

UNIVERSIDAD POLITÉCNICA DE VALENCIA

DEPARTAMENTO DE ORGANIZACIÓN DE EMPRESAS



Problema de Programación de Lotes Cíclicos con Tiempos de Preparación Dependientes e Independientes de la Secuencia con Aplicaciones a Sistemas de Pintura.

TESIS DOCTORAL

PRESENTADA POR:
D. JORGE ARTURO MARTINEZ ORTIZ

DIRIGIDA POR:
DR. D. JOSE PEDRO GARCIA SABATER
DR. D. CARLOS ANDRES ROMANO

VALENCIA, DICIEMBRE 2015

A MIS PADRES Y HERMANOS
A MARIA ELENA

Agradecimientos:

Esta tesis doctoral ha sido el fruto de muchos años de trabajo. El resultado que se presenta ha implicado un esfuerzo continuo tanto en Valencia como en Aguascalientes. Además de los retos intelectuales y académicos ha requerido una gestión constante de aspectos administrativos y logísticos para poder llegar a la meta. El trabajo que se presenta no se habría podido realizar sin el apoyo y ayuda desinteresada de muchas personas que siempre han mantenido la confianza en el resultado final.

Quiero agradecer profunda y sinceramente a mis tutores Dr. José Pedro García Sabater y Dr. Carlos Andrés Romano por el continuo seguimiento, revisiones y consejos. Además de la paciencia y la confianza en que llevaríamos a feliz término este proyecto valoro la amistad y las palabras oportunas a lo largo de todos estos años.

En mis padres, hermanos y amigos cercanos tuve siempre un voto de confianza y alegría para continuar en los momentos difíciles. La memoria de mis abuelos, que vivieron épocas complicadas y que siempre continuaron luchando por conseguir sus metas, han sido mi fuente de inspiración.

Es seguro que sin el apoyo de mis jefes, compañeros y amigos en la Universidad Politécnica de Aguascalientes no hubiese podido lograr mi objetivo de terminar el doctorado. Les doy las gracias por su motivación y ayuda tanto en el ámbito académico como en todos los aspectos logísticos, administrativos y financieros. Especialmente al Mtro. Eulogio Monreal Avila, Rector de la UPA, por su apoyo incondicional.

En los momentos difíciles siempre conté con una red de apoyo de compañeros y amigos. Muchos en España y muchos en México, a ellos también quiero darles las gracias. Han sido muchas las personas que me han ayudado y me sería difícil nombrarlas a todas pero quiero que sepan que estoy muy agradecido y en deuda con todos. En forma especial a Joan Morant Llorca, Francisco Gómez, Rodolfo Reyes Aguillón y Gema Tavares Sánchez_Monge su ayuda invaluable en el proceso informático. A compañeros y colegas en el DOE de la UPV: Cristina Figueroa, Julio García, Cristóbal Miralles, David Pérez, Faustino Alarcón, Thomas Dasaklis, Jairo Coronado. En la UPA al Dr. Marco Paz, Humberto Sosa, Rosy Ramírez, Toño Vázquez, José Luis Gallegos, José Luis Serafín, Rodolfo Medina, Esteban Ogazón. En el Tec de Monterrey al CP. David Noel Ramírez, Mtro. Joaquín Guerra. A colegas y amigos como el Dr. Ricardo Valdez, Gaby López, Leonardo Jiménez, María José Puentes, a todos les doy gracias por su confianza.

En momentos de dificultades o incertidumbre además de a mi familia recibí siempre el apoyo de grandes amigos, personas queridas a quienes agradezco sus palabras oportunas y consejos especialmente a María Elena Vivanco Villarello, Salvador Lupercio Lúevano, Julia López Reyes, Verónica Skinfil y Dietmar Pascher.

Finalmente quiero agradecer a los revisores y miembros del tribunal su dedicación, esfuerzo y sobre todo, sus comentarios.

Valencia, a 4 de diciembre de 2015

RESUMEN

Problema de Programación de Lotes Cíclicos con Tiempos de Preparación Dependientes e Independientes de la Secuencia con Aplicaciones a Sistemas de Pintura.

En el mundo del siglo XXI, en un entorno en general globalizado en el que los métodos y procedimientos aplicados a los procesos industriales son estandarizados y más que probados aún es posible intentar descubrir formas innovadoras de mejorar procesos ampliamente estudiados dentro de las técnicas clásicas de Organización Industrial. Definir una manera eficiente de secuenciar una serie de productos en un proceso productivo impacta en forma relevante en la productividad y en los costes de fabricación.

En los procesos de pintura de las empresas de fabricación de componentes para la industria automotriz se hace la secuenciación de productos con base al programa maestro que indican los clientes. El programa de fabricación se realiza siguiendo los requerimientos de modelos y colores que es necesario entregar. En las fábricas con estos procesos la información sobre las especificaciones de los productos se suele recibir con una ventana de tiempo muy corta, en algunos casos de tan solo seis horas. Esto requiere una flexibilidad de respuesta muy alta. Esta flexibilidad se puede conseguir de diferentes formas, algunas de estas son mediante cambios rápidos de color y de modelo en los procesos de pintura, mediante inventarios de producto terminado que permiten surtir el producto requerido o mediante mejoras en la manera de secuenciar los productos. Esta tesis analiza y busca mejorar los procedimientos de secuenciación para minimizar los cambios que generan un *setup* en los procesos de pintura.

La generación de *setups* en una planta de pintura de componentes para automóvil puede ocurrir por dos situaciones: 1) cambios en el modelo del producto y 2) cambios en el color utilizado para pintar. Estos casos ocurren de forma independiente entre sí, es decir en un momento específico puede ocurrir un caso u otro o ambos al mismo tiempo. Se ha detectado que las técnicas de secuenciación existentes no contemplan la situación de optimizar un proceso productivo de pintura minimizando el número de cambios que se generan por las dos situaciones presentadas.

La motivación principal en esta tesis es desarrollar procedimientos de solución al problema de secuenciación bi_objetivo en los procesos de pintura dado que las técnicas existentes para minimizar el número de *setups* no contemplan este caso. La aplicación de las técnicas clásicas resulta insuficiente para lograr eficiencia al generar una la mejor secuencia.

Se parte del caso de algunas empresas específicas para delimitar el problema en una serie de hipótesis de trabajo. Se hace una revisión de las referencias sobre los temas afines que incluyen la secuenciación en procesos con circuito cerrado, la programación cíclica y la optimización de *setups*. Dadas las hipótesis de trabajo y la literatura existente se confirma la inexistencia de modelos, métodos y procedimientos en este entorno a excepción de dos referencias. Las hipótesis iniciales fueron ampliadas para cambiar el enfoque del problema: no solamente se aborda el caso básico de minimizar el número de cambios de modelo y de color sino que se analiza un segundo problema: minimizar el número de espacios vacíos que ocurren cuando hay un *setup* sea por cambio de modelo o de color. En ambos casos se diseñaron nuevos modelos matemáticos que son explicados ampliamente. Se hace una aportación muy importante en la forma de contar los espacios vacíos con horizonte incierto en el modelo con huecos que se explica a profundidad.

Dado que ambos problemas son *NP_hard* y se exploran mediante software de optimización, rápidamente crecen en el tiempo de resolución. Por esta razón se justifica el desarrollo de distintos procedimientos heurísticos que proporcionan una buena solución computacional en un tiempo razonable. Para todos estos procedimientos se realiza un código de programación que permita simular cada caso. En el problema sin huecos se generaron 27 procedimientos heurísticos. En el caso con huecos fueron 19 procedimientos. Para ambos problemas se presentan conclusiones sobre las heurísticas más eficientes justificándolas tanto desde el punto de vista estadístico como de análisis comparativo con el óptimo teórico.

Finalmente se presentan conclusiones generales donde se indican líneas futuras de investigación, en las que se pueden ampliar los problemas presentados y la posibilidad de extender el enfoque presentado a líneas de fabricación bi_objetivo, no necesariamente de sistemas de pintura.

ABSTRACT

Lots scheduling problem with setup times Cyclic dependent and independent of the sequence Paint Systems Applications.

In this XXI century world, in a generally globalized environment, where the methods and procedures applied to industrial processes are standardized and tested, it is still possible to find innovative ways to improve processes extensively studied within the classical techniques of Industrial Organization. To define an efficient way to sequence a series of products in a production process affects in a relevant form, productivity and manufacturing costs.

In the painting process of manufacturing companies of auto parts for the automotive industry the schedule and sequencing of products is done based on the customer master schedule. The manufacturing program is carried out following the requirements of models and colors that are requested and need to be delivered. In factories with these processes the information on the specifications of the products is often met with a very short window of time, in some cases as little as six hours. This requires a very flexible response. This flexibility can be achieved in different ways, some of these are by rapid changes of color and pattern in the process of painting, finished product inventories which help in the right supply of required products or by improvements in the way of sequencing products. This thesis analyzes and seeks to improve the sequencing procedures to minimize changes that generate a *setup* in painting processes.

The generation of setups in an auto parts painting plant may occur by two situations: 1) changes in the product model and 2) changes in the color used to paint. These cases occur independently from each other, in a specific time it is possible that either one case can occur or both simultaneously. It has been found that existing sequencing techniques do not contemplate the situation to optimize a production process of painting minimizing the number of changes that are generated by the two situations presented.

The main motivation in this thesis is to develop procedures for solving the problem of bi-objective sequencing painting processes because existing techniques to minimize the number of setups do not consider this case. The application of conventional techniques is not sufficient to achieve efficiency at generating the best sequence.

We part from the case of some specific number of companies to define the problem in working hypothesis. A review of the references included in related topics like “Sequencing in closed circuit processes”, “cyclical scheduling” and “Optimization of setups” is done. Having given the hypothesis and reviewed the literature, the lack of models, methods and procedures in this environment is confirmed, with the exception of two references. The initial hypotheses were expanded to a wider focus of the problem: not only the basic case which looks for minimizing the number of changes of pattern and color, but a second problem is analyzed: to minimize the number of gaps that are generated when a setup occur either by changing color or pattern. In both cases, new mathematical models were designed and widely explained. In the model with gaps a very important contribution was done: the form of counting the empty spaces with uncertain horizon. It is explained in depth.

Since both problems are *NP_hard*, if they are explored through optimization software, they quickly grow in resolution time. For this reason it is necessary the development of different heuristic procedures that provide a good solution where a reasonable computational time is warranted. For all of these processes it has been made a program code that simulates each case. In the problem without gaps 27 heuristic procedures were generated. For the problem with gaps, 19 procedures have been designed. For both problems conclusions were presented and the most efficient heuristics were justified by both points of view: statistically and analytically. A comparison with the theoretical optimum was made.

Finally general conclusions are presented indicating possible future research in which the problems explained can be expanded and also indicating the possibility of extending the approach presented to other bi_objetivo manufacturing lines, not necessarily painting systems.

RESUM

Problema de Programació de lots cíclics amb temps de preparació dependents e independents de la seqüència amb Aplicacions a Sistemes de pintura.

En el món del segle XXI, en un entorn en general globalitzat en el que els mètodes i procediments aplicats als processos industrials estan estandarditzats i més que provats, encara es possible intentar descobrir formes innovadores de millorar processos àmpliament estudiats dins de les tècniques clàssiques de Organització Industrial. Definir una manera eficient de seqüenciar una sèrie de productes en un procés productiu impacta de forma rellevant en la productivitat i en els costos de fabricació.

En els processos de pintura de les empreses de fabricació de components per a la indústria automotriu es realitza la seqüenciació de productes en base al programa mestre que indiquen els clients. El programa de fabricació es realitza seguint els requeriments de models i colors que és necessari entregar. En les fabriques amb estos processos la informació sobre les especificacions dels productes es sol rebre amb una finestra de temps molt curta, en alguns casos de tan sols 6 hores. Açò requereix una flexibilitat de resposta molt alta. Esta flexibilitat es pot aconseguir de diferents maneres, algunes de estes son mitjançant canvis ràpids de color i model en els processos de pintura, mitjançant inventaris de producte acabat que permeten sortir el producte requerit o mitjançant millores en la manera de seqüenciar els productes. Esta tesis analitza i busca la manera de millorar els procediments de seqüenciació per a minimitzar els canvis que generen un setup en els processos de pintura.

La generació de *setups* en una planta de pintures de components d'automòbil poden ocórrer per dos circumstancies: 1) canvi en el model del producte i 2) canvi en el color utilitzar per a pintar. Estos casos ocorren de forma independent entre sí, es a dir, en un moment específic pot ocórrer un cas, un altre o ambdós al mateix temps. S'ha detectat que les tècniques de seqüenciació existents no contempen la situació d'optimitzar un procés productiu de pintura minimitzant el número de canvis que es generen per les dues situacions presentades.

La motivació principal en esta tesis es desenvolupar procediments de solució al problema de seqüenciació bi_objectiu en els processos de pintura donat que les tècniques existents per a minimitzar el número de *setups* no contempen este cas. L'aplicació de les tècniques clàssiques resulta insuficient per a aconseguir la eficiència al generar una millor seqüència.

Es parteix del cas d'algunes empreses específiques per a delimitar el problema en una sèrie d'hipòtesis de treball. Es fa una revisió de les referències sobre el tema afins que inclouen la seqüenciació en processos amb circuit tancat, la programació cíclica i l'optimització de *setups*. Donades les hipòtesis de treball i la literatura existent es confirma la inexistència de models, mètodes i procediments en este entorn a excepció de dos referències. Les hipòtesis inicials foren ampliades per a canviar l'enfocament del problema: no únicament s'aborda el cas bàsic de minimitzar els número de canvis de model i de color sinó que s'analitza un segon problema: minimitzar el número d'espais buits que ocorren quan hi ha un *setup* causat per el canvi de model o color. En els dos casos es dissenyen nous models matemàtics en la forma de contar els espais buits amb horitzó incert en el model de buits que s'explica en profunditat.

Donat que els dos problemes son *NP_hard* i s'exploren mitjançant programari d'optimització, ràpidament creixen en el temps de resolució. Per esta raó es justifica el desenvolupament de diferents procediments heurístics que proporcionen una bona solució computacional en un temps raonable. Per a tots estos procediments s'analitza un codi de programació que permet simular cada cas. En el problema sense buits es generen 27 procediments heurístics. En el cas amb buits han sigut 19 procediments. Per ambdós problemes es presenten conclusions sobre les heurístiques més eficients justificant-les tant des de el punt de vista estadístic, com d'anàlisi comparatiu amb el òptim teòric.

Finalment es presenten conclusions general son s'indiquen línies d'investigació futures, en les que es poden ampliar els problemes presentats i la possibilitat d'ampliar l'enfocament presentat a línies de fabricació bi-objectiu no necessàriament de sistemes de pintura.

