International Journal of Sciences: Basic and Applied Research (IJSBAR)

ISSN 2307-4531
(Print & Online)

http://gssrr.org/index.php?journal=JournalOfBasicAndApplied

D-Optimal Design for Mixture Process Variable with Split-Plot Approach Using a Genetic Algorithm

Dea Handayani Juniar[a], Aji Hamim Wigena[b], Utami Dyah Syafitri[c*]

[a,b,c]*Department of Statistics, IPB University, Bogor, 16680, Indonesia*

[a]*Email: deajuniar@apps.ipb.ac.id,* [b]*Email: aji_hw@apps.ipb.ac.id,* [c]*Email: utamids@apps.ipb.ac.id*

Abstract

Mixture-process variables (MPV) is an experiment that responds not only to the proportions of the components but also to the conditions of the process. The impact of MPV is a large number of experimental runs. A large number of experimental runs has a consequence on cost, time, and resource constraints. However, choosing an optimal design with limited runs will ensure efficiency in such cases. In practice, the composition of the mixture experiment will run for each level of the process variables. Therefore, it causes a limitation in randomization. A split-plot approach can be an option to solve the problem, where the whole-plot is the process variables, and the sub-plot is the mixture component. In this research, the authors developed a genetic algorithm (GA) to find the optimal design. The genetic algorithm maintains a population of candidate solutions to a problem. It then selects the candidate point that has the most suitable criteria for solving the problem. The selection criterion used is the D-optimality criterion which is focused on parameter optimization. The case study is an experiment consisting of three ingredients and a process variable with three levels. The result concluded that GA provided an excellent design compared to the coordinate exchange algorithm with the value of D-efficiency for $\eta = 1$ is 1.195, $\eta = 5$ is 1.082, and $\eta = 10$ is 1.078.

Keywords: D-Optimality; Genetic Algorithm; Mixture Experiment; Process Variable; Split-plot.

1. Introduction

A mixture experiment is an experiment that the factors are the components of a mixture. The response is assumed to depend only on the relative proportions of each component and not the total amount of the mixture [1]. The unique feature in the mixture experiment is the mixture constraint.

---

* Corresponding author.

The proportion of each component must be between zero and one and sum to unity. If $x_1, x_2, \dots, x_q$ are the proportions of q components in a mixture experiment, then

$$\sum_{i=1}^{q} x_i = x_1 + x_2 + x_3 + \cdots + x_q = 1 \quad (1.1)$$

The main effect of the equality constraints in equation (1.1) is that the component proportions in the mixture experiment cannot be made orthogonal and cannot vary independently. Besides the primary constraint, it is common in mixture experiments to have limitations on some or all of the component proportions are imposed in the form of lower and upper bounds as follows:

$$0 \le L_i \le x_i \le U_i \le 1; \quad i = 1,2, \dots, q \quad (1.2)$$

Where L_i and U_i are the lower and upper bounds for an *i*th component. The experimental region will become an irregularly-shaped polyhedron contained within a simplex because of these additional constraints.

In some cases, the response depends not only on the relative proportions of each component but also on process variables. These process variables are not part of the mixture, but if the conditions or levels are changed, these changes will affect the response of interest. This combination of mixture experiment and process variables is called Mixture Process Variable (MPV). The level of process variables may be hard to change from one level to another. Therefore, it causes a restriction in randomization. This case is called a split-plot experiment [2]. The split-plot experiment consists of whole plots and sub-plots. The whole plot is a factor with hard to change, while the subplot is a factor with easy to change from run to run. In this case, the whole plot is the process variables, and the subplot is the mixture component. These conditions lead to two randomizations in the split-plot experiment. The first randomization is how to allocate the level of the process variable on the whole plot. Meanwhile, the second randomization puts the mixture compositions as the subplots on a specific whole plot.

The impact of mixture-process variables is a large number of experimental runs because the relative proportions of each component must be run in each level of process variables. Therefore, an optimal design is needed to solve this problem. Optimal design is part of the experimental design that estimates parameters without bias and minimum variance. Practically, the optimal design can reduce the cost and time in the experiment. The optimal design is searching for a good design based on a specific criterion. The standard criterion used is the D-optimality criterion based on parameter prediction. Then, the D-optimal design minimizes the variance value of the estimated model parameters.

Finding an excellent optimal design concerning some optimality criteria can be difficult. An alternative algorithm, such as a genetic algorithm (GA), was proposed. Borkowski [3] used a genetic algorithm to determine the optimal design point in the second-order response surface model. The GA strategies used are blending, creep, sign change, zero genes, and extreme genes. Then, Limmun and his colleagues [4] used a genetic algorithm for optimal design in a mixture experiment with D-optimal criteria. The GA strategies used are blending, crossover, and mutation. Pradubsri and his colleagues [5] used a genetic algorithm for optimal design on mixture-process variables with D-optimality criterion. Pradubsri and his colleagues [5] used Limmun

and his colleagues [4] strategy for mixture components and Borkowski's [3] strategy for process variables. Therefore, in this research, we extend the work of Pradubsri and his colleagues to develop GA in R software for generating designs in mixture-process variables using a split-plot approach. The case study is an experiment consisting of three ingredients and a process variable with three levels. The research has two main aims:

- To build a genetic algorithm in R *software* and obtain a *D*-optimal design for a mixture-process variable (MPV) with a split-plot approach.
- To compare the results of the genetic Algorithm in R software with the coordinate-exchange algorithm in the JMP Software using *D*-efficiency.

