in j have V_{ij} vehicles that may have different capacity (VC_{ijv}). A unit cost C_{ijk} is required if there is a product delivery from entity j in stage i into entity k in the lower level (i+1). A number of products R_{ij} are requested by entity j in stage i from entities in higher level.

3.1.2. Objective function

Based on all identified variables in the distribution process, we determine the problem as total cost minimization in Eq. (1) according to [21] as follow:

$$cost = \sum_{i=1}^{I-1} \sum_{j=1}^{J_i} \sum_{k=1}^{J_{i+1}} C_{ijk} Q_{ijk} \tag{1}$$

Where Q_{ijk} represents the total quantity of items delivered from entity j in stage i into entity k in the lower level (i+1). Q_{ijk} is decision variable that its value will be searched during optimization process.

3.1.3. Constraints to fulfill all request

Request R_{ij} by entity j in stage i must be fulfilled from entities in higher level as shown in Eq. (2).

$$\sum_{k=1}^{J_{i-1}} Q_{i-1,k,j} = R_{ij} \tag{2}$$

This constraint also ensures that total request/available product stock in higher level must be equal or greater than total request in lower level.

3.1.4. Constraints on the vehicle capacity

Total capacity of all vehicles in entity j in stage i is enough to deliver all product into all entities in the lower level (i+1) as shown in Eq. (3).

$$\sum_{k=1}^{J_{i+1}} Q_{ijk} = \sum_{v=1}^{V_{ij}} VC_{ijv} \tag{3}$$

Table 1. Chromosome representation in 1 segment

Entity in Level i												
Distributor Sender j						…	Distributor Sender J_i					
Vehicle v			Vehicle V_{ij}			…	Vehicle v			Vehicle V_{Ji}		
k	..	J_{i+1}	k	..	J_{i+1}	…	k	..	J_{i+1}	k	..	J_{i+1}
30	..	40	27	…	25	…	42	…	38	54	…	31

3.2 Chromosome representation

Determination of chromosome representation is a crucial step in GA implementation. Each chromosome represents a possible solution [22] and a good chromosome representation will enable the GA to efficiently explore vast search space. A real-coded chromosome that contains several segments is used this research. Number of segments is determined according to number of stages and each segment represents a stage or plants/distributors/retailers that are involved in the distribution process. For example, this first segment encodes the plants, its vehicles, and quantity of products delivered by the vehicles. The next segment represents the distributors or retailers in the lower levels. An example of one segment of chromosome is provided in Table 1.

As shown in Table 1, in level i there is J_i entities (plants/distributors/retailers) that deliver final product to J_{i+1} entities in level i+1. Numbers in the last row represent number of products that are delivered by each vehicle. The numbers are treated as real (float) number when are processed in the cycle of the GA. The number will be converted into round number when used for calculation of problem's objective.

3.3 Fitness function

Fitness function is used to measure the quality of solution decoded from a chromosome. Determining the function is an important key in GA process [23]. A chromosome that has higher fitness will have higher probability to be selected for the next generation. As the objective of this optimization problem is minimizing the total distribution cost as stated in Eq. (1), the fitness function is defined as Eq. (5).

$$Fitness = \frac{1}{cost} \tag{5}$$

3.4 Reproduction

The reproduction process is carried out after the initialization population that randomly produce a number (population size or *pop_size*) of initial chromosomes. The reproduction process will produce new chromosomes as new possible solutions. The reproduction process is executed using two genetic operators, namely crossover and mutation. The crossover operator produces a child (offspring) by using two chromosomes as parents. The mutation operator uses only one chromosome as a parent. The number of children produced using the crossover operator is determined by crossover rate (*cr*) while the number of children producing using the crossover operator is determined by mutation rate (*mr*) [24].