INDICE GENERAL

1	INTRODUCCION	16
1.1	PRESENTACION	16
1.2	JUSTIFICACION Y OBJETIVO DE LA TESIS	17
1.3	ESQUEMA GENERAL DE LA TESIS	18
2	DESCRIPCION DEL PROBLEMA DE PROGRAMACION DE LAS LINEAS DE PINTURA EN LA INDUSTRIA AUTOMOTRIZ CON LOTES CICLICOS Y TIEMPOS DE PREPARACION DEPENDIENTES E INDEPENDIENTES DE LA SECUENCIA.....	20
2.1	LA MANUFACTURA AUTOMOTRIZ.....	20
2.1.1	La Producción en línea.....	20
2.1.2	El proceso de pintura de parachoques.....	21
2.1.3	Caso de la empresa Plastic Omnium (Silao, Gto. México):	27
2.2	DEFINICION DEL PROBLEMA.....	31
2.2.1	Generalidades	31
2.2.2	Delimitación del problema. Hipótesis básica	34
2.2.3	Objetivos del problema: variantes del CBBSSP	35
2.3	REVISION DE LA LITERATURA PARA EL PROBLEMA CBBSSP	36
2.3.1	Referencias encontradas en la literatura.....	36
2.3.2	Diferencias de las referencias existentes respecto al planteamiento de esta Tesis sobre el CBBSSP	36
2.4	CONCLUSIONES.....	36
3	ESTADO DEL ARTE	38
3.1	INTRODUCCION.....	38
3.2	Programación en Sistemas de Circuito Cerrado:.....	39
3.2.1	Introducción:	39
3.2.2	Descripción del sistema de circuito cerrado:	40
3.3	SETUP.....	51
3.3.1	Introducción	51
3.3.2	Definición de tiempo/coste de <i>Setup</i>	53
3.3.3	Clasificación.....	55
3.3.4	Problemas <i>Setup</i> de una máquina.	59
3.3.5	Problema de <i>Setup</i> de máquinas en paralelo.	60
3.3.5.1	Restricción de flexibilidad en las máquinas en paralelo.....	60
3.3.5.2	Restricciones de <i>Setup</i> en máquinas en paralelo.....	61
3.3.5.3	Restricciones de disponibilidad de recursos.	63
3.3.6	Problemas de <i>Setup</i> de piso (<i>Flowshop</i>):	64
3.3.6.1	<i>Flowshop</i> general:	64

3.3.6.2	<i>Flowshop</i> sin tiempos de espera:.....	64
3.3.6.3	<i>Flowshop</i> Flexible:.....	65
3.3.6.4	<i>Flowshop</i> de ensamble:.....	66
3.4	SCHEDULING	67
3.4.1	Introducción:	67
3.4.2	Programación Estática.	69
3.4.3	Programación Dinámica:	70
3.4.4	Programación Cíclica:.....	72
3.4.4.1	Introducción:	72
3.4.4.2	Descripción y definiciones:	72
3.4.4.2.1	Programación cíclica de taller (Jobshop):	73
3.4.4.2.2	Programación cíclica del taller de flujo (<i>Flowshop</i>):	77
3.4.4.2.3	Programación cíclica automatizada (Robot):	77
3.4.4.3	Revisión de la literatura:	78
3.5	Programación Multi-Objetivo:	82
3.5.1	Introducción	82
3.5.2	Optimización Multi-Objetivo	82
3.5.3	Frontera de Pareto.....	84
3.5.4	Métodos de solución de problemas Multi-Objetivo.....	86
3.5.4.1	Métodos de optimización no-lineales.	86
3.5.4.2	Métodos basados en la preferencia	86
3.5.4.3	Métodos generadores.....	86
3.5.4.4	Método de ponderación.....	86
3.5.4.5	Método de restricciones	87
3.6	CONCLUSIONES.....	87
4	PROBLEMA CBBSSP SIN HUECOS	88
4.1	MODELADO MATEMATICO PARA EL PROBLEMA CBBSSP SIN HUECOS	88
4.1.1	NOMENCLATURA.....	88
4.1.1.1	Indices:	88
4.1.1.2	Parámetros:	88
4.1.1.3	Variables de Decisión:.....	89
4.1.2	MODELADO DEL PROBLEMA CBBSSP-SH	89
4.1.2.1	Función Objetivo del CBBSSP-SH.....	90
4.1.2.2	Restricciones básicas del problema CBBSSP-SH Opción 1.	90
4.1.2.3	Restricciones básicas del problema CBBSSP-SH Opción 2.	94
4.1.2.4	Modelo general del problema CBBSSP-SH	96
4.1.2.5	Límites al modelo y análisis del entorno en el Problema CBBSSP-SH.	97
4.1.3	CONCLUSION DEL MODELADO	101
4.2	PROCEDMIENTOS EXACTOS PARA LA RESOLUCION DEL PROBLEMA CBBSSP SIN HUECOS.....	101

4.2.1	METODOLOGIA DE LOS PROCEDIMIENTOS EXACTOS DEL PROBLEMA CBBSSP-SH.	102
4.2.2	IMPLEMENTACION DE LOS MODELOS PARA RESOLVER CON GUSEK-GUROBI102	
4.2.2.1	Operacionalización de un ejemplo básico.....	102
4.2.2.2	Implementación del código de programación generalizado para resolver el problema CBBSSP-SH con GUSEK-GUROBI.....	108
4.2.3	RESULTADOS DE LA EXPERIMENTACION DEL PROBLEMA CBBSSP SIN HUECOS EN GUSEK-GUROBI.....	108
4.2.3.1	Definición de los casos analizados en los experimentos.....	109
4.2.3.2	Resultados de los experimentos.	110
4.2.4	ANALISIS DE RESULTADOS DE LOS PROCEDIMIENTOS EXACTOS SIN HUECOS.....	114
4.3	PROCEDIMIENTOS HEURÍSTICOS PARA LA RESOLUCION DEL PROBLEMA CBBSSP SIN HUECOS.....	115
4.3.1	METODOLOGIA DEL DESARROLLO DE HEURISTICAS SIN HUECOS.	116
4.3.2	MARCO GENERAL DE LAS HEURISTICAS DESARROLLADAS PARA EL PROBLEMA DEL CBBSSP-SH.	116
4.3.2.1	Procedimiento General:.....	117
4.3.2.2	Tabla General de las Heurísticas CBBSSP-CH:.....	119
4.3.3	HEURISTICAS HORIZONTALES	120
4.3.4	HEURISTICAS VERTICALES	121
4.3.5	HEURISTICAS DIAGONALES	122
4.3.6	CODIFICACION Y DEFINICION DE LOS PROCEDIMIENTOS EXISTENTES.	123
4.4	ESTUDIO EXPERIMENTAL DE LOS PROCEDIMIENTOS DESARROLLADOS	123
4.4.1	METODOLOGIA DEL ESTUDIO EXPERIMENTAL.....	123
4.4.2	GENERACION DE PROBLEMAS PARA EL ESTUDIO EXPERIMENTAL.....	124
4.4.3	ESTUDIO EXPERIMENTAL DE LOS PROCEDIMIENTOS HEURISTICOS.....	126
4.4.3.1	Análisis de las Listas	126
4.4.3.2	Análisis Algoritmos	130
5	PROBLEMA CBBSSP CON HUECOS	136
5.1	MODELADO MATEMATICO PARA EL PROBLEMA CBBSSP CON HUECOS	136
5.1.1	MODELADO DEL PROBLEMA CBBSSP-CH OPCION 1.....	136
5.1.1.1	NOMENCLATURA	136
5.1.1.2	MODELO DEL PROBLEMA CBBSSPCH OPCION 1.....	137
5.1.2	MODELADO DEL PROBLEMA CBBSSP-CH OPCION 2.....	145
5.1.2.1	NOMENCLATURA	145
5.1.2.2	MODELO DEL PROBLEMA CBBSSPCH OPCION 2.....	146
5.1.3	MODELO GENERAL DEL PROBLEMA CBBSSP-CH.....	151
5.1.3.1	Modelo general del problema CBBSSP Opción 1.....	151

5.1.3.2	Modelo general del problema CBBSSP Opción 2.....	151
5.1.4	LIMITES DEL MODELO Y ANALISIS DEL ENTORNO.	152
5.2	PROCEDIMIENTOS EXACTOS PARA LA RESOLUCION DEL PROBLEMA CBBSSP CON HUECOS.....	156
5.2.1	METODOLOGIA PROCEDIMIENTOS EXACTOS PROBLEMA CON HUECOS. 157	
5.2.2	IMPLEMENTACION DE LOS MODELOS PARA RESOLVER CON GUSEK – GUROBI.....	157
5.2.3	RESULTADOS DE LA EXPERIMENTACION DEL PROBLEMA CBBSSP CON HUECOS EN GUSEK-GUROBI.....	157
5.2.3.1	Definición de los casos analizados en los experimentos.....	157
5.2.3.2	Resultados de los experimentos.	158
5.2.4	ANALISIS DE RESULTADOS DE LOS PROCEDIMIENTOS EXACTOS CON HUECOS.....	164
5.3	PROCEDIMIENTOS HEURÍSTICOS PARA LA RESOLUCION DEL PROBLEMA CBBSSP CON HUECOS.....	164
5.3.1	METODOLOGIA DEL DESARROLLO DE HEURISTICAS CON HUECOS.	165
5.3.2	MARCO GENERAL DE LAS HEURISTICAS DESARROLLADAS PARA EL PROBLEMA DEL CBBSSP-CH.	166
5.3.2.1	Procedimiento General:.....	166
5.3.3	HEURISTICAS HORIZONTALES	169
5.3.4	HEURISTICAS VERTICALES	170
5.3.5	HEURISTICAS DIAGONALES	171
5.3.6	CODIFICACION Y DEFINICION DE LOS PROCEDIMIENTOS EXISTENTES .	171
5.4	ESTUDIO EXPERIMENTAL DE LOS PROCEDIMIENTOS DESARROLLADOS PROBLEMA CON HUECOS	172
5.4.1	INTRODUCCION.....	172
5.4.2	METODOLOGIA DEL ESTUDIO EXPERIMENTAL.....	172
5.4.3	GENERACION DE PROBLEMAS PARA EL ESTUDIO EXPERIMENTAL.....	173
5.4.4	ESTUDIO EXPERIMENTAL DE LOS PROCEDIMIENTOS HEURISTICOS.....	174
5.4.4.1	Validación de los datos generados en los experimentos con las heurísticas. 174	
5.4.4.2	Resultados del Análisis de Varianza	175
5.4.4.3	Resultados de los Algoritmos:	182
5.4.4.4	Resultados de las Listas:	184
6	CONCLUSIONES Y LINEAS FUTURAS DE INVESTIGACION.....	185
6.1	CONCLUSIONES Y APORTACIONES.....	185
6.2	LINEAS FUTURAS DE INVESTIGACION	185
7	BIBLIOGRAFIA	186
	ANEXOS.....	195

Indice de Figuras:

Figura 1: Flujograma del ensamblaje automotriz (Nissan Aguascalientes).....	21
Figura 2: Parachoques delantero	22
Figura 3: Parachoques trasero.....	23
Figura 4: Proceso de Pintado de parachoques.....	24
Figura 5: Pintado de parachoques por medio de robots.....	24
Figura 6: Sistema de pintura con circuito cerrado (2)	26
Figura 7: Secuenciación en circuitos sucesivos (2)	27
Figura 8: Esquema de flujo de requerimientos en un sistema Pull.....	32
Figura 9: Esquema básico de un proceso con transportadores en circuito cerrado.....	33
Figura 10: Estructura General del Estado del Arte	39
Figura 11: Revisión de Literatura en Programación de Sistemas de Circuito Cerrado	40
Figura 12: Sistema General de Circuito Cerrado	42
Figura 13: Proceso Birth –Death.....	45
Figura 14: Proceso Quasi Birth-Death	45
Figura 15: Línea de Producción con dos circuitos y tres máquinas.....	48
Figura 16: Revisión de Literatura en Problemas con <i>setup</i> relevantes a la Tesis.....	52
Figura 17: Gráfico de gantt para un sistema de flowshop sin tiempos de espera.	65
Figura 18: Revisión de Literatura en Problemas con <i>scheduling</i> relevantes a la Tesis	68
Figura 19: Representación del espacio de soluciones y objetivos	84
Figura 20: Frontera de Pareto	85
Figura 21: Comparativo del Tiempo de Optimización CBBSSP-SH Dem 10 Bloques 5	110
Figura 22: Comparativo del Tiempo de Optimización CBBSSP-SH Dem 20 Bloques 5	111
Figura 23: Comparativo del Tiempo de Optimización CBBSSP-SH Dem 10 Bloques 10	112
Figura 24: Comparativo del Tiempo de Optimización CBBSSP-SH Dem 40 Bloques 5	113
Figura 25: Comparativo del Tiempo de Optimización CBBSSP-SH Dem 40 Bloques 10	114
Figura 26: Comportamiento relativo de las Listas respecto al óptimo con Factor "P"	128
Figura 27: Comportamiento Relativo de la Lista 7 en función del valor "P".....	129
Figura 28: Comportamiento Relativo de la Lista 6 en función del valor "P".....	129
Figura 29: Comportamiento Relativo de la Lista 5 en función del valor "P".....	130
Figura 30: Comportamiento relativo de los Algoritmos respecto al óptimo con Factor "P"	134
Figura 31: Comportamiento relativo de los Algoritmos con mejor desempeño respecto al óptimo con Factor "P".....	135
Figura 32: Comparativo del Tiempo de Optimización CBBSSP-CH Dem 10 Bloques 5.....	159
Figura 33: Comparativo del Tiempo de Optimización CBBSSP-CH Dem 20 Bloques 5.....	160
Figura 34: Comparativo del Tiempo de Optimización CBBSSP-CH Dem 20 Bloques 10.....	161
Figura 35: Comparativo del Tiempo de Optimización CBBSSP-CH Dem 40 Bloques 5.....	163
Figura 36: Comparativo del Tiempo de Optimización CBBSSP-CH Dem 40 Bloques 10.....	164

Indice de Tablas:

