

PLANEACIÓN PARA LA PROGRAMACIÓN DE UN ALGORITMO DE CONSTRUCCIÓN DE DISEÑOS FRACCIONADOS DE NIVELES MIXTOS MEDIANTE EL MÉTODO NOBA

*PLANNING FOR THE PROGRAMMING OF A CONSTRUCTION
ALGORITHM OF FRACTIONAL DESIGNS OF MIXED LEVELS BY
MEANS OF THE NOBA METHOD*

Alejandro Silva García

Tecnológico Nacional de México / IT de Celaya, México
M2203049@itcelaya.edu.mx

Armando Javier Ríos Lira

Tecnológico Nacional de México / IT de Celaya, México
armando.rios@itcelaya.edu.mx

Yaquelin Verenice Pantoja Pacheco

Tecnológico Nacional de México / IT de Celaya, México
yaquelinpantoja@hotmail.com

Recepción: 24/noviembre/2023

Aceptación: 4/julio/2024

Resumen

Los diseños de niveles mixtos desempeñan un papel muy importante en la experimentación industrial, ya que la mayoría de los procesos industriales involucran factores cualitativos como cuantitativos, al trabajar con estos factores en diferentes niveles de operación tiene como consecuencia que el número de corridas experimentales aumente dramáticamente. Se han propuesto varios algoritmos para construir diseños ortogonales equilibrados de niveles mixtos, para reducir el número de corridas experimentales, entre los cuales se encuentra el método construir matrices balanceadas casi ortogonales (NOBA). El método NOBA demostró que es posible crear fracciones equilibradas casi ortogonales de tamaño económico, reduciendo el número de corridas experimentales considerablemente. Actualmente no existe un programa que ejecute este método por ello el propósito de esta investigación es plantear el desarrollo de un programa informático para la construcción de diseños fraccionados de niveles mixtos mediante el método NOBA.

Palabras Clave: Algoritmo, diseños de niveles mixtos, diseños factoriales fraccionados, método NOBA, programación.

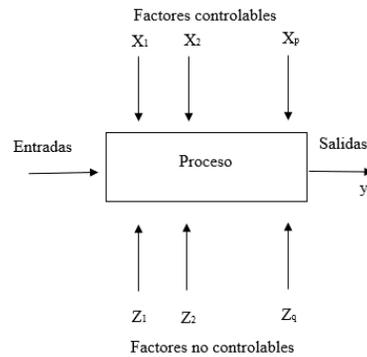
Abstract

Mixed-level designs play a vital role in industrial experimentation, since most industrial processes involve both qualitative and quantitative factors and working with these factors at diverse levels of operation results in a dramatic increase in the number of experimental runs. Several algorithms have been proposed to construct orthogonal balanced mixed-level designs to reduce the number of experimental runs, among which is the near-orthogonal balanced array construction (NOBA) method. The NOBA method demonstrated that it is possible to create near-orthogonal balanced fractions of economic size, reducing the number of experimental runs considerably. Currently there is no program that executes this method, so the purpose of this research is to propose the development of a computer program for the construction of mixed-level fractional designs using the NOBA method.

Keywords: *Algorithm, mixed level designs, fractional factorial designs, NOBA method, programming.*

