

Invariant Property Based Algorithm for Solving Linear Programming Problems

Idorenyin Etukudo*, Mfoniso Umoren**

Abstract

This paper presents an invariant property based algorithm (IPBA) for solving linear programming problems. This new algorithm which uses the principles of optimal designs of experiment shows that the direction vector used for obtaining optimal solution of a given linear programming problem (LPP) is the same as the gradient of the objective function, thereby disregarding the need for partitioning the experimental region, the calculation of the information matrices and their inverses, the calculation of the Hessian matrices and average information matrix as well as the response vector before the direction vector could be obtained as in the case of the modified super convergent line series algorithm (MSCLS_L). This new algorithm further simplifies and guarantees the existence of a solution to the LPP. An illustrative example shows that IPBA for solving linear programming problems is quite efficient and the result obtained was similar to that obtained by using the simplex method or Karmarkar's interior point method.

Keywords

Optimal designs, modified super convergent line series, invariant property, support points, direction vector, objective function gradient, linear programming.

1 INTRODUCTION

In this paper, an invariant property based algorithm (IPBA) for solving linear programming problems is presented. As in the modified super convergent line series algorithm (MSCLS_L) for solving linear programming problems developed by [1], the new approach, also a line search algorithm makes use of the principles of optimal designs of experiment. In MSCLS_L, the support points that make up the initial design matrix obtained from the entire experimental space or feasible region are partitioned into the desired groups of equal sizes. In IPBA, this partitioning is not necessary.

To obtain a D-optimal non-singular design for a p-parameter response function, [2] showed that we do not need more than

$$p \leq N \leq \frac{1}{2}p(p+1) + 1 \quad (1)$$

support points from the entire experimental region. In his contribution, Onukogu in [3], [4] has shown that the number of support points for any response surface is given by

$$n+1 \leq N \leq \frac{1}{2}n(n+1) + 1 \quad (2)$$

In order to partition the entire experimental region into k^* groups of equal sizes, Umoren and Etukudo in [5] showed that the total number of support points required is

$$2k^*(n+1) \leq 2N \leq k^*n(n+1) + 2k^* \quad (3)$$

which now reduces to

$$2(n+1) \leq N \leq n(n+1) + 2 \quad (4)$$

since the partitioning is not necessary.

Etukudo and Umoren [6] used linear transformation of average information matrix and the response vector of the design to prove that the direction vector of a linear programming problem is invariant under choice of design matrices. This new algorithm takes advantage of this invariant property in MSCLS_L since the direction vector for any linear programming problem is equivalent to the gradient vector or the coefficients of the objective function of the linear programming problem.

In MSCLS_L, the calculation of information matrices and their inverses, the Hessian matrices and normalized Hessian matrices, the average information matrix as well as the response vector are required in order to obtain the direction vector. These calculations are not needed in IPBA since the direction vector is the coefficient of the objective function. The importance of this is that, apart from removing the possibility of non-existence of inverses of partitioned matrices which will in turn truncate the existence of the solution to the linear programming problem, the IPBA performs better than MSCLS_L judging from some measures of efficiency such as number of sequential steps, number of experiments performed, computer time required and computer storage space.

2 INVARIANT PROPERTY BASED ALGORITHM FOR SOLVING LINEAR PROGRAMMING PROBLEMS

This new algorithm, namely, invariant property based algorithm (IPBA) for solving linear programming problems, which is a line search algorithm, makes use of the principles of optimal designs of experiment. The sequential steps involved

*Department of Statistics
University of Calabar, Nigeria.
Email: nseobongidorenyin@gmail.com

**Department of Mathematics & Statistics
University of Uyo, Nigeria.

in IPBA are as follows:

Initialization Step: Initial design matrix.
 Choose an initial design matrix, X from a response surface such that

$$2(n+1) \leq N \leq n(n+1) + 2 \quad (5)$$

where n = number of decision variables and N = number of support points, chosen in such a way that the constraints are not violated.

