This article can be cited as 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.

Genetic Algorithms for Optimization of Multi-Level Product Distribution

Asyrofa Rahmi[1], Wayan F. Mahmudy[2] and M. Zoqi Sarwani[3]

[1, 2,3] Faculty of Computer Science, University of Brawijaya,
8 Veteran Road, Malang, Indonesia
[3]Faculty of Information Technology, University of Merdeka,
68 Ir. H. Juanda Road, Pasuruan, Indonesia
Emails: asyrofarahmi@gmail.com, wayanfm@ub.ac.id, zaki_swr@yahoo.com

ABSTRACT

The process of wide product distribution to some areas consumes a very high cost. By minimizing the cost of the distribution process, companies can increase their profits. As algorithms that stochastically provide multiple diverse solutions, Genetic Algorithms (GAs) were proposed to solve the complex problems of multi-level distribution. To prevent premature convergence, preliminary numerical experiments were conducted to obtain the best parameters of GAs. Computational results of GAs that were adjusted by parameter testing were compared with the results of Random Search (RS) computation. The results indicated that there was an enormous difference in costs that resulted from computing GAs compared to RS in multi-level distribution problems.

Keywords: Multi-level distribution, genetic algorithms, random search.

Mathematics Subject Classification: 68T20, 68U01

Computing Classification System: I.2.8

1. INTRODUCTION

Challenges of every company in the scope of the global market force them to be more careful in managing company strategy. If they are not careful so as to not be able to compete with other companies, it is possible that they may not earn profits or even go into bankruptcy. As part of the supply chain, a profitability strategy by minimizing the cost of the distribution process is inevitable, (Sitek and Wikarek, 2012; Qiaolun and Tiegang, 2013). A distribution process that covers a wide area forces entrepreneurs to find ways to reduce costs in logistics and transportation (Sitek and Wikarek, 2012; Guo, Wang and Zhou, 2015). Such a distribution process with wide area coverage implements a multi-level distribution system, which is the process of transfer of finished product units from manufacturing to local distributors in different areas (distributors, retailers, agents, and so on) until the products arrive at the customer (Han and Kim, 2016; Langroodi and Amiri, 2016).

Some approaches had been used in distribution problems. Linear programming (LP) was used to resolve the problem of distribution. By using the Best Candidate Method (BCM) to choose the best candidate from some combination of solutions, the results obtained were nearly optimal with low computational time and reduced in complexity with solutions that are simple and clear. The weakness of the research is applying only one level (Hlayel and Alia, 2012).

Other studies had been done to resolve the distribution problems. In order to satisfy the customer demand, an iterative Lagrange-based heuristic was able to solve the problems of distribution. In small-sized instances, the proposed algorithm provided a better solution than its peers, whereas large-sized instances gave satisfaction performance by 2.2% of average gap. But in a relatively large company with more levels, these algorithms cannot be implemented because distribution problems are solved as distribution problems of two stages starting from manufacturing centers to warehouse and warehouse to retailers (Ardalan *et al.*, 2016).

Gauss Elimination method, Gauss-Jordan and Cramer's rule were also previously implemented to solve distribution problems. The obtained results showed that linearly, all three numerical models can be used in solving the problems of multi-level distribution with an acceptable solution. Although these three methods resulted in the same price, this approach could resolve the issues well. However, the constraints used were one vehicle for each shipper and there was no accounting for the capacity of the vehicle as well as the distribution unit stocks of the shipper.

Evolutionary computing approaches such as Genetic Algorithms (GAs) are widely used for problems in a variety of fields such as economics, industry, engineering, and so on (Ni and Wang, 2013). In addition to the fairly wide scope of being able to solve complex problems (Rahmi and Mahmudy, 2017), Genetic Algorithms could also solve optimization problems. Although there is another approach that can be used in solving optimization problem (Tomescu et al., 2007), GAs is reliable and commonly used such as optimize the parameters of mathematical model (Munyazikwiye, Karimi and Robbersmyr, 2017), parameter values (Guo, Peng and Tang, 2016), combinatorial problem with different levels of difficulty for optimization based on Phylogram Analysis (Soares, Râbelo and Delbem, 2017), ideal gas (IGO) (Shams *et al.*, 2017), bacterial foraging optimization by introducing linear variation and a nonlinear variation of chemotaxis step (Niu *et al.*, 2011), and frequency increments (Xiong *et al.*, 2017). It could also solve combinatorial problems (Abdoun, Abouchabaka and Tajani, 2012; Azim and Rahman, 2014).