2. Literature Review

2.1 Model Mixture Process Variables

The model for modeling the response *Y* of a mixture experiment is the Scheffé-type polynomial model [2]. The first-order Scheffé model is given by

$$Y = \sum_{i=1}^{q} \beta_i x_i + \varepsilon \qquad (2.1)$$

Whereas the second-order Scheffé model is given by

$$Y = \sum_{i=1}^{q} \beta_i x_i + \sum_{i=1}^{q-1} \sum_{j=i+1}^{q} \beta_{ij} x_i x_j + \varepsilon \qquad (2.2)$$

And the special-cubic model can be writeten as

$$Y = \sum_{i=1}^{q} \beta_i x_i + \sum_{i=1}^{q-1} \sum_{j=i+1}^{q} \beta_{ij} x_i x_j + \sum_{i=1}^{q-2} \sum_{j=i+1}^{q-1} \sum_{k=j+1}^{q} \beta_{ijk} x_i x_j x_k + \varepsilon \qquad (2.3)$$

If only two levels of each in the process variables, their main effects, and interaction effects are expressible using the standard polynomial form. With *n* process variables whose levels are defined using coded variables $z_1, z_2, \ldots, z_n$ [6] as follows :

Main effects :

$$Y = \alpha_0 + \alpha_1 z_1 + \cdots + \alpha_n z_n = \alpha_0 + \sum_{l=1}^{n} \alpha_l z_l \qquad (2.4)$$

Main effects plus 2-factor interaction effects:

$$Y = \alpha_0 + \sum_{l=1}^{n} \alpha_l z_l + \sum \sum_{l<m}^{n} \alpha_{lm} z_l z_m \qquad (2.5)$$

2.2 Split Plot and D-optimal Design

The split-plot experiment has limitations in randomization. Therefore, model estimation for a split-plot experiment is more complex than a completely randomized experiment. It is due to the presence of two sources

of error from hard-to-change factor (whole plots) and easy-to-change factor (subplots). The matrix notation of model estimation for a split-plot experiment with sample size *n* and *b* whole plot by [7] is

$$\boldsymbol{Y} = \boldsymbol{X\beta} + \boldsymbol{Z\gamma} + \boldsymbol{\varepsilon} \tag{2.6}$$

where **X** represents ($n \times p$) model matrix containing the settings of both the whole plot factors (**w**) and the subplot variables (**s**). The matrix **Z** represents the $n \times b$ matrix of zeroes, and the ones assigning then runs to the *b* whole plot, $\boldsymbol{\gamma}$ is the random effects of the *b* whole plot, and $\boldsymbol{\varepsilon}$ is the *n*-dimensional vector containing the random errors. The model contains two random components that $\boldsymbol{\gamma}$ is random effect for the main plot, and $\boldsymbol{\varepsilon}$ is the random error for the subplots with the following assumptions:

$$\gamma \sim N(\mathbf{0}_b, \sigma^2 \boldsymbol{I}_b) \tag{2.7}$$

$$\varepsilon \sim N(\mathbf{0}_n, \sigma^2 \boldsymbol{I}_n) \tag{2.6}$$

and

$$cov(\gamma, \varepsilon) = \mathbf{0}_{bxn} \tag{2.8}$$

Under these assumptions, the covariance matrix of the response, *cov*(**Y**), can be written as

$$\boldsymbol{V} = \sigma_\varepsilon^2 \boldsymbol{I_n} + \sigma_\gamma^2 \boldsymbol{ZZ^t} = \sigma_\varepsilon^2 (\boldsymbol{I_n} + \eta \boldsymbol{ZZ^t}) \tag{2.9}$$

where $\eta = \sigma_\gamma^2 / \sigma_\varepsilon^2$ is a statistic to measure whether observations within the same whole plot are correlated. When the same mixture blend is not run at each combination of process variables, to estimate the parameters, the generalized least squares use. The parameter estimating equation is

$$\widehat{\boldsymbol{\beta}} = (\boldsymbol{X^t V^{-1} X})^{-1} \boldsymbol{X^t V^{-1} y} \tag{2.10}$$

With covariance matrix

$$Var(\widehat{\boldsymbol{\beta}}) = (\boldsymbol{X^t V^{-1} X})^{-1} = \boldsymbol{\sigma_\varepsilon^2} \{\boldsymbol{X^t} (\boldsymbol{I_n} + \eta \boldsymbol{ZZ^t})^{-1} \boldsymbol{X}\}^{-1} \tag{2.11}$$

Furthermore, the information matrix is written as

$$\boldsymbol{M} = \boldsymbol{X^t V^{-1} X} = \boldsymbol{\sigma_\varepsilon^{-2}} (\boldsymbol{X^t} (\boldsymbol{I_n} + \eta \boldsymbol{ZZ^t})^{-1} \boldsymbol{X}) \tag{2.12}$$

The split-plot structure can be applied to various criteria such as D, A, and I optimality [2]. The D-Optimality criterion is a generally used criterion to find the optimal design. The D-optimality is a criterion that prioritizes the quality of the parameter estimate that can be shown by the value of Var (β). The D-Optimal expects to get the minimum value of Var (β) by maximizing the determinant of the information matrix or minimizing the inverse determinant of the information matrix [8]. The D-optimality has been applied for constructing split-plot design by Goos and Vandebroek [9,10]. To compare the quality of two design information matrices, we

commonly use D-efficiency [11]. Suppose $\mathbf{M_1}$ denotes the first design information matrix and $\mathbf{M_2}$ denotes the second design information matrix. D-efficiency is defined as

$$D_{eff} = \left(\frac{|\boldsymbol{M_1}|}{|\boldsymbol{M_2}|}\right)^{\frac{1}{p}} \tag{2.13}$$

If the D_{eff} is more than 1, then the design $\mathbf{M_1}$ is more efficient than design $\mathbf{M_2}$, where p is a number of parameters.

2.3 Genetic Algorithm

The Genetic Algorithm (GA) is an optimization and search method that mimics the metaphor of natural biological evolution. John Holland developed the genetic algorithm in the 1960s and detailed it in Holland [12]. In general, GA maintains a population of candidate solutions encoded either in binary or real numbers using a sampling procedure to select the solutions to optimize the objective function.