Step 1: Optimal starting point.
 Use the initial design matrix to determine the optimal starting point, \mathbf{x}_1^* .

Step 2: Direction of movement.
 Input $\mathbf{d} = \mathbf{c}$, where $\mathbf{c} = (c_1, c_2, \dots, c_n)$ is the gradient of the objective function. Normalize \mathbf{d} to obtain \mathbf{d}^* .

Step 3: Determine the optimal step length, ρ_1^* .

For a maximization problem, use

$$\rho_1^* = \rho_{li}^* = \max_i \left\{ \frac{\mathbf{A}_i \mathbf{x}_1^* - b_i}{\mathbf{A}_i \mathbf{d}^*} \right\} \quad (6)$$

and for a minimization problem, use

$$\rho_1^* = \rho_{li}^* = \min_i \left\{ \frac{\mathbf{A}_i \mathbf{x}_1^* - b_i}{\mathbf{A}_i \mathbf{d}^*} \right\} \quad (7)$$

where $\mathbf{A}_i \mathbf{x} = b_i$, $i = 1, 2, \dots, m$ is the i th constraint of the linear programming problem.

Step 4: First movement.
 Make a move to the point

$$\mathbf{x}_2^* = \mathbf{x}_1^* - \rho_1^* \mathbf{d}^* \quad (8)$$

Step 5: Termination criteria

- Compute $f(\mathbf{x}_2^*)$ and $f(\mathbf{x}_1^*)$.
- If $|f(\mathbf{x}_2^*) - f(\mathbf{x}_1^*)| < \varepsilon$ where $\varepsilon = 0.0001$, the algorithm terminates. If not, replace \mathbf{x}_1^* by \mathbf{x}_2^* and determine a new step length using the constraint that gave the optimum step length

in step 3. If the new step length, $\rho_2^* = 0$, then the optimizer had earlier been located in step 4.

3 AN ILLUSTRATIVE EXAMPLE

By using the invariant property based algorithm (IPBA) for solving linear programming problems,

$$\text{Maximize } f(\mathbf{x}) = 850x_1 + 350x_2 \quad (9)$$

Subject to

$$\begin{aligned} 8x_1 + 5x_2 &\leq 200 \\ 4x_1 + x_2 &\leq 48 \\ 5x_1 + 4x_2 &\leq 80 \\ 3x_1 + 5x_2 &\leq 150 \\ x_1, x_2 &\geq 0 \end{aligned}$$

Initialization Step: Initial design matrix
 Given the above response surface and $n = 2$, then

$$\begin{aligned} 2(2+1) &\leq N \leq 2(2+1) + 2 \\ 6 &\leq N \leq 8 \end{aligned} \quad (10)$$

By arbitrarily choosing 8 support points as long as they do not violate any of the constraints, we make up the initial design matrix

$$X = \begin{bmatrix} 1 & 10 & 7 \\ 1 & 8 & 8 \\ 1 & 7 & 4 \\ 1 & 6 & 8 \\ 1 & 5 & 7 \\ 1 & 9 & 4 \\ 1 & 4 & 6 \\ 1 & 11 & 3 \end{bmatrix} \quad (11)$$

Step 1: Optimal starting point.
 Use the initial design matrix to determine the optimal starting point, \mathbf{x}_1^* where

$$\begin{aligned} \mathbf{x}_1^* &= \sum_{m=1}^N u_m^* \mathbf{x}_m^* \\ u_m^* &> 0; \quad \sum_{m=1}^N u_m^* = 1 \end{aligned} \quad (12)$$

$$u_m^* = \frac{a_m^{-1}}{\sum_{m=1}^N a_m^{-1}}, m = 1, 2, \dots, N$$

$$a_m = \mathbf{x}_m \mathbf{x}_m', m = 1, 2, \dots, N$$

Now,

$$a_1 = \mathbf{x}_1 \mathbf{x}_1' = \begin{bmatrix} 1 & 10 & 7 \\ 10 & 10 & 7 \\ 7 & 7 & 7 \end{bmatrix} = 150,$$