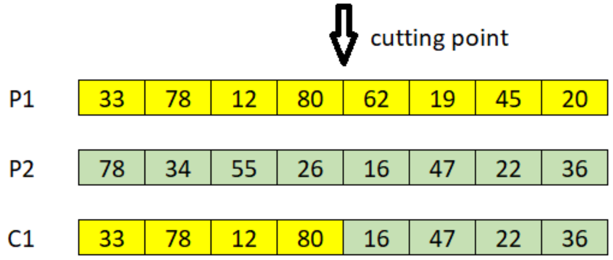
3.4.1. Crossover

Crossover operator is the core and unique operator in GA. This crossover method is a method of reproduction that involves two parents from population pool hat selected randomly. This method is used to get the child with chromosome more varied because it is derived from two parents. In this stage, the operator will produce *cr* x *population size* of children. There are several kinds of methods of crossover in genetic algorithm, but in this study only uses one cut point crossover (OCP) that is modified to fit the multi-stage distribution problem. The mechanism does a crossover on each segment or at the level of each chromosome. So, if there are more than one stage of distribution problem, the crossover process on each chromosome will do as much as n stage. This mechanism will keep part of good solutions for the distribution problem that may be lost if other method such as intermediate extended crossover [25] is used. In Fig. 1 is shown the crossover process using one cut point in a segment.

Fig. 1 shows that *P1* and *P2* are chromosomes that are selected as parents. A cutting point is randomly chosen. A child (*C1*) is performed by copying the left part from *P1* dan the right part from *P2*.

3.4.2. Mutation

The mutation operator uses only one chromosome as a parent to produce a new child. . In this stage, the operator will produce *mr* x *population size* of children. This research will use an exchange mutation and a random mutation. The methods will be randomly selected to generate more diverse population.



P1	33	78	12	80	62	19	45	20
P2	78	34	55	26	16	47	22	36
C1	33	78	12	80	16	47	22	36

Figure. 1 Crossover process using one cut point

The exchange mutation works by exchange two genes in each stage of the distribution problem. The random mutation uses a random number r as a trigger the amount of change in genes value. Random value r has a range 0 to 1 assuming that the exploitation of r did not exceed the limit range of genes when generated. The limit range of gene intended is the difference between the maximum and minimum limits so that the value generated when r multiplied by the difference between the limit range is a variation of the portion. The results obtained are summed with the value of the parent gene so that the changes are not too far away. It is based on the principle of a possible child has a gene that is not too far from its parent. Eq. (6) is a function of the random process of mutation for each gene exchange.

$$g'_a = g_a + r(max - min) \tag{6}$$

Where g_a is an original value of the a-th gene before mutated and g'_a is the resulting gene after mutated. Same as the crossover process, mutation process is also done at each stage as a segment in a chromosome. So, if there is I stage on each chromosome, the mutation process is done as many as I stage.

3.5 Selection

Children (offspring) produced by using the crossover and mutation operators are put the offspring pool. As there are *pop_size* parent chromosomes in the population pool, we need a mechanism to select which chromosomes should be passed into the next generation. In this study, we employ elitism selection by putting together parents and offspring into a temporary pool and sorting it according to its fitness value. The best *pop_size* from the temporary pool will be selected into the next generation. This mechanism guarantees that the best chromosome always be selected.

The cycle of GA is repeated by doing reproduction process using the new chromosomes until a predetermined number of iterations/generations [24].

4. Experimental results

A case of distribution problem used in this study is taken from [6]. The problem has 3 stages. In the first stage there are 3 distribution centres and at the second stage there are 7 distribution agents. In the last stage there are 9 retailers.

4.1 Preliminary experiments

The genetic algorithm will produce satisfactory result if its parameters are properly determined. For example, setting a higher *pop_size* may produce a higher fitness value as more possible solutions are generated to explore the wide are of the search space. However, a higher computational time is also required. Thus, conducting preliminary computational experiments are required to obtain the most suitable of parameter values of the GA.

4.1.1. Testing of population size

The size of the population contained in the genetic algorithm is one of the parameters which have an influence on a given solution. The larger the size of the population, the more varied solutions that can be given to the genetic algorithm so the impact on the possibility to get a better solution. In testing the size of the population, we use the population size in the range between 10 to 100 by 10 times of execution for each population size. We repeat the experiment 10 times for each population size as the stochastic nature of GA that produces different results for each run. Table 2 shows the test result of population size.

The test results to determine the appropriate size of the population are shown in Table 2. From the results obtained the population is able to provide the best solutions are the size of 100. It is possible that the GA may produce better results if using more than 100 of population size, however an excessive computational time is required and the time may be better if used for reproduction process [26].

