



**Universitat Politècnica de Catalunya**

# **Órdenes de experimentación en diseños factoriales**

**Tesis Doctoral**

**Presentada por**  
Alexander Alberto Correa Espinal

**Director**  
Dr. Pere Grima Cintas



A mi esposa, Ángela María por su amor, apoyo y sacrificio en esta dura tarea  
de ver realizado un sueño.

A mis padres, Teresa e Iván por su motivación constante.

A Colombia, tierra querida.



# Índice General

Índice de Tablas .....	v
Índice de Figuras .....	vii
Prólogo .....	xi
Resumen .....	xiii
<b>1. Planteamiento del problema .....</b>	<b>3</b>
1.1 Diseños factoriales a 2 niveles .....	3
1.2 Orden de experimentación .....	7
1.3 Hipótesis sobre la influencia de factores desconocidos .....	8
1.4 Medida de la influencia de los factores desconocidos .....	8
1.5 Aleatorización y neutralización de la influencia de los factores desconocidos .....	10
1.6 Número de cambios de nivel de los factores .....	12
1.7 Objetivos de la tesis .....	15
<b>2. Antecedentes y estado del arte .....</b>	<b>19</b>
2.1 Búsqueda de ordenaciones con el mínimo sesgo .....	20
2.1.1 Cox (1952) .....	20
2.1.2 Box (1952) y Box Hay (1953) .....	21
2.1.3 Hill (1960) .....	22
2.1.4 Daniel y Wilcoxon (1966) .....	23
2.1.5 Cheng y Jacroux (1988) .....	24
2.1.6 John (1990) .....	27
2.1.7 Cheng y Steinberg (1991) .....	28
2.1.8 Bayle <i>et al.</i> (1992) .....	28
2.1.9 Atkinson y Donev (1996) .....	28
2.2 Búsqueda de ordenaciones con el mínimo número de cambios de nivel ..	29

2.2.1	Jan y Wang (1995) .....	29
2.2.2	Cheng <i>et al.</i> (1998) .....	33
2.2.3	Chen y Wang (2001) .....	34
2.3	Búsqueda de ordenaciones que consideran simultáneamente el número de cambios y el sesgo .....	34
2.3.1	Cheng (1985) .....	34
2.3.2	Coster y Cheng (1988) .....	35
2.3.3	Coster (1993) .....	36
2.3.4	Jan y Wang (1995) .....	37
2.3.5	Kim (1997) .....	38
2.3.6	Tack y Vandebroek (2002) .....	39
2.3.7	Dickinson (1974) .....	39
2.3.8	De León (2004) y De León <i>et al.</i> (2005) .....	44
2.3.9	Draper y Stoneman (1968) .....	45
2.3.0	Joiner y Campbell (1976) .....	50
2.4	Resumen .....	52
2.5	Algunas debilidades detectadas .....	55
2.6	Posibles aportes .....	56
<b>3.</b>	<b>Generación de órdenes de experimentación con mínimo número de cambios y sesgo conocido</b> .....	<b>59</b>
3.1	Introducción .....	59
3.2	Obtención de los órdenes de experimentación para diseños $2^4$ .....	60
3.3	Propiedades de la matriz obtenida .....	62
3.3.1	La matriz obtenida es una matriz completa .....	62
3.3.2	La matriz obtenida tiene el mínimo número de cambios posible .....	63
3.3.3	La matriz obtenida permite estimar los efectos con un sesgo conocido .....	64
3.3.4	Resumen de las características del diseño obtenido .....	72
3.4	Aportaciones .....	72
<b>4.</b>	<b>Búsqueda de órdenes con mínimo sesgo y mínimo número de cambios</b> ...	<b>75</b>
4.1	Búsqueda totalmente aleatoria .....	76
4.2	Búsqueda aleatoria con restricciones .....	76
4.2.1	Restricciones en las filas sucesoras .....	77
4.2.2	Algoritmo para la búsqueda de órdenes con sesgo mínimo .....	79
4.2.3	Programa en Free Basic .....	81
4.2.4	Resultados obtenidos .....	81

4.3	Búsqueda exhaustiva de órdenes semilla .....	85
4.3.1	Algoritmo de Dickinson .....	85
4.3.2	Implementación del algoritmo de Dickinson .....	92
4.3.3	Resultados obtenidos .....	93
4.4	Algoritmo de Dickinson con alimentación forzada .....	96
4.4.1	Alimentación y resultados obtenidos para diseños con 64 experimentos .....	96
4.4.2	Alimentación y resultados obtenidos para diseños con 128 experimentos .....	99
4.5	Resumen. Mejores sesgos encontrados .....	100
4.6	Aportaciones .....	102
<b>5.</b>	<b>Obtención de nuevos órdenes a partir de los órdenes semilla .....</b>	<b>105</b>
5.1	Método de Permutación y cambios de signo .....	106
5.2	Método de la columna de expansión .....	109
5.2.1	Paso de un diseño $2^2$ a otro $2^3$ .....	109
5.2.2	Generalización del procedimiento descrito .....	115
5.2.3	Ventajas del método de la columna de expansión .....	118
5.3	Implementación en MINITAB de los métodos propuesto .....	121
5.3.1	Implementación del método de las permutaciones y cambios de signo .....	122
5.3.2	Implementación del método de las columnas de expansión .....	127
5.4	Aportaciones .....	130
<b>6.</b>	<b>Diseños con más de 128 experimentos .....</b>	<b>133</b>
6.1	Comparación de los sesgos máximos para diseños con distinto número de experimentos .....	134
6.2	Nueva medida de la influencia de la tendencia: Coeficiente de correlación entre la respuesta y el efecto .....	135
6.3	Coeficientes de correlación máximo entre factores y respuesta para los diseños con las propiedades deseadas .....	137
6.4	Aportaciones .....	138
<b>7.</b>	<b>Resultados obtenidos y futuras líneas de investigación .....</b>	<b>141</b>
7.1	Resultados obtenidos .....	141
7.2	Futuras líneas de investigación .....	145
<b>Anexos</b>		
A.1	Órdenes de experimentación para diseños con 8 y 16 experimentos propuestos por De León (2004) .....	147

A.2	Uso del lenguaje de programación <i>FreeBasic</i> .....	149
A.3	Programa en <i>FreeBasic</i> para generar órdenes con el mínimo número de cambios de nivel en un diseño $2^6$ .....	153
A.4	Programa en <i>FreeBasic</i> para la búsqueda exhaustiva de órdenes semilla con el mínimo número de cambios en diseños $2^5$ utilizando el algoritmo de Dickinson .....	159
A.5	Orden con las propiedades deseadas para un diseño $2^7$ obtenido con el algoritmo de Dickinson con alimentación forzada .....	167
A.6	Macro en el lenguaje de Minitab para obtener órdenes con las propiedades deseadas usando el método permutación y cambios de signo	169
A.7	Macro en el lenguaje de Minitab para obtener órdenes con las propiedades deseadas usando el método de las columnas de expansión ..	179
A.8	Algunas consideraciones sobre el análisis de los órdenes propuestos	185
	Referencias .....	197

# Índice de tablas

Tabla 1.1	Valores del sesgo máximo en un diseño $2^3$ .....	10
Tabla 1.2	Valores del sesgo máximo en un diseño $2^4$ .....	11
Tabla 1.3	Número de cambios de nivel en dos órdenes de ejecución de un diseño $2^3$ .....	12
Tabla 4.1	Posibles filas sucesoras para tener mínimo número de cambios en un diseño $2^5$ .....	78
Tabla 4.2	Único orden obtenido que proporciona el mínimo sesgo hallado con mínimo número de cambios para un diseño con 64 experimentos ...	83
Tabla 4.3	Resumen de sesgos obtenidos .....	84
Tabla 4.4	Posibles sucesores de cada una de las filas en un diseño $2^3$ .....	87
Tabla 4.5	Órdenes semilla con el mínimo número de cambios de nivel y el mínimo sesgo máximo en diseños con 32 experimentos hallados por búsqueda exhaustiva .....	93
Tabla 4.6	Número de órdenes semilla con las propiedades deseadas para diseños con 32 experimentos .....	94
Tabla 4.7	Orden inicial para el algoritmo de Dickinson en diseños con 64 experimentos .....	97
Tabla 4.8	Órdenes obtenidos que proporcionan el mínimo sesgo hallado con el algoritmo de Dickinson con alimentación forzada para un diseño con 64 experimentos .....	99
Tabla 4.9	Uno de los órdenes que proporciona el mínimo sesgo con el mínimo número de cambios para un diseño con 128 experimentos hallado con el algoritmo de Dickinson con alimentación forzada .....	100
Tabla 4.10	Mejores órdenes hallados para cada tipo de diseño y procedimiento por el que han sido encontrados .....	101
Tabla 5.1	Órdenes semilla obtenidos .....	106

Tabla 5.2	Matriz de diseño e interacciones que cumplen las propiedades deseadas .....	108
Tabla 5.3	Matriz de diseño deducida de la anterior cambiando de orden las columnas A y B, y cambiado los signos de C. Puede comprobarse que las propiedades de número de cambios y mínimo sesgo máximo absoluto se mantienen .....	109
Tabla 5.4	Órdenes de experimentación en el diseño $2^2$ que presentan el mínimo número de cambios (todos los órdenes de este diseño tienen el mismo msma) .....	109
Tabla 5.5	Columnas de expansión para pasar de diseños $2^2$ a diseños $2^3$ . El encabezado de cada columna indica que factor se coloca en último lugar, y si los órdenes del diseño $2^2$ se asignan a los signos positivos o negativos de este último factor .....	113
Tabla 5.6	Órdenes de experimentación en un diseño $2^3$ con las propiedades deseadas generados a partir de los 8 órdenes que reúnen esas propiedades en un diseño $2^2$ y las 6 columnas de expansión .....	114
Tabla 5.7	Columnas de expansión deducidas a partir de la matriz de la Figura 5.6. El título de la columna indica el factor y el signo al que se asignan a primera mitad de los experimentos .....	118
Tabla 6.1	Sesgos máximos que se deducen con el método de duplicación y finalmente hallados para cada número de experimentos .....	133
Tabla 6.2	Coefficiente de correlación $\rho_{E-R}$ para distintos valores del número de experimentos .....	137

# Índice de figuras

Figura 1.1	Matriz de diseño de un diseño factorial $2^3$ en el orden estándar . . . . .	4
Figura 1.2	Matriz ampliada de un diseño $2^3$ . . . . .	5
Figura 1.3	Matriz de diseño en orden estándar . . . . .	7
Figura 1.4	Matriz ampliada y valores de los efectos para un diseño $2^3$ en el que la respuesta es igual al orden de ejecución de los experimentos . . . . .	9
Figura 1.5	Matriz de un diseño $2^{5-2}$ con generadores $D=AB$ y $E=AC$ . . . . .	13
Figura 1.6	Número de cambios en los factores de un diseño $2^{7-4}$ . . . . .	14
Figura 2.1	Plan experimental equivalente al propuesto por Hill (1960) para eliminar la influencia de factores ajenos a la experimentación, presentado por John (1990) . . . . .	22
Figura 2.2	Selección del orden de experimentación para un diseño factorial a dos niveles con el método propuesto por Daniel y Wilcoxon (1966) .	23
Figura 2.3	Orden de experimentación para un diseño factorial a dos niveles obtenido con el método propuesto por Daniel y Wilcoxon (1966) . . .	24
Figura 2.4	Diseño factorial $2^4$ con el orden estándar . . . . .	26
Figura 2.5	Orden de experimentación de un diseño factorial $2^4$ con los factores principales y algunas interacciones dobles libres de sesgo, hallados con el algoritmo de Cheng y Jacroux (1988) . . . . .	27
Figura 2.6	Diseño factorial $2^4$ con el orden estándar y con las columnas en orden creciente del número de cambios . . . . .	30
Figura 2.7	Arreglo ortogonal $AO_{16}(2^{15})$ para el diseño factorial $2^4$ con las columnas ordenadas de menor a mayor número de cambios de nivel .	31
Figura 2.8	Diseño factorial $2^4$ con el mínimo número de cambios de nivel de los factores hallado con el algoritmo de Wang y Jan (1995) . . . . .	32
Figura 2.9	Orden para un diseño $2^{4-0}$ , hallado por el método foldover generalizado propuesto por Coster y Cheng (1988) . . . . .	37
Figura 2.10	Diseño factorial $2^4$ con los efectos principales e interacciones dobles	

	libres de sesgo, hallado con el algoritmo de Wang y Jan (1995) . . . .	38
Figura 2.11	Evaluación del diseño $2^4$ con el orden estándar en cuanto a número de cambios de nivel y sesgo máximo en los efectos principales . . . . .	41
Figura 2.12	Órdenes de experimentación en el diseño factorial $2^4$ con el sesgo máximo en los efectos principales y con 15 cambios de nivel de los factores . . . . .	42
Figura 2.13	Órdenes de experimentación en el diseño factorial $2^5$ con el mínimo sesgo máximo en los efectos principales y con 31 cambios de nivel de los factores . . . . .	43
Figura 2.14	Órdenes de experimentación con pocos cambios de nivel y bajo sesgo en el diseño $2^3$ hallados por Draper Stoneman (1968) . . . . .	47
Figura 2.15	Número de cambios de nivel y sesgo máximo en efectos principales, en uno de los órdenes del diseño $2^3$ , hallados por Draper y Stoneman (1968) . . . . .	47
Figura 2.16	Órdenes de experimentación con pocos cambios de nivel y bajo sesgo en el diseño $2^{4-1}$ hallados por Draper y Stoneman (1968) . . . . .	48
Figura 2.17	Órdenes de experimentación con pocos cambios de nivel y bajo sesgo en el diseño $2^{5-2}$ hallados por Draper y Stoneman (1968) . . . . .	48
Figura 2.18	Órdenes de experimentación con pocos cambios de nivel y bajo sesgo en el diseño $2^{6-3}$ hallados por Draper y Stoneman (1968) . . . . .	49
Figura 2.19	Órdenes de experimentación con pocos cambios de nivel y bajo sesgo en el diseño $2^{7-4}$ hallados por Draper y Stoneman (1968) . . . . .	49
Figura 2.20	Diagrama Bivariante para evaluar 50 órdenes de experimentación en los diseños factoriales $2^3$ obtenido con el método propuesto por Joiner y Campbell (1976) . . . . .	52
Figura 2.21	Estado del arte y antecedentes . . . . .	54
Figura 3.1	Diseño $2^3$ con un orden de ejecución que cumple con la propiedad de mínimo número de cambios en los factores y mínimo sesgo máximo (igual a 2) . . . . .	60
Figura 3.2	Matriz de diseño de la Figura 3.1 con las filas duplicadas . . . . .	61
Figura 3.3	Diseño $2^4$ con un orden de ejecución que presenta las propiedades deseadas . . . . .	62
Figura 3.4	Cambios de signo en el factor añadido al pasar de un diseño $2^3$ a un $2^4$ . . . . .	63
Figura 3.5	Cálculo de los efectos en el diseño $2^3$ de partida . . . . .	64
Figura 3.6	Cálculo de los efectos en el diseño $2^4$ obtenido . . . . .	65
Figura 3.7	Time count para los efectos del diseño $2^3$ de partida y el diseño $2^4$ obtenido . . . . .	66
Figura 3.8	Otra forma de escribir las time count correspondientes al diseño $2^4$ . . . . .	66
Figura 3.9	Generalización del cálculo del time count para el diseño obtenido (1) . . . . .	67

Figura 3.10	Generalización del cálculo del time count para el diseño obtenido (2)	68
Figura 3.11	Valores del time count en las dobles parejas centrales del factor añadido .....	68
Figura 3.12	Sumandos del time count para los efectos X (cualquier efecto de los que aparecen en el diseño de partida) y X•D(1) .....	70
Figura 3.13	Sumandos del time count para los efectos X (cualquier efecto de los que aparecen en el diseño de partida) y X•D(2) .....	70
Figura 3.14	Suma para cada pareja según s sea positivo o negativo .....	71
Figura 3.15	Resumen de las propiedades del diseño obtenido aplicando la metodología propuesta .....	72
Figura 4.1	Posibles filas sucesoras a la número 10 para tener sólo un cambio en una y otra .....	77
Figura 4.2	Diagrama de flujo del algoritmo de búsqueda aleatoria con restricciones .....	80
Figura 4.3	Diagrama de barras del valor de los sesgos obtenidos al generar aleatoriamente 50 millones de órdenes con el mínimo número de cambios en un diseño con 32 experimentos .....	81
Figura 4.4	Diagrama de barras del valor de los sesgos obtenidos al generar aleatoriamente 10 millones de órdenes con mínimo número de cambios en diseños con 64 experimentos .....	81
Figura 4.5	Diagrama de barras del valor de los sesgos obtenidos al generar aleatoriamente 1000 órdenes con el mínimo número de cambios en un diseño con 128 experimentos .....	84
Figura 4.6	Secuencia de órdenes a que obliga el algoritmo de Dickinson en un diseño $2^5$ .....	86
Figura 4.7	Algoritmo de Dickinson para los diseños factoriales a dos niveles con $2^k$ experimentos .....	88
Figura 4.8	Selección de números de fila hasta completar la primera ordenación (de arriba abajo) .....	91
Figura 4.9	Tras completar la primera ordenación, retrocede hasta la posición 6, cambia el sucesor y sigue .....	91
Figura 4.10	Nuevo orden cambiando el sucesor en la posición de la fila 3. No se puede elegir el sucesor fila 5 en la posición de la fila 4 por restricción impuesta en el orden de las filas que tienen un solo signo positivo ..	92
Figura 4.11	Diagrama de barras del valor de los sesgos obtenidos para todos los órdenes semilla con el mínimo número de cambios de nivel para los diseños con 32 experimentos .....	95
Figura 4.12	Posibles sucesores de las filas 2 a la 10 colocados en orden creciente (el utilizado en el algoritmo estándar) .....	97
Figura 4.13	Orden de colocación de los sucesores de cada fila .....	98

Figura 5.1	Las matrices tienen cambiadas las columnas correspondientes a los factores B y C, pero el número total de cambios se mantiene constante .....	107
Figura 5.2	Diseño $2^3$ que cumple con las propiedades deseadas visto como un $2^2$ replicado .....	110
Figura 5.3	Columna de expansión para pasar del diseño $2^2$ al diseño $2^3$ .....	111
Figura 5.4	Aplicación de una columna de expansión a todos los órdenes elegidos del diseño $2^2$ .....	112
Figura 5.5	Columna de expansión complementaria a la que aparece en la Figura 5.3 .....	113
Figura 5.6	Matriz de un diseño $2^4$ en un orden que cumple con las propiedades deseadas .....	115
Figura 5.7	Columna de expansión para pasar de un diseño $2^3$ a otro $2^4$ .....	116
Figura 5.8	Diseño de partida y diseño obtenido al aplicar el método descrito en el capítulo 3 .....	119
Figura 5.9	Columna de expansión que aplica el método descrito en el capítulo 3 .....	120
Figura 5.10	Diseños factoriales disponibles en Minitab .....	121
Figura 5.11	Ventana de diálogo del botón “Opciones” .....	122
Figura 5.12	Hoja de datos con la matriz de diseño tal como la presenta el Minitab, sin aleatorizar .....	123
Figura 5.13	Lanzamiento de la macro para obtener un orden de experimentación con las propiedades deseadas .....	123
Figura 5.14	Orden de experimentación con las propiedades deseadas .....	124
Figura 5.15	Diagrama de flujo del método de permutación y cambios de signo ..	126
Figura 5.16	Diagrama de flujo del método de la columna de expansión .....	129
Figura 6.1	Efectos obtenidos en diseños con 32, 64 y 128 experimentos con las propiedades deseadas y en los que la respuesta es igual al orden de ejecución de los experimentos .....	134

# Prólogo

Cuando se plantea un diseño factorial, en general se recomienda que los experimentos se realicen en orden aleatorio. Esta aleatoriedad tiene como objetivo el protegernos de la posible influencia de factores desconocidos, ya que se espera que esa influencia quede difuminada entre todos los efectos de forma que ninguno se vea especialmente afectado y no se cometan errores al valorar su significación estadística.

Pero este proceder tiene dos inconvenientes importantes:

1. El número de cambios de nivel en los factores que exige la aleatorización puede ser grande (bastante mayor que otras posibles ordenaciones) y difícil de llevar a la práctica, lo que complica y encarece la experimentación.
2. Si se realizan unas hipótesis que parecen muy razonables respecto al tipo de influencia que ejercen los factores desconocidos, existen órdenes claramente mejores que otros para minimizar la influencia de esos factores ajenos a la experimentación.

Numerosos autores han estado trabajando sobre este problema, y ya han sido resueltos algunos aspectos, como el de determinar los órdenes de experimentación que mejor eliminan la influencia de factores desconocidos, aunque sin tener en cuenta el número de cambios en los niveles de los factores que esa ordenación implica. Adicionalmente, se ha resuelto el problema de encontrar órdenes que presentan el mínimo número de cambios de nivel, pero sin considerar de forma simultánea la posible influencia de los factores desconocidos.

Cuando se considera conjuntamente la influencia de los factores desconocidos y el número de cambios en los factores, se ha llegado hasta los diseños con 16 experimentos, pero no más allá porque los procedimientos de búsqueda empleados son inviables

cuando el número de ordenaciones posibles se hace tan grande (con 32 experimentos el número de ordenaciones posibles es  $32! = 2,6 \cdot 10^{35}$ )

La tesis que se presenta tiene por objetivo encontrar un procedimiento que permita obtener órdenes de experimentación con el mínimo número de cambios en los factores y que además minimicen la influencia de los factores desconocidos en la estimación de los efectos para cualquier diseño factorial. Además, se pretende elaborar un procedimiento de forma que dichas ordenaciones, con las propiedades deseadas, puedan ser obtenidas de forma fácil por el experimentador.

## **AGRADECIMIENTOS**

El autor desea presentar los más sinceros agradecimientos:

Al Doctor Pere Grima Cintas, profesor titular de la Universitat Politècnica de Catalunya y director de este trabajo, ya que gracias a él fue posible la adecuada definición del tema investigación y a su valiosa orientación que permitió llevar a buen término este proyecto.

A la Universidad Nacional de Colombia, por apoyar mi proceso de formación doctoral a través de la comisión de estudios.

A la Fundación Carolina de España por su beca de formación de docentes doctores.

A la Universitat Politècnica de Catalunya por brindarme la oportunidad de adelantar mi doctorado en aplicaciones técnicas e informáticas de la estadística, la investigación operativa y la optimización.

Al grupo de profesores y personal administrativo del Departamento de Estadística e Investigación Operativa de la planta 6 del ETSEIB, por brindarme un excelente clima de trabajo y apoyo para el adecuado desarrollo de esta tesis doctoral.

Y a todas aquellas personas que de una u otra forma aportaron a la realización de esta tesis doctoral.



# Resumen

Esta tesis doctoral se compone de 7 capítulos y 8 anexos en relación a los órdenes de ejecución en diseños factoriales a dos niveles para que presenten el mínimo número de cambios de nivel en los factores y que atenúen la influencia en la estimación de los efectos de factores ajenos a la experimentación.

En el capítulo 1, se presenta una breve descripción de los diseños factoriales a dos niveles, se plantea la problemática que implica la aleatorización de los órdenes de experimentación y se presenta la posibilidad de utilizar órdenes sistemáticos que permitan conseguir los objetivos planteados.

En el capítulo 2, se realiza una descripción general de los trabajos que han estudiado el orden de ejecución de los experimentos y su repercusión en el número de cambios de los factores y/o la eliminación de la influencia de factores ajenos a la experimentación. Del estudio de las diferentes estrategias y algoritmos utilizados, se definen las debilidades de las propuestas realizadas y los posibles aportes que pueden ser realizados en esta tesis doctoral.

En el capítulo 3, se presenta la construcción de un nuevo método de búsqueda eficiente de órdenes con el mínimo número de cambios de nivel de los factores y que a su vez atenúan la influencia de factores ajenos a la experimentación. A este procedimiento lo denominamos método de duplicación, ya que permite pasar de un diseño factorial  $2^k$  a un diseño  $2^{k+1}$  que conserve las características del anterior, al duplicar las filas del diseño  $2^k$  y al agregar un factor adicional con una determinada secuencia de signos.

En el capítulo 4, se utilizan diferentes métodos de búsqueda para hallar órdenes de experimentación con el mínimo número de cambios de nivel y mínimo sesgo máximo, tales como el algoritmo de Dickinson, la búsqueda aleatoria restringida, el método de duplicación y el algoritmo de Dickinson con alimentación forzada, con lo cual se logra obtener órdenes con un mínimo sesgo máximo en los diseños factoriales con 32, 64, 128 experimentos que no se conocían.

En el capítulo 5, se presentan dos métodos para obtener órdenes con las características deseadas a partir de los órdenes semilla identificados en el capítulo 4. Estos métodos los denominamos método de permutación y cambios de signo y el método de las columnas de expansión. Ambos métodos fueron programados en macros de Minitab, lo cual permite generar de forma automática y aleatoria los órdenes con las características propuestas.

En el capítulo 6, se presenta una nueva medida de atenuación de la influencia de los factores ajenos a la experimentación en los diseños factoriales con el mínimo número de cambios de nivel, la cual permite comparar la atenuación entre diseños factoriales con diferente número de factores, mostrando que el procedimiento de duplicación del capítulo 3, es adecuado para obtener órdenes de experimentación con las características propuestas en diseños con más de 128 experimentos.

Para finalizar, en el capítulo 7 se presentan las principales conclusiones obtenidas con la realización de esta tesis doctoral y se definen posibles futuras líneas de investigación que podrían ampliar los estudios realizados.

# Capítulo 1



# Capítulo 1

## Planteamiento del problema

Este capítulo tiene como propósito presentar las motivaciones que se consideraron para seleccionar este tema de investigación y definir claramente los elementos fundamentales que se presentan a lo largo del trabajo, tales como los diseños factoriales a dos niveles completos o fraccionales, los problemas que puede causar la aleatorización en estos diseños, y cómo cuantificar la influencia de los factores ajenos a la experimentación en la estimación de los efectos. Asimismo, se plantean las hipótesis y el contexto en que se buscarán los órdenes de ejecución que presentan las propiedades deseadas y, finalmente, se describe el objetivo de la tesis.

### 1.1 Diseños factoriales a dos niveles

Uno de los diseños experimentales más utilizados en la práctica industrial para estudiar el efecto de varios factores (variables de interés) sobre una respuesta (característica estudiada), son los diseños factoriales a dos niveles (valores que puede tomar cada factor), también conocidos como diseños factoriales  $2^k$ .

Estos diseños se caracterizan por plantear experimentos en todas las combinaciones de valores de los factores, y como cada factor toma 2 valores distintos, si el número de factores es  $k$ , el número total de experimentos a realizar será  $2^k$ . Por ejemplo, un diseño factorial  $2^3$  (con 3 factores) requiere la realización de 8 pruebas experimentales, que denominamos “experimentos” o, en Latinoamérica, “corridas”.

Los dos niveles de los factores se representan con los signos  $-$  y  $+$ , o con los valores  $-1$  y  $+1$  (o simplemente 1). Para los factores cuantitativos el signo  $-$  representa al factor en su nivel bajo y el signo  $+$ , en su nivel alto. Para los factores cualitativos, la asignación de los signos  $-$  y  $+$  a cada nivel se realiza de forma arbitraria

La tabla con los niveles a experimentar en cada condición experimental para cada uno de los factores se denomina matriz del diseño. Por ejemplo, la Figura 1.1 representa la matriz del diseño para un diseño factorial  $2^3$ . El orden en que aquí se presentan los experimentos es el orden habitual de presentación y se denomina orden estándar. Este orden es muy fácil de recordar y de escribir, ya que en el primer factor los signos aparecen alternados una vez cada uno empezando por el signo  $-$ . En el segundo factor los signos aparecen alternados de 2 en 2, para el tercer factor de 4 en 4, para el cuarto de 8 en 8, y así sucesivamente.

Exp.	A	B	C
1	-	-	-
2	+	-	-
3	-	+	-
4	+	+	-
5	-	-	+
6	+	-	+
7	-	+	+
8	+	+	+

Figura 1.1: Matriz de diseño, de un diseño factorial  $2^3$  en el orden estándar

Planteada la matriz de diseño, hay que realizar los experimentos para obtener la respuesta en cada condición experimental. Tradicionalmente se recomienda que el orden de realización de los experimentos sea aleatorio, pero este tema, que en realidad es el objeto principal de la tesis, se comenta más adelante.

Una vez se tienen los valores de la respuesta, ya se puede cuantificar la influencia de los factores sobre dicha respuesta. Esta cuantificación se realiza a través del cálculo de los efectos, que pueden ser efectos principales o interacciones, siendo de especial interés el valor de las interacciones de 2 factores.

Para calcular los efectos puede construirse la llamada “matriz de diseño ampliada”, que es la matriz de diseño incluyendo las columnas con las interacciones, las cuales se

construyen multiplicando signo a signo (por filas) los signos de los factores que intervienen en la interacción.

Exp.	<i>A</i>	<i>B</i>	<i>C</i>	<i>AB</i>	<i>AC</i>	<i>BC</i>	<i>ABC</i>	<i>Y</i>
1	-	-	-	+	+	+	-	$y_1$
2	+	-	-	-	-	+	+	$y_2$
3	-	+	-	-	+	-	+	$y_3$
4	+	+	-	+	-	-	-	$y_4$
5	-	-	+	+	-	-	+	$y_5$
6	+	-	+	-	+	-	-	$y_6$
7	-	+	+	-	-	+	-	$y_7$
8	+	+	+	+	+	+	+	$y_8$

Figura 1.2: Matriz ampliada de un diseño  $2^3$  junto con la columna de respuestas

Los efectos se calculan como el producto escalar del vector formado por los elementos de la columna correspondiente a ese factor, por el vector respuestas, partido por la mitad del número de experimentos realizados. Es decir, si designamos por  $A$  el efecto principal del factor  $A$ , por  $AB$  la interacción de los factores  $A$  y  $B$ , y así con todos los efectos, y por  $\bar{A}$ ,  $\bar{AB}$ ... los vectores formados por los elementos de las columnas correspondientes, dichos efectos se calcularán de la forma:

$$A = \frac{\bar{A} \times \bar{Y}}{2^{k-1}}$$

$$AB = \frac{\bar{AB} \times \bar{Y}}{2^{k-1}}$$

donde  $\bar{Y}$  es el vector de de las respuestas.

El mismo procedimiento se sigue para cualquier efecto. Por ejemplo, y de una forma más detallada, el efecto principal del factor  $B$  en un diseño  $2^3$  será:

$$B = \frac{-y_1 - y_2 + y_3 + y_4 - y_5 - y_6 + y_7 + y_8}{4}$$

Las propiedades que presentan este tipo de diseños permiten plantear también diseños factoriales fraccionales. Estos son partes –fracciones– del diseño completo  $2^k$  que permiten evaluar la influencia de los factores sobre la respuesta realizando menos experimentos que los requeridos por el diseño completo. Su notación es  $2^{k-p}$  y el resultado de esta operación indica también el número de experimentos que es necesario realizar.

La contrapartida que presentan los diseños fraccionales es que los efectos no se estiman uno a uno sino en grupos, siendo imposible distinguir la aportación de cada efecto al valor que presenta el grupo al que está incorporado. Estas confusiones en la estimación de los efectos pueden no tener gran importancia si se confunden efectos cuyo valor interesa conocer con otros que se saben no significativos, o si se sabe que algunos de los factores con que se está experimentando resultarán inertes (se sabe, o se supone, que algunos serán inertes pero, naturalmente, no se sabe cuales lo serán). Si después del análisis existen confusiones que es necesario aclarar, habrá que realizar nuevas fracciones del experimento completo pero, en general, se podrá obtener la información que interesa sin necesidad de realizar los  $2^k$  experimentos.

Una excelente referencia para estudiar los diseños factoriales es el texto de Box, Hunter y Hunter (2005). Algunas de las características que este texto atribuye a este tipo de diseños son:

- Son especialmente útiles en las primeras etapas del trabajo experimental, cuando es probable que haya muchos factores por investigar.
- Requieren pocos experimentos elementales por cada factor, y a pesar de que no permiten explorar exhaustivamente una amplia región del espacio de los factores, pueden indicar tendencias y así determinar una dirección prometedora para futuros experimentos.
- Cuando se requiere una exploración más completa, se pueden añadir nuevas tandas de experimentación siguiendo una estrategia secuencial.
- Estos diseños y sus correspondientes fraccionales pueden ser utilizados en bloques para construir diseños con un grado de complejidad que se ajuste a la sofisticación del problema.

- La interpretación de las observaciones producidas por estos diseños se puede realizar en gran parte a base de sentido común, aritmética elemental y graficas de ordenador, en otras palabras, podríamos decir que el análisis de los resultados es bastante simple para estos diseños.

## 1.2 Orden de experimentación

Tradicionalmente se aconseja aleatorizar el orden de ejecución de los experimentos de un diseño factorial. Así, el orden de ejecución de un diseño  $2^3$  podría ser el que se indica en la Figura 1.3.

Orden estándar	Orden de ejecución	A	B	C
5	1	-	-	+
6	2	+	-	+
8	3	+	+	+
4	4	+	+	-
7	5	-	+	+
3	6	-	+	-
2	7	+	-	-
1	8	-	-	-

Figura 1.3: Matriz de diseño en orden aleatorio

La aleatorización persigue neutralizar la influencia de variables desconocidas que afectan a la respuesta y no se pueden mantener constantes a lo largo de todo el plan de experimentación. Es fácil observar que si se realizan los experimentos en orden estándar, la primera mitad de estos experimentos se realizan con el último factor a nivel  $-$ , y la otra mitad con dicho factor a nivel  $+$ , de forma que si existe algún factor ligado al orden de experimentación que afecta a la respuesta, esa influencia será atribuida, erróneamente, al último factor que aparezca en la matriz de diseño.

Aleatorizando el orden de experimentación se espera que esa influencia se difumine entre todos los efectos y evite tomar decisiones equivocadas sobre la significación estadística de alguno de ellos, aunque esto no siempre ocurre, tal como se verá más adelante.

### 1.3 Hipótesis sobre la influencia de factores desconocidos

Para nuestro trabajo utilizaremos la hipótesis de respuesta presentada por de León (2004), para la cual se supone el siguiente modelo:

$$Y = f(\mathbf{X}) + g(\mathbf{Z}) + \varepsilon$$

donde  $f(\mathbf{X})$  es la función que expresa la relación entre los factores en estudio y la respuesta;  $g(\mathbf{Z})$  corresponde a la influencia de los factores desconocidos que no se mantienen constantes a lo largo del plan de experimentación y, por último,  $\varepsilon$  es el error experimental.

Si los valores de las variables representadas por  $\mathbf{Z}$  se mantuvieran constantes, su presencia no afectaría a la estimación de los efectos, sino que simplemente añadirían un valor a la media de las respuestas, independientemente de cual fuera la forma de  $g(\mathbf{Z})$ .

El problema estriba en que los valores de  $\mathbf{Z}$  vayan variando durante el plan de experimentación de acuerdo con algún patrón desconocido. En principio, este patrón puede ser de cualquier tipo, pero si esas variables están relacionadas con las condiciones ambientales, con el deterioro o transformación de las materias primas, con cambios en la maquinaria, o con el proceso de medida, parece razonable suponer que la magnitud de la influencia irá aumentando a medida que se realizan los experimentos.

### 1.4 Medida de la influencia de los factores desconocidos

De acuerdo con las consideraciones realizadas, y suponiendo que en el rango de variación de las  $\mathbf{Z}$  la expresión  $g(\mathbf{Z})$  se puede considerar lineal, podemos asumir que  $g(\mathbf{Z}) = k \cdot i$ , donde  $i$  representa el orden en que se ha realizado el experimento y  $k$  es una constante de proporcionalidad. Para simplificar las expresiones que se plantean, se supone que  $k = 1$ , que todos los factores son inertes y que no existe error experimental. De esta forma, el valor de la respuesta coincide con el orden de experimentación y el sesgo que se produce en la estimación de los efectos coincide con el valor de los efectos calculados.

Por ejemplo, sea un diseño  $2^3$  que se realiza en orden estándar y en el que se obtienen unas respuestas de la forma indicada anteriormente. Su matriz de diseño ampliada junto con los valores de la respuesta y de los efectos se encuentran en la Figura 1.4.

Orden estándar	Orden de ejecución	$A$	$B$	$C$	$AB$	$AC$	$BC$	$ABC$	$Y$
1	1	-1	-1	-1	1	1	1	-1	1
2	2	1	-1	-1	-1	-1	1	1	2
3	3	-1	1	-1	-1	1	-1	1	3
4	4	1	1	-1	1	-1	-1	-1	4
5	5	-1	-1	1	1	-1	-1	1	5
6	6	1	-1	1	-1	1	-1	-1	6
7	7	-1	1	1	-1	-1	1	-1	7
8	8	1	1	1	1	1	1	1	8
Efectos		1	2	4	0	0	0	0	

Figura 1.4: Matriz ampliada y valores de los efectos para un diseño  $2^3$  en que la respuesta es igual al orden de ejecución de los experimentos

En este caso, la influencia de los factores desconocidos, representada por los valores que va tomando la respuesta, hace que el efecto principal de los factores  $A$ ,  $B$  y  $C$  no sea cero (valor que tomaría si no existiera la influencia de esos factores) sino que tenemos:  $A=1$ ,  $B=2$  y  $C=4$ . Recuérdese que estos efectos están midiendo sólo la influencia de los factores desconocidos sobre la respuesta y que en realidad se pueden considerar como sesgos introducidos en la estimación de los efectos.

Como medida de dicha influencia tomaremos el valor del efecto máximo (en valor absoluto) calculado de esta forma, lo cual es lo mismo que decir el sesgo máximo introducido en la estimación de los efectos. Por tanto, buscaremos órdenes de experimentación que produzcan el mínimo sesgo máximo en valor absoluto, expresión que, por comodidad, simplificaremos y nos referiremos a diseños con mínimo sesgo.

Esta medida fue utilizada por de León (2004), quien demostró que minimizar su valor tiene gran interés ya que la suma de los cuadrados de los sesgos es constante, respondiendo a la expresión:

$$\sum_{i=1}^{2^k} (\text{sesgo})^2 = 4 \frac{(2^k - 1)}{2^k} s_Y^2$$

donde  $s_Y^2$  representa la varianza de las respuestas. Esto significa que si un sesgo es muy grande, los demás no pueden ser grandes también, sino que deben ser pequeños para que

la suma de los cuadrados se mantenga constante, y esto hace que sea más probable tomar erróneamente como significativos a los efectos con sesgo grande.

Otros autores, como Draper y Stoneman (1968), Dickinson (1974) o Tack y Vandebroek (2002) utilizan lo que denominan *Time Count*, que es igual al producto escalar de la columna correspondiente al efecto en cuestión y la columna de respuestas., es decir:

$$\text{Efecto} = \frac{\text{Time Count}}{2^{k-1}}$$

Existe por tanto una relación directa entre el valor del efecto y el *Time Count*, por lo que es indiferente usar uno u otro.

## 1.5 Aleatorización y neutralización de la influencia de los factores desconocidos

De León (2004) realizó un análisis exhaustivo del sesgo que se obtiene en la estimación de los efectos al ejecutar un diseño  $2^3$  en todos los ordenes posibles, con un modelo para la respuesta como el descrito anteriormente. Observó que el sesgo máximo solo toma valores entre 2,0 y 4,0 a saltos de 0,5. El resumen de las frecuencias obtenidas para cada sesgo se encuentra en la Tabla 1.1.

Tabla 1.1: Valores del sesgo máximo en un diseño  $2^3$

Sesgo máximo	Frecuencia	%
2,0	1.344	3,33
2,5	9.408	23,33
3,0	13.440	33,33
3,5	8.064	20,00
4,0	8.064 <sup>(1)</sup>	20,00
Total	40.320	100

<sup>(1)</sup> Incluye el orden estándar

Es decir, una ordenación tan mala como el orden estándar, que carga toda la influencia de los factores desconocidos en el efecto principal del último factor de la matriz de

diseño, produce un sesgo en la estimación de este factor, igual al que provocan una quinta parte de los órdenes obtenidos aleatorizando.

Sólo un 3,3% de los ordenes producen un sesgo mínimo en el diseño.

Para diseños  $2^4$ , de León también realizó un estudio, en parte analítico y en parte por simulación, llegando a los valores que se indican en la Tabla 1.2.

Tabla 1.2: Valores del sesgo máximo en un diseño  $2^4$

Sesgo máximo	%
3,00	0,0016
3,25	0,063
3,50	0,669
3,75	2,70
4,00	6,37
4,25	10,16
4,50	12,94
4,75	13,27
5,00	12,62
5,25	10,47
5,50	8,64
5,75	6,48
6,00	5,13
6,25	3,49
6,50	2,56
6,75	1,63
7,00	1,16
7,25	0,704
7,50	0,470
7,75	0,235
8,00	0,235

De León pone de manifiesto que si se consideran los valores comprendidos entre 6,00 y 8,00 como susceptibles de producir un sesgo considerable, se tiene que el porcentaje de veces que aparecen esos valores al aleatorizar el orden de los experimentos es del 15,62%. Al incluir en estos valores el sesgo de 5,75, el porcentaje sube al 22,10%. En cambio, solo aparecen en promedio 16 órdenes que tienen un sesgo igual a 3 en cada millón de aleatorizaciones del orden de experimentación.

Por tanto, parece razonable buscar los órdenes de experimentación que mejor neutralizan la influencia de los factores desconocidos.

## 1.6 Número de cambios de nivel de los factores

El criterio del número de cambios de nivel de los factores es tal vez uno de los más importantes a considerar a la hora de enfrentar un proceso de experimentación. Esto es debido a que existen factores que implican características técnicas de difícil manejo (por ejemplo, cambios de molde), o que requieren un tiempo de estabilización para obtener condiciones de operación adecuadas (p.e.: temperatura de un horno), o por múltiples motivos (p.e.: lavado a fondo de tuberías para una nueva experimentación) y esto puede tener un impacto importante en el esfuerzo y, por tanto, el costo que es necesario para llevar a cabo la experimentación.

Evidentemente, el número de cambios de nivel de los factores depende del resultado de la aleatorización. En la Tabla 1.3 se tienen 2 órdenes de ejecución posibles al aleatorizar un diseño  $2^4$ . El primer resultado exige realizar 42 cambios de nivel, mientras que el segundo sólo requiere 15.

Tabla 1.3 Número de cambios de nivel en dos órdenes de ejecución del diseño  $2^4$

a) 42 cambios de nivel					b) 15 cambios de nivel				
Orden	A	B	C	D	Orden	A	B	C	D
16	+	+	+	+	3	-	+	-	-
10	+	-	-	+	1	-	-	-	-
7	-	+	+	-	2	+	-	-	-
9	-	-	-	+	6	+	-	+	-
12	+	+	-	+	14	+	-	+	+
6	+	-	+	-	10	+	-	-	+
11	-	+	-	+	12	+	+	-	+
4	+	+	-	-	4	+	+	-	-
14	+	-	+	+	8	+	+	+	-
3	-	+	-	-	16	+	+	+	+
2	+	-	-	-	15	-	+	+	+
15	-	+	+	+	7	-	+	+	-
1	-	-	-	-	5	-	-	+	-
5	-	-	+	-	13	-	-	+	+
8	+	+	+	-	9	-	-	-	+
13	-	-	+	+	11	-	+	-	+

Número de cambios	9	13	10	10	Número de cambios	2	4	4	5
-------------------	---	----	----	----	-------------------	---	---	---	---

El menor número de cambios de nivel de los factores se logra cuando sólo se cambia el nivel de un factor entre dos pruebas experimentales consecutivas, o de forma equivalente cuando se cambia un sólo signo al pasar de una fila a la siguiente en la matriz de diseño, tal como ocurre en la Tabla 1.3b. Por tanto, en un diseño  $2^k$  (diseño completo) el mínimo número de cambios de nivel es igual a  $2^k - 1$ , pues si se presentara un número menor de cambios se tendrían filas repetidas.

En los diseños fraccionales la evaluación del número de cambios no es tan sencilla. Primero hay que tener en cuenta que los factores que aparecen en la matriz de diseño pueden considerarse divididos en 2 grupos: 1) los que se acomodan en una matriz completa (por ejemplo, si se hacen 8 experimentos se podrán acomodar los 3 primeros), y 2) los que se generan a partir de interacciones de los primeros. Por ejemplo, en la Figura 1.5 se presenta la matriz de un diseño  $2^{5-2}$  en el que las columnas correspondientes a los factores  $D$  y  $E$  se identifican con las interacciones  $AB$  y  $AC$  respectivamente. A estas identificaciones que sirven para completar las columnas de los factores que no se han acomodado en la matriz completa se les denomina generadores del diseño.

Orden estándar	$A$	$B$	$C$	$D=AB$	$E=AC$
1	-1	-1	-1	1	1
2	1	-1	-1	-1	-1
3	-1	1	-1	-1	1
4	1	1	-1	1	-1
5	-1	-1	1	1	-1
6	1	-1	1	-1	1
7	-1	1	1	-1	-1
8	1	1	1	1	1

Figura 1.5: Matriz de un diseño  $2^{5-2}$  con generadores  $D=AB$  y  $E=AC$

Sean  $NC(X)$  y  $NC(Y)$  el número de cambios correspondientes a los factores  $X$  e  $Y$  acomodados en la matriz completa, la cual se ha ordenado de forma que el número de cambios sea mínimo. Si la secuencia de signos de un nuevo factor se identifica con la interacción  $XY$ , el número de cambios en este factor será  $NC(XY) = NC(X) + NC(Y)$ .

Efectivamente, los cambios de signo al pasar de una fila a la siguiente en la columna  $X$  producen también un cambio de signo en la columna  $XY$ , que no puede ser compensado por un cambio en la columna  $Y$  ya que en la matriz completa se tiene un solo cambio al pasar de una fila a la siguiente.

Por la misma razón, si un factor se genera con la interacción de 3 o más factores de los situados en la matriz completa (recordemos, con mínimo número de cambios) el número de cambios en el nuevo factor será igual a la suma del número de cambios de los factores que intervienen en la interacción con la cual se identifica.

Orden estándar	Orden de ejecución	$A$	$B$	$C$	$D=AB$	$E=AC$	$F=BC$	$G=ABC$
8	1	+	+	+	+	+	+	+
4	2	+	+	-	+	-	-	-
2	3	+	-	-	-	-	+	+
1	4	-	-	-	+	+	+	-
3	5	-	+	-	-	+	-	+
7	6	-	+	+	-	-	+	-
5	7	-	-	+	+	-	-	+
6	8	+	-	+	-	+	-	-
Número de cambios:		2	3	2	5	4	5	7

Figura 1.6: Número de cambios en los factores de un diseño  $2^{7-4}$

Entendemos que la dificultad de cambiar los niveles de los factores no será igual para todos ellos, sino, en general, solo para unos pocos. Por esta razón, en los diseños fraccionales buscaremos órdenes que minimicen el número de cambios en los factores que se sitúan en la matriz completa, y asignaremos a estas posiciones los factores con mayores dificultades de cambio. Con la regla vista anteriormente se puede deducir cuales serán los cambios necesarios en el resto de factores. Si existen más factores que los que se acomodan en la matriz completa que tienen dificultades de cambio de nivel, se podrá utilizar los que presenten un número menor en este segundo grupo. También se

podría actuar sobre los generadores buscando aquellos que minimizan el número de cambios, si su repercusión en las confusiones que se generan en la estimación de los efectos son tolerables.

De esta forma, a efectos de buscar ordenaciones con el mínimo número de cambios, se pueden identificar los diseños por el número de experimentos que requieren, siendo indiferente que se trate de un diseño  $2^3$  o un  $2^{6-3}$ , ya que ambos requieren 8 experimentos.

## **1.7 Objetivos de la tesis**

El objetivo fundamental de esta tesis es ampliar la propuesta que de León (2004) realizó para diseños factoriales a dos niveles con 8 y 16 experimentos, a diseños, también factoriales a dos niveles, pero con cualquier número de experimentos. Además, se pretende que las ordenaciones propuestas puedan ser obtenidas fácilmente con medios a disposición de los experimentadores.

En primer lugar, en el capítulo siguiente, se realiza un análisis de las propuestas existentes en este terreno, para aprovechar todas las buenas ideas que puedan existir sobre el tema y, a partir de ellas, ir elaborando nuestra propuesta.



# Capítulo 2



# Capítulo 2

## Antecedentes y estado del arte

Este capítulo tiene como propósito presentar una revisión bibliográfica exhaustiva de las diferentes propuestas que existen respecto al orden de ejecución de los diseños factoriales para conseguir que las conclusiones del análisis no se vean afectadas por la influencia de factores ajenos a la experimentación y/o que el número de cambios a realizar en los niveles de los factores sea mínimo.

Para facilitar el análisis de estas propuestas se realiza una agrupación en tres categorías que pueden clasificarse así:

El primer grupo corresponde a las propuestas que plantean órdenes para eliminar la influencia de los factores ajenos a la experimentación (sesgo cero), ya sea sólo en los efectos principales o en los efectos principales y las interacciones dobles, pero que no consideran el número de cambios de nivel de los factores. En este grupo están situados los trabajos presentados por Cox (1952), Box (1952), Hill (1960), Daniel y Wilcoxon (1966), John (1990), Cheng y Jacroux (1988), Cheng y Steinberg (1991), Bailey et al. (1992), Atkinson y Donev (1996).

El segundo grupo corresponde a las propuestas que proponen órdenes con el mínimo número de cambios de nivel de los factores sin considerar la influencia que pueden ejercer los factores ajenos a la experimentación. En este grupo se encuentran los

trabajos presentados por Jan y Wang (1995), Cheng et al. (1998) y Chen y Wang (2001).

El tercer grupo corresponde a las propuestas que buscan órdenes teniendo en cuenta simultáneamente el sesgo y el número de cambios de nivel. Esta situación a su vez implica dar prioridad a uno de los dos aspectos o buscar un equilibrio entre ellos, por lo cual este grupo se puede dividir entre los que dan prioridad a la eliminación de la influencia que pueden ejercer factores ajenos a la experimentación y luego evaluar el número de cambios (pocos cambios, pero no el mínimo posible) como los trabajos de Cheng (1985), Coster y Cheng (1988), Coster (1993), Jan y Wang (1995), Kim (1997) y Tack y Vandebroek (2002). Otros autores dan prioridad al número de cambios de nivel y posteriormente evalúan el sesgo (minimizar el sesgo, pero no eliminarlo completamente) como los trabajos de Dickinson (1974), de León (2004) y de León et al. (2005). Por último están las propuestas que buscan un equilibrio entre ambos aspectos (sin obtener el mínimo en ninguno de ellos) como los trabajos de Draper y Stoneman (1968), Joiner y Campbell (1976).

A continuación se presenta una descripción de cada uno de estos grupos y las diferentes propuestas que lo componen. Al final de este capítulo se comentan las debilidades del estado del arte actual y se plantean los aportes previstos en esta tesis.

## **2.1 Búsqueda de ordenaciones con el mínimo sesgo**

En términos generales los diferentes trabajos que se agrupan en esta propuesta tienen como fin la eliminación total del sesgo ya sea sólo en los efectos principales o en los efectos principales y las interacciones sin considerar el número de cambios de nivel de los factores. A continuación se presenta una descripción de estas propuestas por orden cronológico.

### **2.1.1 Cox (1952)**

Presenta por primera vez el concepto de diseños sistemáticos como aquellos en los que la aleatorización del orden de ejecución no es la mejor opción posible, ya que la selección de una secuencia determinada permite contrarrestar la influencia de los factores ajenos a la experimentación. Para mostrar la utilidad práctica de este concepto presenta un caso de aplicación en una fábrica textil, en el cual define como debería ser planeado un experimento con el fin de eliminar la influencia que puede ejercer el tiempo (envejecimiento del algodón) en el proceso de fabricación de la lana.

En particular el experimento consiste en investigar el efecto que tiene el cambiar la humedad relativa en dos etapas de fabricación de la lana. La humedad relativa se estudia en tres niveles 50%, 60% y 70%. La lana se divide en lotes lo más parecidos posible que son numerados de forma aleatoria. Cada semana sólo uno de los lotes es procesado con un cierto tratamiento. El experimentador sospecha que la edad de la lana puede afectar el proceso, por lo que se podría generar una tendencia debido al añejamiento. Por esta razón se desarrolla un plan de ejecución de las pruebas de experimentación que permita eliminar la influencia que pueda tener el envejecimiento de los lotes de lana en el proceso. Para ello se asignan los experimentos a las etapas del proceso en el siguiente orden:  $T_2$ ,  $T_1$ ,  $T_3$ ,  $T_3$ ,  $T_2$ ,  $T_1$ ,  $T_1$ ,  $T_3$  y  $T_2$ , donde  $T_1$  representa un lote con humedad relativa del 50%,  $T_2$  un lote con humedad relativa del 60% y  $T_3$  un lote con humedad relativa del 70%. Este orden garantiza que al realizar el análisis con un método tradicional como los mínimos cuadrados ordinarios y las pruebas de significancia, el efecto de la etapa no se vea afectado por factores ajenos a la experimentación, lo que podría distorsionar los resultados obtenidos.

