Contents lists available at GrowingScience

Decision Science Letters

homepage: www.GrowingScience.com/dsl

A novel crossover operator for genetic algorithm: Stas crossover

Ratchadakorn Poohoi[a*], Kanate Puntusavase[a] and Shunichi Ohmori[b]

[a]*Department of Industrial Engineering and Management, Faculty of Engineering and Industrial Technology, Silpakorn University, Nakhon Pathom, 73000, Thailand*
[b]*Graduate School of Creative Science and Engineering, Waseda University, Tokyo 169-8050, Japan*

CHRONICLE

Article history:
Received: December 12, 2022
Received in revised format: January 12, 2023
Accepted: April 24, 2023
Available online: April 24, 2023

Keywords:
Genetic Algorithm
Stas crossover
Crossover operator

ABSTRACT

The genetic algorithm (GA) is a natural selection-inspired optimization algorithm. It is a population-based search algorithm that utilizes the concept of survival of the fittest. This study creates a new crossover operator called “Stas Crossover” that is a combination of four crossover operators, including Single point crossover, Two points crossover, Arithmetic crossover, and Scattered crossover, and then presents the performance of this crossover operator. The area size and probability of Stas crossover can be adjusted.GA is used to find the optimal solution for this multi-product and multi-period aggregate production planning (APP) problem, which was used to test the algorithm, which provides optimal levels of inventory, backorders, overtime and regular production rates, and other controllable variables. According to the findings of this study, the benefit of stable crossover is that it allows for more variety in the way offspring are created and increases the opportunity for offspring to obtain good genetic information directly.

1. Introduction

The genetic algorithm (GA) is a natural selection-inspired optimization algorithm. Natural selection, according to Darwin's theory, favors the fittest individuals who reproduce. This concept was developed by Goldberg (1989). It is a population-based search algorithm that utilizes the concept of the survival of the fittest. The new populations are created by iterative use of genetic operators on individuals present in the population. GA, according to this idea, is made up of three basic phases: selection, reproduction, and mutation, all of which try to copy the working mechanisms of nature (Burke & Kendall, 2005). GA is mostly used to heuristically locate the global optimal solution. Katoch et al. (2020) discussed genetic algorithms: past, present, and future. This structured review will be helpful for future research.

Aggregate production planning (APP) is a method for developing an overall manufacturing plan that ensures uninterrupted production at a facility. APP is associated with the determination of production, inventory, and labor levels to fulfill varying demand over a planning perspective that ranges from a period of six months to one year (Krajewski & Ritzman, 1999) with limited capacity and resources (Wang & Yeh, 2014). The goal of aggregate planning is to minimize operating costs by matching production demand with production capacity (Pradenas et al., 2004). APP helps manufacturers maximize a facility’s productivity and achieve their financial goals. It helps to get the maximum benefit from the available production capabilities while meeting customer demand and reducing the cost of excess inventory.

In the literature reviewed, the theory and the case study for guidelines of research found that they use the GA to solve the proposed aggregate product model. This algorithm is a heuristic search algorithm based on the idea of natural selection that occurs in the process of evolution and genetic operations, which has become very popular in solving APP. Ioannis (2009)

* Corresponding author.
E-mail address: poohoi_r@silpakorn.edu (R. Poohoi)

doi: 10.5267/dsl.2023.4.010

described a novel genetic algorithm for the problem of constrained optimization. Lambora et al. (2019) discussed a genetic algorithm work and the process is included.

APP is related to determining the optimum production and workforce levels for each period over the medium-term planning horizon. Cheraghalikhania et al. (2019) reviewed the literature on APP models and groups two main purposes. First, a structured classification scheme for APP models is proposed. Second, gaps that exist in the literature have been shown to extract future directions of their research area. Moghaddam and Safaei (2006) have presented a genetic algorithm for solving a generalized model of single-item resource constrained APP with linear cost functions.

Many researchers have developed an integrated approach to solve APP problems and present many models that combine different algorithms and techniques to solve the problem. In this study, the authors used GA to solve the APP problem. Savsani et al. (2016) described a GA approach for solving APP with different selection methods and various crossover phenomena. The principles of natural selection and genetics are the foundation of GA, which are search algorithms. Mahmud and Hossain (2018) developed an interactive possibilistic environment-based GA for multi-product and multi-period APP and it has been solved by the multi-objective genetic algorithm to minimize the production cost and the rate of changing imprecise parameters. Ahmed et al. (2019) proposed a model that attempts to incorporate all the relevant cost factors into the optimization model that are directly or indirectly affected by the APP.

However, researchers have formulated a multi-product and multi period aggregate planning problem model considering various decision variables and using different solution techniques. Many studies point to a future research direction for the aggregate planning large-size problem. The limitation of the existing research is the size of the problem and the diversity of ways to select the way to create offspring in crossover. Chakrabortty and Hasin (2013) studied two products, a two-month period with different population sizes, and a maximum population size of 360. Savsani (2016) presented two products, a two-month period, and a maximum of 100 different experimental runs. Dakka et al. (2017) have formulated a model for five products, 10 months, and a maximum of 50 different experimental runs obtained for 10 runs.