Tabla 1: Tipo de Taller	57
Tabla 2: Características de Taller	57
Tabla 3: Información General del Setup	58
Tabla 4: Tabla de Factores de Medición del Desempeño en Setup	59
Tabla 5: Ejemplo implementado. Demanda por tipo de Producto	103
Tabla 6: Ejemplo implementado. Secuenciación en un sistema con 5 bloques	103
Tabla 7: Resultados Procedimientos Exactos Sin Huecos con Demanda 10 Bloques 5	110
Tabla 8: Resultados Procedimientos Exactos Sin Huecos con Demanda 20 Bloques 5	111
Tabla 9: Resultados Procedimientos Exactos Sin Huecos con Demanda 20 Bloques 10	112
Tabla 10: Resultados Procedimientos Exactos Sin Huecos con Demanda 40 Bloques 5	113
Tabla 11: Resultados Procedimientos Exactos Sin Huecos con Demanda 40 Bloques 10	114
Tabla 12: Marco General de Heurísticas Desarrolladas para el Problema Sin Huecos	120
Tabla 13: Resumen Heurísticas Horizontales Sin Huecos	121
Tabla 14: Resumen Heurísticas Verticales Sin Huecos	122
Tabla 15: Resumen Heurísticas Diagonales Sin Huecos	122
Tabla 16: Instancias del Problema CBBSSP-SH	124
Tabla 17: Resultados Procedimientos Exactos Con Huecos con Demanda 10 Bloques 5	159
Tabla 18: Resultados Procedimientos Exactos Con Huecos con Demanda 20 Bloques 5	160
Tabla 19: Resultados Procedimientos Exactos Con Huecos con Demanda 20 Bloques 10	161
Tabla 20: Resultados Procedimientos Exactos Con Huecos con Demanda 40 Bloques 5	162
Tabla 21: Resultados Procedimientos Exactos Con Huecos con Demanda 40 Bloques 10	163
Tabla 22: Resumen Heurísticas Horizontales Con Huecos	170
Tabla 23: Resumen Heurísticas Verticales Con Huecos	171
Tabla 24: Resumen Heurísticas Diagonales Con Huecos	171
Tabla 25: Instancias del Problema CBBSSP-CH	173
Tabla 26: Frecuencias de los datos por Algoritmo	175
Tabla 27: Frecuencia de los datos por Lista	175
Tabla 28: Efectos inter-sujetos en variables independientes del Problema CBBSSP-CH	176
Tabla 29: Tukey-B. Efecto de variable Algoritmos sobre Datos Generales	178
Tabla 30: Tukey-B. Efecto de variable Listas sobre Datos Generales	178
Tabla 31: Análisis de Varianza de la variable Algoritmos a diferentes niveles de Demanda ...	179
Tabla 32: Tukey-B. Efecto de variable Algoritmos sobre Datos con Demanda 100	180
Tabla 33: Tukey-B. Efecto de variable Algoritmos sobre Datos con Demanda 1000	181
Tabla 34: Tukey-B. Efecto de variable Algoritmos sobre Datos con Demanda 10000	181
Tabla 35: Análisis de Varianza de la variable Listas a diferentes niveles de Demanda	182
Tabla 36: Tukey-B. Efecto de variable Listas sobre Datos con Demanda 100	183
Tabla 37: Tukey-B. Efecto de variable Listas sobre Datos con Demanda 1000	183
Tabla 38: Tukey-B. Efecto de variable Listas sobre Datos con Demanda 10000	184

1 INTRODUCCION

1.1 PRESENTACION

Dadas las características de la Industria Automotriz los procesos que antiguamente se realizaban en una sola planta se realizan actualmente en un conjunto de proveedores que entregan el producto a la empresa ensambladora de acuerdo a un programa maestro de fabricación. Los procesos que se deben adaptar para ser diseñados y ejecutados por varias empresas, separadas y de diferentes características pero que en conjunto conforman una cadena o red de suministro.

Dentro de las empresas señaladas en la situación anterior se encuentra el caso particular de aquellas que tienen procesos de pintura con transportadores de circuito cerrado.

Cada vez son mayores las necesidades de efectuar procesos flexibles que minimicen los costes del producto final. Dada la variedad de productos que el mercado requiere, la flexibilidad en los procesos debe llevarse al límite para ofrecer opciones a los clientes finales. En el caso de los procesos de pintura se realizan cambios para que el producto pueda tener gran cantidad de colores. Los distintos modelos requieren componentes con una geometría distinta entre sí. Combinando la variedad requerida de modelos y colores se genera mayor complejidad que crece exponencialmente con el incremento en el valor de ambas variables.

Existen muy pocos estudios que analicen el problema específico que se menciona. Una de las razones por las que posiblemente no se habrían estudiado estos casos es porque en la Industria Automotriz el Plan de Fabricación es proporcionado por la fábrica ensambladora. Generalmente las empresas que forman parte de la red de fabricación han seguido la secuencia proporcionada por la planta armadora en sus procesos y cuando hay alguna divergencia importante esta es compensada con material o producto del inventario. No existen estudios de campo que hayan buscado mejorar la secuencia de los productos para minimizar el número de cambios. La forma de hacerlo ha sido empírica.

Durante la elaboración de la Tesis se ha tenido oportunidad de conocer la realidad industrial en el tema de secuenciación de algunas empresas en las que existe la oportunidad de aplicación de los conocimientos desarrollados en el presente trabajo. De acuerdo al personal que en ellas trabaja, el beneficio puede ser relevante dado que el coste de cambios es importante y aun mayor el de dejar espacios vacíos en la línea.

Con esta información percibida en las entrevistas tenidas con personal de estas empresas, se hace la justificación que se presenta en el siguiente apartado.

1.2 JUSTIFICACION Y OBJETIVO DE LA TESIS

A continuación se enumeran de manera simplificada las principales razones que justifican la realización de esta tesis Doctoral:

- 1) Dentro de la industria Automotriz, en los procesos de ensamblaje existe cada vez mayor necesidad de contar con proveedores con procesos flexibles que estén dispuestos a entregar una gran variedad en los componentes que fabrican
- 2) En la literatura no se han encontrado modelos que consideren todos los elementos en forma integral para optimizar la secuenciación de productos en los procesos de pintura.
- 3) Existen solamente dos estudios que considera el caso del Problema de Secuenciación con Huecos.
- 4) La importancia de demostrar que los problemas revisados en esta tesis crecen en complejidad exponencialmente al incrementar los valores de las variables independientes del problema, es decir, de los valores del tamaño de la demanda, el número de productos por ciclo dentro del circuito cerrado y el número de colores y modelos que se necesite programar.
- 5) La utilidad de desarrollar heurísticas de aplicación simple que hayan sido debidamente justificadas a través del estudio de simulaciones.
- 6) La certidumbre de que las empresas consultadas han detectado esta problemática pero carecen de herramientas que puedan mejorar la productividad por el motivo analizado en esta Tesis.

Con base en las justificaciones mencionadas surgen los siguientes Objetivos de la Tesis:

- 1) Revisar la literatura existente sobre el tema de secuenciación en procesos de pintura con sistemas en circuito cerrado incluyendo la literatura sobre temas afines aun cuando no contemplen la totalidad de las variables en el mismo estudio.
- 2) Generar dos visiones distintas de modelado, una para cada problema analizado. Tanto para el Problema Sin Huecos como para el Problema Con Huecos. En ambos casos hacer el modelado matemático.
- 3) Comprobar si los casos descritos en el punto anterior crecen en complejidad a medida que se incrementan los valores de las variables independientes.
- 4) Diseñar heurísticas de solución para cada uno de los problemas.
- 5) Diseñar códigos de programación para cada heurística propuesta.
- 6) Hacer experimentos con las heurísticas para determinar las de mejor desempeño relativo y compararlas con el óptimo teórico.
- 7) Describir la metodología para poder aplicar en la industria las heurísticas de mejor desempeño.

- 8) Generar conclusiones sobre el método utilizado y la aplicación de los resultados.

1.3 ESQUEMA GENERAL DE LA TESIS

Después de este primer Capítulo de Introducción. La tesis se estructura en cinco partes: Definición del Problema, Estado del Arte, Propuesta y Análisis del caso Sin Huecos, Propuesta y Análisis del caso Con Huecos, y Conclusiones y Líneas Futuras de Investigación.

Es importante mencionar que en esta Tesis se ha decidido poner el capítulo de Descripción del Problema precediendo al capítulo del Estado del Arte. La razón para hacerlo de esta manera es que dado que no existe bibliografía que considere exclusivamente el tema de la Tesis, el Estado del Arte parte de otros temas conocidos. En la Descripción del Problema se exploran cuáles pueden ser los temas afines sobre los cuales es recomendable hacer exploración. De esta manera se hace primero una definición del problema, se explora la literatura existente sobre el tema en específico y enseguida se procede a hacer una búsqueda en los temas consistentes con el problema descrito.

Como se ha mencionado, en el Capítulo 2 se hace la definición del problema de la Tesis. En este caso como, se explica a continuación y más ampliamente en dicho capítulo se trata de dos problemas. Inicialmente se había considerado solamente el problema sin huecos. Sin embargo dado que durante el proceso de elaboración de la tesis se han generado nuevos artículos, al menos uno, que analiza este problema, la novedad del caso se veía restringida solo a las aportaciones que se hubiesen podido hacer sobre la forma de resolución. Por este motivo se decidió hacer una extensión del problema al caso con huecos. El problema con huecos presenta particularidades de análisis distintas al caso sin huecos. En el aspecto de costes se agrega el que implica tener un espacio vacío al existir un cambio de producto. Este cambio puede ser por causa de la aparición en la secuencia de un modelo distinto o de un color distinto. Sin embargo, el modelo se transforma en uno del tipo mono-objetivo, pues el resultado de ambas situaciones se refleja indistintamente en un espacio vacío en la secuencia de fabricación con su coste asociado.

La parte correspondiente al Estado del Arte se hace en el capítulo 3. En este capítulo se lleva a cabo una revisión del concepto de Programación en Sistemas de Circuito Cerrado, haciendo énfasis en las características particulares de este tipo de sistemas. Cabe destacar que aunque dentro de la industria son sistemas de uso frecuente muchas veces se consideran solo como una caja cerrada en la que la productividad se mide como una entrada y una salida sin hacer un análisis del comportamiento interno.

Se revisan algunos conceptos relevantes a los sistemas de circuito cerrado, las estaciones. El flujo de materiales entre una estación *A*, donde se efectúa la carga y otra estación *B* que es donde se descarga el producto. Se considera circuito cerrado porque en el modelo básico el sistema de transporte regresa a la estación de carga *A* con los dispositivos utilizados vacíos. Adicionalmente se revisan los conceptos de sistemas con procesos *Birth-Death* los cuales se basan en procesos estocásticos del tipo *Markov* donde cada

estado puede ser asociado con un elemento del conjunto y que en su forma más compleja se conocen como *Quasi Birth-Death*.

Dado que el tema central de esta tesis es minimizar el número de *setups* o los espacios vacíos generados por los mismos se revisa en el Estado del Arte este tema. Se revisa la clasificación de este concepto y las formas en que se ha buscado minimizar sus tiempos y sistemas *flowshop* o *jobshop*.

Se revisa una clasificación general del *scheduling* en la industria. Se revisa lo descrito en la literatura sobre los casos de programación estática, dinámica y cíclica. Se pone especial interés al definir un apartado para el caso de la programación multi-objetivo dado que el modelo sin huecos es un problema que presenta dos objetivos a minimizar. El resultado en estos casos es una Frontera de Pareto por lo que se aclara este concepto y se revisan los métodos de solución.

En el Capítulo 4 se analiza el caso del problema sin huecos. Al hacer el análisis de este problema se incluyen 4 apartados que bien podrían haber sido considerados como capítulos independientes. Para efectos de mejor comprensión y una forma más sintetizada de presentar el problema se agrupan en el mismo capítulo como apartados del mismo. Estos son: a) Modelado Matemático del Problema Sin Huecos, b) Procedimientos Exactos de resolución y Experimentación, c) Procedimientos Heurísticos del Problema Sin Huecos y d) Experimentación con Análisis Estadístico de los Resultados, finalmente Conclusiones sobre el Problema Sin Huecos.

En el Capítulo 5 se revisa el problema con huecos. En forma análoga al Capítulo 4 se incluyen 4 apartados que también podrían ser considerados como capítulos independientes y para efectos de claridad y sencillez en la revisión de la tesis se incluyen dentro del mismo. El título es el mismo que los apartados del capítulo anterior aplicado al caso del Problema con Huecos: a) Modelado Matemático del Problema Con Huecos, b) Procedimientos Exactos de resolución y Experimentación, c) Procedimientos Heurísticos del Problema Con Huecos y d) Experimentación con Análisis Estadístico de los Resultados y Conclusiones sobre el Problema Con Huecos.

Finalmente, en el capítulo 6, se recogen las principales Conclusiones y Líneas Futuras de investigación que surgen a partir de este trabajo.

Se cierra la Tesis con un conjunto de Anexos, entre los que figuran, entre otros las tablas de análisis estadístico realizadas en SPSS 17, los códigos de programación en GLPK utilizados en los procedimientos exactos tanto para el caso del problema sin huecos como el problema con huecos.

2 DESCRIPCION DEL PROBLEMA DE PROGRAMACION DE LAS LINEAS DE PINTURA EN LA INDUSTRIA AUTOMOTRIZ CON LOTES CICLICOS Y TIEMPOS DE PREPARACION DEPENDIENTES E INDEPENDIENTES DE LA SECUENCIA

2.1 LA MANUFACTURA AUTOMOTRIZ

2.1.1 La Producción en línea.

La manufactura automotriz masiva se basa en el antiguo paradigma de “producción en cadena”, el cual fue un proceso revolucionario en la producción industrial cuya base es la cadena de montaje o *línea de producción*; una forma de organización de la producción que delega a cada trabajador una función específica y especializada en máquinas también más desarrolladas. Una cadena de montaje es un proceso de fabricación en el que las partes (por lo general piezas intercambiables) se añaden a un producto de una manera secuencial, con la finalidad de incrementar la velocidad de producción y la calidad.

En la industria moderna, dada la complejidad del producto final, es poco práctico implementar una sola línea de producción que incluya todo el proceso de transformación de la materia prima hasta el ensamblaje e inspección del vehículo terminado. Por este motivo el sistema suele dividirse en secciones realizadas por sub-plantas de producción o, en muchos casos, empresas contratadas para completar la cadena de suministros interna y externa.

En el sistema JIT de manufactura automotriz se considera a la línea encargada de ensamblar la carrocería con el resto de los componentes (como llantas, vidrios, plásticos o dispositivos electrónicos) como la línea principal, cuyos procesos son denominados ensamblajes finales. Las líneas a cargo de la transformación de materiales y sub-ensamblajes son consideradas **líneas secundarias**. Ver Figura 1 donde se muestran las líneas secundarias en tono oscuro en el caso de la Empresa Nissan de Aguascalientes, México. Estas son el tipo de líneas a las que se enfoca la tesis dada su vital importancia en el sistema productivo, ya que en sus procesos se definen gran parte de las características de calidad del producto y limitantes de su fabricación.

