Proceedings of the 2nd International Conference on Statistical Theory, Methods and Applications, 25th – 26th April, 2024

Selection of N-Point *D*- and *A*-Optimal Exact Designs Using Genetic Algorithm

Polycarp Emeka Chigbu[1] and Chukwuemeka Anthony Onah[2]

[1,2]Department of Statistics, University of Nigeria, Nsukka, Enugu State, Nigeria
Corresponding author's email: chukwuemeka.onah.pg90926@unn.edu.ng

Abstract

In the search for optimal designs in industrial experiments, situations like non-linearity and non-differentiability often arise that defy classical optimisation algorithms. In this study, Genetic Algorithm (GA) was used to select *n*-point *D*- and *A*- optimal exact designs in regular and irregular experimental regions for standard regression models. Before the application of GA in the selection of the optimal designs, requirements such as design region, optimality criterion, design size, population size, crossover and mutation probabilities were specified. Based on these specifications GA selected *n*-point *D*- and *A*-optimal exact designs by iteratively applying three operators; selection, crossover and mutation probabilities, on an initial population of designs generated from a specified design region. Metrics such as relative efficiency of the *D*- and *A*-optimality criteria were used to determine the performance of GA compared to those obtained from existing algorithms. Results obtained show that GA compared well and even better, with the line search algorithm, Fedorov algorithm, modified Fedorov algorithm, k-exchange algorithm and the computer algorithm reported in the literature in the search for *D*- and *A*-optimal designs. The study recommended among others, that GA should be used for the selection of optimal designs when the model comprised of more than two factors.

Keywords: Algorithms, *A*-optimal designs, *D*-optimal designs, Exact-designs, Genetic-Algorithm, Optimal-designs.

1. Introduction

In many industrial settings, experiments are conducted to characterise the relationships between design factors and the targeted response, with the goal of identifying treatments or design factor concentrations that will produce the targeted response variable's desired effect. Typically, an experiment's size grows quickly as the number of parameters and treatments increases. This, however, will lead to more costs involving time, finance and material, in running the experiment (Upperman, 1993; Collins, Dziak & Li, 2009). Therefore, cutting down on the number of treatments is necessary to save money and time when conducting experiments. Due to this, it is essential to maintain a manageable experimental size, which necessitates the choice for an optimal design (Upperman, 1993; Borkowski, 2003; Collins et al., 2009).

Choosing n points or treatments from a list of candidate-points, *N* for inclusion in a design is one of the goals of optimal design (Berger & Wong, 2009, p. 280; Nsude, 2016). The fundamental principle behind this is based on the theory of optimum design, which holds that by "optimally" choosing treatments of the input factor, statistical inference about measures of interest can be enhanced (Ugbe & Chigbu, 2012). There are two methods for this selection of *n* points design: depending on the type of probability measurements, attached to the levels or treatments of the experimental factors at the design point. It can be either accurate (discrete) or continuous (approximate) (ξ_n or ξ). The definition of an exact design is a n-point design with a probability measure, ξ_n on the region of interest, $\tilde{\chi}$, where each point of the design is assigned an equal weight, 1/n, and $n\xi_n$ is a non-negative integer (Cook & Nachtsheim, 1980). Iwundu and Chigbu (2012) indicated the region of interest of *n*-exact design n the $\tilde{\chi}$ by $\Xi_{\tilde{\chi}}^{n}$. Conversely, though, an approximate design is a probability measure, ξ, on $\tilde{\chi}$ such that $\int_{\tilde{\chi}}(dx) = 1$ and $n\xi$ not have to be a whole number (Iwundu & Chigbu, 2012).

In actuality, before such an ideal approximation design can be put into reality, all the weights at an optimal design point in an approximate design must be multiplied by n and rounded to whole numbers such that they all add to n (Kiefer, 1974; Berger & Wong, 2009, p. 30). Vandenberghe and Boyd (1999) provided details. However, this work will focus on exact designs, because they are the most practical designs that experimenters utilise. In other words, the replicated points at $\underline{x}_i$ are non-negative integers.

As earlier mentioned, the process of creating ideal precise designs necessitates choosing n points, $\underline{x}$, from a given region of interest, $\tilde{\chi}$, $\underline{x} \in \tilde{\chi}$. The method used for this choice optimizes the Fisher's information matrix, $\boldsymbol{X'X}$ (Nsude, 2016). Careful selection of $\underline{x}$ from $\tilde{\chi}$ is attainable by the use of computations to look for the *n*-optimal design points in $\tilde{\chi}$. Algorithms are crucial in the process of choosing near-optimal experimental designs because they minimise the number of runs required to estimate the required parameters, which lowers the cost of conducting experiments.

Algorithms carefully choose an optimal design from the pool of possible designs using the optimality criterion as a guide. The most well-known optimality criteria, which are all based on Fisher's information matrix, are *D*-, *A*-, *E*-, *G*-, and *I*- optimality criteria (Rady, Abd El-Monsef & Seyam, 2009; Oladugba & Madukife, 2009). One of the most widely used criteria, the *D*-optimality criterion, is to maximise the determinant of the matrix for the given problem, or reduce the determinant of the reciprocal of Fisher's information matrix. The confidence volume of each parameter in the model tends to decrease when applying the *D*-optimality criterion.

The *A*-optimality criterion pertains to the minimisation of the trace, which is the sum of the diagonal elements of the information matrix's inverse. The average variance of parameter estimates is lowered as a result of this optimality criterion. The *G*- and *I*-criteria are connected to the design's prediction variance, whilst the *E*-criterion aims to maximise the information matrix's minimum eigenvalue. While the *I*-optimality criterion lowers the average prediction variance across the design space, the *G*-optimality criterion aims to reduce the highest prediction variance overall designs, in the design space.