The main goal of this paper is to introduce a new crossover operator called Stas Crossover, that is a combination of four crossover operators and present the performance of this crossover operator. Stas crossover can adjust the size of the area probability. The rest of this paper is organized as follows: In section 2, definitions and concepts for the various crossover operators are introduced and a proposed crossover operator, Stas Crossover is proposed. Subsequently, Section 3 presents some numerical results and discussion to show the application of the proposed cost optimization model. The final section provides a conclusion.

2. Problem formulation

2.1 Problem Description & Notation

In this paper, A linear programming model is used to describe the multi-product and multi-period APP problem. Here, the assumption is made for the planning horizon t, in which the source manufactures a product to meet people's market demand. GA is used to find the optimal solution for this APP, which provides optimum levels of inventory, backorders, overtime and regular production rates, and other controllable variables.

The following notation is used after reviewing the literature and considering practical situations (Savsani, 2016; Chakrabortty & Hasin, 2013).

Let t = *1, 2, ..., T* be the production planning period set, where T is the number of periods in the planning horizon. Let *i* = *1, 2, ..., I* represents the product category set, where I represents the number of product categories, and *j* represents the source category set, where J represents the number of source categories.

Parameters

R_{ijt}	Regular time production cost for product i manufactured from source j in period t ($/units)
O_{ijt}	Overtime production cost for product i manufactured from source j in period t ($/units)
V_{ijt}	Inventory carrying cost for product i from source j in period t ($/units)
B_{ijt}	Backorder cost for product i m from source j in period t ($/units)

Variables

D_{it}	Forecasted demand of product i in period t (units)
X_{ijt}	Production Quantity of product i manufactured from source j at regular time in period t (units)
Y_{ijt}	Production Quantity of product i manufactured from source j at overtime in period t (units)
W_{ijt}	Inventory of product i in source j at the end of period t (units)
M_{ijt}	Backorder of product i in source j at the end of period t (units)

2.2 Single Objective APP Model

The aim of this study is to find an optimal APP with minimal total cost. The total cost is the sum of the production costs. Accordingly, the objective function of the purpose model is as follows:

$$\min Z = \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{t=1}^{T}\left[R_{ijt}X_{ijt} + O_{ijt}Y_{ijt} + V_{ijt}W_{ijt} + B_{ijt}M_{ijt}\right] \tag{1}$$

The total production cost is mainly considered in this paper, where the total production cost includes four components: regular time production, overtime production, carrying inventory, and backordering cost.

2.3 Constraints

The objective functions formulated in the previous section are restricted by two sets of constraints. They are the carrying inventory constraint and the non-negative constraint.

Constraint on Carrying Inventory

$$W_{ijt} - M_{ijt} = W_{ij(t-1)} - B_{ij(t-1)} + X_{ijt} + Y_{ijt} - D_{it} \quad \text{for } \forall i \forall t \tag{2}$$

$$W_{ijt} \geq W_{ijt\,\min} \quad \text{for } \forall i \forall t \tag{3}$$

$$M_{ijt} \leq M_{ijt\,\min} \quad \text{for } \forall i \forall t \tag{4}$$

where, D_{it} denotes the imprecise forecast demand of ith product in period t. APP problem in the real world, The forecast demand D_{it} cannot be accurately obtained in a dynamic market. In the first constraint equation, the sum of regular and overtime production, inventory levels, and backorder levels essentially should equal the market demand. Demand over a specific period can be either met or backordered, but a backorder must be completed in the next period.

Non-Negative Constraint

$$X_{ijt}, Y_{ijt}, W_{ijt}, M_{ijt} \geq 0 \quad \text{for } \forall i, \forall t \tag{5}$$

2.4 Genetic Algorithm

The term “Genetic Algorithm” describes a set of methods that can be used to optimize complex problems (Mirjalili, 2019). As the name suggests, the processes employed by GA are inspired by natural selection and genetic variation. To achieve this, a GA uses a population of possible solutions to a problem and applies a series of processes to them (Alam et al., 2020)

In keeping with the evolutionary theme, each individual in a GA population is represented by a chromosome. As in nature, this chromosome contains genetic information relating to each individual’s characteristics (Mitchell, 1996). The steps of GA are discussed one by one (Kramer, 2017).

Step 1: Begin by randomly generating a population of n chromosomes.

Step 2: To determine the population's fitness for each chromosome.

Step 3: Repeat the following step:

- From the current population, choose a pair of parent chromosomes.

- At a random point, crossover the pair to produce two offspring.

- Perform a mutation test on new offspring at each locus.

Step 4: Replace the current population with the new population.

Step 5: If the end condition is not satisfied, go back to step 3. If the end condition is satisfied, pause and select the best solution within the current population.

In this study, the production quantity of product for each period PPit and the cost of production Ct, the decision variables, both are positive integers, which are conducive to the implementation of chromosome encoding. For sub-chromosome P, an integer I × T matrix is adopted to represent the chromosome of production quantity of product, where I represents the number of product types and T is the number of periods. Fig. 1. shows a chromosome with T periods and I product types, in which gene PPit represents the production quantity of product and gene Ct represents the cost of production in each period

	Period (t)	1	2	···	T
P	Type 1	PP11	PP12	···	PP1T
	Type 2	PP21	PP22	···	PP2T
	···		···		
	Type I	PPI1	PPI2	···	PPIT
C	Cost of Production	C1	C2	···	CT