### **2.1.2 Box (1952) y Box y Hay (1953)**

Mostraron que cuando se tiene un conocimiento a priori de la superficie de respuesta, es posible seleccionar de forma conveniente las pruebas de experimentación en un diseño factorial a dos niveles de manera tal que se pueda eliminar el sesgo en algunos efectos del modelo. En otras palabras, el orden de experimentación debe escogerse de forma sistemática y no aleatoriamente con el fin de hacer frente a la influencia de factores ajenos a la experimentación sin perder la eficiencia del análisis. Para mostrar la aplicación práctica de su hallazgo utilizaron un experimento que consistía en comparar el efecto que ejercen diferentes dosis de cloruro en la contracción de un nervio de rata, estas dosis fueron aplicadas en días diferentes, por lo cual la influencia del tiempo debía ser eliminada para garantizar un análisis adecuado.

El procedimiento de experimentación consistía en extraer el nervio de la rata y conectarlo a un medidor. A continuación se introducía en una solución donde se le daba un estímulo eléctrico cada 12 segundos que le producía una contracción. Para lograr una medida confiable los aparatos operaban bajo las mismas condiciones todo el tiempo, el promedio de las contracciones era grabado y se usaba para calcular la respuesta. Cuando era conveniente se agrega una dosis de cloruro a la solución, lo que generaba una reducción en las contracciones. Esta reducción se medía a través del denominado “porcentaje de inhibición” y este era el valor que se utilizaba como respuesta en el ensayo. Después de cada prueba, el equipo se lavaba a fondo y se realizaba un nuevo ensayo.

Con lo descrito en el procedimiento se observa que la respuesta (porcentaje de inhibición) se puede hacer cada vez más grande, debido a la fatiga del nervio, a un lavado inadecuado o a otras causas externas que no se conocen claramente. Los autores proponen que una eliminación parcial de esta tendencia podría ser alcanzada con el uso de un bloqueo aleatorizado, donde los bloques serian los periodos sucesivos de tiempo.

### 2.1.3 Hill (1960)

Basado en los trabajos propuestos por Cox (1952) y Box (1952), Hill propone un diseño experimental que permite eliminar la influencia lineal o cuadrática que puedan ejercer los factores desconocidos, y que puede ser utilizado en la industria química para eliminar la influencia ejercida por el tiempo. Para mostrar su aplicación práctica, presentó un ejemplo que consistía en determinar el efecto de un catalizador en un proceso químico, en el cual el tiempo podría tener influencia en los resultados obtenidos por lo que era necesario contrarrestarla con un diseño adecuado.

El diseño usado fue un diseño sistemático compuesto de 4 factores y 14 experimentos que permite proteger de la influencia de factores ajenos a la experimentación que presenten un comportamiento polinomial de alto grado en los efectos del modelo.

Este plan puede ser presentado de forma equivalente como un diseño factorial  $2^4$  con los efectos principales libres de tendencia, usando el método de Daniel y Wilcoxon (1966), tal como lo expresa John (1990), con el orden de ejecución que se indica en la Figura 2.1.

Orden de Ejecución	Orden estándar	A	B	C	D	Orden de Ejecución	Orden estándar	A	B	C	D
1	1	-	-	-	-	9	12	+	+	-	+
2	15	-	+	+	+	10	6	+	-	+	-
3	14	+	-	+	+	11	7	-	+	+	-
4	4	+	+	-	-	12	9	-	-	-	+
5	8	+	+	+	-	13	13	-	-	+	+
6	10	+	-	-	+	14	3	-	+	-	-
7	11	-	+	-	+	15	2	+	-	-	-
8	5	-	-	+	-	16	16	+	+	+	+

Figura 2.1: Plan experimental equivalente al propuesto por Hill (1960) para eliminar la influencia de factores ajenos a la experimentación, presentado por John (1990)

### 2.1.4 Daniel y Wilcoxon (1966)

Se basan en la idea de que el diseño se puede plantear en bloques, cuadrados latinos, etc. con el fin de hacer más preciso el análisis. Pero cuando se tiene información adicional que permite suponer que el comportamiento de la influencia de los factores ajenos presenta una tendencia lineal o cuadrática, se puede construir un diseño no aleatorizado que permite proteger de dicha influencia a los efectos de interés. Daniel y Wilcoxon, definen una forma de medir la influencia de los factores ajenos a la experimentación en cada uno de los efectos del diseño, usando para ello el producto interno del contraste del efecto por la respuesta o *Time Count* definido por Draper y Stoneman (1968). A partir de esta medida se identifican los efectos que presentan un sesgo cero, a los cuales se les asignan los factores que se desea proteger de la influencia no deseada.

Estos autores encontraron órdenes para los diseños factoriales  $2^{k-p}$  con  $k - p = 2, 3, 4$  y 5. A continuación se presenta un ejemplo de este procedimiento para un diseño factorial a dos niveles con ocho experimentos.

En primer lugar se plantea el diseño factorial en el orden estándar y se construye la matriz completa, se calcula el *Time Count* lineal y el cuadrático para todos los efectos, como se presenta en la Figura 2.2.

	A	B	C	AB	AC	BC	ABC	Y	Y <sup>2</sup>
	-	-	-	+	+	+	-	1	1 <sup>2</sup>
	+	-	-	-	-	+	+	2	2 <sup>2</sup>
	-	+	-	-	+	-	+	3	3 <sup>2</sup>
	+	+	-	+	-	-	-	4	4 <sup>2</sup>
	-	-	+	+	-	-	+	5	5 <sup>2</sup>
	+	-	+	-	+	-	-	6	6 <sup>2</sup>
	-	+	+	-	-	+	-	7	7 <sup>2</sup>
	+	+	+	+	+	+	+	8	8 <sup>2</sup>
<i>Time Count</i> Lineal	8	16	32	0	0	0	0		
<i>Time Count</i> Cuadrático	0	0	0	8	16	32	0		

Figura 2.2: Selección del orden de experimentación para un diseño factorial a dos niveles con el método propuesto por Daniel y Wilcoxon (1966)

A partir del cálculo de los *Time Count* de la influencia lineal y cuadrática se seleccionan las tres columnas de contrastes que representan a los tres factores de interés en el diseño, con los cuales se define el orden de experimentación para los ocho experimentos en un diseño factorial a dos niveles, como el que se presenta en la Figura 2.3, el cual permite proteger de la posible influencia lineal o cuadrática de los factores ajenos a la experimentación.

<i>Orden</i>	<i>A(ABC)</i>	<i>B(A)</i>	<i>C(AB)</i>
5	-	-	+
4	+	+	-
2	+	-	-
7	-	+	+
6	+	-	+
3	-	+	-
1	-	-	-
8	+	+	+
<i>Time Count</i> Lineal	0	8	0
<i>Time Count</i> Cuadrático	0	0	8

Figura 2.3: Orden de experimentación para un diseño factorial a dos niveles obtenido con el método propuesto por Daniel y Wilcoxon (1966)

### 2.1.5 Cheng y Jacroux (1988)

Construyeron un procedimiento para obtener órdenes de experimentación en diseños factoriales a dos niveles que eliminan la influencia debida a factores ajenos a la experimentación en los efectos principales y en algunas de las interacciones dobles, pero sin considerar de forma simultanea el número de cambios que presentan dichos órdenes.

La idea de la construcción de los órdenes de experimentación libres de tendencia, partió de un lema planteado por los autores mediante el cual se puede comprobar que en cualquier diseño factorial completo  $2^k$ , el número de contrastes ortogonales a la tendencia lineal es como máximo de  $2^k - k - 1$  [Cheng y Jacroux (1988)]. En otras

palabras, que en un diseño factorial se pueden obtener  $2^k - k - 1$  columnas de signos que representen a los efectos de interés en los cuales el valor del sesgo es cero.

A partir de esta idea construyeron un algoritmo que permite encontrar órdenes de experimentación en diseños factoriales a dos niveles libres de tendencia lineal en los efectos principales y algunas interacciones dobles. Para concretar esta idea es necesario construir  $k$  generadores que representen a los efectos de los  $k$  factores de un nuevo diseño factorial  $2^k$  libre de tendencia. Estos generadores son construidos por medio de la asignación de un efecto principal del nuevo diseño a un producto interno o interacción de las columnas de los efectos principales del diseño factorial  $2^k$  en el orden estándar. La construcción de estas interacciones depende de si el número de factores en el diseño, es par o impar y si se parte de un diseño con 3 o con 4 factores. Esta construcción se puede describir de la siguiente forma:

Partiendo de un diseño con 3 factores:

Si denominamos  $A_i$  al efecto principal del factor  $i$  de un diseño factorial  $2^k$ , la asignación se realiza de la siguiente forma: a) para  $i = 1, 2, 3, \dots, 1-k$ , asignar a  $A_i$  la interacción de todos los factores del orden estándar excepto el  $i$ -ésimo factor. b) Asignar a  $A_k$  (último efecto del diseño libre de tendencia), la interacción de todos los factores del orden estándar. Una vez obtenidas las  $k$  columnas del diseño factorial  $2^k$  se puede encontrar el orden de experimentación asociando las filas en este orden a las filas del diseño factorial en orden estándar.

Partiendo de un diseño con 4 factores:

Si denominamos  $A_i$  al efecto principal del factor  $i$ , entonces la asignación debe ser realizada de la siguiente forma:

Si el número de factores,  $k$ , es par: a) para  $i = 1, 2, 3, \dots, 1-k$ , asignar a  $A_i$  la interacción de todos los factores excepto el  $i$ -ésimo. b) Asignar a  $A_k$  la interacción de todos los factores.

Si  $k$  es impar, se debe hacer lo siguiente: a) para  $i = 1, 2, 3, \dots, 1-k$ , asignar a  $A_i$  la interacción de todos los factores excepto el  $i$ -ésimo y el último. b) Asignar a  $A_k$  la interacción de todos los factores. Una vez obtenidas las  $k$  columnas del diseño factorial

$2^k$  se puede encontrar el orden de experimentación asociando las filas de tratamientos a los tratamientos del diseño factorial con el orden estándar.

A continuación se presenta un ejemplo de cómo funciona el algoritmo. Supongamos que el número de factores es de cuatro ( $k = 4$ ) y deseamos un orden de ejecución de un diseño factorial completo  $2^4$ . Ahora siguiendo los pasos del algoritmo tendríamos:

Se construye el orden estándar para el diseño  $2^4$ , como se muestra en la Figura 2.4.

Orden	$A_1$	$A_2$	$A_3$	$A_4$
1	-	-	-	-
2	+	-	-	-
3	-	+	-	-
4	+	+	-	-
5	-	-	+	-
6	+	-	+	-
7	-	+	+	-
8	+	+	+	-
9	-	-	-	+
10	+	-	-	+
11	-	+	-	+
12	+	+	-	+
13	-	-	+	+
14	+	-	+	+
15	-	+	+	+
16	+	+	+	+

Figura 2.4: Diseño factorial  $2^4$  con el orden estándar

A partir del orden estándar se deben construir los 4 generadores para este diseño, dado que  $k$  es par, debemos hacer la siguiente asignación de interacciones de los efectos a los efectos principales del nuevo diseño libre de tendencia, así:

$$\begin{aligned}
 A_1 &\rightarrow A_2A_3A_4 \\
 A_2 &\rightarrow A_1A_3A_4 \\
 A_3 &\rightarrow A_1A_2A_4 \\
 A_4 &\rightarrow A_1A_2A_3A_4
 \end{aligned}$$

Con lo cual se obtiene el diseño que se presenta en la Figura 2.5.

De la Figura 2.5, podemos observar que el sesgo es eliminado en los factores principales, pero el número de cambios total de cambios en el diseño es de 37, siendo 15 el mínimo posible para este diseño. Por tanto, podemos afirmar que este algoritmo es adecuado para la eliminación de la influencia lineal debida a los factores no controlados en los efectos principales y algunas interacciones dobles del diseño, pero inadecuado en relación al número de cambios de nivel de los factores.

Orden	$A_2A_3A_4$	$A_1A_3A_4$	$A_1A_2A_4$	$A_1A_2A_3A_4$
9	-1	-1	-1	1
7	-1	1	1	-1
6	1	-1	1	-1
12	1	1	-1	1
4	1	1	-1	-1
14	1	-1	1	1
15	-1	1	1	1
1	-1	-1	-1	-1
8	1	1	1	-1
10	1	-1	-1	1
11	-1	1	-1	1
5	-1	-1	1	-1
13	-1	-1	1	1
3	-1	1	-1	-1
2	1	-1	-1	-1
16	1	1	1	1
Cambios	5	13	9	10
Sesgo	0	0	0	0

Figura 2.5: Orden de experimentación de un diseño factorial  $2^4$  con los factores principales y algunas interacciones dobles libres de sesgo, hallados con el algoritmo de Cheng y Jacroux (1988)

### 2.1.6 John (1990)

Generaliza el trabajo de Daniel y Wilcoxon (1966) para obtener algunos órdenes de experimentación de diseños factoriales  $2^k$  y  $3^k$  en los que sus efectos principales no están correlacionados con la tendencia lineal o cuadrática, pero no considera de forma simultánea el número de cambios de nivel. Este trabajo fue la base para la propuesta de Coster y Cheng (1988), aunque su fecha de publicación sea posterior, debido a que en el año 1988 se citaba como manuscrito no publicado.

### **2.1.7 Cheng y Steinberg (1991)**

Presentaron unos procedimientos que permiten obtener órdenes de experimentación en diseños factoriales a dos niveles con 16 y 32 experimentos robustos a influencias simuladas por medio de un modelo autoregresivo de primer orden, completando de esta forma los trabajos existentes que sólo consideraban influencias polinomiales de primer o segundo orden. Los órdenes obtenidos se caracterizan por tener un número elevado de cambios de nivel, lo cual permite atenuar influencias de una complejidad mayor como las series temporales utilizadas.

### **2.1.8 Bailey et al. (1992)**

Usando la teoría algebraica de grupos, presentaron una extensión y unificación de los trabajos de Cheng y Jacroux (1988) y Coster y Cheng (1988) para generar diseños factoriales con cualquier número primo de niveles (diseños  $S^k$ ,  $S$  representa los niveles del factor y  $k$  el número de factores del diseño), donde los efectos principales y los efectos de las interacciones dobles están libres de la influencia debida a factores desconocidos, pero no consideran de forma simultánea el número de cambios de nivel de los factores.

### **2.1.9 Atkinson y Donev (1996)**

Proponen un algoritmo que permite encontrar órdenes de experimentación robustos a diferentes tipos de tendencia a través de un diseño D-óptimo. Se hace énfasis en los parámetros del modelo asociado a los factores y se compara este diseño con el que se obtiene con otro libre de tendencia, lo que permite hacer la evaluación de que tan adecuado es el diseño óptimo con respecto a la eliminación de la tendencia.

Adicionalmente, consideran que los diseños que tienen órdenes de experimentación ortogonales a la tendencia no siempre son adecuados ya que en muchas situaciones prácticas esta ortogonalidad no es posible o razonable, por lo que es necesario elegir las ordenaciones de los tratamientos de forma óptima. Para ejemplificar esta propuesta presentan algunos órdenes de experimentación para 6 y 9 pruebas en presencia de tendencia lineal o cuadrática y con tres factores que pueden ser cualitativos o cuantitativos. También presentan diseños de primer orden para tres factores, con y sin interacciones, en 15 pruebas considerando varias posibilidades con respecto a la asignación de las pruebas en la ejecución: uniformemente espaciadas en el tiempo, o varias pruebas en cada momento en el tiempo.

## 2.2 Búsqueda de ordenaciones con el mínimo número de cambios de nivel

Las propuestas que pertenecen a este grupo son aquellas que generan órdenes con el mínimo número de cambios de nivel de los factores sin considerar la influencia que puede ser ejercida por los factores ajenos a la experimentación. En este grupo se encuentran los trabajos realizados por Jan y Wang (1995), Cheng et al. (1998), Chen y Wang (2001).

### 2.2.1 Jan y Wang (1995)

Wang y Jan (1995), tomaron las ideas de Kacker y Tsui (1990), para asignar los factores a las columnas de un arreglo ortogonal apropiado con el fin de facilitar a los usuarios no expertos la búsqueda de órdenes de experimentación en diseños factoriales a dos niveles que cumplieran la condición de tener el mínimo número de cambios de nivel.

El algoritmo para hallar estos órdenes de experimentación en los diseños  $2^k$  propuesto por Wang y Jan (1995) puede ser resumido en los siguientes pasos:

- Construir la matriz de diseño en el orden estándar para un diseño  $2^k$ .
- Organizar las columnas de la matriz de diseño en el orden ascendente del número de cambios de nivel.
- Construir la matriz completa para el diseño obtenido al organizar las columnas.
- Organizar las columnas de la matriz completa en orden ascendente del número de cambios.
- Seleccionar  $k$  las primeras columnas que permitan obtener el mínimo número de cambios de nivel de los factores en el diseño  $2^k$ .
- Obtener orden de experimentación para el diseño factorial  $2^k$ .

A continuación se describen a manera de ejemplo cada uno de los pasos que componen este algoritmo para la búsqueda de órdenes de experimentación con el mínimo número de cambios:

En primer lugar se debe construir el diseño factorial  $2^k$  en orden estándar, es decir, que el primer factor cambia de signo en cada fila, el segundo factor cambia de signo cada dos filas, en general el factor  $k$  cambia de signo cada  $2^{k-1}$  filas. Por ejemplo para el diseño  $2^4$  la construcción del orden estándar estaría dada por las cuatro columnas que se presentan en la Figura 2.6a.

Orden	A	B	C	D		$X_1$	$X_2$	$X_3$	$X_4$
1	-	-	-	-		-	-	-	-
2	+	-	-	-		-	-	-	+
3	-	+	-	-		-	-	+	-
4	+	+	-	-		-	-	+	+
5	-	-	+	-		-	+	-	-
6	+	-	+	-		-	+	-	+
7	-	+	+	-		-	+	+	-
8	+	+	+	-		-	+	+	+
9	-	-	-	+		+	-	-	-
10	+	-	-	+		+	-	-	+
11	-	+	-	+		+	-	+	-
12	+	+	-	+		+	-	+	+
13	-	-	+	+		+	+	-	-
14	+	-	+	+		+	+	-	+
15	-	+	+	+		+	+	+	-
16	+	+	+	+		+	+	+	+
Cambios						1	3	7	15

(a)
(b)

Figura 2.6: Diseño factorial  $2^4$  con el orden estándar y con las columnas en orden creciente del número de cambios

A continuación se ordenan las columnas de menor a mayor número de cambios, lo cual es equivalente al orden estándar inverso tal como se indica en la Figura 2.6b.

A partir del producto interno de estos vectores o columnas estándar, se puede generar un arreglo ortogonal  $[AO_N(2^{N-1})]$ , donde  $N = 2^k$  con  $2^k - 1$  columnas ortogonales, en el cual, cada una de las columnas presentan un número de cambios entre 1 y  $2^k - 1$ . Por ejemplo, si multiplicamos las columnas  $X_1$  y  $X_2$  de la Figura 2.6b obtendríamos la columna  $X_1X_2$  con dos cambios de nivel, o si multiplicamos las columnas  $X_2$  y  $X_3$  se

obtiene la columna  $X_2X_3$  con cuatro cambios de nivel. Si en nuestro ejemplo multiplicamos de forma conveniente las columnas y las ordenamos de menor a mayor número de cambios de nivel, se obtiene el arreglo ortogonal  $AO_{16}(2^{15})$ , es decir, se tienen 15 columnas como se muestra en la Figura 2.7.

$X_1$	$X_1 X_2$	$X_2$	$X_2 X_3$	$X_1 X_2 X_3$	$X_1 X_3$	$X_3$	$X_3 X_4$	$X_1 X_3 X_4$	$X_1 X_2 X_3 X_4$	$X_2 X_3 X_4$	$X_2 X_4$	$X_1 X_2 X_4$	$X_1 X_4$	$X_4$
-1	1	-1	1	1	1	-1	1	1	1	1	1	1	1	-1
-1	1	-1	1	1	1	-1	-1	-1	-1	-1	-1	-1	-1	1
-1	1	-1	-1	-1	-1	1	-1	-1	-1	-1	1	1	1	-1
-1	1	-1	-1	-1	-1	1	1	1	1	1	-1	-1	-1	1
-1	-1	1	-1	-1	1	-1	1	1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	1	-1	-1	-1	1	1	1	1	-1	1
-1	-1	1	1	1	-1	1	-1	-1	1	1	-1	-1	1	-1
-1	-1	1	1	1	-1	1	1	1	-1	-1	1	1	-1	1
1	-1	-1	1	-1	-1	-1	1	-1	-1	1	1	-1	-1	-1
1	-1	-1	1	-1	-1	-1	-1	1	1	-1	-1	1	1	1
1	-1	-1	-1	1	1	1	-1	1	1	-1	1	-1	-1	-1
1	-1	-1	-1	1	1	1	1	-1	-1	1	-1	1	1	1
1	1	1	-1	1	-1	-1	1	-1	1	-1	-1	1	-1	-1
1	1	1	1	-1	1	1	-1	1	-1	1	1	-1	1	1
1	1	1	1	-1	1	1	1	-1	1	-1	1	-1	1	-1
1	1	1	1	-1	1	1	1	-1	1	-1	1	-1	1	1
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Número de cambios en las columnas														

Figura 2.7: Arreglo ortogonal  $AO_{16}(2^{15})$  para el diseño factorial  $2^4$  con las columnas ordenadas de menor a mayor número de cambios de nivel.

Una vez se tiene el arreglo ortogonal para el diseño factorial a dos niveles con todas las columnas organizadas de menor a mayor en relación al número de cambios, se procede a seleccionar las columnas que conjuntamente presenten el mínimo número de cambios para un diseño factorial a dos niveles, que como sabemos corresponde al valor  $2^k - 1$ . Por ejemplo, para el caso particular del diseño  $2^4$  si deseamos obtener un diseño factorial completo con el mínimo número de cambios de nivel, en primera instancia debemos escoger las cuatro primeras columnas de la Figura 2.7 que corresponden a los cuatro factores necesarios para este tipo de diseño, así:  $A = X_1$ ,  $B = X_1 X_2$ ,  $C = X_2$  y

$D = X_2X_3$ , sin embargo, estas asignaciones tienen un problema, ya que se presenta la confusión  $B = AC$ . Para evitar esta situación debemos mover  $C$  a  $X_2X_3$ . Ahora podríamos asignar  $D$  a  $X_1X_2X_3$ ,  $X_1X_3$  o  $X_3$ , pero resulta que se generan las confusiones  $AC = D$ ,  $B = CD$  o  $AB = D$ , respectivamente. Todos estos diseños serían inadecuados y por lo tanto  $D$  debe ser asignado a la próxima columna disponible, es decir,  $X_3X_4$ . Por lo cual el diseño completo con el mínimo número de cambios estaría conformado por las siguientes columnas  $A = X_1$ ,  $B = X_1X_2$ ,  $C = X_2X_3$  y  $D = X_3X_4$  al que corresponde el orden de experimentación que se presenta en la Figura 2.8.

Orden	$X_1$	$X_1X_2$	$X_2X_3$	$X_3X_4$
15	-1	1	1	1
7	-1	1	1	-1
3	-1	1	-1	-1
11	-1	1	-1	1
9	-1	-1	-1	1
1	-1	-1	-1	-1
5	-1	-1	1	-1
13	-1	-1	1	1
14	1	-1	1	1
6	1	-1	1	-1
2	1	-1	-1	-1
10	1	-1	-1	1
12	1	1	-1	1
4	1	1	-1	-1
8	1	1	1	-1
16	1	1	1	1
Núm. cambios	1	2	4	8
Sesgo	8	0	0	0

Figura 2.8: Diseño factorial  $2^4$  con el mínimo número de cambios de nivel de los factores hallado con el algoritmo de Wang y Jan (1995)

Para calcular la influencia ejercida por los factores ajenos a la experimentación utilizaremos la medida de sesgo propuesta por de León (2004) y el vector de respuestas  $(1, 2, 3, \dots, N)$ , con lo cual podemos comprobar que el valor de sesgo obtenido para el orden hallado con este algoritmo corresponde al máximo valor (8) que puede presentar

un orden con estas características, ya que concentra toda la influencia de los factores ajenos a la experimentación en el primer efecto del diseño, es decir, que este orden es adecuado en cuanto al número de cambios de nivel de los factores, pues presenta el mínimo posible, pero no es adecuado para atenuar la influencia ejercida por factores ajenos a la experimentación, ya que presenta el máximo sesgo posible que puede ser obtenido en este tipo de diseños.

Debido a que este algoritmo siempre utiliza las columnas del diseño factorial  $2^k$  en orden estándar, todos los órdenes obtenidos con este procedimiento en cualquier diseño factorial van a presentar el máximo sesgo, pues siempre se va a seleccionar la primera columna del arreglo ortogonal que presenta un solo cambio de nivel de los factores, por lo cual el sesgo se concentra en este factor. Así pues, los órdenes generados por este algoritmo son adecuados desde un punto de vista económico ya que reducen el número de cambios de nivel al mínimo, pero no son adecuados cuando puede existir una posible influencia de factores no controlados, pues la concentra en el efecto de un sólo factor lo que podría conducir a un análisis con conclusiones erróneas.

### **2.2.2 Cheng et al. (1998)**

Presentaron una propuesta más formal desde el punto de vista matemático del procedimiento propuesto Jan y Wang (1995), acompañándola de las demostraciones que no fueron aportadas por estos autores. Dicha propuesta consiste en la construcción de órdenes de experimentación de diseños factoriales a dos niveles con el mínimo número de cambios basado en el hecho que los  $2^k$  experimentos de un diseño factorial fraccional pueden ser ordenados de forma tal que el número de cambios de nivel de los factores sea igual a cada uno de los enteros entre 1 y  $2^k - 1$ . Esta propuesta no considerara simultáneamente el problema de la tendencia debida a factores no controlados, por lo que todos los órdenes obtenidos al usar este método presentan el máximo sesgo posible.

Adicionalmente, en este trabajo se presenta un algoritmo para hallar órdenes de experimentación en diseños factoriales con el máximo número de cambios de nivel, los cuales son adecuados para eliminar influencia polinomial de alto grado en este tipo de diseño, como lo demostraron Cheng y Steinberg (1991). Así, al usar un diseño con el máximo número de cambios de nivel de los factores se garantiza la eliminación de múltiples tipos de tendencia, pero el costo experimental es el más elevado posible.

### 2.2.3 Chen y Wang (2001)

Usando un planteamiento matemático muy formal y abstracto, generalizaron los trabajos de Jan y Wang (1995) y Cheng et al. (1998) para la búsqueda de órdenes con el mínimo número de cambios en diseños factoriales fraccionales con cualquier número primo de niveles ( $S^k$ ), aunque sin considerar de forma simultánea la influencia debida a factores no controlados. El método, en esencia, consiste en construir un arreglo ortogonal para un diseño factorial con un número primo de niveles  $S$  ( $S^k$ ), al cual se le puedan asignar las columnas a los factores de un diseño con  $S$  niveles y obtener el mínimo número de cambios de nivel para el diseño correspondiente.

## 2.3 Búsqueda de ordenaciones que consideran simultáneamente el número de cambios de nivel y el sesgo

Las propuestas que figuran en este grupo son aquellas que consideran que se pueden llegar a obtener órdenes con buenas características al evaluar simultáneamente el número de cambios de nivel y la atenuación de la influencia de los factores ajenos a experimentación (sesgo). Estas propuestas se pueden dividir en 3 grupos:

- Las que dan prioridad a la eliminación del sesgo, como en los trabajos de Cheng (1985), Coster y Cheng (1988), Jan y Wang (1995), Kim (1997) y Tack y Vandebroek (2002).
- Las que dan prioridad a minimizar el número de cambios de nivel como los trabajos de Dickinson (1974), de León (2004) y de León et al. (2005).
- Aquellas que buscan un equilibrio en ambos aspectos como los trabajos de Draper y Stoneman (1968), Joiner y Campbell (1976).

A continuación se describen cada una de estas propuestas:

### 2.3.1 Cheng (1985)

Presentó una descripción teórica sobre la estructura de costes en los diseños factoriales a dos niveles para encontrar órdenes de experimentación que eliminan la tendencia lineal debida a factores no controlados en los efectos principales y que a su vez presenten un bajo número de cambios de nivel de los factores. Esta nueva forma de enfrentar el

problema se crea como una alternativa para ampliar la búsqueda de órdenes de experimentación en diseños factoriales con un mayor número de factores sin considerar de forma prioritaria el número de cambios de nivel como lo venían haciendo Draper y Stoneman (1968) y Dickinson (1974), ya que se concentra principalmente en la eliminación de la influencia lineal debida a factores desconocidos con la ayuda de las estructuras algebraicas que presentan los diseños factoriales. Adicionalmente, Cheng presenta un planteamiento teórico en el que demuestra que en un diseño factorial a dos niveles pueden existir múltiples órdenes con el mínimo número de cambios de nivel y dentro de estos es posible que exista un orden que proteja a los efectos principales del diseño de la influencia de los factores ajenos a la experimentación.

Vale la pena destacar el siguiente comentario expresado por el autor: *“Cuando un conjunto de órdenes con el mínimo número de cambios de nivel es lo suficientemente grande, uno podría esperar que alguno cualquiera de esos órdenes se comporte bien en relación a la eliminación del sesgo. En esta situación uno podría adoptar el método de Dickinson (1974) y buscar entre todos los órdenes con el mínimo número de cambios de nivel uno que también sea resistente a la influencia de factores ajenos a la experimentación. Otro método útil es el de aumentar el número de cambios de nivel de los factores para disminuir el valor del sesgo”*. Los trabajos realizados por este autor se han concentrado fundamentalmente en el segundo aspecto, es decir, en aumentar el número de cambios del diseño para garantizar la eliminación del sesgo en los efectos principales y en algunas interacciones..

### **2.3.2 Coster y Cheng (1988)**

Utilizando una estructura matemática compleja, basada en la teoría de los grupos de Galois, Coster y Cheng desarrollaron una técnica conocida con el nombre de método *foldover* generalizado (GFS, de sus siglas en inglés) que permite obtener órdenes de experimentación en diseños factoriales fraccionados que minimizan una función de coste basada en el número de veces que los factores cambian de nivel en la ejecución del experimento y, simultáneamente, también consigue que todos los efectos principales sean ortogonales a la tendencia lineal.

En términos generales este procedimiento consiste en construir la matriz de diseño de un orden de experimentación libre de tendencia en factores principales, a partir de una matriz generadora de un cierto número de factores (generadores) de la siguiente forma: Inicialmente se ubica en la primera fila aquella que presenta todos los factores a nivel bajo, luego se multiplica esta fila por la primera fila de la matriz generadora (o primer generador) y se obtiene una nueva fila que es agregada a la matriz de diseño. Las dos filas de la matriz de diseño obtenidas en el paso anterior son multiplicadas por la

segunda fila de la matriz generadora (o segundo generador), con lo cual se obtienen dos nuevas filas que son agregadas a la matriz de diseño, para un total de cuatro filas y así se continua hasta agotar todas las filas de la matriz generadora y a su vez obtener todas las filas de la matriz de diseño, a partir de esta se puede deducir el orden del diseño factorial fraccional obtenido. Esquemáticamente el procedimiento podría resumirse tal como se muestra en los siguientes pasos:

- Seleccionar las filas (generadores) de la matriz generadora para un determinado número de factores.
- Ubicar la primera fila de la matriz de diseño como aquella en la que todos los factores están a nivel bajo.
- Usar las filas de la matriz generadora y el método *foldover* generalizado para obtener la matriz de diseño de un diseño factorial fraccional.
- Obtener el orden para el diseño factorial fraccional a dos niveles con los efectos principales libres de tendencia.

Adicionalmente, en la Figura 2.9 se presenta la matriz de diseño y el orden de experimentación para un diseño  $2^4$  que fue obtenido por la matriz generadora compuesta por las filas 4, 7, 14 y 11 de la matriz con el orden estándar, la cual presenta un orden con todos los efectos principales libres de tendencia y 27 cambios de nivel de los factores. Este valor del número de cambios es bastante elevado si lo comparamos con el mínimo posible que sería de 15. Por lo cual podríamos afirmar que este orden es bastante bueno para contrarrestar los efectos de factores ajenos a la experimentación pero poco adecuado desde el punto de vista económico, dado el elevado número de cambios de nivel que es necesario realizar.

### 2.3.3 Coster (1993)

Publica 125 planes de experimentación para diseños factoriales fraccionales a dos y tres niveles con resolución V, similares a los presentados por Addelman (1965) en la *National Bureau of Standards Applied Mathematics Series*, obtenidos con el plan *foldover* generalizado desarrollado por Coster y Cheng (1988) en los cuales los efectos principales están libres de tendencia y sus factores tienen el mínimo número de cambios de nivel.

Orden	A	B	C	D
1	-	-	-	-
4	+	+	-	-
7	-	+	+	-
6	+	-	+	-
14	+	-	+	+
15	-	+	+	+
12	+	+	-	+
9	-	-	-	+
11	-	+	-	+
10	+	-	-	+
13	-	-	+	+
16	+	+	+	+
8	+	+	+	-
5	-	-	+	-
2	+	-	-	-
3	-	+	-	-
Cambios	12	9	4	2
Sesgo	0	0	0	0

Figura 2.9: Orden para un diseño  $2^{4-0}$ , hallado por el método foldover generalizado propuesto por Coster y Cheng (1988)

### 2.3.4 Jan y Wang (1995)

Presentaron algunas reglas para construir arreglos ortogonales que permiten obtener diseños factoriales completos y fraccionados libres de tendencia lineal o cuadrática en factores principales e interacciones dobles considerando de forma simultánea el número de cambios de nivel de los factores, pero sin alcanzar el mínimo posible en este último aspecto. El procedimiento consiste en 1) construir la matriz de diseño para un diseño completo en el orden estándar. 2) Organizar en orden ascendente el número de cambios de nivel de las columnas de diseño. 3) Generar la matriz completa para el diseño organizado y 4) Seleccionar las  $k$  columnas de las interacciones que presenten el menor número de cambios de nivel y asociarlas a los  $k$  factores para obtener una matriz de diseño que genere órdenes de experimentación en los diseños factoriales a dos niveles libres de la influencia de factores ajenos a la experimentación (sesgo cero) en factores principales e interacciones dobles con pocos cambios de nivel en los factores. A continuación se presenta una descripción esquemática de los pasos correspondientes a este algoritmo:

- Construir el orden estándar para el diseño  $2^k$ .
- Organizar las columnas de la matriz diseño en el orden estándar de menor a mayor número de cambios de nivel.

- Obtener el arreglo ortogonal con  $2^{k-1}$  columnas (matriz completa de la matriz del diseño anterior).
- Organizar las  $2^{k-1}$  columnas del arreglo ortogonal de menor a mayor número de cambios de nivel.
- Seleccionar  $k$  columnas de interacciones.
- Obtener el orden de experimentación para un diseño factorial  $2^k$  con los efectos principales libres de tendencia.

En la Figura 2.10 se presenta el orden obtenido por este procedimiento para el caso de un diseño  $2^4$ . Tiene todos factores principales y las interacciones dobles libres de tendencia, pero presenta 27 cambios de nivel de los factores, cuando el mínimo posible es 15.

Orden	A	B	C	D
1	-	-	-	-
13	-	-	+	+
7	-	+	+	-
11	-	+	-	+
12	+	+	-	+
8	+	+	+	-
14	+	-	+	+
2	+	-	-	-
6	+	-	+	-
10	+	-	-	+
4	+	+	-	-
16	+	+	+	+
15	-	+	+	+
3	-	+	-	-
9	-	-	-	+
5	-	-	+	-
Núm. Cambios	2	4	9	12
Sesgo	0	0	0	0

Figura 2.10: Diseño factorial  $2^4$  con los efectos principales e interacciones dobles libres de sesgo hallado con el algoritmo de Wang y Jan (1995)

### 2.3.5 Kim (1997)

Presenta en su tesis doctoral una modificación al plan *foldover* generalizado desarrollado por Coster y Cheng (1988), que denomina el método de selección sistemática de vectores *foldover*. Este método consiste en seleccionar dentro de un conjunto de posibles generadores (filas de la matriz generadora), sólo aquellos que

permitan obtener órdenes con los efectos principales y los efectos de las interacciones dobles libres de la influencia de factores ajenos a la experimentación y que a su vez provean el menor número de cambios de nivel posible en estas condiciones. De esta forma el método *foldover* no sólo considera la eliminación del sesgo en los efectos de importancia, sino que a su vez contempla el número de cambios como un elemento de interés en la construcción del orden. Adicionalmente propone un método de análisis basado en el criterio AIC (*Akaike's Information Criterion*) para los diseños obtenidos, debido a la restricción en la aleatorización que estos órdenes presentan.

### 2.3.6 Tack y Vandebroek (2002)

Presentan una modificación del algoritmo de Atkinson y Donev (1996) para obtener diseños de costo eficiente (es decir, bajo número de cambios en los niveles de los factores) y libres de tendencia en los efectos principales. El algoritmo se compone de tres fases: en la primera se asignan unas filas al diseño de acuerdo con los deseos del experimentador, en la segunda fase se completa el número total de filas de forma que se obtenga un criterio D-óptimo bajo y por último en la tercera fase se realiza el intercambio de filas del diseño con el fin de mejorar el criterio de D-óptimo con el que fue alimentado.

Este método ofrece una forma general de resolver problemas en los que es necesario encontrar un balance entre unos diseños baratos pero ineficientes, y otros muy costosos pero con buena resistencia a la tendencia. Sin embargo, no ofrece una guía de actuación fácil de utilizar y de poner en práctica por los experimentadores.



A continuación se presentan las propuestas que priorizan el número de cambios de nivel y que a la vez atenúan la influencia de factores ajenos a la experimentación en los efectos principales o en los efectos principales y las interacciones tales como los trabajos de Dickinson (1974), de León (2004) y de León et al. (2005).

### 2.3.7 Dickinson (1974)

Dada la necesidad de ampliar la propuesta de Draper y Stoneman (1968), que presentaba órdenes de experimentación para diseños  $2^3$  con las propiedades de tener pocos cambios de nivel y los efectos principales casi libres de la influencia lineal debida a factores ajenos a la experimentación, Dickinson (1974) continuó la búsqueda de órdenes con características similares para diseños  $2^4$  y  $2^5$ , salvando de manera

adecuada la restricción computacional debida a la elevada cantidad de órdenes que era necesario explorar. Así, limitó la búsqueda de los órdenes que atenuaban la influencia lineal en los efectos principales a sólo aquellos que presentaban el mínimo número de cambios de nivel, pues con base en los trabajos de Tiahrt y Weeks (1970) se conocía que este número estaba predeterminado para los diseños factoriales a dos niveles, por la expresión  $2^k - 1$ , ya que cuando sólo cambia un nivel de los factores en cada par de pruebas consecutivas de un diseño factorial a dos niveles el número de cambios es mínimo.

Para evaluar la influencia de los factores no controlados en los efectos principales Dickinson utilizó como medida el criterio “*Time Count*”<sup>1</sup>, presentado por Draper y Stoneman (1968), que consistía en el producto interno entre la columna de un efecto particular y vector de respuesta que simula la influencia lineal de los factores no controlados. Para mantener la notación de nuestro trabajo utilizaremos como medida de esta influencia el valor del sesgo de forma similar a la utilizada por de León (2004), el cual puede ser calculado con la siguiente expresión:

$$\text{Sesgo} = \frac{\text{Time Count}}{2^{k-1}}$$

En la Figura 2.11 se presenta un ejemplo de cómo pueden ser utilizadas estas medidas (número de cambios y sesgo) para evaluar el diseño factorial  $2^4$  en orden estándar en relación a estos dos criterios. Puede observarse que el número de cambios de nivel que presenta este diseño es de 26 y la influencia de los factores no controlados en los cuatro efectos principales, medida con el sesgo es de 1, 2, 4, y 8, respectivamente. Obsérvese que el número de cambios de nivel es grande, si lo comparamos con el mínimo número de cambios posibles en este tipo de diseños, que es de 15 cambios. Adicionalmente, el sesgo máximo obtenido en el factor D, es de ocho (8), el mayor valor del sesgo que se puede presentar en un diseño  $2^4$ , ya que las ocho primeras observaciones se realizan con este factor a nivel bajo y las ocho restantes con el nivel alto.

---

<sup>1</sup> Este término fue creado por Draper y Stoneman (1968), para denominar el resultado del producto interno de la columna de respuesta (1, 2, 3,...,N) y un efecto deseado con el nombre de contador de tiempo o medida de la influencia lineal de los factores ajenos a la experimentación.

Orden de experimentación	A	B	C	D	Respuesta
1	-	-	-	-	1
2	+	-	-	-	2
3	-	+	-	-	3
4	+	+	-	-	4
5	-	-	+	-	5
6	+	-	+	-	6
7	-	+	+	-	7
8	+	+	+	-	8
9	-	-	-	+	9
10	+	-	-	+	10
11	-	+	-	+	11
12	+	+	-	+	12
13	-	-	+	+	13
14	+	-	+	+	14
15	-	+	+	+	15
16	+	+	+	+	16
Sesgo	1	2	4	8	
Número de cambios	15	7	3	1	

Figura 2.11: Evaluación del diseño  $2^4$  con el orden estándar en cuanto a número de cambios de nivel y el sesgo máximo en los efectos principales

Ahora si deseamos hallar órdenes en los diseños factoriales  $2^k$ , en los cuales el número de cambios de nivel sea el mínimo y la influencia lineal ejercida por los factores no controlados también lo sea, es necesario evaluar las  $k!$  permutaciones de las columnas o factores, lo que genera un número de posibilidades exageradamente grande, por lo que es necesario limitar la búsqueda a un conjunto estratégico de órdenes, como aquellos en los cuales el número de cambios de nivel de los factores sea mínimo, para posteriormente evaluar los sesgos en los efectos principales de los órdenes hallados.

En el capítulo 4 se presenta una descripción detallada, a nivel operativo, del algoritmo que propone Dickinson. Con su aplicación a los diseños factoriales  $2^4$  se pueden hallar la totalidad de órdenes semilla (238) que presentan el mínimo número de cambios de nivel (15 cambios) en este tipo de diseños. Cada uno de estos órdenes fue evaluado con base en el sesgo máximo presente en los efectos principales y los siete órdenes con el valor más bajo del sesgo máximo se pueden observar en la Figura 2.12. Estos órdenes

tienen la característica de poseer el mínimo costo de experimentación y proteger de la influencia lineal de factores ajenos al experimento en los diseños factoriales  $2^4$ , simultáneamente. Es necesario aclarar que con cada uno de los órdenes presentados en la Figura 2.12 se pueden generar 384 órdenes adicionales, al permutar las columnas y/o cambiar los signos de los elementos de una columna o conjunto de columnas.

De manera similar Dickinson utilizó el algoritmo descrito anteriormente para hallar órdenes de experimentación con el mínimo número de cambios de nivel de los factores en los diseños factoriales  $2^5$ , pero dada la excesiva cantidad de órdenes que deben ser explorados ( $2.63 \times 10^{35}$ ), la búsqueda exhaustiva aún en el caso de restringirla sólo a los diseños con el mínimo número de cambios de nivel era inviable para esa época, por lo cual se truncó la búsqueda. Para el caso particular del trabajo realizado por Dickinson (1974) la búsqueda fue interrumpida cuando se habían obtenido 26.000 órdenes de experimentación con el mínimo número de cambios de nivel de los factores y con estos se realizó la evaluación del mínimo sesgo máximo en los efectos principales. El mínimo sesgo máximo obtenido para los efectos principales fue de 2,25 en cuatro órdenes de experimentación, 2,5 en un orden de experimentación y 2,75 en diez órdenes de experimentación, en la Figura 2.13 se presentan dichos órdenes.

		Órdenes						
		a	b	c	d	e	f	g
		1	1	1	1	1	1	1
		2	2	2	2	2	2	2
		10	10	6	6	10	10	10
		14	14	14	14	14	14	12
		16	16	16	16	16	16	11
		15	15	15	15	15	12	15
		7	7	11	11	11	11	16
		3	3	3	3	12	15	14
		11	4	4	7	4	7	6
		12	8	8	5	3	3	8
		4	6	7	13	7	4	4
		8	5	5	9	8	8	3
		6	13	13	10	6	6	7
		5	9	9	12	5	5	5
		13	11	10	4	13	13	13
		9	12	12	8	9	9	9
Mínimo sesgo máximo en efectos principales		2	2	3	3	3	3	3,5

Figura 2.12: Órdenes de experimentación en el diseño factorial  $2^4$  con el mínimo sesgo máximo en los efectos principales y con 15 cambios de nivel de los factores

Órdenes de experimentación														
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
18	18	18	18	18	18	18	18	18	18	18	18	18	18	18
20	20	20	20	20	20	20	20	20	20	20	20	20	20	20
28	28	28	28	28	28	28	28	28	28	28	28	28	28	28
32	32	32	32	32	32	32	32	32	32	32	32	32	32	32
31	31	31	31	31	31	31	31	31	31	31	31	31	31	31
15	15	15	15	15	15	15	15	15	15	15	15	15	15	15
11	11	11	11	16	11	11	11	11	11	16	16	11	16	16
3	3	3	3	14	3	3	3	3	3	8	12	3	12	14
7	7	7	7	6	7	7	7	7	7	4	4	7	4	13
5	5	5	5	22	5	5	5	5	5	3	3	5	3	29
6	6	6	6	24	21	6	21	21	6	7	7	6	7	25
22	22	22	22	23	29	22	29	29	22	5	5	22	5	27
21	21	30	30	19	13	21	30	30	30	13	13	30	21	19
29	29	29	26	27	9	29	26	26	29	9	29	26	22	3
30	30	25	25	11	10	30	10	10	13	25	21	10	30	7
26	26	26	29	3	26	26	9	9	9	17	22	9	29	5
10	10	10	13	7	30	10	13	13	25	21	30	25	13	6
9	9	9	9	5	22	9	14	14	26	29	26	29	9	8
13	13	13	10	13	6	13	16	16	10	30	25	13	11	4
14	14	14	14	9	14	14	12	12	14	26	27	14	27	12
16	16	16	16	25	16	16	4	4	16	10	11	16	25	11
12	12	12	12	17	12	12	8	8	12	14	9	12	26	9
4	4	4	4	21	4	4	6	6	4	6	10	4	10	10
8	8	8	8	29	8	8	22	22	8	22	14	8	14	26
24	24	24	24	30	24	24	24	24	24	24	6	24	6	30
23	23	23	23	26	23	23	23	23	23	23	8	23	8	22
19	19	21	21	10	19	19	19	19	21	19	24	21	24	24
27	17	17	17	12	27	17	17	27	17	27	23	17	23	23
25	25	19	19	4	25	25	25	25	19	11	19	19	19	21
17	27	27	27	8	17	27	27	17	27	12	17	27	17	17
Mínimo Sesgo máximo En efectos principales	2,25	2,25	2,25	2,25	2,5	2,75	2,75	2,75	2,75	2,75	2,75	2,75	2,75	2,75

Figura 2.13: Órdenes de experimentación en el diseño factorial  $2^5$  con el mínimo sesgo máximo en los efectos principales y con 31 cambios de nivel de los factores

Dada las evidentes restricciones que este método presenta para la búsqueda de órdenes con el mínimo número de cambios de nivel, debido al elevado número de combinaciones que deben ser analizadas, la búsqueda de órdenes con características similares en diseños factoriales a dos niveles con un mayor número de factores ( $2^6$ ,  $2^7$ , ...), no ha sido posible, por lo cual sigue teniendo vigencia el comentario hecho por Dickinson (1974) en el numeral de posibles extensiones, donde afirma que: “La carga computacional podría ser reducida considerablemente si se pudiera encontrar una forma sistemática de generar órdenes que posean buenas propiedades sin generar

*órdenes adicionales con propiedades no adecuadas*". Este comentario es uno de los puntos de partida para el desarrollo de esta tesis doctoral.

### **2.3.8 de León (2004) y de León et al. (2005)**

Una reciente modificación al trabajo propuesto por Dickinson (1974) fue presentada por de León et al. (2005), la cual consistió en hallar órdenes de experimentación en los diseños factoriales a dos niveles con 8 y 16 experimentos que presentan el mínimo número de cambios de nivel de los factores, utilizando para ello el algoritmo de búsqueda propuesto por Dickinson (1974). Una vez obtenidos dichos órdenes se procedió a evaluar el mínimo sesgo máximo en todos los efectos del diseño y no sólo el de los efectos principales como lo proponía Dickinson (1974), con el fin de que estos órdenes no sólo fueran útiles para la atenuación de la influencia de los factores ajenos a la experimentación en diseños factoriales a dos niveles completos, sino también para diseños fraccionados, de forma similar al trabajo propuesto por Draper y Stoneman (1966), pero restringiendo los órdenes al mínimo número de cambios de nivel de los factores. El total de órdenes que presentan las características de tener el mínimo número de cambios de nivel y el mínimo sesgo máximo en valor absoluto en todos los efectos, fueron 48 en el diseño con 8 experimentos y 16 en el diseño con 16 experimentos (ver Anexo 1). Dos órdenes del diseño  $2^4$  hallados por de León (2004) coinciden con los órdenes e y f de la Figura 2.13, hallados por Dickinson (1974), lo que nos permite afirmar que dichos órdenes no solo atenúan la influencia en los efectos principales, sino también en todos los efectos del modelo.

Uno de los aportes de este trabajo fue el reinterpretar el concepto del *Time Count* propuesto por Draper y Stoneman (1968) para medir de la influencia de los factores ajenos a la experimentación a través de un concepto más claro para usuarios no expertos, como es el sesgo, y a partir de este concepto determinar la forma en que su influencia podía ser homogenizada entre todos los efectos del modelo y no sólo en los efectos principales.

Dado que los órdenes hallados por de León et al. (2005) son el resultado de una búsqueda exhaustiva, se puede identificar claramente que el mínimo sesgo máximo en todos los efectos del modelo obtenidos en los diseños factoriales con 8 experimentos (diseños factoriales  $2^3$ ) es igual 2, cuando se restringe al mínimo número de cambios de nivel de los factores. De forma similar el mínimo sesgo máximo para los diseños factoriales con 16 experimentos (diseños factoriales  $2^4$ ) es igual a 4. Dichos valores permiten atenuar de forma importante la influencia de los factores ajenos a la experimentación cuando se fuerza el mínimo número de cambios de nivel de los

factores. Esta situación será de gran importancia a la hora de evaluar la eficacia de los órdenes obtenidos con un nuevo algoritmo.

La búsqueda de órdenes con las características de tener el mínimo número de cambios de nivel y el mínimo sesgo máximo en diseños factoriales a dos niveles para un mayor número de experimentos o factores, presenta los mismos inconvenientes descritos en la propuesta de Dickinson (1974), dado que el algoritmo de búsqueda es el mismo y lo que varía es la forma de evaluar el sesgo en los efectos del modelo, por lo cual para continuar con la búsqueda de este tipo de órdenes es necesario construir un nuevo algoritmo que salve dicha restricción.

Vale la pena destacar el comentario realizado por León (2004) en el capítulo 6, de resultados y futuras líneas de investigación, de su tesis doctoral, donde dice: “*Sería conveniente tener órdenes de experimentación con el mínimo sesgo y con un mínimo número de cambios de nivel, para diseños con un número mayor de experimentos*”. Siendo este uno de los principales elementos que motivan a la realización de esta tesis doctoral.



Seguidamente se presentan las propuestas que persiguen un equilibrio entre el número de cambios de nivel de los factores y el sesgo en los efectos principales, de forma que no se minimice ninguno de los dos aspectos, sino que se obtenga un valor bajo en ambos, tal como los trabajos presentados por Draper y Stoneman (1968) y Joiner y Campbell (1976).

### **2.3.9 Draper y Stoneman (1968)**

Tal vez la primera insinuación acerca de la necesidad de tener órdenes de experimentación con el doble propósito de proteger contra la influencia de factores ajenos a la experimentación y con un bajo costo, se debe al comentario realizado por el señor Grant Wernimont, citado en Daniel y Wilcoxon, 1966, en cuanto hacer más económicos los órdenes de ejecución propuestos en el trabajo de estos mismos autores, los cuales sólo buscaban la eliminación de la influencia de los factores no controlados en los efectos principales del diseño factorial.

Pero fueron Draper y Stoneman (1968), los primeros autores en proponer la búsqueda de órdenes de experimentación en diseños factoriales  $2^3$  buscando un equilibrio entre el número de cambios de nivel de los factores y la atenuación de la influencia lineal que

los factores no controlados pueden ejercer en los efectos principales. Hasta ese momento los esfuerzos se habían concentrado en hallar órdenes de experimentación que eliminaran completamente la influencia de factores no controlados en los efectos principales del modelo, en especial el trabajo presentado por Daniel y Wilcoxon (1966).