In practical situations, the most significant and often used design criterion is the *D*-optimality criterion (Rady et al., 2009; Ugbe & Chigbu, 2012). Scholars believe that, in certain situations, the *A*-optimality criterion does better than the *D*-optimality criterion. According to research presented by Jones, Allen-Moyer, and Goose (2020), screening designs that meet the *A*-optimality criterion are deemed more desirable than those that meet the *D*-, *I*-, and G-optimality criteria. They also concluded that, in comparison to the *D*-optimality criterion, *A*-optimal designs typically contain more independent columns in their model matrices. Because the *A*-optimal criterion is preferred for screening designs and the *D*-optimal criterion has a wider use, this work is restricted to *D*- and *A*-optimality criteria. Screening designs constitute one of the most often utilised strategies in industrial testing. Screening designs are usually utilised in the early phases of an experiment to narrow down numerous potentially significant variables and their interactions to a select few important ones. Finding an active part of the components is the goal of screening designs (Jones et al., 2020).

The *D*- and *A*-optimal designs have been identified using a variety of techniques. Nonetheless, the majority of research was concentrated on determining the *D*-optimal design in comparison to the *A*-optimal design. Among the techniques used are the exchange algorithms (Fedorov, 1972; Mitchell, 1974; Cook & Nachtsheim, 1980; Johnson & Nachtsheim, 1983). Generally, an exchange algorithm starts with an approximate initial design, and then iteratively changes the current design by exchanging current design points with new points from the region of interest. The algorithm ends when a predetermined condition is met, such as a specified number of iterations or the observation of an insignificant alteration in the objective criteria value across iterations. The exchange-type algorithms employ direct search methods, and there is always a possibility that the solution found is not globally optimal; in other words, there is no assurance that the "true" optimum design will be identified. So, it is also necessary to restart the algorithms several times with different starting points to increase the chance of finding a design that is closer to the global optimum (Montepiedra, 1998). Furthermore, the exchange-type algorithms will fail to reach global optimum when the regression model representing the problem to be optimised has three or more explanatory variables (Broudiscou et al., 1996). Other issues associated with the exchange algorithms as highlighted by Iwundu and Chigbu (2012) consist of cycling, delayed convergence, and inability to reach the intended optimum. In the same vein, Nsude (2016) noted that obtaining *D*-optimal designs presents a challenge for all exchange algorithms. It is either trapped at the local optimum or slow and computationally challenging.

In order to mitigate some of the problems associated with exchange-type algorithms in finding *D*- optimal designs, researchers have done several works in the past. For example, Iwundu and Chigbu (2012) constructed *D*-optimal *n*-point exact designs using a hill-climbing combinatorial algorithm. In finding *D*-optimal designs, Iwundu and Chigbu (2012) adopted the method of Onukogu and Iwundu (2007); it consists of organizing the $\tilde{\chi}$'s support points according to how far apart they are from the middle of $\tilde{\chi}$. Finding a collection of squared information matrices having matching diagonal elements was possible and rapid by grouping support points into H concentric balls. This made it possible to compare elements inside the group according to their absolute values that are off-diagonal (Onukogu & Iwundu, 2007). Nsude (2016) obtained D-optimal designs in continuous space by applying a line search technique; a *D*-optimal design was reached with minimal iteration. The results of Nsude (2016) compete favourably well with the works of Atkinson and Donev (1992) and Atkinson, Donev and Tobias

(2007). However, line search algorithms apply when the objective function to be optimized is differentiable (Jason, 2020). If not, it will be challenging for the algorithm to determine the global optimum for these kinds of issues (Nsude, 2016; Civicioglu, 2013).

There are non-exchange-type algorithms as well as algorithms that are not constrained by differentiability; for instance, the genetic algorithms (GAs). The GAs are gaining widespread popularity among scientists owing to their many advantages over exchange-type algorithms, as well as the line search algorithm. Optimisation problems like non-linear and non-differentiable functions, irregular design space structure, can be solved with genetic algorithms, etc; more so, due to their ability to find a near-optimal design where the conventional exchange algorithms fail to do so: see, for example, Limmun, Borkowski and Chomtee (2012), Borkowski (2003) and Nambiar (2015).

A genetic algorithm (GA) is an evolutionary search approach based on condensed biological population genetics laws and evolution ideas (Borkowski, 2003). A GA keeps track of several viable solutions and selects the most efficient ones from this population to arrive at a close-to-optimal solution. Following the selection process, reproduction operators integrate and/or modify the top candidate solutions to create new solutions for the following generation. Up until a workable solution is found, the procedure is repeated with each generation producing better ones (Borkowski, 2003). Near-optimal *D*- and *A*-exact n-point designs are chosen using a GA in this work. The chosen designs' efficiencies are computed in comparison to well-known optimal designs with identical design sizes, models, and spaces.

The aim of the study is to select *n*-point *D*- and *A*-optimal designs using genetic algorithm. Specifically, the objectives are to: select *n*-point *D*- and *A*-optimal design using genetic algorithm; and compare the efficiency of the selected designs with those selected using some of the existing algorithms.

2. Methodology

2.1 Application of GA in the Selection of *n*-point Exact Designs

A typical experimental design is defined in the experimental space $\{\tilde{\chi}, F_x, \Sigma_x\}$;
where,

$$\tilde{\chi} = \{\underline{x}\} \tag{2.1}$$

is the space of all possible trials or design region or experimental region; $\underline{x}$ is a row of *p*-factor experiment;

$$F_x = \{f(\underline{x})\} \tag{2.2}$$

is a linear space of finite dimensional continuous function defined on $\tilde{\chi}$ so that at any given design point, $\underline{x} \in \tilde{\chi}$, a random variable,

$$y(\underline{x}) = \boldsymbol{\beta} f(\underline{x}) + \varepsilon \tag{2.3}$$

is defined on $\tilde{\chi}$;
where,
$y(\underline{x})$ is the response variable observed at a designed point, $\underline{x} \in \tilde{\chi}$;
$\boldsymbol{\beta}$ is a vector of *p* unknown constant coefficients referred to as parameters;
ε is the random experimental error; and

$$\Sigma_x = \{\sigma_{\underline{x}}^2\} \tag{2.4}$$

is the space of non-negative continuous random observation error also defined on $\tilde{\chi}$.