After this selection process, the most suitable candidate solutions are combined or modified by the "reproduction" operator, such as crossover and mutation, to generate new solutions for the next generation. Thus, the process continues, with the next generation developing a more proper solution until an acceptable solution has developed. In practice, Davis [13] found that the performance of the real number has out-performed with binary encoding in numerical optimization problems. The theory about the performance of GA with real numbers is developed in Golberg [14]. In addition, the use of real-valued encodings is described in detail by Michalewicz [15]. So in this research, real number encoding is used.

The genetic algorithm was applied to a wide range of scientific and industrial applications. Recently, the genetic algorithm also has been used to find optimal or nearly optimal in experimental designs. The GA can be an exciting tool in optimization. The advantages of the GA algorithm are easy to implement, reduces the possibility of reporting local optima, does not require the experimenter to provide a finite candidate set, finds good solutions in a reasonable amount of time, gives differentiability, and also convexity of the objective function [4]. For example, Borkowski [3] used a GA to generate near-optimal D, A, G, and IV exact N-point response surface designs in the hypercube and found the exact optimal designs by applying a local search algorithm to these near-optimal designs. Heredia-Langer [16] presented a technique to generate D-efficient designs using genetic algorithms (GA) and found that the GA approach eliminates the need to explicitly consider a candidate set of experimental points. In addition, it can be used in highly constrained regions while maintaining a level of performance comparable to more traditional design construction techniques. Limmun and his colleagues [4] proposed a GA to generate D-optimal designs in which the experimental region is an irregularly shaped polyhedral region. The final designs produced by GA are superior to the extreme vertice designs for all cases and can be an alternative approach when EV designs are insufficient. Pradubsri and his colleagues [5] used a GA to generate D-Optimal designs in mixture process variables and found that GA designs have superior prediction variance properties than DETMAX and k-exchange algorithm designs when the design space is the simplex or is a highly-constrained subspace of the simplex.

Although GA can be successfully applied to an optimal experimental design problem, the efficiency depends on several factors such as the choice of design region, the choice of the evaluation function, and the size of initial population designs (*M*). The initial population design (*M*) size should be large enough to allow the GA to perform an intensive search. Thus, the GA may not find the optimal global neighborhood and get trapped at a locally optimal design if the population is too small. Besides that, the selection of M will also depend on practical computing time considerations, and the population consists of an odd number (*M*) of designs for the GA used in this paper.

3. Materials and Methods

3.1. Materials

The case study consisted of three mixture components and a process variable with three levels. The proportion of the first, second, third ingredient was x_1, x_2, and x_3, respectively. In detail, Table 1 shows the constraints of each ingredient. The levels of the process variable were -1, 0, 1. Thus, code -1 represents the lowest level, 0 represents the middle level, and 1 represents the highest level.

Table 1: The constraints of components.

Components	Constraints
1st	$x_1 \geq 0.1$
2nd	$x_2 \geq 0.1$
3rd	$x_3 \geq 0.6$

Figure 1 shows the experimental region of the mixture experiment. The experimental region is under red lines. Furthermore, the design points of the mixture experiments would be run on each level of process variables.

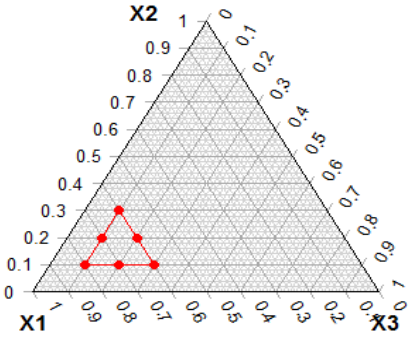

Figure 1: The experimental region of the mixture experiment.

3.2. Methods

The analytical procedure to be carried out to obtain the optimal design in this research is as follows:

Step 1. Determine the model of mixture-process variables (MPV). This research uses Scheffé's quadratic

model for mixture models and linear models for process variables.

The Scheffé's Quadratic Model :

$$Y = \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_{12} x_1 x_2 + \beta_{13} x_1 x_3 + \beta_{23} x_2 x_3 + \varepsilon \quad (3.1)$$

Linear Models for Process Variables.:

$$Y = \alpha_0 + \alpha_1 z_1 \quad (3.2)$$

The MPV model is the product of a Scheffé's quadratic mixture model crossed with a linear model in the process variables :

$$Y = \gamma_1^0 x_1 + \gamma_2^0 x_2 + \gamma_3^0 x_3 + \gamma_{12}^0 x_1 x_2 + \gamma_{13}^0 x_1 x_3 + \gamma_{23}^0 x_2 x_3 + \gamma_1^1 x_1 z + \gamma_2^1 x_2 z + \gamma_3^1 x_3 z + \gamma_{12}^1 x_1 x_2 z + \gamma_{13}^1 x_1 x_3 z + \gamma_{23}^1 x_2 x_3 z \quad (3.3)$$

where, $\gamma_i^0 = \beta_i \times \alpha_0$ dan $\gamma_i^1 = \beta_i \times \alpha_1$

Step 2. Determine the variance ratios η = 1, 5, and 10. The larger η, the more the observations within the same whole plot are correlated [7]. Furthermore, the different variance ratios lead to different designs, but these designs' quality is almost equal [2].

Table 2: The Variance of Whole plot and Subplot.

Variance	$\eta = 1$	$\eta = 5$	$\eta = 10$
Whole plot	1	5	10
Subplot	1	1	1

Step 3. Determine the number of experimental runs. This case study assumed that there are seven whole plots and three subplots, so the number of experimental runs determined is 21 design points.

Step 4. Determine *D*-optimal design by the genetic Algorithm (GA). The Genetic Algorithm was built in R software.