Este trabajo fue pionero también, en definir que la influencia lineal de los factores no controlados podría ser simulada al colocar como respuesta de cada uno de los tratamientos el valor de la fila de ejecución de los experimentos. Por ejemplo, para el caso de los diseños factoriales  $2^3$  estas respuestas serían: 1, 2, ..., 8, respectivamente. Adicionalmente definieron que la influencia de los factores no controlados podría ser obtenida a través del producto interno entre la columna del efecto deseado y la columna de respuesta, a esta medida la denominaron "*Time Count*", que puede ser interpretada como el contador de la influencia lineal ejercida en el tiempo.

Para realizar la búsqueda de los órdenes de experimentación que cumplen el doble propósito de proteger de la influencia de factores no controlados y a su vez tener un bajo número de cambios de nivel de los factores en los diseños factoriales  $2^3$ , construyeron un programa de ordenador que permitiera explorar la totalidad de órdenes ( $8! = 40.320$ ), clasificándolos en dos aspectos, el número de cambios de nivel y el máximo *Time Count* en valor absoluto, para nuestro caso particular este criterio sería el mínimo sesgo máximo en valor absoluto, tal como lo usó de León (2004).

Los 40.320 órdenes fueron divididos en 840 grupos de 48 órdenes equivalentes cada grupo. Los 48 órdenes equivalentes que pueden ser obtenidos al permutar las columnas de los factores (seis formas) y al cambiar la posición de los signos de una o más columnas (8 formas;  $6 \times 8 = 48$ ). En la Figura 2.14 se presentan los mejores órdenes obtenidos para el diseño factorial completo  $2^3$  en cuanto a un número bajo de cambios de nivel y un bajo sesgo máximo en los efectos principales.

Si tomamos uno de los órdenes presentados en la Figura 2.14, por ejemplo, el orden 1, 4, 8, 6, 5, 2, 3, 7 y lo analizamos en relación al número de cambios de nivel de los factores y el sesgo máximo, se obtienen un total de 10 cambios de nivel y un sesgo máximo de 1,5 en los efectos principales, pero un sesgo de 3,5 en el efecto de la interacción *AB* si consideramos todos los efectos del modelo, en la Figura 2.15 se pueden observar los efectos principales y el número de cambios en los factores correspondientes al orden descrito.

Órdenes											
1	1	1	1	1	1	1	1	1	1	1	1
4	2	4	2	4	4	4	4	4	4	8	4
8	8	8	8	8	8	6	6	8	8	4	6
6	7	5	3	6	5	8	8	6	6	6	8
5	3	6	7	5	6	7	5	5	7	5	7
7	5	2	5	2	7	5	7	3	5	7	3
3	6	3	6	3	3	3	3	7	3	3	5
2	4	7	4	7	2	2	2	2	2	2	2

Figura 2.14: Órdenes de experimentación con pocos cambios de nivel y bajo sesgo en el diseño  $2^3$  hallados por Draper y Stoneman (1968)

Orden	A	B	C
1	-1	-1	-1
4	1	1	-1
8	1	1	1
6	1	-1	1
5	-1	-1	1
2	1	-1	-1
3	-1	1	-1
7	-1	1	1
Cambios	4	3	3
Sesgo	1,5	1	1

Figura 2.15: Número de cambios de nivel y sesgo máximo en efectos principales en uno de los órdenes del diseño  $2^3$ , hallados por Draper y Stoneman (1968)

Como se puede observar en la Figura 2.15 estos órdenes no presentan el valor mínimo en ninguno de los aspectos, ya que, el mínimo número de cambios que pueden ser alcanzado en estos diseños es de 7 y el mínimo sesgo máximo un valor 0 en los efectos principales. Pero estas dos condiciones no pueden ser alcanzadas de forma simultánea por lo cual se debe buscar un equilibrio en estos dos aspectos, según la propuesta de Draper y Stoneman (1968).

Procediendo de forma similar a lo presentado en el caso de los diseños factoriales completos se realizó el estudio para los diseños factoriales fraccionales, comenzando por los diseños  $2^{4-1}$ . Ahora los 40.320 órdenes fueron divididos en 210 grupos distintos

con 192 órdenes equivalentes por grupo. El número equivalente de órdenes puede ser obtenido por la permutación de las columnas de los factores A,B,C y la interacción ABC que representa al cuarto factor o factor D ( $4! = 24$  formas) y cambiar la posición de los signos de una o más columnas (8 formas;  $24 \times 8 = 192$ ). En la Figura 2.16, se presentan los mejores órdenes obtenidos para este tipo de diseños.

Órdenes																
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
2	2	2	2	2	8	2	8	8	2	8	8	2	2	2	8	8
8	8	7	8	8	2	8	2	2	8	2	2	7	7	8	2	2
7	3	8	3	7	3	7	3	7	7	3	7	8	3	3	3	3
3	7	3	7	3	7	3	7	3	3	7	3	3	8	6	6	7
5	5	5	6	5	5	6	6	5	6	6	4	5	6	7	5	5
6	6	6	5	4	6	5	5	6	4	4	6	4	5	5	7	4
4	4	4	4	6	4	4	4	4	5	5	5	6	4	4	4	6

Figura 2.16: Órdenes de experimentación con pocos cambios de nivel y bajo sesgo en el diseño  $2^{4-1}$  hallados por Draper y Stoneman (1968)

Continuando con la búsqueda de los órdenes de experimentación en diseños factoriales fraccionales con ocho experimentos, Draper y Stoneman examinaron los órdenes para los diseños  $2^{5-2}$ , obteniéndose los órdenes que se muestran en la Figura 2.17. Adicionalmente se evaluaron los órdenes para los diseños  $2^{6-3}$ , donde fueron encontrados los órdenes que se presentan en la Figura 2.18. Finalmente se evaluaron los órdenes para los diseños saturados  $2^{7-4}$  obteniéndose los órdenes que se presentan en la Figura 2.19.

Órdenes															
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
2	3	3	2	7	2	7	2	2	7	2	2	2	2	2	
4	7	2	7	2	7	2	7	7	2	7	7	8	8	8	
8	2	6	3	4	8	4	4	4	4	8	8	7	3	3	
6	6	7	8	8	4	8	8	8	8	3	4	3	7	7	
5	8	5	6	6	5	6	5	6	5	6	5	6	5	6	5
7	4	8	5	5	6	3	6	5	6	5	3	5	6	6	
3	5	4	4	3	3	5	3	3	3	4	6	4	4	4	

Figura 2.17: Órdenes de experimentación con pocos cambios de nivel y bajo sesgo en el diseño  $2^{5-2}$  hallados por Draper y Stoneman (1968)

Órdenes															
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	4	2	4	2	4	4	4	4	4	4	4	4	2	4	2
4	2	4	2	3	6	6	2	6	6	2	6	2	6	4	6
5	6	6	6	6	2	2	5	2	2	6	2	6	2	6	2
7	8	8	8	7	7	7	7	7	7	7	8	7	8	7	8
8	7	7	5	8	8	5	6	5	8	5	7	8	7	8	5
6	5	5	7	5	5	3	8	8	3	8	3	5	5	5	7
3	3	3	3	4	3	8	3	3	5	3	5	3	3	3	4

Figura 2.18: Órdenes de experimentación con pocos cambios de nivel y bajo sesgo en el diseño  $2^{6-3}$ , hallados por Draper y Stoneman (1968)

Órdenes															
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
8	7	7	8	7	8	7	8	6	6	6	7	6	7	6	8
7	6	8	6	6	6	8	7	4	4	8	4	7	4	7	4
6	4	4	7	8	4	6	4	8	7	4	8	4	6	8	6
4	8	6	4	4	7	4	6	7	8	7	6	8	8	4	7

Figura 2.19: Órdenes de experimentación con pocos cambios de nivel y bajo sesgo en el diseño  $2^{7-4}$  hallados por Draper y Stoneman (1968)

Al finalizar su trabajo en los diseños factoriales a dos niveles con ocho experimentos, Draper y Stoneman hicieron el siguiente comentario: “Con los actuales equipos de computación, el análisis exhaustivo de los diseños con 16 o más experimentos parece ser un tarea formidable en términos de tiempo computacional, por el gran número de posibles órdenes que pueden ocurrir. Por esta razón no hemos extendido este trabajo a diseños factoriales a dos niveles con más experimentos”. Este comentario fue la base de la propuesta de Dickinson (1974) para búsqueda de órdenes de experimentación en diseños factoriales  $2^4$  y  $2^5$  que presentaran el mínimo número de cambios de nivel y el mínimo sesgo máximo en los efectos principales.

### 2.3.10 Joiner y Campbell (1976)

Presentan una serie de ejemplos en los cuales se observa claramente la influencia que pueden ejercer factores desconocidos relacionados con el tiempo, tales como la descripción del ejemplo propuesto por Draper y Smith (1966) en el que se presenta un experimento para evaluar el efecto de tres factores sobre la cantidad de vitamina B en los nabos. Los tres factores son: la radiación solar, la mezcla de tierra y la temperatura. Al analizar detalladamente los datos obtenidos en la respuesta se encuentra que al establecer una regresión lineal simple entre  $y_i$  (respuesta) y  $i$  (posición) se obtiene un 90% de correlación. En otras palabras, que los valores de la vitamina B disminuyen casi linealmente con el orden en que se realizan los experimentos. Lo que nos lleva a pensar que por alguna razón los resultados disminuyen con el tiempo (¿disminuye la vitamina? o ¿Disminuyen los reactivos?).

Otro ejemplo, corresponde a un trabajo encomendado por la agencia de estándares a Brian Joiner para diseñar un experimento que permita evaluar la precisión de una instalación que había sido construida para calibrar los medidores usados para venta de fluidos criogénicos (nitrógeno líquido, oxígeno líquido, etc.). Se sabe que los medidores se deterioran con el uso, y se consideraba tanto la instalación como los medidores que podían tener la posibilidad de ser sensibles a la temperatura, presión y a la velocidad de flujo del líquido que estaba siendo medido. Otra variable de interés en la valoración de la instalación fue el peso total del líquido bombeado durante la prueba.

Comúnmente un estadístico puede sugerir que las medidas deben hacerse en orden aleatorio. Pero en este caso, cambiar la temperatura consume mucho tiempo (alto costo) y hasta cierto punto la presión también, pero cambiar la velocidad y el peso podría hacerse de forma instantánea. Una vez se tiene preparadas todas las variables para una prueba, la medición podría hacerse en 5 minutos. Pero cambiar la temperatura podría consumir medio día, por lo cual una aleatorización del orden fue descartada, pero el orden óptimo de experimentación no es obvio.

Para solucionar los problemas mencionados en los párrafos anteriores, se propone una estrategia de búsqueda de diseños factoriales a dos niveles, a través de la selección aleatoria de 50 órdenes de experimentación y a partir de estos escoger el mejor. Para realizar esta selección, todos los órdenes son evaluados con una medida de eficiencia definida como la razón entre una medida de calidad y un coste, que puede ser representada por la expresión  $\frac{|\mathbf{X}'\mathbf{X}|^{1/p}}{\text{Costo}}$ , donde  $p$  es el número de vectores en  $\mathbf{X}$  y el valor del coste representa la sumas de los costes incurridos en el cambio de nivel de los factores y el costo de las mediciones. Esta medida se complementa con un gráfico

bivariante entre la calidad y el coste de cada uno de los órdenes obtenidos, con el fin de buscar un equilibrio entre el coste de la experimentación y la atenuación del sesgo, para facilitar la identificación y selección del mejor o mejores órdenes, el diagrama bivariante pueden ser acotado. Esquemáticamente el procedimiento es definido así:

- Seleccionar aleatoriamente 50 órdenes de experimentación de un diseño factorial  $2^k$ .
- Calcular la medida de calidad  $|\mathbf{X}'\mathbf{X}|^{1/p}$ .
- Calcular el número de cambios de nivel de los factores.
- Hacer un gráfico bivariante con la medida de calidad y el número de cambios de nivel.
- Seleccionar el orden que presente simultáneamente la medida de calidad alta y el número de cambios de nivel de los factores bajo.

En la Figura 2.20, reproducida a partir de los valores que figuran en Joiner y Campbell (1976), se presenta un ejemplo del diagrama bivariante al evaluar la medida de calidad ( $=|\mathbf{X}'\mathbf{X}|^{1/p}$ ) y los costos (igual al número de cambios en los factores) en los efectos principales sobre 50 órdenes de un diseño factorial  $2^3$ . El punto marcado con una flecha corresponde al orden que debiera ser escogido, ya que presenta una medida de calidad alta con un número de cambios de nivel bajo. De acuerdo con los comentarios dados por Joiner y Campbell (1976), este orden coincide con uno de los órdenes hallados por la búsqueda exhaustiva realizada por Draper y Stoneman (1968).

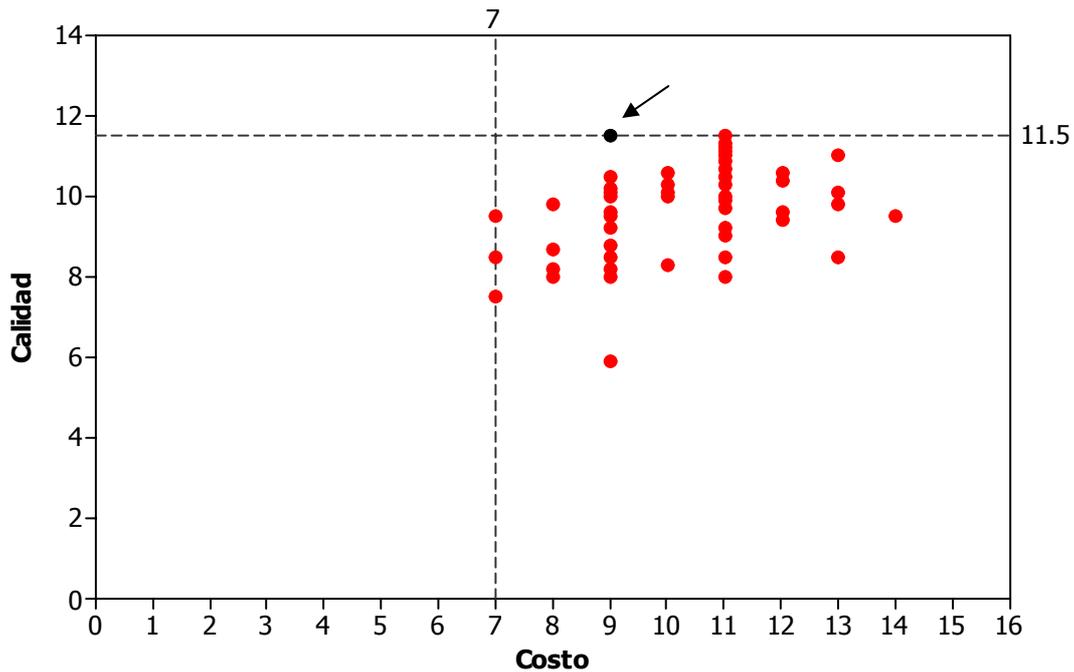


Figura 2.20 Diagrama Bivariante para evaluar 50 órdenes de experimentación en los diseños factoriales  $2^3$  obtenido con el método propuesto por Joiner y Campbell (1976)

## 2.4 Resumen

De lo descrito en la revisión bibliográfica, se puede deducir que existen tres formas de abordar el problema de la búsqueda de órdenes de experimentación cuando se quieren cumplir simultáneamente las condiciones de eliminar o minimizar la tendencia debida a los factores no controlados y mantener un coste de experimentación bajo. La primera da prioridad al problema de eliminar la tendencia y posteriormente considera el número de cambios de nivel de los factores. La segunda, da prioridad al mínimo número de cambios de nivel de los factores y luego se ocupa de la atenuación de la tendencia debida a los factores no controlados. Y la tercera que busca un equilibrio en ambos aspectos. Estas estrategias se han desarrollado debido a que no es posible alcanzar de forma conjunta el mínimo en ambos criterios, como fue demostrado por Cheng (1985).

Adicionalmente, con base en la revisión bibliográfica exhaustiva del problema de aleatorización de los órdenes de experimentación cuando son importantes el coste y la influencia de los factores no controlados, podemos decir, que hasta el momento ha sido resuelto de forma general para diseños factoriales  $2^{k-p}$ , el problema que busca eliminar

completamente la influencia de los factores no controlados tanto en efectos principales como interacciones dobles, pero sin minimizar de forma simultánea el número de cambios de nivel en los factores. Igualmente, ha sido resuelto de forma general el problema que busca minimizar el número de cambios de nivel de los factores, pero sin considerar la influencia de factores no controlados. Falta todavía resolver el problema de minimizar el número de cambios de nivel y a su vez proteger de la influencia debida a los factores no controlados en todos los efectos del modelo, ya que hasta el momento sólo se ha resuelto para diseños factoriales a dos niveles con 8 y 16 experimentos.

Teniendo presente estas tres formas de abordar el problema y lo descrito en el párrafo anterior consideramos que es posible continuar el proceso de investigación en el tema de aleatorización de los órdenes de experimentación, con el segundo enfoque. Desarrollando un método que permita encontrar órdenes de experimentación que cumplan simultáneamente los criterios de poseer el mínimo número de cambios en los niveles de los factores y el mínimo sesgo máximo en valor absoluto en la estimación de los efectos. Como un nuevo aporte en esta línea de trabajo. Por ejemplo para el caso de un diseño factorial con ocho experimentos se podrían hallar la totalidad (48) órdenes de experimentación que cumplen estas características con respecto a los 40.320 posibles órdenes que posee este diseño.

Las aportaciones comentadas en esta revisión bibliográfica se presentan resumidas en la Figura 2.21.

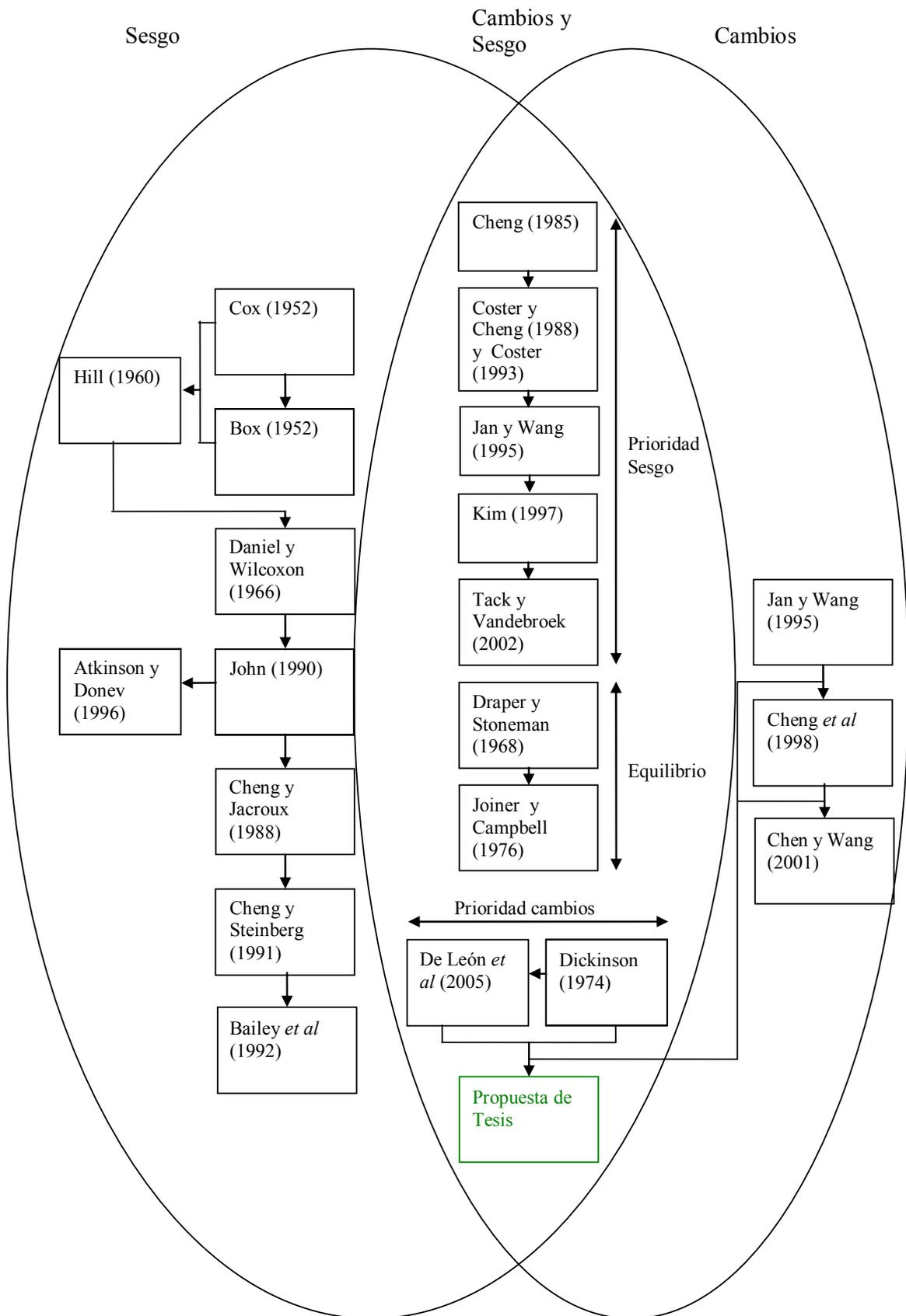


Figura 2.21: Estado del arte y antecedentes

## 2.5 Algunas debilidades detectadas

Después del análisis de las propuestas, se encuentran las siguientes debilidades que pueden ser el punto de partida de la tesis que se propone:

- Las metodologías propuestas que dan prioridad al problema de la tendencia y no al coste, generan demasiados cambios de nivel de los factores para realizar la experimentación (70% o más cambios con respecto al mínimo número de cambios posibles), lo que representa un aumento importante de los costes a la hora de realizar la experimentación.
- Con relación a las metodologías que dan prioridad al mínimo número de cambios de nivel y posteriormente a la minimización de la tendencia, sólo se han encontrado hasta el momento todos los órdenes para diseños factoriales con 8 y 16 experimentos por búsqueda exhaustiva y algunos órdenes para diseños con 32 experimentos con un método poco eficiente de búsqueda computacional restringida.
- Para los diseños factoriales  $2^{k-p}$  con  $k-p > 5$  no se conocen órdenes de experimentación que presenten el mínimo número de cambios y el mínimo sesgo máximo en valor absoluto para la estimación de los efectos.
- No se han elaborado hasta la fecha algoritmos eficientes, que permitan obtener órdenes de experimentación con el mínimo número de cambios de nivel que protejan de la influencia de factores desconocidos.
- Las metodologías propuestas para la búsqueda de órdenes de experimentación libres de tendencia son demasiado complejas, lo que ha impedido su uso práctico en la industria.
- Existe también un motivo económico para la búsqueda de órdenes con un mayor número de factores, ya que la diferencia entre el número mínimo de cambios posibles ( $2^{k-1}$ ) y el número esperado de cambios ( $k \cdot 2^{k-1}$ ) crece, al aumentar la cantidad de factores [Dickinson (1974)]. Así el potencial de ahorro aumenta cuando el número de factores aumenta. Ventaja que no se ha podido aprovechar, dado que no se conocen dichos órdenes para  $k > 5$ .
- Los algoritmos conocidos para obtener órdenes con el mínimo número de cambios de nivel no consideran simultáneamente la tendencia como criterio a tener en cuenta, lo que genera órdenes de experimentación con el máximo sesgo posible, es decir, no adecuados cuando existe tendencia en los factores no controlados.

## 2.6 Posibles aportes

Con base en las debilidades detectadas podemos definir los siguientes aportes que pueden ser alcanzados con esta tesis:

- Simplificar la búsqueda de diseños factoriales a dos niveles que presentan el mínimo número de cambios de nivel de los factores y el mínimo sesgo máximo en valor absoluto.
- Obtener órdenes de experimentación en diseños factoriales  $2^k$  con el mínimo número de cambios de nivel de los factores que protejan contra la tendencia lineal para cualquier valor de  $k$ .
- Facilitar la utilización de estos órdenes en la práctica industrial con la ayuda de software que haga posible su uso por parte de usuarios no expertos.
- Combinar y generalizar las propuestas presentadas por Dickinson (1974) y de De León et al. (2005) para cualquier diseño factorial a dos niveles, ya que hasta el momento sólo se conocen órdenes para diseños factoriales a dos niveles con 8, 16 y, parcialmente, 32 experimentos.
- Permitir que estos desarrollos investigativos trasciendan del campo teórico al aplicado y puedan ser utilizados en las experimentaciones industriales cotidianas sin requerir grandes conocimientos estadístico para que sean aprovechadas las ventajas que estos diseños pueden ofrecer.

# Capítulo 3



# Capítulo 3

## Generación de órdenes de experimentación con mínimo número de cambios y sesgo conocido

En este capítulo se presenta un procedimiento que permite encontrar órdenes de experimentación para diseños factoriales a 2 niveles, con el mínimo número de cambios en los niveles de los factores y un sesgo conocido. La característica más importante de este procedimiento es que puede ser aplicado a diseños factoriales a 2 niveles con cualquier número de factores.

### 3.1 Introducción

La búsqueda de órdenes de experimentación con las propiedades deseadas de mínimo número de cambios de nivel y mínimo sesgo máximo en valor absoluto, se ha realizado hasta ahora a través de una exploración computacional exhaustiva [Dickinson (1974) y de León et al. (2005)] lo que ha limitado su hallazgo hasta los diseños  $2^4$  inclusive. Por otra parte, se han desarrollado algoritmos complejos para hallar órdenes en diseños factoriales  $2^k$  con el mínimo número de cambios nivel, pero sin considerar de forma simultánea la influencia debida a factores no controlados [Wang y Jan (1995); Cheng et al. (1998) y Chen y Wang (2001)].

También se han desarrollado procedimientos para hallar órdenes que protegen contra tendencias en los factores desconocidos, pero que no consideran el número de cambios de nivel de los factores y presentan un número muy alto de este valor respecto al mínimo posible. Autores que han trabajado con este tipo de enfoque son Daniel y

Wilcoxon (1966), Cheng y Jacroux (1988), John (1990), Cheng y Steinberg (1991), Atkinson y Donev (1996) y Bailey (1992).

El objetivo de este capítulo es plantear un procedimiento que contemple simultáneamente las dos propiedades deseadas y que sea generalizable a cualquier diseño factorial, independiente del número de factores que contenga.

### 3.2 Obtención de los órdenes de experimentación para diseños $2^4$

Partimos de los órdenes para un diseño con 8 experimentos que cumplen con las propiedades deseadas y que pueden ser obtenidos mediante una exploración exhaustiva de todos los órdenes posibles tal como realizó de León (2004).

Sea, por ejemplo, el orden 1, 2, 6, 8, 4, 3, 7, 5 (Figura 3.1) que aparece en primer lugar en la tabla propuesta por de León, ver Anexo 1.

Orden estándar	Factores		
	A	B	C
1	-1	-1	-1
2	1	-1	-1
6	1	-1	1
8	1	1	1
4	1	1	-1
3	-1	1	-1
7	-1	1	1
5	-1	-1	1
Cambios	2	2	3

Figura 3.1: Diseño  $2^3$  con un orden de ejecución que cumple con la propiedad de mínimo número de cambios en los factores y mínimo sesgo máximo (igual a 2)

Partiendo de esta matriz construimos otra duplicando las filas, tal como se presenta en la Figura 3.2.

Orden	Factores		
	A	B	C
1	-1	-1	-1
1	-1	-1	-1
2	1	-1	-1
2	1	-1	-1
6	1	-1	1
6	1	-1	1
8	1	1	1
8	1	1	1
4	1	1	-1
4	1	1	-1
3	-1	1	-1
3	-1	1	-1
7	-1	1	1
7	-1	1	1
5	-1	-1	1
5	-1	-1	1
Cambios	2	2	3

Figura 3.2: Matriz de diseño de la Figura 3.1 con las filas duplicadas

Evidentemente, esta operación siempre mantiene el número de cambios en los factores de la matriz original (A, B y C en nuestro ejemplo), ya que entre las filas duplicadas el número de cambios es cero.

Ahora creamos la secuencia de signos para el nuevo factor (D en el ejemplo), de la siguiente forma:

- Colocamos -1 en la primera y en la última fila
- Llenamos las filas intermedias repitiendo la secuencia 1, 1, -1, -1.

Y de esta forma obtenemos la matriz de diseño de la Figura 3.3.

Orden estándar	Factores			
	A	B	C	D
1	-1	-1	-1	-1
9	-1	-1	-1	1
10	1	-1	-1	1
2	1	-1	-1	-1
6	1	-1	1	-1
14	1	-1	1	1
16	1	1	1	1
8	1	1	1	-1
4	1	1	-1	-1
12	1	1	-1	1
11	-1	1	-1	1
3	-1	1	-1	-1
7	-1	1	1	-1
15	-1	1	1	1
13	-1	-1	1	1
5	-1	-1	1	-1
Cambios	2	2	3	8

Figura 3.3: Diseño  $2^4$  con un orden de ejecución que presenta las propiedades deseadas

Esta nueva matriz, así como la que se obtendría multiplicando la última columna por -1, cumple las propiedades deseadas para un diseño con 16 experimentos, tal como a continuación se pone de manifiesto.

### 3.3 Propiedades de la matriz obtenida

#### 3.3.1 La matriz obtenida es una matriz completa

Efectivamente, la matriz obtenida tiene 16 filas y todas son distintas ya que para cada pareja de 2 filas iguales en los 3 primeros factores, el cuarto factor tiene siempre distinto signo. Por tanto, se trata de una matriz completa para un diseño  $2^4$ .

El argumento es el mismo para pasar de cualquier diseño  $2^k$  a un diseño  $2^{k+1}$  aplicando la metodología propuesta.

### 3.3.2 La matriz obtenida tiene el mínimo número de cambios posible

Sabemos que el mínimo número de cambios en un diseño  $2^k$  es  $2^k - 1$ . Así, en un diseño  $2^3$  es igual a 7, y en un  $2^4$  será igual a 15, etc.

Hemos visto que al duplicar las filas en un diseño  $2^k$  se mantiene el número de cambios en sus factores, y al añadir el factor  $k+1$  con el procedimiento definido, este nuevo factor siempre presentará  $2^k$  cambios de nivel.

En efecto, puede verse en el ejemplo que en el factor D existen 7 parejas de signos y, por tanto, hay 6 cambios de signo entre parejas. Además, hay un cambio de signo al principio y otro al final. En total 8 cambios tal como se observa en la Figura 3.4.

Orden	Factores				7 parejas	6 cambios de signo entre parejas	Cambio inicial + cambio final
	A	B	C	D			
1	-1	-1	-1	-1			
9	-1	-1	-1	1			
10	1	-1	-1	1			
2	1	-1	-1	-1			
6	1	-1	1	-1			
14	1	-1	1	1			
16	1	1	1	1			
8	1	1	1	-1			
4	1	1	-1	-1			
12	1	1	-1	1			
11	-1	1	-1	1			
3	-1	1	-1	-1			
7	-1	1	1	-1			
15	-1	1	1	1			
13	-1	-1	1	1			
5	-1	-1	1	-1			
Cambios	2	2	3	8			

Figura 3.4: Cambios de signo en el factor añadido al pasar de un diseño  $2^3$  a un  $2^4$

En general, en un diseño  $2^{k+1}$  tendremos  $2^k - 1$  parejas de signos y, por tanto,  $2^k - 2$  cambios de signos entre ellas más los cambios en la primera y la última fila dan un total de  $2^k$  cambios. Tendremos, por tanto, los  $2^k - 1$  de los  $k$  primeros factores más los  $2^k$  del factor  $k+1$  lo cual suma un total de  $2^{k+1} - 1$  cambios de nivel que, tal como hemos visto, es el mínimo posible en el diseño  $2^{k+1}$ .

### 3.3.3 La matriz obtenida permite estimar los efectos con un sesgo conocido

A continuación se deducen los valores que toman los efectos de la matriz obtenida en función de los que se tienen en la matriz de partida.

#### 3.3.3.1 Todos los efectos, ya sean principales o interacciones, que ya aparecen en el diseño de partida, duplican su valor.

Veamos el caso del paso del diseño  $2^3$  al  $2^4$  en el ejemplo que hemos visto anteriormente.

Orden estándar	A	B	C	AB	AC	BC	ABC	y
1	-1	-1	-1	1	1	1	-1	1
2	1	-1	-1	-1	-1	1	1	2
6	1	-1	1	-1	1	-1	-1	3
8	1	1	1	1	1	1	1	4
4	1	1	-1	1	-1	-1	-1	5
3	-1	1	-1	-1	1	-1	1	6
7	-1	1	1	-1	-1	1	-1	7
5	-1	-1	1	1	-1	-1	1	8
Efectos	-2	2	2	0	-2	-2	1	

Figura 3.5: Cálculo de los efectos en el diseño  $2^3$  de partida

Orden estándar	A	B	C	D	AB	AC	AD	BC	BD	CD	ABC	ABD	ACD	BCD	ABCD
1	-1	-1	-1	-1	1	1	1	1	1	1	-1	-1	-1	-1	1
9	-1	-1	-1	1	1	1	-1	1	-1	-1	-1	1	1	1	-1
10	1	-1	-1	1	-1	-1	1	1	-1	-1	1	-1	-1	1	1
2	1	-1	-1	-1	-1	-1	-1	1	1	1	1	1	1	-1	-1
6	1	-1	1	-1	-1	1	-1	-1	1	-1	-1	1	-1	1	1
14	1	-1	1	1	-1	1	1	-1	-1	1	-1	-1	1	-1	-1
16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	1	1	1	-1	1	1	-1	1	-1	-1	1	-1	-1	-1	-1
4	1	1	-1	-1	1	-1	-1	-1	-1	1	-1	-1	1	1	1
12	1	1	-1	1	1	-1	1	-1	1	-1	-1	1	-1	-1	-1
11	-1	1	-1	1	-1	1	-1	-1	1	-1	1	-1	1	-1	1
3	-1	1	-1	-1	-1	1	1	-1	-1	1	1	1	-1	1	-1
7	-1	1	1	-1	-1	-1	1	1	-1	-1	-1	1	1	-1	1
15	-1	1	1	1	-1	-1	-1	1	1	1	-1	-1	-1	1	-1
13	-1	-1	1	1	1	-1	-1	-1	-1	1	1	1	-1	-1	1
5	-1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	1	-1
Efectos	-4	4	4	0	0	-4	0	-4	0	0	2	0	0	0	-1

Figura 3.6: Cálculo de los efectos en el diseño  $2^4$  obtenido

Los *Time Count* (numerador de la expresión que da los efectos) para cualquier efecto en el diseño  $2^3$  y el mismo efecto en el diseño  $2^4$  obtenido siguiendo el procedimiento descrito, son los que se indican en la Figura 3.7, en la que  $s_i$  representa el signo en la fila  $i$  de la columna correspondiente al efecto calculado en la matriz de diseño ampliada (matriz de diseño a la que se han añadido las columnas correspondientes a las interacciones).

<i>Time Count</i> para un efecto de un diseño $2^3$	<i>Time Count</i> para el mismo efecto en un diseño $2^4$ obtenido del $2^3$ anterior aplicando la metodología propuesta
$s_1 \cdot 1$	$s_1 \cdot (1 + 2) = s_1 \cdot 3$
$s_2 \cdot 2$	$s_2 \cdot (3 + 4) = s_2 \cdot 7$
$s_3 \cdot 3$	$s_3 \cdot (5 + 6) = s_3 \cdot 11$
$s_4 \cdot 4$	$s_4 \cdot (7 + 8) = s_4 \cdot 15$
$s_5 \cdot 5$	$s_5 \cdot (9 + 10) = s_5 \cdot 19$
$s_6 \cdot 6$	$s_6 \cdot (11 + 12) = s_6 \cdot 23$
$s_7 \cdot 7$	$s_7 \cdot (13 + 14) = s_7 \cdot 27$
$s_8 \cdot 8$	$s_8 \cdot (15 + 16) = s_8 \cdot 31$

Figura 3.7: *Time Count* para los efectos del diseño  $2^3$  de partida y el diseño  $2^4$  obtenido

La figura anterior también puede escribirse de la siguiente forma:

<i>Time Count</i> para un efecto de un diseño $2^3$	<i>Time Count</i> para el mismo efecto en un diseño $2^4$ obtenido del $2^3$ anterior aplicando la metodología propuesta
$s_1 \cdot 1$	$s_1 \cdot 4 - s_1$
$s_2 \cdot 2$	$s_2 \cdot 8 - s_2$
$s_3 \cdot 3$	$s_3 \cdot 12 - s_3$
$s_4 \cdot 4$	$s_4 \cdot 16 - s_4$
$s_5 \cdot 5$	$s_5 \cdot 20 - s_5$
$s_6 \cdot 6$	$s_6 \cdot 24 - s_6$
$s_7 \cdot 7$	$s_7 \cdot 28 - s_7$
$s_8 \cdot 8$	$s_8 \cdot 32 - s_8$

Figura 3.8: Otra forma de escribir los *Time Count* correspondientes al diseño  $2^4$

Puede observarse que el *Time Count* en el diseño  $2^4$  es igual a 4 veces su valor en el diseño  $2^3$  menos  $\sum_{i=1}^8 s_i$ , y como  $\sum_{i=1}^8 s_i = 0$ , ya que existen la mitad de valores -1 y la otra mitad de +1, resulta el *Time Count* para un efecto en un diseño  $2^4$  es 4 veces el valor del mismo efecto en un diseño  $2^3$ .

Respecto al valor de los efectos, recordemos que responden a la expresión:

$$\text{Efecto} = \frac{\text{Time Count}}{2^k}$$

Como el numerador (*Time Count*) se multiplica por 4, y el denominador ( $2^k$ ) se multiplica por 2, el efecto estará multiplicado por 2.

Para generalizar a cualquier tipo de diseño, veamos la Figura 3.9 que es una generalización de la Figura 3.7.

<i>Time Count</i> para un efecto de un diseño $2^k$	<i>Time Count</i> para el mismo efecto en un diseño $2^{k+1}$ obtenido del $2^k$ anterior aplicando la metodología propuesta
$s_1 \cdot 1$	$s_1 \cdot (1 + 2) = s_1 \cdot 3$
$s_2 \cdot 2$	$s_2 \cdot (3 + 4) = s_2 \cdot 7$
$s_3 \cdot 3$	$s_3 \cdot (5 + 6) = s_3 \cdot 11$
...	...
$s_{2^k-3} \cdot 2^k - 3$	$s_{2^k-3} \cdot (2^{k+1} - 7 + 2^{k+1} - 6)$
$s_{2^k-2} \cdot 2^k - 2$	$s_{2^k-2} \cdot (2^{k+1} - 5 + 2^{k+1} - 4)$
$s_{2^k-1} \cdot 2^k - 1$	$s_{2^k-1} \cdot (2^{k+1} - 3 + 2^{k+1} - 2)$
$s_{2^k} \cdot 2^k$	$s_{2^k} \cdot (2^{k+1} - 1 + 2^{k+1})$

Figura 3.9: Generalización del cálculo del *Time Count* para el diseño obtenido (1)

Esta tabla también se puede escribir de la forma:

<i>Time Count</i> para un efecto de un diseño $2^k$	<i>Time Count</i> para el mismo efecto en un diseño $2^{k+1}$ obtenido del $2^k$ anterior aplicando la metodología propuesta
$s_1 \cdot 1$	$s_1 \cdot 4 - s_1$
$s_2 \cdot 2$	$s_2 \cdot 8 - s_2$
$s_3 \cdot 3$	$s_3 \cdot 12 - s_3$
...	...
$s_{2^k-3} \cdot 2^k - 3$	$s_{2^k-3} \cdot (2^{k+2} - 12) - s_{2^k-3}$
$s_{2^k-2} \cdot 2^k - 2$	$s_{2^k-2} \cdot (2^{k+2} - 8) - s_{2^k-2}$
$s_{2^k-1} \cdot 2^k - 1$	$s_{2^k-1} \cdot (2^{k+2} - 4) - s_{2^k-1}$
$s_{2^k} \cdot 2^k$	$s_{2^k} \cdot 2^{k+2} - s_{2^k}$

Figura 3.10: Generalización del cálculo del *Time Count* para el diseño obtenido (2)

Como  $4 \cdot 2^k = 2^{k+2}$ , tenemos que el *Time Count* para los efectos en el nuevo diseño es siempre 4 veces su valor en el diseño original. Por tanto, el valor de sus efectos se duplica.

### 3.3.3.2 El efecto del factor que se añade es siempre igual a cero.

Observemos la secuencia de signos utilizada en el ejemplo para el factor que añadimos:

...	D	...	y
	-1		1
	1		2
	1		3
	-1		4
	-1		5
	1		6
	1		7
	-1		8
	-1		9
	1		10
	1		11
	-1		12
	-1		13
	1		14
	1		15
	-1		16

$-4 - 5 + 6 + 7 = 4$   
 $-8 - 9 + 10 + 11 = 4$   
 $-12 - 13 + 14 + 15 = 4$

Figura 3.11: Valores del *Time Count* en las dobles parejas centrales del factor añadido

Dejando las 3 primeras y la última, el resto de filas se pueden dividir en 3 grupos en este caso y, en general, en  $\frac{2^k}{4} - 1$  grupos y dentro de cada grupo la suma de las respuestas por los signos que les corresponden es siempre igual a 4.

Por otra parte, el valor correspondiente a las 3 primeras filas siempre será igual a 4  $(-1+2+3)$  independientemente del número de filas que se tengan. El valor correspondiente a la última fila será siempre  $-2^k$ . Por tanto, el *Time Count* para esta columna será:

$$\left(\frac{2^k}{4} - 1\right) \cdot 4 + 4 - 2^k$$

Y es inmediato comprobar que este valor es siempre igual a cero. También es obvio que si cambiamos los signos a todas las filas (segunda opción para la secuencia de signos del factor que se añade) el *Time Count* seguirá siendo cero.

### 3.3.3.3 Las interacciones en las que interviene el factor añadido tienen, como máximo, un valor absoluto igual a 1

La Figura 3.12 muestra los sumando que permiten calcular el *Time Count* para cualquier efecto de un diseño  $2^4$  obtenido mediante la metodología propuesta y que ya aparecen en el diseño de partida  $2^3$ , así como los correspondientes a cualquier efecto del nuevo diseño en el que interviene el factor añadido (D, en este caso).

Los signos en la columna de X están duplicados, es decir:  $s_1 = s_2$ ,  $s_3 = s_4$ , ...,  $s_{15} = s_{16}$ . Por otra parte, las respuestas son  $y_1 = 1$ ,  $y_2 = 2$ ,  $y_3 = 3$ , etc., así que el valor de X·D también se puede escribir tal como se indica en la Figura 3.13.

X	...	D	...	X · D
$s_1y_1$		-1		$-s_1y_1$
$s_2y_2$		1		$s_2y_2$
$s_3y_3$		1		$s_3y_3$
$s_4y_4$		-1		$-s_4y_4$
$s_5y_5$		-1		$-s_5y_5$
$s_6y_6$		1		$s_6y_6$
$s_7y_7$		1		$s_7y_7$
$s_8y_8$		-1		$-s_8y_8$
$s_9y_9$		-1		$-s_9y_9$
$s_{10}y_{10}$		1		$s_{10}y_{10}$
$s_{11}y_{11}$		1		$s_{11}y_{11}$
$s_{12}y_{12}$		-1		$-s_{12}y_{12}$
$s_{13}y_{13}$		-1		$-s_{13}y_{13}$
$s_{14}y_{14}$		1		$s_{14}y_{14}$
$s_{15}y_{15}$		1		$s_{15}y_{15}$
$s_{16}y_{16}$		-1		$-s_{16}y_{16}$

Figura 3.12: Sumandos del time count para los efectos X (cualquier efecto de los que aparecen en el diseño de partida) y X·D (1)

X · D	Suma de la pareja
$-s_11$	$\pm 1$
$s_12$	
$s_33$	$\pm 1$
$-s_34$	
$-s_55$	$\pm 1$
$s_56$	
$s_77$	$\pm 1$
$-s_78$	
$-s_99$	$\pm 1$
$s_910$	
$s_{11}11$	$\pm 1$
$-s_{11}12$	
$-s_{13}13$	$\pm 1$
$s_{13}14$	
$s_{15}15$	$\pm 1$
$-s_{15}16$	

Figura 3.13: Sumandos del time count para los efectos X (cualquier efecto de los que aparecen en el diseño de partida) y X·D (2)

El valor máximo del *Time Count* se obtiene cuando la suma de todas las parejas es igual a +1, lo cual se da cuando:

$$s_1 = s_5 = s_9 = s_{13} = 1 ,$$

$$s_3 = s_7 = s_{11} = s_{15} = -1$$

Y el valor mínimo lo tendremos cuando la suma de cada pareja sea igual a -1, lo cual se da cuando se tienen los signos:

$$s_1 = s_5 = s_9 = s_{13} = -1$$

$$s_3 = s_7 = s_{11} = s_{15} = 1$$

Tal como se pone de manifiesto en la Figura 3.14.

X · D	s positivo	s negativo
-s <sub>1</sub> 1	1	-1
s <sub>1</sub> 2		
s <sub>3</sub> 3	-1	1
-s <sub>3</sub> 4		
-s <sub>5</sub> 5	1	-1
s <sub>5</sub> 6		
s <sub>7</sub> 7	-1	1
-s <sub>7</sub> 8		
-s <sub>9</sub> 9	1	-1
s <sub>9</sub> 10		
s <sub>11</sub> 11	-1	1
-s <sub>11</sub> 12		
-s <sub>13</sub> 13	1	-1
s <sub>13</sub> 14		
s <sub>15</sub> 15	-1	1
-s <sub>15</sub> 16		

Figura 3.14: Suma para cada pareja según s sea positivo o negativo

Por tanto, en este caso el *Time Count* puede valer como máximo 8 y como mínimo -8, lo cual significa que los valores del efecto estarán entre 1 y -1.

En general, el *Time Count* tomará un valor absoluto máximo igual al número de parejas que se forman, es decir, si se tienen  $k$  factores será  $2^{k-1}$ . Como los efectos se calculan dividiendo el *Time Count* por la mitad del número de experimentos, el valor absoluto máximo de este tipo de efectos será igual a 1.

### 3.3.4 Resumen de las características del diseño obtenido

Aplicando la metodología propuesta a un diseño  $2^k$  con las propiedades deseadas, se obtiene un diseño  $2^{k+1}$  con unas propiedades que están resumidas en la Figura 3.15.

Número de cambios:	El mínimo posible
Sesgo máximo en valor absoluto:	Doble que en el diseño de partida
Efectos:	
Los que ya aparecen en el diseño de partida:	Duplica su valor
Efecto principal del factor que se añade:	Nulo
Interacciones en las que participa el factor que se añade:	Valor absoluto entre 0 y 1

*Figura 3.15: Resumen de las propiedades del diseño obtenido aplicando la metodología propuesta*

Este método conduce a los mejores órdenes posibles para diseños  $2^4$  (sesgo máximo absoluto igual a 4) a partir de los diseños  $2^3$  (sesgo máximo absoluto igual a 2). Pero si se aplica a la obtención de diseños  $2^5$  se obtiene un sesgo igual a 8 y, de entrada, no se tiene la seguridad de que este sea el mínimo posible ya que para este tipo de diseño no se han realizado análisis exhaustivos.

En el próximo capítulo se analiza esta situación y se presentan alternativas para disminuir el sesgo cuando  $k > 4$ .

## 3.4 Aportaciones

En este capítulo se ha presentado un procedimiento para obtener órdenes de experimentación con el mínimo número de cambios y un sesgo máximo conocido a partir de los órdenes de un diseño con un factor menos y que tenga las propiedades deseadas.

Se han analizado con detalle las propiedades que presenta el método planteado, y se ha deducido cuales serán los valores del sesgos en el nuevo diseño en función de los valores en el diseño de partida.

# Capítulo 4



# Capítulo 4

## Búsqueda de órdenes con mínimo sesgo y mínimo número de cambios

En el capítulo anterior se presentó un método que partiendo de un diseño con las propiedades deseadas, permite construir otro diseño con un factor más que mantiene el mínimo número de cambios en los niveles de los factores y duplica el sesgo máximo absoluto del diseño de partida.

Duplicar el sesgo máximo puede ser la mejor opción, como ocurre al obtener diseños  $2^4$  a partir de los diseños  $2^3$ . Para estos diseños ya se conocen todos los órdenes con las propiedades deseadas, y en el caso de los diseños  $2^3$  tienen un mínimo sesgo máximo igual a 2, y en los  $2^4$  este valor es igual a 4. Exactamente el que proporciona el método propuesto, que es capaz de obtener todos los órdenes con las propiedades deseadas para los diseños  $2^4$ .

Siguiendo con el método, pasar de diseños  $2^4$  a  $2^5$  produce ordenaciones que tienen un sesgo máximo igual a 8 para estos últimos diseños. La literatura sobre este tema no indica cual es el valor mínimo posible de este sesgo máximo, lo cual obliga a explorar la posibilidad de que existan órdenes que presenten un sesgo menor y, en el caso de que existan, identificar los mejores.

Esto es lo que se hace en este capítulo aplicando distintos procedimientos, desde búsquedas aleatorias hasta búsquedas exhaustivas aplicando, en ambos casos, diferentes limitaciones para hacerlos viables en la práctica.

## 4.1 Búsqueda totalmente aleatoria

Una primera idea para detectar órdenes con las propiedades deseadas, conceptualmente muy sencilla y fácil de implementar en un programa informático, puede consistir en:

1. Generar un orden aleatoriamente
2. Contar el número de cambios de nivel en los factores. Si este valor es el mínimo posible, seguir. Si no, volver al punto 1.
3. Calcular el sesgo máximo asociado a ese orden. Si el valor obtenido está por debajo de un cierto valor que se considera bueno, se guarda el orden, en caso contrario se descarta.
4. Volver al punto 1 hasta que se generen todos los órdenes deseados.

Pero este método es totalmente inviable ya que los órdenes que presentan mínimo número de cambios son escasísimos, y aunque se generen los órdenes muy rápidamente, tarda mucho en encontrar órdenes con mínimo número de cambios.

Por ejemplo, en el caso de los diseños  $2^5$ , se tienen  $32! = 2,63 \cdot 10^{35}$  órdenes posibles. Por un método de búsqueda exhaustiva que se verá más adelante, se ha podido determinar que el número de órdenes que presentan mínimo número de cambios es  $1,87 \cdot 10^{11}$ , lo cual significa que la probabilidad de que al generar un orden al azar este presente el mínimo número de cambios es de  $7 \cdot 10^{-25}$ . No hace falta hacer más cuentas para poner de manifiesto que este método no es viable.

## 4.2 Búsqueda aleatoria con restricciones

Se ha diseñado un procedimiento que permite obtener rápidamente órdenes con mínimo número de cambios. Este procedimiento se basa en ir generando los órdenes fila a fila atendiendo a las restricciones que se presentan al asignar las posibles filas sucesoras a las que se van colocando.

### 4.2.1 Restricciones en las filas sucesoras

Sea un diseño  $2^5$  y elijamos un número de orden (correspondiente a la matriz en orden estándar) para el primer experimento a realizar (el que figura en primer lugar en la matriz de diseño). Supongamos que ese primer número de orden es el 10. Si deseamos que la matriz del diseño presente el mínimo número de cambios, no podemos poner en la segunda fila cualquier otro orden tomado al azar.

Efectivamente, para que sólo haya un cambio entre la primera fila y la segunda, sólo se puede elegir entre las filas 9, 12, 14, 2 y 26, tal como se indica en la Figura 4.1.

	Orden de ejecución	Orden estándar	Factores				
			A	B	C	D	E
	1	10	1	-1	-1	1	-1
Posibles órdenes para la segunda fila	}	9	-1	-1	-1	1	-1
		12	1	1	-1	1	-1
		14	1	-1	1	1	-1
		2	1	-1	-1	-1	-1
		26	1	-1	-1	1	1

Figura 4.1: Posibles filas sucesoras a la número 10 para tener sólo un cambio en una y otra

Elegiríamos una al azar, por ejemplo la 9, a continuación habría que elegir otra, de entre las posibles sucesoras de la 9, pero que no sea la 10 porque ya ha salido, y así hasta completar la matriz o llegar a una fila en la que no hay sucesores posibles porque ya todos se han utilizado en las filas anteriores, y en ese caso hay que descartar la secuencia y volver a empezar.

Las posibles sucesoras para cada fila, de forma que sólo se presente un cambio entre una y otra, se encuentran en la Tabla 4.1.