1. Introducción

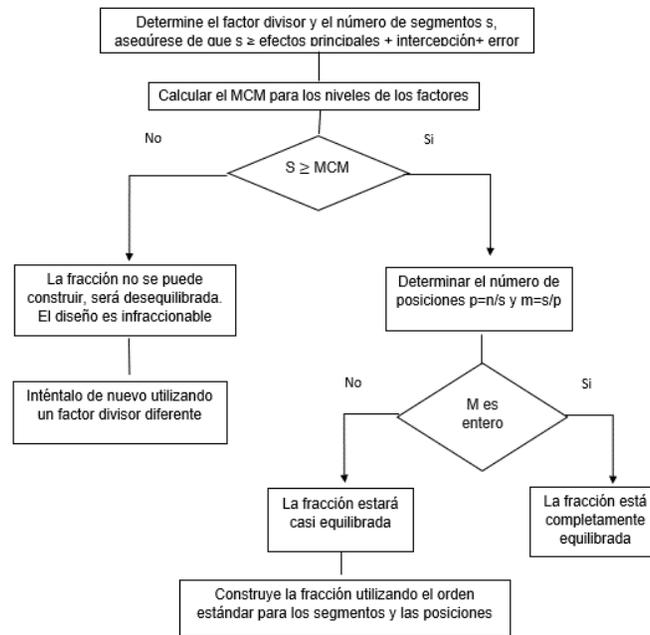
Un diseño de experimentos consiste en planear y realizar un conjunto de pruebas con el objetivo de generar datos que, al ser analizados estadísticamente, proporcione evidencia objetiva que permita responder las siguientes interrogantes planteadas por el experimentador sobre determinada situación [Gutierrez & De la Vara, 2012]. Los diseños experimentales comúnmente se emplean para estudiar sistemas o procesos, Montgomery lo representa como se muestra en la figura 1. El proceso se visualiza como un conjunto de componentes que transforman los materiales (entradas) en una o más variables de respuesta (salidas) y que son observables, en dicho proceso existen factores controlables (x) y no controlables (z) que influyen en la transformación de la variable de respuesta (y) [Montgomery, 2004].



Fuente: Adaptada de Montgomery [2004]

Figura 1 Modelo general de un proceso o sistema.

En el ámbito de la ingeniería los diseños de experimentos se consideran como una herramienta clave para el desarrollo y mejora de procesos de manufactura, en dichos procesos se puede estudiar los efectos que varios factores pueden tener en una respuesta. Al realizar un experimento, variar los niveles de todos los factores al mismo tiempo en lugar de uno a la vez, permite estudiar las interacciones entre los factores, a estos experimentos se les denominan diseños factoriales [Minitab, 2019]. Cuando en un experimento involucra factores de tipo cualitativos y cuantitativos y son empleados a diferentes niveles, el número de corridas experimentales crece dramáticamente. Existen varios métodos para construir diseños factoriales de niveles mixtos (Wang & Wu, 1991; Wang & Wu, 1992; Nguyen, 1996; Wang, 1996; DeCock & Stufken 2000; Xu, 2002; Guo, Simpson & Pignatiello, 2007; Pantoja, Ríos & Tapia, 2018) entre los cuales se basa el comportamiento del algoritmo genérico utilizado para generar las matrices eficientes desarrollado por Guo; se detectó un patrón y este condujo a la generación de un algoritmo capaz de construir fracciones de manera simple. Estas fracciones se denominan matrices balanceadas casi ortogonales (NOBA). El método NOBA demostró que es posible crear fracciones equilibradas casi ortogonales de tamaño económico, reduciendo el número de corridas experimentales considerablemente. El método hace uso de factores divisores y vectores permutados para crear tablas de orden estándar que contienen segmentos y posiciones; los segmentos y posiciones se utilizan para extraer ejecuciones de factorial completo para crear una fracción [Pantoja Y. R., 2018]. En la figura 2 se muestra el algoritmo para la ejecución del método NOBA.



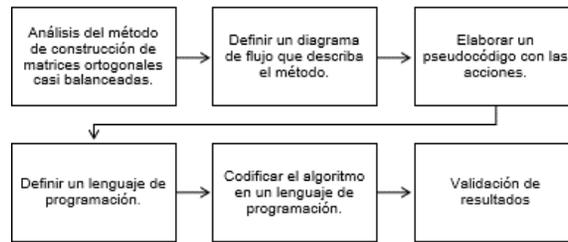
Fuente: Adaptada de Pantoja [2018]