Fig. 1. A simple example of chromosome

2.4.1 Crossover operators

Crossover operators specify how the GA form a new individual or offspring by combining two individuals, or parents for the next generation (Chakrabortty & Hasin, 2013; Mathew, 2012). For the purpose of this work, only crossover operators that operate on two parents and combination of four crossover operators including Single point crossover, Two points crossover, Arithmetic crossover, and Scattered crossover. Single point crossover chooses a random position in the entire chromosomal length of the individual. This point is called a cross-site or crossover point. For the first individual the values of the bits for the head of the chromosomes are taken from the first parent and the tail section is taken from the second parent. For example, if p1 and p2 are the parents such as p1 = [s t u v w x y z] and p2 = [1 2 3 4 5 6 7 8] and the random integer is 3. Then, the offspring would have [s t u 4 5 6 7 8] (Kramer, 2017). Two points crossover is chosen and these points are exchanged between two mated parents. For the first individual the values of the bits for head and tail of the chromosomes are taken from the first parent and the middle section is taken from the second parent. For example, if p1 and p2 are the parents such as p1 = [s t u v w x y z] and p2 = [1 2 3 4 5 6 7 8] and the random values are 2 and 4. Then, the offspring would have [s t u 4 5 6 y z] (Sivanandam & Deepa, 2008; Shukla et al., 2010). Arithmetic crossover is a crossover operator that linearly combines chromosome vectors of both the parents to produce two new offspring according to the following equations (Eiben & Smith, 2003).

Offspring1=a×Parent1 + (1-a) ×Parent2

Offspring2=(1–a) ×Parent1 + a×Parent2

when a is a random weighting factor (chosen before each crossover operation).

Scattered crossover is to select the parent bits from randomly made at the time of the crossover. Some bit positions are randomly selected, these bits from the first parent and the other from the second parent another individual is to select the converse bit. Two new individuals are made for the next generation. For example, if p1 and p2 are the parents such as p1 = [s t u v w x y z] and p2 = [1 2 3 4 5 6 7 8] and the binary vector is [1 1 0 0 0 1 0 0 0]. Then, the function returns the following offspring = [s t 3 4 w 6 7 8] (Shukla et al., 2010).

2.5 *A proposed crossover operator: Stas Crossover*

The new crossover operator in this study is a combination of four crossover operators, including Single point crossover, Two points crossover, Arithmetic crossover, and Scattered crossover. The crossover operator is used to combine the genetic information of two parents to generate new offspring. The probability of each operator being the same to generate new offspring is equal to 25%. The new crossover operator is called Stas Crossover. The following Fig. 2. illustrates Stas crossover process, which shows that the new offspring would have an equal probability of occurrence. Stas crossover can be adjusted to the size of the area probability in each operator. The probability of generating new offspring for each operator is unequal. That results in greater variety than traditional crossover. To adjust the size of the area probability, Single point crossover and Two points crossover don't have an area probability because it is assumed that both of these operators have too few improvements and more improvements are needed. It is divided into two parts with a 50% area probability for Arithmetic crossover and Scattered crossover. It is called the Stas0055 crossover. The numbers after Stas represent the probability of each operator. There is also an adjusted probability of Single point crossover of 10%, Two points crossover of 10%, Arithmetic crossover of 10%, and Scattered crossover of 70%. It is called Stas1117 crossover. Fig. 3. Illustrates Stas1117 crossover process, which shows that the offspring would not have an equal probability of occurrence. Its random chance of occurrence depends on the size of the area probability.

Stas Crossover operator selects two parents, such as p1 and p2, for the crossover process to place all the crossover operators on a roulette wheel with an equal area probability. The roulette is turned and the crossover operators are selected at random crossover operators, which chooses single point crossover, two points crossover, arithmetic crossover, and scattered crossover to determine the direction of the way to create offspring. The corresponding crossover options are selected to create the new offspring according to the following conditions: Each experimental run is random, which means that if an experimental run is obtained for 10 runs, it selects different crossover operators in each run. For example, if the first experimental run is selected as a Scattered crossover, that means the second experimental run may be to choose Scattered crossover again or choose the other crossover operators.