This study is a development of the previous study. By using a variety of more developed problem definitions and data (each distributor having a wide range of vehicles), multi-level distribution problems are solved using Genetic Algorithms. The proposed solution is to create a model of a chosen vehicle-based distribution network so that the costs incurred in the distribution process are minimal. The results of the costs derived from Genetic Algorithms are compared with the random search algorithm as baseline to test how efficient and optimal Genetic Algorithms are in solving distribution problems based on the minimum cost.

2. PROBLEM DESCRIPTION

The data used in this study is in the form of simulated data that was designed based on results of surveys and interviews with distribution experts of a company. The data consists of the stock capacity of each company and distributor at every level, as well as data on the number of vehicles, vehicle capacity and cost spent.

The issue raised in the study was a multi-level distribution problem. The multiple levels in this study consist of 4 levels of plant/manufacturing, distributor center, agent, and retailer. For example, each level has some distributor unit such as 3 units of plants, 5 units of distribution centers, 8 units of agents, and 14 units of retailers. Each distributor unit has 1 to 3 vehicles. Each vehicle has a capacity from 250 to 1500 units and each distributor unit has available stock and minimal stock for ordering the product units.

Mathematically formulated, there are *L* levels of distribution. Each level has distributor units numbering to *U.* Each distributor unit has a vehicle number of *K* for serving product delivery and each vehicle has a capacity of K_{cap} as well as a fixed cost of *C* for each vehicle. Every distributor unit *D* that requests an order of *O* is serviced by the distributor unit located on the level above it. If the level above is not able to fulfill the order, it will initiate the order request on the level above that. *ST* is the status of the level that would serve the order request. If the *ST* has a value of 1 then the distribution level serves the request, and if *ST* is 0 the request is not served at the distribution level. To serve the order request, each distributor unit at every level has a stock of products numbering to *CAP*.

In the distribution process, the objective function of the problem is minimizing expenditures, because cost minimization is the solution to the problem of multi-level distribution. The objective function is formulated in equation (1).

$$Z = \sum_{i=0}^{l} \sum_{j=0}^{u} \sum_{m=0}^{d} X_{ijm} C_{imj} ST_i \tag{1}$$

Here *l* is the number of levels of distribution, *u* is the number of the distribution units of the shipper, *d* is the number of distribution unit customers, X_{ijm} is the number of product units sent by distributor unit *j* to distributor unit *m*, C_{imj} is a fixed cost that is incurred to send products from distributor unit *m* to distributor unit *j*, and ST_i is the status of whether or not distribution level *l* served the request.

The objectives to be achieved in resolving multi-level distribution is indicated by the objective function. However, before calculating the objective function, there are several things to note regarding the constraints. Such constraints are mathematically formulated, making them easier to process and depict.

2.1 THE CONSTRAINT FUNCTION FOR THE NUMBER OF ORDERS

The first constraint is related to the limit of the number of orders from the customers of the distribution units. The amount ordered by the customer should be the same as those sent to the customer. If the number of orders is not sufficient, then the system automatically processes the shortage of product units, taking from the level above. The function of the constraints for the number of orders is shown in equation (2).

$$\sum_{i=0}^{l}\sum_{j=0}^{u}\sum_{m=0}^{d} X_{ijm} = O_m \tag{2}$$

Here O_m is the request of orders on distributor *m.*

2.2 THE CONSTRAINT FUNCTION FOR VEHICLE CAPACITY

The second constraint is the capacity of the vehicles used. When the product units are sent to the customer or the level below, the product units are transported by vehicles. The problem is that the vehicles of each distributor unit shipper have a capacity limit that should not be exceeded. The influence is not only on the quality of the product units being delivered, but also the durability of the vehicle to prevent rapid deterioration. The function for vehicle capacity is shown in equation (3).

$$\sum_{n=0}^{u} X_{inm} \leq Kcap_n \tag{3}$$

Here X_{inm} is the number of product units to be shipped to distributor *m* by vehicle *n* and $Kcap_n$ is the transport capacity of the vehicle to *n*.