Figura 2 Procedimiento método NOBA

Actualmente el método se ejecuta en hojas de cálculo de Excel, Minitab y Desing Expert, al realizar el método en distintos programas existe la probabilidad de cometer errores de cálculo y al aplicar el método manualmente, considerando también que el tiempo de ejecución es considerablemente largo. Un algoritmo es un conjunto de pasos ordenados y finitos que permiten resolver un problema o una tarea específica. Es importante mencionar que el diseño de un algoritmo es un proceso independiente al lenguaje de programación en el que este vaya a ser codificado [Joyanes, 2008]. El desarrollo de un algoritmo de programación se puede ser representado con un pseudocódigo o un diagrama de flujo, donde se diferencian procesos que se realizan en la computadora. Bajo el enfoque de planeación como propósito de este documento se propone las etapas de la planeación para el desarrollo de un programa que tiene como objetivo facilitar la construcción de diseños fraccionados de niveles mixtos mediante el método NOBA.

2. Métodos

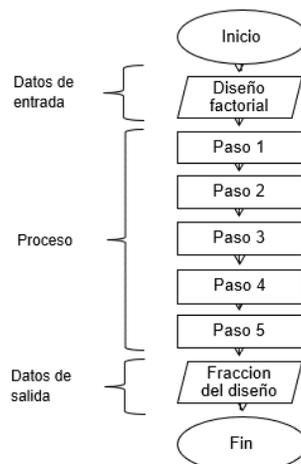
La planeación propuesta en esta investigación consta de seis etapas las cuales se muestran en la figura 3:



Fuente: Elaboración propia

Figura 3 Planeación para la generación del programa

- Etapa 1. Análisis del método de construcción de matrices ortogonales casi balanceadas. En esta primera etapa se estudiarán el funcionamiento de construcción de diseños fraccionados de niveles mixtos mediante el método NOBA, lo cual implica reconocer los datos del diseño factorial, factores, niveles y posibles combinaciones del experimento. Esta etapa involucra el dominio de los factores divisores y la identificación de vectores permutados para poder segmentar y posicionar la fracción del diseño factorial completo.
- Etapa 2. Definir el diagrama de flujo que describa el método. Esta etapa es fundamental, ya que permite definir los procesos para construir la fracción del diseño factorial, el diagrama de flujo tiene como objetivo representar el procedimiento, indicando de manera secuencial los procesos a realizar y la toma de criterios para aplicar el método. La figura 4 muestra el diagrama de flujo que define cada paso del método.



Fuente: Elaboración propia

Figura 4 Diagrama de flujo del método

El diagrama de flujo muestra la secuencia a seguir para el algoritmo, donde se indica la entrada de datos, información que solicitara el programa, diseño factorial completo, número de factores y número de niveles por factor. El procedimiento del método se puede dividir en 5 paso, para el primer paso calcula en número de posibles combinaciones de los factores, el paso 2 consiste expresar los segmentos en los que se puede dividir el diseño factorial. En el paso 3 se calcula el mínimo común múltiplo de los posibles segmentos calculados en el paso anterior, en el paso 4 implica determinar el número de posiciones con ayuda de factores divisores y la tabla de vectores permutados (Tabla1). En el paso 5 con los valores obtenidos de los factores divisores se construye la fracción del diseño factorial completo, y será el dato de salida del programa.

Tabla 1 Vectores permutados

	729	729	729	729	729	729	729	729	729	729	.	.	729	729	729	729	729
	728	728	728	728	728	728	728	728	728	728	.	.	728	728	728	728	728
	727	727	727	727	727	727	727	727	727	727	.	.	727	727	727	727	727