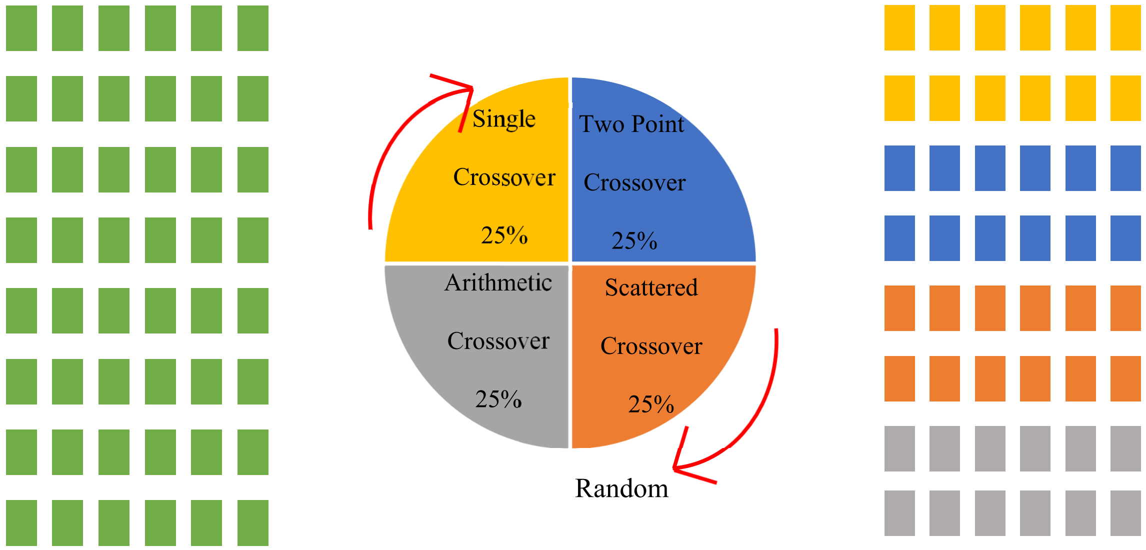

Fig. 2. Illustrates Stas crossover process

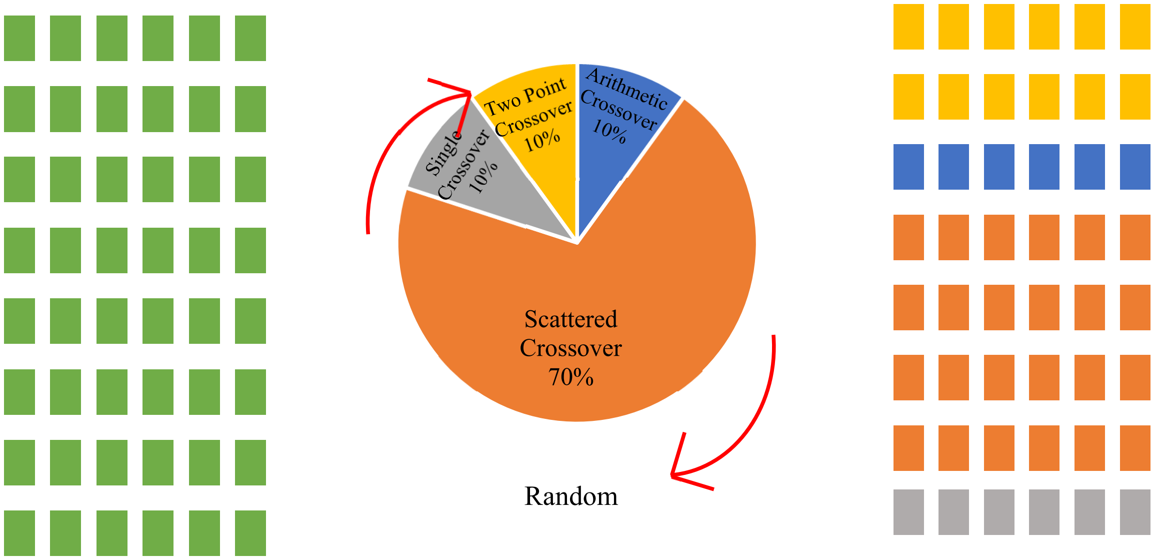

Fig. 3. Illustrates Stas1117 crossover process

3. Model Implementation

In this study, an APP problem with multi-product and multi-period was used to test the algorithm. Problems are generated with various condition parameters by defining the size of the problem. This study focuses on the size of the problem and optimal solution by changing production costs, inventory costs, backorder costs, production capacity, and forecast demand. Therefore, the actual value of the answer is ignored.

According to the introductory information, Table 1 and Table 2 summarize production costs, inventory costs, backorder costs, and the sources of production capacities in the planning horizon for different products. The problem has 6 periods, 12 periods, and 24 periods planning horizon. The forecast demand quantities are shown in Table 3. Following that, P denotes the period and S denotes the source. Therefore, 2P12P6S means 2 products, 12 periods, and 6 sources.

Table 1
Production, inventory and backorder cost

Product i	Cost of RT (bath/unit)	Cost of OT (bath/unit)	Inventory cost (bath/unit)	Backorder cost (bath/unit)
1	10	12	2	2
2	12	14	2	3
3	14	16	2	4
4	10	12	2	2
5	12	14	2	3

Table 2
Production capacity

Source	Capacity per Period							
	2 Products		3 Products		4 Products		5 Products	
6 Sources	160		200		180		220	
12 Sources	80		150		120		150	
24 Source	RT	OT	RT	OT	RT	OT	RT	OT
	100	50	100	50	100	50	120	60