2.3 THE CONSTRAINT FUNCTION FOR THE DISTRIBUTION UNIT STOCKS OF THE SHIPPER

The last constraint is the stocks of every distributor unit. Each distributor always has a stock of availability of the product units. When a distributor unit acts as the shipper, it checks the stock so that the number of product units shipped does not exceed the stock. The function constraint to the stock of product unit of distributor unit shipper is shown in equation (4).

$$\sum_{j=0}^{l} X_{ij} \leq Cap_j \tag{4}$$

Here X_{ij} is the number of product units to be shipped by the distributor unit *j* and Cap_j is the stock supply of distributor unit *j.*

3. GENETIC ALGORITHMS (GAs)

Genetic algorithm is widely used to solve complex optimization problems. The imitation of the evolutionary process of a living organism in order to survive the selection process is the working principle of the Genetic Algorithm (Sen *et al.*, 2011). The working principle of evolution using the crossover, mutation, and selection operators have produced many solutions that are superior, efficient and of a good quality (Soni and Kumar, 2014; Rahmi and Mahmudy, 2017). A candidate solution is represented by a chromosome with a measurement of the solution quality, which is the *fitness* value. In multi-level product distribution case, the chromosome contains a combination of the number of products distribution from each distribution unit. A good *fitness* shows the superiority of the solution candidate. The objective function of the problem is minimizing expenditure. The lowest cost of multi-level distribution of a chromosome reflects that the chromosome has good fitness. Iteratively, GAs will provide nearly optimal solution candidates from the population of individuals / parents and the pool of new individuals / offspring based on the magnitude of the resulting *fitness* value.

3.1. CHROMOSOME REPRESENTATION AND CALCULATION OF *FITNESS*

The process of mapping the solution of the problem into a series of genes in a chromosome is known as chromosome representation. This chromosome representation is a core process in Genetic Algorithms. With a matching representation of chromosomes, a problem is solved by providing near-optimal solutions (Lianshuan and Huahui, 2015).

Real-code representation is appropriate for this problem because the purpose of product distribution is to suit the amount of product requested by the customer. To find the representation of chromosomes that are designed to solve the problems of genetic algorithm, it is necessary to find out the list of orders for the distributor units which represent the customers. Table 1 is the example list of orders placed across several distributor unit customers of different levels. It serves as the suitability of the structural design of the chromosome representation to be built.

Table 1. List of Orders of the Customer Distributor Unit

No.	Distributor Unit Customer	Number of Orders
1	Distributor Center 1	300
2	Distributor Center 2	260
3	Distributor Center 3	550
4	Agent 1	250
5	Agent 5	550
6	Agent 6	175
7	Agent 7	330
8	Retailer 8	100
9	Retailer 9	250

Based on the list of orders in Table 1, it can be seen that the multi-level process occurs in the distribution process. Figure 1 is the structure of chromosome representation of one level. The structure only shows the distributor center level to the plant level.

<table>
<tr><td colspan="8">Level *L*</td></tr>
<tr><td colspan="6">Pt1</td><td>..</td><td>Pt3</td></tr>
<tr><td colspan="3">V1</td><td colspan="3">V2</td><td></td><td></td></tr>
<tr><td>DC1</td><td>DC2</td><td>DC3</td><td>DC1</td><td>DC2</td><td>DC3</td><td></td><td></td></tr>
<tr><td>150</td><td>90</td><td>100</td><td>20</td><td>250</td><td>110</td><td>..</td><td></td></tr>
</table>

Figure 1. Chromosome Representation.

Figure 1 depicts a representation of a chromosome or a candidate solution. One chromosome is made up of several segments of a number of levels used in the distribution process (*l*). Figure 1 shows the chromosome with one segment because it uses only one level from plants (Pt) to the Distributor Center (DC). It can be said to be one level because DC1 acts as the distributor that should send product units from the plant to the next level until received by distributor unit customers.

The first gene contains 150, which indicates that Plant 1 distributes 150 product units to DC1 using Vehicle 1 (V1). The second gene indicates that Plant 1 distributes 90 product units to DC2 using Vehicle 1, and so on in the whole gene. In the process of chromosome representation, after having acquired the real value, the value should be checked first by the limiting constraints of the problem, such as constraints on vehicle capacity as represented by equation (2), the capacity limits of the vehicles as in equation (3) and the stock distribution unit stocks of the shipper as in equation (4). When calculated based on the overall level, the length of chromosomes in multi-level distribution problems are determined by the number of distributors that order product units on each level multiplied by the total number of vehicles owned by each distributor unit which serve orders on each level and multiplied by the number of levels.