$$a_1^{-1} = 0.0067$$

$$a_2 = \mathbf{x}_2 \mathbf{x}_2' = \begin{bmatrix} 1 & 8 & 8 \\ 8 & 8 & 8 \\ 8 & 8 & 8 \end{bmatrix} = 129,$$

$$a_2^{-1} = 0.0078$$

$$a_3 = \mathbf{x}_3 \mathbf{x}_3' = \begin{bmatrix} 1 & 7 & 4 \\ 7 & 7 & 4 \\ 4 & 4 & 4 \end{bmatrix} = 66,$$

$$a_3^{-1} = 0.0152$$

$$a_4 = \mathbf{x}_4 \mathbf{x}_4' = \begin{bmatrix} 1 & 6 & 8 \\ 6 & 6 & 8 \\ 8 & 8 & 8 \end{bmatrix} = 101,$$

$$a_4^{-1} = 0.0099$$

$$a_5 = \mathbf{x}_5 \mathbf{x}_5' = \begin{bmatrix} 1 & 5 & 7 \\ 5 & 5 & 7 \\ 7 & 7 & 7 \end{bmatrix} = 75,$$

$$a_5^{-1} = 0.0133$$

$$a_6 = \mathbf{x}_6 \mathbf{x}_6' = \begin{bmatrix} 1 & 9 & 4 \\ 9 & 9 & 4 \\ 4 & 4 & 4 \end{bmatrix} = 98,$$

$$a_6^{-1} = 0.0102$$

$$a_7 = \mathbf{x}_7 \mathbf{x}_7' = \begin{bmatrix} 1 & 4 & 6 \\ 4 & 4 & 6 \\ 6 & 6 & 6 \end{bmatrix} = 53,$$

$$a_7^{-1} = 0.0189$$

$$a_8 = \mathbf{x}_8 \mathbf{x}_8' = \begin{bmatrix} 1 & 11 & 3 \\ 11 & 11 & 3 \\ 3 & 3 & 3 \end{bmatrix} = 131,$$

$$a_8^{-1} = 0.0076$$

$$\sum_{m=1}^8 a_m^{-1} = 0.0067 + 0.0078 + \dots + 0.0189 + 0.0076 = 0.0896$$

Since

$$u_m^* = \frac{a_m^{-1}}{\sum_{m=1}^N a_m^{-1}}, m = 1, 2, \dots, N, \text{ then}$$

$$u_1^* = \frac{0.0067}{0.0896} = 0.0748,$$

$$u_2^* = \frac{0.0078}{0.0896} = 0.0871,$$

$$u_3^* = \frac{0.0152}{0.0896} = 0.1696,$$

$$u_4^* = \frac{0.0099}{0.0896} = 0.1105,$$

$$u_5^* = \frac{0.0133}{0.0896} = 0.1484,$$

$$u_6^* = \frac{0.0102}{0.0896} = 0.1138,$$

$$u_7^* = \frac{0.0189}{0.0896} = 0.2109,$$

$$u_8^* = \frac{0.0076}{0.0896} = 0.0848$$

$$d^* = \begin{bmatrix} d_1^* \\ d_2^* \end{bmatrix} = \begin{bmatrix} \frac{850}{\sqrt{850^2 + 350^2}} \\ \frac{350}{\sqrt{850^2 + 350^2}} \end{bmatrix} = \begin{bmatrix} 0.9247 \\ 0.3808 \end{bmatrix}$$