4.1.2. Testing number of generations

As testing the population size, testing number of generations is required to determine its suitable value that enable the GA to get the best possible solutions in reasonable amount of time. As shown in Table 3, we test the value from 100 to 1000. After 300

Table 2. Testing result of population size

No	Population size	Average cost	Average fitness
1	10	6270	0.0001684984
2	20	5080	0.0001995103
3	30	4775	0.0002126277
4	40	4955	0.0002048416
5	50	4375	0.0002346217
6	60	4700	0.0002188333
7	70	4285	0.0002389802
8	80	4190	0.0002386630
9	90	4290	0.0002331000
10	100	3910	0.0002559861

Table 3. Testing result of the number of generations

No	Generation Number	Average cost	Average fitness
1	100	3945	0.0002579337
2	200	4200	0.0002434614
3	300	3685	0.0002761948
4	400	4125	0.0002488171
5	500	4440	0.0002330662
6	600	4105	0.0002478166
7	700	4090	0.0002472525
8	800	4245	0.0002424727
9	900	4190	0.0002484967
10	1000	3910	0.0002559861

Table 4. Result testing of *crossover rate* and *mutation rate* combination

Crossover rate (*cr*)	Mutation rate (*mr*)	Average cost	Average fitness
0.1	0.9	4550	0.0002398221
0.2	0.8	4285	0.0002398221
0.3	0.7	3945	0.0002593855
0.4	0.6	3955	0.0002586965
0.5	0.5	3910	0.0002639413
0.6	0.4	3925	0.0002627761
0.7	0.3	4070	0.000250042
0.8	0.2	4145	0.0002478912
0.9	0.1	4430	0.0002336377

Table 5. The best GA parameters from preliminary experiments

Parameter of GA	Value
Population size	100
Number of generations	300
Crossover rate	0.5
Mutation rate	0.5

generations, the GA has low probability to get better solutions. Thus, we determine that the best number of generations is 300.

4.1.3. Testing combination of crossover rate (cr) and mutation rate (mr)

A proper value of crossover rate (*cr*) and mutation rate (*mr*) will enable the GA to balance its exploration and exploitation power in huge and complex search space. To get fair results, the value of *cr* and *mr* is set so that *cr*+*mr*=1 in each run. Using this scenario, the GA will produce a same number of offspring (possible solutions) in each iteration. In this test, we use population size 0f 100 and number of generations of 300 as obtained in the previous experiments. The testing result is presented in Table 4. Here, we get that the best combination is *cr*=0.5 and *mr*=0.5.

4.2 Result analysis

From the preliminary experiments, the best parameter values of the GA are obtained as shown in Table 5. By using the values, the GA will efficiently explore and exploit the search space of the multi-stage distribution problem.

Using the same distribution problem data, solutions using the proposed method (improved genetic algorithm, IGA) is compared to previous version of genetic algorithm (GA) [6] and a random search (RS) algorithm as a baseline method. A simulated annealing that has been known as popular optimization method is also used as a comparison method.

We focus on comparing the improved GA with the previous version of GA as the GA has been proved as a superior method for complex problems such as the product distribution problem. For example, the GA produced better results than simulated annealing for network design problem that is considered as large-scale optimization problem [27]. A similar finding was also proved in the chemical composition optimization that is considered as a complex problem [28].

The random search (RS) algorithm has two main processes to enable exploiting and exploring the search space of the distribution problem. The first process works by randomly exchange number of items unit transported by any vehicle within a distribution stage. The second process exchange the number of items between two different distribution stages.

The simulated annealing (SA) has three main processes to solve the distribution problem as follows:

- Generating a random initial solution.
- Producing an alternative solution by using two main processes as applied in the RS.
- Applying a function to reduce the probability of accepting a new worse solution. This mechanism enables the SA to escape local optimum areas.

Computational experiments of SA, RS, GA and IGA are performed 10 times and the results are shown in Fig. 2.