Each chromosome has a *fitness* value as a reference of the ability to provide a solution. Because the main goal is to minimize the cost of distribution problems, as opposed to the *fitness* function in genetic algorithm to maximize the value, it can be concluded that the *fitness* function is obtained from the opposite objective function. As explained previously, the objective function is described in equation (1) with the *Z* value as the final value. The explanation based on meal *fitness* value is shown in equation (5).

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

3.2. INITIALIZATION POPULATION AND REPRODUCTION

The population is a set of chromosomes, individuals, or parents in accordance with the population size (*popSize*) at the start of the determination. If the specified population size is 10, then the population contains 10 chromosomes. In addition to population size, also requiring initial determination are the number of iterations or generations, as well as the value of the crossover rate (*cr*) and the mutation rate (*mr*).

The next process of the genetic algorithm and the main part of the calculation process is reproduction. Reproduction aims to generate several new individuals from parents in the population so that a variety of solution or offspring candidates are formed (Wayan Firdaus Mahmudy, Marian and Luong, 2013).

3.2.1 CROSSOVER OPERATOR

One of the operators for forming new individuals (offspring) in the reproduction process is the crossover operator. The number of offspring is obtained by multiplying the population size and the crossover rate (*popSize* * *cr*). Thus if the *cr* value is 0.4, then the number of offspring is 4.

This study uses an extended intermediate crossover model that uses random variables to determine how far the changes of the offspring are expected by the range limit (Wayan F Mahmudy, Marian and Luong, 2013). The equation for the extended intermediate crossover model is presented in equation (6).

$$C_1 = P_1 + \alpha(P_2 - P_1) \tag{6}$$

Here P is a parent and C is the offspring. α is a random variable that serves as a determinant of change in the offspring as produced by a certain range. The range used was between -0.25 and 1.25 which means that the exploration process was undertaken at an interval of 0.25 below 0 and above 1. 0.25 was chosen with the intention that the exploration process produced offspring that do not differ greatly from the parents that shape it. Figure 2 shows the crossover process.

In Figure 2, (C1) is from an extended intermediate process using two random parents, P1 and P2. The value of α is also generated and then calculated using equation (6).

P1	150	90	100	20	250	110	..	..
P2	100	30	40	350	150	120	..	..
α	0.14	0.5	0.26	0.78	0.39	0.98	..	..
C1	143	59	84	270	211	120	..	..

Figure 2. Intermediate Extended Crossover.

3.2.2 MUTATION OPERATOR

The insertion model is used for the mutation operator. The number of offspring was also obtained by multiplying the population size and mutation rate (*popSize* * *mr*). The process of insertion adds one gene randomly into another gene. Figure. 3 shows the insertion mutation process.

P1	150	90	100	20	250	110	..	..
C5	150	20	90	100	250	110	..	..

Figure. 3 Insertion Mutation.

The process in Figure. 3 begins with determining a parent randomly and P1 was selected. Next, randomly selected were 2 gene points and the second and fourth genes were selected. The offspring (C5) was obtained from the fourth gene being inserted to the second gene.

3.3 SELECTION

The selection used in solving the problems of multi-level distribution optimization is the Roulette wheel. The selection is done by calculating the cumulative probability of the *fitness* value. Then a number is randomly generated and this serves as a determinant for the specific individual cumulative probability *fitness* range.

4. EXPERIMENTAL RESULTS & DISCUSSION

Problems in the distribution of multi-level optimization were solved using genetic algorithms. In the computation process, the genetic algorithm had a few parameters that influenced in generating better candidate solutions. Some of these parameters are usually set in advance before the genetic algorithm process, namely the population size, the number of iterations, and the combined value of *cr* and *mr*. In order that the obtained results become close to optimal, tests were performed on several parameters. The parameter results that were close to optimal were seen from the average *fitness* value. The average *fitness* was obtained from *fitness* value of each size, which was tested ten times. The average *fitness* was used because genetic algorithms are stochastic in nature, because the value obtained constantly changes.