In Genetic Algorithm, specific terminology is usually used. A chromosome is a potential solution (design) in mixture-process variables that can be encoded as an $n \times (q + b)$ matrix, where *n* is the number of design points, q is the number of mixture components, and b is the number of process variables. Each position of a design point on a chromosome is called a gene, $g_i = [x_i, z_i]$. Thus, g_i represents the real-value encoding in the *i-th* experimental run (i =1,2,..., n) in the mixture-process variable design matrix and z_i represents the process variables. The following step of Genetic Algoritma in this research is as follows :

1. Define *N* initial design point population according to the constraints of the mixture process variable.
2. Generate an initial population of *M* chromosomes. The more the number of *M* chromosomes, the longer the

computational time in the genetic algorithm.

3. Measure the fitness value of the *M* chromosome using the D-optimal criteria.
4. Selection of the *M* chromosome using elitism. Designs that have the highest D-optimal value are called elite designs. In this research, the select design is retained and left unchanged. However, it was not allowed to follow the reproduction process until a new design chromosome with better criteria replaced it. The other design (*M*-1) chromosomes are randomly divided into (*M*-1)/2 pairs of parent chromosomes.
5. The next stage is the reproduction process. If a probability test is passed (PTIP), the reproduction operator will be applied to the parent chromosomes to create their offspring chromosomes. Statistically, let u be a random number from a uniform distribution [0,1]. Then, the reproduction process is Bernoulli distribution with probability of success is α. If $0 \leq u \leq \alpha$, the reproduction operator will be applied to the gene or set of genes. If $u > a$, then the operator is not used. The steps of reproduction for the mixture component [4] are as follows.

- Blending is the reproduction process between two parents' chromosomes. Let A_a and B_b are two parents paired in the selection process, then if a PTIP for A_a and B_b, then A_a and B_b are to be blended and create offspring chromosomes A^*dan B^* which is formed from a linear combination as follows:

$x_a^* = \beta x_a + (1-\beta)x_b$ dan $x_b^* = (1-\beta)x_a + \beta x_b$,

Where $\beta \sim U(0,1)$ is the random blending factor

- The next step is crossover. The method used is a single cut point, randomly selecting one position in the parent chromosomes and then exchanging genes. There are two types of crossover: within-parent and between-parent.

✓ Between-parent Crossover: if a PTIP, then a crossover is applied to A_a of A with random gene B_b of B.
✓ Within-parent Crossover: if a PTIP, then a crossover is applied within A_a.

- The last step of the recombination process in a mixture component is mutation. Mutations are used to change genes randomly. The method used is uniform mutation. If a PTIP is for A_a, then one of the q components of a gene, for example, r, is chosen randomly. Then, the value of x_r. According to the constraint function, Gene is replaced with a uniform random value chosen between the upper and lower limits. The other components in the x_a gene must be repaired so that the constraints of the mixture experiment are still fulfilled; that is, the sum of all components is one.

The process variables according to [3] are as follows:

- *Sign change*: if a PTIP, then process variable vector z_a is set to $-z_a$
- *Zero gene*s: if a PTIP, then process variable vector z_a is set to 0.

6. Measure the fitness value for the offspring chromosomes and compare it with the parent chromosomes' fitness value. If the value of the offspring chromosomes is higher than parent chromosomes, then the offspring

chromosomes replace the parent chromosomes in the next generation.

7. Iterate over steps 4 to 6 for a large number of generations, which shows convergence to a near-optimal design.

Step 5. Comparing the genetic algorithm results with the results of the coordinate-exchange Algorithm (JMP software) using D-efficiency.

4. Results and Discussion

Based on the constraint in the mixture experiment, the experimental region for this case can be seen in Figure 1. The experimental region is part of the entire simplex design, which forms a triangle space. The red point in Figure 1 is the main point in the simplex lattice design. Then, the design points of the mixture experiment will run in each level of the process variable. So, the total design points are 36, originating from 6 main points in each level of the process variable and two replications. Table 2 shows the variance ratio (η) influences the optimal design for mixture-process variables in a split-plot approach using the D-optimal criteria. This case study consisted of seven main plots and three subplots. In total, there were 21 design points to run. To generate the optimal design in this research using a genetic algorithm with the probability of success (α) used for each reproduction process is shown in Table 3.

Table 3: The operator reproduction of genetic algorithm.

The Reproduction Operators	Case A	Case B
Blending	0.01	0.1
Crossover	0.2	0.2
Mutation	0.01	0.1
Sign Change	0.2	0.2
Zero Change	0.2	0.2
M	51	51
Iteration	1000	1000

There are no rules for selecting the probability of success (α). The choice of alpha (α) will not prevent convergence. However, alpha (α) choice affects the convergence speed to the optimal design [3]. Based on the variance ratio (η) in Table 2 and the reproduction operators in Table 3, Table 4 shows the final D-optimal design using a genetic algorithm for case A.

Table 4: The D-optimal Design Using Genetic Algorithm For Case A.