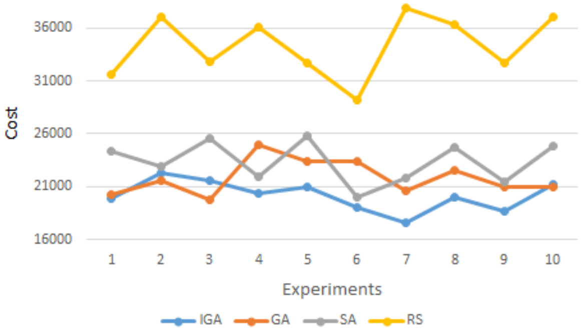


Figure. 2 Comparison cost of all methods

Table 6. Comparison of Average Cost and Fitness

Methods	Average Fitness	Average Cost (Rp)
Improved GA (IGA)	0.000050	20,167.80
Genetic Algorithm (GA)	0.000046	21,860.40
Simulated Annealing (SA)	0.000043	23,354.30
Random Search (RS)	0.000029	34,328.00

Table 7. Comparison of Coefficient of Variation

Methods	Coefficient of Variation
Improved GA (IGA)	6.719
Genetic Algorithm (GA)	7.151
Simulated Annealing (SA)	8.041
Random Search (RS)	8.016

Table 8. Comparison of Average Cost and Fitness for Big Size Problem

Methods	Average Cost (Rp)
Improved GA (IGA)	5,937,148.20
Genetic Algorithm (GA)	6,016,943.30
Simulated Annealing (SA)	6,219,221.10
Random Search (RS)	6,837,240.30

Based on the line chart in Fig. 2 it, can be seen that IGA provides superior results (the lower cost) comparable to those achieved by other methods. The random search (RS) as a baseline method gives the worst results (the higher cost). The simulated annealing (SA) has better results than RS but its performance is still bellow than the previous version of GA.

Experiments for GA, SA and RS are done 10 times considering the methods are stochastic algorithms. A stochastic algorithm will produce various results for each run and its average is used to measure the robustness of the algorithm [24]. The average fitness and distribution costs of all methods are shown in Table 6.

According to the Table 6, the improved genetic algorithm produces better results in the average cost and fitness in comparable to those achieved by the previous version of the genetic algorithm. These results prove that the improvisation of the genetic operators of GA is effective to provide a lower cost of distribution process.

We also calculate coefficient of variation (CV) of cost that are resulted by all methods in 10 runs. The CV is used to measure the volatility of each method. As shown in Table 7, the improved genetic algorithm has the lowest value of CV. It means that the improved genetic algorithm produces more stable results and less variation if it is run several times.

To prove the effectiveness of the proposed algorithm in solving bigger size problems, we generate a bigger case distribution problem that has 5 stages. The problem involves 1, 2, 6, 18, 54 entities in each stage from the first stage to the fifth stage, respectively. Eighty vehicles are used to deliver the products. The average distribution costs of all methods for 10 runs are shown in Table 8. The result proves the robustness of the improved GA for solving big size problems.

5. Conclusion

This study proves that the genetic algorithm with the proposed operator models can be employed to solve multi-stage distribution problems. To enable the genetic algorithm obtaining good results in reasonable amount of time, some parameters' values of the genetic algorithm are determined in the preliminary computational. The computational experiments found that the best population size is 100 and the best number of generations is 300. The experiment also found that the best crossover rate (*cr*) and mutation rate (*mr*) is 0.5.

Using the best parameters values, the improved GA produces the best results comparable to those achieved by all comparison methods when viewed from the average cost and fitness. The improved genetic algorithm produces solution with the average cost of Rp 20,167.8, lower than cost of Rp 21,860.4, Rp 23,354.3 and Rp 34,328.0 obtained by original genetic algorithm, simulated annealing, and random search, respectively. The improved GA also produces solutions that has the lowest value of coefficient of variation. It means that the improved GA is more stable and produces less variation if it is run several times.

A bigger size problem is generated to prove the effectiveness of the proposed algorithm. The improved genetic algorithm produces solution with the average cost of Rp 5,937,148.20, lower than cost of Rp 6,016,943.30, Rp 6,219,221.10 and Rp 6,837,240.30 obtained by original genetic algorithm, simulated annealing, and random search, respectively. The result proves the robustness of the improved GA for solving big size problems.