4.1 POPULATION TESTING

The initial test was to test population size. The most influential control parameter on GAs performance was population size (Wright and Alajmi, 2016). The initial parameters that were used were iteration 100, *cr* 0.7 and *mr* 0.3. The selected population size was at the point where the average *fitness* began to show a slight difference from the average *fitness* of the next population sizes. The slight difference showed that the points thereafter did not give a significant difference. Figure 4 shows the average *fitness* of population size testing.

Population size testing in Figure 4 used population sizes from 10 to 100 in increments of 10. Population 60 was the starting point of the average *fitness* which showed that for the next population size, the obtained difference of the average *fitness* was not too significant.

4.2 GENERATION/ITERATION TESTING

Iteration testing used the population size of the previous test results (60) with the values of *cr* as 0.7 and *mr* as 0.3. The numbers of iterations tested were in multiples of 100 from 100 to 500. Figure 5 shows the graph of iteration testing.

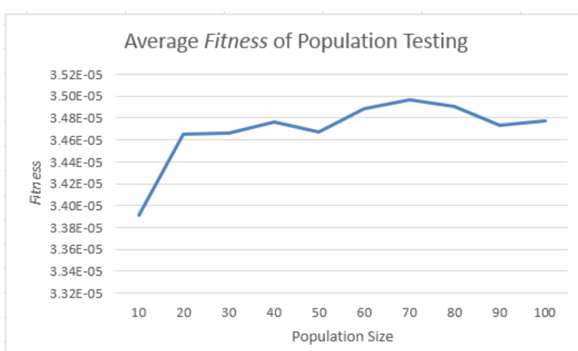

Figure 4. Population Testing.

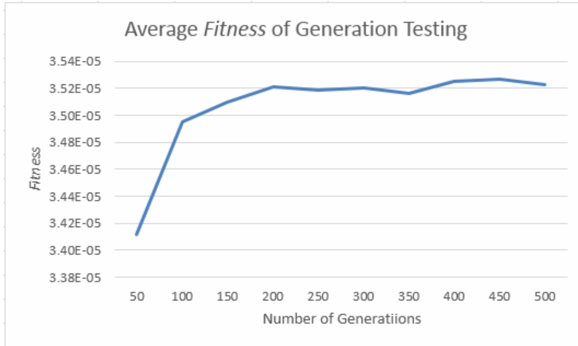

Figure 5. Iteration Testing.

The graph in Figure 5 showed that the 200th iteration was the starting point of average *fitness* which was not very significant for the following iterations. Thus, 200 iterations were selected.

4.3 CROSSOVER RATE AND MUTATION RATE TESTING

The last parameter tested was to find the best combination of the crossover rate (*cr*) and the mutation rate (*mr*) values. The population size and iteration parameters used were the selected values in earlier tests (60 and 200 respectively). The combinations of *cr* and *mr* used were in the range of 0-1 by multiples of 0.1. Figure 5 shows the graph of *cr* and *mr* testing.

Tested combinations of *cr* and *mr* are shown in Figure 5. For the problem of multi-level distribution optimization, the best combination with the average highest *fitness* was for the values of *cr* as 0.3 and *mr* as 0.7. This showed that for the problem of multi-level distribution, a deeper exploitation process gave more candidates for a nearly optimal solution.

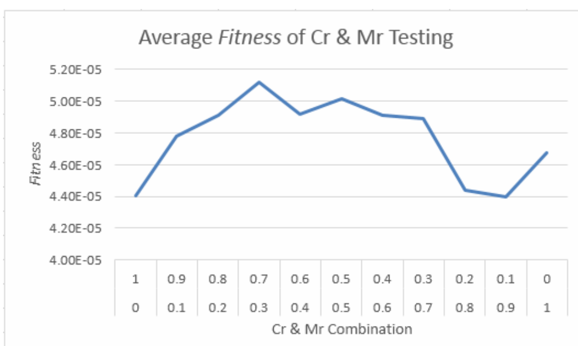

Figure 6. *Cr* & *Mr* Testing.

4.4 RESULT ANALYSIS

For testing the parameters of GAs, the previously obtained best parameters for solving the optimization problem of multi-level distribution are shown in Table 2.

Table 2. The Best GAs Parameters

GAs Parameter	**Number**
Population Size	60
Generation	200
Cr	0.3
Mr	0.7

The best parameters in Table 2 were used to test how well the GAs were in solving the multi-level distribution optimization problem. Based on the parameters in Table 2, the GAs were processed 10 times to obtain more accurate test results. The test results of GAs were compared to a heuristic algorithm, Random Search (RS). As with the GAs, RS was run 10 times; the results of the comparison of both methods are shown in Figure 7.