10	10	10	10	10	10	10	10	10	10	10	.	.	10	10	10	10	10
9	9	9	9	9	9	9	9	9	9	9	.	.	9	9	9	9	9
8	8	8	8	8	8	8	8	8	8	8	.	.	8	8	8	8	8
7	7	7	7	7	7	7	7	7	7	7	.	.	7	7	7	7	7
6	6	6	6	6	6	6	6	6	6	6	.	.	6	6	6	6	6
5	5	5	5	5	5	5	5	5	5	5	.	.	5	5	5	5	5
4	4	4	4	4	4	4	3	3	3	3	.	.	1	1	1	1	1
3	3	3	2	2	1	1	4	4	2	2	.	.	3	2	2	4	4
2	2	1	3	1	2	3	2	1	4	4	.	.	4	3	4	2	3
1	1	2	1	3	3	2	1	2	1	1	.	.	2	4	3	3	2
	1	2	3	4	5	6	7	8	9	.	.	.	20	21	22	23	24
	m																

Fuente: Adaptada de Pantoja [2018]

- Etapa 3. Elaborar un pseudocódigo con las instrucciones. La elaboración del pseudocódigo es una forma de expresa las acciones descritas detalladamente, esta herramienta se utiliza para visualizar cada paso y acción. Esta etapa facilita el desarrollo de código ya el traspaso de información del diagrama de flujo al código de programación.
- Etapa 4. Definir el lenguaje de programación. El software que frecuentemente es usado para para resolver problemas de ingeniería y ciencias es MATLAB, existen otros softwares capaces de resolver estos problemas como C++, FORTAN, PYTHON entre otros, siendo MATLAB el

óptimo para trabajo con matrices, vectores y cuando el tamaño y la complejidad de las operaciones crece.

- Etapa 5. Codificar el algoritmo en un lenguaje de programación. Al haber definido los pasos para el desarrollo del algoritmo, los procesos del pseudocódigo elaborado en la etapa 3, se reescribirán en el lenguaje de programación de MATLAB. Siendo la programación modular la adecuada para codificar los procesos del método, esta técnica permite comparar, depurar y combinar los módulos entre sí, facilitando la programación.
- Etapa 6. Validación de resultados. Para el procedimiento de validación de resultados se utilizará los diseños factoriales mixtos comparados en [Pantoja Y. R., 2018], realizando la construcción de las fracciones de los diseños factoriales manualmente y comparando con las fracciones generadas por el programa, detectando posibles errores en la programación para corregir y validar el programa nuevamente.

3. Resultados

Únicamente se muestra la planeación para el desarrollo de un programa para construir diseños factoriales fraccionados mediante el método NOBA:

- La primera etapa permitió el análisis detallado del método para construir matrices balanceadas casi ortogonales y así desarrollar una estrategia para elaborar un algoritmo para la ejecución de este.
- La segunda etapa segunda etapa sirvió para el desarrollo del diagrama de flujo, dividido en tres secciones, la primera sección donde se solicita los datos de entrada correspondientes al diseño factorial completo, la segunda sección donde se realizan los procedimientos para segmentar y posicionar los datos, y la tercera sección donde se genera la fracción del diseño factorial correspondiente a los datos de salida.

En la figura 5 se muestra un ejemplo ($3^16^17^1$ Diseño factorial) de la interfaz del programa, para introducir las variables de entrada. La figura 6 muestra la interfaz del programa con las posibles combinaciones del ejemplo, dando un total de 126 corridas experimentales.

- La tercera etapa permitió el desarrollo del pseudocódigo, con esta técnica se describió detalladamente los procesos para facilitar la programación del código, posteriormente en la cuarta etapa se encontró la mejor alternativa para desarrollar el código, debido a su capacidad para resolver problemas complejos de matrices y vectores fue MATLAB, también se encontró que la programación modular es la óptima para generar, editar y detectar errores. La figura 7 muestra la fracción del diseño factorial del ejemplo, reduciendo de 126 a 42 corridas experimentales.

```
>> Ejemplo
Teclee el numero de factores    3
Teclee los niveles para cada factor
3
6
7
```

Fuente: Elaboración propia

Figura 5 Interfaz del programa para ingresar variables de entrada, diseño $3^16^17^1$