For stochastic algorithm such as GA, the average fitness becomes a reference to determine the solution. This describes that the multi-stage distribution problems highly recommended solved using the improved genetic algorithm in combination with the modified one cut point crossover, exchange and random mutation, and elitism selection.

Conflicts of Interest

The authors declare no conflict of interest

Author Contributions

Conceptualization, Wayan Firdaus Mahmudy; methodology, Wayan Firdaus Mahmudy and Mohammad Zoqi Sarwani; software, Mohammad Zoqi Sarwani and Asyrofa Rahmi; validation, Wayan Firdaus Mahmudy and Agus Wahyu Widodo; investigation, Wayan Firdaus Mahmudy and Mohammad Zoqi Sarwani; writing—original draft preparation, Mohammad Zoqi Sarwani and Asyrofa Rahmi; writing—review and editing, Wayan Firdaus Mahmudy and Mohammad Zoqi Sarwani.

Acknowledgments

This work was supported by the Faculty of Computer Science, Brawijaya University.

References

[1] E. Kusrini, E. Sugito, Z. M. Rahman, T. N. Setiawan, and R. P. Hasibuan, “Risk mitigation on product distribution and delay delivery : A case study in an Indonesian manufacturing company”, *IOP Conference Series: Materials Science and Engineering*, Vol. 722, No. 1, pp. 12015–12020, 2020.

[2] F. Ghassemi Tari and Z. Hashemi, “A priority based genetic algorithm for nonlinear transportation costs problems”, *Computers & Industrial Engineering*, Vol. 96, No. 1, pp. 86–95, 2016.

[3] R. Nourifar, I. Mahdavi, N. Mahdavi-Amiri, and M. M. Paydar, “Optimizing decentralized production–distribution planning problem in a multi-period supply chain network under uncertainty”, *Journal of Industrial Engineering International*, Vol. 14, No. 2, pp. 367–382, 2018.

[4] W. Hapsari, U. Sumarwan, and H. Wijayanto, “analysis of the effect of production, promotion and distribution costs towards company’s sales of the fast moving consumer goods sector”, *Russian Journal of Agricultural and Socio-Economic Sciences*, Vol. 87, No. 1, pp. 175–182, 2019.

[5] S.-Z. Lou and X.-C. Tian, “Inventory control in a fluctuating demand and return environment”, *Control Theory and Applications*, Vol. 32, No. 1, pp. 934–941, 2015.

[6] A. Rahmi, W. F. Mahmudy, and M. Z. Sarwani, “Genetic Algorithms for Optimization of Multi-Level Product Distribution”, *International Journal of Artificial Intelligence*, Vol. 18, No. 1, pp. 135–147, 2020.

[7] K. Antony Arokia Durai Raj and C. Rajendran, “A genetic algorithm for solving the fixed-charge transportation model: Two-stage problem”, *Computers and Operations Research*, Vol. 39, No. 9, pp. 2016–2032, 2012.

[8] R. Masson, N. Lahrichi, and L. M. Rousseau, “A two-stage solution method for the annual dairy transportation problem”, *European Journal of Operational Research*, Vol. 251, No. 1, pp. 36–43, 2016.

[9] Z. Indra and Subanar, “Optimization of Supply Chain Distribution Costs Three Levels Using Adaptive Genetic Algorithm and Distributed”, *Indonesian Journal of Computing and Cybernetics Systems*, Vol. 8, No. 2, pp. 189–200, 2014.

[10] A. Syarif, D. Anggraini, K. Muludi, W. Wamiliana, and M. Gen, “Comparing Various Genetic Algorithm Approaches for Multiple-choice Multi-dimensional Knapsack Problem (mm-KP)”, *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 5, pp. 455–462, 2020.

[11] L. Bertazzi, A. Bosco, and D. Laganà, “Min–Max exact and heuristic policies for a two-echelon supply chain with inventory and transportation procurement decisions”, *Transportation Research Part E: Logistics and Transportation Review*, Vol. 93, No. 1, pp. 57–70, 2016.

[12] N. S. Dhakry and P. A. Bangar, “Minimization of Inventory & Transportation Cost Of an Industry ” -A Supply Chain Optimization”, *International Journal of Engineering Research and Applications*, Vol. 3, No. 5, pp. 96–101, 2013.