The interest here is to find *n*-point designs such that the variance-covariance matrix is minimised or the resultant information matrix, $\boldsymbol{X'X}$, is maximised. Given the space of possible trials, $\tilde{\chi}$, the objective function or optimality criterion, $\Phi\{.\}$, design points, n, crossover probability, P_c, and mutation probability, P_m, the GA iteratively search for an *n*-point design, ξ_n^*, from $\tilde{\chi}$ which maximises the determinant of $\boldsymbol{M}(\xi_n)\ \forall\ \xi_n \in \Xi$, and also a design, ξ_n^*, that minimises the trace of $\{\boldsymbol{M^{-1}}(\xi_n)\}\ \forall\ \xi_n \in \Xi$. Thus, given the space of possible trials or experimental region, $\tilde{\chi}$, the objective function or optimality criterion, $\Phi\{.\}$, population of designs, M, design points, n, crossover probability, P_c, and mutation probability, P_m, GA generates optimal exact designs via the following steps:

Step 1. Randomly generate the initial population.
Here, M designs, $\{\xi_n^1, \ldots, \xi_n^M\}$, were randomly generated uniformly from $\tilde{\chi} = [.]$ as the initial population of designs.

Step 2. Selection probability.
For each $i = 1, 2, \ldots, M$, calculate the fitness of each design in the initial population as

$$fit_i = \frac{\Phi\{\xi_n^i\}}{\sum_{i=1}^{M}\Phi\{\xi_n^M\}} \quad (2.5)$$

where $\Phi\{\xi_n^i\}$ is the value of the objective criterion of the design, ξ_n^i, after which the corresponding cumulative probabilities, $q_m = \sum_{i=1}^{M} fit_i$, are computed and a random number from U [0, 1] is drawn. Hence, selecting designs from the initial population into the starting population is guided by q_m and the random values drawn from U [0, 1]. That is, if a particular cumulative probability, q_m, is higher than or equal to the random number, r, then that design is selected. Due to this selection process, designs with lesser fitness will not be selected; whereas, designs with better fitness can even be selected more than once.

Step 3. Crossover probability.
Designs that made it into the starting population, M', are randomly altered by reproduction. Two designs say, $\xi_n^{1\prime}$and $\xi_n^{2\prime}$, are randomly selected from the starting population of designs, M', as two parents that will mate or do crossover or exchange their genes (design point) to form two new offsprings or designs. The selection is based on a random number, r , generated from U [0, 1], and a specified crossover probability, P_c*. Assuming that r is associated with $\xi_n^{1\prime}$ given that $r \leq P_c$*, then select $\xi_n^{1\prime}$ as parent 1. After selecting $\xi_n^{1\prime}$ as parent 1, $\xi_n^{2\prime}$ is randomly selected from M' as parent 2. After selecting $\xi_n^{1\prime}$ and $\xi_n^{2\prime}$, a random integer, $r*$, within $1\ to\ n$ is generated to indicate the position where $\xi_n^{1\prime}$and $\xi_n^{2\prime}$ are to split up for crossover. At that point, part of $\xi_n^{1\prime}$ is joined with part of $\xi_n^{2\prime}$ to produce offsprings or designs denoted as M'^{of}. Then, M'^{of} and M' are combined to form the population of designs, M'', whose members were sorted based on the objective function or optimality criterion, $\Phi\{.\}$, so that the population of designs, M''', with the same size as the starting population of designs, M', are extracted from, M'' for mutation.

Step 4. Mutation.
Mutation is done according to a specified mutation probability, P_m*. This operation is done on an individual design point (gene) that is randomly selected. First, a random integer, $r*$, within 1 to n is generated for the mutation position. Again, a random number, r, is generated uniformly. If for any given design, $P_m* \geq r$, then, that design point is replaced with a random point from $\tilde{\chi} = [.]$ which will yield new population of designs M''' that will undergo step 2 to 4.

Step 5. Repeat step 2 to 4 for jth generations, for $j = 1, \ldots, n^{th}$ generations.

Step 6. Stopping criteria (Stop).
Stop when there is no improvement with respect to the values of the optimality criterion, $\Phi\{.\}$, in the subsequent generations.

2.2 Number of Distinct Design Points

Various ideal designs may necessitate varying quantities of design points. It is evident that any model aiming to estimate p parameters must have a minimum of p distinct design points. The ideal number of distinct design points, according to various models and optimality criteria, falls between p and $\frac{p(p+1)}{2}$ (Pukelsheim, 1993, p. 190). Designs with less than p distinct design points and more than $\frac{p(p+1)}{2}$ distinct design points may suffer from a serious loss of efficiency (Berger & Wong, 2009, p. 43).