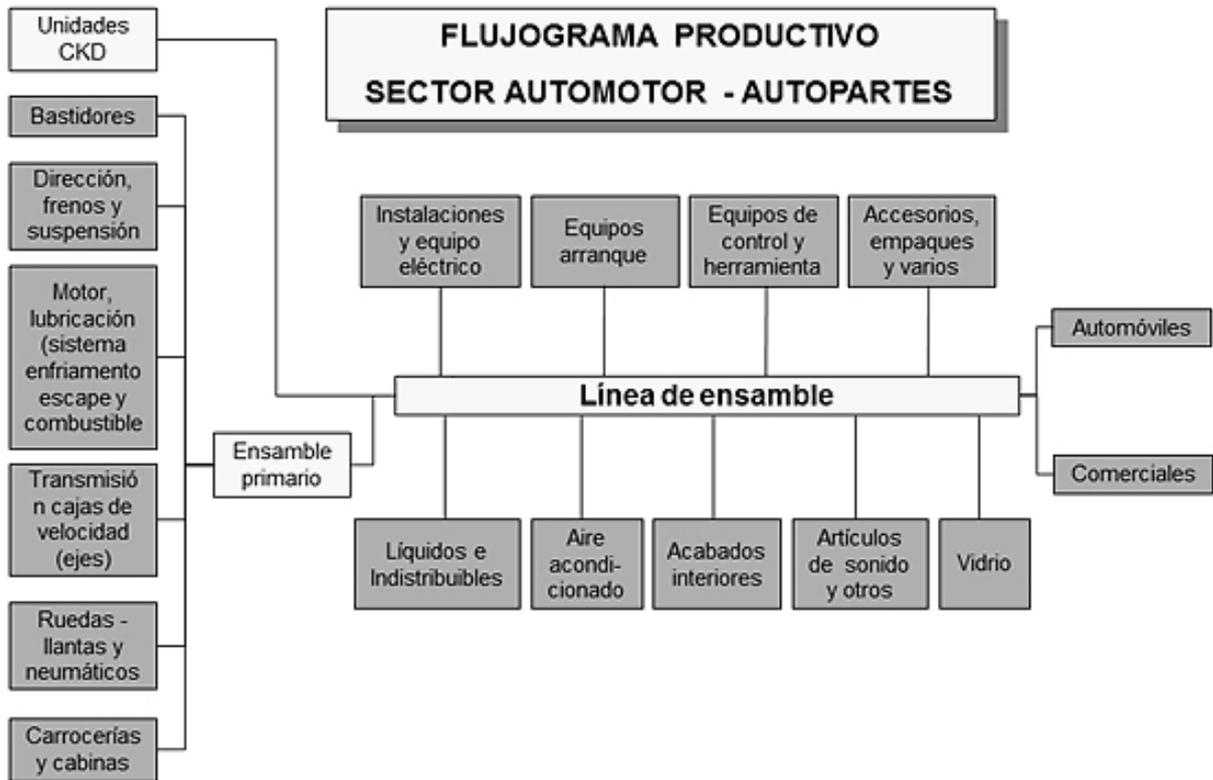


Figura 1: Flujograma del ensamblaje automotriz (Nissan Aguascalientes).

El sistema JIT propone la eliminación de desperdicios. Con esta finalidad, la línea principal marca la pauta del secuenciado de producción de la planta, al cual se sincronizan las líneas secundarias; reduciendo la necesidad de mantener inventarios entre un proceso y el otro. Esto requiere un alto grado de comunicación entre cliente y proveedor, y un suministro constante y fluido. La secuenciación de la línea principal busca cumplir con el objetivo principal de suministrar el producto cuando lo solicita el cliente, ni antes ni después (ambos casos son calificados como desperdicios y generan costos adicionales para la empresa) tomando en consideración todas las restricciones de las líneas sincronizadas. Ante esto, la organización se encuentra con el predicamento de incluir la mayor cantidad de procesos al sistema sincronizado, para reducir costos de manejo de inventarios; a costa de incluir posibles restricciones a la secuenciación de producción.

Frecuentemente en los sistemas de manufactura actuales, minimizar los costos de producción y minimizar la complejidad del sistema suelen ser objetivos contrapuestos. Esto puede tener un impacto negativo en el desempeño y bienestar de las personas involucradas.

2.1.2 El proceso de pintura de parachoques

Los automóviles son elaborados mediante la unión de piezas de materiales variados, en los que predominan el acero, el aluminio y distintos tipos de plástico. A pesar de que la estructura del vehículo está hecha casi en su totalidad de acero, existen dos piezas exteriores que comúnmente son

elaboradas con plástico: los parachoques. Ver Figura 2 y Figura 3. A diferencia de lo que su nombre podría indicar, la finalidad de los parachoques no es detener alguna colisión, sino servir de soporte a elementos funcionales como luces y placa de matrícula y además cumplir una función estética en la forma del diseño integral. Estas piezas son elaboradas con plásticos termoplásticos sea del tipo llamado uretano elastomérico moldeado o del polipropileno. Con estos materiales se busca reducir el peso del vehículo y facilitar la fabricación de diseños más complejos que los que permite el estampado del metal. A su vez el diseño ayuda a lograr una mejor fijación del parachoque en el vehículo al considerar distintos puntos de inserción como puede verse en la Figura 3.

Los parachoques son elaborados mediante un proceso de inyección y son pintados dentro de la misma sub-planta; la cual puede formar parte de un gran complejo automotriz (caso de Nissan en Aguascalientes, México) donde se manufacturan distintas partes del automóvil, además de ensamblar el vehículo completo, o de una planta independiente (caso de Plastic Omnium planta Silao, Guanajuato, México que manufactura algunos de los modelos de parachoques para las plantas de General Motors, Chrysler y Fiat en México y Arlington, Texas, USA). Es el proceso de pintura de estas piezas en el que se centra el problema.

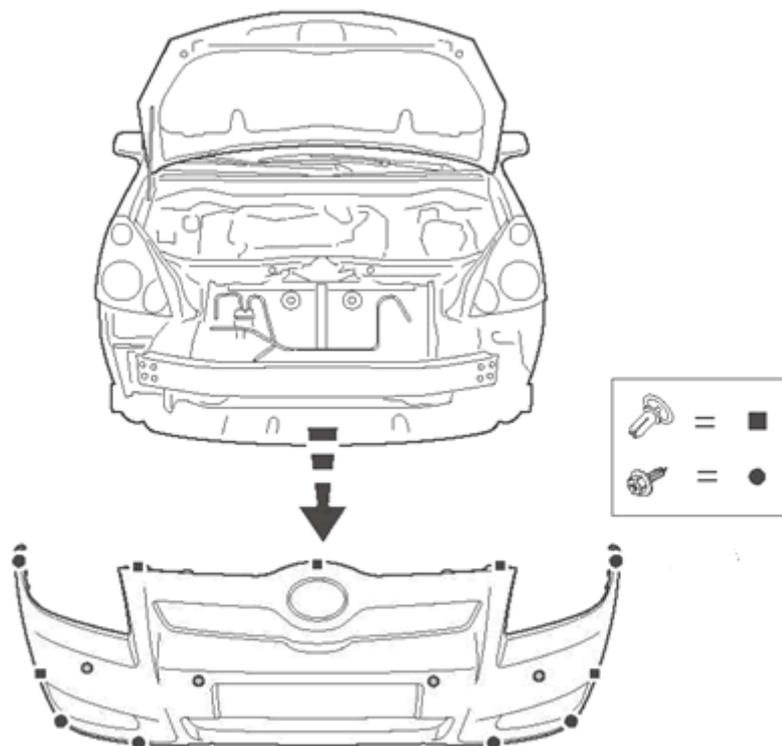


Figura 2: Parachoques delantero

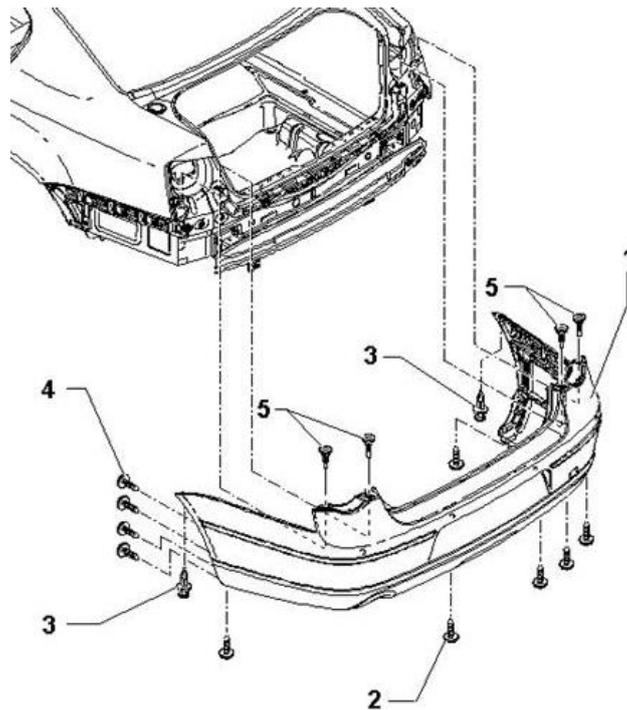


Figura 3: Parachoques trasero

Las piezas, cuyo plástico tiene un color específico, son almacenadas en un sistema automático antes de ser pintadas. Esto se debe a que las características del proceso de inyección en moldes hacen que se torne tardado y costoso manejar lotes muy pequeños. Éste proceso de inyección, es el único proceso de la planta de fabricación y pintura de parachoques que no está sincronizado con la línea de ensamblaje principal.

El proceso de pintura de parachoques cuenta con un sistema de transportadores en circuito cerrado. Las estructuras de acero, conocidas como bastidores (*jigs*), no solo trasladan un parachoques trasero y un parachoques delantero correspondientes al mismo vehículo de un puesto de trabajo a otro, sino que cumplen con la importante función de evitar la deformación de la pieza durante el proceso de horneado y secado de la pintura. Por lo que cada pieza debe ser transportada por un tipo de bastidor (*jig*) específico. Ver Figura 4.

Los transportadores son movidos individualmente por medio de sistemas de cadenas subterráneas a una velocidad constante por la mayoría del circuito, sin embargo algunas secciones no cuentan con cadenas para que los trabajadores puedan manipular los *jigs* y puedan formarse colas.

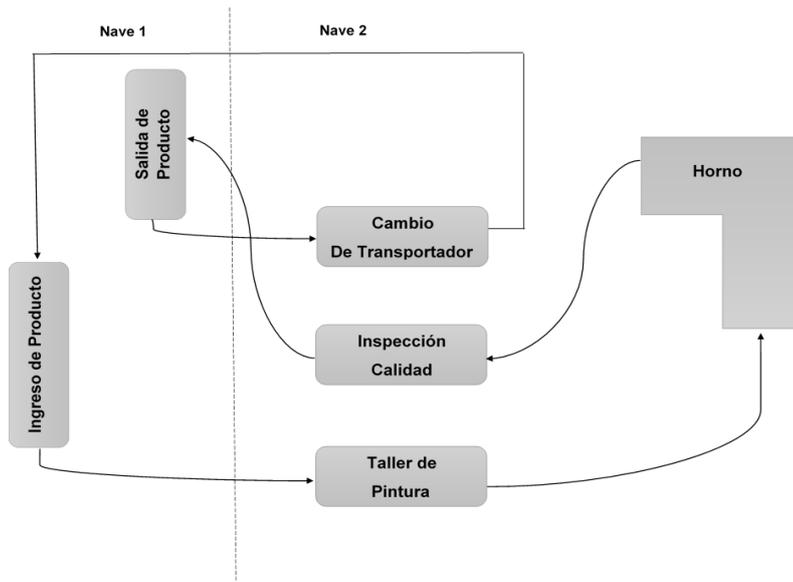


Figura 4: Proceso de Pintado de parachoques

El ciclo comienza con la recepción de las piezas que provienen del almacén, donde dos trabajadores se encargan de preparar la pieza para ser pintada; esto incluye la limpieza, remoción de rebabas y aplicación de coberturas para las zonas que no recibirán pintura. Una vez listas, las piezas son colocadas en los transportadores y llevadas al cuarto de pintura. Las distintas capas de pintura son aplicadas de forma automática por medio de robots, los cuales pueden ajustar el color de la pintura necesaria para cada pieza haciendo una purga entre una tarea y la otra. Ver Figura 5.

Tras ser pintadas, las piezas son llevadas al horno para afirmar la adherencia del color y darles un mejor acabado. Esta etapa cobra vital importancia ya que el extender la estancia de los parachoques dentro del horno causaría deformaciones en el plástico, lo cual limita al sistema completo a no poder detenerse.



Figura 5: Pintado de parachoques por medio de robots

A continuación los parachoques arriban a un pasillo destinado al control de calidad y tareas de retoque, y tras confirmar que se cumplen todos los requerimientos son llevados a la estación de trabajo donde son retirados del bastidor. Los transportadores siguen su camino hasta llegar a la estación donde comenzó el ciclo, pero antes son evaluados para confirmar que corresponden con el producto que portarán el ciclo siguiente y, en caso contrario, son cambiados manualmente por el modelo adecuado. Esta actividad genera dos situaciones no deseadas. La primera es que el remplazar los dos moldes de aproximadamente 35kg y llevarlos a su lugar es la tarea con mayor duración del ciclo. A pesar de que esta tarea no es requerida para todos los transportadores, cuando la secuenciación de producción es poco homogénea, se genera un cuello de botella. La segunda situación es que la tarea en sí misma es poco ergonómica e incluso peligrosa, a medida que los transportadores se comienzan a acumular esperando a ser remplazados y la persona a cargo se ve obligada a manipular materiales pesados en movimiento, de forma cada vez más rápida.

Una de las posibles soluciones a esta situación es automatizar el proceso, lo cual reduciría el tiempo requerido para llevar a cabo los ajustes necesarios en la secuencia de los transportadores y aumentaría la seguridad de la tarea.

Esta es una solución común dentro de la industria automotriz, sin embargo la implementación de esta estrategia requeriría de una inversión de alto coste, implicando no solo los gastos adicionales por la compra y operación del equipo, sino la reestructuración del complejo productivo. Las distintas implicaciones a tomar en cuenta pueden hacer que esta posibilidad sea impráctica o incluso imposible, dependiendo de los recursos con los que cuente la organización. Por lo tanto se requiere ofrecer una solución metodológica en lugar de tecnológica al problema, que se pueda implementar con los recursos disponibles en la planta y que los administradores de procesos puedan manipular.

Como se ha mencionado, los sistemas de pintura en la industria automotriz, especialmente en los componentes de plástico que pueden tener una amplia variedad de formas y de colores. Cada producto está definido por su geometría y por su color. El fabricante tiene en sus instalaciones un sistema cerrado que consiste en un transportador en circuito que contienen un número específico de posiciones o espacios en los que se colocará el producto o un grupo de productos.

Los productos se fijan en cada espacio con una herramienta especial llamada percha, bastidor o jig. Cada bastidor puede contener un número determinado y limitado de piezas con la misma geometría. Se especifica que sea con la misma geometría por la naturaleza del proceso. Con vista a mantener la claridad y sin perder generalidad se considerará un solo producto en cada bastidor. En el mundo real esto se puede conseguir dividiendo el número de productos por la demanda requerida. Es importante hacer hincapié en que cada tipo de geometría de producto requiere un bastidor distinto, aunque el mismo puede ser utilizado para múltiples colores. Esto significa que cuando el modelo a ser pintado es cambiado los bastidores se deben cambiar, pero cuando el cambio es en el color entonces no se necesita dicho cambio.

Si el color a pintar en las unidades sucesivas es diferente, puede ser necesario utilizar solvente en el sistema, que ha de purgar las tuberías. En algunos casos el PLC que controla los tubos de pintura debe ser reiniciado en el caso de que haya geometrías diferentes. Este cambio puede requerir un tiempo superior al disponible para dos posiciones consecutivas. Por lo tanto cuando un cambio de color va a ser programado ocurre lo que en esta tesis se llamará “*Setup Horizontal*” el cual implica una pérdida sea en tiempo de capacidad de producción o en coste. Por lo explicado, también sucede que en el cambio de geometría pueda existir una pérdida sea en capacidad o en coste. La novedad que no ha sido analizada en la literatura y que es objeto de análisis en este estudio es que cuando en ciclos sucesivos ocurre un cambio de geometría en la misma posición existe lo que llamaremos “*Setup Vertical*” que depende del producto que haya sido programado en el ciclo anterior. Las piezas pasan a través de una estación de pintura en una posición fija en la línea. Ver Figura 6. En el caso estudiado la programación por lotes es deseable para reducir al mínimo el tiempo de *setup* entre lotes consecutivos de productos similares y poder mantener un buen nivel de servicio al cliente. La programación cíclica es recomendable para reducir el trabajo en proceso (WIP) entre las instalaciones de la empresa y los clientes. Por lo mostrado se consideran dos tipos de configuraciones para el *setup*: *Setup Horizontal* y *Setup Vertical*.