Whole Plot	$\eta = 1$				$\eta = 5$				$\eta = 10$			
	X1	X2	X3	Z	X1	X2	X3	Z	X1	X2	X3	Z
1	0.1	0.1	0.8	1	0.1	0.1	0.8	-1	0.1	0.3	0.6	-1
1	0.1	0.3	0.6	1	0.2	0.1	0.7	-1	0.3	0.1	0.6	-1
1	0.3	0.1	0.6	1	0.1	0.2	0.7	-1	0.1	0.2	0.7	-1
2	0.2	0.2	0.6	1	0.1	0.3	0.6	-1	0.1	0.1	0.8	-1
2	0.1	0.2	0.7	1	0.3	0.1	0.6	-1	0.2	0.1	0.7	-1
2	0.2	0.1	0.7	1	0.2	0.2	0.6	-1	0.1	0.3	0.6	-1
3	0.2	0.2	0.6	-1	0.1	0.3	0.6	1	0.1	0.2	0.7	1
3	0.1	0.3	0.6	-1	0.2	0.2	0.6	1	0.1	0.3	0.6	1
3	0.3	0.1	0.6	-1	0.3	0.1	0.6	1	0.3	0.1	0.6	1
4	0.1	0.1	0.8	0	0.1	0.1	0.8	0	0.2	0.1	0.7	1
4	0.3	0.1	0.6	0	0.2	0.2	0.6	0	0.1	0.3	0.6	1
4	0.2	0.1	0.7	0	0.1	0.2	0.7	0	0.1	0.1	0.8	1
5	0.2	0.2	0.6	-1	0.1	0.2	0.7	1	0.1	0.2	0.7	-1
5	0.1	0.3	0.6	-1	0.1	0.1	0.8	1	0.2	0.1	0.7	-1
5	0.1	0.2	0.7	-1	0.289	0.111	0.6	1	0.2	0.2	0.6	-1
6	0.1	0.1	0.8	-1	0.2	0.1	0.7	1	0.3	0.1	0.6	0
6	0.1	0.2	0.7	-1	0.1	0.3	0.6	1	0.2	0.2	0.6	0
6	0.2	0.1	0.7	-1	0.1	0.2	0.7	1	0.1	0.2	0.7	0
7	0.2	0.2	0.6	1	0.1	0.3	0.6	-1	0.2	0.2	0.6	1
7	0.1	0.3	0.6	1	0.3	0.1	0.6	-1	0.1	0.3	0.6	1
7	0.3	0.1	0.6	1	0.2	0.1	0.7	-1	0.1	0.1	0.8	1
D-Opt	1.70E-26				5.15E-28				1.09E-28			

The design points in Table 4 will be more representative to show in simplex space, as shown in Figure 2.

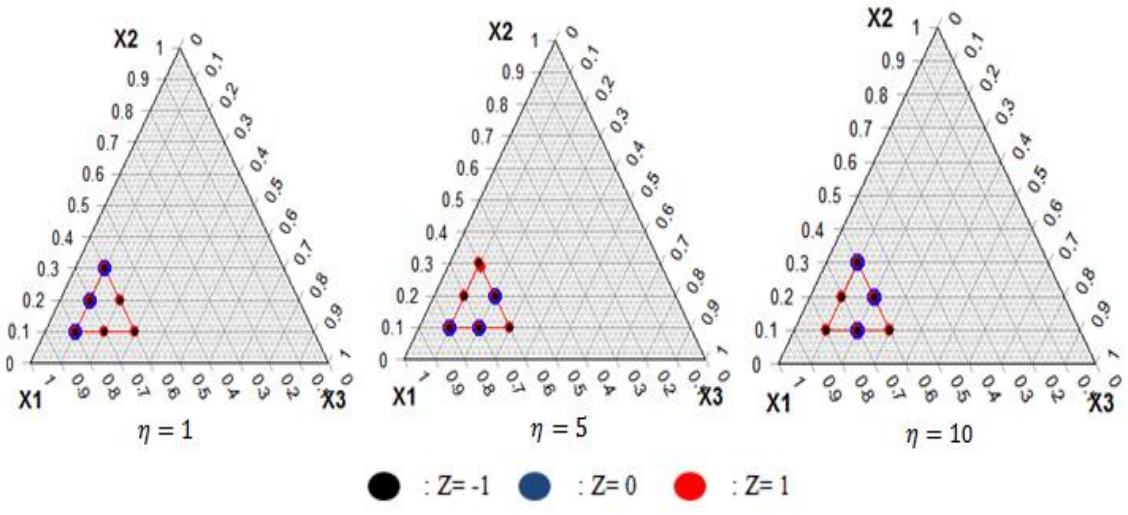

Figure 2: The Experimental Region of The MPV Using GA for Case A.

In Figure 2, in each variance ratio for process variables with z = 1 (low level), the optimal design points are spread over all the main points in the simplex lattice design and consist of 3 main plots so that there are nine design points. The same thing happened for the design z= -1 (high level), while at z = 0 (middle level), it only occurs one time in the main plot so that there is three design point. In addition, Table 4 shows that that the D-optimal value will decrease if the variance ratio (η) of the main plots and sub-plots increases. Then, Table 5 shows the D-optimality design of case B with 1000 iteration.

Table 5: D-optimal Design Using Genetic Algorithm For Case B.

Petak Utama	$\eta = 1$				$\eta = 5$				$\eta = 10$			
	X1	X2	X3	Z	X1	X2	X3	Z	X1	X2	X3	Z
1	0.1	0.1	0.8	-1	0.1	0.27	0.63	-1	0.1	0.1	0.8	-1
1	0.2	0.1	0.7	-1	0.1	0.1	0.8	-1	0.2	0.1	0.7	-1
1	0.3	0.1	0.6	-1	0.3	0.1	0.6	-1	0.3	0.1	0.6	-1
2	0.2	0.2	0.6	-1	0.194	0.104	0.702	-1	0.165	0.212	0.624	-1
2	0.1	0.3	0.6	-1	0.146	0.207	0.646	-1	0.1	0.3	0.6	-1
2	0.1	0.2	0.7	-1	0.3	0.1	0.6	-1	0.1	0.2	0.7	-1
3	0.1	0.1	0.8	-1	0.2	0.2	0.6	-1	0.2	0.2	0.6	-1
3	0.1	0.3	0.6	-1	0.1	0.3	0.6	-1	0.3	0.1	0.6	-1
3	0.3	0.1	0.6	-1	0.1	0.2	0.7	-1	0.1	0.2	0.7	-1
4	0.2	0.1	0.7	0	0.289	0.1	0.611	0	0.1	0.2	0.7	0
4	0.1	0.3	0.6	0	0.2	0.1	0.7	0	0.2	0.1	0.7	0
4	0.1	0.2	0.7	0	0.115	0.285	0.6	0	0.1	0.2	0.7	0
5	0.2	0.2	0.6	1	0.2	0.2	0.6	1	0.3	0.1	0.6	1
5	0.2	0.1	0.7	1	0.1	0.3	0.6	1	0.1	0.3	0.6	1
5	0.3	0.1	0.6	1	0.3	0.1	0.6	1	0.1	0.2	0.7	1
6	0.1	0.1	0.8	1	0.1	0.1	0.8	1	0.1	0.247	0.653	1
6	0.1	0.3	0.6	1	0.2	0.1	0.7	1	0.2	0.1	0.7	1
6	0.1	0.2	0.7	1	0.1	0.2	0.7	1	0.2	0.2	0.6	1
7	0.1	0.1	0.8	1	0.2	0.2	0.6	1	0.1	0.1	0.8	1
7	0.2	0.1	0.7	1	0.155	0.245	0.6	1	0.123	0.254	0.623	1
7	0.3	0.1	0.6	1	0.1	0.2	0.7	1	0.1	0.2	0.7	1
D-Opt	1.53E-26				1.75E-28				2.97 E-29			