Table 3
Forecast Demand

Size of Problem	Product i	Period i											
		1	2	3	4	5	6	7	8	9	10	11	12
2P12P6S	Product 1	45	40	60	25	45	55	30	40	35	35	40	30
	Product 2	55	40	55	30	30	35	45	30	55	30	40	35
2P12P12S	Product 1	45	40	60	25	45	55	30	40	35	35	40	30
	Product 2	55	40	55	30	30	35	45	30	55	30	40	35
2P12P24S	Product 1	70	85	85	60	75	80	65	75	65	85	80	80
	Product 2	75	60	90	70	90	75	70	75	60	70	85	75
3P12P6S	Product 1	35	25	35	30	40	30	35	30	35	35	30	40
	Product 2	30	25	30	35	40	35	25	30	25	30	40	35
	Product 3	30	35	45	30	25	30	35	40	45	40	35	30
3P12P12S	Product 1	60	65	45	60	35	65	45	60	55	50	45	35
	Product 2	40	35	50	55	45	50	65	35	60	35	45	55
	Product 3	35	55	35	60	60	65	55	35	55	65	35	55
3P12P24S	Product 1	50	65	50	45	55	40	55	45	50	55	45	50
	Product 2	65	40	50	65	50	50	55	45	40	40	65	40
	Product 3	50	40	45	45	40	45	60	65	60	45	45	50
4P12P6S	Product 1	20	25	15	30	25	25	30	15	15	25	20	25
	Product 2	30	15	30	25	15	30	25	25	30	20	15	30
	Product 3	20	20	30	15	15	20	25	15	30	30	15	25
	Product 4	25	15	25	20	15	25	15	30	20	30	25	15
4P12P12S	Product 1	30	20	35	30	35	25	30	30	40	35	25	30
	Product 2	25	35	35	40	35	25	25	40	45	40	25	40
	Product 3	25	35	20	25	30	25	40	25	35	20	30	25
	Product 4	40	25	25	25	40	25	30	25	20	25	25	25
4P12P24S	Product 1	35	45	40	30	45	35	40	35	35	30	45	40
	Product 2	45	40	45	30	30	45	30	40	40	35	30	40
	Product 3	30	35	45	40	45	35	35	40	45	50	30	35
	Product 4	40	35	30	30	40	45	30	45	40	35	30	30
5P12P6S	Product 1	20	25	15	30	25	25	30	15	15	25	20	25
	Product 2	30	15	30	25	15	30	25	25	30	20	15	30
	Product 3	20	20	30	15	15	20	25	15	30	30	15	25
	Product 4	25	15	25	20	15	25	15	30	20	30	25	15
	Product 5	25	20	25	20	15	25	15	20	20	15	20	20
5P12P12S	Product 1	30	20	35	30	35	25	30	30	40	35	25	30
	Product 2	25	35	35	40	35	25	25	40	45	40	25	40
	Product 3	25	35	20	25	30	25	40	25	35	20	30	25
	Product 4	40	25	25	25	40	25	30	25	20	25	25	25
	Product 5	25	30	30	20	25	30	30	25	40	40	30	35
5P12P24S	Product 1	35	45	40	30	45	35	40	35	35	30	45	40
	Product 2	45	40	45	30	30	45	30	40	40	35	30	40
	Product 3	30	35	45	40	45	35	35	40	45	50	30	35
	Product 4	40	35	30	30	40	45	30	45	40	35	30	30
	Product 5	30	35	25	30	25	25	25	35	40	35	35	20

The aim of this research is to introduce a new crossover operator called Stas Crossover on multi-product and multi-period APP problems, that is a combination of four crossover operators. Various combinations of crossovers are tested for APP problems. Results are obtained for 10 iterations and compared according to different statistical values with a line chart. The population size is 1500, 1000, 600, 500, 150, and 100, respectively. The number of generations is 20, 30, 50, 60, 200, and 300, respectively, and the number of runs which have been considered for the experimental run is 30,000.

4. Results and Discussion

The aim Stas crossover for the GA approach can solve multi-product and multi-period APP problems with an interactive decision variable. In this research, problems are defined depending on the size of the problem, and 40 different crossover options are tested for 12 types of problems, and 10 runs are reported for each combination. These four scenarios have been shown in Fig. 4.