Tabla 4.1. Posibles filas sucesoras para tener mínimo número de cambios en un diseño  $2^5$

<b>1</b>	-	-	-	-	-	<b>2</b>	+	-	-	-	-	<b>3</b>	-	+	-	-	-	<b>4</b>	+	+	-	-	-
2	+	-	-	-	-	1	-	-	-	-	-	1	-	-	-	-	-	2	+	-	-	-	-
3	-	+	-	-	-	4	+	+	-	-	-	4	+	+	-	-	-	3	-	+	-	-	-
5	-	-	+	-	-	6	+	-	+	-	-	7	-	+	+	-	-	8	+	+	+	-	-
9	-	-	-	+	-	10	+	-	-	+	-	11	-	+	-	+	-	12	+	+	-	+	-
17	-	-	-	-	+	18	+	-	-	-	+	19	-	+	-	-	+	20	+	+	-	-	+
<b>5</b>	-	-	+	-	-	<b>6</b>	+	-	+	-	-	<b>7</b>	-	+	+	-	-	<b>8</b>	+	+	+	-	-
1	-	-	-	-	-	2	+	-	-	-	-	3	-	+	-	-	-	4	+	+	-	-	-
6	+	-	+	-	-	5	-	-	+	-	-	5	-	-	+	-	-	6	+	-	+	-	-
7	-	+	+	-	-	8	+	+	+	-	-	8	+	+	+	-	-	7	-	+	+	-	-
13	-	-	+	+	-	14	+	-	+	+	-	15	-	+	+	+	-	16	+	+	+	+	-
21	-	-	+	-	+	22	+	-	+	-	+	23	-	+	+	-	+	24	+	+	+	-	+
<b>9</b>	-	-	-	+	-	<b>10</b>	+	-	-	+	-	<b>11</b>	-	+	-	+	-	<b>12</b>	+	+	-	+	-
1	-	-	-	-	-	2	+	-	-	-	-	3	-	+	-	-	-	4	+	+	-	-	-
10	+	-	-	+	-	9	-	-	-	+	-	9	-	-	-	+	-	10	+	-	-	+	-
11	-	+	-	+	-	12	+	+	-	+	-	12	+	+	-	+	-	11	-	+	-	+	-
13	-	-	+	+	-	14	+	-	+	+	-	15	-	+	+	+	-	16	+	+	+	+	-
25	-	-	-	+	+	26	+	-	-	+	+	27	-	+	-	+	+	28	+	+	-	+	+
<b>13</b>	-	-	+	+	-	<b>14</b>	+	-	+	+	-	<b>15</b>	-	+	+	+	-	<b>16</b>	+	+	+	+	-
5	-	-	+	-	-	6	+	-	+	-	-	7	-	+	+	-	-	8	+	+	+	-	-
9	-	-	-	+	-	10	+	-	-	+	-	11	-	+	-	+	-	12	+	+	-	+	-
14	+	-	+	+	-	13	-	-	+	+	-	13	-	-	+	+	-	14	+	-	+	+	-
15	-	+	+	+	-	16	+	+	+	+	-	16	+	+	+	+	-	15	-	+	+	+	-
29	-	-	+	+	+	30	+	-	+	+	+	31	-	+	+	+	+	32	+	+	+	+	+
<b>17</b>	-	-	-	-	+	<b>18</b>	+	-	-	-	+	<b>19</b>	-	+	-	-	+	<b>20</b>	+	+	-	-	+
1	-	-	-	-	-	2	+	-	-	-	-	3	-	+	-	-	-	4	+	+	-	-	-
18	+	-	-	-	+	17	-	-	-	-	+	17	-	-	-	-	+	18	+	-	-	-	+
19	-	+	-	-	+	20	+	+	-	-	+	20	+	+	-	-	+	19	-	+	-	-	+
21	-	-	+	-	+	22	+	-	+	-	+	23	-	+	+	-	+	24	+	+	+	-	+
25	-	-	-	+	+	26	+	-	-	+	+	27	-	+	-	+	+	28	+	+	-	+	+
<b>21</b>	-	-	+	-	+	<b>22</b>	+	-	+	-	+	<b>23</b>	-	+	+	-	+	<b>24</b>	+	+	+	-	+
5	-	-	+	-	-	6	+	-	+	-	-	7	-	+	+	-	-	8	+	+	+	-	-
17	-	-	-	-	+	18	+	-	-	-	+	19	-	+	-	-	+	20	+	+	-	-	+
22	+	-	+	-	+	21	-	-	+	-	+	21	-	-	+	-	+	22	+	-	+	-	+
23	-	+	+	-	+	24	+	+	+	-	+	24	+	+	+	-	+	23	-	+	+	-	+
29	-	-	+	+	+	30	+	-	+	+	+	31	-	+	+	+	+	32	+	+	+	+	+
<b>25</b>	-	-	-	+	+	<b>26</b>	+	-	-	+	+	<b>27</b>	-	+	-	+	+	<b>28</b>	+	+	-	+	+
9	-	-	-	+	-	10	+	-	-	+	-	11	-	+	-	+	-	12	+	+	-	+	-
17	-	-	-	-	+	18	+	-	-	-	+	19	-	+	-	-	+	20	+	+	-	-	+
26	+	-	-	+	+	25	-	-	-	+	+	25	-	-	-	+	+	26	+	-	-	+	+
27	-	+	-	+	+	28	+	+	-	+	+	28	+	+	-	+	+	27	-	+	-	+	+
29	-	-	+	+	+	30	+	-	+	+	+	31	-	+	+	+	+	32	+	+	+	+	+
<b>29</b>	-	-	+	+	+	<b>30</b>	+	-	+	+	+	<b>31</b>	-	+	+	+	+	<b>32</b>	+	+	+	+	+
13	-	-	+	+	-	14	+	-	+	+	-	15	-	+	+	+	-	16	+	+	+	+	-
21	-	-	+	-	+	22	+	-	+	-	+	23	-	+	+	-	+	24	+	+	+	-	+
25	-	-	-	+	+	26	+	-	-	+	+	27	-	+	-	+	+	28	+	+	-	+	+
30	+	-	+	+	+	29	-	-	+	+	+	29	-	-	+	+	+	30	+	-	+	+	+
31	-	+	+	+	+	32	+	+	+	+	+	32	+	+	+	+	+	31	-	+	+	+	+

#### 4.2.2 Algoritmo para la búsqueda de órdenes con sesgo mínimo

Esquemáticamente, el procedimiento utilizado sigue los pasos que a continuación se detallan:

1. Generar un número aleatorio entre 1 y  $2^k$ . Este será el número de fila de la matriz en orden estándar, correspondiente al experimento que se realizará en primer lugar.
2. Identificar los posibles sucesores a la fila que aparece en primer lugar y elegir al azar uno de esos posibles sucesores.
3. De los posibles sucesores a la última fila seleccionada, eliminar los números de orden que ya se han asignado a filas anteriores. Si no quedan sucesores posibles volver al punto 1, si sí quedan, elegir uno al azar.
4. Cuando ya se han completado las  $2^k$  filas, calcular los efectos e identificar el máximo en valor absoluto.
5. Guardar (de forma tabulada, para que no ocupe mucho espacio) el sesgo obtenido. Si dicho sesgo está por debajo de un cierto valor que se considera interesante, guardar también el orden que conduce a ese sesgo
6. Volver al punto 1 hasta finalizar todas las iteraciones deseadas.

La Figura 4.2 muestra el diagrama de flujo del algoritmo utilizado.

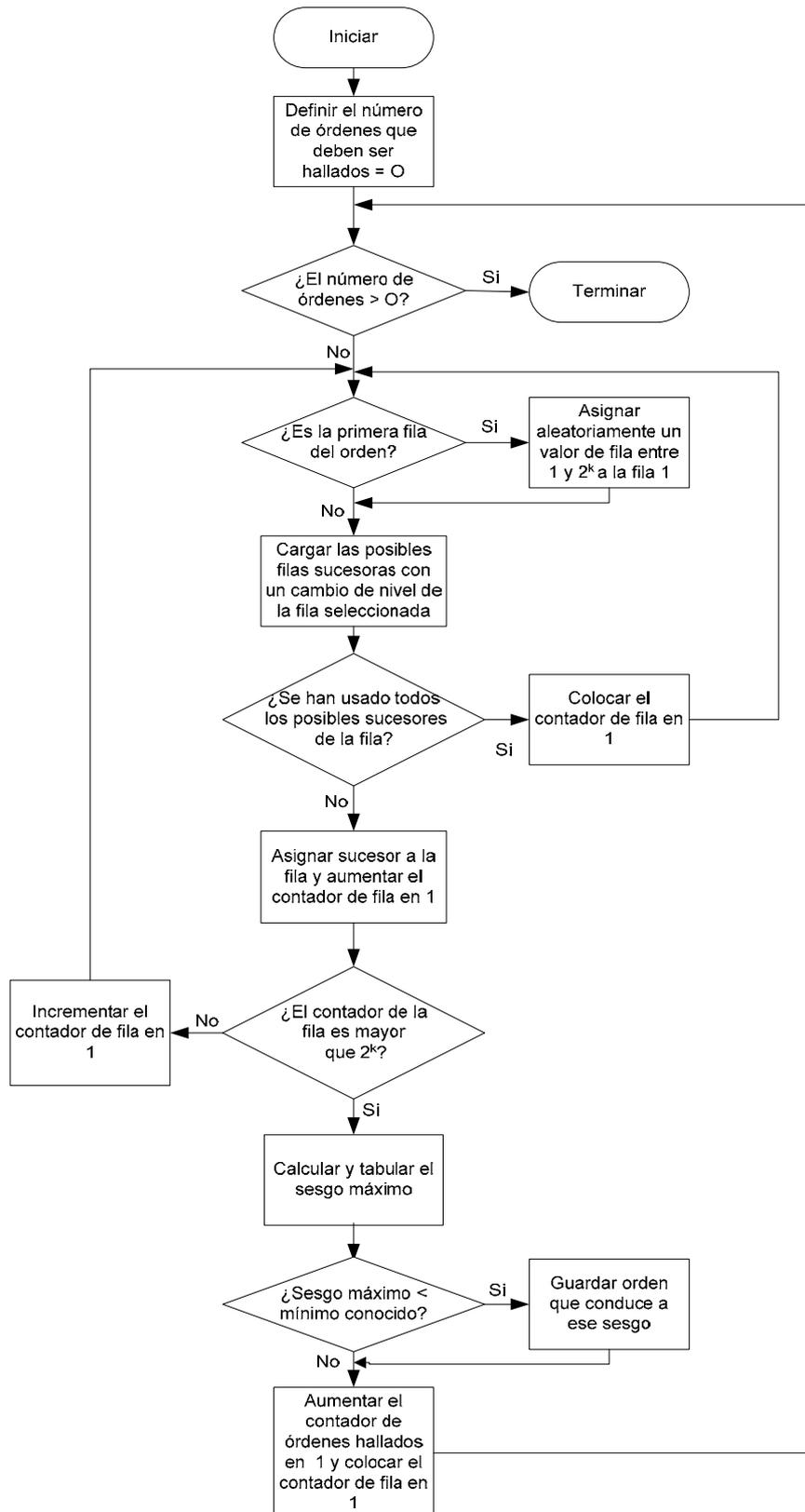


Figura 4.2 Diagrama de flujo del algoritmo de búsqueda aleatoria con restricciones

### 4.2.3 Programa en Free Basic

La primera idea era realizar los programas utilizando el lenguaje que incorpora Minitab (macros). Entre otras ventajas, hacerlo así evita el tener que programar el cálculo de los efectos, ordenación de valores, detectar máximos y mínimos, y otras rutinas que si se utiliza Minitab ya vienen dadas. Pero el inconveniente es que la ejecución de los programas es muy lenta, y esta lentitud es incompatible con los objetivos de búsqueda y exploración de órdenes que nos hemos propuesto.

Es necesario utilizar un lenguaje compilable de ejecución rápida, y entre las alternativas disponibles se ha elegido Free Basic por las siguientes razones:

1. Es un programa de libre uso y, por tanto, accesible a todo el mundo, que puede ser bajado del sitio web: <http://www.freebasic.net/>
2. El lenguaje es fácil de aprender y es bastante parecido al que se utiliza para realizar las macros de Minitab
3. Dispone de una cómoda interficie para escribir los programas
4. Está también disponible un manual muy completo y claro sobre el uso del lenguaje y de cada uno de los comandos. (Ver Anexo 2)

Aunque se podía haber realizado un sólo programa válido para un cierto rango del número de factores, se ha preferido, por comodidad, realizar un programa diferente para  $k = 5$ ,  $6$  y  $7$ . En el Anexo 3 se incluye el código del programa para  $k = 6$ .

### 4.2.4 Resultados obtenidos

#### 4.2.4.1 Diseños con 32 experimentos

Se han generado 50.000.000 de órdenes con el mínimo número de cambios de nivel. La obtención de estos órdenes, y los análisis correspondientes, se ha demorado 5 horas y 30 minutos. El menor sesgo obtenido es 5,25. En la Figura 4.3 se presenta la distribución de los sesgos obtenidos.

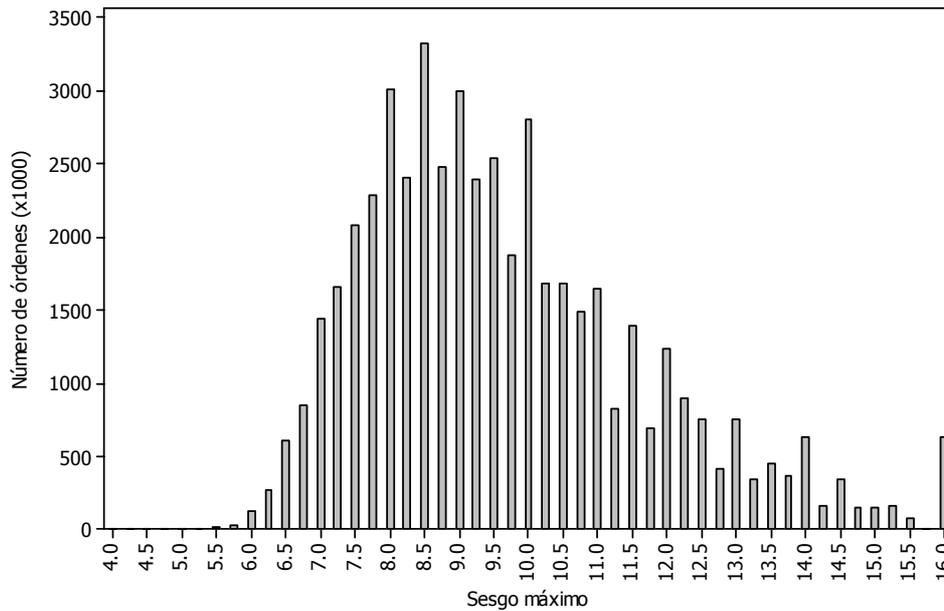


Figura 4.3: Diagrama de barras del valor de los sesgos obtenidos al generar aleatoriamente 50 millones de órdenes con mínimo número de cambios en diseños con 32 experimentos

#### 4.2.4.2 Diseños con 64 experimentos

Para 64 experimentos se ha generado un total de 10.000.000 de órdenes (unas 80 horas de computación) habiéndose obtenido los sesgos que se indican en la Figura 4.4.

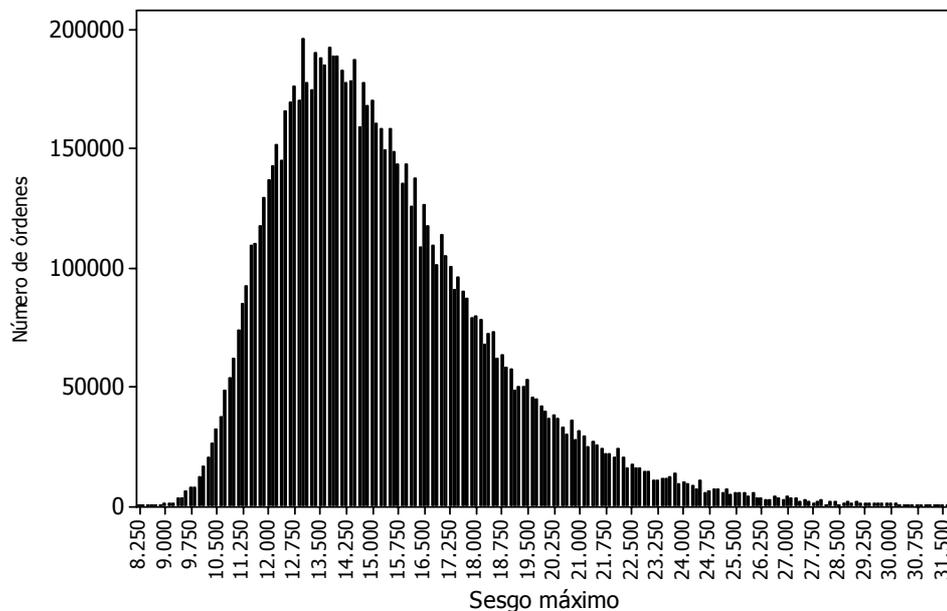


Figura 4.4: Diagrama de barras del valor de los sesgos obtenidos al generar aleatoriamente 10 millones de órdenes con mínimo número de cambios en diseños con 64 experimentos

El valor mínimo para el sesgo es igual a 8.25. Por debajo de 10.5 (valor que obtendríamos duplicando el mejor orden obtenido para 32 experimentos) se tiene solo el 1.27% de ordenaciones. Aproximadamente el 80% de los órdenes obtenidos se concentra entre los sesgos 12 y 20.

Se ha obtenido una sola ordenación, aunque repetida varias veces, que da el sesgo mínimo. Esta es la que se indica en la Tabla 4.2.

*Tabla 4.2: Único orden obtenido que proporciona el mínimo sesgo hallado con el mínimo número de cambios para un diseño con 64 experimentos*

Posición	# Orden						
1	44	17	4	33	28	49	25
2	48	18	36	34	20	50	57
3	64	19	34	35	19	51	59
4	63	20	38	36	3	52	60
5	47	21	54	37	1	53	52
6	45	22	22	38	2	54	56
7	13	23	30	39	18	55	40
8	15	24	26	40	17	56	8
9	7	25	58	41	21	57	24
10	39	26	50	42	29	58	32
11	35	27	49	43	61	59	16
12	33	28	51	44	62	60	14
13	41	29	55	45	46	61	6
14	43	30	23	46	42	62	5
15	11	31	31	47	10	63	37
16	12	32	27	48	9	64	53

#### 4.2.4.3 Diseños con 128 experimentos

De manera similar se procedió a la búsqueda de órdenes con el mínimo número de cambios de nivel en los diseños con 128 experimentos. En este caso la generación de órdenes es mucho más lenta, tardando un promedio de unos 4 minutos en generar cada uno. Debido a esta dificultad, y al hecho de que los mejores órdenes salen sistemáticamente repetidos, se han generado solo 1000 órdenes, lo cual ha requerido unas 70 horas de computación.

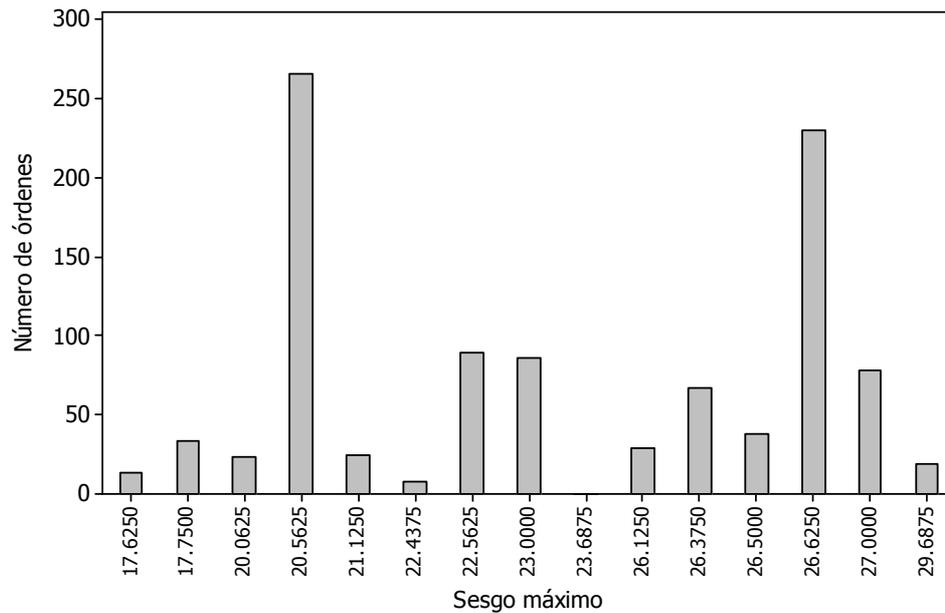


Figura 4.5: Diagrama de barras del valor de los sesgos obtenidos al generar aleatoriamente 1000 órdenes con mínimo número de cambios en diseños con 128 experimentos

La Figura 4.4 presenta el diagrama de barras de los sesgos obtenidos. Obsérvese que el sesgo mínimo obtenido por este método (igual a 17.625) es mayor que el doble del obtenido para el diseño con 64 experimentos (igual a 8.25) por lo cual este método de búsqueda no ha proporcionado resultados relevantes para este tipo de diseños.

#### 4.2.4.4 Resumen de resultados obtenidos

Tras aplicar el método de la búsqueda aleatoria, la situación en cuanto a los sesgos mínimos es la que se indica en la Tabla 4.3. Puede observarse que para diseños de 32 y 64 experimentos se ha obtenido un resultado mejor al que resultaría de duplicar el obtenido en un diseño con un factor menos.

Tabla 4.3: Resumen de sesgos obtenidos

# Experimentos	Sesgo obtenido	
	Expansión del diseño anterior	Búsqueda aleatoria
32	8	5.25
64	10.5	8.25
128	16.5	17.625

Sin embargo, en ningún caso existe garantía de que estos sesgos sean los mínimos posibles. La deducción teórica de cuales son esos valores mínimos para cada tipo de diseño es una tarea que ha demostrado ser inviable incluso para diseños con un menor número de experimentos (véase de León et al., 2005), de manera que la única forma de identificar los mínimos absolutos es a través de la generación exhaustiva de todos los órdenes posibles.

### **4.3 Búsqueda exhaustiva de órdenes semilla**

A. W. Dickinson publicó en 1974 un algoritmo que permite explorar de forma exhaustiva todos los órdenes de un diseño factorial con mínimo número de cambios, aunque el desarrollo de la informática en esa época restringía sus posibilidades de aplicación más allá de diseños muy pequeños.

De León (2004) utilizó un procedimiento basado en este algoritmo implementado en una macro de Minitab para obtener todos los órdenes para diseños con 16 experimentos. Pero el tiempo de computación que exige este trabajo es muy alto y lo hace inviable para diseños con un mayor número de factores. La posibilidad de utilizar lenguajes de programación mucho más rápidos que Minitab, y también la introducción de algunos cambios en el algoritmo de Dickinson nos hicieron vislumbrar la posibilidad de obtener unos resultados que, como mínimo, mejoraran los obtenidos con la búsqueda aleatoria.

#### **4.3.1 Algoritmo de Dickinson**

La característica más destacable de este algoritmo, y la que hace viable su aplicación, es que reduce la búsqueda de los órdenes a un grupo estratégico a partir del cual se pueden deducir todos los demás. Los órdenes que se encuentra son los que llamamos “órdenes semilla”, ya que a partir de cada uno de estos órdenes se pueden obtener órdenes adicionales, que presentan las mismas características en cuanto a sesgo y número de cambios, a base de permutar el orden de las columnas de la matriz de diseño y/o multiplicar por -1 algunas de ellas. La explicación detallada de cómo pueden generarse nuevos órdenes a partir de uno conocido se trata en detalle en el capítulo 5.

Para obtener sólo órdenes semilla y evitar redundancias en la búsqueda, el algoritmo de Dickinson obliga a que la búsqueda se realice con las siguientes restricciones:

1. Solo debe haber un cambio de signo entre una fila y la siguiente.  
Evidentemente esto debe ser así para asegurar el mínimo número de cambios en los niveles de los factores.
2. Debe darse una secuencia determinada en las filas que presentan un solo signo positivo. En concreto, exige que los signos positivos se vayan desplazando de izquierda a derecha, lo cual corresponde a la secuencia 2, 3, 5, 9 y 17 para un diseño  $2^5$ .

Orden	Factores				
	A	B	C	D	E
1	-1	-1	-1	-1	-1
2	①	-1	-1	-1	-1
...					
3	-1	①	-1	-1	-1
...					
5	-1	-1	①	-1	-1
...					
9	-1	-1	-1	①	-1
...					
17	-1	-1	-1	-1	①
...					

Figura 4.6: Secuencia de órdenes a que obliga el algoritmo de Dickinson en un diseño  $2^5$

De esta forma se asegura que no se obtienen órdenes que se podrían deducir unos de otros a base de permutar el orden de las columnas. Forzando esta secuencia se obtiene una sola de entre todas las que pueden obtenerse permutando las columnas.

3. La primera fila solo debe incluir signos negativos. Es decir, es la que corresponde a la primera posición en el orden estándar.

De esta forma se asegura que no se obtienen dos órdenes que se puedan deducir uno del otro a base de multiplicar por -1 alguna columna

En la Figura 4.7 se presenta el diagrama de flujo que representa el funcionamiento de este algoritmo. Esquemáticamente, funciona de la siguiente forma:

- Se fijan las primeras dos filas. Las restricciones comentadas anteriormente imponen que estas filas siempre sean la 1 y la 2 del orden estándar.

- Se buscan las posibles filas sucesoras (que cumplan con las restricciones impuestas) a partir de la segunda fila. De esta forma se genera una primera ordenación de forma similar a como se hacía en la búsqueda aleatoria con restricciones, con la condición añadida de la secuencia en las filas que tienen un solo signo positivo.
- Una vez obtenido el primer orden se inicia una generación exhaustiva de órdenes evaluando todos los posibles sucesores de las filas empezando por abajo y hasta llegar a la fila 2, momento en que ya se han generado todos los órdenes semilla.

#### 4.3.1.1 Ejemplo de aplicación del algoritmo de Dickinson

Para ilustrar el funcionamiento de este algoritmo, veamos como funcionaría aplicado a un diseño  $2^3$ . En primer lugar hay que plantear la lista de posibles sucesores para cada fila, estos sucesores son los que se indican en la Tabla 4.4.

Tabla 4.4: Posibles sucesores de cada una de las filas en un diseño  $2^3$

Número de fila (en el orden estándar de la matriz de diseño)	Posibles filas sucesoras
1	2
2	4 - 6
3	4 - 7
4	3 - 8
5	6 - 7
6	5 - 8
7	3 - 5 - 8
8	4 - 6 - 7

A continuación, veamos como se van seleccionando los números de fila para determinar la primera ordenación (selección de las filas de arriba abajo).

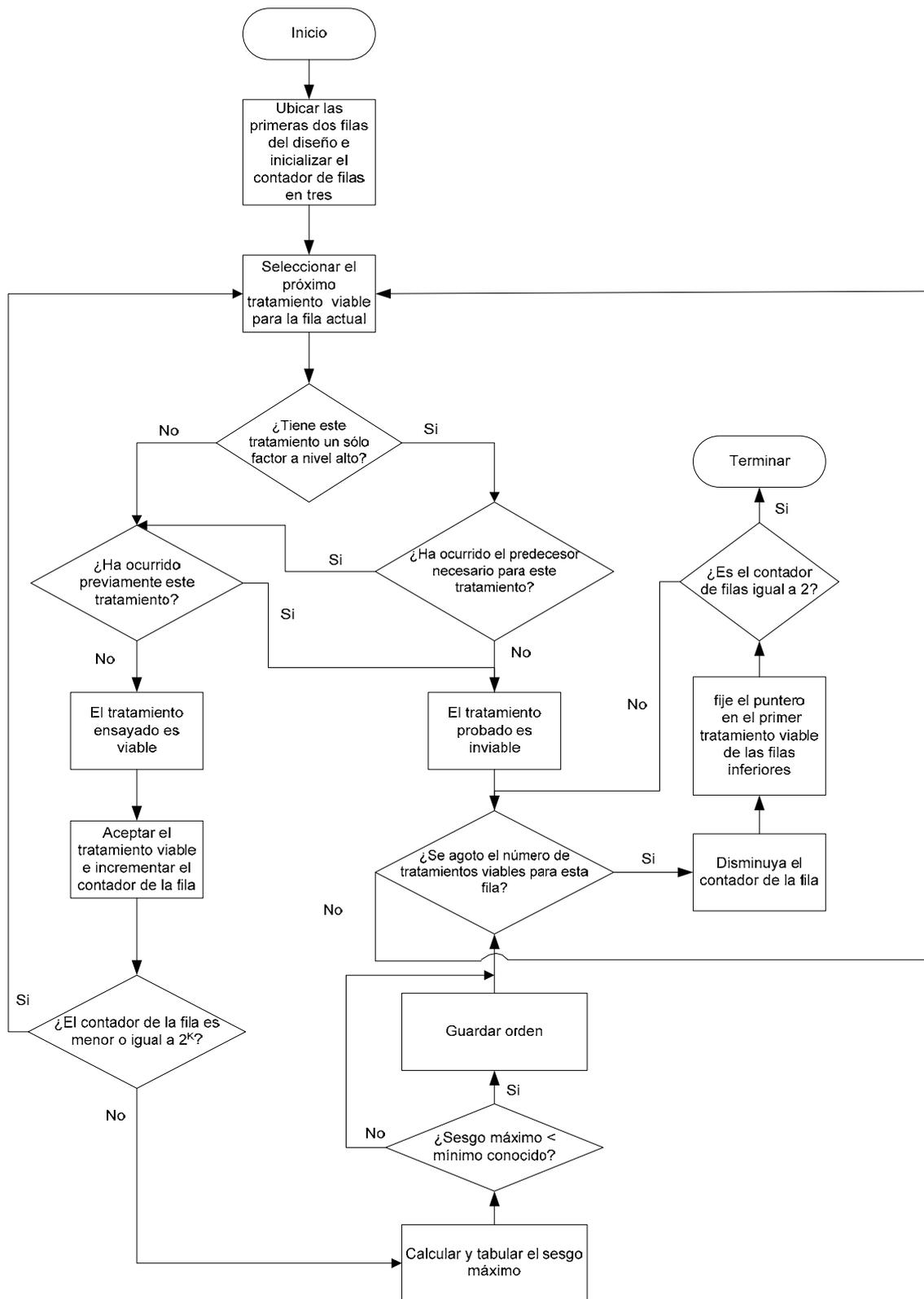


Figura 4.7 Algoritmo de Dickinson para los diseños factoriales a dos niveles con  $2^k$  experimentos

Filas en las posiciones 1 y 2:

Siempre son las 1 y 2 en el orden estándar de la matriz de diseño

Posición de la fila	Números de fila (orden estándar)
1	1
2	2
:	:

Filas en la posición 3:

Posibles sucesores de la fila 2 (en el orden estándar) son la 4 y la 6. Elegimos la 4

Posición de la fila	Números de fila (orden estándar)
1	1
2	2
3	4
:	:

Fila en la posición 4:

Posibles sucesores de la fila 4 son la 3 y la 8. Elegimos la 3. Esta ordenación ya cumplirá la condición de que aparezca primero la fila 3 que la 5.

Posición de la fila	Números de fila (orden estándar)
1	1
2	2
3	4
4	3
:	:

Fila en la posición 5:

Posibles sucesores de la fila 3 son la 4 y la 7. Elegimos la 7 porque la 4 ya ha salido.

Posición de la fila	Números de fila (orden estándar)
1	1
2	2
3	4
4	3
5	7
:	:

Fila en la posición 6:

Posibles sucesores de la fila 7 son la 3, la 5 y la 8. Elegimos la 5 porque la 3 ya ha salido.

Posición de la fila	Números de fila (orden estándar)
1	1
2	2
3	4
4	3
5	7
6	5
:	:

Filas en las posición 7:

Posibles sucesores de la fila 5 son la 6 y la 7.

Elegimos la 6.

Posición de la fila	Números de fila (orden estándar)
1	1
2	2
3	4
4	3
5	7
6	5
7	6
:	:

Filas en las posición 8:

Posibles sucesores de la fila 6 son la 5 y la 8. Como la 5 ya ha salido, elegimos la 8.

Posición de la fila	Números de fila (orden estándar)
1	1
2	2
3	4
4	3
5	7
6	5
7	6
8	8

En este caso de los diseños  $2^3$  se completa la primera ordenación sin tener que volver hacia atrás. Para diseños con mayor número de factores puede ocurrir que con los sucesores que se van seleccionando se llegue a una posición de fila en que los sucesores posibles no se pueden usar porque ya han aparecido todos en filas anteriores. En este caso hay que volver hacia atrás y cambiar la selección en la primera fila que se encuentra en la que se ha seleccionado un sucesor pero había otros posibles.

Una vez tenemos la primera ordenación completada, nos situamos en la antepenúltima<sup>1</sup> fila y observamos si había otras alternativas posibles. Si en esta fila no se puede cambiar de sucesor, se va subiendo hasta llegar a una fila en la que este cambio es posible. En nuestro ejemplo, hay que subir a la fila 6 y cambiar el sucesor fila 5 por el sucesor fila 8, y explorar todas las posibilidades de este cambio hacia abajo.

Siguiendo con el mismo procedimiento, después de completar la segunda ordenación subimos y vamos explorando las alternativas que hemos dejado. En nuestro caso, en la fila 4, cambiando del sucesor fila 3 al sucesor fila 8 se entra en un camino que no se puede finalizar (llega un momento en que no hay sucesores posibles). En la fila 3 sí se

<sup>1</sup> ¿Por qué la antepenúltima? Seguro que para la penúltima se ha seleccionado el único sucesor disponible (evidentemente, es imposible que hayan dos opciones para colocar en la última fila)

puede cambiar del sucesor fila 4 al fila 6 y se llega a una nueva ordenación que también cumple el requisito de mínimo número de cambios.

La Figura 4.8 resume el camino seguido en la primera ordenación de arriba abajo.

Posición de la fila	Posibles números de fila	Números de fila (orden estándar)
1	①	1
2	②	2
3	④ 6	4
4	③ 8	3
5	<del>4</del> ⑦	7
6	<del>3</del> ⑤ 8	5
7	⑥ <del>7</del>	6
8	<del>5</del> ⑧	8

Figura 4.8: Selección de números de fila hasta completar la primera ordenación (de arriba abajo)

Después de la primera ordenación, retrocedemos hasta la primera fila que encontramos en que es posible cambiar el sucesor. En nuestro caso retrocedemos hasta la posición de la fila 6 y cambiamos el sucesor 5 por el 8. A continuación volvemos hacia abajo igual que si fuera la primera ordenación.

Posición de la fila	Posibles números de fila	Números de fila (orden estándar)	
1	①	1	
2	②	2	
3	④ 6	4	
4	③ 8	3	
5	<del>4</del> ⑦	7	
6	<del>3</del> 5 ⑧ ←	8	Primera opción (empezando desde abajo) de cambiar el sucesor
7	<del>4</del> ⑥ 7	6	
8	⑤ <del>8</del>	5	

Figura 4.9: Tras completar la primera ordenación, retrocede hasta la posición 6, cambia el sucesor y sigue.

Después de encontrar esta segunda ordenación, retrocedemos igual que cuando encontramos la primera hasta encontrar una fila en que se puede cambiar de sucesor. Se puede cambiar en la posición de fila 4, cambiando la fila 3 por la 8, pero si se realiza este cambio no se consigue completar la ordenación (se llega a una fila en la que no hay sucesores posibles). Sin embargo, cambiando en la posición de la fila 3 el sucesor fila 4 por el fila 6, se completa un nuevo orden tal como se ilustra en la Figura 4.10.

Posición de la fila	Posibles números de fila	Números de fila (orden estándar)
1	①	1
2	②	2
3	4 ⑥	6 ← Cambio de sucesor
4	<del>5</del> ⑧	8
5	④ <del>6</del> 7	4
6	③ <del>5</del>	3
7	<del>4</del> ⑦	7
8	<del>3</del> ⑤ <del>6</del>	5

Figura 4.10: Nuevo orden cambiando el sucesor en la posición de la fila 3. No se puede elegir el sucesor fila 5 en la posición de la fila 4 por la restricción impuesta en el orden de las filas que tienen un solo signo positivo.

Después de encontrar esta última ordenación podríamos seguir con el procedimiento subiendo hasta la posición 5 y cambiando el sucesor fila 4 por el fila 7 pero tras realizar este cambio no se logra completar ninguna nueva ordenación.

En este caso ya se han encontrado los únicos 3 órdenes semilla que producen el mínimo número de cambios. El último de ellos tiene, además, sesgo mínimo, y este es el orden semilla de los 48 órdenes existentes con las propiedades deseadas de mínimo número de cambios y mínimo sesgo.

### 4.3.2 Implementación del algoritmo de Dickinson

Al igual que en el caso de la búsqueda aleatoria con restricciones, aunque la primera idea era utilizar macros de Minitab para aprovechar las subrutinas y cálculos que ya vienen hechos, la lentitud con que se ejecutan estas obligó a trabajar con el lenguaje Free Basic. En el Anexo 3 se encuentra el código del programa realizado para encontrar los órdenes semilla de los diseños con 32 experimentos.

### 4.3.3 Resultados obtenidos

#### 4.3.3.1 Diseños con 32 experimentos

Para 32 experimentos el método ha resultado un éxito y ha permitido obtener todos los órdenes semilla para diseños con mínimo número de cambios.

En total se han obtenido  $N = 48.828.036$  órdenes, lo cual significa que existen un total de  $187.499.658.240$  órdenes que presentan el mínimo número de cambios ( $N \cdot 2^k \cdot k!$ ). Este número es muy grande, pero representa una proporción muy pequeña<sup>2</sup> ( $7,13 \times 10^{-25}$ ) respecto al total de órdenes posibles.

De todos los órdenes semilla obtenidos sólo 2 presentan el mínimo sesgo. A partir de estos 2 se pueden obtener  $2 \times 2^5 \times 5! = 7680$  órdenes, lo cual representa una proporción de  $2,92 \times 10^{-32}$  respecto al total de órdenes posibles. En la Tabla 4.5 se presentan estos dos órdenes que poseen el mínimo número de cambios de nivel (31) y el mínimo sesgo máximo en valor absoluto (4,75).

Tabla 4.5. Órdenes semilla con el mínimo número de cambios de nivel y el mínimo sesgo máximo en diseños con 32 experimentos, hallados por búsqueda exhaustiva

Ordenación 1				Ordenación 2			
Posición	# Orden	Posición	# Orden	Posición	# Orden	Posición	# Orden
1	1	17	30	1	1	17	16
2	2	18	26	2	2	18	15
3	4	19	28	3	18	19	7
4	8	20	27	4	22	20	5
5	16	21	31	5	30	21	13
6	12	22	23	6	29	22	9
7	11	23	19	7	25	23	10
8	15	24	3	8	27	24	26
9	13	25	7	9	19	25	28
10	29	26	5	10	3	26	20
11	21	27	6	11	11	27	24
12	22	28	14	12	12	28	32
13	18	29	10	13	4	29	31
14	20	30	9	14	8	30	23
15	24	31	25	15	6	31	21
16	32	32	17	16	14	32	17

<sup>2</sup>

$$\frac{48.828.036 \times 2^5 \times 5!}{32!} = 7,13 \times 10^{-25}$$

En la Tabla 4.6 se muestra un resumen de la salida del programa, en la cual se presenta el valor del sesgo máximo y el número de órdenes que presentan este valor.

Tabla 4.6. Número de órdenes semilla con las propiedades deseadas para diseños con 32 experimentos

Sesgo máximo	Número de órdenes	Porcentaje individual	Porcentaje acumulado
4,75	2	0,0000041	0,0000041
5	44	0,0000901	0,0000942
5,25	399	0,0008172	0,0009114
5,5	3363	0,0068874	0,0077988
5,75	16467	0,0337245	0,0415233
6	74657	0,1528978	0,1944211
6,25	164951	0,3378203	0,5322414
6,5	396025	0,8110607	1,3433020
6,75	579064	1,1859252	2,5292273
7	1033676	2,1169723	4,6461996
7,25	1217079	2,4925823	7,1387819
7,50	1571870	3,2191956	10,3579775
7,75	1791012	3,6679993	14,0259768
8	2479835	5,0787113	19,1046881
8,25	1992271	4,0801784	23,1848666
8,5	2961583	6,0653330	29,2501996
8,75	2209437	4,5249352	33,7751348
9	2829397	5,7946156	39,5697505
9,25	2204881	4,5156045	44,0853550
9,5	2487841	5,0951077	49,1804626
9,75	1806800	3,7003331	52,8807958
10	2962194	6,0665844	58,9473801
10,25	1690501	3,4621524	62,4095325
10,5	1782419	3,6504008	66,0599333
10,75	1557125	3,1889978	69,2489311
11	1793529	3,6731541	72,9220852
11,25	845350	1,7312800	74,6533651
11,5	1550469	3,1753663	77,8287314
11,75	725196	1,4852041	79,3139355
12	1537742	3,1493014	82,4632369
12,25	1074988	2,2015794	84,6648163
12,5	910024	1,8637325	86,5285489
12,75	502498	1,0291178	87,5576667
13	956531	1,9589791	89,5166457
13,25	411510	0,8427740	90,3594197
13,5	565427	1,1579966	91,5174163
13,75	449580	0,9207415	92,4381579
14	1003858	2,0559049	94,4940628
14,25	212664	0,4355367	94,9295995
14,5	424137	0,8686342	95,7982336
14,75	193296	0,3958709	96,1941046
15	194928	0,3992133	96,5933178
15,25	190672	0,3904970	96,9838148
15,5	113288	0,2320142	97,2158290
15,75	0	0,0000000	97,2158290
16	1359456	2,7841710	100,0000000
Total	48828036		

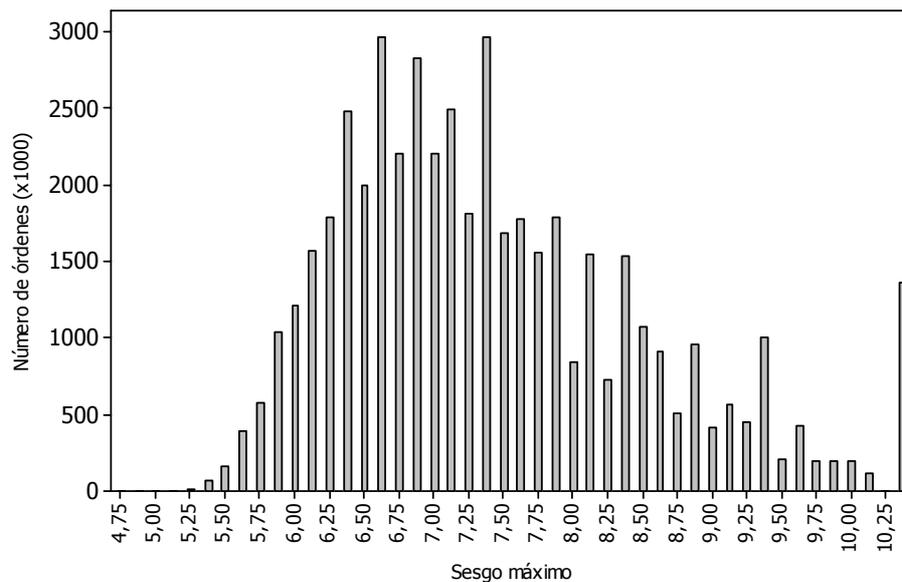


Figura 4.11 Diagrama de barras del valor de los sesgos obtenidos para todos los órdenes semilla con el mínimo número de cambios de nivel para diseños con 32 experimentos

Con base en los resultados obtenidos, podemos decir que el sesgo máximo que presentan los órdenes con el mínimo número de cambios de nivel en el diseño factorial con 32 experimentos se encuentra entre 4.75 y 16 (mínimo y máximo, respectivamente). Entre los sesgos de 4.75 a 8 se encuentran el 19% de los órdenes y entre 8.25 y 16 se concentran el 81% de los órdenes.

#### 4.3.3.2 Diseños con 64 experimentos y con 128 experimentos

Este método no ha funcionado para diseños con 64 ni 128 experimentos. En el caso de 64 experimentos, después de 2 semanas de computación y de generar unos 700 millones de ordenaciones con mínimo número de cambios, el sesgo de todas ellas siempre fue 32, es decir, el máximo posible. Otro dato relevante es que en la exploración ascendente de todos los posibles sucesores, solo se llegó a la fila 23.

En cuanto a los diseños con 128 experimentos, después de una semana de computación se generaron unos 6 millones de órdenes con mínimo número de cambios, pero los sesgos tomaron valores muy altos, situándose todos ellos en el intervalo de 53,25 a 48,120. Durante este periodo se subió solo hasta la fila 85.

## 4.4 Algoritmo de Dickinson con alimentación forzada

El problema del algoritmo de Dickinson, tal y como ha sido implementado anteriormente, es que debido al orden en que se han introducido los sucesores de cada fila, de menor a mayor, la primera ordenación que se encuentra (en torno a la cual se van realizando los cambios) da un sesgo muy grande.

Si realmente fuera posible la búsqueda exhaustiva este hecho no tendría importancia ya que aun partiendo de un sesgo malo, al final se encontraría la ordenación que da el menor posible. Pero como en la práctica esa exhaustividad no es posible para diseños con 64 o más experimentos, si se parte de una ordenación que da un sesgo muy grande, durante el tiempo en que el programa está en marcha (unas 2 semanas) realizando cambios a partir del orden de partida, no es capaz de encontrar las mejores ordenaciones.

La solución a este problema consiste en colocar los sucesores en un orden tal que la primera ordenación que encuentre sea la que proporciona el menor sesgo de todas las encontradas por los procedimientos utilizados anteriormente. De esta forma, realizando cambios sobre esa primera ordenación, que ya es buena, probablemente se obtengan otras mejores. A esta forma de proceder la hemos denominado “algoritmo de Dickinson con alimentación forzada”

### 4.4.1 Alimentación y resultados obtenidos para diseños con 64 experimentos

Con los métodos utilizados hasta ahora el mejor orden para diseños con 64 experimentos es el que proporciona un sesgo de 8,25 y es el que ha sido obtenido a través de la búsqueda aleatoria con restricciones (Tabla 4.2).

Para alimentar el algoritmo con este orden en primer lugar hay que identificar a que orden semilla corresponde. Para ello es necesario buscar la permutación que se debe realizar en los factores y cuales de ellos hay que multiplicar por -1 para que la cumpla con las restricciones de los órdenes semilla (empezar por las filas 1 y 2, y tener en orden creciente las que contienen un solo signo positivo). En nuestro caso los factores quedan de la siguiente forma:

C -A -F -D -B E

Y conducen al orden semilla que se indica en la Tabla 4.7

Tabla 4.7: Orden inicial para el algoritmo de Dickinson en diseños con 64 experimentos

Posición	# Orden						
1	1	17	13	33	37	49	55
2	2	18	9	34	45	50	51
3	34	19	25	35	47	51	35
4	36	20	26	36	15	52	33
5	4	21	58	37	31	53	41
6	20	22	62	38	29	54	42
7	24	23	54	39	61	55	10
8	8	24	53	40	63	56	14
9	16	25	49	41	64	57	46
10	12	26	57	42	56	58	38
11	11	27	59	43	52	59	6
12	27	28	43	44	50	60	22
13	19	29	44	45	18	61	30
14	3	30	48	46	17	62	32
15	7	31	40	47	21	63	28
16	5	32	39	48	23	64	60

Para que este orden sea el primero en ser obtenido, es necesario que en la lista de posibles sucesores de cada fila se coloque en primer lugar aquel que conduce a la ordenación buscada. A título de ejemplo, los sucesores de las filas 2 a la 10 (el sucesor de la fila 1 siempre es la 2) empleados en la primera versión del algoritmo de Dickinson son los que se indican en la Figura 4.12 en los que el criterio ha sido, simplemente, colocarlos en orden creciente.

Fila	Posibles sucesores
2	4, 6, 10, 18, 34
3	4, 7, 11, 19, 35
4	3, 8, 12, 20, 36
5	6, 7, 13, 21, 37
6	5, 8, 14, 22, 38
7	3, 5, 8, 15, 23, 39
8	4, 6, 7, 16, 24, 40
9	10, 11, 13, 25, 41
10	9, 12, 14, 26, 42

Figura 4.12: Posibles sucesores de las filas 2 a la 10 colocados en orden creciente (el utilizado en el algoritmo estándar)

Pero para obtener la ordenación deseada como orden de partida, los sucesores deben colocarse tal como se indica en la Figura 4.13. Si no existe un orden inferior al que se coloca en primer lugar, se coloca toda la secuencia en orden creciente, y si no existe un orden superior al primero, después del segundo se colocan el resto de sucesores en orden creciente, tal como ocurre para los sucesores de la fila 2.

Fila	Primer sucesor (el que conduce al orden de partida)	Segundo sucesor (el inmediato inferior al orden de partida)	Tercer sucesor (el inmediato superior al orden de partida)	Resto de sucesores
2	34	18		1, 4, 6, 10
3	7	4	11	19, 35
4	20	12	36	3, 8
5	13	7	21	6, 37
6	22	14	38	5, 8
7	5	3	8	15, 23, 39
8	16	7	24	4, 6, 40
9	25	13	41	10, 11
10	14	12	26	9, 42

Figura 4.13: Orden de colocación de los sucesores de cada fila

Ejecutando el programa de esta forma, se obtienen rápidamente (dentro de la primera hora) dos ordenaciones diferentes que dan un sesgo de 8. Dejando el programa en marcha durante 2 semanas se evaluaron 75 millones de órdenes, pero ninguno dio un sesgo menor. Estos órdenes de sesgo 8, que es el menor de todos los obtenidos, son los que se indican en la Tabla 4.8 (las 2 ordenaciones solo se diferencian en 4 posiciones).

Tabla 4.8: Órdenes obtenidos que proporcionan el mínimo sesgo máximo hallado con el algoritmo de Dickinson con alimentación forzada para un diseño con 64 experimentos

Posición	# Orden	Posición	# Orden	Posición	# Orden	Posición	#Orden
1	1	17	13	33	37	49	52
2	2	18	9	34	45	50	51
3	34	19	25	35	47 - 61	51	35
4	36	20	26	36	15 - 29	52	33
5	4	21	58	37	31	53	41
6	20	22	62	38	29 - 15	54	42
7	24	23	54	39	61 - 47	55	10
8	8	24	53	40	63	56	14
9	16	25	49	41	64	57	46
10	12	26	57	42	56	58	38
11	11	27	59	43	55	59	6
12	27	28	43	44	23	60	22
13	19	29	44	45	21	61	30
14	3	30	48	46	17	62	32
15	7	31	40	47	18	63	28
16	5	32	39	48	50	64	60

#### 4.4.2 Alimentación y resultados obtenidos para diseños con 128 experimentos

Con los procedimientos aplicados hasta ahora, para diseños con 128 experimentos, el mejor sesgo se obtiene aplicando el método de duplicación descrito en el capítulo 3 al orden obtenido anteriormente para 64 experimentos. De esta forma, se obtiene un ordenación que produce un sesgo máximo de 16 (para 64 el sesgo era de 8).

Utilizando un procedimiento idéntico al plantando anteriormente para los diseños de 64 experimentos, se ha ejecutado el programa de búsqueda con una disposición de los sucesores de cada fila que hace que se encuentre inmediatamente el mejor orden disponible hasta el momento. De esta forma, al igual que en el caso anterior, se empiezan a buscar nuevas ordenaciones en torno a una que ya proporciona un buen resultado.

Después de 2 semanas de computación y unos 45 millones de órdenes generados, se hallaron 2 órdenes con un sesgo mínimo de 15,0625, la primera ordenación se presenta en la Tabla 4.9 y la segunda en el Anexo 5.

Tabla 4.9: Uno de los órdenes que proporciona el mínimo sesgo máximo con el mínimo número de cambios para un diseño con 128 experimentos hallado con el algoritmo de Dickinson con alimentación forzada

Posición	# Orden						
1	1	17	25	65	73	97	44
2	2	18	26	66	74	98	12
3	4	19	18	67	90	99	76
4	3	20	17	68	89	100	92
5	67	21	49	69	93	101	91
6	68	22	50	70	94	102	83
7	72	23	52	71	30	103	84
8	71	24	51	72	29	104	82
9	7	25	115	73	61	105	81
10	8	26	116	74	62	106	65
11	40	27	124	75	58	107	66
12	39	28	123	76	57	108	70
13	47	29	107	77	121	109	69
14	48	30	108	78	122	110	101
15	16	31	106	79	126	111	102
16	15	32	105	80	125	112	104
17	31	49	97	81	127	113	100
18	32	50	98	82	128	114	99
19	24	51	114	83	120	115	35
20	23	52	113	84	119	116	43
21	21	53	117	85	103	117	59
22	22	54	118	86	111	118	63
23	54	55	86	87	112	119	55
24	53	56	85	88	110	120	56
25	37	57	87	89	109	121	64
26	38	58	88	90	45	122	60
27	6	59	96	91	46	123	28
28	5	60	95	92	42	124	20
29	13	61	79	93	41	125	19
30	14	62	80	94	33	126	27
31	10	63	78	95	34	127	11
32	9	64	77	96	36	128	75

#### 4.5 Resumen. Mejores sesgos encontrados

La Tabla 4.10 muestra los mejores órdenes encontrados para cada tipo de diseño y el procedimiento por el cual han sido obtenidos.