Based on Table 5, each variance ratio for process variables with z = 1 (low level) consists of 3 main plots so that there are nine design points. The same thing happened for the design z= -1 (high level), while at z = 0 (middle level), it only occurs one time in the main plot so that there is three design point. In conclusion, the D-optimality criterion decreased if the variance ratio (η) of the main plots and sub-plots increased. The optimal design points in Table 5 will be more representative if shown in a simplex space in Figure 3.

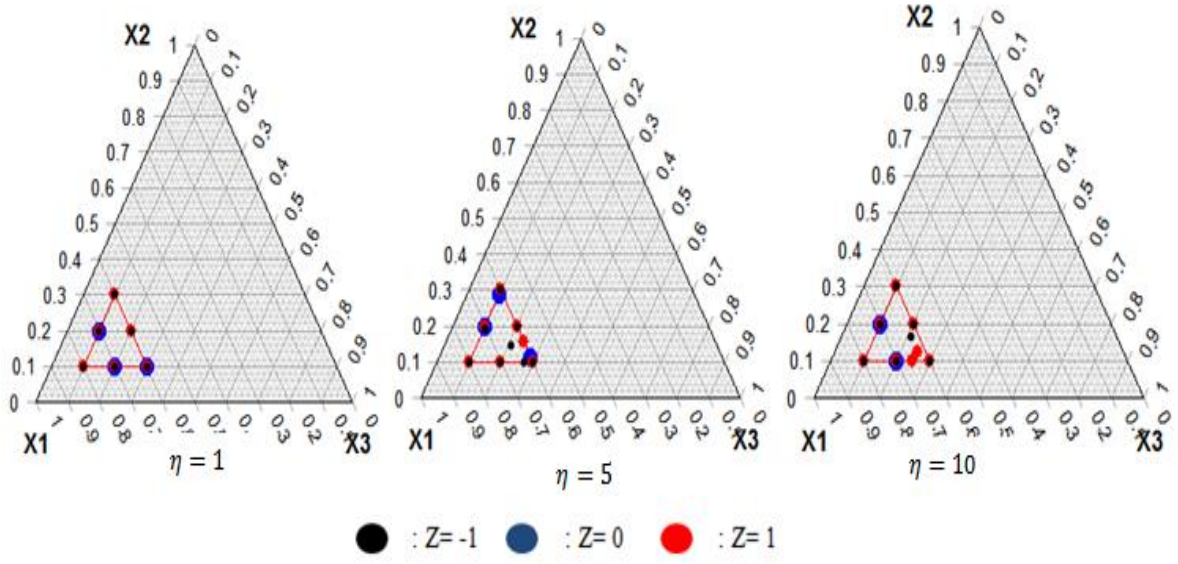

Figure 3: The Experimental Region of The MPV Using GA For Case B.

For $\eta = 1$, the points spread over the main points in the simplex lattice design. However, for $\eta = 5$ and $\eta = 10$, several points are shifted from the main points in the simplex lattice design. In conclusion, that the design has not been reached the optimal point yet. Hence, more iterations were needed. Those indicate a relationship between the probability of success (α), especially in operator blending and operator mutation, with the number of iterations to reach the optimal design point. It is remarkable from Table 4 and Table 5 that the D-optimality criterion decreased if the value of the variance ratio (η) of the whole plot and sub-plot increased. That condition is related to the split-plot design; the D-optimal value will generally depend on the V matrix's variance ratio (η) assumption. Based on the D-optimal value, the design in Table 4 has a higher D-optimal value in each variance ratio (η) than the D-optimal value in Table 5. Therefore, the optimal design in Table 4 (case A) is better than the optimal design in Table 5 (case B). In cases A and B, the proportions at each level in the process variables and the variance ratio (η) have the same proportions; the design points tend to choose points from the extreme level (±1). The reason is that the process variable model is the first-order model, and the number of process variable's levels is three. As an optimal design theory, the design points are in the corner of the experimental region for the first-order model. If the process variable model's assumption is linear, then theoretically, the number of levels in the process variable used is two (2). For simulation, the GA algorithm was run for the D-optimal MPV with two levels in the process variable. Table 6 shows the optimal design resulted from the GA algorithm 1000 iterations.

Table 6: D-optimal Design with 2 Level Process Variable Using GA for Case A.