2.3 Performance of GA

Verifying that a design is optimal is crucial after one has been proposed. This is because not all optimal designs obtained numerically are error-free (Nsude, 2016). As a result, a technique for comparing the "optimal design" and a user-specified design with each other must be used. In this work, the *D*-efficiency and *A*-efficiency were used to compare algorithms. A metric called relative efficiency allows one to compare the efficiencies of two different designs (Berger & Wong, 2009, p. 44). The best design is compared to a user-specified design using design efficiency. Design efficiency is a number that, in theory, ranges from 0 to 1. The closer an arbitrary design is to 1, the better. For instance, if the model in question has p parameters and the *D*-criterion is the goal of the experiment, the relative effectiveness of the user-specified design, ξ^* , in comparison to the *D*-optimal design, ξ, is:

$$D_{eff} = \left\{\frac{|\boldsymbol{M}(\xi^*)|}{|\boldsymbol{M}(\xi)|}\right\}^{1/P} \quad (2.6)$$

Where $|\boldsymbol{M}(\xi^*)|$ and $|\boldsymbol{M}(\xi)|$ are the determinant of the Fisher information matrices of the user specified design, ξ^* and optimal design, ξ, respectively, and p is the number of parametres in the regression equation (Drain, Carlye, Montgomery, Borror & Cook, 2004). *D*-efficiency provides a way to compare different designs, even if they have varying numbers of runs. One can formulate comparable relative efficiency for the other optimality criteria, omitting the power $(1/p)$ in the process. For instance, the relative efficiency for the *A*-criterion is:

$$A_{eff} = \left\{\frac{trace|\boldsymbol{M^{-1}}(\xi)|}{trace|\boldsymbol{M^{-1}}(\xi^*)|}\right\} \quad (2.7)$$

3. Results and Discussion

3.1 Problem 1

Consider a quadratic regression function in one factor on a regular geometric space:

$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 \; ; -1 \leq x \leq 1; \quad (3.1)$$

following the Pukelsheim (1993) rule, 3-point, 4-point, 5-point, and 6-point exact *D*-optimal and *A*-optimal designs were obtained as shown on Tables 1 and 2, respectively.

Table 1. Summary of 3-, 4-, 5- and 6-point *D*-optimal designs for $(\beta_0, \beta_1 x, \beta_2 x^2)$

s/n	n	ξ_n^*	$max\{det[\boldsymbol{M}(\xi_n^*)]\}$
1	3	(0, 1, -1)	3.997855
2	4	(-1.00, 0.00, 1.00, -0.01)	7.997562
3	5	(-1, 0, 0, 1, -1)	15.99058
4	6	(1, 0, -1, -1, 1, 0)	31.95087

Table 2. Summary of 3-, 4-, 5- and 6-point *A*-optimal designs for $(\beta_0, \beta_1 x, \beta_2 x^2)$

s/n	n	ξ_n^*	$min\{trace[\boldsymbol{M^{-1}}(\xi_n^*)]\}$
1	3	(1, 1, 1)	0.3333376
2	4	(1, 1, 1, 1)	0.2500694
3	5	(-1, 1, -1, -1)	0.2000622
4	6	(1, 1, 1, 1, 1, -1)	0.1667201

Results in Table 1 show the generated *D*-optimal designs with their associated determinant values. Similarly, the generated *A*-optimal designs with their associated traces can also be seen displayed in Table 2. Comparing the design, (- 1.00, 0.00, 1.00, -0.01) obtained by GA in this work for 4-point *D*-optimal design, to the design, (- 1, 0.0048, 0, 1) obtained by Nsude (2016), it was observed that the design generated by GA in this work and the work of Nsude (2016) are not the same. This could be attributed to the already made starting design points adopted by Nsude (2016) in generating the optimal design as used by Atkinson and Donev (1992) and Atkinson, Donev and Tobias (2007). It was also observed that a determinant value of 7.9999 was attained in generating 4-point *D*-optimal designs in the work of Nsude (2016), while a determinant value of 7.997562 was attained by GA in generating 4-point *D*-optimal designs. This indicates that GA compete favourably well with the line search algorithm of Nsude (2016) in the selection of 4-point *D*-optimal designs.

3.2 Problem 2

Consider a quadratic regression function in two factors on a regular geometric space:

$$f(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2 \; ; -1 \leq x_1, x_2 \leq 1; \quad (3.2)$$

Following the Pukelsheim (1993) rule, 6-point, 7-point, 8-point and 9-point exact *D*-optimal and *A*-optimal designs were obtained as shown on Tables 3 and 4, respectively.

Table 3. Summary of 6-, 7-, 8- and 9-point *D*-optimal designs for $(\beta_0,\ \beta_1 x_1,\ \beta_2 x_2,\ \beta_{12} x_1 x_2,\ \beta_{11} x_1^2,\ \beta_{22} x_2^2)$

s/n	n	ξ_n^*	$max\ \{det[\boldsymbol{M}(\xi_n^*)]\}$
1	6	(1.00, 0.39, -1.00, -1.00, 1.00, -0.13), (1.00, -1.00, -1.00, 1.00, -0.40, 0.13)	265.8401
2	7	(1.00, 1.00, -1.00, -0.09, 1.00, 0.07, -1.00), (1.00, -1.00, -1.00, -0.09, 0.07, 1.00, 1.00)	976.1774
3	8	(0.08, -1.00, 1.00, -1.00, 1.00, 0.08, -0.21, 1.00), (1.00, 1.00, -1.00, -1.00, 0.00, -1.00, 0.00, 1.00)	2345.087
4	9	(1, 1, 0, -1, 0, -1, 0, -1, 1), (1, -1, 1, 0, 0, 1, -1, -1, 0)	5137.131

Table 4. Summary of 6-, 7-, 8- and 9-point *A*-optimal designs for $(\beta_0,\ \beta_1 x_1,\ \beta_2 x_2,\ \beta_{12} x_1 x_2,\ \beta_{11} x_1^2,\ \beta_{22} x_2^2)$

s/n	n	ξ_n^*	$min\{trace[\boldsymbol{M^{-1}}(\xi_n^*)]\}$
1	6	(0.65, -0.96, 0.95, 0.18, 0.15, -0.06), (0.98, -0.79, -0.23, -0.95, 0.51, -0.04)	3.908065
2	7	(-0.01, 0.06, -0.35, 0.94, 0.02, -0.90, -0.85), (0.08, 0.95, -1.00, -0.75, -0.18, 0.96, 0.05)	3.080716
3	8	(-0.04, 0.99, 0.06, -0.99, 0.96, 0.18, -0.98, -0.18), (0.04, 0.97, -0.02, -0.98, -0.18, -1.00, 0.15, 1.00)	2.053511
4	9	(-0.05, -1.00, -1.00, -0.99, -0.08, 0.99, 0.99, -0.08, 0.13), (1.00, -0.98, 0.06, 1.00, 0.09, 0.99, -0.10, 0.09, -0.97)	1.79235