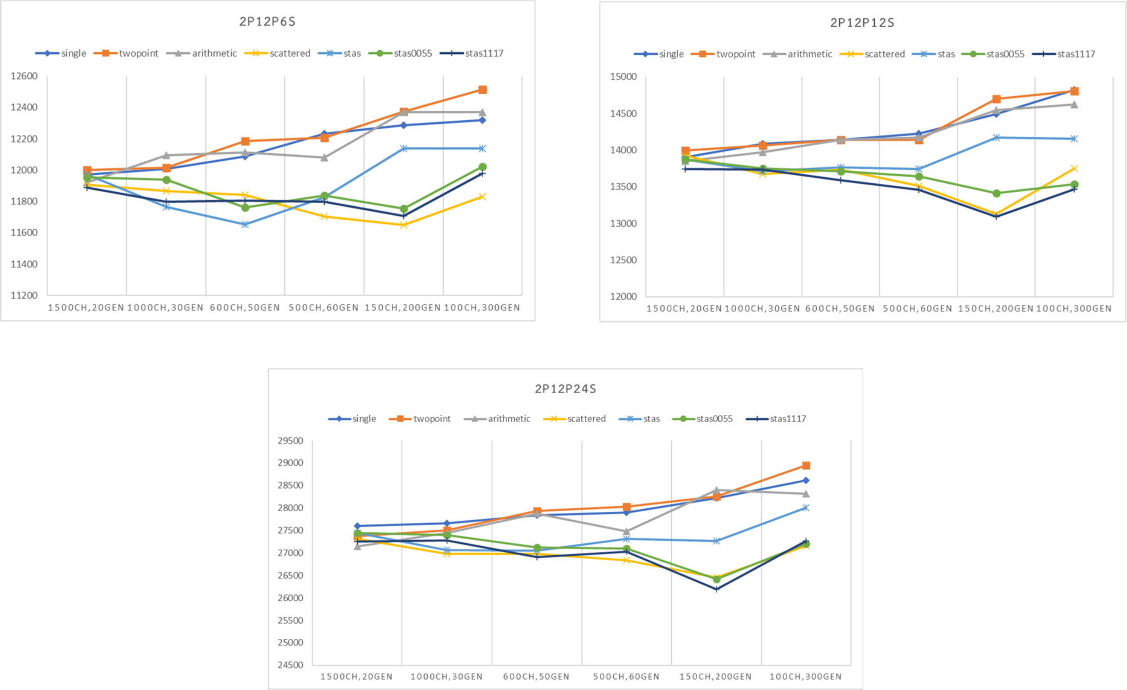

Fig. 4. (a) Scenario of 2 products

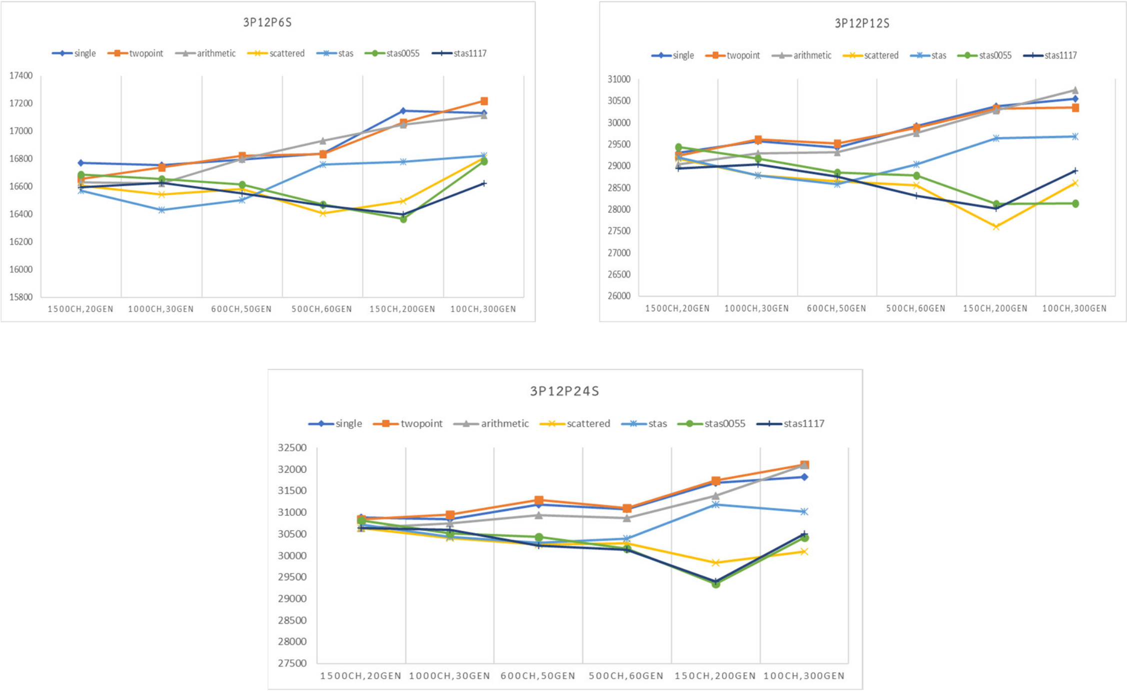

Fig. 4. (b) Scenario of 3 products

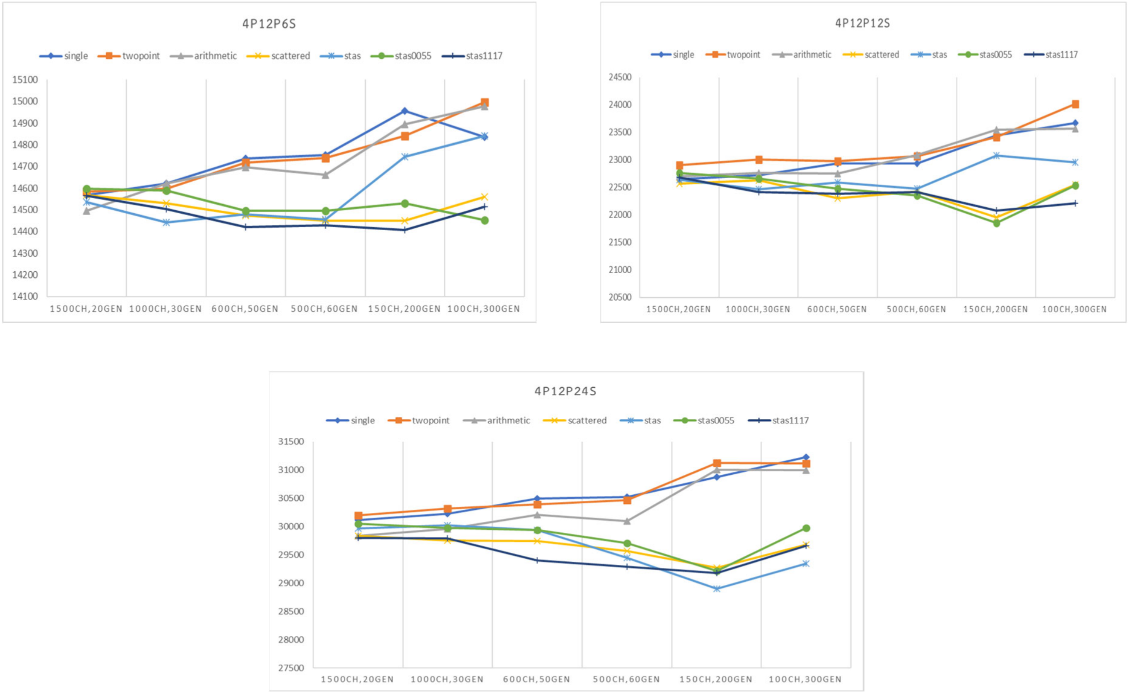

Fig. 4. (c) Scenario of 4 products

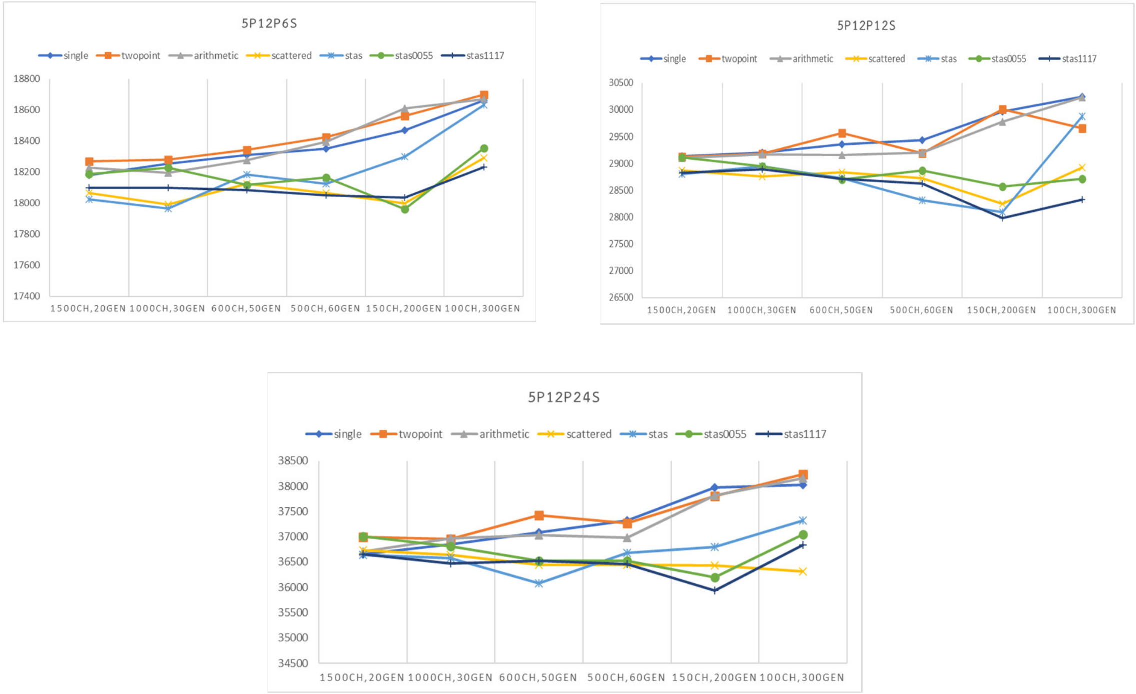

Fig. 4. (d) Scenario of 5 products

Fig. 5. shows the number of times to find the best performance for each option. Stas1117 crossover is able to find the best performance 28 out of 72 times. It can be found that by changing the area probability, it’s better than other operators.