1	1	1	3	5	2	2	4	4	1	3	6
2	1	1	1	6	2	3	4	4	2	3	6
3	1	1	2	6	2	1	5	4	3	3	6
1	2	1	3	6	2	2	5	4	1	4	6
2	2	1	1	1	3	3	5	4	2	4	6
3	2	1	2	1	3	1	6	4	3	4	6
1	3	1	3	1	3	2	6	4	1	5	6
2	3	1	1	2	3	3	6	4	2	5	6
3	3	1	2	2	3	1	1	5	3	5	6
1	4	1	3	2	3	2	1	5	1	6	6
2	4	1	1	3	3	3	1	5	2	6	6
3	4	1	2	3	3	1	2	5	3	6	6
1	5	1	3	3	3	2	2	5	1	1	7
2	5	1	1	4	3	3	2	5	2	1	7
3	5	1	2	4	3	1	3	5	3	1	7
1	6	1	3	4	3	2	3	5	1	2	7
2	6	1	1	5	3	3	3	5	2	2	7
3	6	1	2	5	3	1	4	5	3	2	7
1	1	2	3	5	3	2	4	5	1	3	7
2	1	2	1	6	3	3	4	5	2	3	7
3	1	2	2	6	3	1	5	5	3	3	7
1	2	2	3	6	3	2	5	5	1	4	7
2	2	2	1	1	4	3	5	5	2	4	7
3	2	2	2	1	4	1	6	5	3	4	7
1	3	2	3	1	4	2	6	5	1	5	7
2	3	2	1	2	4	3	6	5	2	5	7
3	3	2	2	2	4	1	1	6	3	5	7
1	4	2	3	2	4	2	1	6	1	6	7
2	4	2	1	3	4	3	1	6	2	6	7
3	4	2	2	3	4	1	2	6	3	6	7
1	5	2	3	3	4	2	2	6			
2	5	2	1	4	4	3	2	6			

Fuente: Elaboración propia

Figura 6 Posibles combinaciones de los factores, diseño $3^16^17^1$

3	1	1	3	3	3	1	5	5
2	2	1	1	4	3	1	1	6
1	3	1	3	5	3	1	1	6
3	4	1	2	6	3	2	2	6
1	5	1	1	2	4	1	4	6
2	6	1	2	2	4	1	4	6
2	1	2	1	3	4	1	6	6
3	2	2	1	5	4	2	6	6
1	3	2	1	5	4	3	1	7
2	4	2	2	6	4	1	3	7
1	5	2	2	1	5	1	3	7
3	6	2	1	3	5	3	4	7
1	1	3	1	3	5	1	5	7
2	2	3	2	4	5	3	6	7

Fuente: Elaboración propia

Figura 7 Fracción del diseño $3^1 6^1 7^1$

Una vez en funcionamiento correcto el programa se comparará las fracciones construidas manualmente en [Pantoja Y. R., 2018], fraccionando diseños factoriales desde 3 a 9 factores mixtos. Se validará el programa siendo las mismas fracciones construidas, así como también se llevará a la práctica con diseños de experimentos de niveles mixtos de un proceso real.

4. Discusión

Los diseños factoriales con factores cualitativos, cuantitativos y niveles mixtos tienen un número alto de corridas experimentales lo que se traduce como un gasto de recursos, la alternativa para para minimizar el empleo de recursos es segmentar el diseño factorial para solo realizar una fracción del diseño. El método NOBA es un método para construir diseños fraccionados de niveles mixtos, el método reduce el número de corridas experimentales. La planeación mostrada en este documento describe las etapas para desarrollar el programa, con el análisis del método, se pudo definir un diagrama de flujo, un pseudocódigo y posteriormente en el software MATLAB realizar la codificación del algoritmo para validar el código haciendo una comparación de las fracciones obtenidas manualmente y la fracción obtenida con el programa, cabe destacar que el tiempo de ejecución del método y la posibilidad de cometer un error humano se reduce significativamente. Actualmente no existe un programa que ejecute el método NOBA, por ello existe la necesidad de generar la herramienta que facilite la construcción de diseños fraccionados de niveles mixtos.