Tables 3 and 4 are the summary reports for the selected sample optimal points for both *D*- and *A*-optimal designs. Comparing the 6-point *D*-optimal design, (1, 1), (0.39, -1), (-1, -1), (-1, 1), (1, -0.4), (-0.13, 0.13) obtained by GA in this work with a determinant value of 265.8401, and the design (-1, 1), (-1, -1), (1, -1), (1, 1), (0, -1), (-0.0178, 0.0001) obtained by Nsude (2016) with a determinant value of 256.0000. The determinant value for the 6-point *D*-optimal design generated by GA is higher than the determinant value generated by the line search algorithm of Nsude (2016).

3.3 Problem 3

Consider a quadratic regression function in three factors on a regular geometric space:

$$f(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_{12} x_1 x_2 + \beta_{13} x_1 x_3 + \beta_{23} x_2 x_3 + \beta_{123} x_1 x_2 x_3 + \beta_{111} x_1^2 + \beta_{222} x_2^2 + \beta_{333} x_3^2 \ ; -1 \le x_1 x_2 x_3 \le 1 \quad (3.3)$$

Following the Pukelsheim (1993) rule, 11-point, 12-point, 13-point and 14-point exact *D*-optimal and *A*-optimal designs were obtained as shown on Tables 5 and 6, respectively.

Table 5. Summary of 11-, 12-, 13- and 14-point *D*-optimal designs for $(\beta_0,\ \beta_1 x_1,\ \beta_2 x_2,\ \beta_3 x_3,\ \beta_{12} x_1 x_2,\ \beta_{13} x_1 x_3,\ \beta_{23} x_2 x_3,\ \beta_{123} x_1 x_2 x_3,\ \beta_{111} x_1^2,\ \beta_{222} x_2^2,\ \beta_{333} x_3^2)$

s/n	n	ξ_n^*	$max\ \{det[\boldsymbol{M}(\xi_n^*)]\}$
1	11	(-1.00, -1.00, 1.00, 1.00, 1.00, -0.08, 1.00, -1.00, -1.00, 0.21, 1.00), (-1.00, 1.00, 0.21, 1.00, -1.00, -0.08, -1.00, -1.00, 1.00, 1.00, 1.00), (1.00, 1.00, -1.00, 1.00, 1.00, 0.09, -1.00, -1.00, -1.00, -1.00, -0.21)	12643.18
2	12	(-1.00, 1.00, 1.00, 1.00, -1.00, -1.00, 0.01, -1.00, 1.00, -0.05, 1.00, -1.00), (1.00, -1.00, -1.00, 1.00, -0.05, -1.00, 0.02, 1.00, 1.00, -1.00, 0.02, -1.00), (1.00, 1.00, -1.00, -1.00, -1.00, 1.00, 1.00, -1.00, 1.00, -0.05, 0.01, -1.00)	27859.1
3	13	(1.00, -1.00, 1.00, 1.00, -1.00, 0.00, -1.00, 0.09, -1.00, -1.00, 1.00, 1.00, -0.04), (1.00, 1.00, 1.00, -1.00, 1.00, 0.00, 0.10, -1.00, -1.00, -1.00, -1.00, -0.03, 1.00), (-1.00, -1.00, 1.00, 1.00, 1.00, -1.00, 0.03, 0.03, 1.00 -1.00, -1.00, 1.00, 1.00)	53118.09
4	14	(1.00, -1.00, 1.00, 1.00, -0.26, 1.00, -1.00, -1.00, -0.06, 1.00, -1.00, 1.00, -1.00, -0.06), (-1.00, -1.00, 1.00, -1.00, 1.00, 0.06, 1.00, 0.26, 0.06, -1.00, -1.00, 1.00, 1.00, -1.00), (1.00, 1.00, -1.00, -1.00, -1.00, -0.06, 1.00, -1.00, 1.00, 1.00, -1.00, 1.00, -0.26, -0.06)	71707.34

Table 6. Summary of 11-, 12-, 13- and 14-point *A*-optimal designs for ($\beta_0,\ \beta_1 x_1,\ \beta_2 x_2,\ \beta_3 x_3,\ \beta_{12} x_1 x_2,\ \beta_{13} x_1 x_3,\ \beta_{23} x_2 x_3,\ \beta_{123} x_1 x_2 x_3,\ \beta_{111} x_1^2,\ \beta_{222} x_2^2,\ \beta_{333} x_3^2$)