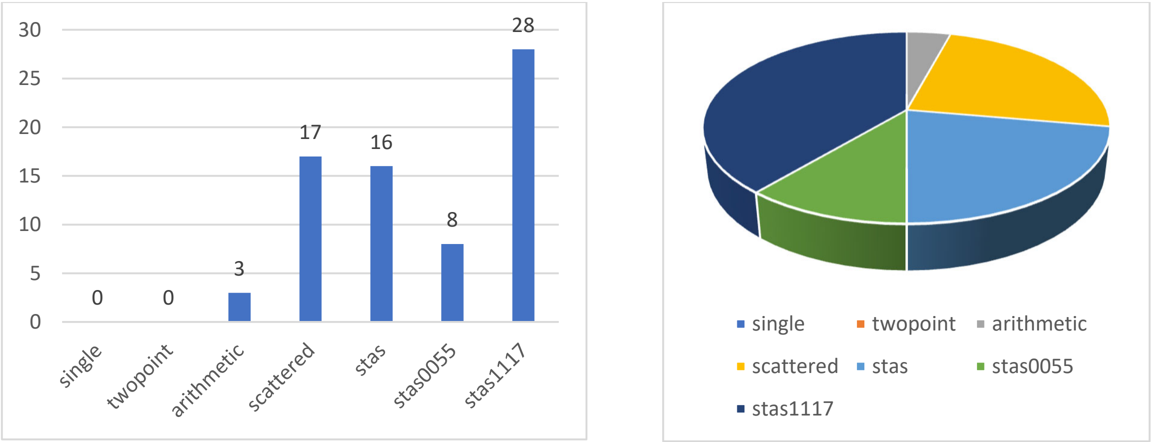

Fig. 5. Proportion of the best performance for each option

Table 4 shows the cost value obtained by using different combinations of crossover procedures. From the results, it can be seen that Stas0055 crossover and Stas1117 crossover perform better than Single point crossover, Two points crossover, and Arithmetic crossover. In addition, it can be seen that the population size and number of generations of the best possible answer are 150 chromosomes and 200 generations, respectively. It can be obtained 8 times out of 12 times the problem size.