Whole Plot	$\eta = 1$				$\eta = 5$				$\eta = 10$			
	X1	X2	X3	Z	X1	X2	X3	Z	X1	X2	X3	Z
1	0.2	0.2	0.6	1	0.1	0.2	0.7	1	0.1	0.1	0.8	-1
1	0.1	0.2	0.7	1	0.2	0.2	0.6	1	0.2	0.2	0.6	-1
1	0.3	0.1	0.6	1	0.3	0.1	0.6	1	0.3	0.1	0.6	-1
2	0.1	0.3	0.6	1	0.1	0.1	0.8	1	0.3	0.1	0.6	-1
2	0.2	0.2	0.6	1	0.3	0.1	0.6	1	0.1	0.2	0.7	-1
2	0.2	0.1	0.7	1	0.2	0.1	0.7	1	0.2	0.1	0.7	-1
3	0.3	0.1	0.6	1	0.2	0.2	0.6	-1	0.2	0.1	0.7	1
3	0.1	0.1	0.8	1	0.1	0.2	0.7	-1	0.1	0.2	0.7	1
3	0.2	0.1	0.7	1	0.3	0.1	0.6	-1	0.1	0.3	0.6	1
4	0.1	0.2	0.7	-1	0.1	0.3	0.6	1	0.3	0.1	0.6	1
4	0.1	0.1	0.8	-1	0.2	0.2	0.6	1	0.1	0.3	0.6	1
4	0.2	0.1	0.7	-1	0.2	0.1	0.7	1	0.1	0.1	0.8	1
5	0.2	0.2	0.6	-1	0.1	0.3	0.6	1	0.1	0.3	0.6	-1
5	0.1	0.3	0.6	-1	0.1	0.2	0.7	1	0.2	0.1	0.7	-1
5	0.2	0.1	0.7	-1	0.1	0.1	0.8	1	0.1	0.1	0.8	-1
6	0.1	0.3	0.6	1	0.1	0.2	0.7	-1	0.2	0.2	0.6	1
6	0.1	0.2	0.7	1	0.2	0.1	0.7	-1	0.3	0.1	0.6	1
6	0.1	0.1	0.8	1	0.1	0.3	0.6	-1	0.1	0.2	0.7	1
7	0.2	0.2	0.6	-1	0.1	0.1	0.8	-1	0.2	0.2	0.6	-1
7	0.1	0.2	0.7	-1	0.2	0.2	0.6	-1	0.1	0.2	0.7	-1
7	0.3	0.1	0.6	-1	0.2	0.1	0.7	-1	0.1	0.3	0.6	-1
D-Opt	5.530E-26				1.923E-27				4.626E-28			

The design points in Table 6 will be more representative to show in simplex space, as shown in Figure 4.

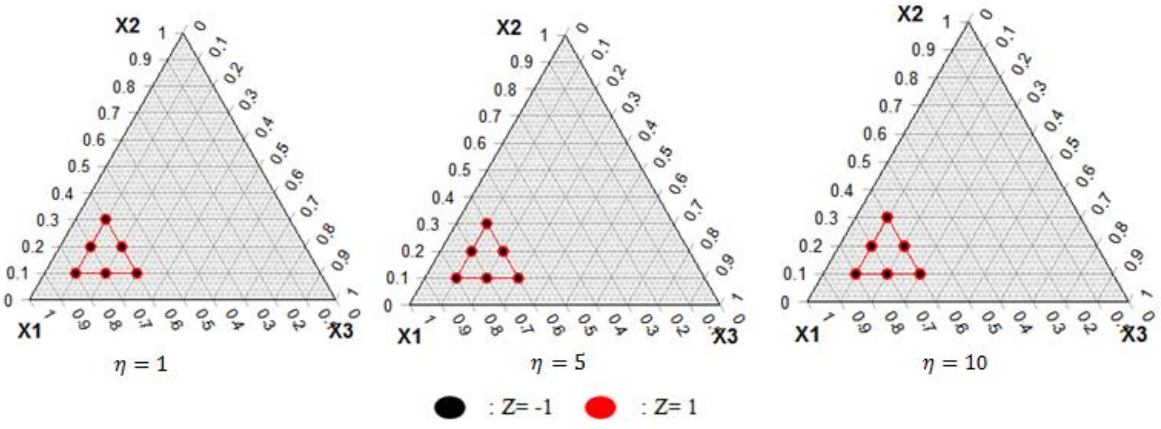

Figure 4: The Experimental Region of The MPV with 2 Level Process Variable for Case A.

Based on Table 6 and Figure 4, the design points spread to the main points in the simplex lattice design. For

$\eta = 1$ and $\eta = 5$, the process variable z = -1 is located in 3 whole plots so that it consists of 9 design points, and the process variable z = 1 is located in 4 whole plots so that it consists of 12 design points. Then for $\eta = 10$, the process variable z = -1 is located in 4 whole plots so that it consists of 12 design points, and the process variable z = 1 is in 3 whole plots so that it consists of 9 design points. The D-optimality criterion for the D-optimal design MPV with two levels on the process variable was more prominent than three levels in all the variance ratios (η).

Table 7: D-optimal of MPV with 3 Levels of Process Variables and 2 Levels of Process Variables.

Process Variable	$\eta = 1$	$\eta = 5$	$\eta = 10$
3 Level	1.70E-26	5.15E-28	1.09E-28
2 Level	5.53E-26	1.92E-27	4.63E-28

To evaluate the performance of the genetic algorithm, the D-optimal MPV designs of the Genetic Algorithm (Case A) compared to the D-optimal designs of the coordinate exchange (CE) algorithm. Figure 5 shows the visualization of the optimal design based on the coordinate exchange algorithm in the Trial JMP Software.