s/n	n	ξ_n^*	$min\{trace[\boldsymbol{M}^{-1}(\xi_n^*)]\}$
1	11	(-0.86, 0.32, 0.85, -0.44, -0.49, -0.25, 0.46, 0.79, -0.98, 0.21, 0.51), (-0.86, 0.90, 0.50, 0.89, 0.92, -0.32, -0.50, -0.82, -0.02, 0.68, 0.11), (-0.17, -0.76, -0.11, 0.83, 0.20, -0.68, -0.74, 0.19, -0.80, 0.74, -0.45)	4.074543
2	12	(-0.58, -0.92, 0.73, 0.23, -1.00, -0.06, -0.70, 0.91, -0.17, 0.04, 0.91, 0.73), (0.73, 0.00, 0.94, 0.75, 0.75, -0.85, -0.85, 0.10, -0.69, 0.01, 0.80, -0.85), (0.87, 0.53, -0.81, 0.82, -0.75, 0.65, -0.44, 0.88, -0.88, -0.20, 0.99, 0.14)	3.147544
3	13	(-0.41, 0.96, -0.81, 0.90, 0.02, 0.80, -0.42, 1.00, -0.99, 0.25, -1.00, -0.98, -0.05), (-1.00, -0.96, 0.99, 0.96, 0.93, 0.83, 0.90, -0.94, 0.01, -0.04, -0.98, 0.96, -0.10), (0.95, -0.80, -0.01, -0.92, -0.22, 0.78, 0.98, 0.97, 0.57, -0.05, -0.78, -1.00, -0.98)	1.988962
4	14	(0.95, 0.98, -0.02, -0.97, 0.92, 0.98, -0.02, 1.00, -0.98, -1.00, -0.96, -0.07, -0.98, -0.08), (-0.91, -0.03, -0.03, -0.16, 0.99, -0.89, -0.05, 0.99, 0.98, -0.98, -0.93, -0.98, 1.00, 0.95), (0.96, -0.06, 0.98, -0.99, 0.95, -0.99, -0.04, -0.99, -0.95, -0.16, 0.97, -1.00, 0.88, -0.07)	1.595788

In Tables 5 and 6 are the *D*- and *A*-optimal designs with their associated determinant and trace values obtained by GA.

3.4 Problem 4

A polynomial regression function in one factor on regular geometric space:

$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 x^4 + \beta_5 x^5 \ ; -1 \le x \le 1; \qquad (3.4)$$

Following the Pukelsheim (1993) rule, 6-point, 7-point, 8-point and 8-point exact *D*-optimal and *A*-optimal designs were obtained as shown on Tables 7 and 8, respectively.

Table 7. Summary of 6-, 7-, 8- and 9-point *D*-optimal designs for ($\beta_0,\ \beta_1 x,\ \beta_2 x^2,\ \beta_3 x^3,\ \beta_4 x^4,\ \beta_5 x^5$)

s/n	n	ξ_n^*	$max\ \{det[\boldsymbol{M}(\xi_n^*)]\}$
1	6	(-0.28, -1.00, 1.00, 0.29, 0.77, -0.76)	0.004123225
2	7	(-0.31, -1.00, 1.00, 0.15, 0.39, 0.77, -0.77)	0.008402192
3	8	(-0.31, -1.00, 1.00, 0.77, -0.77, 0.16, 0.39, -1.00)	0.01680674
4	9	(0.39 0.18 -0.77 -1.00 -0.30 -0.77 1.00 -1.00 0.77)	0.03340839

Table 8. Summary of 6-, 7-, 8- and 9-point *A*-optimal designs for ($\beta_0,\ \beta_1 x,\ \beta_2 x^2,\ \beta_3 x^3,\ \beta_4 x^4,\ \beta_5 x^5$)

s/n	n	ξ_n^*	$min\{trace[\boldsymbol{M}^{-1}(\xi_n^*)]\}$
1	6	(-1.00, 1.00, -1.00, 1.00, 1.00, -0.99)	0.1674673
2	7	(1, 1, -1, 1, -1, 1, -1)	0.1436902
3	8	(1.00, 1.00, -0.99, -1.00, 1.00, 1.00, 1.00, -1.00)	0.1258784
4	9	(1.00, 0.99, 1.00, 0.99, -1.00, 1.00, -1.00, -0.99, -1.00)	0.1118684

Tables 7 and 8 are the summary reports depicting the selected sample optimal points for both *D*- and *A*-optimal designs, with their determinants and trace values, respectively. Comparing the determinant value 0.01680674 attained by GA for 8-point *D*-optimal design in this work, to the determinant value obtained by Fedorov algorithm (0.00000006170), modified Fedorov algorithm (0.00000006437) and k-exchange algorithm (0.00000006403) (Johnson & Nachtshem, 1983, p. 275), it can be seen that the GA achieved a higher determinant value relative to the three algorithms; implying that GA was able to search for better 6-, 7-, 8- and 9-point *D*- and *A*-optimal designs relative to Fedorov algorithm, modified Fedorov algorithm and k-exchange algorithm.

3.5 Problem 5

Consider a Scheffé regression function in three factors on an irregular geometric space:

$$f(x) = \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 \; ; \{0.2 \leq x_1 \leq 0.8; 0.2 \leq x_2 \leq 0.8; 0.0 \leq x_3 \leq 0.6\}; \quad (3.5)$$

9-point exact *D*-optimal and *A*-optimal designs were obtained as shown on Tables 9 and 10.

Table 9. Summary of 9-point *D*-optimal designs for $(\beta_1 x_1,\ \beta_2 x_2,\ \beta_3 x_3)$

s/n	n	ξ_n^*	$max\ \{det[\boldsymbol{M}(\xi_n^*)]\}$
1	9	(0.8, 0.8, 0.8, 0.2, 0.2, 0.8, 0.8, 0.2, 0.8), (0.2, 0.8, 0.2, 0.8, 0.8, 0.8, 0.2, 0.8, 0.8), (0.6, 0.0, 0.6, 0.6, 0.6, 0.0, 0.6, 0.6, 0.0)	8.9283

Table 10: Summary of 9-point *A*-optimal designs for $(\beta_1 x_1,\ \beta_2 x_2,\ \beta_3 x_3)$

s/n	n	ξ_n^*	$min\{trace[\boldsymbol{M^{-1}}(\xi_n^*)]\}$
1	9	(0.80, 0.80, 0.80, 0.20, 0.80, 0.80, 0.20, 0.21, 0.20), (0.80, 0.20, 0.80, 0.80, 0.20, 0.20, 0.80, 0.80, 0.20), (0.00, 0.60, 0.00, 0.60, 0.60, 0.00, 0.60, 0.00, 0.60)	2.0597