1) *Setup Horizontal*: Es el que se genera por los cambios convencionales entre lotes consecutivos. Están relacionados con dos tipos de cambios en el producto: Cambio de color y cambio de geometría o diseño. Estos últimos se requieren por ejemplo cuando el software que controla las tuberías de pintura necesita ser modificado o actualizado por el cambio en el diseño del producto. Los cambios de color pueden requerir o no el uso de solventes para limpiar las tuberías o los espacios del transportador dependiendo de los colores que interviene, el que se deja de producir y el que se inicia.

2) *Setup Vertical*: Se tiene cuando un cambio de geometría se requiere en la misma posición del transportador pero en el bucle o circuito sucedáneo. En este caso se debe eliminar de la posición una percha o jig e instalar en ese mismo sitio uno distinto. En este caso se utiliza la mano de obra disponible, es decir tiempo de trabajo del operador. El uso de un trabajador tiene asociado un coste y esta es la razón por la que a cada cambio de geometría corresponde un coste de *Setup*.

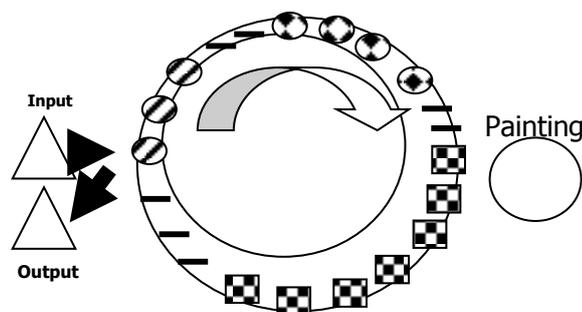


Figura 6: Sistema de pintura con circuito cerrado (2)

Una mejor descripción gráfica se puede observar en la Figura 7. En ella se muestran tres ciclos con un número de geometrías distintas (G1, G2, G3) y colores diferentes (C1, C2, C3) en cada bucle. Aparecen costes de *setup* debido al cambio de bastidor en bucles consecutivos y costes de *setup* por el cambio de color en posiciones consecutivas.

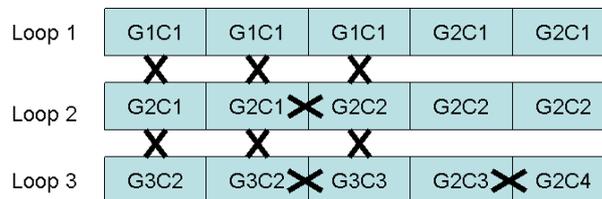


Figura 7: Secuenciación en circuitos sucesivos (2)

2.1.3 Caso de la empresa Plastic Omnium (Silao, Gto. México):

Generalidades:

Plastic Omnium fue fundada en Francia en 1946 por Pierre Burelle. Al 2014 cuenta con 111 plantas distribuidas en 30 países y tiene 2 divisiones:

- 1) Autopartes plásticas, que incluyen la producción de parachoques, partes estructurales y semiestructurales, de carrocería, etc.
- 2) Sistemas de innovación, entre otros, considera sistemas de carburación en plástico y reducción de emisiones en *diesel*.

La planta que la empresa tiene en Silao, Guanajuato, México está ubicada en el parque industrial “La Colina” de dicha población. Es una de las 8 que esta empresa tiene en México dedicadas a la fabricación de autopartes. Esta planta está enfocada a la producción de parachoques para las plantas de *General Motors* (fábricas en Arlington, Texas, San Luis Potosí y Silao), *Chrysler* y *Fiat* (Toluca, Estado de México).

La programación de la producción en esta empresa hace las siguientes consideraciones: existe un pronóstico con una anticipación de 9 meses en el número de piezas requerido. Tal como ocurre también en fábricas de Alemania y Japón, el pronóstico de demanda normalmente es realizado por el departamento de ventas y es un proceso dinámico. El pronóstico se hace con datos de los modelos y cantidades solicitadas por los distribuidores, el conocimiento del mercado dado por unidades descentralizadas que indican preferencias y tendencias, las campañas de mercadeo del departamento de mercadotecnia, los contratos planeados con clientes finales importantes (por ejemplo aquellos que compran flotillas de vehículos) y el análisis de datos históricos (1). El programa en firme con colores y modelos tiene una ventana de 8 semanas que puede tener variaciones de más o menos 10% al momento de la ejecución. La experiencia muestra que esto depende del cliente. En el caso de *General Motors* las variaciones son mínimas mientras que la frecuencia de dichos cambios es mayor en *Chrysler* y *Fiat*.

Dentro de la planta de los clientes se mantiene un inventario base a nivel mínimo. En las plantas de México entre 1.5 y 3 días de consumo y en la de Arlington 4 días. El cliente es quien determina la secuencia. Hacer un cambio

de última hora es caro. En todo momento se mantiene comunicación entre cliente y fabricante con una visión del programa de seis horas lo que permite hacer entrega de las piezas en cualquiera de las plantas de México desde el almacén en Plastic Omnium, Silao en el caso de existir cambios o modificaciones en la secuencia. En la planta del cliente se tiene un agente del fabricante que está pendiente de los cambios y modificaciones.

Sistema Comercial:

El producto se entrega en la planta del cliente. Se manejan dos tipos de contrato comercial:

- 1) *Pay on Production* (PoP). El cliente paga conforme utiliza el producto que ha sido entregado en su planta. El proveedor debe entregar los parachoques en el orden preciso que solicita el cliente directamente en la línea de fabricación. Este sistema requiere que la facturación y contabilidad se realicen solo de lo que se entrega en el piso de producción.
- 2) *Advanced Shipping Notice* (ASN) el material se factura al momento de emitir el "ASN" y posteriormente se entrega el producto. Es una notificación de la entrega pendiente similar a una lista de embarque. Normalmente se envía en un formato electrónico utilizando EDI. Adicionalmente a la lista de contenidos es posible incluir información adicional relevante para la recepción del envío tal como: información de la orden, descripción del producto, características físicas. tipo de empaque, configuración del producto en el transporte.

En ambos casos el coste del flete, que se hace en contenedores, lo paga el proveedor.

El Producto:

Parchoques plásticos elaborados con polipropileno. En la planta de Silao la variedad de colores que en el 2015 se tiene son 40 distintos. Los modelos de parachoques para los clientes actuales: *General Motors*, *Chrysler* y *Fiat* son entre 15 y 18. De acuerdo a datos actuales de resultados de calidad al final del proceso de pintura el 80% del producto se obtiene sin ningún defecto al momento de salir, 14% requieren un retrabajo por discrepancias menores que se pueden corregir y el 6% es rechazado por errores no corregibles. Entre los defectos más frecuentes se tiene: 1) Espesor no homogéneo en la capa de pintura, 2) Discrepancia con el color solicitado, 3) Defectos por mal horneado.

El Proceso:

La fabricación de parachoques en la planta de Silao incluye 3 procesos principales: Inyección de Plástico, Pintura y Ensamble de componentes adicionales. Para efecto de esta tesis el único relevante es el Proceso de Pintura. Las etapas para efectuar el pintado de los parachoques son las siguientes:

- 1) Aplicación de Primer: Es un proceso en que a la parte moldeada por inyección con polipropileno de un color específico, que depende del

diseño final producto, se le hace un lavado con agua y algunos productos químicos. El objetivo es abrir los poros para que penetre y permanezca la pintura en la pieza. Una vez efectuado este lavado se hace el secado eliminando humedad.

- 2) Aplicación de Base: En este proceso se hace la aplicación de la pintura automotriz que da el color a los parachoques. Se hace mediante 4 robots automatizados. Cada bastidor pasa por los 4 robots. Cada uno está programado para pintar una sección del parachoque.
- 3) Aplicación de Brillo. Aplicación de la capa de brillo. Se conoce como "Clear Coat". Se utilizan 3 robots automatizados y cada bastidor deberá pasar por los 3.
- 4) Curado en Horno.

El Equipo:

Los robots:

Se clasifican por su modo de operación:

- a. En Campana: El aplicador de pintura del robot tiene forma del perímetro de un círculo, lo que permite hacer una aplicación general de la pintura abarcando un gran espacio.
- b. En Pistola: Ayuda a que la capa sea más profunda, es decir, que pueda aplicarse el pigmento en los espacios con mayor nivel de profundidad. El alcance de la profundidad es una relación proporcional al espacio disponible para la aplicación, es decir un agujero o forma pequeña no permite pintar a mucha profundidad.

Ambos tipos de robots están preparados para manejar hasta 50 colores distintos en el brazo de aplicación que es la variedad completa de los colores solicitados por los clientes.

El tiempo de purga para cambiar de un color a otro es de máximo 3 segundos. El brazo del robot está conectado mediante mangueras muy finas a los contenedores de pintura. La purga se efectúa solo en el brazo, por eso el tiempo de limpieza es tan corto.

En la aplicación de la pintura se utiliza el efecto Faraday. Los bastidores de metal a través de un campo electromagnético hacen que la pintura sea atraída al parachoque.

Existe un coeficiente de transferencia, es decir un porcentaje de la pintura que se queda en la superficie y un porcentaje de pérdida. La pérdida es recogida en una fosa de lodos en el piso por donde pasa el transportador.

Cada robot tiene un rango de operación específico el cuál es de 1.8 X 2 metros. Este rango o ventana determina el alcance máximo que puede ser pintado en este proceso. Los valores son dados, sobre todo la altura, por las condiciones ergonómicas requeridas considerando las medidas del trabajador promedio.

Existe un robot para pintar prototipos. Se le conoce como "Teach Robot". Las actividades de prueba de nuevos diseños requieren que se asegure que el producto cumpla las normas de calidad. Este robot

ayuda a definir los movimientos que el brazo realizará para cubrir toda la superficie del parachoque con las especificaciones de grosor de capa establecidas en cada modelo.

Los bastidores (Jigs):

Deben ser lo suficientemente fuertes para soportar el peso del producto, cada parachoque pesa entre 4 y 6 kilogramos. En cada bastidor se montan entre 2 y 4 parachoques. Se utilizan de dos tipos:

- a. *Wyer Prime Jig*. Bastidores elaborados en alambre de acero grueso y se utilizan para procesos de pintura a baja temperatura. Baja temperatura es el proceso de pintado a 80 grados Centígrados o menos.
- b. *Casting Jig*. Bastidores fabricados en acero y tienen la forma interior del parachoque. Incluyen perforaciones o agujeros (*lighter holes*) que hacen que el mismo sea más ligero y fácil de manipular por los operarios. Se utilizan para procesos de alta temperatura en el horneado. Alta temperatura se considera en esta planta al rango de 80 a 120 grados Centígrados. La diferencia con el *Wyer Prime Jig* es que el *Casting Jig* permite al parachoque conservar la forma del molde dentro del horno a temperaturas altas. Es importante considerar que la pieza tiene un coeficiente de contracción del 4% la cual es considerada desde el diseño del vehículo para conservar las dimensiones requeridas y que tienen para estas piezas una tolerancia de +/- 1 mm.

El transportador de circuito cerrado (*Closed Loop Conveyor*):

Tiene capacidad para 220 bastidores. El equipo es fijo en el número de posiciones disponibles. La distancia entre cada poste es de 3.5 metros. La velocidad es fija y no depende del tipo de bastidor instalado. El tiempo para que una posición complete el circuito es de 4.5 horas aproximadamente. Lo que da un tiempo de 1.2 minutos por bastidor en cada etapa del proceso. Este es el tiempo del que dispone el operario para montar y desmontar los parachoques que como se ha mencionado son de 2 a 4 por bastidor. La altura y la distancia que tiene que recorrer el trabajador está diseñada para que pueda efectuar como máximo 500 repeticiones por turno, por lo que la ergonomía en esta etapa ha sido diseñada de acuerdo a estándares internacionales.

El horno de curado:

Se detiene una sola vez a la semana por el coste de arranque y operación. Para el curado de la pintura utiliza el rango entre 80 y 120 grados Centígrados. El tiempo de duración de cada bastidor con sus parachoques correspondientes es de 30 minutos. La longitud del horno son aproximadamente 90 metros con lo que se tienen 25 de los 220 bastidores en proceso de curado simultáneo en el transportado. Cada 1.2 sale un bastidor con producto terminado y al mismo tiempo está entrando otro.

Problemática en la planta Plastic Omnium de Silao:

- 1) La secuencia es definida por el cliente y así se debe hacer la entrega. El orden de fabricación del proveedor no debe afectarla.
El cliente indica la cantidad en firme por modelo y color con una antelación de 8 semanas. Algunos cambios pueden introducirse como máximo 6 horas antes por comunicación electrónica.
Es posible justificar una secuencia óptima para el proceso de pintura que minimice los espacios vacíos si no se afecta la secuencia requerida por el cliente. Definir una secuencia óptima en el proceso de pintura no cambiaría la decisión del cliente sobre el orden de entrega en el que requiere el producto. Una opción para cumplir el orden del cliente puede ser mantener un inventario mínimo de algunas combinaciones de modelo y color de los parachoques.
En los casos en que pintar un color distinto en la secuencia implica dejar un bastidor sin producto para evitar defectos en la pintura, los espacios sin producto se pueden minimizar asegurando el suministro en la secuencia requerida con producto tomado del inventario mínimo.
- 2) El coste de dejar un espacio sin producto es alto, reduce la productividad y genera desperdicios de energía y mano de obra no utilizada.
- 3) El cambio de color no implica en forma natural la pérdida de un espacio. Este se genera por los defectos ocasionados por la secuencia entre colores específicos.
- 4) El cambio de modelo de una posición implica que el trabajador tenga una carga de trabajo adicional en el tiempo especificado por posición. La carga se genera no solo por cambiar los parachoques, sino por desmontar un bastidor y montar uno distinto.

2.2 DEFINICION DEL PROBLEMA

El problema que se analiza en esta tesis puede enunciarse de la siguiente manera: En un sistema de ciclo cerrado secuenciar un grupo de productos. Estos son definidos por su geometría y por su color y con diferentes cantidades de demanda cada uno. En cada posición del sistema solo se puede programar una unidad. Se debe minimizar el número de cambios de color entre unidades consecutivas y el número de cambios de bastidores o perchas en cada posición en ciclos consecutivos. Los costes asociados por cambio de color son a causa del uso de solventes principalmente y los costes por cambio de jig o bastidor se tienen por el uso de mano de obra. En este problema el cambio de geometría en unidades de producto consecutivas no es relevante.

2.2.1 Generalidades

Partiendo de la definición del problema se puede observar una limitante importante propia de los sistemas de producción sincronizados: no todos los procesos son completamente flexibles. Cuando se incrementa la complejidad por aumentar el número de procesos sincronizados el impacto generado en la programación es relevante en mayor o menor medida. Una restricción en la secuenciación de un proceso se convierte en una limitante para el sistema

completo. En un sistema Pull esto es evidente al recibir del cliente una determinada señal de demanda o especificación. Ver Figura 8.