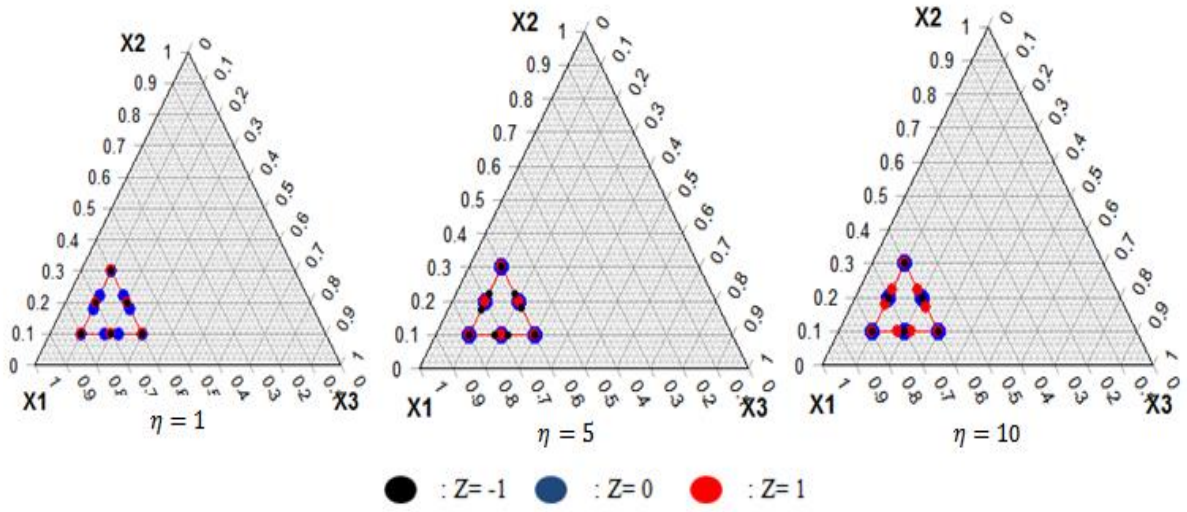

Figure 5: The Experimental Region of The MPV Using CE.

Based on Figure 5, the design points in the coordinate exchange algorithm do not tend to come from the extreme level (±1) compared to the genetic algorithm. However, with the same number of iterations, design points in the coordinate exchange algorithm are still slightly shifted from the main points in the simplex lattice design. Table 8 shows the comparison between the GA and CE algorithms based on the D-optimality criteria. The D-efficiency of the D-optimal MPV designs of the GA algorithm out-performed the design of the CE algorithm.

Table 8: D-Efficiency of Coordinate Exchange and Genetic Algorithm.

Algorithm	$\eta = 1$	$\eta = 5$	$\eta = 10$
GA	1.70E-26	5.15E-28	1.09E-28
CE	2.00E-27	1.99E-28	4.43 E-29
D_{eff}	1.195	1.082	1.078

6. Conclusions

In conclusion, the genetic algorithm thrived on finding the D-optimal MPV design with a split-plot approach. The strategies used in the genetic algorithm were blending, crossover, mutation, sign change, and zero genes. Blending and mutation operators used less probability of success (α) than other operators to obtain more stable D-optimal MPV designs. It is also remarkable that the genetic algorithm provided optimal design as good as the coordinate-exchange algorithm, which is already implemented in commercial software.

Acknowledgments

The authors thank the Higher Education of Ministry of Education, Culture, Research, and Technology for the funding. The authors also thank Irene Triyani and Cindy Firiera Darwis from PT Nutrifood Indonesia as collaboration partners in this research.

References

[1] Cornell, J. (2002). *Experiments with Mixtures: Designs, Models, and the Analysis of Mixture Data* (3rd ed.). Wiley.

[2] Goos, P., Jones, B. 2011. *Optimal Design of Experiment: A Case Study Approach*. Wiley.

[3] John J. Borkowski. 2003. Using a Genetic Algorithm to Generate Small Exact Response Surface Designs. Journal of Probability and Statistical Science.1(1), 65-88.

[4] Limmun, W., Borkowski, J. J., & Chomtee, B. (2012). Using a Genetic Algorithm to Generate D-optimal Designs for Mixture Experiments. *Quality and Reliability Engineering International*, *29*(7), 1055–1068. https://doi.org/10.1002/qre.1457

[5] Pradubsri, W., Chomtee, B., & Borkowski, J. J. (2019). Using a genetic algorithm to generate D-optimal designs for mixture-process variable experiments. *Quality and Reliability Engineering International*, *35*(8), 2657–2676. https://doi.org/10.1002/qre.2549

[6] Cornell, J. A. (1988). Analyzing Data from Mixture Experiments Containing Process Variables: A Split-Plot Approach. *Journal of Quality Technology*, *20*(1), 2–23. https://doi.org/10.1080/00224065.1988.11979079

[7] Goos, P., & Donev, A. N. (2007). Tailor-Made Split-Plot Designs for Mixture and Process Variables. *Journal of Quality Technology*, *39*(4), 326–339. https://doi.org/10.1080/00224065.2007.11917699

[8] Kiefer, J., & Wolfowitz, J. (1960). The Equivalence of Two Extremum Problems. *Canadian Journal of Mathematics*, *12*, 363–366. https://doi.org/10.4153/cjm-1960-030-4

[9] Goos, P., & Vandebroek, M. (2001). Optimal Split-Plot Designs. *Journal of Quality Technology*, *33*(4), 436–450. https://doi.org/10.1080/00224065.2001.11980103

[10] Goos, P., & Vanderbroek, M. (2003). D-Optimal Split-Plot Designs With Given Numbers and Sizes of Whole Plots. *Technometrics*, *45*(3), 235–245. https://doi.org/10.1198/004017003000000050

[11] Atkinson, Donev, Tobias. 2007. Optimum Experimental Designs with SAS. Oxford: Oxford University Press.

[12] Holland, J. H.(1975). *Adaption in Natural and Artificial Systems,* Universitas of Michigan Press, Ann Arbor, MI.

[13] Davis, L. (1991). *Handbook of Genetic Algorithms, Van Nostrand Reinhold,* New York.

[14] Golberg, D. E.(1991). Real-coded Genetic Algorithms, Virtual Alphabets, and Blocking, Complex System, 5, 139-168.

[15] Michalewicz, Z. (1992). Genetic algorithm + Data Structures = Evolution Programs, Springer-Verlag, New York.

[16] Heredia-Langner, A., Carlyle, W. M., Montgomery, D. C., Borror, C. M., & Runger, G. C. (2003). Genetic Algorithms for the Construction of D-optimal Designs. *Journal of Quality Technology*, *35*(1), 28–46. https://doi.org/10.1080/00224065.2003.11980189