Tabla 4.10: Mejores órdenes hallados para cada tipo de diseño y procedimiento por el que han sido encontrados

Número de experimentos	Conocido	Sesgos			¿Valor mínimo asegurado?
		Expansión del mejor orden del diseño anterior	Búsqueda aleatoria con restricciones	Búsqueda exhaustiva	
8	2	-	-	-	Si
16	4	4	-	-	Si
32	-	8	5,25	4,75	Si
64	-	9,5	8,25	-	No
128	-	16	17,625	-	No

Para diseños con 32 experimentos se ha obtenido el mínimo sesgo posible realizando una búsqueda exhaustiva con el algoritmo de Dickinson. Para diseños con mayor número de experimentos esta estrategia no es viable porque las posibles ordenaciones crecen de forma exponencial y el tiempo de computación se hace demasiado grande. Para encontrar buenos órdenes a pesar de esa limitación, se ha forzado que el algoritmo genere en primer lugar el mejor orden de los que se habían obtenido hasta el momento, para que explorando en torno a ese orden se puedan encontrar otros mejores.

En el caso de diseños con 64 experimentos el orden de alimentación es el que se ha obtenido utilizando la estrategia de búsqueda aleatoria con restricciones, logrando pasar del sesgo 8,25 del orden de alimentación a 8,0 para el mejor orden obtenido.

Para diseños con 128 experimentos la mejor ordenación se obtiene al expandir el mejor orden obtenido para diseños con 64 experimentos. Así se consigue una ordenación que da un sesgo de 16, que tras pasar por el algoritmo de Dickinson con alimentación forzada se ha transformado en un orden con el sesgo de 15,0625.

## **4.6 Aportaciones**

Este capítulo contiene las siguientes aportaciones:

- Se han hallado los dos únicos órdenes semilla que producen diseños con el mínimo número de cambios en los factores y mínimo sesgo máximo para diseños con 32 experimentos. Hasta ahora estos órdenes solo se tenían para diseños con 8 y 16 experimentos (además de los de 4 experimentos, que es un caso elemental).
- Para diseños con 64 y 128 experimentos se han hallado órdenes que presentan el mínimo número de cambios y sesgos máximos 4 veces menores que los hallados hasta el momento por las propuestas de Jan y Wang (1995) y Cheng et al. (1998).
- Se han desarrollado metodologías (búsqueda aleatoria con restricciones y algoritmo de Dickinson con alimentación forzada) que utilizadas con ordenadores más rápidos o durante mayor tiempo, podrían producir órdenes mejores para los diseños con 64 y 128 experimentos.

# Capítulo 5



# Capítulo 5

## Obtención de nuevos órdenes a partir de los órdenes semilla

Una vez obtenido un orden, al que llamamos “orden semilla”, que cumple las propiedades de mínimo número de cambios en los factores y mínimo sesgo máximo en valor absoluto, es posible deducir a partir de éste nuevos órdenes que cumplan igualmente con esas propiedades. Para diseños con 4, 8 y 16 experimentos, ya se conocían los órdenes semilla, y para diseños con 32, 64 y 128 experimentos, se han deducido en el capítulo anterior. La Tabla 5.1 muestra la situación en que nos encontramos.

Tal como se justifica más adelante, si para un diseño  $2^k$  se tiene un orden que cumpla con las propiedades deseadas, a partir de este se pueden deducir otros  $k!2^k$  con las mismas propiedades. Esta posibilidad de deducir nuevos órdenes a partir de uno dado permite que la selección del orden en que finalmente se realizará la experimentación no se realice sólo entre los pocos órdenes que se hayan encontrado con las propiedades deseadas, sino que se podrá seleccionar uno al azar de entre un gran conjunto.

En este capítulo se presenta cómo deducir los nuevos órdenes a través de dos métodos que tienen distintas peculiaridades, y se detalla también la forma de implementar estos procedimientos en el paquete de software estadístico Minitab.

Tabla 5.1: Órdenes semilla obtenidos

# Exp.	# Órdenes semilla	Comentarios
4	1	Fácil de deducir. Solo existen un total de 24 órdenes.
8	1	En el trabajo de de León (2004) aparecen todos los órdenes que se pueden deducir de esta semilla. Estos órdenes son todos, y a la vez los únicos, que tienen las propiedades deseadas.
16	16	En el trabajo de de León (2004) aparecen estos 16 órdenes semilla. A partir de estos se pueden deducir todos los órdenes que tienen las propiedades deseadas.
32	2	Obtenidos utilizando el algoritmo de Dickinson para la búsqueda exhaustiva. Los órdenes obtenidos eran desconocidos hasta ahora y se puede garantizar que no existen otros que tengan sesgos inferiores.
64	2	Obtenidos utilizando el algoritmo de Dickinson con alimentación forzada. El orden de alimentación ha sido obtenido por el método de la búsqueda aleatoria con restricciones. No se conocen órdenes que den un sesgo menor, pero no se puede garantizar que este sea el mínimo posible.
128	2	Obtenidos utilizando el algoritmo de Dickinson con alimentación forzada. El orden de alimentación ha sido obtenido por duplicación de uno de los mejores órdenes obtenidos para 64 experimentos. Al igual que en el caso anterior, no se conocen órdenes que den un sesgo menor, pero no se puede garantizar que este sea el mínimo posible.

## 5.1 Método de permutación y cambios de signo

En cualquier matriz de un diseño completo se cumplen las siguientes propiedades:

1. Al permutar el orden de las columnas se mantiene el valor de los efectos.

Obviamente, el valor de los efectos está relacionado con la secuencia de signos, no con el nombre del efecto a que corresponden. Así, si intercambiamos la secuencia de signos correspondientes a los efectos principales de A y B, el valor que antes correspondía al efecto principal de A pasará a ser el de B y viceversa. También cambiará la asignación de efectos a las interacciones en las que intervienen estos factores, pero no cambiarán los números concretos, es decir, no cambiarán los valores de los efectos, independientemente de cual sea su nueva asignación.

- Al permutar el orden de las columnas se obtiene otra matriz con el mismo número de cambios de los factores.

La Figura 5.1 muestra, en la izquierda, la matriz de un diseño  $2^3$  con mínimo número de cambios en los factores y en la derecha, la matriz que resulta de cambiar el orden de las columnas B y C.

Evidentemente, el número de cambios se mantiene. En general, si entre dos filas consecutivas hay un cambio de signos, este cambio está situado en una columna concreta, y se mantiene con independencia del lugar donde se coloque esa columna.

Figura 5.1: Las matrices tienen cambiadas las columnas correspondientes a los factores B y C, pero el número total de cambios se mantiene constante.

Orden	Factores			Orden	Factores		
	A	B	C		A	B	C
1	-1	-1	-1	1	-1	-1	-1
2	1	-1	-1	2	1	-1	-1
6	1	-1	1	4	1	1	-1
8	1	1	1	8	1	1	1
4	1	1	-1	6	1	-1	1
3	-1	1	-1	5	-1	-1	1
7	-1	1	1	7	-1	1	1
5	-1	-1	1	3	-1	1	-1
Cambios	2	2	3	Cambios	2	3	2

- Al multiplicar por -1 una o varias columnas de la matriz de diseño, se mantiene el valor absoluto de los efectos

Efectivamente, al multiplicar por -1 los signos de una columna de la matriz de diseño se multiplica también por -1 el valor del efecto correspondiente, así como las interacciones en las que participa dicho factor. Si cambian los signos de un número par de los factores que intervienen en la interacción, esta mantiene su valor.

En cualquier caso, los valores absolutos se mantienen para todos los efectos al multiplicar una o varias columnas de la matriz de diseño por -1. Por tanto, también se mantendrá el sesgo máximo en valor absoluto, que es el valor que nos interesa.

- Al multiplicar por -1 una o varias columnas de la matriz de diseño, se mantiene el número de cambios de nivel de los factores.

Los cambios de nivel (cambios de signo) se producen al pasar de una fila a otra a lo largo de una columna. Evidentemente, cambiar todos los signos a una columna mantiene el número de cambios de signo.

5. En una matriz de diseño, permutar el orden de las columnas o multiplicar algunas de ellas por -1, conduce a otra matriz de diseño.

En la matriz de diseño de un diseño factorial  $2^k$ , al multiplicar por -1 los signos de una columna no quedará ninguna fila repetida, ya que antes (se trataba de una matriz de diseño) tampoco la había. Cambiar el orden de las columnas es como cambiar el nombre de los factores y eso, obviamente, no deshace las propiedades de la matriz de diseño.

En definitiva, cambiar el orden de las columnas en la matriz de diseño o multiplicar algunas de ellas por -1 nos conduce a otra matriz de diseño en la que cambia el orden de realización de los experimentos, pero mantiene el número de cambios de nivel y también el valor absoluto de los efectos. A título de ejemplo en las Tabla 5.2 se presenta una matriz de diseño con el cálculo de los efectos, y en la Tabla 5.3 se tiene a misma matriz cambiando el orden de las columnas A y B y cambiando los signos de C. Puede comprobarse que las propiedades de número de cambios y mínimo sesgo máximo absoluto se mantienen.

Tabla 5.2: Matriz de diseño e interacciones que cumplen las propiedades deseadas

Orden	A	B	C	AB	AC	BC	ABC	y
1	-1	-1	-1	1	1	1	-1	1
2	1	-1	-1	-1	-1	1	1	2
6	1	-1	1	-1	1	-1	-1	3
8	1	1	1	1	1	1	1	4
4	1	1	-1	1	-1	-1	-1	5
3	-1	1	-1	-1	1	-1	1	6
7	-1	1	1	-1	-1	1	-1	7
5	-1	-1	1	1	-1	-1	1	8
Cambios	2	2	3					
Efectos	-2	2	2	0	-2	-2	1	

Tabla 5.3: Matriz de diseño deducida de la anterior cambiando de orden las columnas A y B, y cambiando los signos de C.

Orden	A	B	C	AB	AC	BC	ABC	Y
5	-1	-1	1	1	1	1	-1	1
7	-1	1	1	-1	-1	1	1	2
3	-1	1	-1	-1	1	-1	-1	3
4	1	1	-1	1	1	1	1	4
8	1	1	1	1	-1	-1	-1	5
6	1	-1	1	-1	1	-1	1	6
2	1	-1	-1	-1	-1	1	-1	7
1	-1	-1	-1	1	-1	-1	1	8
Cambios	2	2	3					
Efectos	2	-2	-2	0	2	2	-1	

## 5.2 Método de la columna de expansión

Entre un orden con las propiedades deseadas (mínimo número de cambios y mínimo sesgo máximo en valor absoluto) en un diseño  $2^k$  y otro orden con las mismas propiedades en un diseño  $2^{k+1}$  puede definirse lo que hemos denominado “columna de expansión” que sirve para pasar de uno a otro. En este apartado veremos su construcción, las propiedades en que basa y las posibilidades que presenta.

### 5.2.1 Paso de un diseño $2^2$ a otro $2^3$

Los diseños  $2^2$  son muy sencillos y todos los órdenes en que se pueden presentar (4!) pueden estudiarse rápidamente de forma exhaustiva. Se puede comprobar que los únicos órdenes que presentan el mínimo número de cambios son los que aparecen en la Tabla 5.4. En este caso no tiene sentido hablar de mínimo sesgo máximo en valor absoluto (msma), ya que para todos los órdenes se tiene el mismo valor.

Tabla 5.4. Órdenes de experimentación en el diseño  $2^2$  que presentan el mínimo número de cambios (todos los órdenes de este diseño tienen el mismo msma).

Orden 1	Orden 2	Orden 3	Orden 4	Orden 5	Orden 6	Orden 7	Orden 8
1	1	2	2	3	3	4	4
2	3	1	4	4	1	2	3
4	4	3	3	2	2	1	1
3	2	4	1	1	4	3	2

Tomemos un diseño  $2^3$  en el orden que aparece en primer lugar en la tabla de de León (2004), ver Anexo 1. Este diseño se puede considerar como un diseño  $2^2$  replicado para los 2 primeros factores, tal como se muestra en la Figura 5.2.

Orden Estándar	Factores		
	A	B	C
1	-1	-1	-1
2	1	-1	-1
6	1	-1	1
8	1	1	1
4	1	1	-1
3	-1	1	-1
7	-1	1	1
5	-1	-1	1
Cambios	2	2	3

Primer diseño  $2^2$  (filas 1, 2, 6, 8)

Segundo diseño  $2^2$  (filas 4, 3, 7, 5)

Figura 5.2: Diseño  $2^3$  que cumple con las propiedades deseadas visto como un  $2^2$  replicado

Sobre este sencillo ejemplo se pueden hacer las siguientes consideraciones, fácilmente generalizables.

1. El primer diseño corresponde a las filas que tienen signo  $-$  en el tercer factor, y el segundo diseño corresponde a las filas que tienen signo  $+$  en ese último factor.

Evidentemente esto siempre es así. Un diseño  $2^k$  se puede ver como un diseño  $2^{k-1}$  replicado, sea cual fuere el factor que se elimina. Esto es un caso particular de la propiedad proyectiva de los diseños factoriales.

2. Uno de los diseños  $2^2$  contiene las 4 primeras filas (en el orden estándar de matriz de diseño) del diseño  $2^3$ , y el otro diseño  $2^2$  contiene las otras 4.

En general, pensemos en el orden estándar de la matriz de un diseño  $2^k$ . Las  $2^{k-1}$  primeras filas corresponden a los signos  $-$  del último factor. Por tanto, siempre tendremos un diseño  $2^{k-1}$  completo en las  $2^{k-1}$  primeras filas para los  $k-1$  primeros factores.

3. Las réplicas están alejadas 4 unidades en la escala de orden de la matriz de diseño.

Efectivamente, la matriz de un diseño  $2^3$  está constituida por 2 matrices idénticas para los 2 primeros factores, de forma que para esos 2 primeros factores la fila 1 (primera fila con signo – en el tercer factor) es idéntica a la fila 5 (primera fila con signo + en el tercer factor) y así sucesivamente. En general, considerando solo los  $k-1$  primeros factores de un diseño  $2^k$ , y siendo  $i$  una de las primeras  $\frac{2^k}{2}$  filas de la matriz de diseño en orden estándar, tendremos que para esos  $k-1$  primeros factores, la fila  $i$  coincidirá con la fila  $\frac{2^k}{2} + i$ .

En la Figura 5.2 el primer diseño  $2^2$  presenta un orden que tiene las propiedades deseadas (es el primero que aparece en la Tabla 5.4) y a la vista de las propiedades comentadas puede definirse un procedimiento para pasar de ese diseño  $2^2$  al diseño  $2^3$  de la siguiente forma:

- En el diseño de partida (en nuestro ejemplo  $2^2$ ) definimos  $O_1, O_2, O_3$  y  $O_4$  como los números de fila de la matriz de diseño en orden estándar que corresponden a las filas 1, 2, 3, y 4 del orden de ejecución planteado. Es decir,  $O_1$  es el número de orden (en la matriz de diseño en orden estándar) de la fila que aparece en primar lugar,  $O_2$  es el de la segunda fila, etc. En nuestro ejemplo,  $O_1 = 1, O_2 = 2, O_3 = 4$  y  $O_4 = 3$ .
- Partiendo de los órdenes del diseño  $2^2$  definimos una columna de expansión tal como se indica en la Figura 5.3 para pasar a los órdenes del diseño  $2^3$

Orden de partida Diseño $2^2$	Columna de expansión	Nuevo orden Diseño $2^3$
	$O_1$	<b>1</b>
	$O_2$	<b>2</b>
1	$O_2 + 4$	6
2	$O_3 + 4$	8
4	$O_3$	<b>4</b>
3	$O_4$	<b>3</b>
	$O_4 + 4$	7
	$O_1 + 4$	5

Figura 5.3: Columna de expansión para pasar del diseño  $2^2$  al diseño  $2^3$

Puede observarse que en la columna de expansión, el cambio de un número de orden de la primera mitad a otro de la segunda mitad que viene correlativo en el orden de ejecución siempre es del tipo  $O_i$  y  $O_i + 4$  ya que si una fila está en la primera mitad de la matriz de diseño en orden estándar, y la otra fila está en la segunda mitad, seguro que cambia el signo del tercer factor, y como solo puede cambiar un signo entre dos filas consecutivas, los demás signos deberán ser iguales.

La lista de asignación de órdenes de la Figura 5.3 puede aplicarse a todos los órdenes de diseños  $2^2$  que cumplan con las propiedades deseadas, tal como se indica en la Figura 5.4.

Ordenes de partida Diseño $2^2$	Columna de expansión	Nuevos ordenes Diseño $2^3$
	$O_1$	<b>1 1 2 2 3 3 4 4</b>
	$O_2$	<b>2 3 1 4 4 1 2 3</b>
1 1 2 2 3 3 4 4	$O_2+4$	6 7 5 8 8 5 6 7
2 3 1 4 4 1 2 3	$O_3+4$	8 8 7 7 6 6 5 5
4 4 3 3 2 2 1 1	$O_3$	<b>4 4 3 3 2 2 1 1</b>
3 2 4 1 1 4 3 2	$O_4$	<b>3 2 4 1 1 4 3 2</b>
	$O_4+7$	7 6 8 5 5 8 7 6
	$O_1+4$	5 5 6 6 7 7 8 8

Figura 5.4: Aplicación de una columna de expansión a todos los órdenes elegidos del diseño  $2^2$

De forma totalmente análoga a la que hemos visto, en vez de colocar los números de orden del diseño  $2^2$  en las filas correspondientes a los signos - del tercer factor, los podemos colocar en los signos +, obteniéndose una columna de expansión complementaria a la anterior, y que aplicada a los órdenes de partida permite obtener otros 8 órdenes para un diseño  $2^3$ , todos ellos con las propiedades deseadas.

Órdenes de partida Diseño $2^2$	Columna de expansión	Nuevos órdenes Diseño $2^3$
	$O_1 + 4$	5 5 6 6 7 7 8 8
	$O_2 + 4$	7 6 8 5 5 8 7 6
3 2 4 1 1 4 3 2	$O_2$	3 2 4 1 1 4 3 2
4 4 3 3 2 2 1 1	$O_3$	4 4 3 3 2 2 1 1
2 3 1 4 4 1 2 3	$O_3 + 4$	8 8 7 7 6 6 5 5
1 1 2 2 3 3 4 4	$O_4 + 4$	6 7 5 8 8 5 6 7
	$O_4$	2 3 1 4 4 1 2 3
	$O_1$	1 1 2 2 3 3 4 4

Figura 5.5: Columna de expansión complementaria a la que aparece en la Figura 5.3

Si en vez de colocar en último lugar el factor C, colocamos el A o el B, de forma análoga podemos obtener nuevas columnas de expansión. En total se pueden obtener las 6 que figuran en la Tabla 5.5

Tabla 5.5. Columnas de expansión para pasar de diseños  $2^2$  a diseños  $2^3$ . El encabezado de cada columna indica que factor se coloca en último lugar, y si los órdenes del diseño  $2^2$  se asignan a los signos positivos o negativos de este último factor

A -	A +	B -	B +	C -	C +
$O_1$	$O_1 + 4$	$O_1$	$O_1 + 4$	$O_1$	$O_1 + 4$
$O_1 + 4$	$O_1$	$O_2$	$O_2 + 4$	$O_2$	$O_2 + 4$
$O_4 + 4$	$O_4$	$O_3$	$O_3 + 4$	$O_2 + 4$	$O_2$
$O_3 + 4$	$O_3$	$O_3 + 4$	$O_3$	$O_3 + 4$	$O_3$
$O_2 + 4$	$O_2$	$O_2 + 4$	$O_2$	$O_3$	$O_3 + 4$
$O_2$	$O_2 + 4$	$O_1 + 4$	$O_1$	$O_4$	$O_4 + 4$
$O_3$	$O_3 + 4$	$O_4 + 4$	$O_4$	$O_4 + 4$	$O_4$
$O_4$	$O_4 + 4$	$O_4$	$O_4 + 4$	$O_1 + 4$	$O_1$

Aplicando cada una de estas columnas de expansión (6 en total) a cada uno de los órdenes del diseño  $2^2$  con las propiedades deseadas (son 8), se tienen los 48 órdenes de un diseño  $2^3$  que igualmente presentan las propiedades deseadas, y que son los que aparecen en la tabla de de León (2004), ver Anexo 1.

Tabla 5.6. Órdenes de experimentación en un diseño  $2^3$  con las propiedades deseadas generados a partir de los 8 órdenes que reúnen esas propiedades en un diseño  $2^2$  y las 6 columnas de expansión

A -	Órdenes							
$O_1$	1	1	2	2	3	3	4	4
$O_1+4$	5	5	6	6	7	7	8	8
$O_4+4$	7	6	8	5	5	8	7	6
$O_3+4$	8	8	7	7	6	6	5	5
$O_2+4$	6	7	5	8	8	5	6	7
$O_2$	2	3	1	4	4	1	2	3
$O_3$	4	4	3	3	2	2	1	1
$O_4$	3	2	4	1	1	4	3	2

A +	Órdenes							
$O_1+4$	5	5	6	6	7	7	8	8
$O_1$	1	1	2	2	3	3	4	4
$O_4$	2	3	1	4	4	1	2	3
$O_3$	4	4	3	3	2	2	1	1
$O_2$	3	2	4	1	1	4	3	2
$O_2+4$	7	6	8	5	5	8	7	6
$O_3+4$	8	8	7	7	6	6	5	5
$O_4+4$	6	7	5	8	8	5	6	7

B -	Órdenes							
$O_1$	1	1	2	2	3	3	4	4
$O_2$	2	3	1	4	4	1	2	3
$O_3$	4	4	3	3	2	2	1	1
$O_3+4$	8	8	7	7	6	6	5	5
$O_2+4$	6	7	5	8	8	5	6	7
$O_1+4$	5	5	6	6	7	7	8	8
$O_4+4$	7	6	8	5	5	8	7	6
$O_4$	3	2	4	1	1	4	3	2

B +	Órdenes							
$O_1+4$	5	5	6	6	7	7	8	8
$O_2+4$	6	7	5	8	8	5	6	7
$O_3+4$	8	8	7	7	6	6	5	5
$O_3$	4	4	3	3	2	2	1	1
$O_2$	2	3	1	4	4	1	2	3
$O_1$	1	1	2	2	3	3	4	4
$O_4$	3	2	4	1	1	4	3	2
$O_4+4$	7	6	8	5	5	8	7	6

C -	Órdenes							
$O_1$	1	1	2	2	3	3	4	4
$O_2$	2	3	1	4	4	1	2	3
$O_2+4$	6	7	5	8	8	5	6	7
$O_3+4$	8	8	7	7	6	6	5	5
$O_3$	4	4	3	3	2	2	1	1
$O_4$	3	2	4	1	1	4	3	2
$O_4+4$	7	6	8	5	5	8	7	6
$O_1+4$	5	5	6	6	7	7	8	8

C +	Órdenes							
$O_1+4$	5	5	6	6	7	7	8	8
$O_2+4$	7	6	8	5	5	8	7	6
$O_2$	3	2	4	1	1	4	3	2
$O_3$	4	4	3	3	2	2	1	1
$O_3+4$	8	8	7	7	6	6	5	5
$O_4+4$	6	7	5	8	8	5	6	7
$O_4$	2	3	1	4	4	1	2	3
$O_1$	1	1	2	2	3	3	4	4

### 5.2.2 Generalización del procedimiento descrito

De forma análoga a la planteada al pasar de un diseño  $2^2$  a otro  $2^3$ , se verá como pasar de este último a un  $2^4$  siguiendo los pasos que a continuación se indican.

1. Tomar el orden del diseño  $2^3$  elegido. En este caso será: 1, 2, 6, 8, 4, 3, 7 y 5 (obtenido en la Tabla 5.3).
2. Elegir un orden para el diseño  $2^4$  que cumpla con las propiedades deseadas. Por ejemplo, elegiremos el: 1, 2, 10, 14, 16, 15, 7, 3, 11, 12, 4, 8, 6, 5, 13, 9. La matriz de diseño que corresponde a este orden es la que se indica en la Figura 5.6.

Orden de ejecución	Orden estándar	Factores			
		A	B	C	D
1	1	-1	-1	-1	-1
2	2	1	-1	-1	-1
3	10	1	-1	-1	1
4	14	1	-1	1	1
5	16	1	1	1	1
6	15	-1	1	1	1
7	7	-1	1	1	-1
8	3	-1	1	-1	-1
9	11	-1	1	-1	1
10	12	1	1	-1	1
11	4	1	1	-1	-1
12	8	1	1	1	-1
13	6	1	-1	1	-1
14	5	-1	-1	1	-1
15	13	-1	-1	1	1
16	9	-1	1	-1	1

Figura 5.6: Matriz de un diseño  $2^4$  en un orden que cumple con las propiedades deseadas

3. Deducir la columna de expansión entre los órdenes de uno y otro diseño.

Al igual que en el ejemplo anterior al pasar del diseño  $2^2$  al  $2^3$ , asignamos primero los órdenes correspondientes a los 8 primeros experimentos (en orden estándar) y a continuación asignamos el resto.

Para la primera mitad de filas basta con identificar en que posición se presentan esos órdenes en el diseño de partida. Por ejemplo, el orden 3 se presenta en la fila 6 del diseño de partida, por tanto, junto al orden 3 en el diseño final colocaremos  $O_6$ . El 4 se presenta en 5ª posición, luego junto a ese 4 colocaremos  $O_5$ , y así sucesivamente.

Para la segunda mitad hay que identificar a cual de la primera mitad hay que sumarle 8 posiciones. Así, por ejemplo, en la tercera fila, el orden 10 es el correspondiente al de  $O_2$  (en  $O_2$  el orden es igual a 2) más 8. En la cuarta fila será el orden que da la posición 6, que es  $O_5$ , situado en la fila 13, y así sucesivamente.

Orden de Ejecución Diseño $2^3$	Orden de partida Diseño $2^3$	Columna de expansión	Nuevo orden Diseño $2^4$
		$O_1$	1
		$O_2$	2
		$O_2 + 8$	10
		$O_3 + 8$	14
1	1	$O_4 + 8$	16
2	2	$O_7 + 8$	15
3	6	$O_7$	7
4	8	$O_6$	3
5	4	$O_6 + 8$	11
6	3	$O_5 + 8$	12
7	7	$O_5$	4
8	5	$O_4$	8
		$O_3$	6
		$O_8$	5
		$O_8 + 8$	13
		$O_1 + 8$	9

Figura 5.7: Columna de expansión para pasar de un diseño  $2^3$  a otro  $2^4$

- Aplicar la columna de expansión a cualquier orden que tenga las propiedades deseadas en un diseño  $2^3$ . Por ejemplo, si lo aplicamos al segundo que aparece en la tabla de de León (2004), tendremos:

Orden de partida Diseño $2^3$	Columna de expansión	Nuevo orden Diseño $2^4$
	$O_1$	5
	$O_2$	7
	$O_2 + 8$	15
	$O_3 + 8$	11
5	$O_4 + 8$	12
7	$O_7 + 8$	10
3	$O_7$	2
4	$O_6$	6
8	$O_6 + 8$	14
6	$O_5 + 8$	16
2	$O_5$	8
1	$O_4$	4
	$O_3$	3
	$O_8$	1
	$O_8 + 8$	9
	$O_1 + 8$	13

Aplicando la columna de expansión a cualquier ordenación de un diseño  $2^3$  que tenga las propiedades deseadas, conduce a un orden del diseño  $2^4$  con las mismas propiedades. El procedimiento es el mismo y se puede generalizar para pasar de diseños  $2^k$  a diseños  $2^{k+1}$ .

Obsérvese que hemos ubicado la primera mitad de las filas atendiendo a los signos – en el último factor de la matriz del diseño final. También la podríamos haber asignado a los signos + de esta última columna, o a los – o los + de cualquiera de las otras 3, obteniéndose las columnas de expansión que se indican en la Tabla 5.7

Tabla 5.7: Columnas de expansión deducidas a partir de la matriz de la Figura 5.6. El título de la columna indica el factor y el signo al que se asignan a primera mitad de los experimentos

A -	A +	B -	B +	C -	C +	D -	D +
$O_1$	$O_1 + 8$						
$O_1 + 8$	$O_1$	$O_2$	$O_2 + 8$	$O_2$	$O_2 + 8$	$O_2$	$O_2 + 8$
$O_6 + 8$	$O_6$	$O_3$	$O_3 + 8$	$O_5$	$O_5 + 8$	$O_2 + 8$	$O_2$
$O_7 + 8$	$O_7$	$O_4$	$O_4 + 8$	$O_5 + 8$	$O_5$	$O_5 + 8$	$O_5$
$O_4 + 8$	$O_4$						
$O_4$	$O_4 + 8$	$O_7 + 8$	$O_7$	$O_7 + 8$	$O_7$	$O_7 + 8$	$O_7$
$O_3$	$O_3 + 8$	$O_6 + 8$	$O_6$	$O_8 + 8$	$O_8$	$O_7$	$O_7 + 8$
$O_2$	$O_2 + 8$	$O_1 + 8$	$O_1$	$O_8$	$O_8 + 8$	$O_8$	$O_8 + 8$
$O_5$	$O_5 + 8$	$O_8 + 8$	$O_8$	$O_7$	$O_7 + 8$	$O_8 + 8$	$O_8$
$O_5 + 8$	$O_5$	$O_3 + 8$	$O_3$	$O_4$	$O_4 + 8$	$O_3 + 8$	$O_3$
$O_2 + 8$	$O_2$	$O_2 + 8$	$O_2$	$O_3$	$O_3 + 8$	$O_3$	$O_3 + 8$
$O_3 + 8$	$O_3$	$O_5 + 8$	$O_5$	$O_3 + 8$	$O_3$	$O_4$	$O_4 + 8$
$O_8 + 8$	$O_8$	$O_5$	$O_5 + 8$	$O_2 + 8$	$O_2$	$O_5$	$O_5 + 8$
$O_8$	$O_8 + 8$	$O_6$	$O_6 + 8$	$O_1 + 8$	$O_1$	$O_6$	$O_6 + 8$
$O_7$	$O_7 + 8$	$O_7$	$O_7 + 8$	$O_6 + 8$	$O_6$	$O_6 + 8$	$O_6$
$O_6$	$O_6 + 8$	$O_8$	$O_8 + 8$	$O_6$	$O_6 + 8$	$O_1 + 8$	$O_1$

De cada uno de los órdenes del diseño  $2^4$  se pueden deducir 8 columnas de expansión, y a cada una estas se le pueden aplicar los 48 órdenes del diseño  $2^3$  que cumplen con las propiedades deseadas, de forma que a partir de cada orden de un diseño  $2^4$  se pueden deducir otros 384 ( $=8 \times 48$ ). Este es también el valor que se obtendría con el método de las permutaciones y cambios de signo presentado en el apartado anterior ( $4! \cdot 2^4 = 384$ ).

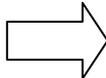
### 5.2.3 Ventajas del método de la columna de expansión

Este método también permite pasar de diseños  $2^k$  a diseños  $2^{k+1}$  aunque de estos últimos no se conozca ninguna ordenación que cumpla con las propiedades deseadas.

En el capítulo 3 se presentó una forma de “ampliar” un diseño  $2^k$  con las propiedades deseadas para obtener otro  $2^{k+1}$  que también presenta mínimo número de cambios en los niveles de los factores y un sesgo que es el doble del que tenía el diseño de partida. Sea por ejemplo el diseño  $2^3$  con el orden 1, 2, 6, 8, 4, 3, 7 y 5, que cumple con las

propiedades deseadas. Aplicando el método propuesto se llega al diseño  $2^4$  con orden: 1, 9, 10, 2, 6, 14, 16, 8, 4, 12, 11, 3, 7, 15, 13 y 5, que tiene mínimo número de cambios y sesgo máximo doble que en el diseño anterior, el cual en este caso es también el mínimo posible.

Orden	Factores		
	A	B	C
1	-1	-1	-1
2	1	-1	-1
6	1	-1	1
8	1	1	1
4	1	1	-1
3	-1	1	-1
7	-1	1	1
5	-1	-1	1
Cambios	2	2	3



Orden	Factores			
	A	B	C	D
1	-1	-1	-1	-1
9	-1	-1	-1	1
10	1	-1	-1	1
2	1	-1	-1	-1
6	1	-1	1	-1
14	1	-1	1	1
16	1	1	1	1
8	1	1	1	-1
4	1	1	-1	-1
12	1	1	-1	1
11	-1	1	-1	1
3	-1	1	-1	-1
7	-1	1	1	-1
15	-1	1	1	1
13	-1	-1	1	1
5	-1	-1	1	-1
Cambios	2	2	3	8

Figura 5.8: Diseño de partida y diseño obtenido al aplicar el método descrito en el capítulo 3

Esta “duplicación” de la matriz de partida, equivale a aplicar un tipo concreto de columna de expansión, que siempre conduce a órdenes con mínimo número de cambios y sesgo bajo, y es el único método que es posible aplicar cuando el número de factores es mayor de 7.

La Figura 5.9 muestra la columna de expansión que hace crecer el diseño aplicando el método descrito. El patrón que sigue está muy claro y es inmediato generalizarlo a cualquier número de factores.

Orden de partida Diseño $2^3$	Columna de expansión	Nuevo orden Diseño $2^4$
	$O_1$	1
	$O_1 + 8$	9
	$O_2 + 8$	10
	$O_2$	2
1	$O_3$	6
2	$O_3 + 8$	14
6	$O_4 + 8$	16
8	$O_4$	8
4	$O_5$	4
3	$O_5 + 8$	12
7	$O_6 + 8$	11
5	$O_6$	3
	$O_7$	7
	$O_7 + 8$	15
	$O_8 + 8$	13
	$O_8$	5

Figura 5.9: Columna de expansión que aplica el método descrito en el capítulo 3

Además, este procedimiento permite una fácil implementación en un programa informático y permite también forzar determinados caminos en la selección de la ordenación deseada.

Algunos autores como Dickinson (1974), Joiner y Campbell (1976), Cheng (1985), Coster y Cheng (1988), John (1990), Steinberg y Cheng (1991), Coster (1993) y Tack y Vandebroek (2002) muestran especial interés en que el sesgo máximo no esté ubicado en los efectos principales. En el planteamiento realizado en esta tesis no se ha prestado atención al lugar (efecto) donde aparece el sesgo máximo ya que forzar a que dicho sesgo máximo esté situado en interacciones de orden superior puede acabar colocándolo en un efecto principal si ese orden se utiliza para un diseño fraccional y la interacción “perjudicada” se utiliza como generador.

Sin embargo, si se conocen órdenes que cumplan con las propiedades deseadas, pueden seleccionarse, de entre todas las posibles columnas de expansión, aquellas que conducen a diseños que tienen los efectos principales libres del sesgo máximo. La forma de implantación informática que se utiliza para aplicar este procedimiento permite conseguir esto fácilmente, dejando que la selección aleatoria transcurra solo por los

caminos (columnas de expansión) que conducen a los órdenes con las propiedades que interesen.

### 5.3 Implementación en MINITAB de los métodos propuestos

Minitab es un paquete de software estadístico de uso extendido en los ambientes industriales y de control de calidad. A esta extensión ha contribuido el que se haya convertido en el software estadístico “oficial” (recomendado por los mayores consultores y utilizado por casi todas las grandes empresas) en la implantación de programas de mejora Seis Sigma. Aunque no es un paquete especializado en diseño de experimentos constituye una excelente ayuda tanto para elegir y plantear el diseño más adecuado como para su posterior análisis. Además es, seguramente, uno de los más utilizados.

Permite seleccionar un diseño para entre 2 y 15 factores con entre 4 y 128 experimentos. Estos diseños están resumidos en el cuadro de la Figura 5.10.

Available Factorial Designs (with Resolution)															
	Factors														
Runs	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
4	Full	III													
8		Full	IV	III	III	III									
16			Full	V	IV	IV	IV	III							
32				Full	VI	IV	IV	IV	IV	IV	IV	IV	IV	IV	
64					Full	VII	V	IV							
128						Full	VIII	VI	V	V	IV	IV	IV	IV	

Available Resolution III Plackett-Burman Designs					
Factors	Runs	Factors	Runs	Factors	Runs
2-7	12,20,24,28,...,48	20-23	24,28,32,36,...,48	36-39	40,44,48
8-11	12,20,24,28,...,48	24-27	28,32,36,40,44,48	40-43	44,48
12-15	20,24,28,36,...,48	28-31	32,36,40,44,48	44-47	48
16-19	20,24,28,32,...,48	32-35	36,40,44,48		

Figura 5.10: Diseños factoriales disponibles en Minitab

En el proceso de selección del diseño existe un botón de “Opciones” que conduce a la ventana de diálogo de la Figura 5.11. En esta ventana está marcado por defecto la opción de presentar los experimentos en orden aleatorio. Si se quita esta opción se presentan en orden estándar.

Vimos en el capítulo 1 que el orden aleatorio no garantiza las propiedades deseadas, y puede encarecer notablemente el desarrollo del plan experimental al requerir muchos cambios de nivel. Si se elimina esta opción, el diseño se presenta en orden estándar, el cual es bien sabido que presenta inconvenientes importantes.

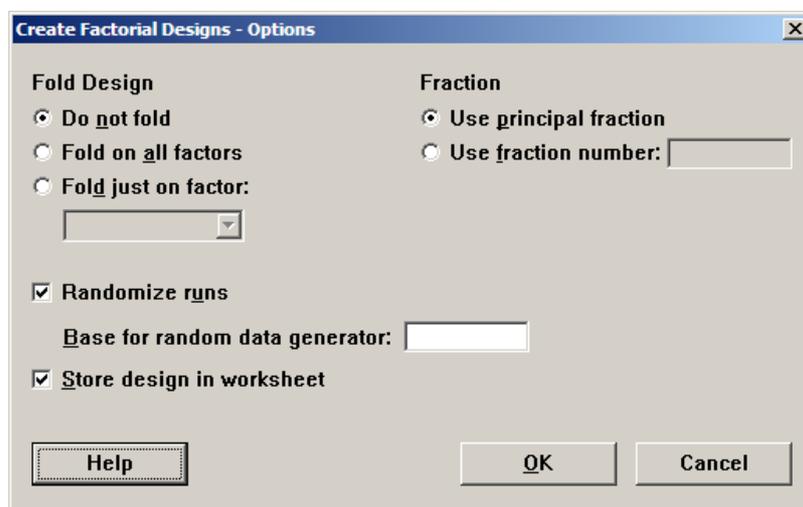


Figura 5.11: Ventana de diálogo del botón “Opciones”

La opción más razonable es plantear un orden aleatorio pero restringido entre aquellos órdenes conocidos que cumplen las propiedades deseadas. Esto es lo que hacen las macros que hemos desarrollado, utilizando los métodos descritos.

### 5.3.1 Implementación del método de las permutaciones y cambios de signo

#### 5.3.1.1 Procedimiento

En primer lugar hay que seleccionar el diseño deseado, con todas las opciones por defecto, excepto la de aleatorizar, que la quitaremos para que la matriz de diseño aparezca en orden estándar. Por ejemplo, si deseamos realizar un diseño  $2^{10-6}$  obtendremos la matriz de diseño tal como se presenta en la Figura 5.12.

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15
	StdOrder	RunOrder	CenterPt	Blocks	A	B	C	D	E	F	G	H	J	K	
1	1	1	1	1	-1	-1	-1	-1	-1	-1	-1	-1	1	1	
2	2	2	1	1	1	-1	-1	-1	1	-1	1	1	-1	-1	
3	3	3	1	1	-1	1	-1	-1	1	1	-1	1	-1	-1	
4	4	4	1	1	1	1	-1	-1	-1	1	1	-1	1	1	
5	5	5	1	1	-1	-1	1	-1	1	1	1	-1	-1	1	
6	6	6	1	1	1	-1	1	-1	-1	1	-1	1	1	-1	
7	7	7	1	1	-1	1	1	-1	-1	-1	1	1	1	-1	
8	8	8	1	1	1	1	1	-1	1	-1	-1	-1	-1	1	
9	9	9	1	1	-1	-1	-1	1	-1	1	1	1	-1	1	
10	10	10	1	1	1	-1	-1	1	1	1	-1	-1	1	-1	
11	11	11	1	1	-1	1	-1	1	1	-1	1	-1	1	-1	
12	12	12	1	1	1	1	-1	1	-1	-1	-1	1	-1	1	
13	13	13	1	1	-1	-1	1	1	1	-1	-1	1	1	1	
14	14	14	1	1	1	-1	1	1	-1	-1	1	-1	-1	-1	
15	15	15	1	1	-1	1	1	1	-1	1	-1	-1	-1	-1	
16	16	16	1	1	1	1	1	1	1	1	1	1	1	1	
17															

Figura 5.12: Hoja de datos con la matriz de diseño tal como la presenta Minitab, sin aleatorizar

Para obtener un orden con las propiedades deseadas basta con ejecutar la macro tecleando su nombre en la ventana de Sesión, tal como se indica en la Figura 5.13. A continuación se obtiene un orden con las propiedades deseadas tal como el de la Figura 5.14.

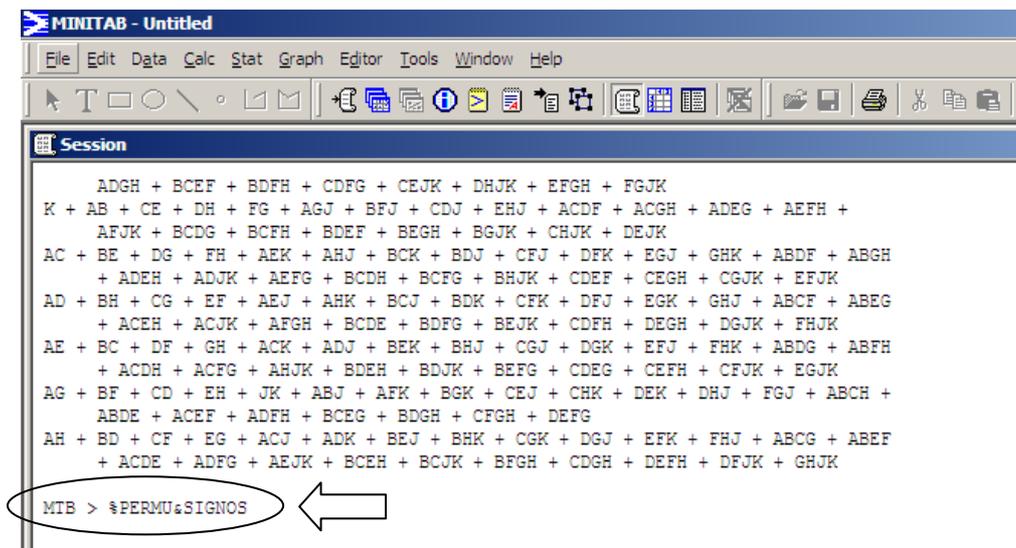


Figura 5.13: Lanzamiento de la macro para obtener un orden de experimentación con las propiedades deseadas

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15
	StdOrder	RunOrder	CenterPt	Blocks	A	B	C	D	E	F	G	H	J	K	
1	9	1	1	1	-1	-1	-1	1	-1	1	1	1	1	1	
2	10	2	1	1	1	-1	-1	1	1	1	-1	-1	-1	-1	
3	2	3	1	1	1	-1	-1	-1	1	-1	1	1	-1	-1	
4	6	4	1	1	1	-1	1	-1	-1	1	-1	1	-1	-1	
5	8	5	1	1	1	1	1	-1	1	-1	-1	-1	1	1	
6	7	6	1	1	-1	1	1	-1	-1	-1	1	1	-1	-1	
7	15	7	1	1	-1	1	1	1	-1	1	-1	-1	-1	-1	
8	11	8	1	1	-1	1	-1	1	1	-1	1	-1	-1	-1	
9	3	9	1	1	-1	1	-1	-1	1	1	-1	1	-1	-1	
10	4	10	1	1	1	1	-1	-1	-1	1	1	-1	1	1	
11	12	11	1	1	1	1	-1	1	-1	-1	-1	1	1	1	
12	16	12	1	1	1	1	1	1	1	1	1	1	1	1	
13	14	13	1	1	1	-1	1	1	-1	-1	1	-1	-1	-1	
14	13	14	1	1	-1	-1	1	1	1	-1	-1	1	1	1	
15	5	15	1	1	-1	-1	1	-1	1	1	1	-1	1	1	
16	1	16	1	1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	
17															

Figura 5.14: Orden de experimentación con las propiedades deseadas

### 5.3.1.2 Esquema de la macro

1. Determina el diseño de que se trata.

Para mayor comodidad, la macro se pone en marcha sin indicar de que diseño se trata. Por tanto, la propia macro, a través de la hoja de datos, debe identificar el número de experimentos, el número de factores y el número de columnas a ordenar.

2. Lee los órdenes semilla correspondientes al diseño identificado. Si se tiene más de un orden se elige uno al azar.

La selección al azar se realiza mediante la generación de un número aleatorio.

3. Reordena la matriz de diseño en el orden semilla

4. Permuta las columnas

Para ello se reordena al azar la secuencia de números desde 1 al  $k$  (siendo  $k$  el número de factores que corresponde a la matriz completa) y se colocan las columnas en el orden obtenido.

## 5. Cambia los signos

Se generan  $k$  valores de una distribución de Bernoulli con  $p=0,5$ . De forma ordenada, a cada columna le corresponde un valor (a la primera columna el primer valor, etc). Si el valor es igual a 1, la columna se multiplica por -1 y si es igual a cero se deja como está.

## 6. Introduce los generadores del diseño identificado, si este es un diseño factorial

Tras haber permutado las columnas de la matriz completa, y haber cambiado los signos a algunas de ellas, es necesario recalcular los signos de las columnas obtenidas a través de los generadores. Para realizar esta operación no hay más remedio que utilizar los generadores del diseño, por lo que se han debido introducir todos en la macro, siendo esta su parte más larga, aunque sería prescindible si se insertara en el código del propio programa.

## 7. Identifica el nuevo orden en el que quedan los experimentos

La identificación del orden resultante, referido al orden en la matriz estándar, no es estrictamente necesario para llevar a cabo la experimentación, ni para analizar los resultados, pero es conveniente, sobretodo en la fase de pruebas, para asegurarnos que no se ha cometido ningún error. El orden se identifica a través de unas expresiones polinómicas que dan un resultado para cada fila en función de la secuencia de signos que contenga.

En la Figura 5.15 se presenta el diagrama de flujo correspondiente a este método y en el Anexo 6 se encuentra el código completo de la macro.

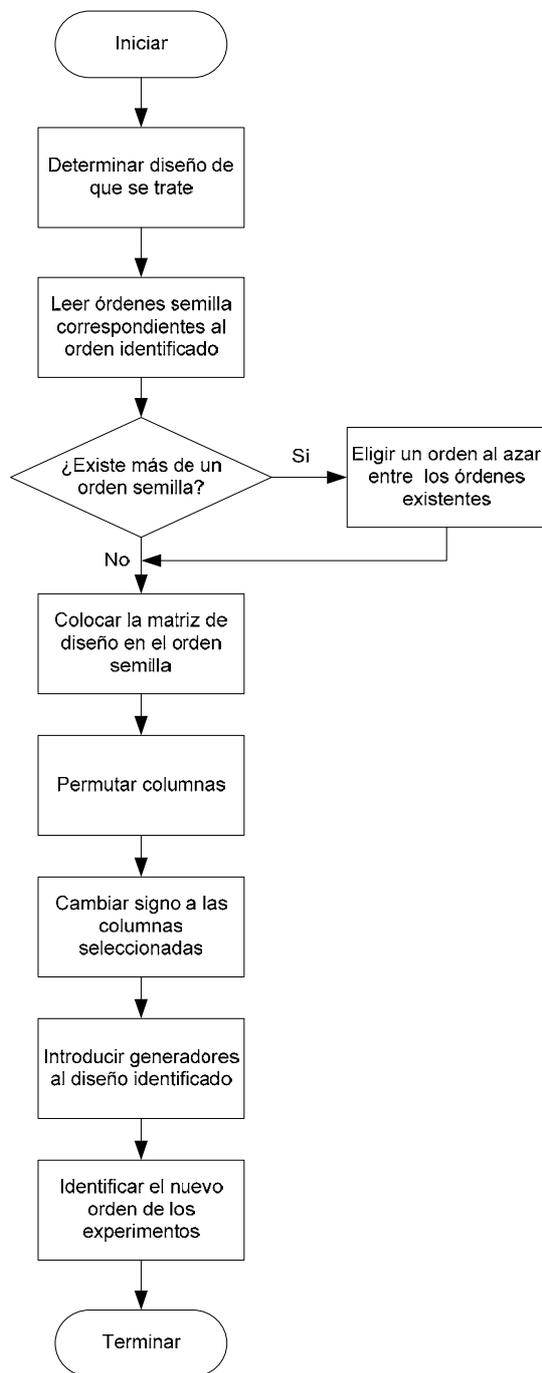


Figura 5.15: Diagrama de flujo del método de permutación y cambios de signo

### 5.3.2 Implementación del método de las columnas de expansión

El procedimiento de uso es idéntico al de la macro anterior. Se plantea el diseño deseado eliminando la opción de aleatorización para que aparezca la matriz de diseño en orden estándar, y a continuación se ejecuta la macro.

#### 5.3.2.1 Esquema de la macro

1. Determina el número de experimentos y el número de columnas a ordenar.

Igual que en la macro anterior, este paso evita que se tenga que introducir el tipo de diseño de que se trata.

2. Se elige un orden al azar de entre los 8 disponibles en un diseño  $2^2$ .

Se leen los 8 órdenes con el mínimo número de cambios de nivel en los diseños  $2^2$  (ver, Tabla 5.4) y se elige uno al azar. A partir de aquí, si se trata de un diseño con 4 experimentos, ya se puede ir al paso 5.

3. Se elige al azar una columna de expansión, de entre las disponibles, para llegar al diseño  $2^3$ .

Se leen 6 columnas de expansión (ver, Tabla 5.5) que permiten generar un orden para diseños factoriales a dos niveles con 8 experimentos que presentan el mínimo número de cambios de nivel y mínimo sesgo máximo a partir de orden elegido del diseño  $2^2$  en el paso 2. Se elige al azar una de las columnas y se genera el orden en que deben ser realizados los experimentos. A partir de aquí, si se trata de un diseño con 8 experimentos, ya se puede ir al paso 5.

4. Igual que el paso anterior hasta llegar al diseño deseado.

De acuerdo al número de experimentos ( $2^k$ , con  $k=4, \dots, 7$ ) determinado en el paso 1, Se leen las columnas de expansión disponibles para los diseños con igual número de experimentos y se selecciona aleatoriamente una de ellas. Estas columnas fueron obtenidas del orden de experimentación que presenta el mínimo número de cambios de nivel y el mínimo sesgo máximo en cada diseño (ver capítulo 4). El número de columnas de expansión utilizadas para construir los órdenes fueron: 8 columnas para el diseño  $2^4$ , 2 columnas para los diseños  $2^5$ ,  $2^6$  y  $2^7$ . Aunque para estos tres últimos diseños el total de columnas de expansión serían: 10, 12 y 14 respectivamente, pero el número de órdenes que pueden ser obtenidos con dos columnas de expansión es lo suficientemente grande para los fines de aplicación

práctica. A partir de la columna de expansión elegida se obtiene el orden en que deben ser realizados los experimentos en el diseño determinado y se va al paso 5.

5. Ordenar la matriz de diseño de acuerdo con el orden obtenido en el último paso.

Con base el orden obtenido el paso anterior se construye la matriz de diseño que presenta el mínimo número de cambios de nivel y el mínimo sesgo máximo en valor absoluto y se termina la búsqueda.

En la Figura 5.16 se presenta el diagrama de flujo correspondiente a este método y en el Anexo 7 se encuentra el código completo de esta macro.