Hence, the optimal starting point is

$$x_1^* = \sum_{m=1}^N u_m^* x_m$$

$$= 0.0748 \begin{bmatrix} 1 \\ 10 \\ 7 \end{bmatrix} + 0.0871 \begin{bmatrix} 1 \\ 8 \\ 8 \end{bmatrix} + 0.1696 \begin{bmatrix} 1 \\ 7 \\ 4 \end{bmatrix}$$

$$+ 0.1105 \begin{bmatrix} 1 \\ 6 \\ 8 \end{bmatrix} + 0.1484 \begin{bmatrix} 1 \\ 5 \\ 7 \end{bmatrix} + 0.1138 \begin{bmatrix} 1 \\ 9 \\ 4 \end{bmatrix}$$

$$+ 0.2109 \begin{bmatrix} 1 \\ 4 \\ 6 \end{bmatrix} + 0.0848 \begin{bmatrix} 1 \\ 11 \\ 3 \end{bmatrix}$$

$$x_1^* = \begin{bmatrix} 0.9999 \\ 6.8376 \\ 5.7966 \end{bmatrix}$$

Step 2: Direction of movement.

We note that the direction vector, d is equivalent to the coefficient of the objective function, hence

$$d = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = \begin{bmatrix} 850 \\ 350 \end{bmatrix}$$

and by normalizing d such that $d^* \cdot d^* = 1$, we have

Step 3: Determine the optimal step length, ρ_1^* from

$$\rho_1^* = \rho_{1i}^* = \max_i \left\{ \frac{A_i x_1^* - b_i}{A_i d^*} \right\}$$

where $A_i x = b_i$, $i = 1, 2, \dots, m$ is the i th constraint of the linear programming problem.

For $A_1 = [8 \ 5]$ and $b_1 = 200$

$$\rho_1^* = \rho_{11}^* = \left\{ \frac{[8 \ 5] \begin{bmatrix} 6.8376 \\ 5.7966 \end{bmatrix} - 200}{[8 \ 5] \begin{bmatrix} 0.9247 \\ 0.3808 \end{bmatrix}} \right\} = -12.5050$$

For $A_2 = [4 \ 1]$ and $b_2 = 48$

$$\rho_1^* = \rho_{12}^* = \left\{ \frac{[4 \ 1] \begin{bmatrix} 6.8376 \\ 5.7966 \end{bmatrix} - 48}{[4 \ 1] \begin{bmatrix} 0.9247 \\ 0.3808 \end{bmatrix}} \right\} = -3.6408$$

For $A_3 = [5 \ 4]$ and $b_3 = 80$

$$\rho_1^* = \rho_{13}^* = \left\{ \frac{[5 \ 4] \begin{bmatrix} 6.8376 \\ 5.7966 \end{bmatrix} - 80}{[5 \ 4] \begin{bmatrix} 0.9247 \\ 0.3808 \end{bmatrix}} \right\} = -3.6809$$

For $A_4 = [3 \ 5]$ and $b_4 = 150$

$$\rho_1^* = \rho_{14}^* = \left\{ \frac{[3 \ 5] \begin{bmatrix} 6.8376 \\ 5.7966 \end{bmatrix} - 150}{[3 \ 5] \begin{bmatrix} 0.9247 \\ 0.3808 \end{bmatrix}} \right\} = -21.4840$$

Hence, we choose the maximum step length,

$$\rho_1^* = \rho_{12}^* = -3.6408$$

Step 4: First movement
 Make a move to the point

$$\begin{aligned} \mathbf{x}_2^* &= \mathbf{x}_1^* - \rho_1^* \mathbf{d}^* \\ &= \begin{bmatrix} 6.8376 \\ 5.7966 \end{bmatrix} - (-3.6408) \begin{bmatrix} 0.9247 \\ 0.3808 \end{bmatrix} \\ &= \begin{bmatrix} 10.2042 \\ 7.1830 \end{bmatrix} \end{aligned}$$

Step 5: Termination criteria

- (a) Computing $f(\mathbf{x}_2^*)$ and $f(\mathbf{x}_1^*)$, we have
- $$f(\mathbf{x}_2^*) = 850(10.2042) + 350(7.1830) = 11,187.62$$
- $$f(\mathbf{x}_1^*) = 850(6.8376) + 350(5.7966) = 7,840.77$$