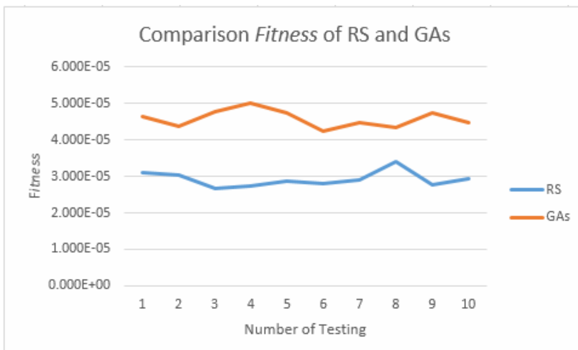

Figure 7. Comparison of RS and Gas.

Figure 7 showed that although both methods were run as many as 10 times, the overall *fitness* trial results of the GAs process performed better. From the 10 trials for each method, the average *fitness* as well as the average cost were obtained. The average *fitness* and cost are presented in

Table **3.** The Average *Fitness* and Cost of Both Methods

Methods	***Average Fitness***	***Average Cost***
Genetic Algorithms	4.585485563E-05	21860.4
Random Search	2.92655777E-05	34328

From **Error! Not a valid bookmark self-reference.** it could be seen that the average *fitness* and cost for each method differed greatly. Specifically, the average cost had a difference of 12467.6. This showed that GAs were able to produce better results compared to the results of the Random Search.

Table 3. The Average *Fitness* and Cost of Both Methods

Methods	***Average Fitness***	***Average Cost***
Genetic Algorithms	4.585485563E-05	21860.4

Random Search	2.92655777E-05	34328

From **Error! Not a valid bookmark self-reference.** it could be seen that the average *fitness* and cost for each method differed greatly. Specifically, the average cost had a difference of 12467.6. This showed that GAs were able to produce better results compared to the results of the Random Search.

5. CONCLUSION

In this study, the problem of multi-level distribution optimization aimed to produce a minimal cost. The used constraints of multi-level distribution were the number of orders, the capacity of the vehicles, as well as the distribution unit stocks of the shipper. The research problem was solved using Genetic Algorithms. An extended intermediate model was used as the crossover model and the insertion model was used for the mutation models. To obtain near-optimal solutions, the parameters of GAs were tested to obtain a population size of 60, 200 iterations, and *cr* and *mr* values of 0.3 and 0.7.

The best parameters of GAs were tested for results by processing them 10 times for near-optimal results. For tests repeated 10 times and the average yield of the final *fitness*, GAs were compared with RS. The obtained *fitness* results showed that GAs were better than RS. Based on the cost, the result of GAs was 21860.4 and RS was 34328. The result of GAs was smaller with a difference of 12467.6.

In this study, the products used were only of one type, the cost for *fitness* is considered as a fixed cost only, and the genetic algorithms used were general. Suggestions for the development of further research are to search for other genetic operators that give better results or to update genetic algorithms for optimization, modification, hybridization, and adaptiveness. In the real case of distribution, the costs to be considered should not only be the fixed cost but also the variable cost and the resulting products of a company could be more than one type.

REFERENCES

Abdoun, O., Abouchabaka, J. and Tajani, C. (2012) ‘Analyzing the performance of mutation operators to solve the travelling salesman problem’, *International Journal of Emerging Sciences*, 2(1), pp. 61–77.

Ardalan, Z., Karimi, S., Naderi, B. and Khamseh, A. A. (2016) ‘Supply chain networks design with multi-mode demand satisfaction policy’, *Computers & Industrial Engineering*, 96, pp. 108–117.

Azim, I. and Rahman, F. (2014) ‘Genetic algorithm based reactive power management by SVC’, *International Journal of Electrical and Computer Engineering*, 4(2), pp. 200–206.

Guo, F., Peng, H. and Tang, J. (2016) ‘Genetic algorithm-based parameter selection approach to single image defogging’, *Information Processing Letters*, 116(10), pp. 595–602.

Guo, H., Wang, X. and Zhou, S. (2015) ‘A Transportation problem with uncertain costs and random supplies’, *International Journal of e-Navigation and Maritime Economy*, 2, pp. 1–11.