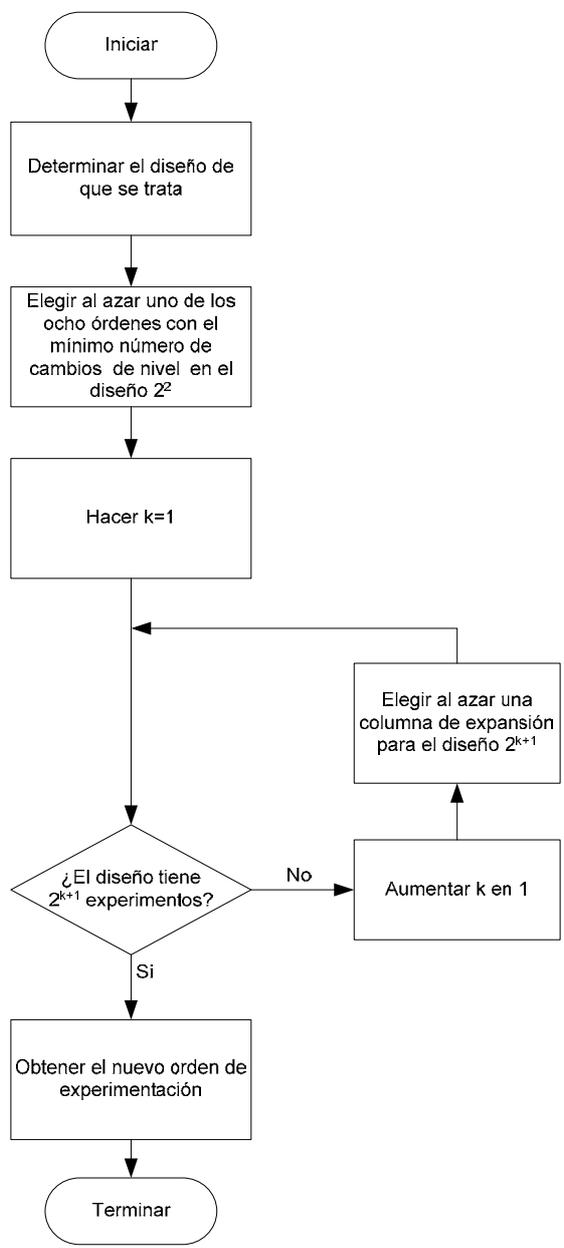


Figura 5.16: Diagrama de flujo del método de la columna de expansión

## **5.4 Aportaciones**

- Se ha planteado el método de las permutaciones y cambios de signo para obtener nuevos órdenes con las propiedades deseadas a partir de un orden semilla. Se ha concretado esta metodología en un algoritmo fácil de implementar en un lenguaje informático.
- Se ha desarrollado el método de las columnas de expansión para obtener los órdenes de un diseño a través de órdenes con las propiedades deseadas obtenidas para diseños con menos factores.
- Se han implementado los dos procedimientos en el programa de software estadístico Minitab. Ambos procedimientos son equivalentes y puede utilizarse uno u otro según convenga. Mediante estas macros se resuelve de forma práctica el problema de utilizar órdenes de experimentación que cumplan con las propiedades deseadas.

# Capítulo 6



# Capítulo 6

## Diseños con más de 128 experimentos

Para diseños con más de 128 experimentos, la búsqueda de los mejores órdenes utilizando las técnicas vistas en el capítulo 4 se convierte en una tarea imposible debido al aumento exponencial de las posibilidades que se presentan (en realidad, la búsqueda exhaustiva ya no es posible para diseños con más de 32 experimentos). En este caso, la única técnica viable es duplicar la matriz del diseño con un factor menor y con las propiedades deseadas.

Hemos visto en el capítulo 3 que duplicar una matriz de diseño con la metodología propuesta, duplica también el sesgo máximo, y por diversos procedimientos hemos visto que es posible disminuir ese sesgo duplicado en diseños con 32 y 64 experimentos. En la tabla Tabla 6.1 se indican los valores de los menores sesgos obtenidos, y los que se hubieran obtenido con el método de la duplicación.

*Tabla 6.1: Mínimos sesgos máximos que se deducen con el método de duplicación y finalmente hallados para cada número de experimentos*

Número de Experimentos	Mínimo sesgo máx. con el método de duplicación	Mínimo sesgo máx. finalmente hallado.
8	-	2
16	4	4
32	8	4,75
64	16	8
128	32	15,0625

En este capítulo se pondrá de manifiesto que duplicar el sesgo en diseños con más de 128 experimentos conduce a buenos resultados y que, por tanto, este método es una buena opción (en este momento la mejor posible) para diseños de esa magnitud.

## 6.1 Comparación de los sesgos máximos para diseños con distinto número de experimentos

Tal como puede observarse en la Tabla 6.1, el mínimo sesgo máximo va aumentando a medida que lo hace el número de experimentos, pero esto no quiere decir que la situación que sea cada vez peor. La Figura 6.1 representa los valores obtenidos para los efectos en diseños con 32, 64 y 128 experimentos ejecutados en un orden con las propiedades deseadas y con una respuesta que es igual al orden de ejecución.

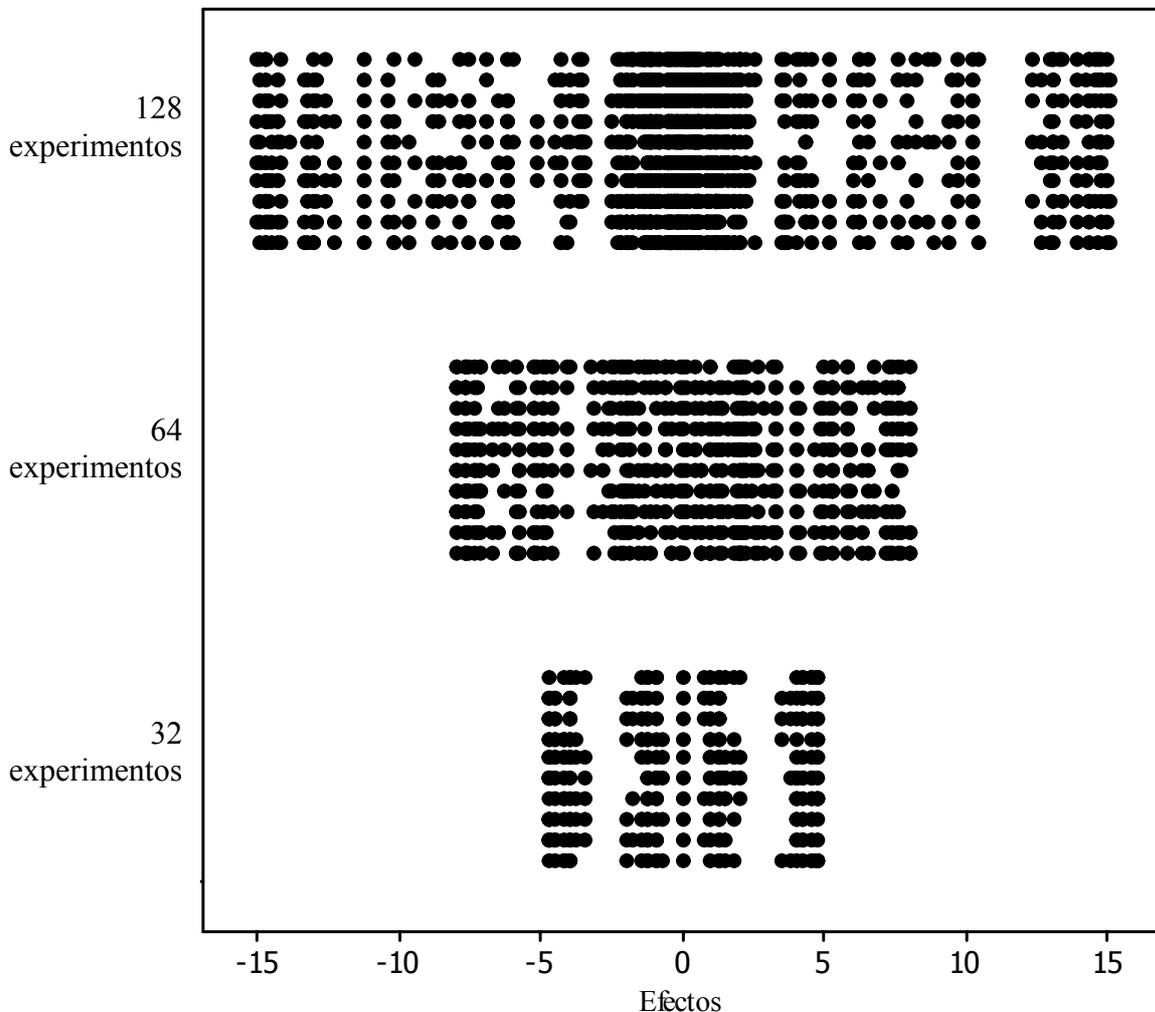


Figura 6.1: Efectos obtenidos en diseños con 32, 64 y 128 experimentos con las propiedades deseadas y en los que la respuesta es igual al orden de ejecución de los experimentos

Aunque el sesgo (valor de los efectos) aumenta a medida que lo hace el número de experimentos del diseño, esto no significa que el resultado sea peor. El sesgo puede servir para comparar la capacidad de eliminar tendencias de distintas ordenaciones para un mismo diseño, pero no para comparar diseños con distinto número de experimentos.

Para poder realizar esta comparación se ha definido una nueva medida de la capacidad de una ordenación para neutralizar la influencia de tendencias, y que puede servir para comparar diseños con distinto número de experimentos.

## 6.2 Nueva medida de la influencia de la tendencia: Coeficiente de correlación entre la respuesta y el efecto

El coeficiente de correlación entre dos variables aleatorias  $X$  e  $Y$  responde a la expresión:

$$\rho = \frac{\sum_{i=1}^N (x_i - \mu_X)(y_i - \mu_Y)}{N\sigma_X\sigma_Y}$$

Si  $Y$  es el vector que contiene las respuestas y  $X$  el correspondiente a un efecto en la matriz de diseño expandida (incluye las columnas correspondientes a las interacciones), tendremos que:

$$\begin{aligned} \sum_{i=1}^N (x_i - \mu_X)(y_i - \mu_Y) &= \sum_{i=1}^N (x_i y_i - x_i \mu_Y - \mu_X y_i + \mu_X \mu_Y) = \\ &= \sum_{i=1}^N x_i y_i - \mu_Y \sum_{i=1}^N x_i - \mu_X \sum_{i=1}^N y_i + \sum_{i=1}^N \mu_X \mu_Y \end{aligned}$$

Y como  $\sum_{i=1}^N x_i = 0$ , nos queda para el numerador solo la expresión  $\sum_{i=1}^N x_i y_i$

Por otra parte, tenemos que:

$$\sigma_X^2 = \frac{\sum_{i=1}^N x_i^2}{N} = \frac{N}{N} = 1$$

$$\begin{aligned}\sigma_Y^2 &= \frac{\sum_{i=1}^N (y_i - \mu_Y)^2}{N} = \\ &= \frac{\sum_{i=1}^N y_i^2}{N} - 2\mu_Y \frac{\sum_{i=1}^N y_i}{N} + \frac{\sum_{i=1}^N \mu_Y^2}{N} = \\ &= \frac{\sum_{i=1}^N y_i^2}{N} - \mu_Y^2\end{aligned}$$

Obsérvese que  $\sum_{i=1}^N y_i^2$  equivale en nuestro caso a la suma de los primeros  $N$  números naturales al cuadrado y, por tanto, responde a la expresión:

$$\sum_{i=1}^N y_i^2 = \frac{1}{3}N^3 + \frac{1}{2}N^2 + \frac{1}{6}N$$

Por otra parte,

$$\begin{aligned}\mu_Y^2 &= \left(\frac{N+1}{2}\right)^2 = \\ &= \frac{N^2 + 2N + 1}{4} = \\ &= \frac{N^2}{4} + \frac{N}{2} + \frac{1}{4}\end{aligned}$$

Luego tenemos que:

$$\begin{aligned}\sigma_Y^2 &= \frac{N^2}{3} + \frac{N}{2} + \frac{1}{6} - \frac{N^2}{4} - \frac{N}{2} - \frac{1}{4} = \\ &= \frac{N^2 - 1}{12}\end{aligned}$$

Por tanto, sustituyendo  $2^k = N$ , tendremos el siguiente coeficiente de correlación entre un efecto cuya columna correspondiente en la matriz de diseño ampliada sea el vector  $X$ , y el vector de respuestas  $Y$ :

$$\rho_{E-R} = \frac{\sum_{i=1}^N x_i y_i}{2^k \sqrt{\frac{2^{2k} - 1}{12}}}$$

Por tanto, nos queda una fórmula cerrada en función sólo de lo que hemos denominado “*Time Count*” y del número de experimentos de que consta el diseño.

### 6.3 Coeficiente de correlación máximo entre factores y respuesta para los diseños con las propiedades deseadas

De forma análoga al sesgo máximo, puede definirse el coeficiente de correlación máximo para un diseño y para este valor utilizaremos la notación  $\rho_{max}$ . La Tabla 6.2 muestra el coeficiente de correlación máximo  $\rho_{max}$  para los diseños que se han considerado, desde 8 hasta 128 experimentos, y también se muestra cual es el valor de este coeficiente para diseños con 216 y 512 experimentos.

Tabla 6.2: Coeficiente de correlación  $\rho_{E-R}$  para distintos valores del número de experimentos

Número de Experimentos	Sesgo máximo	<i>Time Count</i> máximo	Denominador del coeficiente $\rho_{max}$	Coeficiente $\rho_{max}$
8	2	8	18,33	0,4364
16	4	32	73,76	0,4338
32	4,75	76	295,46	0,2572
64	8	256	1182,27	0,2165
128	15,0625	964	4729,51	0,2038
256	30,125	3856	18918,47	0,2038
512	60,250	15424	75674,31	0,2038
...	...	...	...	...

Hemos visto que el sesgo máximo se multiplica por 2 al añadir un factor al diseño y, por tanto, el *Time Count* se multiplicará por 4. Siguiendo con la duplicación del sesgo, para diseños con más de 7 factores el *Time Count* máximo responde a la expresión:

$$Time\ Count = 964 \cdot 2^{2(k-7)}$$

Por tanto, el coeficiente de correlación máximo será:

$$\rho_{max} = \frac{964 \cdot 2^{2(k-7)}}{2^k \sqrt{\frac{2^{2k} - 1}{12}}}$$

Con  $k$  grande, podemos considerar  $2^{2k} - 1 = 2^{2k}$ , y nos queda:

$$\rho_{max} = \frac{964 \cdot 2^{2k} \sqrt{12}}{2^{2k} \cdot 2^{14}} =$$

$$= \frac{964 \sqrt{12}}{2^{14}} =$$

$$0,2038$$

Habíamos definido este coeficiente de correlación como una buena medida de la bondad de una ordenación para eliminar la influencia de variables que introducen una influencia en la respuesta, y acabamos de ver que este coeficiente se mantiene constante al aplicar el método de la duplicación del diseño visto en el capítulo 3.

Por tanto, para diseños con más de 128 experimentos podemos usar el método de duplicación presentado en el capítulo 3, sin temor a que una posible tendencia en la respuesta acabe afectando de forma considerable a la estimación de los efectos.

## 6.4 Aportaciones

En este capítulo se ha definido el coeficiente de correlación entre la respuesta y la columna de la matriz de diseño ampliada con la que presenta un coeficiente de correlación más alto, como una medida de la bondad de una ordenación para neutralizar la posible influencia de factores que introducen una tendencia en la respuesta.

Utilizando esta medida se ha demostrado que la capacidad de la ordenación para neutralizar las tendencias en la respuesta no se deteriora al aplicar la metodología propuesta en el capítulo 3. En este caso, el coeficiente de correlación se mantiene prácticamente constante e igual al obtenido al mejor de los casos de todos los diseños aparecidos anteriormente.

# Capítulo 7



# Capítulo 7

## Resultados obtenidos y futuras líneas de investigación

### 7.1 Resultados obtenidos

Los logros alcanzados a lo largo del desarrollo de este trabajo se pueden resumir en los siguientes aspectos:

- Se han presentado algunos argumentos que permiten sustentar la búsqueda de órdenes en diseños factoriales a dos niveles con ciertas características como el mínimo número de cambios de nivel y el mínimo sesgo máximo en valor absoluto. Lo cual permite hacer más económica la experimentación y a su vez proteger de la influencia de factores ajenos a la misma, en relación con los órdenes de experimentación construidos de forma completamente aleatoria.
- Se ha hecho una revisión exhaustiva de la literatura existente acerca de la eliminación o atenuación de la influencia lineal debida a factores desconocidos en los efectos principales e interacciones dobles. Adicionalmente, se ha revisado la documentación concerniente a la disminución o minimización del número de cambios de nivel en diseños factoriales a dos niveles. De lo cual podemos afirmar que el problema de la eliminación de la influencia lineal en los efectos principales e interacciones dobles ha sido resuelto de forma general para los diseños  $2^k$ . Así como también el problema de minimizar el número de cambios de nivel sin considerar la influencia de factores desconocidos. Pero hasta ahora el problema que

combina el mínimo número de cambios de nivel y la atenuación de la influencia lineal solo ha sido resuelto para diseños factoriales  $2^k$ , con  $k = 3, 4, 5$ . Por lo cual el objetivo principal de nuestro trabajo fue darle una solución general a este problema como un nuevo aporte al conocimiento dada su necesidad e importancia en la práctica industrial.

- Se ha construido un procedimiento de duplicación que permite obtener órdenes de experimentación en diseños factoriales a dos niveles que presentan el mínimo número de cambios de nivel que protegen contra la influencia lineal debida a factores desconocidos, los cuales son de gran importancia para la experimentación industrial dado el bajo coste de experimentación y la atenuación del sesgo que estos presentan. Con lo cual se resuelve de forma general el trabajo iniciado por Dickinson (1974) y complementado por De León et al. (2005). Adicionalmente, este trabajo es una mejora a los algoritmos construidos por Wang y Jan (1995) y Cheng et al. (1998) que buscan órdenes de experimentación diseños factoriales con el mínimo número de cambios de nivel de los factores sin considerar de forma simultánea la posible influencia de factores desconocidos.
- Con la construcción del procedimiento de duplicación se han podido identificar propiedades que permiten conservar las características deseadas de un diseño factorial  $2^k$  al expandirlo a un nuevo diseño factorial  $2^{k+1}$ .
- Usando diferentes métodos de búsqueda tales como: aleatoria restringida, el algoritmo de Dickinson, algoritmo de Dickinson con alimentación forzada y el método de duplicación, se hallaron órdenes con el mínimo número de cambios de nivel y mínimo sesgo máximo para los diseños factoriales a dos niveles con 64 y 128 experimentos. Adicionalmente, se han hallado los dos únicos órdenes semilla que producen diseños con el mínimo número de cambios en los factores y mínimo sesgo máximo para diseños con 32 experimentos. Hasta ahora estos órdenes solo se tenían para diseños con 8 y 16 experimentos (además de los de 4, que es un caso elemental).
- Con el método de duplicación construido, es posible escoger el número de cambios que se desean realizar en cada uno de los factores teniendo como base las restricciones del proceso o los costos que estos cambios implican en cada uno de los factores y a su vez la cantidad de sesgo que puede ser atenuada. Por ejemplo. Si partimos de un  $2^2$  el factor con menor número de cambios tendrá 1 cambio. Si partimos de  $2^3$  ó  $2^4$  el factor con menor número de cambios tendrá 2. Si partimos de un  $2^5$  el factor con menor número de cambio tendrá 5 y si partimos de un diseño  $2^6$  ó  $2^7$  el factor con menor número de cambio tendrá 9.

- Con el método de las columnas de expansión es posible generalizar la búsqueda de órdenes con el mínimo número de cambios de nivel que protegen de la influencia lineal debida a factores ajenos a la experimentación para cualquier diseño factorial  $2^k$  partiendo de ocho órdenes del diseño  $2^2$ . Con el procedimiento de duplicación o el método de columnas de expansión es posible obtener órdenes con el mínimo número de cambios de nivel que protegen de la influencia de factores ajenos a la experimentación para diseños factoriales  $2^k$  con  $k > 5$ , los cuales no se conocían hasta el desarrollo de esta tesis doctoral.
- En los órdenes de experimentación con el mínimo número de cambios de nivel obtenidos con el método de duplicación para diseños factoriales con 32 o más experimentos, se disminuye el valor del mínimo sesgo máximo en todos los efectos a más de una cuarta parte, comparado con los órdenes que presentan el mínimo número de cambios de nivel, obtenidos con los algoritmos propuestos por Wang y Jan (1995) y Cheng *et al.* (1998).
- Se han construido dos macros en Minitab que permiten generar automáticamente y de forma aleatoria, órdenes que presentan el mínimo número de cambios de nivel y que protegen de la influencia lineal de los factores ajenos a la experimentación, lo que permite la utilización de estos órdenes hallados por parte de usuarios no expertos.
- Se ha definido el coeficiente de correlación entre la respuesta y la columna de la matriz de diseño ampliada con la que presenta un coeficiente de correlación más alto, como una medida de la bondad de una ordenación para neutralizar la posible influencia de factores que introducen una tendencia en la respuesta.
- El análisis de los órdenes obtenidos con el método de duplicación puede ser realizado con los métodos tradicionales, tales como los mínimos cuadrados ordinarios (OLS) o la prueba de significación F, ya que estos permiten identificar efectos significativos en el modelo aunque los factores no sean reajustados antes de cada prueba [Webb *et al.* (2004)], lo que facilita su uso a nivel industrial por parte de usuarios no expertos, dado que no se requieren modelos complejos (modelo mixto) para llevarlo a cabo. Aunque si se desea hacer un análisis preciso es necesario utilizar el método de mínimos cuadrados generalizados (GLS), ya sea usando el modulo MIXED de SAS como lo propone Goos *et al.* (2006) o haciendo algunas manipulaciones con las herramientas de MINITAB como lo propone Simpson *et al.* (2004).

## 7.2 Futuras líneas de investigación

- Para diseños con 64 o más experimentos no podemos garantizar que los sesgos obtenidos sean los mínimos posibles. Aquí existe, por tanto, una posible línea de investigación para buscar métodos analíticos o algoritmos más eficientes que permitan encontrar esos valores mínimos o demostrar que ya lo son los aquí obtenidos.
- También para diseños de 64 o más experimentos, pueden existir más órdenes semilla de los que aquí se presentan. Se podría ampliar la búsqueda para encontrar nuevos órdenes.
- Esta tesis solo considera una influencia lineal del orden en el valor de la respuesta. Una posible línea de ampliación sería considerar influencias de tipo polinomial, aunque consideramos que en la práctica siempre será difícil realizar hipótesis sobre este tipo de influencia, más allá del modelo planteado con influencia lineal.
- Un tema que en esta tesis no se trata con detalle es el referido al análisis de los resultados obtenidos cuando existen restricciones en los cambios de nivel de algunos factores. ¿Hasta que punto es necesario utilizar técnicas de análisis más complejas que las habituales cuando se utilizan los planes de experimentación propuestos? ¿En qué situaciones, o bajo qué suposiciones, es razonable utilizar los métodos habituales, y en cuales no lo es?

Este último es sin duda un interesante tema a estudiar. No es nuevo y hay trabajos sobre el mismo, algunos muy actuales. En el anexo 8 incluimos un resumen de los trabajos más relevantes que hemos encontrado y que pueden servir de punto de partida para esta nueva investigación.

# **Anexos**



# Anexo 1

## Órdenes de experimentación para diseños con 8 y 16 experimentos propuestos por de León (2004)

De León (2004) obtuvo a través de una búsqueda exhaustiva de todas las posibles ordenaciones de un diseño factorial con 8 experimentos los únicos 48 órdenes que presentan mínimo sesgo y a la vez mínimo número de cambios en los niveles de los factores.

Asimismo, mediante un programa basado en el algoritmo de Dickinson encontró los únicos 16 órdenes semilla para diseños con 16 experimentos que presentan el mínimo número de cambios y el menor sesgo posible entre estos.

A continuación se reproducen las tablas con estos órdenes, que se citan en varias ocasiones a lo largo de la tesis.

*Tabla A.1.1: Únicos 48 órdenes para diseños  $2^{k-p}$  con 8 experimentos que presentan el valor mínimo para el sesgo máximo (SMA=2) y mínimo número de cambios en los niveles de los factores en la matriz completa. Las filas A, B y C indican el número de cambios para cada factor.*

	1	5	1	1	6	6	1	7	8	4	4	8	1	5	1	7	7	8	4	3	5	3	4	6
	2	7	2	5	2	8	5	8	7	2	8	7	3	6	3	8	3	6	3	7	7	7	2	2
	6	3	4	7	4	4	6	6	3	6	6	5	7	2	4	4	4	2	7	5	8	8	1	1
	8	4	8	8	3	3	8	2	1	5	5	1	8	4	8	2	2	1	5	6	4	6	5	3
	4	8	6	6	1	7	7	4	5	1	7	3	4	8	7	6	1	5	1	8	3	5	6	4
	3	6	5	2	5	5	3	3	6	3	3	4	2	7	5	5	5	7	2	4	1	1	8	8
	7	2	7	4	7	1	4	1	2	7	1	2	6	3	6	1	6	3	6	2	2	2	7	7
	5	1	3	3	8	2	2	5	4	8	2	6	5	1	2	3	8	4	8	1	6	4	3	5
A	2	2	2	2	2	2	3	2	2	2	2	2	2	2	3	2	3	2	2	2	3	3	3	3
B	2	2	3	3	3	2	2	3	2	2	3	3	2	2	2	2	2	2	2	3	2	2	2	2
C	3	3	2	2	2	3	2	2	3	3	2	2	3	3	2	3	2	3	3	2	2	2	2	2
Total	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7

(sigue...)

... continuación

	2	2	5	5	2	6	2	8	7	2	2	6	6	4	4	7	7	3	3	5	3	3	8	8
	1	1	1	6	6	5	4	4	5	4	6	5	8	3	8	5	3	1	1	1	4	4	6	4
	3	5	3	8	8	7	3	3	6	8	5	1	7	1	7	1	1	2	5	2	2	8	5	2
	7	7	4	4	7	3	7	1	2	7	7	3	3	5	5	2	2	6	6	4	6	6	1	1
	5	3	2	2	5	1	8	2	1	3	8	7	4	7	6	6	4	5	2	3	8	2	2	3
	6	4	6	1	1	2	6	6	3	1	4	8	2	8	2	8	8	7	4	7	7	1	4	7
	8	8	8	3	3	4	5	5	4	5	3	4	1	6	1	4	6	8	8	8	5	5	3	5
	4	6	7	7	4	8	1	7	8	6	1	2	5	2	3	3	5	4	7	6	1	7	7	6
A	2	2	2	2	2	2	3	3	3	2	3	2	3	2	3	2	2	3	2	3	2	2	3	2
B	3	2	3	3	3	3	2	2	2	2	2	2	2	3	2	2	3	2	2	2	3	2	2	3
C	2	3	2	2	2	2	2	2	2	3	2	3	2	2	2	3	2	2	3	2	2	3	2	2
Total	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7

Tabla A.1.2: Únicos órdenes semilla para diseños  $2^{k-p}$  con 16 experimentos que producen el mínimo número de cambios en los niveles de los factores en la matriz completa y un valor del sesgo máximo igual a 4 (el mínimo posible con el mínimo número de cambios). Las filas A, B, C y D indican el número de cambios para cada factor

	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
	4	4	4	4	4	4	4	4	4	4	4	10	10	10	10	10	10	10	10	10	10	10	10	10
	3	3	3	8	8	12	12	12	12	12	12	12	12	14	14	14	14	14	14	14	14	14	14	14
	11	11	11	16	16	11	11	11	11	16	4	4	16	16	16	16	16	16	16	16	16	16	16	16
	12	12	15	12	15	3	3	3	3	8	3	3	12	12	15	15	15	15	15	15	15	15	15	15
	16	16	7	11	7	7	7	7	7	7	7	7	7	4	11	7	11	11	11	11	11	11	11	11
	8	15	8	3	3	5	8	15	15	3	5	8	3	15	3	12	12	12	12	12	12	12	12	
	7	7	16	7	11	13	16	13	16	11	13	16	11	7	11	4	11	11	11	11	11	11	11	
	15	8	12	15	12	15	15	5	8	15	14	14	15	3	12	3	12	12	12	12	12	12	12	
	13	6	10	13	10	16	13	6	6	13	6	6	7	4	4	7	7	7	7	7	7	7	7	
	5	5	14	5	14	8	5	8	5	5	8	5	8	8	8	8	8	8	8	8	8	8	8	
	6	13	6	6	6	6	6	16	13	6	16	13	6	6	6	6	6	6	6	6	6	6	6	
	14	14	5	14	5	14	14	14	14	14	14	14	14	15	15	5	5	5	5	5	5	5	5	
	10	10	13	10	13	10	10	10	10	10	10	10	10	11	11	13	13	13	13	13	13	13	13	
	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	
A	6	8	4	4	4	4	6	4	6	4	4	4	4	4	4	4	4	4	4	4	4	4	6	
B	2	2	2	2	2	4	2	4	2	2	4	4	2	2	2	2	2	2	2	2	2	2	2	
C	2	2	4	4	4	2	2	2	2	4	2	2	4	6	4	4	4	4	4	4	4	4	4	
D	5	3	5	5	5	5	5	5	5	5	5	5	5	3	5	3	5	5	5	5	5	5	3	
Total	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	

## Anexo 2

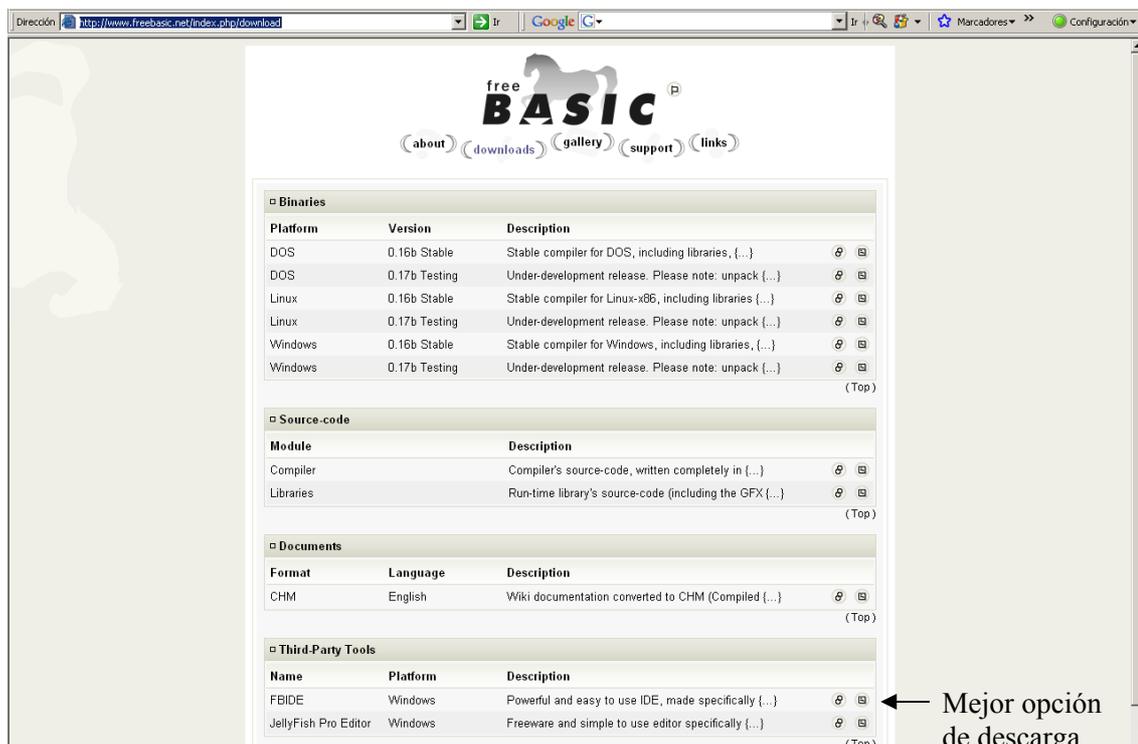
### Uso del lenguaje de programación *FreeBasic*

Los programas realizados en esta tesis han sido realizados utilizando el lenguaje de programación FreeBasic, excepto los que están dirigidos al experimentador para su uso en el contexto del software estadístico Minitab, que se han realizado mediante macros de este programa.

En principio la idea era realizar todos los programas en el lenguaje de la macros de Minitab, ya que de esta forma se evitaba el tener que programar todos los análisis estadísticos que Minitab ya realiza automáticamente (generación de los diseños, cálculo de los efectos,...) además de otras acciones, como la ordenación de valores, que en Minitab son triviales pero que programarlas exige un cierto esfuerzo. Sin embargo, la lentitud (comparada con los lenguajes de programación compilables) con que se ejecutan las macros, hacía totalmente inviable su uso para la consecución de en nuestros objetivos. Había que utilizar un lenguaje compilable, y el elegido fue FreeBasic.

Características destacables de este lenguaje, que nos hizo decidir su utilización son:

- Es un lenguaje de uso libre, que puede ser descargado a través de Internet en la dirección: <http://www.freebasic.net/index.php/download>



The screenshot shows the website <http://www.freebasic.net/index.php/download>. The page features the FreeBasic logo and navigation links: (about) (downloads) (gallery) (support) (links). The main content is organized into four sections:

- Binaries**: A table listing compilers for DOS, Linux, and Windows in both Stable and Testing versions.
- Source-code**: A table listing source code for the Compiler and Libraries.
- Documents**: A table listing CHM documentation in English.
- Third-Party Tools**: A table listing IDEs like FBIDE and JellyFish Pro Editor.

An arrow points to the 'FBIDE' entry in the 'Third-Party Tools' table, with the text 'Mejor opción de descarga' (Best download option).

Platform	Version	Description		
DOS	0.16b Stable	Stable compiler for DOS, including libraries, {...}	🔗	📄
DOS	0.17b Testing	Under-development release. Please note: unpack {...}	🔗	📄
Linux	0.16b Stable	Stable compiler for Linux-x86, including libraries {...}	🔗	📄
Linux	0.17b Testing	Under-development release. Please note: unpack {...}	🔗	📄
Windows	0.16b Stable	Stable compiler for Windows, including libraries, {...}	🔗	📄
Windows	0.17b Testing	Under-development release. Please note: unpack {...}	🔗	📄

Module	Description		
Compiler	Compiler's source-code, written completely in {...}	🔗	📄
Libraries	Run-time library's source-code (including the GFX {...})	🔗	📄

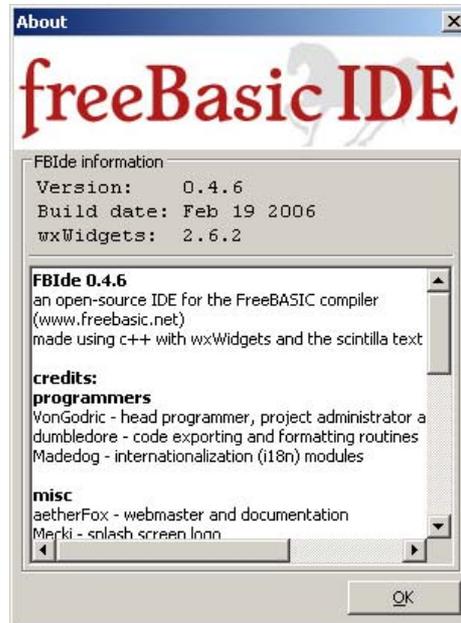
  

Format	Language	Description		
CHM	English	Wiki documentation converted to CHM (Compiled {...})	🔗	📄

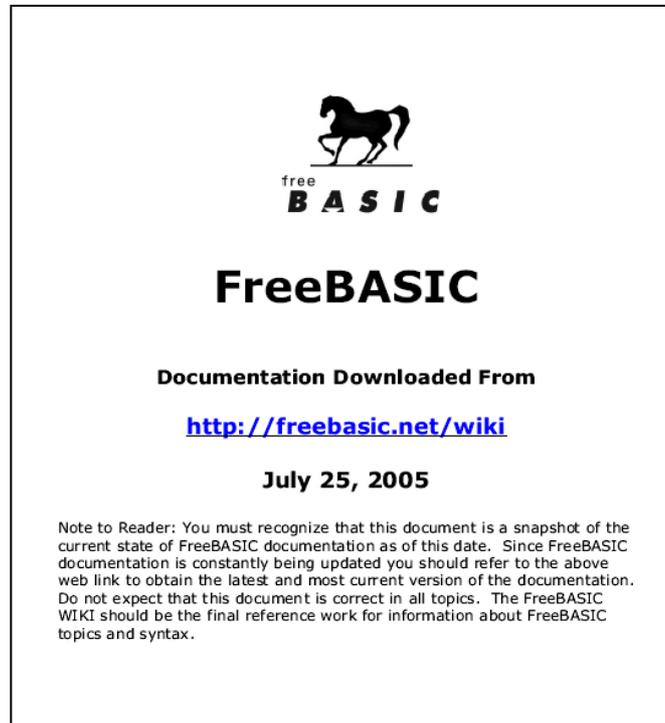
Name	Platform	Description		
FBIDE	Windows	Powerful and easy to use IDE, made specifically {...}	🔗	📄
JellyFish Pro Editor	Windows	Freeware and simple to use editor specifically {...}	🔗	📄

- Posee un interfaz de programación muy cómodo y completo, y también de libre distribución. El usado en esta tesis es el FBIDE 0.4.6 que se puede obtener en la misma dirección de Internet citada anteriormente.



```
FBide - G:\ACTUAL\Alexander\PROGRAMAS\Busqueda aleatoria\aleat64_230906.bas
File Edit Search View Run Help
[!] aleat64_230906.bas
38 DIM FILA(1 TO 64) AS INTEGER
39 DIM SUCESORES(1 TO 64, 1 TO 6) AS INTEGER
40 DIM POSIBLES(1 TO 6) AS INTEGER
41 DIM RESE(1 TO 64) AS INTEGER
42 DIM EFEC(1 TO 63) AS SINGLE
43 DIM EFMAX AS SINGLE
44 DIM MATSTD(1 TO 64, 1 TO 6) AS INTEGER
45 DIM MATORD(1 TO 64, 1 TO 6) AS INTEGER
46 DIM MATCOM(1 TO 64, 1 TO 63) AS INTEGER
47 DIM TABEFE(1 TO 9602, 1 TO 2) AS DOUBLE
48 DIM ORDBU(1 TO 1000, 1 TO 66) AS INTEGER
49 DIM E(1 TO 1000) AS SINGLE
50 DIM M, S, O, I, J, K, L, F AS INTEGER
51 DIM MM AS DOUBLE
52
53 RANDOMIZE TIMER
54
55 'Lee la matriz de diseño en orden estándar
56 FOR I=1 TO 6
57   FOR J=1 TO 64
58     READ MATSTD(J,I)
59   NEXT
60 NEXT
61
62 'Prepara la tabla en la que presentará los resultados
63
64 TABEFE(1,1)=99
65 FOR I=2 TO 9602
66   TABEFE(I,1)=8+(I-2)/400
67 NEXT
68
69 'Genera un número aleatorio entre 1 y Z
70 FUNCTION ALEAT (Z)
71   ALEAT = INT (RND(1)*Z + 1)
72 END FUNCTION
73
74 'Lee los sucesores posibles para cada fila
75 FOR I=1 TO 64
76   FOR J=1 TO 6
77     READ SUCESORES(I,J)
78   NEXT
79 NEXT
80
81 'Asigna los valores de las respuestas
82 FOR I=1 TO 64
83   RESP(I)=T
```

- Finalmente, una ventaja importante es que el tipo de instrucciones y la sintaxis utilizada por este lenguaje es muy parecida al lenguaje BASIC tradicional, y también a las macros de Minitab, por lo su aprendizaje resulta ser muy fácil. Además, existe un manual de instrucciones muy completo, claro y con ejemplos sobre el uso de cada instrucción.





## Anexo 3

### Programa en *FreeBasic* para generar órdenes con el mínimo número de cambios de nivel en un diseño $2^6$

Este programa genera órdenes de ejecución con el mínimo número de cambios en los niveles de los factores para diseños factoriales  $2^6$ . Para cada uno de los órdenes obtenidos se calcula el sesgo máximo en la estimación de los efectos y se guardan estos sesgos en forma de tabla. Si el sesgo obtenido está por debajo de un cierto valor (en nuestro caso hemos elegido este valor igual a 9) se guarda también el orden que genera dicho sesgo.

```
.....
' aleat64_230906.bas
'
' ¿QUÉ HACE?
' Genera órdenes con el mínimo número de cambios de nivel en un diseño 2^6.
' La generación se realiza basándose en la idea de que cada fila tiene solo
' un cierto número de sucesores posibles para asegurar un solo cambio de
' nivel entre una fila y la siguiente.
' Almacena en una tabla (TablaSesgos64_x.txt) el sesgo máximo para cada orden
' y guarda los órdenes con sesgo máximo < 9 (Ordenes64_x.txt)
'
' ¿CÓMO UTILIZARLO?
' Ejecutarlo a través de una interfaz como FBide 0.4.6 (www.freebasic.net)
' Los archivos de salida se generan en formato ASCII en el directorio FreeBasic
'
.....
'
' FILA:      Vector que contiene los números de fila
' SUCEORES: Matriz que contiene todos los posibles sucesores para cada fila
' POSIBLES: Posibles sucesores para una determinada fila
' ALEAT:     Función que genera números aleatorios de una distribución
'            equiprobable entre 1 y 64
' RESP:      Vector que contiene las respuestas (1, 2,... 64)
' EFEC:      Vector que contiene los valores de cada uno de los efectos
' EFMAX:     Efecto máximo
' MATSTD:    Matriz de diseño en el orden estándar
' MATORD:    Matriz de diseño en el orden hallado
' MATCOM:    Matriz completa o del modelo
' TABEFE:    Tabla de efectos
' ORDBU:     Orden bueno o con efecto bajo (por debajo del valor asignado)
' El resto de variables son contadores, o variables indicadoras
'
'
DIM FILA(1 TO 64) AS INTEGER
DIM SUCEORES(1 TO 64, 1 TO 6) AS INTEGER
DIM POSIBLES(1 TO 6) AS INTEGER
DIM RESP(1 TO 64) AS INTEGER
DIM EFEC(1 TO 63) AS SINGLE
DIM EFMAX AS SINGLE
DIM MATSTD(1 TO 64, 1 TO 6) AS INTEGER
DIM MATORD(1 TO 64, 1 TO 6) AS INTEGER
DIM MATCOM(1 TO 64, 1 TO 63) AS INTEGER
DIM TABEFE(1 TO 9602, 1 TO 2) AS DOUBLE
DIM ORDBU(1 TO 1000, 1 TO 66) AS INTEGER
DIM E(1 TO 1000) AS SINGLE
```

Anexo 3: Programa para generar órdenes con el mínimo número de cambios

```
DIM M, S, O, I, J, K, L, F AS INTEGER
DIM MM AS DOUBLE
'
RANDOMIZE TIMER
'
'Lee la matriz de diseño en orden estándar
FOR I=1 TO 6
  FOR J=1 TO 64
    READ MATSTD(J,I)
  NEXT
NEXT
'Prepara la tabla en la que presentará los resultados
'
TABEFE(1,1)=99
FOR I=2 TO 9602
  TABEFE(I,1)=8+(I-2)/400
NEXT
'
'Genera un número aleatorio entre 1 y Z
FUNCTION ALEAT (Z)
  ALEAT = INT (RND(1)*Z + 1)
END FUNCTION
'
'Lee los sucesores posibles para cada fila
FOR I=1 TO 64
  FOR J=1 TO 6
    READ SUCESTORES(I,J)
  NEXT
NEXT
'
'Asigna los valores de las respuestas
FOR I=1 TO 64
  RESP(I)=I
NEXT
'
'Inicializa los valores de E(I)
FOR I=1 TO 1000
  E(I)=0
NEXT
'
M=0
O=0
'
'Hasta el número de órdenes que queremos encontrar
WHILE M<2000000
  '
  'Primera fila
  FOR F=1 TO 64
    IF F=1 THEN
      FILA(F)=ALEAT(64)
    ELSE
      '
      'Resto de filas
      S=6
      '
      'Carga todos los posibles sucesores de la fila
      FOR J=1 TO 6
        POSIBLES(J)=SUCESTORES(FILA(F-1),J)
      NEXT
      '
      'Pone cero en los posibles sucesores ya utilizados
      FOR L=1 TO F-1
        IF POSIBLES(J)=FILA(L) THEN
          POSIBLES(J)=0
          S=S-1
        END IF
      NEXT
      NEXT
      '
      'Si una fila no tiene sucesores, sale
      IF S=0 THEN
        GOTO 20
      END IF
    '
  '

```



Anexo 3: Programa para generar órdenes con el mínimo número de cambios

```
DATA 2,9,12,14,26,42 '10
DATA 3,9,12,15,27,43 '11
DATA 4,10,11,16,28,44 '12
DATA 5,9,14,15,29,45 '13
DATA 6,10,13,16,30,46 '14
DATA 7,11,13,16,31,47 '15
DATA 8,12,14,15,32,48 '16
DATA 1,18,19,21,25,49 '17
DATA 2,17,20,22,26,50 '18
DATA 3,17,20,23,27,51 '19
DATA 4,18,19,24,28,52 '20
DATA 5,17,22,23,29,53 '21
DATA 6,18,21,24,30,54 '22
DATA 7,19,21,24,31,55 '23
DATA 8,20,22,23,32,56 '24
DATA 9,17,26,27,29,57 '25
DATA 10,18,25,28,30,58 '26
DATA 11,19,25,28,31,59 '27
DATA 12,20,26,27,32,60 '28
DATA 13,21,25,30,31,61 '29
DATA 14,22,26,29,32,62 '30
DATA 15,23,27,29,32,63 '31
DATA 16,24,28,30,31,64 '32
DATA 1,34,35,37,41,49 '33
DATA 2,33,36,38,42,50 '34
DATA 3,33,36,39,43,51 '35
DATA 4,34,35,40,44,52 '36
DATA 5,33,38,39,45,53 '37
DATA 6,34,37,40,46,54 '38
DATA 7,35,37,40,47,55 '39
DATA 8,36,38,39,48,56 '40
DATA 9,33,42,43,45,57 '41
DATA 10,34,41,44,46,58 '42
DATA 11,35,41,44,47,59 '43
DATA 12,36,42,43,48,60 '44
DATA 13,37,41,46,47,61 '45
DATA 14,38,42,45,48,62 '46
DATA 15,39,43,45,48,63 '47
DATA 16,40,44,46,47,64 '48
DATA 17,33,50,51,53,57 '49
DATA 18,34,49,52,54,58 '50
DATA 19,35,49,52,55,59 '51
DATA 20,36,50,51,56,60 '52
DATA 21,37,49,54,55,61 '53
DATA 22,38,50,53,56,62 '54
DATA 23,39,51,53,56,63 '55
DATA 24,40,52,54,55,64 '56
DATA 25,41,49,58,59,61 '57
DATA 26,42,50,57,60,62 '58
DATA 27,43,51,57,60,63 '59
DATA 28,44,52,58,59,64 '60
DATA 29,45,53,57,62,63 '61
DATA 30,46,54,58,61,64 '62
DATA 31,47,55,59,61,64 '63
DATA 32,48,56,60,62,63 '64
'
'
SESGO:
'Construye la matriz ordenada
FOR I=1 TO 64
  FOR J=1 TO 6
    MATORD(I,J)=MATSTD(FILA(I), J)
  NEXT
NEXT
'
'Construye la matriz completa
FOR I=1 TO 64
  MATCOM(I,1)=MATORD(I,1)
  MATCOM(I,2)=MATORD(I,2)
  MATCOM(I,3)=MATORD(I,3)
  MATCOM(I,4)=MATORD(I,4)
  MATCOM(I,5)=MATORD(I,5)
  MATCOM(I,6)=MATORD(I,6)
  MATCOM(I,7)=MATORD(I,1)*MATORD(I,2)
  MATCOM(I,8)=MATORD(I,1)*MATORD(I,3)
  MATCOM(I,9)=MATORD(I,1)*MATORD(I,4)
```

Anexo 3: Programa para generar órdenes con el mínimo número de cambios

```
MATCOM(I,10)=MATORD(I,1)*MATORD(I,5)
MATCOM(I,11)=MATORD(I,1)*MATORD(I,6)
MATCOM(I,12)=MATORD(I,2)*MATORD(I,3)
MATCOM(I,13)=MATORD(I,2)*MATORD(I,4)
MATCOM(I,14)=MATORD(I,2)*MATORD(I,5)
MATCOM(I,15)=MATORD(I,2)*MATORD(I,6)
MATCOM(I,16)=MATORD(I,3)*MATORD(I,4)
MATCOM(I,17)=MATORD(I,3)*MATORD(I,5)
MATCOM(I,18)=MATORD(I,3)*MATORD(I,6)
MATCOM(I,19)=MATORD(I,4)*MATORD(I,5)
MATCOM(I,20)=MATORD(I,4)*MATORD(I,6)
MATCOM(I,21)=MATORD(I,5)*MATORD(I,6)
MATCOM(I,22)=MATORD(I,1)*MATORD(I,2)*MATORD(I,3)
MATCOM(I,23)=MATORD(I,1)*MATORD(I,2)*MATORD(I,4)
MATCOM(I,24)=MATORD(I,1)*MATORD(I,2)*MATORD(I,5)
MATCOM(I,25)=MATORD(I,1)*MATORD(I,2)*MATORD(I,6)
MATCOM(I,26)=MATORD(I,1)*MATORD(I,3)*MATORD(I,4)
MATCOM(I,27)=MATORD(I,1)*MATORD(I,3)*MATORD(I,5)
MATCOM(I,28)=MATORD(I,1)*MATORD(I,3)*MATORD(I,6)
MATCOM(I,29)=MATORD(I,1)*MATORD(I,4)*MATORD(I,5)
MATCOM(I,30)=MATORD(I,1)*MATORD(I,4)*MATORD(I,6)
MATCOM(I,31)=MATORD(I,1)*MATORD(I,5)*MATORD(I,6)
MATCOM(I,32)=MATORD(I,2)*MATORD(I,3)*MATORD(I,4)
MATCOM(I,33)=MATORD(I,2)*MATORD(I,3)*MATORD(I,5)
MATCOM(I,34)=MATORD(I,2)*MATORD(I,3)*MATORD(I,6)
MATCOM(I,35)=MATORD(I,2)*MATORD(I,4)*MATORD(I,5)
MATCOM(I,36)=MATORD(I,2)*MATORD(I,4)*MATORD(I,6)
MATCOM(I,37)=MATORD(I,2)*MATORD(I,5)*MATORD(I,6)
MATCOM(I,38)=MATORD(I,3)*MATORD(I,4)*MATORD(I,5)
MATCOM(I,39)=MATORD(I,3)*MATORD(I,4)*MATORD(I,6)
MATCOM(I,40)=MATORD(I,3)*MATORD(I,5)*MATORD(I,6)
MATCOM(I,41)=MATORD(I,4)*MATORD(I,5)*MATORD(I,6)
MATCOM(I,42)=MATORD(I,1)*MATORD(I,2)*MATORD(I,3)*MATORD(I,4)
MATCOM(I,43)=MATORD(I,1)*MATORD(I,2)*MATORD(I,3)*MATORD(I,5)
MATCOM(I,44)=MATORD(I,1)*MATORD(I,2)*MATORD(I,3)*MATORD(I,6)
MATCOM(I,45)=MATORD(I,1)*MATORD(I,2)*MATORD(I,4)*MATORD(I,5)
MATCOM(I,46)=MATORD(I,1)*MATORD(I,2)*MATORD(I,4)*MATORD(I,6)
MATCOM(I,47)=MATORD(I,1)*MATORD(I,2)*MATORD(I,5)*MATORD(I,6)
MATCOM(I,48)=MATORD(I,1)*MATORD(I,3)*MATORD(I,4)*MATORD(I,5)
MATCOM(I,49)=MATORD(I,1)*MATORD(I,3)*MATORD(I,4)*MATORD(I,6)
MATCOM(I,50)=MATORD(I,1)*MATORD(I,3)*MATORD(I,5)*MATORD(I,6)
MATCOM(I,51)=MATORD(I,1)*MATORD(I,4)*MATORD(I,5)*MATORD(I,6)
MATCOM(I,52)=MATORD(I,2)*MATORD(I,3)*MATORD(I,4)*MATORD(I,5)
MATCOM(I,53)=MATORD(I,2)*MATORD(I,3)*MATORD(I,4)*MATORD(I,6)
MATCOM(I,54)=MATORD(I,2)*MATORD(I,3)*MATORD(I,5)*MATORD(I,6)
MATCOM(I,55)=MATORD(I,2)*MATORD(I,4)*MATORD(I,5)*MATORD(I,6)
MATCOM(I,56)=MATORD(I,3)*MATORD(I,4)*MATORD(I,5)*MATORD(I,6)
MATCOM(I,57)=MATORD(I,1)*MATORD(I,2)*MATORD(I,3)*MATORD(I,4)*MATORD(I,5)
MATCOM(I,58)=MATORD(I,1)*MATORD(I,2)*MATORD(I,3)*MATORD(I,4)*MATORD(I,6)
MATCOM(I,59)=MATORD(I,1)*MATORD(I,2)*MATORD(I,3)*MATORD(I,5)*MATORD(I,6)
MATCOM(I,60)=MATORD(I,1)*MATORD(I,2)*MATORD(I,4)*MATORD(I,5)*MATORD(I,6)
MATCOM(I,61)=MATORD(I,1)*MATORD(I,3)*MATORD(I,4)*MATORD(I,5)*MATORD(I,6)
MATCOM(I,62)=MATORD(I,2)*MATORD(I,3)*MATORD(I,4)*MATORD(I,5)*MATORD(I,6)

MATCOM(I,63)=MATORD(I,1)*MATORD(I,2)*MATORD(I,3)*MATORD(I,4)*MATORD(I,5)*MATORD(I,6)
NEXT
'
' Calcula los efectos
FOR I=1 TO 63
  EFEC(I)=0
NEXT
FOR J=1 TO 63
  FOR I=1 TO 64
    EFEC(J)=EFEC(J)+MATCOM(I,J)*RESP(I)
  NEXT
NEXT
'
' Busca el efecto máximo
EFMAX=ABS(EFEC(1))
FOR I=2 TO 63
  IF ABS(EFEC(I))>EFMAX THEN
    EFMAX=ABS(EFEC(I))
  END IF
NEXT
EFMAX=EFMAX/32
RETURN
```

*Anexo 3: Programa para generar órdenes con el mínimo número de cambios*

```
'  
' Guarda las ordenaciones que dan sesgo bajo  
GORDEN:  
OPEN "Ordenes64_4.txt" FOR OUTPUT AS #2  
FOR I=1 TO 0  
  FOR J=1 TO 64  
    WRITE #2, ORDBU(I,J),  
  NEXT  
  WRITE #2, E(I)  
NEXT  
CLOSE #2  
RETURN  
'  
' Actualizar la tabla de sesgos  
ACTUALIZA:  
IF EFMAX<8 THEN  
  TABEFE(1,2)=TABEFE(1,2)+1  
ELSE  
  I=-3198+400*EFMAX  
  TABEFE(I,2)=TABEFE(I,2)+1  
END IF  
RETURN  
'  
' Graba la tabla de sesgos  
GRABA:  
OPEN "TablaSesgos64_4.txt" FOR OUTPUT AS #1  
FOR I=1 TO 9602  
  WRITE #1, TABEFE(I,1),TABEFE(I,2)  
NEXT  
WRITE #1, M  
CLOSE #1  
RANDOMIZE TIMER  
RETURN
```

## Anexo 4

### Programa en *FreeBasic* para la búsqueda exhaustiva de órdenes semilla con mínimo número de cambios en diseños $2^5$ utilizando el algoritmo de Dickinson

La generación aleatoria de órdenes con el mínimo número de cambios, tal como se realiza con el programa del Anexo 3, puede proporcionar órdenes con sesgo bajo, pero no se tiene la seguridad de que no exista otro orden que proporcione un sesgo todavía menor.