(b) Since

$$\begin{aligned} |f(\mathbf{x}_2^*) - f(\mathbf{x}_1^*)| &= |11,187.62 - 7,840.77| \\ &= 3,346.85 > \varepsilon = 0.0001, \end{aligned}$$

we make a second move and replace \mathbf{x}_1^* by \mathbf{x}_2^* to determine a new step length using the constraint that gave the optimum step length in step 3. This is obtained as follows:

$$\rho_2^* = \rho_{22}^* = \left\{ \frac{[4 \ 1] \begin{bmatrix} 10.2042 \\ 7.1830 \end{bmatrix} - 48}{[4 \ 1] \begin{bmatrix} 0.9247 \\ 0.3808 \end{bmatrix}} \right\} = -0.000049 \approx 0$$

Since the new step length, $\rho_2^* = 0$, then the optimizer had earlier been located in step 4.

Hence, $\mathbf{x}_2^* = \begin{bmatrix} 10.2042 \\ 7.1830 \end{bmatrix}$ and $f(\mathbf{x}_2^*) = 11,187.62$ or

$$\mathbf{x}_2^* = \begin{bmatrix} 10 \\ 7 \end{bmatrix} \text{ and } f(\mathbf{x}_2^*) = 10,950.$$

4 CONCLUSION

In this work, the primary objective of the study has been successfully executed, namely, the development of an invariant property based algorithm for solving linear programming problems. This method guarantees the existence of an optimizer with less computational effort since inverse of the constraint coefficient is not needed. In this solution technique of solving LPP, there is no need of partitioning the experimental region into segments since the calculation of the information matrices and their inverses, the calculation of the Hessian matrices and the average information matrix of those segments which are required in any algorithm using the principles of optimal designs of experiment are not necessary in IPBA.

Again, the calculation of the response vector in order to obtain the direction vector was omitted here since it had been shown that the direction vector of a given LPP is the same as the gradient of the objective function of the LPP. The importance of this is that, apart from removing the possibility of non-existence of inverses of partitioned matrices which will in turn truncate the existence of the solution to the linear programming problem, the IPBA simplifies the attainment of an optimizer and converges at a faster rate. Result

obtained from the numerical illustration gives $\mathbf{x}_2^* = \begin{bmatrix} 10 \\ 7 \end{bmatrix}$ and

$f(\mathbf{x}_2^*) = 10,950$ which agrees with that obtained by using either the simplex method or the Karmarkar's interior point method.

REFERENCES

- [1] Etukudo, I. A. and Umoren, M. U., 2008, "Modified Super Convergent Line Series Algorithms For Solving Linear Programming Problems", Journal of Mathematical Sciences, Vol. 19, No. 1, 73 – 88, International Centre for Advanced Studies, West Bengal, India.
- [2] Pazman, A. (1986): Foundations of Optimum Experimental Design, D-Reidal Publishings Company, Boston.
- [3] Onukogu, I. B. (1997): Foundations Optimal Exploration of Response Surfaces, Epharata Press, Nsukka, Nigeria.
- [4] Onukogu, I. B. (2002): Super Convergent Line Series in Optimal Designs of Experiment and Mathematical Programming, AP Express Publishers, Nigeria.
- [5] Etukudo, I. A. and Umoren, M. U., 2009, "Modified Super Convergent Line Series Algorithms For Solving Quadratic Programming Problems", Journal of Mathematical Sciences, Vol. 20, No. 1, 55 – 66, International Centre for Advanced Studies, West Bengal, India.
- [6] Etukudo, I. A. and Umoren, M. U., 2009, "Invariant Property of the Direction Vector of Linearly Constrained Optimization Problems with Linear Objective Function", Journal of Mathematical Sciences, Vol. 19, No. 1, 73 – 88, International Centre for Advanced Studies, West Bengal, India.