Han, J.-H. and Kim, Y.-D. (2016) ‘Design and operation of a two-level supply chain for production-time-dependent products using Lagrangian relaxation’, *Computers & Industrial Engineering*, 96, pp. 118–125.

Hlayel, A. A. and Alia, M. A. (2012) ‘Solving transportation problems using the best candidates method’, *Computer Science & Engineering*, 2(5), pp. 23–30.

Langroodi, R. R. P. and Amiri, M. (2016) ‘A system dynamics modeling approach for a multi-level, multi-product, multi-region supply chain under demand uncertainty’, *Expert Systems with Applications*, 51, pp. 231–244.

Lianshuan, S. and Huahui, W. (2015) ‘A multi-agent self-adaptive multi-objective genetic algorithm’, *International Journal of Intelligent Engineering & Systems*, 8(2), pp. 7–13.

Mahmudy, Wayan F, Marian, R. M. and Luong, L. H. S. (2013) ‘Hybrid genetic algorithms for multi-period part type selection and machine loading problems in flexible manufacturing system’, in *IEEE International Conference on Computational Intelligence and Cybernetics*. Yogyakarta, Indonesia, 3-4 December: IEEE, pp. 126–130.

Mahmudy, Wayan Firdaus, Marian, R. M. and Luong, L. H. S. (2013) ‘Real coded genetic algorithms for solving flexible job-shop scheduling problem - Part I: Modelling’, *Advanced Materials Research*, 701, pp. 359–363.

Munyazikwiye, B. B., Karimi, H. R. and Robbersmyr, K. G. (2017) ‘Optimization of vehicle-to-vehicle frontal crash model based on measured data using genetic algorithm’, *IEEE Access*, 5, pp. 3131–3138.

Ni, H. and Wang, Y. (2013) ‘Stock index tracking by Pareto efficient genetic algorithm’, *Applied Soft Computing*, 13(12), pp. 4519–4535.

Niu, B., Fan, Y., Wang, H., Li, L. and Wang, X.-F. (2011) ‘Novel bacterial foraging optimization with time-varying chemotaxis step’, *International Journal of Artificial Intelligence*, 7(11), pp. 257–273.

Qiaolun, G. and Tiegang, G. (2013) ‘Profits management for reverse supply chain with supply disruption’, *International Journal of Intelligent Engineering & Systems*, 6(1), pp. 1–9.

Rahmi, A. and Mahmudy, W. F. (2017) ‘Regression modelling for precipitation prediction using genetic algorithms’, *Telkomnika*, 15(3), pp. 1290–1300.

Sen, S., Roy, P., Chakrabarti, A. and Sengupta, S. (2011) ‘Generator contribution based congestion management using multiobjective genetic algorithm’, *Telkomnika*, 9(1), pp. 1–8.

Shams, M., Rashedi, E., Dashti, S. M. and Hakimi, A. (2017) ‘Ideal gas optimization algorithm’, *International Journal of Artificial Intelligence*, 15(2), pp. 116–130.

Sitek, P. and Wikarek, J. (2012) ‘Mathematical programming model of cost optimization for supply chain from perspective of logistics provider’, *Management and Production Engineering Review*, 3(2), pp. 49–61.

Soares, A., Râbelo, R. and Delbem, A. (2017) ‘Optimization based on phylogram analysis’, *Expert Systems with Applications*, 78, pp. 32–50.

Soni, N. and Kumar, T. (2014) ‘Study of various mutation operators in genetic algorithms’, *International Journal of Computer Science and Information Technology*, 5(3), pp. 4519–4521.

Tomescu, M. L., Preitl, S., Precup, R.-E. and Tar, J. K. (2007) ‘Stability analysis method for fuzzy control systems dedicated controlling nonlinear processes‘, *Acta Polytechnica Hungarica*, 4(3), pp. 127–141.

Wright, J. and Alajmi, A. (2016) ‘Efficient genetic algorithm sets for optimizing constrained building design problem’, *International Journal of Sustainable Built Environment*, 5(1), pp. 123–131.

Xiong, J. *et al.* (2017) ‘Frequency diverse array transmit beampattern optimization with genetic algorithm', *IEEE Antennas and Wireless Propagation Letters*, 16, pp. 469–472.