El programa que aquí se presenta para diseños  $2^5$  (pero fácilmente convertible a otros tipos de diseño), realiza una búsqueda exhaustiva de todas las ordenaciones con el mínimo número de cambios, y para cada una de ellas calcula su sesgo máximo, por lo que es posible saber cual de las ordenaciones con mínimo número de cambios presenta mínimo sesgo máximo. Su ejecución tarda unas 5 horas en completarse.

```
.....
'DICKINSON32_171006.bas
'
' ¿QUÉ HACE?
' Genera, de una forma sistemática y ordenada, los órdenes semilla de un
' diseño  $2^5$  que permiten obtener TODOS los órdenes que presentan el mínimo
' número de cambios en los niveles de los factores. Se utiliza el algoritmo
' de Dickinson.
' Guarda e imprime en un archivo los órdenes que presentan un sesgo máximo <5
' y una tabla con el sesgo de todos los órdenes obtenidos (Salida32.txt)
'
' ¿CÓMO UTILIZARLO?
' Ejecutarlo a través de una interfaz como FBIDE 0.4.6 (www.freebasic.net)
' Los archivos de salida se generan en formato ASCII en el directorio FreeBasic
'
'
' VP: Posible fila sucesora o tratamiento a ser evaluado
' AP(I): Indica cuantas opciones se han probado
' AX(I): Número de posibles tratamientos sucesores
' CAM: Número de cambios en el orden obtenido
' NC: Número de ordenes hallados
' NF: Número de fila en la que se busca el tratamiento
' IND: Indicador de la posición en el vector SEC
' SEC: Vector de secuencia de posibles tratamientos sucesores
' para cada tratamiento de la matriz de diseño
' CORR: Tratamiento o prueba experimental elegida en una fila
' RESP: Vector de respuesta con tendencia lineal
' EFMAX: Efecto o sesgo máximo
' EFEC: Efecto o sesgo
' MATSTD: Matriz de diseño en el orden estándar
' MATORD: Matriz de diseño del orden hallado
' MATCOM: Matriz completa del orden hallado
' TABEFE: Tabla de efectos
' ORDBU: Orden bueno o con efecto máximo mínimo
' C: Define que cada cierto número de órdenes se debe
```

Anexo 4: Programa para la búsqueda exhaustiva de órdenes semilla

```

'          imprimir un archivo de salida
'
DIM NC, NF, VP, IND, NF1, J, I, K AS INTEGER
DIM EFMAX AS SINGLE
DIM SEC(1 TO 146) AS INTEGER
DIM AP(1 TO 31) AS INTEGER
DIM AX(1 TO 31) AS INTEGER
DIM ACUM(1 TO 32) AS INTEGER
DIM CORR(1 TO 32) AS INTEGER
DIM RESP(1 TO 32) AS INTEGER
DIM EFEC(1 TO 32) AS SINGLE
DIM MATSTD(1 TO 32, 1 TO 5)
DIM MATORD(1 TO 32, 1 TO 5)
DIM MATCOM(1 TO 32, 1 TO 31) AS INTEGER
DIM TABEFE(1 TO 50, 1 TO 2) AS SINGLE
'
'Se inicializan las variables
'
NC=0      ' Número de órdenes obtenidos
CORR(1)=1 ' Se fija el tratamiento 1 en la fila 1, para evitar cambios
          ' de signo en las columnas del diseño
CORR(2)=2 ' Se fija el tratamiento 2 en la fila 2 para garantizar
          ' un sólo cambio de nivel en las primeras dos filas
NF=3      ' Ubica el puntero en la fila 3 para iniciar la búsqueda
VP=2      ' Se define como primer tratamiento probable,el tratamiento
          ' ubicado en la segunda fila
'
' Se presentan los datos que componen los vectores o matrices
'
' Vector SEC que contiene la secuencia de valores de los posibles
' tratamientos sucesores con un cambio de nivel para cada una de
' las filas de la matriz de diseño del orden estándar
' para 32 experimentos
'
DATA 4,6,10,18      '2
DATA 4,7,11,19     '3
DATA 3,8,12,20     '4
DATA 6,7,13,21     '5
DATA 5,8,14,22     '6
DATA 3,5,8,15,23   '7
DATA 4,6,7,16,24   '8
DATA 10,11,13,25   '9
DATA 9,12,14,26    '10
DATA 3,9,12,15,27  '11
DATA 4,10,11,16,28 '12
DATA 5,9,14,15,29  '13
DATA 6,10,13,16,30 '14
DATA 7,11,13,16,31 '15
DATA 8,12,14,15,32 '16
DATA 18,19,21,25   '17
DATA 17,20,22,26   '18
DATA 3,17,20,23,27 '19
DATA 4,18,19,24,28 '20
DATA 5,17,22,23,29 '21
DATA 6,18,21,24,30 '22
DATA 7,19,21,24,31 '23
DATA 8,20,22,23,32 '24
DATA 9,17,26,27,29 '25
DATA 10,18,25,28,30 '26
DATA 11,19,25,28,31 '27
DATA 12,20,26,27,32 '28
DATA 13,21,25,30,31 '29
DATA 14,22,26,29,32 '30
DATA 15,23,27,29,32 '31
DATA 16,24,28,30,31 '32
'
' Vector AX que contiene el número de posibles tratamientos
' sucesores para cada fila del diseño con el orden estándar
'
DATA 4,4,4,4,4,5,5,4,4,5,5,5,5,5,5,4
DATA 4,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5
'
' Vector ACUM que contiene el número acumulado de posibles
' tratamientos sucesores
'
DATA 0,4,8,12,16,20,25,30,34,38,43,48,53,58,63,68,72,76,81,86,91

```

Anexo 4: Programa para la búsqueda exhaustiva de órdenes semilla

```
DATA 96,101,106,111,116,121,126,131,136,141,146
'
' MATSTD que contiene los signos de las columnas de la matriz
' de diseño con el orden estándar para un diseño factorial
' a dos niveles con 32 experimentos
'
DATA -1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1
DATA -1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1
DATA -1,-1,1,1,-1,-1,1,1,-1,-1,1,1,-1,-1,1,1
DATA -1,-1,1,1,-1,-1,1,1,-1,-1,1,1,-1,-1,1,1
DATA -1,-1,-1,-1,1,1,1,1,-1,-1,-1,-1,1,1,1,1
DATA -1,-1,-1,-1,1,1,1,1,-1,-1,-1,-1,1,1,1,1
DATA -1,-1,-1,-1,-1,-1,-1,-1,-1,1,1,1,1,1,1,1
DATA -1,-1,-1,-1,-1,-1,-1,-1,-1,1,1,1,1,1,1,1
DATA -1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1
DATA 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
'
' Se abre el archivo de salida donde se almacenan los resultados
'
OPEN "SALIDA32.txt" FOR OUTPUT AS #1
'
' Se leen los datos de los posibles tratamientos sucesores
' que presentan un sólo cambio de nivel del vector SEC
'
FOR I=1 TO 146
  READ SEC(I)
NEXT I
'
' Se lee el número de posibles tratamientos sucesores que tienen
' cada uno de los tratamientos del orden estándar del vector AX
' y se inicializan los apuntadores en cero
'
FOR I=1 TO 31
  AP(I)=0
  READ AX(I)
NEXT I
'
' Se lee el número acumulado de tratamientos sucesores
' del vector ACUM
'
FOR I=1 TO 32
  READ ACUM(I)
NEXT I
'
' Se leen los 1 y -1's que componen la matriz de diseño estándar
' para 32 experimentos
'
FOR I=1 TO 5
  FOR J=1 TO 32
    READ MATSTD(J,I)
  NEXT
NEXT
'
' Se genera el vector de respuestas con tendencia lineal (1,2...,32)
'
FOR I=1 TO 32
  RESP(I)=I
NEXT
'
' Se imprime la matriz de diseño en el orden estándar para un diseño
' factorial a dos niveles con 32 experimentos
' (sólo una vez para probar)
'
'FOR I=1 TO 32
'  FOR J=1 TO 5
'    IF J<5 THEN
'      PRINT MATSTD(I,J);
'    END IF
'    IF J=5 THEN
'      PRINT MATSTD(I,J)
'    END IF
'  NEXT
'NEXT
'
' Se inicia el proceso de búsqueda
'
```

Anexo 4: Programa para la búsqueda exhaustiva de órdenes semilla

```
WHILE NF>2 ' Si el número de fila es 2, se han explorado todos
            ' los órdenes semilla con el mínimo número
            ' de cambios de nivel para 32 experimentos
            ,
            'Se Inicializan los punteros
            ,
            K = AP(VP-1) ' Al principio K=0, no se ha probado ninguna opción
            J = AX(VP-1) ' Número de sucesores posibles en esta posición
            IF K >= J THEN
                GOTO 40 ' Ya se han mirado todos los sucesores de esa fila
            END IF
            ,
            ' Se selecciona una posible fila sucesora
            ,
            AP(VP-1) = AP(VP-1)+1          ' Va a probar una opción y suma 1
                                           ' al contador del apuntador
            IND = ACUM(VP-1) + AP(VP-1)    ' Indica la posición donde está la
                                           ' secuencia a probar en el vector SEC
            VP = SEC(IND)                  ' Valor del tratamiento a ser probado
            ,
            ' Identificado el tratamiento, hay que comprobar que
            ' cumple 2 condiciones:
            ' Condición 1: Secuencia de los tratamientos con sólo signo +
            ' debe ser en orden ascendente para evitar la permutación
            ' de las columnas en la búsqueda
            ,
            NF1 = NF-1 ' Indica el número de filas evaluadas antes del
                       ' último tratamiento posible
            ,
            ' Para ubicar el tratamiento 5, el tratamiento 3 debe haber
            ' sido seleccionado previamente
            ,
            IF VP = 5 THEN
                J=0
                FOR I=1 TO NF1          ' evalúa si el tratamiento 3 fue seleccionado
                                         ' en las posiciones anteriores
                    IF CORR(I)=3 THEN
                        J=1              ' el tratamiento 5 es posible en esta
                                         ' ubicación
                    END IF
                NEXT
                IF J=0 THEN
                    VP=CORR(NF-1)      ' Se ubica en el último tratamiento posible
                    GOTO 10             ' el tratamiento 5 no es factible en esta
                                         ' ubicación
                END IF
            END IF
            ,
            ' Para ubicar el tratamiento 9, el tratamiento 5 debe haber sido
            ' seleccionado previamente
            ,
            IF VP = 9 THEN
                J=0
                FOR I=1 TO NF1
                    IF CORR(I)=5 THEN
                        J=1
                    END IF
                NEXT
                IF J=0 THEN
                    VP=CORR(NF-1)
                    GOTO 10
                END IF
            END IF
            ,
            ' Para ubicar el tratamiento 17, el tratamiento 9 debe haber
            ' sido seleccionado previamente
            ,
            IF VP = 17 THEN
                J=0
                FOR I=1 TO NF1
                    IF CORR(I)=9 THEN
                        J=1
                    END IF
                NEXT
                IF J=0 THEN
                    VP=CORR(NF-1)
                    GOTO 10
                END IF
            END IF
```

Anexo 4: Programa para la búsqueda exhaustiva de órdenes semilla

```

END IF
'
' Condición 2: Verificar que el tratamiento seleccionado
' no ha aparecido previamente
'
FOR I=1 TO NF1
  IF VP=CORR(I) THEN ' el tratamiento ha sido seleccionado
                    ' previamente
                    VP=CORR(NF-1) ' Volver al anterior tratamiento factible
                    GOTO 10        ' El tratamiento evaluado no es factible
                                    ' en esta posición
  END IF
NEXT
'
' El tratamiento es posible. Guardarlo, aumentar
' el contador de la fila y seguir
'
CORR(NF)=VP ' Guardar tratamiento en la posición de la fila
NF=NF+1    ' Aumentar el contador de la fila
IF NF=33 THEN ' Se han completado los tratamientos de un orden
  GOTO 20    ' ir a guardar el orden hallado
END IF
GOTO 50     ' Terminar
'
' El tratamiento seleccionado no es factible. Se colocan
' los apuntadores en la posición del tratamiento posible
' anterior para buscar un nuevo sucesor
'
10
K=AP(VP-1)
J=AX(VP-1)
IF K<J THEN
  GOTO 50
END IF
'
' Modificar los apuntadores cuando se han probado todos
' los posibles sucesores de una fila y registrar la fila
' en que se encuentra el último tratamiento factible
'
40
NF=NF-1
AP(VP-1)=0
VP=CORR(NF-1)
IF NF=2 THEN
  GOTO 50
END IF
GOTO 10
'
' Guardar el nuevo orden hallado
'
20
' Ha terminado de encontrar un nuevo orden
' Se Construye la matriz de diseño en el orden propuesto
'
FOR I=1 TO 32
  FOR J=1 TO 5
    MATORD(I,J)=MATSTD(CORR(I), J)
  NEXT
NEXT
'
' Se imprime la matriz de diseño del orden obtenido
'
FOR I=1 TO 32
  FOR J=1 TO 5
    IF J<5 THEN
      'PRINT MATORD(I,J);
    END IF
    IF J=5 THEN
      'PRINT MATORD(I,J)
    END IF
  NEXT
NEXT
'
' Cuenta el número de cambios de nivel de los factores
' en el orden hallado
'

```

Anexo 4: Programa para la búsqueda exhaustiva de órdenes semilla

```

CAM=0
FOR I=2 TO 32
  FOR J=1 TO 5
    IF NOT MATORD(I,J) = MATORD(I-1,J) THEN
      CAM = CAM+1
    END IF
  NEXT
NEXT
'
' Cuenta el sesgo máximo (máximo efecto)
'
' Se construye la matriz completa con todos los factores
' e interacciones
'
FOR I=1 TO 32
  MATCOM(I,1)=MATORD(I,1)
  MATCOM(I,2)=MATORD(I,2)
  MATCOM(I,3)=MATORD(I,3)
  MATCOM(I,4)=MATORD(I,4)
  MATCOM(I,5)=MATORD(I,5)
  MATCOM(I,6)=MATORD(I,1)*MATORD(I,2)
  MATCOM(I,7)=MATORD(I,1)*MATORD(I,3)
  MATCOM(I,8)=MATORD(I,1)*MATORD(I,4)
  MATCOM(I,9)=MATORD(I,1)*MATORD(I,5)
  MATCOM(I,10)=MATORD(I,2)*MATORD(I,3)
  MATCOM(I,11)=MATORD(I,2)*MATORD(I,4)
  MATCOM(I,12)=MATORD(I,2)*MATORD(I,5)
  MATCOM(I,13)=MATORD(I,3)*MATORD(I,4)
  MATCOM(I,14)=MATORD(I,3)*MATORD(I,5)
  MATCOM(I,15)=MATORD(I,4)*MATORD(I,5)
  MATCOM(I,16)=MATORD(I,1)*MATORD(I,2)*MATORD(I,3)
  MATCOM(I,17)=MATORD(I,1)*MATORD(I,2)*MATORD(I,4)
  MATCOM(I,18)=MATORD(I,1)*MATORD(I,2)*MATORD(I,5)
  MATCOM(I,19)=MATORD(I,1)*MATORD(I,3)*MATORD(I,4)
  MATCOM(I,20)=MATORD(I,1)*MATORD(I,3)*MATORD(I,5)
  MATCOM(I,21)=MATORD(I,1)*MATORD(I,4)*MATORD(I,5)
  MATCOM(I,22)=MATORD(I,2)*MATORD(I,3)*MATORD(I,4)
  MATCOM(I,23)=MATORD(I,2)*MATORD(I,3)*MATORD(I,5)
  MATCOM(I,24)=MATORD(I,2)*MATORD(I,4)*MATORD(I,5)
  MATCOM(I,25)=MATORD(I,3)*MATORD(I,4)*MATORD(I,5)
  MATCOM(I,26)=MATORD(I,1)*MATORD(I,2)*MATORD(I,3)*MATORD(I,4)
  MATCOM(I,27)=MATORD(I,1)*MATORD(I,2)*MATORD(I,3)*MATORD(I,5)
  MATCOM(I,28)=MATORD(I,1)*MATORD(I,2)*MATORD(I,4)*MATORD(I,5)
  MATCOM(I,29)=MATORD(I,1)*MATORD(I,3)*MATORD(I,4)*MATORD(I,5)
  MATCOM(I,30)=MATORD(I,2)*MATORD(I,3)*MATORD(I,4)*MATORD(I,5)
  MATCOM(I,31)=MATORD(I,1)*MATORD(I,2)*MATORD(I,3)*MATORD(I,4)*MATORD(I,5)
NEXT
'
' Se imprime la matriz completa en un archivo de salida
' (sólo para probarla una vez)
'
FOR I=1 TO 32
  FOR J=1 TO 31
    IF J<31 THEN
      'PRINT MATCOM(I,J);
    END IF
    IF J=31 THEN
      'PRINT MATCOM(I,J)
    END IF
  NEXT
NEXT
'
' Calcula los efectos de la matriz completa
'
FOR I=1 TO 31
  EFEC(I)=0
NEXT
FOR J=1 TO 31
  FOR I=1 TO 32
    EFEC(J)=EFEC(J)+MATCOM(I,J)*RESP(I)
  NEXT
NEXT
'
' Busca el efecto o sesgo máximo
'

```

Anexo 4: Programa para la búsqueda exhaustiva de órdenes semilla

```

EFMAX=ABS(EFEC(1))
FOR J=2 TO 31
  IF ABS(EFEC(J))>EFMAX THEN
    EFMAX=ABS(EFEC(J))
  END IF
NEXT
EFMAX=EFMAX/16
'
' Tabulación del número de órdenes obtenidos en cada efecto
' Iniciando en un efecto 4 y terminando en 16 con incrementos
' de 0,25
'
' Construye los valores correspondientes a posibles efectos
'
TABEFE(1,1)=99 ' En la primera fila de la tabla se almacena
               ' el número de órdenes con el efecto máximo
               ' menor que 4, si existen

FOR I=2 TO 50   ' Número de filas necesarias para almacenar
               ' los efectos
  TABEFE(I,1)=4+(I-2)/4 ' Genera los incrementos de 0,25
                       ' en el sesgo
NEXT
'
' Si el valor del efecto máximo es menor que 4 entonces
' ubíquelo al frente del efecto 99 en la primera fila
' de la tabla y aumente el contador en 1
'
IF EFMAX<4 THEN
  TABEFE(1,2)=TABEFE(1,2)+1
'
' De lo contrario ubíquelo en la posición del efecto
' correspondiente y aumente el contador en 1
'
ELSE
  I=-14+4*EFMAX ' Se determina la posición en la que se ubica
                ' el sesgo máximo hallado
  TABEFE(I,2)=TABEFE(I,2)+1 ' Se incrementa el contador para
                            ' cada sesgo máximo hallado en 1
END IF
'
' Si el valor del efecto es menor que 5 imprima el orden en
' el archivo de salida
'
IF EFMAX<5 THEN
  FOR I=1 TO 32
    WRITE #1, CORR(I),
  NEXT I
  WRITE #1, EFMAX
END IF
'
'
' Se actualizan las variables después de que un nuevo
' orden ha sido hallado
'
NC=NC+1
PRINT NC; CAM; EFMAX
NF=31
VP=CORR(30) ' Se ubica el último tratamiento posible
            ' en la antepenúltima fila
K=CORR(31)  ' Se actualiza el apuntador para la penúltima fila
AP(K-1)=0
K=CORR(32)
AP(K-1)=0 ' Se actualiza el apuntador para la última fila
'
' Para parar después de un cierto número de órdenes encontrados
' (usado sólo en pruebas iniciales)
'
'IF NC=1 THEN
'  NF=1
'END IF
'
' Se finaliza el proceso de búsqueda
'
50
WEND

```

*Anexo 4: Programa para la búsqueda exhaustiva de órdenes semilla*

```
'  
' Imprime archivo de salida con la tabla de efectos y el número  
' de órdenes hallados  
'  
FOR I=1 TO 50  
  WRITE #1, TABEFE(I,1),TABEFE(I,2)  
NEXT  
WRITE #1, NC  
CLOSE #1  
  
SLEEP
```

## Anexo 5

### **Orden con las propiedades deseadas para un diseño $2^7$ obtenido con el algoritmo de Dickinson con alimentación forzada**

La búsqueda exhaustiva no es viable para diseños  $2^6$  y  $2^7$  debido al excesivo tiempo de computación que precisan.

Para diseños  $2^7$  la ordenación encontrada que produce un menor sesgo (con el mínimo número de cambios) no se obtiene a través de la búsqueda aleatoria sino duplicando el mejor orden hallado para diseños  $2^6$ , obteniéndose un sesgo de 16. Para mejorar este valor se ha organizado el programa que aplica el algoritmo de Dickinson de forma que el primer orden que encuentre sea el mejor de los conocidos, de forma que introduciendo cambios sobre este orden de partida se puedan encontrar otros con un sesgo menor.

Por este procedimiento se han obtenido dos ordenaciones que tienen un sesgo de 15,0625 y este es el valor mínimo obtenido. La primera de estas ordenaciones está incluida en el capítulo 4, y la segunda se presenta en este anexo.

Ver tabla en la página siguiente.

Anexo 5: Orden con las propiedades deseadas para un diseño  $2^7$

Posición	# Orden	Posición	# Orden	Posición	# Orden	Posición	#Orden
1	1	17	25	65	73	97	103
2	2	18	26	66	74	98	101
3	4	19	18	67	90	99	69
4	3	20	17	68	89	100	65
5	67	21	49	69	93	101	81
6	68	22	50	70	94	102	83
7	72	23	52	71	30	103	91
8	71	24	51	72	29	104	75
9	7	25	115	73	61	105	76
10	8	26	116	74	62	106	92
11	40	27	124	75	58	107	84
12	39	28	123	76	57	108	82
13	47	29	107	77	121	109	66
14	48	30	108	78	122	110	70
15	16	31	106	79	126	111	102
16	15	32	105	80	125	112	104
17	31	49	97	81	127	113	100
18	32	50	98	82	119	114	99
19	24	51	114	83	120	115	35
20	23	52	113	84	128	116	43
21	21	53	117	85	112	117	11
22	22	54	118	86	110	118	12
23	54	55	86	87	46	119	28
24	53	56	85	88	42	120	60
25	37	57	87	89	44	121	64
26	38	58	88	90	36	122	56
27	6	59	96	91	34	123	55
28	5	60	95	92	33	124	63
29	13	61	79	93	41	125	59
30	14	62	80	94	45	126	27
31	10	63	78	95	109	127	19
32	9	64	77	96	111	128	20

## Anexo 6

### Macro en el lenguaje de Minitab para obtener órdenes con las propiedades deseadas usando el método de permutación y cambios de signo

Esta macro, integrada en el entorno del paquete de software estadístico Minitab, convierte el orden estándar en un nuevo orden de experimentación que tiene las propiedades deseadas para cualquiera de los diseños que se pueden plantear con este software.

La macro ya tiene incorporados los órdenes semilla obtenidos para cada uno de los diseños (para diseños con hasta 16 experimentos ya eran conocidos) y a partir de estos se generan los órdenes de experimentación utilizando el método de las permutaciones y/o cambio de signo de algunas columnas. De esta forma, a partir de un orden semilla de un diseño  $2^k$  pueden obtenerse  $k! \cdot 2^k$  órdenes con las mismas propiedades. La tabla A6.1 resume el número de órdenes semilla que tenemos para cada tipo de diseños (identificado por el número de experimentos que requiere) y cuantos se pueden obtener a partir de estos. La macro selecciona uno al azar del número total de órdenes.

Tabla A6.1: Número total de órdenes de entre los que la macro elige uno al azar

Número de Experimentos	Número de órdenes semilla	Número de órdenes por orden semilla	Número total de órdenes <sup>(1)</sup>	
4	1	8	8	<sup>(2)</sup>
8	1	48	48	<sup>(2)</sup>
16	16	384	6.144	<sup>(2)</sup>
32	2	3.840	7.680	<sup>(2)</sup>
64	2	46.080	92.160	
128	2	645.120	1.290.240	

<sup>(1)</sup> De entre los cuales la macro elige uno

<sup>(2)</sup> Estos son todos los posibles

Se empieza leyendo un orden semilla al azar de entre los disponibles según sea el diseño de que se trate. A continuación la macro está estructurada en subrutinas.

La subrutina “GENERADORES” es muy larga porque es necesario introducir en la macro todos los generadores que utiliza Minitab para que en los diseños fraccionales, una vez ordenados los factores que se acomodan en la matriz completa, puedan ordenarse también los factores que se obtienen a partir de los generadores.

La subrutina “IDENTIFICA” la utilizamos para colocar en la columna “StdOrder” los órdenes que corresponden a la matriz en orden estándar. Se realiza mediante una fórmula polinómica distinta para cada diseño, de manera que aplicando esta fórmula a los valores que toman los factores en cada fila se obtiene un número distinto para cada una de ellas, lo cual sirve para identificarlas.

A pesar de su longitud, la macro funciona muy rápido incluso para los mayores diseños.

```
#####
# PERMU021206.MAC #
# #
# ¿QUÉ HACE? #
# A partir de órdenes semilla ya introducidos, genera al azar #
# otro orden que cumple con las propiedades: #
# -Mínimo sesgo máximo absoluto #
# -Mínimo número de cambios en los factores de la matriz completa #
# a base de permutar el orden de las columnas y/o cambiar los #
# signos ninguna o a alguna de ellas #
# #
# ¿CÓMO UTILIZARLO? #
# Generar con Minitab cualquier diseño factorial completo o #
# o fraccional de entre todos los disponibles, con todas las #
# opciones por defecto excepto la de aleatorización, que debe #
# quitarse para que el diseño se presente en orden estándar. #
# Ejecutar la macro. El diseño queda en el orden adecuado para #
# introducir las respuestas y ser analizado #
# #
# Preparado para Minitab versión 14 #
# #
#####

GMACRO
PERMU
BRIEF 0# Para que no salgan resultados intermedios por la pantalla
#
# K1: Número de experimentos
# K2: Número de columnas a ordenar
# K3: Número de factores en diseño completo
# K12: Número aleatorio para seleccionar orden semilla

# C1: Orden semilla seleccionado
# c99: Coloca el núm. aleatorio para seleccionar orden semilla
# c101: Orden semilla seleccionado
# C102: Secuencia para ordenar
# C103: Números de 1 a k3. Primero seguidos, después desordenados
# c104: Números de 1 a K1
# C105: Valores para identificar las filas
# c110: Todos los órdenes semilla

# Cuenta el número de experimentos
COUNT C1 K1

# Cuenta el número de columnas a ordenar
LET K4=0
DO K5=7:19
  COUNT CK5 K3
  LET K4=K4+K3
ENDDO
LET K2=6+K4/K1

# Selecciona e introduce órdenes semilla
```

Anexo 6: Macro para obtener órdenes. Método de la permutación y cambios de signo

```

# Diseños con 4 experimentos
IF K1=4
  SET C101
  1 2 4 3
  END
ENDIF
#
# Diseños con 8 experimentos
IF K1=8
  SET C101
  1 2 6 8 4 3 7 5
  END
ENDIF
#
# Diseños con 16 experimentos
IF K1=16
  SET C110
  1 2 4 3 11 12 16 8 7 15 13 5 6 14 10 9 #1
  1 2 4 3 11 12 16 15 7 8 6 5 13 14 10 9 #2
  1 2 4 3 11 15 7 8 16 12 10 14 6 5 13 9 #3
  1 2 4 8 16 12 11 3 7 15 13 5 6 14 10 9 #4
  1 2 4 8 16 15 7 3 11 12 10 14 6 5 13 9 #5
  1 2 4 12 11 3 7 5 13 15 16 8 6 14 10 9 #6
  1 2 4 12 11 3 7 8 16 15 13 5 6 14 10 9 #7
  1 2 4 12 11 3 7 15 13 5 6 8 16 14 10 9 #8
  1 2 4 12 11 3 7 15 16 8 6 5 13 14 10 9 #9
  1 2 4 12 16 8 7 3 11 15 13 5 6 14 10 9 #10
  1 2 10 12 4 3 7 5 13 14 6 8 16 15 11 9 #11
  1 2 10 12 4 3 7 8 16 14 6 5 13 15 11 9 #12
  1 2 10 14 16 12 4 3 11 15 7 8 6 5 13 9 #13
  1 2 10 14 16 12 11 15 7 3 4 8 6 5 13 9 #14
  1 2 10 14 16 15 7 3 11 12 4 8 6 5 13 9 #15
  1 2 10 14 16 15 11 12 4 3 7 8 6 5 13 9 #16
  END
  RANDOM 1 C99; # Número aleatorio para elegir uno entre los órdenes semilla
  INTEGER 1 16.
  LET K12=C99(1)
  DO K5=1:16
    LET C101(K5)=C110(16*(K12-1)+K5)
  ENDDO
ENDIF
#
# Diseños con 32 experimentos
IF K1=32
  SET C110
  1 2 4 8 16 12 11 15 13 29 21 22 18 20 24 32 #1
  30 26 28 27 31 23 19 3 7 5 6 14 10 9 25 17
  1 2 18 22 30 29 25 27 19 3 11 12 4 8 6 14 #2
  16 15 7 5 13 9 10 26 28 20 24 32 31 23 21 17
  END
  RANDOM 1 C99; # Número aleatorio para elegir uno entre los órdenes semilla
  INTEGER 1 2.
  LET K12=C99(1)
  DO K5=1:32
    LET C101(K5)=C110(32*(K12-1)+K5)
  ENDDO
ENDIF
#
# Diseños con 64 experimentos
IF K1=64
  SET C110
  1 2 34 36 4 20 24 8 16 12 11 27 19 3 7 5 #1
  13 9 25 26 58 62 54 53 49 57 59 43 44 48 40 39
  37 45 47 15 31 29 61 63 64 56 55 23 21 17 18 50
  52 51 35 33 41 42 10 14 46 38 6 22 30 32 28 60
  1 2 34 36 4 20 24 8 16 12 11 27 19 3 7 5 #2
  13 9 25 26 58 62 54 53 49 57 59 43 44 48 40 39
  37 45 61 29 31 15 47 63 64 56 55 23 21 17 18 50
  52 51 35 33 41 42 10 14 46 38 6 22 30 32 28 60
  END
  RANDOM 1 C99; # Número aleatorio para elegir uno entre los órdenes semilla
  INTEGER 1 2.
  LET K12=C99(1)
  DO K5=1:64
    LET C101(K5)=C110(64*(K12-1)+K5)
  ENDDO
ENDIF

```

Anexo 6: Macro para obtener órdenes. Método de la permutación y cambios de signo

```

ENDDO
ENDIF
#
# Diseños con 128 experimentos
IF K1=128
  SET C110
  1 2 4 3 67 68 72 71 7 8 40 39 47 48 16 15          #1
  31 32 24 23 21 22 54 53 37 38 6 5 13 14 10 9
  25 26 18 17 49 50 52 51 115 116 124 123 107 108 106 105
  97 98 114 113 117 118 86 85 87 88 96 95 79 80 78 77
  73 74 90 89 93 94 30 29 61 62 58 57 121 122 126 125
  127 128 120 119 103 111 112 110 109 45 46 42 41 33 34 36
  44 12 76 92 91 83 84 82 81 65 66 70 69 101 102 104
  100 99 35 43 59 63 55 56 64 60 28 20 19 27 11 75
  1 2 4 3 67 68 72 71 7 8 40 39 47 48 16 15          #2
  31 32 24 23 21 22 54 53 37 38 6 5 13 14 10 9
  25 26 18 17 49 50 52 51 115 116 124 123 107 108 106 105
  97 98 114 113 117 118 86 85 87 88 96 95 79 80 78 77
  73 74 90 89 93 94 30 29 61 62 58 57 121 122 126 125
  127 119 120 128 112 110 46 42 44 36 34 33 41 45 109 111
  103 101 69 65 81 83 91 75 76 92 84 82 66 70 102 104
  100 99 35 43 11 12 28 60 64 56 55 63 59 27 19 20
END
RANDOM 1 C99;      # Número aleatorio para elegir uno entre los órdenes semilla
  INTEGER 1 2.
  LET K12=C99(1)
  DO K5=1:128
    LET C101(K5)=C110(128*(K12-1)+K5)
  ENDDO
ENDIF
#
# SUBROUTINAS
#
# Coloca en el orden semilla
CALL COLOCA

# Permutación
CALL PERMUTA

# Cambios de signo
CALL SIGNOS

# Recalcular columnas de generadores (en diseños fraccionales)
CALL GENERADORES

# Identificación del nuevo orden
CALL IDENTIFICA

BRIEF 2
ENDMACRO

GMACRO
COLOCA
SET C104
  1:K1
END
CONVERT C101 C104 C1 C1
SORT C1 C3-CK2 C1 C3-CK2;      # Ordena
BY C1.
ENDMACRO

GMACRO
PERMUTA
LET K3=LOGE(K1)/LOGE(2)      # Calcula el número de factores
LET K4=K3+4
LET K24=K3+24
COPY C5-CK4 C25-CK24      # Copia la matriz de diseño en C25-CK24
SET C103
  1:K3
END
SAMPLE K3 C103 C103
DO K5=5:K4
  LET K6=C103(K5-4)+24
  COPY CK5 CK6
ENDDO
ENDMACRO

```

```
GMACRO
SIGNOS
RANDOM K3 C103;
  BERNOUILLI 0.5.
DO K5=5:K4
  IF C103(K5-4)=1
    LET CK5=-1*CK5
  ENDIF
ENDDO
ENDMACRO

GMACRO
GENERADORES
LET K4=K2-4

IF K4=2
  GOTO 10
ENDIF

IF K4=3
  IF K1=4
    LET 'C'='A'*'B'
  ENDIF
  GOTO 10
ENDIF

IF K4=4
  IF K1=8
    LET 'D'='A'*'B'*'C'
  ENDIF
  GOTO 10
ENDIF

IF K4=5
  IF K1=8
    LET 'D'='A'*'B'
    LET 'E'='A'*'C'
  ENDIF
  IF K1=16
    LET 'E'='A'*'B'*'C'
  ENDIF
  GOTO 10
ENDIF

IF K4=6
  IF K1=8
    LET 'D'='A'*'B'
    LET 'E'='A'*'C'
    LET 'F'='B'*'C'
  ENDIF
  IF K1=16
    LET 'E'='A'*'B'*'C'
    LET 'F'='B'*'C'*'D'
  ENDIF
  IF K1=32
    LET 'F'='A'*'B'*'C'*'D'*'E'
  ENDIF
  GOTO 10
ENDIF

IF K4=7
  IF K1=8
    LET 'D'='A'*'B'
    LET 'E'='A'*'C'
    LET 'F'='B'*'C'
    LET 'G'='A'*'B'*'C'
  ENDIF
  IF K1=16
    LET 'E'='A'*'B'*'C'
    LET 'F'='B'*'C'*'D'
    LET 'G'='A'*'C'*'D'
  ENDIF
  IF K1=32
    LET 'F'='A'*'B'*'C'*'D'
    LET 'G'='A'*'B'*'D'*'E'
```

Anexo 6: Macro para obtener órdenes. Método de la permutación y cambios de signo

```
ENDIF
IF K1=64
  LET 'G'='A'*'B'*'C'*'D'*'E'*'F'
ENDIF
GOTO 10
ENDIF

IF K4=8
  IF K1=16
    LET 'E'='B'*'C'*'D'
    LET 'F'='A'*'C'*'D'
    LET 'G'='A'*'B'*'C'
    LET 'H'='A'*'B'*'D'
  ENDIF
  IF K1=32
    LET 'F'='A'*'B'*'C'
    LET 'G'='A'*'B'*'D'
    LET 'H'='B'*'C'*'D'*'E'
  ENDIF
  IF K1=64
    LET 'G'='A'*'B'*'C'*'D'
    LET 'H'='A'*'B'*'E'*'F'
  ENDIF
  IF K1=128
    LET 'H'='A'*'B'*'C'*'D'*'E'*'F'*'G'
  ENDIF
  GOTO 10
ENDIF

IF K4=9
  IF K1=16
    LET 'E'='A'*'B'*'C'
    LET 'F'='B'*'C'*'D'
    LET 'G'='A'*'C'*'D'
    LET 'H'='A'*'B'*'D'
    LET 'J'='A'*'B'*'C'*'D'
  ENDIF
  IF K1=32
    LET 'F'='B'*'C'*'D'*'E'
    LET 'G'='A'*'C'*'D'*'E'
    LET 'H'='A'*'B'*'D'*'E'
    LET 'J'='A'*'B'*'C'*'E'
  ENDIF
  IF K1=64
    LET 'G'='A'*'B'*'C'*'D'
    LET 'H'='A'*'C'*'E'*'F'
    LET 'J'='C'*'D'*'E'*'F'
  ENDIF
  IF K1=128
    LET 'H'='A'*'C'*'D'*'F'*'G'
    LET 'J'='B'*'C'*'E'*'F'*'G'
  ENDIF
  GOTO 10
ENDIF

IF K4=10
  IF K1=16
    LET 'E'='A'*'B'*'C'
    LET 'F'='B'*'C'*'D'
    LET 'G'='A'*'C'*'D'
    LET 'H'='A'*'B'*'D'
    LET 'J'='A'*'B'*'C'*'D'
    LET 'J'='A'*'B'
  ENDIF
  IF K1=32
    LET 'F'='A'*'B'*'C'*'D'
    LET 'G'='A'*'B'*'C'*'E'
    LET 'H'='A'*'B'*'D'*'E'
    LET 'J'='A'*'C'*'D'*'E'
    LET 'K'='B'*'C'*'D'*'E'
  ENDIF
  IF K1=64
    LET 'G'='B'*'C'*'D'*'F'
    LET 'H'='A'*'C'*'D'*'F'
    LET 'J'='A'*'B'*'D'*'E'
    LET 'K'='A'*'B'*'C'*'E'
  ENDIF
```

Anexo 6: Macro para obtener órdenes. Método de la permutación y cambios de signo

```
ENDIF
IF K1=128
  LET 'H'='A'*'B'*'C'*'G'
  LET 'J'='B'*'C'*'D'*'E'
  LET 'K'='A'*'C'*'D'*'F'
ENDIF
GOTO 10
ENDIF

IF K4=11
  IF K1=16
    LET 'E'='A'*'B'*'C'
    LET 'F'='B'*'C'*'D'
    LET 'G'='A'*'C'*'D'
    LET 'H'='A'*'B'*'D'
    LET 'J'='A'*'B'*'C'*'D'
    LET 'K'='A'*'B'
    LET 'L'='A'*'C'
  ENDIF
  IF K1=32
    LET 'F'='A'*'B'*'C'
    LET 'G'='B'*'C'*'D'
    LET 'H'='C'*'D'*'E'
    LET 'J'='A'*'C'*'D'
    LET 'K'='A'*'D'*'E'
    LET 'L'='B'*'D'*'E'
  ENDIF
  IF K1=64
    LET 'G'='C'*'D'*'E'
    LET 'H'='A'*'B'*'C'*'D'
    LET 'J'='A'*'B'*'F'
    LET 'K'='B'*'D'*'E'*'F'
    LET 'L'='A'*'D'*'E'*'F'
  ENDIF
  IF K1=128
    LET 'H'='A'*'B'*'C'*'G'
    LET 'J'='B'*'C'*'D'*'E'
    LET 'K'='A'*'C'*'D'*'F'
    LET 'L'='A'*'B'*'C'*'D'*'E'*'F'*'G'
  ENDIF
  GOTO 10
ENDIF

IF K4=12
  IF K1=16
    LET 'E'='A'*'B'*'C'
    LET 'F'='A'*'B'*'D'
    LET 'G'='A'*'C'*'D'
    LET 'H'='B'*'C'*'D'
    LET 'J'='A'*'B'*'C'*'D'
    LET 'K'='A'*'B'
    LET 'L'='A'*'C'
    LET 'M'='A'*'D'
  ENDIF
  IF K1=32
    LET 'F'='A'*'C'*'E'
    LET 'G'='A'*'C'*'D'
    LET 'H'='A'*'B'*'D'
    LET 'J'='A'*'B'*'E'
    LET 'K'='C'*'D'*'E'
    LET 'L'='A'*'B'*'C'*'D'*'E'
    LET 'M'='A'*'D'*'E'
  ENDIF
  IF K1=64
    LET 'G'='D'*'E'*'F'
    LET 'H'='A'*'B'*'C'
    LET 'J'='B'*'C'*'D'*'E'
    LET 'K'='B'*'C'*'D'*'F'
    LET 'L'='A'*'B'*'E'*'F'
    LET 'M'='A'*'C'*'E'*'F'
  ENDIF
  IF K1=128
    LET 'H'='A'*'C'*'D'*'G'
    LET 'J'='A'*'B'*'C'*'D'
    LET 'K'='B'*'C'*'F'*'G'
    LET 'L'='A'*'B'*'D'*'E'*'F'*'G'
  ENDIF

```

Anexo 6: Macro para obtener órdenes. Método de la permutación y cambios de signo

```
    LET 'M'='C'*'D'*'E'*'F'
  ENDIF
  GOTO 10
ENDIF

IF K4=13
  IF K1=16
    LET 'E'='A'*'B'*'C'
    LET 'F'='A'*'B'*'D'
    LET 'G'='A'*'C'*'D'
    LET 'H'='B'*'C'*'D'
    LET 'J'='A'*'B'*'C'*'D'
    LET 'K'='A'*'B'
    LET 'L'='A'*'C'
    LET 'M'='A'*'D'
    LET 'N'='B'*'C'
  ENDIF
  IF K1=32
    LET 'F'='A'*'C'*'E'
    LET 'G'='B'*'C'*'E'
    LET 'H'='A'*'B'*'C'
    LET 'J'='C'*'D'*'E'
    LET 'K'='A'*'B'*'C'*'D'*'E'
    LET 'L'='A'*'B'*'E'
    LET 'M'='A'*'C'*'D'
    LET 'N'='A'*'D'*'E'
  ENDIF
  IF K1=64
    LET 'G'='A'*'B'*'C'
    LET 'H'='D'*'E'*'F'
    LET 'J'='B'*'C'*'D'*'F'
    LET 'K'='B'*'C'*'D'*'E'
    LET 'L'='A'*'B'*'E'*'F'
    LET 'M'='A'*'C'*'E'*'F'
    LET 'N'='B'*'C'*'E'*'F'
  ENDIF
  IF K1=128
    LET 'H'='D'*'E'*'F'*'G'
    LET 'J'='B'*'C'*'E'*'G'
    LET 'K'='B'*'C'*'D'*'E'*'G'
    LET 'L'='A'*'B'*'D'*'E'*'F'
    LET 'M'='A'*'C'*'E'*'F'
    LET 'N'='A'*'B'*'C'
  ENDIF
  GOTO 10
ENDIF

IF K4=14
  IF K1=16
    LET 'E'='A'*'B'*'C'
    LET 'F'='A'*'B'*'D'
    LET 'G'='A'*'C'*'D'
    LET 'H'='B'*'C'*'D'
    LET 'J'='A'*'B'*'C'*'D'
    LET 'K'='A'*'B'
    LET 'L'='A'*'C'
    LET 'M'='A'*'D'
    LET 'N'='B'*'C'
    LET 'O'='B'*'D'
  ENDIF
  IF K1=32
    LET 'F'='A'*'B'*'C'
    LET 'G'='A'*'B'*'D'
    LET 'H'='A'*'B'*'E'
    LET 'J'='A'*'C'*'D'
    LET 'K'='A'*'C'*'E'
    LET 'L'='A'*'D'*'E'
    LET 'M'='B'*'C'*'D'
    LET 'N'='B'*'C'*'E'
    LET 'O'='B'*'D'*'E'
  ENDIF
  IF K1=64
    LET 'G'='B'*'E'*'F'
    LET 'H'='B'*'C'*'F'
    LET 'J'='D'*'E'*'F'
    LET 'K'='C'*'E'*'F'
```

Anexo 6: Macro para obtener órdenes. Método de la permutación y cambios de signo

```

LET 'L'='B'*'C'*'E'
LET 'M'='C'*'D'*'F'
LET 'N'='A'*'C'*'D'*'E'
LET 'O'='B'*'C'*'D'*'E'*'F'
ENDIF
IF K1=128
LET 'H'='E'*'F'*'G'
LET 'J'='B'*'C'*'F'*'G'
LET 'K'='B'*'C'*'E'*'G'
LET 'L'='A'*'B'*'E'*'F'
LET 'M'='A'*'C'*'E'*'F'
LET 'N'='B'*'C'*'D'*'E'*'F'
LET 'O'='A'*'B'*'C'
ENDIF
GOTO 10
ENDIF

IF K4=15
IF K1=16
LET 'E'='A'*'B'*'C'
LET 'F'='A'*'B'*'D'
LET 'G'='A'*'C'*'D'
LET 'H'='B'*'C'*'D'
LET 'J'='A'*'B'*'C'*'D'
LET 'K'='A'*'B'
LET 'L'='A'*'C'
LET 'M'='A'*'D'
LET 'N'='B'*'C'
LET 'O'='B'*'D'
LET 'P'='C'*'D'
ENDIF
IF K1=32
LET 'F'='A'*'B'*'C'
LET 'G'='A'*'B'*'D'
LET 'H'='A'*'B'*'E'
LET 'J'='A'*'C'*'D'
LET 'K'='A'*'C'*'E'
LET 'L'='A'*'D'*'E'
LET 'M'='B'*'C'*'D'
LET 'N'='B'*'C'*'E'
LET 'O'='B'*'D'*'E'
LET 'P'='C'*'D'*'E'
ENDIF
IF K1=64
LET 'G'='A'*'B'*'C'
LET 'H'='A'*'B'*'D'
LET 'J'='A'*'B'*'E'
LET 'K'='A'*'B'*'F'
LET 'L'='A'*'C'*'D'
LET 'M'='A'*'C'*'E'
LET 'N'='A'*'C'*'F'
LET 'O'='A'*'D'*'E'
LET 'P'='A'*'D'*'F'
ENDIF
IF K1=128
LET 'H'='A'*'B'*'F'*'G'
LET 'J'='A'*'C'*'D'*'E'*'F'
LET 'K'='B'*'E'*'F'
LET 'L'='A'*'B'*'C'*'E'*'G'
LET 'M'='C'*'D'*'F'*'G'
LET 'N'='A'*'C'*'D'*'E'*'G'
LET 'O'='E'*'F'*'G'
LET 'P'='A'*'B'*'D'*'E'*'F'*'G'
ENDIF
ENDIF

MLABEL 10
ENDMACRO

GMACRO
IDENTIFICA
IF K1=4
LET C1=10*'A'+ 'B'
SET C105
-11 9 -9 11
END

```

Anexo 6: Macro para obtener órdenes. Método de la permutación y cambios de signo

```

ENDIF
IF K1=8
  LET C1=100*'A'+10*'B'+'C'
  SET C105
  -111 89 -91 109 -109 91 -89 111
  END
ENDIF
IF K1=16
  LET C1=1000*'A'+100*'B'+10*'C'+'D'
  SET C105
  -1111 889 -911 1089 -1091 909 -891 1109
  -1109 891 -909 1091 -1089 911 -889 1111
  END
ENDIF
IF K1=32
  LET C1=10000*'A'+1000*'B'+100*'C'+10*'D'+'E'
  SET C105
  -11111 8889 -9111 10889 -10911 9089 -8911 11089
  -11091 8909 -9091 10909 -10891 9109 -8891 11109
  -11109 8891 -9109 10891 -10909 9091 -8909 11091
  -11089 8911 -9089 10911 -10889 9111 -8889 11111
  END
ENDIF
IF K1=64
  LET C1=100000*'A'+10000*'B'+1000*'C'+100*'D'+10*'E'+'F'
  SET C105
  -111111 88889 -91111 108889 -109111 90889 -89111
  110889 -110911 89089 -90911 109089 -108911 91089
  -88911 111089 -111091 88909 -91091 108909 -109091
  90909 -89091 110909 -110891 89109 -90891 109109
  -108891 91109 -88891 111109 -111109 88891 -91109
  108891 -109109 90891 -89109 110891 -110909 89091
  -90909 109091 -108909 91091 -88909 111091 -111089
  88911 -91089 108911 -109089 90911 -89089 110911
  -110889 89111 -90889 109111 -108889 91111 -88889
  111111
  END
ENDIF
IF K1=128
  LET C1=1000000*'A'+100000*'B'+10000*'C'+1000*'D'+100*'E'+10*'F'+'G'
  SET C105
  -1111111 888889 -911111 1088889 -1091111 908889 -891111
  1108889 -1109111 890889 -909111 1090889 -1089111 910889
  -889111 1110889 -1110911 889089 -910911 1089089 -1090911
  909089 -890911 1109089 -1108911 891089 -908911 1091089
  -1088911 911089 -888911 1111089 -1111091 888909 -911091
  1088909 -1091091 908909 -891091 1108909 -1109091 890909
  -909091 1090909 -1089091 910909 -889091 1110909 -1110891
  889109 -910891 1089109 -1090891 909109 -890891 1109109
  -1108891 891109 -908891 1091109 -1088891 911109 -888891
  1111109 -1111109 888891 -911109 1088891 -1091109 908891
  -891109 1108891 -1109109 890891 -909109 1090891 -1089109
  910891 -889109 1110891 -1110909 889091 -910909 1089091
  -1090909 909091 -890909 1109091 -1108909 891091 -908909
  1091091 -1088909 911091 -888909 1111091 -1111089 888911
  -911089 1088911 -1091089 908911 -891089 1108911 -1109089
  890911 -909089 1090911 -1089089 910911 -889089 1110911
  -1110889 889111 -910889 1089111 -1090889 909111 -890889
  1109111 -1108889 891111 -908889 1091111 -1088889 911111
  -888889 1111111
  END
ENDIF
CONVERT C105 C104 C1 C1
ENDMACRO

```

## Anexo 7

### Macro en el lenguaje de Minitab para obtener órdenes con las propiedades deseadas usando el método de las columnas de expansión

Esta macro, también integrada en el entorno del paquete de software estadístico Minitab, realiza la misma tarea que la macro presentada en el anexo anterior, aunque utilizando un método distinto que puede aprovecharse para orientar la búsqueda solo hacia aquellos órdenes que reúnan unas propiedades especiales (como concentrar el sesgo máximo fuera de los efectos principales).

Dado un orden de un diseño  $2^k$  con las propiedades deseadas, de él se pueden deducir  $2k$  columnas de expansión para llegar a un orden con las mismas propiedades que este a partir del orden de un diseño  $2^{k-1}$  con esas propiedades. Por tanto, los vectores de expansión que se pueden deducir de cada diseño son los que figuran en la Tabla A7.1.