Lo anterior es una de las razones principales por la que no se logra la sincronización con los demás procesos en el ensamblaje final. Cada una de las restricciones que se añaden a la estructura de la secuenciación del sistema tiene un impacto negativo en la reducción del número de cambios en la secuenciación establecida.



Figura 8: Esquema de flujo de requerimientos en un sistema Pull.

Uno de los objetivos en la industria relacionada con los procesos automotrices es el disminuir el impacto de los procesos poco flexibles en la cadena de suministros y mantener los beneficios logrados por las actuales filosofías de producción (como el bajo nivel de inventarios requeridos) y al mismo tiempo establecer y mantener un bajo coste.

Las líneas de producción flexibles con una secuenciación sincronizada “Justo a Tiempo” requieren una mayor cantidad de ajustes del proceso. En los sistemas de producción con transportadores en circuito cerrado estos ajustes provocan incrementos no deseados en algunos factores tales como: el riesgo para el personal, el tiempo de proceso, el trabajo en mano de obra y uso de tiempo máquina, el costo y la complejidad del sistema, así como el número de cambios en la secuenciación. El retraso en el tiempo de entrega de una parte en alguno de los procesos sincronizados puede generar incrementos significativos en los niveles de inventario y reducciones en los niveles de eficiencia.

La propuesta de solución para este problema es determinar una secuencia alternativa al orden general de producción que minimice el número de cambios de modelo y de color en una línea secundaria en un sistema de producción con transportadores en circuito cerrado. Ver Figura 9.

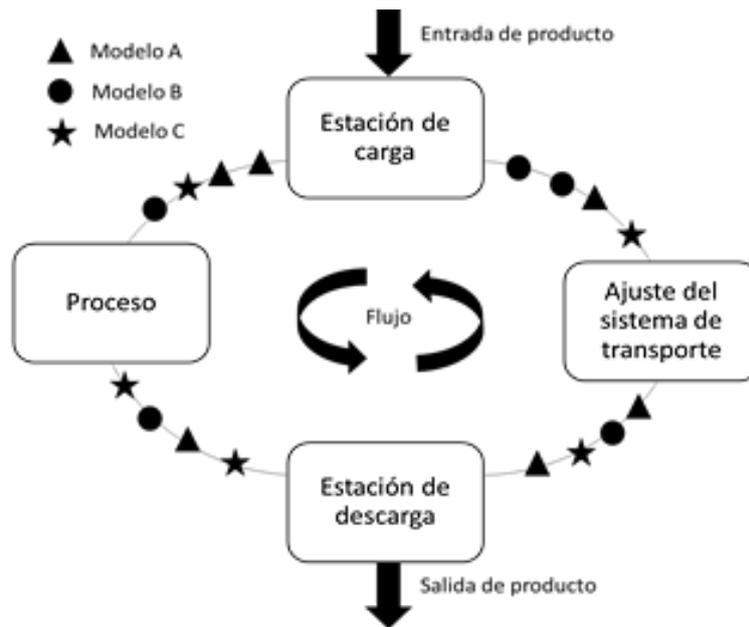


Figura 9: Esquema básico de un proceso con transportadores en circuito cerrado

Consideremos una línea de producción con un sistema de transporte en circuito cerrado compuesto por distintos modelos de transportadores intercambiables correspondientes a una tarea específica. Este sistema cuenta con una cantidad definida de transportadores y una estación de ajuste de modelos.

Hay una serie de características que se tienen que tomar en cuenta para identificar y diferenciar el problema:

- La línea de producción a secuenciar es una línea secundaria. Esto significa que la demanda exacta ya está definida y no se consideran las posibles restricciones de los procesos paralelos.
- El tiempo del proceso es despreciable. Se considera que las tareas requieren el mismo tiempo independientemente del modelo del producto, por lo cual el modificar la secuencia no impacta en el balanceo de la línea.
- La entrega tardía o temprana de producto no está permitida y se debe cumplir toda la demanda.
- Se considera un cambio de modelo o *setup* de modelo cada vez que un transportador requiera ser cambiado por otro.
- Se considera un cambio de color o *setup* de color cada vez que se procesa un producto que debe ser pintado con un color distinto.

Problemas como el que se presenta pueden ser clasificados por su complejidad para ser resueltos con un algoritmo computacional en tiempos polinomiales. La idea intuitiva de problema "difícil de resolver" queda reflejada en el término científico *NP-hard* utilizado en el contexto de la complejidad

algorítmica. En términos coloquiales podemos decir que un problema de optimización difícil es aquel para el que no podemos garantizar el encontrar la mejor solución posible en un tiempo razonable. Su complejidad puede ser probada matemáticamente o conocida en base a problemas similares ya estudiados. De forma general se pueden clasificar los problemas *NP-hard* en los siguientes tipos o áreas de estudio:

- Teoría de grafos
- Diseño de redes
- Conjuntos y particiones
- Almacenaje y recuperación
- Secuenciación y programación
- Programación matemática
- Álgebra y teoría de números
- Juegos y acertijos
- Lógica
- Autómatas y lenguaje
- Optimización de programas

Dadas las características del problema ya expuestas se puede clasificar como un problema de secuenciación y programación o programación matemática.

2.2.2 Delimitación del problema. Hipótesis básica.

En esta tesis, buscando la generalidad, se hará referencia al problema mencionado utilizando un término en inglés junto con su acrónimo:

“CBBSSP” – Cyclic Batch and Bi-Objective Setups Scheduling Problem

El problema de establecer una secuenciación alternativa de producción en un sistema JIT no es sencillo. En la práctica existen una gran cantidad de consideraciones y limitaciones que hacen de la aplicación de una solución, un proyecto demandante en términos de tiempo y recursos.

La solución de un problema particular puede partir de la modelación y análisis de sus características generales, lo cual se intenta en esta investigación. La hipótesis principal se plantea como sigue:

“Se puede generar una secuenciación alternativa de producción en la que se reduzcan el número de cambios de modelo y el número de cambios de color en un transportador de circuito cerrado mediante el desarrollo de heurísticas aplicables al proceso”.

Se generan algunas preguntas cuyas respuestas son relevantes para la investigación:

- ¿Puede generarse un modelo matemático que minimice los ajustes de proceso ocasionados por las variables número de cambios de modelo y número de cambios de color en forma simultánea?
- ¿Puede generarse un segundo modelo matemático que minimice el número de espacios vacíos que se generen por los ajustes de proceso ocasionados por los cambios de modelo y cambios de color?

- ¿Puede obtenerse de forma precisa la frontera de Pareto para el primer caso que es bi-objetivo?
- ¿Qué grado de reducción en el número de cambios se obtiene al implementar las heurísticas que sean desarrolladas?
- ¿Cuál es el factor que más altera la cantidad de ajustes en el proceso?

2.2.3 Objetivos del problema: variantes del CBBSSP

El objetivo principal de esta investigación es el desarrollar un modelo de programación matemática que resuelva el problema de secuenciación de la producción de un sistema JIT con transportadores cíclicos, minimizando la cantidad de ajustes de proceso. En un inicio la propuesta involucraba solo el caso del problema de resolver la minimización de ajuste sea por cambio de modelo o por cambio de color. Dado que en el año 2014 se publicó un artículo sobre el tema, el cual se presenta en la siguiente sección, se han ampliado los objetivos de esta Tesis añadiendo una restricción adicional: considerar la generación de espacios vacíos cuando se presentan los ajustes en la secuenciación. De esta manera los problemas analizados y de los que se hacen propuestas son dos: revisar el caso sin huecos y revisar el caso con huecos. Para efectos de la Tesis se utilizará la siguiente nomenclatura:

- 1) "CBBSSP-SH": Cyclic Batch and Bi-Objective Setups Scheduling Problem without Holes. Problema de Programación de Lotes Cíclicos con Tiempos de Preparación Dependientes e Independientes de la Secuencia Sin Huecos.
- 2) "CBBSSP-CH": Cyclic Batch and Bi-Objective Setups Scheduling Problem with Holes. Problema de Programación de Lotes Cíclicos con Tiempos de Preparación Dependientes e Independientes de la Secuencia Con Huecos.

Los objetivos específicos de esta Tesis son:

- Diseñar un modelo matemático para el problema CBBSSP-SH que minimice el número de ajustes de proceso generados por los cambios de modelo y los cambios de modelo en forma simultánea.
- Diseñar un modelo matemático para el problema CBBSSP-CH que minimice el número de espacios vacíos generados por los ajustes requeridos en el proceso cuando existen cambios de modelo y/o cambios de modelo.
- Para cada variante:
 - Diseñar un modelo matemático que pueda determinar cuáles son las soluciones que minimizan los ajustes por cambio de color y/o de modelo considerando según sea la variante, solo los ajustes o los espacios vacíos generados por los ajustes.
 - Programar e implementar una herramienta computacional capaz de resolver el problema de forma exacta.
 - Desarrollar y evaluar heurísticas que minimicen el problema mediante experimentación.
 - Evaluar el impacto de los factores involucrados en la complejidad y valor de la solución de cada modelo propuesto.

- Proponer futuras líneas de investigación.

2.3 REVISION DE LA LITERATURA PARA EL PROBLEMA CBBSSP

2.3.1 Referencias encontradas en la literatura

Como se ha mencionado desde el Capítulo de Introducción. Se ha hecho una revisión de la literatura y en el Estado del Arte se hace una selección de los temas afines a esta Tesis para elaborar una revisión completa de los mismos. Al inicio del proceso de elaboración de este trabajo solo existía una investigación publicada sobre el tema, la cual fue desarrollada por los tutores de esta Tesis (2). En 2014 apareció una publicación que también revisa el tema central. La referencia se encuentra en (3).

2.3.2 Diferencias de las referencias existentes respecto al planteamiento de esta Tesis sobre el CBBSSP.

En (2) García-Sabater et. al. hicieron la primera aproximación al problema central de esta Tesis en el Onceavo Taller Internacional de Administración de Proyectos y Programación en Estambul 2008 (Proceedings of the 11th International Workshop on Project Management and Scheduling). En esta referencia se hace una descripción clara del problema CBBSSP sin Huecos, introduce una formulación matemática y resuelve una instancia del problema con software comercial.

Ganguly y Laguna en (3) hacen la segunda publicación en una revista sobre el caso específico del problema bi-objetivo con tiempos dependientes e independientes de la secuencia. En este artículo, se hace un modelo matemático general que incluye el concepto de variación en el coste por variable es decir el coste de hacer un cambio de un color específico a otro en forma horizontal y el coste de hacer un cambio de modelo específico a otro en sentido vertical. La forma de abordar la resolución del problema es con dos enfoques:

- 1) Utilizando una adaptación de OptQuest.
- 2) Utilizando Búsqueda por Entorno Variable (VNS, *Variable Neighborhood Search*).

2.4 CONCLUSIONES

En este capítulo se ha presentado el problema general de esta Tesis utilizando como nomenclatura para denotarlo "CBBSSP". Se ha justificado la separación del problema básico en el análisis de dos casos. El primer caso es considerar el problema CBBSSP sin Huecos donde el enfoque es hacer un

análisis bi-objetivo buscando minimizar las variables Número de cambios de modelo y Número de cambio de color. El segundo caso es mono-objetivo pues reduce la situación de existencia de cambios sean estos por ajustes de modelo o de color a la generación de espacios vacío en la secuencia. El objetivo se enfoca a la minimización de espacios vacíos o huecos en la secuencia óptima. Este capítulo sirve de marco a la tesis para justificar los tres capítulos siguientes. El capítulo 3 que busca en la literatura los temas afines para crear un Estado del Arte, el capítulo 4 que busca resolver el Problema CBBSSP Sin Huecos, y el capítulo 5 que tiene como objetivo resolver el Problema CBBSSO con Huecos.

3 ESTADO DEL ARTE

3.1 INTRODUCCION

En el presente capítulo se busca dar el contexto conceptual sobre el problema generando un estado del arte general sobre los modelos, algoritmos y métodos utilizados para resolver problemas de secuenciación en los que se consideren restricciones dadas por características particulares y generales de los sistemas productivos. El problema de las líneas de pintura en la industria automotriz con lotes cíclicos y tiempos de preparación, dependientes e independientes de la secuencia, ha sido estudiado previamente solamente en dos casos, siendo la primera ocasión en 2008 cuando se presenta en un congreso en Turquía (2) por los tutores de esta Tesis y el segundo caso en 2014 por Ganguly y Laguna en (3). Además de la revisión de los artículos citados se busca una aproximación al problema a través de la descomposición de este tipo de sistemas en elementos que son comunes con otros sistemas productivos. Se revisarán aquellos aspectos relevantes y la forma en que han sido abordados en otros estudios con temas en los que algún aspecto del problema ha sido analizado. También se consideran los temas que por la definición presentada en el capítulo 2 forman parte del área de exploración del problema.

De acuerdo al análisis de la Definición del Problema, se consideran como temas relevantes los que se presentan en la Figura 10. En esta figura se menciona también cuál es la importancia de cada uno de estos aspectos.

La profundidad en el análisis de los temas no es la misma en todos los casos. La razón para hacer esta distinción es que solo se toman en cuenta los aspectos relevantes al problema. Se hace hincapié nuevamente que en esta Tesis son dos los problemas analizados. El caso de la secuenciación sin huecos y el caso con huecos. Esta distinción tiene también impacto en la selección temática. En el caso del problema sin huecos la forma de explorar soluciones es buscando optimizar un problema bi-objetivo: la minimización de los cambios los cuales tienen dos naturalezas diferentes entre sí, los cambios de geometría o modelo y los cambios de colore. En el caso del problema con huecos no se tiene la situación de análisis bi-objetivo, sino mono-objetivo. La solución del problema busca minimizar la generación de espacios vacíos sea esta ocasionada por cambios de color o cambio de modelo.

En otras palabras, en la resolución del problema con huecos se reduce la optimización a la minimización de los espacios vacíos, los cuales se generan por las dos mismas causas que en el problema sin huecos. Dada esta situación la profundidad de opciones de análisis bi-objetivo en el problema con huecos no es relevante y en el Estado del Arte se presentará tan solo una exploración general.