Tables 9 and 10 are the summary reports depicting the selected sample optimal points for both *D*- and *A*-optimal designs, with their determinants and trace values, respectively. Comparing the determinant value 8.9283 attained by GA for 9-point *D*-optimal design in this work, to the determinant value obtained by a computer algorithm (3.4992) in the work of Fasoranbaku and Daramola (2018), it can be seen that the GA achieved a higher determinant value relative to the computer algorithm reported by Fasoranbaku and Daramola (2018). In similar vein, comparing the trace value 2.0597 attained by GA for 9-point *A*-optimal design in this work, to the trace value obtained by a computer algorithm (2.2593) in the work of Fasoranbaku and Daramola (2018), shows that the GA achieved a lower trace value relative to the computer algorithm reported by Fasoranbaku and Daramola (2018). This goes to show that GA was able to search for better 9-point *D*- and *A*-optimal designs relative to the computer algorithm reported by Fasoranbaku and Daramola (2018).

3.6 Performances of the GA

Here, GA's performance on one- and two-factor quadratic models is compared to Nsude (2016) line search approach. On a one-factor polynomial model, GA's performance was also contrasted with that of the Fedorov, modified Fedorov, and k-exchange algorithms. Lastly, the efficacy of GA is contrasted with the computer algorithm used in the three-factor Scheffé regression model by Fasoranbaku and Daramola (2018).

Table 11. Summary results of the performance of GA relative to algorithms studied in this work

Design problem			Algorithms	Performance of GA relative to the algorithms used in this work	
Regions	*p*	*n*		*D*-efficiency	*A*-efficiency
Regular region	3	4	LSA	0.9997	-
	6	6	LSA	≥ 1	-
	5	8	FA	≥ 1	-
			MFA	≥ 1	-
			KEA	≥ 1	-
Irregular region	3	9	CA	≥ 1	≥ 1

p = number of parameters in model, n = number of design points, LSA= Line Search Algorithm, FA = Fedorov Algorithm, MFA = Modified Fedorov Algorithm, KEA = K-Exchange Algorithm, CA = Computer Algorithm

Table 11 summarizes the performance of GA relative to line search algorithm (Nsude, 2016), Fedorov algorithm, modified Fedorov algorithm, k-exchange algorithm and computer algorithm (Fasoranbaku & Daramola, 2018), in the search for *D*- and *A*-optimal designs. For a simple design problem, results in Table 11 show that GA compares favourably well with the line search algorithm reported by Nsude (2016), in the search for 4-point *D*-

optimal designs. Also, further results in Table 11, reveal that GA compares even better than the line search algorithm (Nsude, 2016), Fedorov algorithm, modified Fedorov algorithm, k-exchange algorithm and the computer algorithm reported by Fasoranbaku and Daramola (2018), in the search for *D*- and *A*-optimal design points.

4. Discussion of Results

Results from the study revealed that for a one factor quadratic model, the 4-point *D*-optimal design obtained by GA in this work, was as efficient as the 4-point *D*-optimal design obtained by the line search algorithm, reported by Nsude (2016). This is because both algorithms do not seem to differ in their determinant values obtained in the search for *D*-optimal design. However, for a two-factor quadratic model, it was also discovered that the 6-point *D*-optimal design obtained by GA was more efficient than the 6-point *D*-optimal design obtained by the line search algorithm reported by Nsude (2016). The differences in the determinant values obtained by both algorithms in the search for *D*-optimal designs, was an indication that GA ($|\boldsymbol{M}(\xi_6)| = 265.8401$) performed better than the line search algorithm ($|\boldsymbol{M}(\xi_6)| = 256.0000$) in the search for the *D*-optimal design.

It was also established here that for a one-factor regression model of order five, GA generated 6-, 7-, 8- and 9-point *D*-optimal designs which were more efficient than the 6-, 7-, 8- and 9-point *D*-optimal designs obtained by the exchange-type algorithms such as the Fedorov algorithm, modified Fedorov algorithm and k-exchange algorithm. This finding is in line with an earlier study by Limmun et al. (2012) who reported that GA performed better than the exchange type algorithms in generating *D*-optimal designs.

Further findings from the study show that for a three-factor Scheffé regression model, GA generated 9-point *D*- and *A*-optimal designs that were more efficient than the 9-point *D*- and *A*-optimal designs generated by the computer algorithm reported in the work of Fasoranbaku and Daramola (2018). Supporting this finding, Limmun et al. (2012) concluded that GA is an alternative algorithm for generating optimal designs where the design region is highly constrained.

Lastly, the results from this work reveal that for a simple design problem, GA compares favourably well with the line search algorithm reported by Nsude (2016), in the search for 4-point *D*-optimal designs. Also, it was revealed that GA compares even better than the line search algorithm (Nsude, 2016), Fedorov algorithm, modified Fedorov algorithm, k-exchange algorithm and the computer algorithm reported by Fasoranbaku and Daramola (2018), in the search for *D*- and *A*-optimal design points as the design problem get more complex. That is, from a regular to irregular experimental region, from a regression model with one explanatory variable to regression models with more than two explanatory variables, from a model of order 1 to a model of order ≥ 2 and as the design size, *n* increases.

5. Conclusion

It has been demonstrated that the genetic algorithm employed in this work is capable of choosing *D*- and *A*-optimal designs in both regular and irregular regions. In the search for *D*-optimal designs, the genetic algorithm compares favourably, even better than the k-exchange, Fedorov, modified Fedorov, and line-search algorithms. The results also demonstrate that, while looking for *D*- and *A*-optimal designs, GA compares better than the computer technique presented by Fasoranbaku and Daramola (2018).