Tabla A7.1: Columnas de expansión que se pueden deducir de cada diseño

Diseño	Columnas de expansión	Número de órdenes a que se llega (por orden semilla)	
$2^3$	6	$8 \times 6 = 48$	( <sup>1</sup> )
$2^4$	8	$8 \times 6 \times 8 = 384$	
$2^5$	10	$8 \times 6 \times 8 \times 10 = 3.840$	
$2^6$	12	$8 \times 6 \times 8 \times 10 \times 12 = 46.080$	
$2^7$	14	$8 \times 6 \times 8 \times 10 \times 12 \times 14 = 645.120$	

(<sup>1</sup>) 8 son los órdenes de un diseño  $2^2$  a los que aplicar las columnas de expansión.

La macro que aquí se presenta está simplificada de forma que se aplica a un solo orden semilla por tipo de diseño, y en los diseños  $2^5$ ,  $2^6$  y  $2^7$  sólo se utilizan 2 columnas de expansión, de entre las disponibles. Así pues, esta macro genera al azar un orden con las propiedades deseadas de entre los que se indican en la Tabla A7.2. A efectos prácticos esta selección nos parece suficiente aunque, naturalmente, se puede completar la macro para poder considerar todas las posibilidades existentes.

Tabla A7.2: Número total de órdenes de entre los que la macro elige uno al azar

Número de Experimentos	Número de órdenes de entre los que esta macro selecciona uno al azar
4	8
8	48
16	384
32	768
64	1.536
128	3.072

```
#####
# COLUM161006.mac #
# #
# ¿QUE HACE? #
# A partir de 8 órdenes con el mínimo número de cambios de nivel #
# de los factores para un diseño 2^2 y las columnas de expansión #
# para los diseños 2^k con k=3,4,5,6,7 se obtienen aleatoriamente #
# órdenes con las siguientes propiedades: #
# - Mínimo sesgo máximo absoluto #
# - Mínimo número de cambios de nivel en los factores #
# de la matriz completa #
# #
# ¿CÓMO UTILIZARLO? #
# Generar con Minitab cualquier diseño factorial completo o #
# o fraccional de entre todos los disponibles, con todas las #
# opciones por defecto excepto la de aleatorización, que debe #
# quitarse para que el diseño se presente en orden estándar. #
# Ejecutar la macro. El diseño queda en el orden adecuado para #
# introducir las respuestas y ser analizado #
# #
# Preparado para Minitab versión 14 #
# #
#####

GMACRO
COLUM
brief 0 # Para que no salgan resultados intermedios por la pantalla
#
# K1: Número de experimentos
# k2: Número de columnas a ordenar
# K3 a K6: Contadores y auxiliares
#
# K12 Núm. aleatorio para el 2^2
# K13 Núm. aleatorio para el 2^3
# K14 Núm. aleatorio para el 2^4
# k15 Núm. aleatorio para el 2^5
# k16 Núm. aleatorio para el 2^6
# k17 Núm. aleatorio para el 2^7

# C101: Órdenes para el diseño 2^2
# C102: columna expansión para diseño 2^3
# c103: Sumas a la columna de expansión para el diseño 2^3
# c104: columna expansión para diseño 2^4
# c105: Sumas a la columna de expansión para el diseño 2^4
# c106: columna expansión para diseño 2^5
# c107: Sumas a la columna de expansión para el diseño 2^5
# c108: columna expansión para diseño 2^6
# c109: Sumas a la columna de expansión para el diseño 2^6
# c110: columna expansión para diseño 2^7
# c111: Sumas a la columna de expansión para el diseño 2^7
#
# c112: Lugar donde se depositan los números aleatorios
#
```

Anexo 7: Macro para obtener órdenes. Método de las columnas de expansión

```

# c122: Valores de la columna seleccionada para diseño 2^2
# c123: ... 2^3
# c124: ... 2^4
# c125: ... 2^5
# c126: ... 2^6
# c127: ... 2^7
#
# c142: Posiciones para el diseño 2^3
# C143: ... 2^4
# C144: ... 2^5
# C145: ... 2^6
# C146: ... 2^7

# c152: Columna donde se colocan antes de la presentación final

# Cuenta el número de experimentos
count c1 k1

# Cuenta el número de columnas a ordenar
let k4=0

DO K5=7:19
  COUNT CK5 K3
  LET K4=K4+K3
ENDDO

LET K2=6+K4/K1

# *****2^2*****

SET C101      # Para 2^2 (Total: 8), sesgo 2
1 2 4 3
1 3 4 2
2 1 3 4
2 4 3 1
3 4 2 1
3 1 2 4
4 2 1 3
4 3 1 2
END

RANDOM 1 C112;
  INTEGER 1 8.
LET K12=C112(1)

do k5=1:4      # Valores seleccionados para el 2^2
  let c122(k5)=c101(4*(k12-1)+k5)
enddo

if k1=4
  let c152=c122      # Para la presentación final
  goto 10
endif

# *****2^3*****

SET C102      # Para 2^3 (Total: 6), sesgo 2
1 1 4 3 2 2 3 4 # columna de expansión del factor A
1 1 4 3 2 2 3 4 # columna de expansión del factor A-
1 2 3 3 2 1 4 4 # columna de expansión del factor B
1 2 3 3 2 1 4 4 # columna de expansión del factor B-
1 2 2 3 3 4 4 1 # columna de expansión del factor C
1 2 2 3 3 4 4 1 # columna de expansión del factor C-
END

SET C103
0 4 4 4 4 0 0 0 # sumas columna A
4 0 0 0 0 4 4 4 # sumas columna A-
0 0 0 4 4 4 4 0 # sumas columna B
4 4 4 0 0 0 0 4 # sumas columna B-
0 0 4 4 0 0 4 4 # sumas columna C
4 4 0 0 4 4 0 0 # sumas columna C-
END

RANDOM 1 C112;
  INTEGER 1 6.

```

Anexo 7: Macro para obtener órdenes. Método de las columnas de expansión

```

LET K13=C112(1)

DO K5=1:8          # Valores seleccionados para el 2^3
  LET C123(K5)=C102(8*(K13-1)+K5)
ENDDO

DO K5=1:8          # Coloca en c142 las posiciones de acuerdo
  LET k6=c123(K5)  # con la columna de expansión
  LET c142(k5)=c122(K6)
ENDDO

DO k5=1:8          # Suma 4 donde corresponde
  LET c142(K5)=c142(K5)+c103(8*(K13-1)+K5)
ENDDO

IF K1=8
  LET c152=c142    # Para la presentación final
  GOTO 10
ENDIF

# *****2^4*****

SET C104          # Para 2^4 (Total 8: 4 col + sus inversas), sesgo 4
1 1 6 7 4 4 3 2 5 5 2 3 8 8 7 6 #columna de expansión del factor A
1 1 6 7 4 4 3 2 5 5 2 3 8 8 7 6 #columna de expansión del factor A-
1 2 3 4 4 7 6 1 8 3 2 5 5 6 7 8 #columna de expansión del factor B
1 2 3 4 4 7 6 1 8 3 2 5 5 6 7 8 #columna de expansión del factor B-
1 2 5 5 4 7 8 8 7 4 3 3 2 1 6 6 #columna de expansión del factor C
1 2 5 5 4 7 8 8 7 4 3 3 2 1 6 6 #columna de expansión del factor C-
1 2 2 5 4 7 7 8 8 3 3 4 5 6 6 1 #columna de expansión del factor D
1 2 2 5 4 7 7 8 8 3 3 4 5 6 6 1 #columna de expansión del factor D-
END

SET C105
0 8 8 8 8 0 0 0 0 8 8 8 8 0 0 0 # sumas columna A
8 0 0 0 0 8 8 8 8 0 0 0 0 8 8 8 # sumas columna A-
0 0 0 0 8 8 8 8 8 8 8 8 0 0 0 0 # sumas columna B
8 8 8 8 0 0 0 0 0 0 0 8 8 8 8 # sumas columna B-
0 0 0 8 8 8 8 0 0 0 0 8 8 8 8 0 # sumas columna C
8 8 8 0 0 0 0 8 8 8 8 0 0 0 0 8 # sumas columna C-
0 0 8 8 8 8 0 0 8 8 0 0 0 8 8 # sumas columna D
8 8 0 0 0 0 8 8 0 0 8 8 8 8 0 0 # sumas columna D-
END

RANDOM 1 C112;
  INTEGER 1 8.
LET K14=C112(1)

do K5=1:16        # Valores seleccionados para el 2^4
  LET C124(K5)=C104(16*(K14-1)+K5)
ENDDO

DO K5=1:16        # Coloca en c143 las posiciones de acuerdo
  LET K6=c124(K5)  # con el columna de expansión
  LET C143(K5)=C142(K6)
ENDDO

do K5=1:16        # Suma 8 donde corresponde
  LET C143(K5)=C143(K5)+C105(16*(K14-1)+K5)
ENDDO

IF K1=16
  LET C152=C143    # Para la presentación final
  GOTO 10
ENDIF

# *****2^5*****

SET C106          # Para 2^5 (Total 2: 1 + inverso), sesgo 4.75
1 2 11 12 5 10 9 6 15 15 14 13 2 11 12 5 # columna E
4 3 10 9 6 7 8 8 7 14 13 4 3 16 16 1 # sigue
1 2 11 12 5 10 9 6 15 15 14 13 2 11 12 5 # columna E-
4 3 10 9 6 7 8 8 7 14 13 4 3 16 16 1 # sigue
END

SET C107

```

Anexo 7: Macro para obtener órdenes. Método de las columnas de expansión

```

0 0 0 0 0 0 0 0 0 16 16 16 16 16 16 16 16 # sumas columna E
16 16 16 16 16 16 16 0 0 0 0 0 0 0 0 16 16 # sigue
16 16 16 16 16 16 16 16 16 0 0 0 0 0 0 0 0 # sumas columna E-
0 0 0 0 0 0 0 16 16 16 16 16 16 16 0 0 # sigue
END

RANDOM 1 C112;
  INTEGER 1 2.
LET K15=C112(1)

DO K5=1:32 # Valores seleccionados para el 2^5
  LET C125(K5)=C106(32*(K15-1)+K5)
ENDDO

DO K5=1:32 # Coloca en c144 las posiciones de acuerdo
  LET K6=C125(K5) # con la columna de expansión
  LET C144(K5)=C143(K6)
ENDDO

DO K5=1:32 # Suma 16 donde corresponde
  LET C144(K5)=C144(K5)+C107(32*(K15-1)+K5)
ENDDO

IF K1=32
  LET C152=c144 # Para la presentación final
  GOTO 10
ENDIF

# *****2^6*****

SET C108 # Para 2^6 (Total: 1 + el inverso), sesgo 8
1 2 2 3 3 14 15 4 5 6 7 20 23 24 25 26 #columna F
9 30 31 18 18 17 12 11 32 31 20 7 6 5 4 25 #sigue
26 9 8 8 21 10 10 21 16 15 22 22 11 32 13 13 #sigue
14 23 24 1 30 29 29 28 28 27 27 12 17 16 19 19
1 2 2 3 3 14 15 4 5 6 7 20 23 24 25 26 #columna F-
9 30 31 18 18 17 12 11 32 31 20 7 6 5 4 25 #sigue
26 9 8 8 21 10 10 21 16 15 22 22 11 32 13 13 #sigue
14 23 24 1 30 29 29 28 28 27 27 12 17 16 19 19
END

SET C109
0 0 32 32 0 0 0 0 0 0 0 0 0 0 0 0 # sumas columna F
0 0 0 0 32 32 32 32 32 32 32 32 32 32 32 # sigue
32 32 32 0 0 0 32 32 32 32 32 32 0 0 0 0 # sigue
32 32 32 32 32 32 32 0 0 32 32 0 0 0 0 0 32
32 32 0 0 32 32 32 32 32 32 32 32 32 32 32 32 # sumas columna F-
32 32 32 32 0 0 0 0 0 0 0 0 0 0 0 0 # sigue
0 0 0 32 32 32 0 0 0 0 0 32 32 32 32 0 # sigue
0 0 0 0 0 0 32 32 0 0 32 32 32 32 32 0
END

RANDOM 1 C112;
  INTEGER 1 2.
let K16=C112(1)

DO k5=1:64 # Valores seleccionados para el 2^6
  LET C126(K5)=C108(64*(K15-1)+K5)
ENDDO

DO K5=1:64 # Coloca en c144 las posiciones de acuerdo
  LET K6=C126(K5) # con la columna de expansión
  LET C145(K5)=C144(K6)
ENDDO

DO K5=1:64 # Suma 32 donde corresponde
  LET C145(K5)=C145(K5)+C109(64*(K15-1)+K5)
ENDDO

IF K1=64
  LET C152=C145 # Para la presentación final
  GOTO 10
ENDIF

# *****2^7*****

```

Anexo 7: Macro para obtener órdenes. Método de las columnas de expansión

```

SET C110          # Para 2^7 (Total: 1 + el inverso), sesgo 15.0625
1 2 5 14 14 5 8 15 15 8 31 32 35 30 9 36          #columna G
37 62 7 44 45 60 23 24 33 58 59 16 17 56 55 18    #sigue
19 20 47 46 25 48 49 50 50 49 64 27 28 29 54 53    #sigue
52 3 48 25 24 23 60 45 44 7 62 37 36 9 56 17      #sigue
18 55 20 19 38 61 61 38 39 22 21 26 26 21 22 39    #sigue
40 41 42 43 32 35 30 57 34 34 57 54 53 52 3 4      #sigue
29 10 10 63 12 13 6 47 46 1 2 59 16 33 58 31      #sigue
4 51 51 28 27 40 43 42 41 64 63 6 13 12 11 11
1 2 5 14 14 5 8 15 15 8 31 32 35 30 9 36          #columna G-
37 62 7 44 45 60 23 24 33 58 59 16 17 56 55 18    #sigue
19 20 47 46 25 48 49 50 50 49 64 27 28 29 54 53    #sigue
52 3 48 25 24 23 60 45 44 7 62 37 36 9 56 17      #sigue
18 55 20 19 38 61 61 38 39 22 21 26 26 21 22 39    #sigue
40 41 42 43 32 35 30 57 34 34 57 54 53 52 3 4      #sigue
29 10 10 63 12 13 6 47 46 1 2 59 16 33 58 31      #sigue
4 51 51 28 27 40 43 42 41 64 63 6 13 12 11 11
END

SET C111
0 0 0 0 64 64 64 64 0 0 0 0 0 0 0 0 0 0          # sumas columna G
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0            #sigue
0 0 0 0 0 0 0 0 64 64 64 64 64 64 64 64 64      #sigue
64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 #sigue
64 64 64 64 64 64 64 0 0 0 0 0 0 0 64 64 64 64  #sigue
64 64 64 64 64 64 64 64 64 64 0 0 0 0 0 0 0 0  #sigue
0 0 64 64 64 64 64 64 64 64 64 64 64 64 64 64  #sigue
64 64 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 64          #sigue
64 64 64 64 64 0 0 0 0 64 64 64 64 64 64 64 64 # sumas columna G-
64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 #sigue
64 64 64 64 64 64 64 64 0 0 0 0 0 0 0 0 0 0    #sigue
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0            #sigue
0 0 0 0 0 0 64 64 64 64 64 64 64 0 0 0 0 0 0    #sigue
0 0 0 0 0 0 0 0 0 64 64 64 64 64 64 64 64      #sigue
64 64 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0          #sigue
0 0 64 64 64 64 64 64 64 64 64 64 64 64 64 64 0
END

RANDOM 1 C112;
  INTEGER 1 2.
LET K17=C112(1)

DO K5=1:128          # Valores seleccionados para el 2^7
  LET C127(K5)=C110(128*(K17-1)+K5)
ENDDO

DO K5=1:128          # Coloca en c146 las posiciones de acuerdo
  LET K6=C127(K5)    # con la columna de expansión
  LET C146(K5)=C145(K6)
ENDDO

DO K5=1:128          # Suma 64 donde corresponde
  LET C146(K5)=C146(K5)+C111(128*(K17-1)+K5)
ENDDO

LET C152=C146          # Para la presentación final

# *****
MLABEL 10
DO K5=1:K1            # Presenta la secuencia para ordenar
  LET K6=C152(K5)
  LET C160(K6)=K5
ENDDO

SORT C1 C3-CK2 C1 C3-CK2; # Ordenación
BY C160.

BRIEF 2
ENDMACRO

```

## Anexo 8

### Algunas consideraciones sobre el análisis de los órdenes propuestos

El hecho de que algunos factores cambien pocas veces de valor a lo largo de la experimentación puede provocar que su aportación a la variabilidad de la respuesta esté minusvalorada, lo cual puede conducir a errores en la selección de los efectos significativos. Esta circunstancia es bien conocida, y están descritos los procedimientos de análisis rigurosos que pueden aplicarse cuando existen restricciones en la aleatorización como las que aquí se proponen. También es verdad que este tema presenta aspectos paradójicos, ya que, conceptualmente, el problema que puede presentar un determinado orden de experimentación no tiene que ver tanto con el orden en sí, como con la forma en que dicho orden ha sido obtenido (las hipótesis del modelo exigen que se haya obtenido de forma totalmente aleatoria).

Nuestra percepción, que coincide con la de autores como Webb et al. (2004), es que en la experimentación industrial es muy difícil respetar la ortodoxia de lo que debería ser un plan de experimentación totalmente aleatorizado y partiendo de cero en cada corrida. Y esto es así porque hacerlo “correctamente” no sería viable, y se asume que lo que se pierde en cuanto a “finura” en la interpretación de los resultados compensa sobradamente lo que se ahorra en la ejecución del experimentos.

La propuesta que se plantea en esta tesis hace frente a esta situación y ante factores difíciles de cambiar de nivel, propone ordenaciones que consiguen pocos cambios para esos factores y logran neutralizar la influencia de posibles factores ajenos a la experimentación. Esto difícilmente se consigue actuando por intuición y menos cuando, ya puestos a violar las reglas, se hacen primero todos los experimentos con el factor a un nivel y después el resto con el factor al otro nivel.

En este anexo incluimos las consideraciones y las propuestas de análisis más relevantes que hemos encontrado en la literatura. Respecto a encontrar un compromiso entre rigor y posibilidad de uso éste es seguramente un tema todavía abierto, y presentamos esta recopilación básicamente como punto de partida para futuras investigaciones.

#### **Dickinson (1974)**

Este trabajo de Dickinson, una de las referencias que más influencia ha tenido en el desarrollo de nuestra tesis, centra su interés en la búsqueda de órdenes de experimentación con las propiedades deseadas (mínimo número de cambios de nivel y

protección contra la influencia de factores ajenos a la experimentación), y no trata con detalle el análisis de los resultados de los experimentos.

Sin embargo, plantea que si los errores de ajuste al fijar las condiciones de los factores son insignificantes, el método de mínimos cuadrados ordinarios puede ser razonable, mientras que si los errores de ajuste son grandes, este tipo de análisis puede distorsionar de forma importante los resultados del análisis, ya que algunos efectos pueden ser declarados falsamente como significativos, mientras que otros realmente significativos no pueden ser identificados. Para disminuir estos errores recomienda asignar los factores que se considera que pueden presentar grandes errores de ajuste a las columnas del diseño que contienen un mayor número de cambios de nivel.

También indica que aunque el método de los mínimos cuadrados ordinarios no sea estrictamente válido para el análisis de los diseños con aleatorización restringida, puede servir para indicar la magnitud y dirección de los efectos y el experimentador debe hacer un balance entre las ganancias obtenidas al eliminar el efecto de una posible influencia lineal debida a factores desconocidos y también la de minimizar el número de cambios de nivel de los factores frente a la posible pérdida en que se puede incurrir al hacer un análisis inexacto.

### **Kim (1997)**

Kiho Kim (1997), en su tesis doctoral presentó una propuesta de análisis para diseños factoriales  $2^k$  libres de tendencia en efectos principales e interacciones dobles y también con pocos cambios de nivel, lo que genera una aleatorización restringida en algunos factores. Para resolver este problema propone el siguiente modelo mixto con efectos fijos y efectos aleatorios:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \sum_{i=1}^n \mathbf{Z}_i \mathbf{u}_i + \mathbf{e},$$

Donde,  $\mathbf{y}$  es el vector de respuestas;  $\mathbf{X}$  es la matriz de diseño del modelo;  $\boldsymbol{\beta}$  es el vector de los parámetros de efectos fijos;  $\mathbf{u}_i$  es el vector de errores de los tratamientos o vector de efectos aleatorios debido al  $i$ -ésimo factor con aleatorización restringida;  $\mathbf{Z}$  es la matriz de coeficientes de  $\mathbf{u}_i$ ; y  $\mathbf{e}$  es el vector de errores. Este modelo permite considerar la variabilidad debida a los factores con aleatorización restringida en el análisis, al estimar los errores en que se puede incurrir con esta restricción. De las condiciones dadas a este modelo es claro que la esperanza  $E(\mathbf{y})$  y la varianza ( $\mathbf{V}$ ) se pueden representar como:

$$E(\mathbf{y}) = \mathbf{X}\boldsymbol{\beta}$$

$$\mathbf{V} = \text{Var}(\mathbf{y}) = \sum_{i=1}^n \mathbf{Z}_i \mathbf{Z}_i' \sigma_i^2 + \sigma_e^2 \mathbf{I}_N.$$

Ahora el mejor estimador lineal insesgado para  $\beta$  es el estimador de los mínimos cuadrados generalizados (GLS), donde el estimador para  $\beta$  esta dado por

$$\hat{\beta} = (\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})^{-1} \mathbf{X}'\mathbf{V}^{-1}\mathbf{y}$$

Y la matriz de varianza-covarianza es

$$\text{Cov}(\hat{\beta}) = (\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})^{-1}.$$

El estimador para  $\beta$  es idéntico al estimador de mínimos cuadrados cuando  $\mathbf{V} = \sigma^2 \mathbf{I}$ , en otras palabras cuando no existen errores asociados a los ajustes los dos estimadores son iguales. Como los parámetros para los efectos aleatorios son desconocidos, para obtener los valores de la  $\hat{\beta}$  y  $\text{Cov}(\hat{\beta})$ , se deben estimar los parámetros de los efectos aleatorios a través del método estimación de máxima verosimilitud restringido (REML, *Restricted Maximum Likelihood Estimation*), el cual puede ser resuelto de forma computacional a través del procedimiento PROC MIXED de SAS.

Esta propuesta presenta el inconveniente de tener una alta complejidad, lo que la haría poco efectiva para fines de aplicación práctica por parte usuarios no expertos, aunque su grado de precisión es superior al análisis tradicional de mínimos cuadrados ordinarios y sus correspondientes pruebas de significación, ya que considera las restricciones en el reajuste de cada unos de los factores.

### **Ju y Lucas (2002)**

Este trabajo se centra en diseños  $2^k$  en los que se restringe la aleatorización de un solo factor del diseño. Para evaluar el efecto que puede tener esta restricción sobre la estimación de los coeficientes de regresión se construyen las matrices de covarianza esperada para estos diseños en varios escenarios: completamente aleatorizado, órdenes de ejecución aleatorizado con un factor difícil de cambiar no reajustado (RRO), completamente restringido, parcialmente restringido y el bloqueo *split plot*. La matriz de covarianza esperada ( $\mathbf{V}$ ) para un diseño factorial  $2^k$  con un factor difícil de ajustar puede ser representada por la siguiente ecuación.

$$\mathbf{V} = (\sigma_s^2 + (1-p)\sigma_w^2) \mathbf{I} + (p\sigma_w^2) \mathbf{U}_r \mathbf{U}_r'$$

Donde  $\sigma_s^2$  es la varianza del error asociado a los ajustes de los factores fáciles de cambiar,  $\sigma_w^2$  es la varianza del error asociado a los ajustes del factor difícil de cambiar,  $\mathbf{I}$  es la matriz identidad,  $\mathbf{U}_p$  representa la matriz de la estructura de bloqueo y  $p$  es la probabilidad de que un par de pruebas cualquiera tengan el mismo ajuste del factor difícil de cambiar en el mismo bloque, el cual depende del tipo de diseño que se está evaluando así: 1, para un diseño con aleatorización completamente restringida; 0, para un diseño completamente aleatorizado y  $2/2^{k-1} + 2$ , para un orden aleatorio con un factor no reajustado.

Como se observa en la descripción anterior el análisis de estos diseños requiere incluir en el modelo un término del error asociado con el factor que presenta la mayor dificultad de ajuste y otro término que represente el error de los factores restantes. Además si se quiere conocer con un mayor grado de detalle la forma correcta de analizar este tipo de diseños, es decir, a través de un modelo mixto, se recomienda ver el trabajo de Lorenzen y Anderson (1993).

Una de las principales conclusiones que se obtiene de este trabajo, es que la forma más adecuada de ejecutar y analizar un diseño con un factor difícil de ajustar no reiniciado después de cada prueba es el diseño RNR (*Randomized not Reset*) ya que este mejora al diseño completamente aleatorizado. Adicionalmente se comenta que un orden de experimentación aleatorio con un factor difícil de ajustar no reiniciado (RRO, *random run order with hard to change factor not reset*) o diseños RNR, pueden ser una estrategia razonable de ejecución para los experimentos en un mejoramiento de procesos porque el número de ajustes en los factores (costo) disminuye más rápidamente que lo que se incrementa la varianza [Weeb (1999)]. Vale la pena destacar que este trabajo se concentró principalmente en órdenes en los cuales un solo factor es difícil de ajustar (factor no reiniciado), pero esta situación no es la que se presenta con los órdenes obtenidos por el algoritmo de expansión, donde existen varios factores con restricción en la aleatorización, por lo cual no sería recomendable utilizar esta metodología de análisis en los órdenes obtenidos por el algoritmo de expansión si no se generaliza esta propuesta.

### **Webb et al. (2004)**

Tal vez, el trabajo que mejor coincide con los propósitos de los órdenes obtenidos por nuestro algoritmo de expansión es el presentado por Webb et al. (2004), ya que permite definir la forma más práctica de enfrentar el análisis en los diseños factoriales con aleatorización restringida en múltiples factores. Esta propuesta generaliza los trabajos de Ju y Lucas (2002), con lo cual es posible mejorar el entendimiento de las propiedades de la estimación de la varianza en los diseños factoriales que contienen uno

o más factores no reajustados, para compararlo con la de un diseño completamente aleatorizado con los factores reajustados después de cada prueba, y a partir de esta comparación definir cual de las dos formas de experimentación sería más razonable tanto desde el punto de vista estadístico como industrial.

El trabajo parte de la idea de que en la práctica industrial la mayoría de la experimentación no se realiza de forma completamente aleatoria, así lo recomienda el plan experimental, debido a que uno o varios factores no pueden ser reajustados después de cada prueba experimental, ya sea por motivos económicos o restricciones del proceso, y a pesar de ello su análisis se realiza de forma tradicional como si la experimentación se hubiera realizado de forma aleatoria.

Sabemos que cuando los experimentadores no reconocen la presencia de factores que no han sido reiniciados después de cada prueba experimental se asume el modelo de efectos fijos  $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ , donde  $\mathbf{y}$  es el vector de respuestas,  $\mathbf{X}$  es la matriz del modelo,  $\boldsymbol{\beta}$  es el vector de los parámetros de efectos fijos y  $\boldsymbol{\varepsilon}$  es el vector de errores aleatorios. En estas circunstancias el estimador de mínimos cuadrados ordinarios  $\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1} \mathbf{X}'\mathbf{y}$ , es usado para estimar el vector de parámetros  $\boldsymbol{\beta}$ .

Este modelo falla al no considerar las posibles correlaciones presentes entre las filas adyacentes inducidas por no reajustar los factores. Para considerar las restricciones en la aleatorización en un experimento con  $c$  factores que no son reajustados, el modelo más indicado sería, el modelo mixto  $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{u} + \boldsymbol{\varepsilon}$ , donde,  $\mathbf{y}$ , es el vector de respuestas,  $\mathbf{X}$  es la matriz del modelo,  $\boldsymbol{\beta}$  es el vector de los parámetros de efectos fijos;  $\mathbf{Z} = (\mathbf{Z}_{x_1} | \dots | \mathbf{Z}_{x_c})$  es la matriz de diseño de efectos aleatorios de los  $c$  factores no reajustados;  $\mathbf{u}$  es el vector de efectos aleatorios de los factores con restricción en la aleatorización y  $\boldsymbol{\varepsilon}$  es el vector de errores aleatorios. Ejemplos de cómo se construyen las matrices  $\mathbf{Z}$ , pueden ser visto en Ju y Lucas (2002), además para una minuciosa discusión del análisis de los modelos mixtos ver Goos *et al.* (2002).

También se procedió a realizar las comparaciones para los diseños factoriales  $2^k$  con  $k = 2$  hasta  $k = 7$ , de lo cual concluyen que a efectos prácticos la diferencia en la estimación de la varianza esperada que se obtiene al considerar o no los ajustes de los factores, es muy pequeña, es decir, que en los diseños en los cuales los factores no reajustados es una forma practica de conducir experimentos, en especial cuando algunos factores son difíciles de reajustar o se incurren en grandes costos al hacer los ajustes. Además el método de mínimos cuadrados ordinarios es un método razonable y sencillo para el análisis de diseños factoriales  $2^k$  cuando se restringe la aleatorización de

múltiples factores en la ejecución del experimento, ya que permite detectar efectos significativos en los efectos.

Los autores concluyen que los experimentos en los cuales se restringe la aleatorización de más de un factor son razonables para el mejoramiento de procesos, ya que, el número de ajustes (costo) disminuye más rápido de lo que la varianza aumenta, aunque no para experimentos científicos en los cuales sea muy importante asegurar la validez de las pruebas de significación.

### **Ganju y Lucas (2004)**

Discuten en este trabajo como en los diferentes libros de texto y artículos desarrollados tanto para los ambientes universitarios como para la industria se distorsiona el concepto de aleatorización, ya que por aleatorizar se entiende como obtener un plan de experimentación en el cual el orden de las corridas es obtenido de forma aleatoria. Pero aleatorizar también requiere independizar los ajustes de los niveles de los factores cuando los niveles de la prueba anterior son los mismos, es decir, reajustar los valores de los factores después de cada prueba. Además insisten en que en la primera situación (aleatorizar los órdenes de experimentación) se hace mucho énfasis en los libros de texto y publicaciones, más la segunda situación (reajustar los valores después de cada prueba) permanece ausente.

Sin embargo, los autores reconocen que en la práctica industrial existen situaciones en las cuales no es posible realizar un reajuste de los niveles de los factores después de cada prueba experimental debido a dificultades físicas del proceso, por lo cual recomiendan como un método razonable para realizar la experimentación, los órdenes aleatorizados con múltiples factores no reajustados para el mejoramiento de procesos, propuestos por Webb et al. (2004), ya que estos órdenes permiten obtener grandes ahorros con respecto a la precisión que se pierde en el análisis realizado con el método de mínimos cuadrados.

Adicionalmente, los autores recomiendan que dada la importancia que tiene conocer la forma en que fueron realizados los experimentos para mejorar su análisis, se debería fomentar el que las revistas especializadas solicitaran a los autores de artículos que informen con el máximo detalle sobre como fueron realizados los experimentos, especialmente en los que concierne a como fue obtenido el orden de realización y si realmente se partió de cero después de cada corrida.

### **Bisgaard y Pinho (2004)**

Presentan un ejemplo sobre la cocción de pasteles en un horno de túnel, en el que se realiza un diseño  $2^3$  con los factores: cantidad de azúcar (A), la cantidad de harina (B), y la velocidad de banda transportadora (C). Los factores A y B son difíciles de reajustar por razones económicas mientras que el factor C es fácil de reajustar ya que sólo precisa pulsar un botón. Bisgaard y Pinho (2004), demuestran a partir de la generalización de este ejemplo como los diseños  $2^k$  con factores no reajustados (diseños *split plot*) pueden ser analizados usando dos graficas de probabilidad normal, una para el grupo de factores no reajustados o difíciles de reajustar en la experimentación (factores *whole plot*) y otra para los factores que fueron reajustados o factores fáciles de reajustar (factores *subplot*). Esto debido a que los efectos caen en dos grupos con diferentes varianzas. Una varianza para los factores *whole plot* o no reajustados y sus interacciones con un valor de:

$$\frac{4}{2^k} (2^{k-p} \sigma_1^2 + \sigma_0^2),$$

Donde  $k$ , es el número de factores del diseño,  $p$  es el número de factores no reajustados,  $\sigma_1^2$  es la varianza entre los factores no reajustados (factores *whole plot*) y  $\sigma_0^2$ , es la varianza entre los factores reajustados o factores *subplot*. La varianza para los factores *subplot* y sus interacciones dada por

$$\frac{4}{2^k} \sigma_0^2,$$

Ya que independientemente del número de factores no reajustados que existan en los diseños factoriales  $2^k$  las varianzas de los efectos caen en los grupos con las varianzas descritas previamente, una forma de análisis adecuado y simple es usar dos graficas de probabilidad normal para obtener resultados confiables. Una grafica para los factores que no han sido reajustados y otra para los que se han reajustado.

Varios ejemplos de aplicación en la industria de los diseños factoriales que presentan factores difíciles de reajustar (diseños *split plot*) pueden verse en Bisgaard (2000) y Kowalski y Potcner (2003).

### **Simpson et al. (2004)**

La mayoría de propuestas formales recomiendan el método de los mínimos cuadrados generalizados para analizar los resultados de diseños  $2^k$  que presentan restricciones en

la aleatorización. Este análisis puede ser desarrollado de forma automatizada con procedimientos del paquete de software estadístico SAS, pero en la práctica este programa no se encuentra disponible para los usuarios no expertos. Los autores desarrollaron un método que permite realizar este análisis a través del uso del programa MINITAB en dos etapas para estimar apropiadamente todos los términos del modelo.

### **Goos *et al.* (2006)**

Para verificar los resultados presentados por Webb *et al.* (2004), Goos *et al.* (2006), realizaron un estudio de simulación comparando la eficiencia de dos métodos de análisis, el método de mínimos cuadrados ordinarios (OLS) contra el método de estimación de mínimos cuadrados generalizados (GLS) cuando se tiene un diseño *split plot* en el cual el orden de experimentación presenta el mismo nivel en varias pruebas consecutivas para uno o más factores bajo investigación, ya sea que en la ejecución del experimentos se reinicien o no los niveles de los factores después de cada prueba de experimentación.

Para llevar a cabo el estudio usaron diseños experimentales pequeños (factorial, central compuesto y D-óptimo con 16 corridas) y se centraron en los dos aspectos clave para tomar una decisión estadística. Primero, se investigó cómo la probabilidad de encontrar efectos significativos depende del método usado. Para este propósito desarrollaron un estudio de simulación con el método OLS y cuatro métodos diferentes de GLS (método de contención, residual, Satterthwaite y Kenward y Roger). Los cuatro métodos GLS, están disponibles en SAS 8.02 y difieren en la forma en que se calculan los grados de libertad y el test estadístico usado. Segundo, se investigó si se obtienen diferentes respuestas al usar el OLS ó GLS en relación a las condiciones óptimas para las variables experimentales en una superficie de respuesta.

Un resultado importante de este trabajo, es que en la medida en que disminuya el número de ajustes en los factores difíciles de cambiar, mayor será la imprecisión en la estimación de los efectos de estos factores, aunque si el efecto es grande su detección es posible. Este resultado también fue confirmado analíticamente por Goos y Vandebroek (2004), cuando afirman que al permitir un aumento en el número de cambios de nivel de los factores difíciles de ajustar (factores *whole plot*) aumenta la eficiencia de diseño *split plot* en términos del criterio D-óptimo, reduce la pérdida de precisión en la estimación de los coeficientes de los factores difíciles de ajustar, permite más grados de libertad para la estimación de los errores de los factores *whole plot* y finalmente reducen el riesgo de errores sistemáticos y tendencias en los efectos que podrían distorsionar los resultados del experimento.

Otro resultado del estudio es que no hay una enorme diferencia entre las condiciones óptimas de operación de los niveles de los factores obtenidos al analizar el modelo con el método OLS y uno estimado por método GLS. Especialmente en el caso en que las observaciones del experimento *split plot* están poco correlacionadas.

En conclusión, puede decirse que a pesar de que los diseños en los cuales los factores no se reajustan después de cada prueba experimental pueden presentar algunos errores de precisión en la detección de efectos significativos, en general las condiciones óptimas de operación de los factores que intervienen en el experimento son detectadas por el modelo OLS, lo que confirma el éxito que se presenta en la industria al desarrollar procesos de mejoramiento con órdenes de experimentación en los cuales no se reajustan los niveles de los factores después de cada prueba y posteriormente son analizados de la forma tradicional. Lo que coincide plenamente con lo que había identificado Webb *et al.* (2004).



# Referencias



# Referencias

Addelman, S., (1965) "The Construction of  $2^{17-9}$  Resolution V Plan in Eight Blocks of 32" *Technometrics*. Vol. 7.

Anderson y Mclean (1974) "Design of Experiments: A realistic Approach". *Marcel Dekker, Inc.* New York.

Arvidsson M. y Gremyr I. (2003) "Deliberate Choices of Restrictions in Complete Randomization" *Quality and Reliability Engineering International*. Vol 19.

Atkinson, A. y Donev, A. (1996) "Experimental Designs Optimally Balanced For Trend" *Technometrics*. Vol. 38, No.4.

Atkinson, A. y Donev, A. (1996)a. "Optimum Experimental Designs" *Oxford University Press*. New York.

Bailey (1983)."Restricted Randomization" *Biometrika*. Vol. 70, No. 1.

Bailey, R.A., Cheng C.S., Kipnis P. (1992) "Construction of Trend-Resistant Factorial Designs" *Statistica Sinica*. No. 2.

Bingham, D.R. y Sitter, R.R. (2001). "Design Issues in Fractional Factorial Split Plot Experiments" *The Journal of Quality Technology*. Vol. 33.

Bingham, D.R. y Sitter, R.R. (2003)."Fractional Factorial Split Plot Designs for Robust Parameter Experiments". *Technometrics*. Vol. 45, No. 1.

Bisgaard, S.; Fuller, H. T. y Barrios E. (1995) "Two Level Factorials Runs as Split Plot Experiments". *Quality Engineering*. Vol. 8, No. 4.

Bisgaard, S. y Steinberg (1997) "The Design and Analysis of  $2^{k-p} \times s$  Prototype Experiments". *Technometrics*. Vol.39, No. 1.

Bisgaard, S, (2000). "The Design and Analysis of  $2^{k-p} \times 2^{q-r}$  Split Plot Experiments". *Journal of Quality Technology*. Vol. 32, No. 1.

Bisgaard, S. y De Pinho A. (2004) "The Error Structure of Split Plot Experiments" *Quality Engineering*. Vol. 16. No. 4.

Borowski, J. y Lucas, J. (1997) "Designs of Mixed Resolution for Process Robustness Studies". *Technometrics*. Vol. 39, No.1.

Box, G.E.P. (1952) "Multifactor Design of First Order". *Biometrika*, Vol. 39.

Box, G. E. P. and Hay, W. A. (1953). "A Statistical Design for the Removal of Trends Occurring in a Comparative Experiment with an Application in Biological Assay". *Biometrics*, Vol. 9, No. 3.

Box G.E.P. y Hunter. J. S (1961) "The  $2^{k-p}$  Fractional Factorial Designs" *Technometrics*. Vol. 3.

Box G.E.P y Guttman I. (1966) "Some Aspects of Randomization". *Journal Royal of the Statistical Society*. No. 3.

Box G.E.P. y Jones S. (1990) "Designing Products that are Robust to the Environment". *Technical Report CQPI*. No. 56.

Box G.E.P. y Jones S. (1992) "Split Plot Designs for Robust Parameters Design Approach". *Journal of Applied Statistics*. Vol. 19.

Box G.E.P; Hunter G. W. y Hunter J. S. (2005) "Statistics for Experimenters: Design, Innovation, and Discovery" Second edition. *John Wiley & sons*. New Jersey, USA.

Chen M. H. and Wang P. C. (2001) "Multi-Level Factorial Designs with Minimum Numbers of Level Changes". *Communications in Statistics: Theory and Methods* Vol. 30, No. 5

Cheng, C. S. (1985). "Run Orders of Factorial Designs. In *Proceedings of the Berkeley Conference in Honor of Jerzy Neyman and Jack Kiefer* (L. M. Le Cam and R. A. Olshen, eds.)". Vol. 2. Wadsworth, Monterrey, California.

Cheng, C. S. y Jacroux M. (1988) "The Construction of Trend-Free Run Orders of Two-Level Factorial Designs". *Journal of American Statistical Association*. Vol. 83, No. 404.

Cheng, C. S. (1990)," Construction of Run Orders of Factorial Designs, in *Statistical Design and Analysis of Industrial Experiments*". *Marcel Dekker*. New York.

Cheng, C. S. y Steinberg, D. M. (1991) "Trend Robust Two Level Factorial Designs". *Biometrika*. Vol. 78, No.2.

Cheng C. S. (1995) "Some Projection Properties of Orthogonal Arrays". *The Annals of Statistics*, Vol. 23, No.4.

Cheng C. S.; Martin R. J. y Tang B. (1998) "Two-Level Factorial Designs With Extreme Numbers of Level Changes" *The Annals of Statistics*. Vol. 26, No. 4.

Coster, D. y Cheng Ch. (1988) "Minimum Cost Trend-free Run Orders of Fractional Factorial Designs" *The Annals of Statistics*. Vol. 16, No. 3.

Coster, D.C. (1993). "Tables of Minimum Cost, Linear Trend-Free Run Sequences for Two and Three-Level Fractional Factorial Design". *Computational Statistics and Data Analysis*, Vol. 16.

Coster, D. C. (1993). "Trend-free Run Orders of Mixed-Level Fractional Factorial Designs". *The Annals of Statistics*, Vol. 21.

Cox, D.R. (1952) "Some Recent Work on Systematic Experimental Design". *Journal of Royal Statistical Society*. Vol. 14, No. 2.

Daniel, C. y Wilcoxon, F. (1966) "Factorial  $2^{p-q}$  Plans Robust against Linear and Quadratic Trends" *Technometrics*. Vol. 8, No. 2.

De León, G. (2004) “Análisis y Propuestas Sobre Algunos Aspectos de la Aplicación del Diseño de Experimentos en la Industria” *Tesis doctoral Universidad Politécnica de Cataluña*.

De León G., Grima P. y Tort-Martorell X. (2005) “Experimentation Order in Factorial Designs with 8 or 16 Runs”. *Journal of Applied Statistics*. Vol. 32, No. 3

Dickinson, A. W. (1974) “Some Orders Requiring a Minimum Number of Factor Level Changes for  $2^4$  and  $2^5$  Main Effects Plans” *Technometrics*. Vol. 16, No. 1.

Draper, N y Stoneman, D. (1968) “Factor Changes and Linear Trends in Eight-Run Two-Level Factorial Designs”. *Technometrics*. Vol. 10, No. 2.

Draper, N. y Smith h. (1966) “Applied Regression Analysis”. John Wiley & Sons. New York.

Ganju, J. y Lucas, J. M. (1997) “Bias in Test Statistics When Restrictions on Randomization are Cause by Factors”. *Communications in Statistics Theory and Methods*. Vol. 26.

Ganju, J. y Lucas, J. M. (1999) “Detecting Randomization Restrictions Caused by Factors”. *Journal of Statistical Planning and Inference*. Vol. 81.

Ganju, J. y Lucas, J. M. (2000) “Analysis of Unbalanced Data from an Experiment with Block Effects and Unequally Space Factor Levels” *The American Statistician*. Vol. 54. No. 1.

Ganju, J. y Lucas, J. M. (2004) “Randomized and Random Run Orders Experiments” *Journal of Statistical Planning and Inference*. Vol. 133.

Goos, P. y Vandebroek M. (2001) “Optimal Split Plot Designs”. *Journal of Quality Technology*. Vol 33.

Goos, P. (2002) “The Optimal Design of blocked and Split Plot Experiments” Springer. Berlin.

Goos, P. y Vandebroek M. (2004) “Outperforming Completely Randomized Designs”. *Journal of Quality Technology*. Vol. 36. No. 1.

Goos, P., Langhans I. y Vandebroek M. (2005) "Identifying Effects under a Split Plot Design Structure". *Journal of Chemometrics*. Vol. 19.

Goos, P., Langhans I. y Vandebroek M. (2006) "Practical Inference from Industrial Split Plot". *Journal of Quality Technology*. Vol. 38, No.2

Grima, P.; Marco Ll. y Tort-Martorell X. (2004) "Estadística Práctica con Minitab" *Pearson-Prentice Hall*. Madrid.

Hahn G. J. (1984) "Experimental Design in the Complex World" *Technometrics*. Vol. 26. No. 1.

Harville, D. A. (1975) "Experimental Randomization: Who Needs It?" *The American Statistician*, Vol. 29, No. 1.

Hill, H.M. (1960) "Experimental Designs to Adjust for Time Trend". *Technometrics*. Vol. 2. No. 1.

Hinkelmann K. y Kempthorne O. (1994). "Design and Analysis of Experiments" John Wiley and Sons, Inc. New York.

Jacroux M. (1990) y Ray R.S., "On Construction of Trend-free Run Orders of Treatments" *Biometrika*. Vol. 77.

Jacroux M. (1994) "On the Construction of Trend Resistant Mixed Level Factorial Run Orders". *The Annals of Statistics*, Vol. 22, No. 2.

Jacroux M., Majumdar D. y Shah K. (1997) "On the Determination and Construction of Optimal Block Designs in Presence of Linear Trends". *Journal of the American Statistical Association*. Vol. 92, No. 437

John, P. W. (1990) "Time Trends and Factorial Experiments" *Technometrics*. Vol. 32, No. 3.

Joiner, B. y Campbell C. (1976) "Designing Experiments When Run Order is Important". *Technometrics*. Vol. 18, No. 3.

Ju, H. y Lucas, J. (2002) “ $L^k$  Factorial Experiment with Hard to Change and Easy to Change Factors”. *Journal of Quality Technology*. Vol. 34, No. 4.

Kiefer J. (1959) “Optimal Experimental Designs” *Journal of the Royal Statistical Society*. Vol. 21, No.2.

Kiefer J. y Wolfowitz (1960) “The Equivalence of Two Extremum Problem” *Canadian Journal of Mathematics*. Vol. 12.

Kim, K. (1997) “Construction and Analysis of Linear Trend-Free Factorial Designs Under a General Cost Structure” *Ph. D. Dissertation at Virginia Polytechnic Institute and State University*.

Kowalski S. M. (2002) “ $2^4$  Run Split Plot Experiments for Robust Parameter Design”. *Journal of Quality Technology*, Vol. 34, No. 4.

Kowalski, S.M.; Cornell, J.A. y Vining, G.G. (2002) “Split Plot Designs and Estimation Methods for Mixture Experiments with Process Variables” *Technometrics*. Vol. 44.

Kowalski S. M. y Potcner K. J. (2003) “How to Recognize a Split Plot Experiment”. *Quality Progress*. November.

Kracker, R.N. y Tsui, K. (1990) “Interaction Graphs: Graphical Aids for Planning Experiments” *Journal of Quality Technology*, Vol. 22, No. 1.

Kulahci M.; Ramírez J. y Tobias R. (2006) “Split Plot Fractional Designs: Is Minimum Aberration Enough?” *Journal of Quality Technology*, Vol. 38, No. 1.

Li W. y Lin D. K. (2003) “Optimal Foldover Plans for Two Level Fractional Factorial Designs” *Technometrics*. Vol.45, No.2.

Letsinger, J.D; Myers, R. H. y Lentner M. (1996) “Response Surface Methods for Bi-randomization Structures” *Journal of Quality Technology*, Vol. 28, No. 4.

Lonrenzen T. J. y Anderson V.L. (1993) “Design of Experiments: A No Name Approach” New York, NY: *Marcel Dekker, Inc.*

Lucas J. M. (2003) "New Results about Randomization and Split-Plotting" Quality & Productivity Research Conference. New York.

Martin, R.J.; Jones, G. y Eccleston, J.A. (1998) "Some Results on Two-level Factorial Designs with Dependent observations" *Journal of statistical Planning Inference*. Vol. 66.

Mason R. L., Gunst, R.F. y Hess J.L. (2003) "Statistical Design and Analysis of Experiments with Applications to Engineering and Science". Second edition. Wiley, New York.

McLean, R. A. y Anderson, V.L. (1974). "Design of Experiments a Realistic Approach." New York, NY: Marcel Dekker, Inc.

McLeod R. G. y Brewster J.F. (2004) "The Design of Blocked Fractional Factorial Split Plot Experiments" *Technometrics*. Vol. 46, No. 2.

Miller A. (1997) "Strip Plot Configurations of Fractional Factorials" *Technometrics*. Vol. 39, No. 2.

Montgomery D. C. (2005) "Design and Analysis of Experiments" 6<sup>th</sup> Edition, *John Wiley and Sons*.

Mood, A. M. (1940) "The Distribution Theory of Runs". *Annals of Mathematical Statistics*. Vol. 11.

Nelson, L.S. (1985) "What do Low F Ratios Tell you?" *Journal of Quality Technology*. Vol. 17.

Neter, J.; Kutner M. H.; Nachtsheim C. J. y Wasserman W. (1996) "Applied Linear Statistical Models" *McGraw Hill*. U.S.A.

Prat, A.; Tort-Martorell, X.; Grima, P. y Pozueta, L. (1997) "Métodos Estadísticos: Control y Mejora de la Calidad" Primera Edición. Edicions UPC. Barcelona

Rao C. R. (1947) "Factorial Experiments Derivable from Combinatorial Arrangements of Arrays". *Journal of the Royal Statistical Society*. Vol. 9, No.1.

Tack L. y Vandebroek M. (2002) "Trend Resistant and Cost-efficient Block Designs with Fixed or Random Block Effects". *Journal of Quality Technology*. Vol .34, No.4.

Tiahrt, K.J. y Weeks, P.L. (1970). "A Method of Constrained Randomization for  $2^k$  Factorial". *Technometrics*. Vol. 12.

Trinca, L.A. y Gilmour S.G. (2001), "Multistratum Response Surface Designs" *Technometrics*. Vol. 43, No. 1.

Schott, J. R. (1997) "Matrix Analysis for Statistics". *John Wiley and Sons*. New York.

Searle, S. R. (1982) "Matrix Algebra Useful for Statistics". *John Wiley and Sons*. New York.

Searle, S.R.; Casella, G.; y McCulloch, C.E. (1992) "Variance Components". *John Wiley & Sons*, New York.

Simpson J. R., Kowalski S.M. y Landman D. (2004) "Experimentation with Randomization Restrictions: Targeting Practical Implementation" *Quality and Reliability Engineering International*. Vol. 20.

Steinberg, D. (1988). "Factorial Experiments with Time Trends". *Technometrics*. Vol. 30. No. 3.

Wang P.C. (1991) "Symbol Changes and Trend Resistance in Orthogonal Plans of Symmetrical Factorial". *Sankhya: The Indian Journal of Statistics*. Vol. 53. Series B.

Wang J. C. (1992) "Nearly Orthogonal Arrays with Mixed Levels and Small Runs". *Technometrics*. Vol.34, No. 4.

Wang, P.C. y Jan, H. W. (1995). "Designing Two-Level Factorial Experiments Using Orthogonal Arrays When The Run Order Is Important". *The Statistician*. Vol. 44, No. 3.

Wang, P. C. (1996) "Level Changes and Trend Resistance in  $L_N(2^p4^q)$  Orthogonal Arrays" *Statistica Sinica* Vol. 6.

Wang, P. C. y Chen, M.H. (1998) "Level Changes and Trend Resistance on Replacement in Asymmetric Orthogonal Arrays". *Journal of Statistical Planning and Inference*. Vol. 69.

Webb, D. (1999) "Randomization Restrictions and the Inadvertent Split Plot in Industrial Experimentation". *Ph. D. Dissertation, Montana State University*.

Webb, D. F.; Lucas, J. M. y Borkowski, J. J. (2004) "Factorial Experiments When Factor Levels Are Not Necessarily Reset" *Journal of Quality Technology*. Vol. 36. No. 1.

Wooding, W.M. (1973) "The Split Plot Design" *Journal of Quality Technology*. Vol. 5

Wu C. F. y Hamada M. (2000) "Experiments Planning, Analysis, and Parameter Design Optimization". John Wiley and Sons. New York.

Wynn H.P. (1970), "The Sequential Generation of D-optimal Designs" *Annals of Mathematical Statistics*. Vol. 41.

Wynn H.P. (1972), "Results in the Theory and Construction of D- Optimum Experiments Designs" *Journal of the Royal Statistical Society*. Vol. 34

Yeh, C.M., Bradley, R. A. y Notz W.I. (1985) "Nearly Trend Free-Block Design" *Journal of the American Statistical Association*. Vol. 80.

Youden W.J. (1956) "Randomization and Experimentation" *Annals of Mathematical Statistics*. Vol. 27.