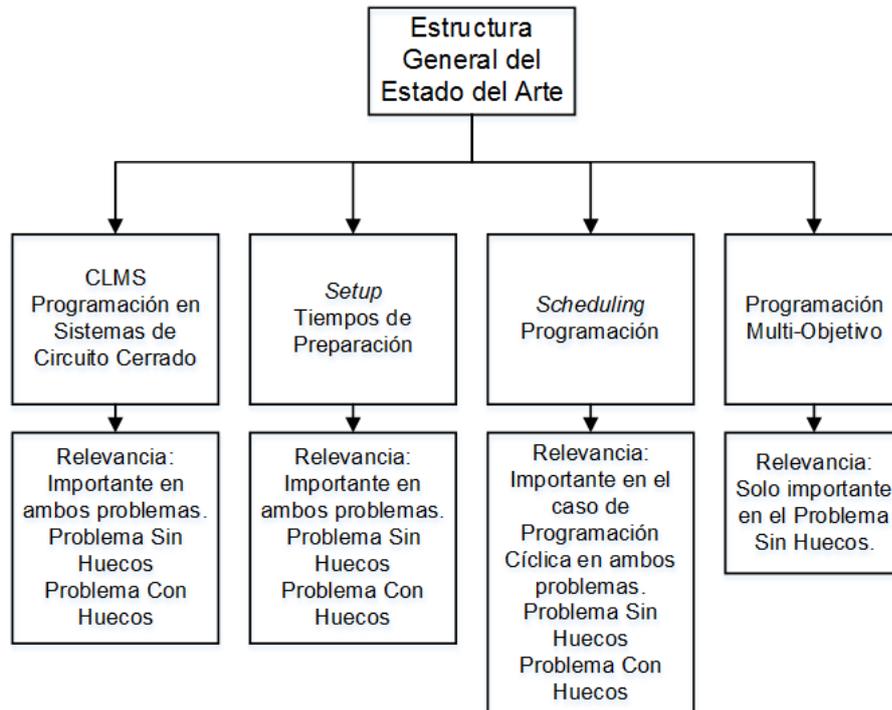


Figura 10: Estructura General del Estado del Arte

Se define la estructura del presente capítulo considerando los siguientes aspectos relevantes:

- 1) CLMS. En el próximo apartado se muestra la definición de los Sistemas de circuito cerrado y las investigaciones más importantes sobre los mismos.
- 2) Setups. Por las características de los dos problemas es necesario revisar el problema de secuenciación con tiempos de preparación o *setups* y la forma en que se han estudiado en otras investigaciones las características y aspectos que tienen impacto en la secuenciación.
- 3) Scheduling. La programación se revisa en el tercer apartado. Para contextualizar el tipo de programación o *scheduling* se presenta una clasificación de este concepto y se hace mención de la parte relevante a los problemas de la Tesis que es sobre todo la programación Cíclica que es a la que se da mayor relevancia.
- 4) Programación Multi-Objetivo. Se revisa en forma general este concepto que es importante para el Problema Sin Huecos. Se consideran los métodos y algoritmos más frecuentes en la literatura.

3.2 Programación en Sistemas de Circuito Cerrado:

3.2.1 Introducción:

Los sistemas de manufactura de circuito cerrado son ampliamente usados en la industria, en sectores tales como la industria automotriz, la

fabricación de semiconductores y el ensamble de aparatos eléctricos y dispositivos electrónicos. Existen varios factores que impactan el desempeño de estos sistemas. Los factores más relevantes son: la capacidad de los bastidores, tarimas o dispositivos de soporte, el número de estos dispositivos, la confiabilidad de la máquina, el tiempo de proceso, la velocidad del equipo y la posición de los puntos de carga y descarga. Estos factores hacen que el análisis preciso de estos sistemas sea demandante y un reto para su optimización. En esta sección se presentan las características principales de estos sistemas, los métodos utilizados para su análisis, las dificultades que se presentan en la aplicación de estos y la literatura que existe sobre el tema. A continuación, en la Figura 11 se presentan las principales investigaciones en artículos publicados sobre el tema de Sistemas de Circuito Cerrado que son relevantes a los problemas analizados en esta Tesis.

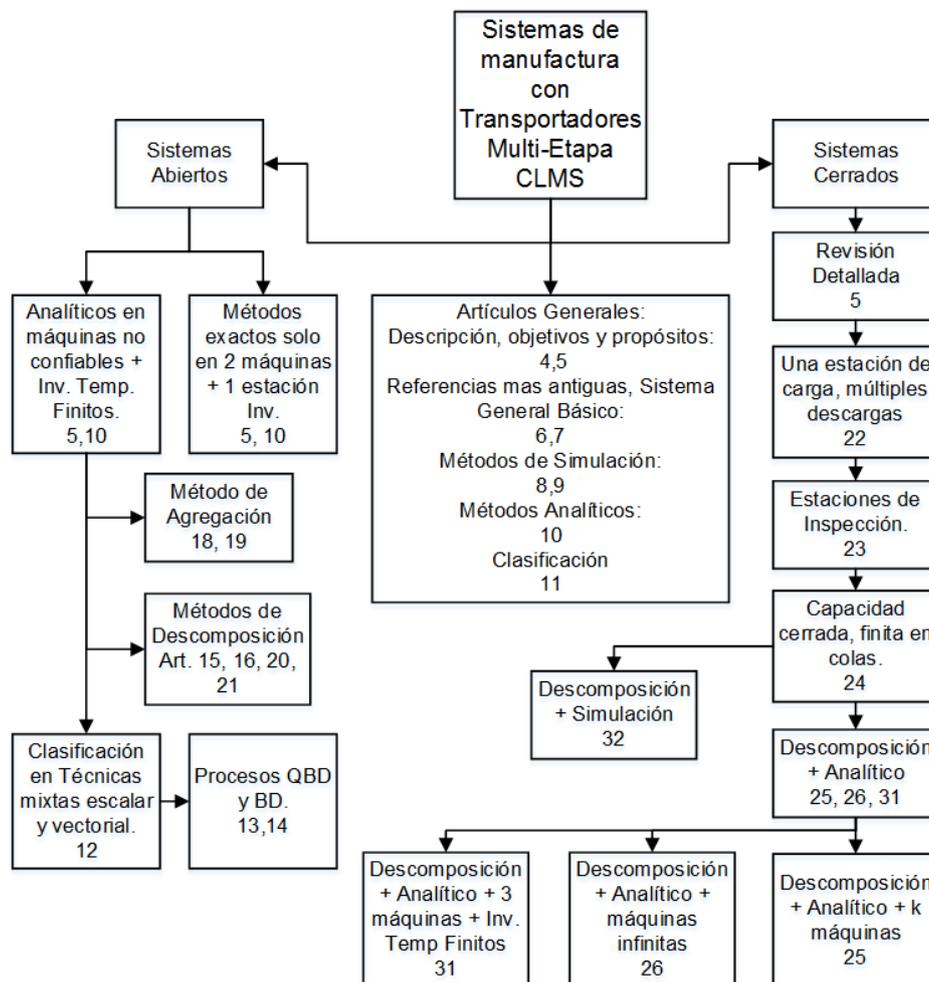


Figura 11: Revisión de Literatura en Programación de Sistemas de Circuito Cerrado

3.2.2 Descripción del sistema de circuito cerrado:

Los sistemas de manufactura en circuito cerrado son descritos en la literatura en inglés con las siglas CLMS (*Closed Loop Manufacturing Systems*) (4). Estos sistemas son utilizados en una gran variedad de sectores industriales. Entre otros, se conocen la industria de manufactura de automóviles, tanto en la parte de ensamblaje como en la de pintura de

autopartes, la fabricación de semiconductores, el ensamblaje de aparatos eléctricos y electrónicos tales como teléfonos móviles o fijos, televisiones, equipo de sonido, articulas para el hogar. etc.

Uno de los propósitos más importantes al utilizar equipos con sistemas de manejo de materiales en circuito cerrado es mover las partes en forma confiable y segura a lo largo de las distintas etapas del proceso al mismo tiempo que se mejora el uso del sistema de transporte, se ahorra espacio, se reduce el tiempo de *setup* y se reduce en conjunto el coste de manejo de materiales (4). Se pueden utilizar circuitos adicionales para realizar retrabajos, lo cual contribuye a mejorar la calidad y el rendimiento del sistema (5). Esto se hace al permitir que existan puntos de carga y descarga adicionales o en algunos casos al utilizar los existentes para hacer este ciclo adicional.

El sistema básico CLMS considera el siguiente proceso:

- 1) Las partes son colocadas en un punto de carga utilizando para el efecto, transportadores o dispositivos de carga tales como tarimas, contenedores, bastidores, carros o perchas. Para el efecto de esta tesis se utilizará el término "*bastidores*".
- 2) Las piezas se mueven colocadas en los bastidores a través de un sistema transportador específico, del punto *A* al punto *B* sin intervención manual. El sistema de transporte puede ser un sistema de rodillos automatizado, un sistema de rodillos con banda, un transportador de cadena u otras formas de estructura.
- 3) Cuando todas las partes han terminado con todas las operaciones requeridas al final de la línea, se descargan del bastidor y salen del sistema o salen del sistema junto con el bastidor y se descargan cuando ya no están circulando.
- 4) Los bastidores vacíos continúan en el sistema y regresan al sitio de carga inicial en forma automatizada o si fueron desmontados en el sitio de descarga junto con las partes son trasladados a un punto de entrada (este puede ser el descrito en el paso 1 o en uno previo). De esta manera queda completo un circuito de manejo de materiales.

Es importante mencionar que un sistema de producción puede tener más de un circuito que separa el sistema general en varias etapas y esto depende de la configuración del diseño del proceso.

Muth Eginhard en 1974 (6) presenta algunos de los artículos más antiguos que hacen referencia al tema de los transportadores de circuito cerrado. Cita a (Kwo, T.T., "A Theory of Conveyors", *Management Science*, 6, 1, Octubre, 1958) por haber desarrollado una teoría intuitiva, Los resultados que obtiene fueron interpretados en el libro de Morris en 1962 (Morris, William T., *Analysis for Materials Handling Management*, Richard D. Irwin, Inc., Homewood, Illinois, pp. 129-138, 1962.) Ambas referencias fueron buscadas pero no están disponibles.

El trabajo de Muth Eginhard había iniciado con un análisis general sobre los sistemas de transporte de circuito cerrado en (7). En este artículo considera el caso del manejo de materiales con flujo continuo, hace referencia a un sistema con una estación de carga y una de descarga de partes. La operación del sistema tiene variación en el tiempo de entrada y en la tasa de salida. Sin embargo el flujo es continuo. La frecuencia de respuesta describe el proceso. Se presentan gráficamente algunas soluciones para casos específicos.

En (6) Muth presenta el caso de flujo discreto, es una continuación del problema presentado en (7). El modelo esquemático que describe la forma general de los sistemas de manufactura con transporte en circuito cerrado se muestra en la Figura 12. La parte móvil del transportador se conoce como “*banda de transporte*” (*belt* en inglés). Esta banda tiene un movimiento continuo con una velocidad constante v . El material es transportado en puntos equidistantes a lo largo de la banda de transporte. A estos puntos se les llama “*cangilones*” (*buckets*). El número de cangilones en la banda transportadora es k y la capacidad en término de material por cangilón es C . El material ingresa al cubo en la estación de carga A y deja el sistema en la estación de descarga B . La distancia entre las estaciones de carga y descarga, A y B , medidas a lo largo de la banda de transporte es L_1 . Así cada cangilón se mueve de la estación A a la estación B en el intervalo de tiempo: $T_1 = L_1/v$. La distancia de regreso de la estación B a la estación A es L_2 , y el intervalo de tiempo correspondiente es: $T_2 = L_2/v$. Se asume que la entrega de material en la estación de carga A y la demanda para el material en la estación de descarga B están sincronizadas con el tiempo de paso de los cangilones. Esto significa que el material llega a la estación A en puntos discretos en el tiempo y es descargado de la estación B en puntos discretos en el tiempo. No hay espacios o instalaciones disponibles para almacenamiento temporal ni en la estación A ni en la B . Por tanto, la operación del transportador debe ser compatible con la oferta de materiales del proveedor o almacén de materiales y con la demanda. Debe haber suficiente capacidad en la banda transportadora con los cangilones para cargar y mover el material que llega a la estación A al mismo tiempo que suficiente material o partes deben ser entregados en la estación B para satisfacer la demanda.

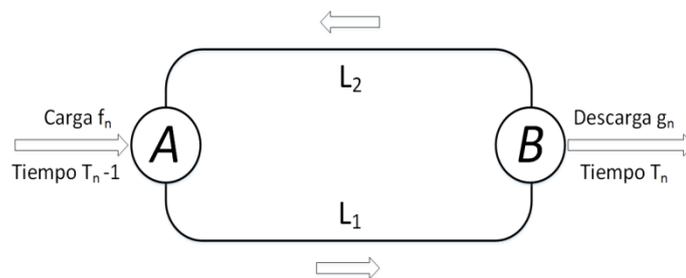


Figura 12: Sistema General de Circuito Cerrado

Para establecer la notación en (6) se considera un origen arbitrario para el tiempo t y después de un tiempo cero, se cuentan todos los cangilones que pasan por la estación B . La cantidad de material requerido en la estación B en el tiempo t_n se denota con g_n . El n -ésimo cangilón pasa por la estación A en el tiempo $t_n - T_1$. El total del material entregado en la estación A en el tiempo $t_n - T_1$ se denota por f_n . Se puede observar que la cantidad f_n se carga en la estación A en el mismo cangilón del cual la cantidad g_n se va a retirar T_1 unidades de tiempo después.

El flujo de materiales entre la estación A , donde se efectúa la carga y la estación B que es donde se descarga está representado por la secuencia (f_n) y (g_n) . Se tienen dos casos:

- 1) Caso continuo: Cuando el flujo de material se entrega en una tasa uniforme, es decir $f_n = g_n = \text{constante}$.
- 2) Caso discreto: Cuando el flujo de material tiene patrones variables.

Existen múltiples factores que impactan el rendimiento de los CLMS, algunos de ellos son: la capacidad total de bastidores en el sistema (el número máximo de bastidores que el sistema tiene o puede tener), el número real de bastidores liberados en el sistema, el índice de velocidad del bastidor, las posiciones de carga y descarga (A y B), la velocidad de proceso de la máquina, la confiabilidad de la máquina, etc. La evaluación del desempeño de los CLMS (Rendimiento, inventario de trabajo en proceso, retrasos, tiempo de ciclo total, etc.) es un trabajo difícil por la compleja interacción de varios parámetros y múltiples elementos. Existen situaciones como la falta de confiabilidad de la maquinaria y equipo, bastidores y tarimas compartidas, partes sin terminar y partes terminadas, longitud limitada del transportador, etc. Sin embargo y a pesar de estas dificultades es crítico y necesario hacer un análisis preciso de estos factores para el óptimo diseño de los CLMS, así como la mejora continua y el control de los mismos.

Los métodos que se han utilizado para estimar y analizar el desempeño de los CLMS son principalmente dos: Simulación y Métodos Analíticos. Los experimentos de simulación han sido ampliamente utilizados en el análisis de sistemas de manufactura por su robustez y la capacidad de modelar procesos y sistemas complejos y grandes. En (8) se puede encontrar una revisión detallada y completa en el uso de la simulación en el diseño y operación de los sistemas de manufactura. Sin embargo es importante considerar que el desarrollo y experimentación de los modelos de simulación consumen mucho tiempo de planeación, son más caros y requieren un tiempo de ejecución prolongado para obtener resultados que sean estadísticamente válidos (9). Por otro lado, un método de modelación analítico y accesible puede facilitar resultados que sean consistentes y rápidos, y además proporcionar los fundamentos o la razón subyacente del funcionamiento del desempeño del sistema de forma más eficiente y concisa que otros métodos (10).