[13] S. Rekha, B. Srividhya, and S. Vidya, “Transportation Cost Minimization : Max Min Penalty Approach”, *IOSR Journal of Mathematics (IOSR-JM)*, Vol. 10, No. 2, pp. 6–8, 2014.

[14] A. Ebrahimnejad, “A simplified new approach for solving fuzzy transportation problems with generalized trapezoidal fuzzy numbers”, *Applied Soft Computing Journal*, Vol. 19, No. 1, pp. 171–176, 2014.

[15] P. Kundu, S. Kar, and M. Maiti, “Fixed charge transportation problem with type-2 fuzzy variables”, *Information Sciences*, Vol. 255, No. 1, pp. 170–186, 2014.

[16] J. C. P. Yu, Y. S. Lin, K. J. Wang, and H. M. Wee, “Using AI Approach to Solve an Integrated Three-Echelon Supply Chain Model with Strategic Alliances”, In: *Proc. of First*

Asian Conference on Intelligent Information and Database Systems, Dong Hoi, Vietnam, pp. 259–264, 2009.

[17] İ. Küçükoğlu and N. Öztürk, "Simulated Annealing Approach for Transportation Problem of Cross-docking Network Design", *Procedia - Social and Behavioral Sciences*, Vol. 109, No. 2012, pp. 1180–1184, 2014.

[18] A. Király, T. Varga, and J. Abonyi, "Constrained Particle Swarm Optimization of Supply Chains", *International Journal of Industrial and Manufacturing Engineering*, Vol. 6, No. 7, pp. 1113–1121, 2012.

[19] Z. Qiongbing, "A New Crossover Mechanism for Genetic Algorithms with Variable-length Chromosomes for Path Optimization Problems", *Expert Systems With Applications*, Vol. 60, No. 1, pp. 183–189, 2016.

[20] M. Thakur and A. Kumar, "Electrical Power and Energy Systems Optimal coordination of directional over current relays using a modified real coded genetic algorithm : A comparative study", *International Journal of Electrical Power and Energy Systems*, Vol. 82, No. 1, pp. 484–495, 2016.

[21] H. Guo, X. Wang, and S. Zhou, "A Transportation Problem with Uncertain Costs and Random Supplies", *International Journal of e-Navigation and Maritime Economy,* Vol. 2, No. 1, pp. 1–11, 2015.

[22] W. F. Mahmudy, R. M. Marian, and L. H. S. Luong, "Hybrid genetic algorithms for multi-period part type selection and machine loading problems in flexible manufacturing system", In: *Proc. IEEE CYBERNETICSCOM 2013: IEEE International Conference on Computational Intelligence and Cybernetics*, 2013, pp. 126–130.

[23] Z.-Q. Chen and R.-L. Wang, "Solving the m-way graph partitioning problem using a genetic algorithm", *IEEJ Transactions on Electrical and Electronic Engineering,* Vol. 6, No. 5, pp. 483–489, 2011.

[24] V. N. Wijayaningrum and W. F. Mahmudy, "Optimization of ship's route scheduling using genetic algorithm", *Indonesian Journal of Electrical Engineering and Computer Science,* Vol. 2, No. 1, pp. 180-186, 2016.

[25] A. Rahmi, W. F. Mahmudy, and S. Anam, "A crossover in simulated annealing for population initialization of genetic algorithm to optimize the distribution cost", *Journal of Telecommunication, Electronic and Computer Engineering,* Vol. 9, No. 2–8, pp. 177-182, 2017.

[26] A. Rahmi and W. F. Mahmudy, "Regression modelling for precipitation prediction using genetic algorithms", *Telkomnika (Telecommunication Computing Electronics and Control),* Vol. 15, No. 3, pp. 1290–1300, 2017.

[27] F. Jia and D. Lichti, "A comparison of simulated annealing, genetic algorithm and particle swarm optimization in optimal first-order design of indoor tls networks", *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences,* Vol. IV-2/W4, No. 1, pp. 75–82, 2017.

[28] A. Kerr and K. Mullen, "A comparison of genetic algorithms and simulated annealing in maximizing the thermal conductance of harmonic lattices", *Computational Materials Science*, Vol. 157, No. 1, pp. 31–36, 2019.