5. 1 Recommendations

1. GA should be used for the selection of optimal designs when the model comprise of more than two factors.
2. The selection of the best designs for polynomial models ought to be done using GA.
3. GA should be used for the selection of optimal designs when the design region is irregular.
4. More work is needed to be done using genetic algorithm to search for optimal designs based on other optimality criteria not considered herein.
5. Work also needs to be carried out on how GA can be hybridized with other algorithms that search for optimal designs in order to possibly gain some improvements in the search for optimal designs.

5.2 Contribution to Knowledge

This study added to the body of knowledge by empirically demonstrating that GA can be utilised as a substitute algorithm for the selection of *D*- and *A*-optimal exact designs in regular and irregular experimental regions while taking regression models with three or more explanatory variables into consideration.

References

Atkinson, A. C., & Donev, A. C. (1992). *Optimum experimental designs*. Oxford, England: Oxford University Press.

Atkinson, A. C., Donev, A. N., & Tobias, R. D. (2007), *Optimum experimental designs, with SAS*. Oxford, England: Oxford University Press.

Berger, M. P. F., & Wong, W. K. (2009). *An introduction to optimal designs for social and biomedical research*. United Kingdom: John Wiley & Sons Ltd.

Borkowski, J. J. (2003). Using a genetic algorithm to generate a small exact response surface designs. *Journal of Probability and Statistical Science, 1*(1), 65-88.

Broudiscou, A., Leardi, R., & Phan-Tan-Luu, R. (1996). Genetic algorithm as a tool for selection of D-optimal design. *Chemometrics and Intelligent Laboratory Systems, 35*, 105−116.

Civicioglu, P. (2013). Backtracking search optimization algorithm for numerical optimization problems. *Applied Mathematics and Computation, 219* (15), 8121-8144.

Collins, L. M., Dziak, J. J., & Li, R. (2009). Design of experiments with multiple independent variables: A resource management perspective on complete and reduced factorial designs. *Psychological Methods, 14*(3), 202–224.

Cook, R. D., & Nachtsheim, C. J. (1980). A comparison of algorithms for constructing exact D-optimal designs. *Technometrics, 22*(3), 315−324.

Drain, D., Carlye, M. W., Montgomery, D. C., Borror, C., & Cook, A. C. (2004). A genetic algorithm hybrid for constructing optimal response surface designs. *Quality and Reliability Engineering International, 20*, 637 – 650.

Fasoranbaku, O.A., & Daramola, G.O. (2018). Towards a better estimation of the parameters of linear regression models: The optimal designed experiment approach. *International Journal of New Technology and Research, 4*(5), 31-36.

Fedorov, V. V. (1972), *Theory of optimal experiments.* New York, USA: Academic Press Inc.

Iwundu, M., & Chigbu, P. (2012). A hill-climbing combinatorial algorithm for constructing N-point D-optimal exact designs. *Journal of Statistics Applications & Probability, 1*(2), 133-146.

Jason, B. (2020). How to choose an optimization algorithm. Retrieved from https://machinelearningmastery.com/tour-of-optimization-algorithms/

Johnson, M. E., & Nachtsheim, C. J. (1983). Some guidelines for constructing exact D-optimal designs on convex design spaces. *Technometrics, 25*(3), 271 - 277.

Jones, B., Allen-Moyer, K., & Goos, P. (2020). A-optimal versus D-optimal design for screening experiments. *Journal of Quality Technology.* doi: https://doi.org/10.1080/00224065.2020.1757391.

Kiefer, J. (1974). General Equivalence theory for designs (Approximate theory). *Annals of Mathematical Statistics, 2*, 849-879.

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*, 1055-1068.

Mitchell, T. J. (1974). An algorithm for the construction of D-optimal experimental designs. *Technometrics, 16*(2), 203-210.

Montepiedra, G. (1998). Application of genetic algorithms in the construction of exact D-optimal designs. *Journal of Applied Statistics, 25*, 817−826.

Nambiar, P. M. (2015). *Genetic algorithms developed in R software for finding optimal experimental designs* (Master's thesis, University of Georgia, USA). Retrieved from https://esploro.libs.uga.edu/esploro/outputs/graduate/Genetic-algorithms-developed-in-R-software-for-finding-optimal-experimental-designs/9949333759602959.

Nsude, F. I. (2016). *Line search algorithms for constructing D-optimal exact designs in continuous experimental regions* (Doctoral thesis, University of Nigeria Nsukka, Nigeria). Retrieved from https://oer.unn.edu.ng/read/line-search-algorithms-for-constructing-d-optimal-exact-designs-in-continuous-experimental-regions.

Oladugba, A. V., & Madukaife, M. S. (2009). D-optimality and D_L-optimality criteria for incomplete block designs. *Global Journal of Mathematical Sciences, 8*(2), 107-115.

Onukogu, I. B., & Iwundu, M. P. (2007). A combinatorial procedure for constructing D-optimal exact designs. *Statistica, anno LXVII, N.4*, 416-423.

Pukelsheim, F. (1993). *Optimal experimental design.* New York, USA: John Wiley & Son.

Rady, E. A., Abd El-Monsef, M. M. E., & Seyam, M. M. (2009). *Relationships among several optimality criteria*. Retrieved from https://scholar.cu.edu.eg/?q=elhoussainy/files/10.1.1.586.5137.pdf.

Ugbe, T.A., & Chigbu, P.E. (2012). On D-optimality criterion of non-overlapping and overlapping segmentation of the response surfaces. *Research Journal of Mathematics and Statistics 4*(1), 1-5.

Upperman, P. M. (1993). *Designs with a small number of runs for factorial experiments* (Doctoral thesis, Technische Universiteit Eindhoven, Netherlands). Retrieved from https://doi.org/10.6100/IR392978.

Vandenberghe, L., & Boyd, S. (1999). Applications of semi-definite programming. *Applied Numerical Mathematics: Transactions of IMACS 29*(3), 283-299.