Se puede concluir que, al ejecutar el método NOBA con un programa se puede reducir considerablemente el tiempo requerido para ejecutar el método para la construcción de diseños factoriales de niveles mixtos, así como reducir la posibilidad de errores de cálculo al aplicar el método manualmente. Una vez validado el programa facilitara al experimentador la creación de tablas de orden estándar que contengan segmentos y posiciones para la creación de las fracciones del diseño factorial completo, sin requerimientos de un dominio de estadística, matemáticas o programación, así como equipos o software costosos.

6. Bibliografía y Referencias

- [1] Ataurima Arellano, M. (2013). *Matlab & Simulink para Ingeniería Nivel I*. Lima: Universidad de. Ciencias y Humanidades.
- [2] Corona, M. A., & Ancona, M. Á. (2011). *Diseño de algoritmos y su codificación en lenguaje C*. México: McGraw-Hill/Interamericana Editores, SA de CV.
- [3] De Burgos, J. (2006). *Algebra lineal y geometria cartesiana*. Ciudad de Mexico: Mc Graw Hill.
- [4] Deitel, H., & Deitel, P. (2003). *Como programar en C++*. México: Pearson Educación de México. Fernandez, S. (2020). *Diseño experimentos, Diseño factorial*. Tesis de Maestria. Catalunya: Universidad Politecnica de Catalunya.
- [5] García, J., Rodríguez, J., & Brazález, A. (2001). *Aprenda Matlab 6.1 como si estuviera en primero*. Madrid: Universidad Politécnica de Madrid.
- [6] Goicoechea, H. (2017). *Diseño experimental*. <https://www.fccb.unl.edu.ar/laboratorios/ladaq/wp-content/uploads/2016/06/2-Selecci%C3%B3n-factores-2017-pdf.pdf>.
- [7] Gutierrez, H., & De la Vara, R. (2012). *Analisis y diseño de experimentos*. México: The McGraw-Hill Companies, Inc.
- [8] Joyanes, L. (2008). *Fundamentos de programación*. Madrid: McGrawHill/Interamericana de España, SAU.
- [9] Minitab. (2019). *¿Qué es la resolución de diseño en un diseño factorial?* <https://support.minitab.com/es-mx/minitab/18/help-and-how-to/modeling->

statistics/doe/supporting-topics/factorial-and-screening-designs/what-is-design-resolution/

- [10] Minitab. (2019). Diseños factoriales y factoriales fraccionados. <https://support.minitab.com/es-mx/minitab/18/help-and-how-to/modeling-statistics/doe/supporting-topics/factorial-and-screening-designs/factorial-and-fractional-factorial-designs/>.
- [11] Montgomery. (2004). Diseño y análisis de experimentos. Ciudad de Mexico: Limusa Wiley.
- [12] Moore, H. (2007). MATLAB para Ingenieros. México: Pearson Educación México.
- [13] Pantoja, Y. R. (2018). A method for construction of mixed-level fractional designs. Memorias del Congreso Internacional de Investigación Academia Journals Celaya, 2712-2875. doi:ISSN 1946-5351
- [14] Pantoja, Y., Rios, A., Perez, L., Hernandez, D., & Tapia, M. (2019). Aplicacion del metodo NOBA para construccion de fracciones de diseños de niveles mixtos. Memorias del Congreso Internacional de Investigación Academia Journals Celaya 2019, (págs. 2712-2875). Celaya.
- [15] Psicometodos. (2019). Balanceo y contrabalanceo. <https://www.psicometodos.com/?p=1846>.
- [16] Salmoran, E. (2020). Programación de un algoritmo para la construcción de estructuras de alias de diseños factoriales fraccionados de niveles mixtos. Tesis de Maestria. Celaya: Tecnológico Nacional de México en Celaya.
- [17] Weisstein, E. (2009). CRC Encyclopedia of Mathematics. California: CR Press.