Table 4
Cost value obtained by using different combinations of crossover procedures

Size of Problem	Chromosome/Generation	Crossover Operator	Cost value
2p12p6s	150ch,200gen	Scattered	11,652
2p12p12s	150ch,200gen	Stas1117	13,093
2p12p24s	150ch,200gen	Stas1117	26,194
3p12p6s	500ch,60gen	Stas0055	16,368
3p12p12s	150ch,200gen	Scattered	27,606
3p12p24s	150ch,200gen	Stas0055	29,347
4p12p6s	1000ch, 30gen	Stas1117	14,407
4p12p12s	150ch, 200gen	Stas0055	21,852
4p12p24s	150ch, 200gen	Stas	28,901
5p12p6s	1000ch, 30gen	Stas0055	17,960
5p12p12s	150ch, 200gen	Stas1117	27,983
5p12p24s	600ch, 50gen	Stas1117	35,940

5. Conclusion

In this study, the new crossover operator is called Stas Crossover. It is tested by multi-product and multi-period APP problems with an interactive decision variable. Stas crossover operator performs better than Single point crossover, Two points crossover, and Arithmetic crossover, but in APP problems, Scattered crossover is shown to have the best performance. Then adjust the size of the area probability Stas crossovers, on the other hand, include Stas0055 crossover and Stas1117 crossover. These options could be able to show better performance than every other operator. That means in the beginning it was assumed that a reduction in the percentage of Single point crossover and Two points crossover resulted in an improvement in the answer. The most important advantage of Stas crossover is that it allows more diversity to select the way to create offspring and increase opportunity for offspring that can obtain good genetic information directly.

References

Ahmed, S. A., Biswas, T. K., & Nundy, C. K. (2019). An Optimization Model for Aggregate Production Planning and Control: A Genetic Algorithm Approach. *International Journal of Research in Industrial Engineering, 8*(3), 203-224.

Alam, T., Qamar, S., Dixit, A., & Benaida, M. (2020). Genetic algorithm: Reviews, implementations, and applications. *arXiv preprint arXiv:2007.12673*.

Burke, E. K., & Kendall, G. (2005). *Search methodologies Introductory Tutorials in Optimization and Decision Support Techniques*. Springer Science-i-Business Media, LLC.

Chakrabortty, R.K., & Hasin, Md.A. (2013). Solving an aggregate production planning problem by using multi-objective genetic algorithm (MOGA) approach. *International Journal of Industrial Engineering Computations, 4*(1), 1-12.

Cheraghalikhania, A., Khoshalhana, F., & Mokhtari, H. (2019). Aggregate production planning: A literature review and future research directions. *International Journal of Industrial Engineering Computations, 10*(2), 309-330.

Dakka, F., Aswin, M., & Siswojo, B. (2017). Multi-Plant Multi-Product Aggregate Production Planning Using Genetic Algorithm. *International Journal of Engineering Research And Management, 4(7), 48-52.*

Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization & Machine Learning. Pearson Education Pvt. Ltd., Singapore.

Ioannis, G.T. (2009). Solving constrained optimization problems using a novel genetic algorithm. *Applied Mathematics and Computation, 208*, 273-283.

Katoch, S., Chauhan, S. S., & Kumar, V. (2020). A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications, 80*(5), 8091–8126.

Krajewski, L. J., & Ritzman, L. P. (1999). *Operations Management: Strategy and Analysis*. Wesley Publishing Company, United States of America.

Kramer, O. (2017). *Genetic Algorithm Essentials*. Springer International Publishing AG.

Lambora, A., Gupta, K., & Chopra, K. (2019). Genetic Algorithm- A Literature Review. *In International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (Com-IT-Con) (pp. 380-384).*

Mahmud, S., Hossain, M. S., & Hossain, M. M. (2018). Application of multi-objective genetic algorithm to aggregate production planning in a possibilistic environment. *International Journal of Industrial and Systems Engineering, 30*(1), 40-59.

Mathew, T. V. (2012). *Genetic algorithm.* Report submitted at IIT Bombay.

Mirjalili, S. (2019). *Evolutionary Algorithms and Neural Networks*. Springer International Publishing AG.

Mitchell, M. (1996). *An Introduction to Genetic Algorithms*. Massachusetts Institute of Technology. England.

Moghaddam, R.T., & Safaei, N. (2006). Solving a generalize aggregate production planning problem by genetic algorithm. Journal of Industrial Engineering International, Islamic Azad University, South Tehran Branch, 2(1), 53-64.

Pradenas, L., Penailillo, F., & Ferland, J. (2004). Aggregate production planning problem. A new algorithm. *Electronic Notes in Discrete Mathematics, 18*, 193–199.

Savsani, P., Banthia, G., Gupta, J., & Vyas, R. (2016). Optimal Aggregate Production Planning by using Genetic Algorithm. *In Proceedings of the 2016 Conference on Industrial Engineering and Operations Management Kuala Lumpur, Malaysia (pp. 863-874)*.

Shukla, A., Tiwari, R., & Kala, R. (2010). *Towards Hybrid and Adaptive Computing. Studies in Computational Intelligence, 307*. Springer-Verlag Berlin Heidelberg.

Sivanandam, S. N. & Deepa, S. N. (2008). *Introduction to Genetic Algorithms*. Springer-Verlag Berlin Heidelberg.

Wang, S. C., & Yeh, M. F. (2014). A modified particle swarm optimization for aggregate production planning. *Expert Systems with Applications, 41*(6), 3069-3077.