En la literatura se puede encontrar que se ha hecho un trabajo significativo utilizando métodos analíticos para evaluar el rendimiento del desempeño de sistemas de manufactura abiertos con máquinas no confiables y con inventario finito. Un sistema de manufactura abierto se considera aquel en el que se tiene una línea serial en el que las máquinas e inventarios temporales están relacionados a lo largo de un solo patrón de flujo uno después de otro. Otro caso que también se considera abierto es cuando se tiene una estructura de árbol de un solo sentido. En este caso existe solamente una secuencia única que conecta dos estaciones, sean estas máquinas o inventarios temporales en el sistema. Los sistemas de manufactura abiertos incluyen líneas de transporte, sistemas de ensamble o desensamble, líneas en paralelo, sistemas que se dividen en varias líneas, líneas que se unen o combinan, etc.

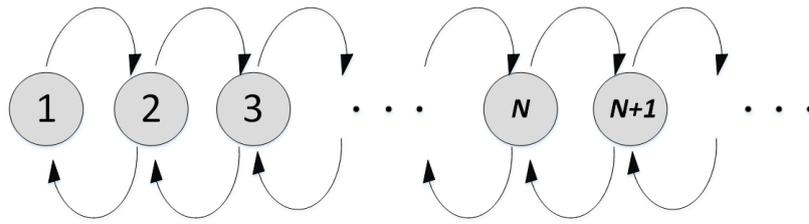
Como contraparte a los sistemas abiertos, existen los sistemas de ciclo cerrado, en los cuales existen puntos de reentrada. Estos puntos crean circuitos cerrados y hacen imposible clasificar y ordenar las máquinas con base en una relación de etapas proveedor-cliente. En términos en inglés hablan de ordenar con base en una relación *upstream-downstream*, que se refiere a la obtención de materias primas en el *upstream* que son transformadas en el

downstream, es decir la secuencia natural de transformar materiales en producto terminado. En el libro de Koren (11) se encuentran métodos efectivos para clasificar la configuración de los sistemas de manufactura y se estudian los impactos que estas generan en la productividad, la capacidad de respuesta, convertibilidad y escalabilidad. Los autores en (10) y en (5) proporcionan revisiones extensas y amplias en el análisis de sistemas abiertos de manufactura con máquinas no confiables e inventarios temporales finitos. Las soluciones exactas solamente están disponibles para el caso simple de una línea con transporte de dos máquinas y una estación de inventario temporal.

Este caso incluye modelos de tiempo de procesamiento exponencial, modelos de flujo continuo y modelos determinísticos.

En (12) Yang et al. utilizan técnicas mixtas escalar y vectorial para modelar ecuaciones genéricas de balanceo de líneas de transporte multi-etapa y estudiaron sus propiedades en forma acotada a este caso.

Liu et al. en (13) modelaron y analizaron el rendimiento de un sistema de manufactura con dos etapas y un inventario temporal de tamaño finito entre cada etapa. Las etapas de manufactura se acomodan en paralelo con múltiples máquinas en cada una, independientes y no confiables. En la literatura es frecuente el uso de variables aleatorias binarias para modelar máquinas no confiables en las líneas de transporte y en algunos otros sistemas de producción. En este artículo se muestra la importancia de utilizar más de dos estados para modelar sistemas de máquinas no confiables en paralelo. La razón para esto último es que las operaciones en estos sistemas son asincrónicas e independientes. Las ecuaciones de balanceo del sistema fueron formuladas con base en un conjunto de nuevas notaciones de manipulaciones de vector, y fueron transformadas en una matriz cumpliendo con las propiedades del proceso QBD (*Quasi-Birth-Death*). Se utilizó el método de la *Matriz Analítica* (MA) para resolver el proceso genérico QBD y se calcula con él la probabilidad de tener un estado específico y del rendimiento. Se demuestra a través de casos numéricos que el método de solución es rápido y preciso para analizar sistemas de manufactura paralelos probando así la efectividad y la aplicabilidad del método MA. Estos modelos Multi-estado y sus técnicas de solución pueden ser utilizados para analizar sistemas de manufactura más grandes y complejos. Como complemento a este artículo se hace referencia al artículo de Kharoufeh (14) donde se explica el proceso QBD (*Quasi-Birth-Death*). El proceso QBD es una generalización del proceso Birth-Death. Los procesos Birth-Death son procesos estocásticos del tipo Markov donde cada estado puede ser asociado con un elemento del conjunto que se cuenta, lo cual significa que los diferentes estados puede ser ordenado en forma lineal. Una vez que el orden correcto es establecido, las transiciones pueden ocurrir solamente con los estados vecinos. Lo que quiere decir que si un proceso está en el estado n , entonces solamente puede haber una transición al estado $n+1$ o al estado $n-1$, ver Figura 13.

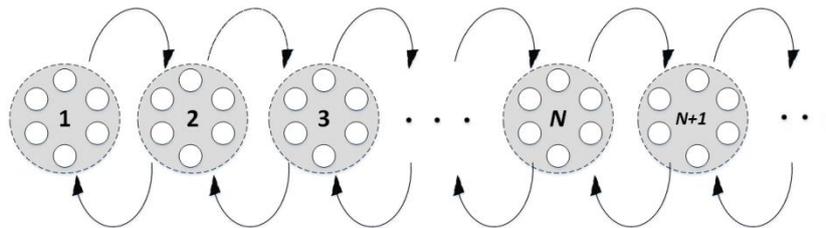


Proceso Birth-Death

Figura 13: Proceso Birth –Death

Adicionalmente, los tiempos de transición están distribuidos exponencialmente lo que permite su clasificación bajo la categoría de procesos de Markov.

En el caso de los procesos QBD, ver Figura 14, en vez de un estado que corresponde a cada elemento del conjunto que se analiza, existe un grupo de estados correspondientes. De esta manera para un grupo de estados específico, el proceso puede tener transiciones al grupo de estados que corresponden al elemento previo del conjunto que cuenta o al grupo de estados correspondiente al elemento siguiente del conjunto que cuenta (15).



Proceso Quasi Birth-Death

Las líneas punteadas representan una colección de transiciones que se originan en un estado de un grupo y terminan en un estado de otro grupo.

Figura 14: Proceso Quasi Birth-Death

En el caso de los sistemas más grandes tales como líneas complejas de transporte o sistemas de ensamble y armado con estructura de árbol, no existen métodos analíticos exactos por el incremento explosivo en los posibles estados del sistema y las relaciones complejas de acoplamiento (4), como se observa en teoría de colas. Se han desarrollado métodos para la evaluación de estos sistemas complejos, se consideran entre estos a los métodos de aproximación que utilizan descomposición y agregación. Se pueden ver en los artículos de (16) que utiliza descomposición y de (5). La idea básica en la descomposición es descomponer el análisis del sistema original en uno con un conjunto más pequeño de subsistemas pero con un comportamiento equivalente que sea más sencillo de manipular matemáticamente. En el caso contrario, la agregación consiste en reemplazar un grupo de máquinas e inventarios temporales por una sola máquina que sea equivalente al sistema original. La máquina equivalente hace la misma o similar operación, el mismo tipo de descomposturas y formas de reparación como la que tiene el grupo de máquinas al cual equivale. Ambos procedimientos: descomposición y

agregación, se realizan a través de métodos recursivos. Si el procedimiento converge, se registran la tasa de producción del sistema y otras medidas de desempeño adicionales.

En (17) se tienen algoritmos recursivos que los autores desarrollaron para generar automáticamente configuraciones del sistema de ensamble con selección de equipo para la manufactura de baterías en el sector automotriz.

Um et al. en (18) presentaron el diseño de la simulación y el análisis de un sistema de manufactura flexible con un sistema de vehículos guiados automáticamente.

Koren y Shpitalni en (19) estudiaron el diseño de un sistema de manufactura reconfigurable,

Huang en (20) señaló que utilizar el rendimiento de la estación promedio puede originar que se haga una sobreestimación en el diseño del sistema de producción.

En Huang et al (21) se estudian técnicas de modelación de sistemas de manufactura con el propósito de mejorar la productividad.

En comparación con los sistemas abiertos, existe un número muy limitado de trabajos realizados en el análisis de sistemas de manufactura con circuito cerrado CLMS:

Li et al. (5) hicieron una revisión detallada de los trabajos anteriores relacionados con el tema de los sistemas de producción con circuito cerrado con un número constante de puntos de transporte en el CLMS.

Bastani en (22) analizó un sistema de transporte de circuito cerrado considerando que este tuviese una sola estación de carga y múltiples estaciones de descarga. Propone una solución analítica para medir los efectos de los tiempos de recirculación de unidades, además de otros parámetros del sistema.

Narahari y Khan en (23) modelaron un sistema de manufactura reentrante con estaciones de inspección.

Bouhchouch et al. en (24) modelaron un sistema de circuito cerrado como una red de cola cerrada con capacidad finita en la cola, y propusieron un método de aproximación para calcular el desempeño de estas redes de colas. Sin embargo, los modelos de colas frecuentemente se modelan como procesos *Poisson* a través del uso de la distribución exponencial para hacer a los modelos tratables analíticamente. Estas limitaciones matemáticas pueden hacer una imitación relativamente pobre de los tiempos constantes de proceso y de otras restricciones en los sistemas de manufactura reales.

Frein et al. (25) presentaron un enfoque de descomposición en este artículo. El objetivo que ellos tuvieron fue proponer un método analítico para evaluar el desempeño de líneas de producción con circuito cerrado con máquinas no confiables e inventarios temporales finitos. Este sistema está compuesto por un conjunto de k máquinas separadas por inventarios temporales de capacidad finita. Existe además un número fijo de tarimas circulando en el circuito. Asumen que las máquinas tienen tiempos de proceso determinísticos pero que están sujetos a fallas. Las fallas y los tiempos de reparación están distribuidos exponencialmente. La aproximación al comportamiento de este sistema es a través de un modelo de flujo continuo. El modelo de flujo continuo a su vez es analizado mediante una técnica de descomposición similar a las que se utilizan en los sistemas abiertos de líneas de producción. En los resultados se puede obtener una aproximación a la tasa

de producción del sistema para líneas homogéneas de circuito cerrado. La situación que su análisis no considera es que no toma en cuenta el número de tarimas en los inventarios temporales reales, y hacen una estimación aleatoria, lo cual lleva a errores en algunas situaciones. Este modelo considera al sistema CLMS como una línea en la cual la primera máquina y la última son la misma. En relación al número de máquinas e inventarios temporales es preciso y funciona bien para sistemas grandes no para sistemas pequeños. En cuanto al tamaño de población los resultados son precisos para tamaños intermedios de partes, el desempeño es inferior en el caso de poblaciones muy pequeñas o muy grandes.

Gershwin y Werner en (26) hicieron una mejora al Sistema de Frein et al. (25), presentaron un nuevo método de descomposición que considera tanto la correlación entre la población y la probabilidad de bloqueo y falta de suministro. Adicionalmente el modelo que ellos desarrollaron permite considerar cualquier número de máquinas en el sistema. Los procesos de manufactura que utilizan tarimas o bastidores pueden ser analizados como circuitos en sí mismos. La razón para esto es que el número de tarimas o bastidores permanece constante en el sistema y están circulando constantemente, no son infinitos. De la misma manera las políticas de control tales como el *conwip* o el *kanban* establecen circuitos conceptuales al imponer una limitante en el número de partes que puede haber en el sistema en todo momento. Los autores afirman que los métodos de descomposición se han desarrollado para hacer la evaluación del desempeño de algunos sistemas muy grandes, con gran cantidad de máquinas para los cuales los métodos analíticos no existían y los métodos numéricos eran inviables. Las técnicas de descomposición aproximan el sistema de gran escala original a un conjunto de sistemas más pequeños que cumplen y satisfacen un conjunto de relaciones o conexiones cuidadosamente elegidas. Las redes de mayor interés para analizar son aquellas con un inventario temporal finito.

Li et al. en (27) estudiaron un Sistema de producción con dos circuitos cerrados. Los sistemas de producción con múltiples circuitos cerrados son representativos de las líneas de ensamble de semiconductores. Las líneas de ensamble en esta industria son la parte final o última etapa en la manufactura de semiconductores. Estas líneas están divididas en dos partes formando dos bucles de circuito cerrado. Las dos partes del sistema están diferenciadas dependiendo del tipo de carro transportador que utilicen para circular por el sistema de producción. Existen ejemplos similares de este tipo de sistemas con dos circuitos cerrados en la industria de fabricación de automóviles y en la industria empacadora de alimentos. Por otro lado, las políticas de control tales como la estrategia de control de segmentos múltiples *conwip* crean también ciclos conceptuales múltiples al imponer limitantes como el número de partes permitidas en el sistema productivo. En este artículo el análisis se hace mediante una cadena de Markov con estructura de bloques. Los autores propusieron una solución de factorización RG para evaluar las medidas de desempeño del sistema incluyendo el rendimiento y el nivel del trabajo en proceso WIP. Con base en esto, el sistema puede ser analizado numéricamente en relación a varios parámetros cruciales. Esto llevo a una mejora en las cantidades de los transportadores los cuales son siempre factores relevantes en las líneas de producción de circuito cerrado.

Un ejemplo de una línea con estas características se puede observar en la Figura 15. Los rectángulos y los círculos representan las máquinas y los inventarios temporales en ese orden. Se asume que existe siempre suficiente materia prima para ingresar a la máquina M_1 . Cada operación en cada una de las tres máquinas debe efectuarse con el apoyo de carros de transporte. Las operaciones en las máquinas M_1 y M_3 requieren los carros A y B respectivamente. La operación en la máquina M_2 requiere los carros A y B simultáneamente. El proceso es como sigue: En la primera estación la materia prima se coloca en un carro vacío tipo A. Este material proviene del almacén temporal B_a y es enviado a la máquina M_1 para que sea procesado. Después de la operación en M_1 , el carro A llena con producto se coloca en el almacén temporal B_1 para esperar la siguiente operación. Si M_2 y un carro B están disponibles cuando el carro A llega, tanto el carro A como el carro B se colocan en máquinas M_2 para una operación asociada. Una vez que la operación en M_2 ha finalizado todos los productos del carro A se mueven al carro B. El carro A entonces regresa al inventario temporal B_a y el carro B lleno con todos los productos se coloca en el inventario temporal B_2 . Una vez que la máquina M_3 está disponible, el carro B sale del inventario temporal B_2 y entra en la máquina M_3 para la operación correspondiente. Después de que se realizan estos procesos, el producto deja el sistema en la máquina M_3 y el carro B vacío regresa al almacén temporal B_b . Todas las máquinas son confiables y nunca producen partes con fallas. Cada máquina puede procesar el producto de un carro a la vez. La máquina M_1 se paraliza si no existe producto en el inventario temporal B_a . La máquina M_2 se paraliza cuando el inventario B_1 o el inventario B_b están vacíos. La máquina M_3 se paraliza cuando el inventario temporal B_2 está vacío. Algunos de los parámetros que son básicos en el modelo descrito son:

- 1) Los tiempos de servicio de las máquinas M_i son mutuamente independientes con distribución exponencial idéntica, donde i es el número de máquina para $i=1,2,3$.
- 2) El número de carros tipo A es m , y el número total de carros tipo B es n . Este sistema incluye 3 máquinas y 4 estaciones de inventario en proceso y forma 2 circuitos cerrados con un número constante de carros.

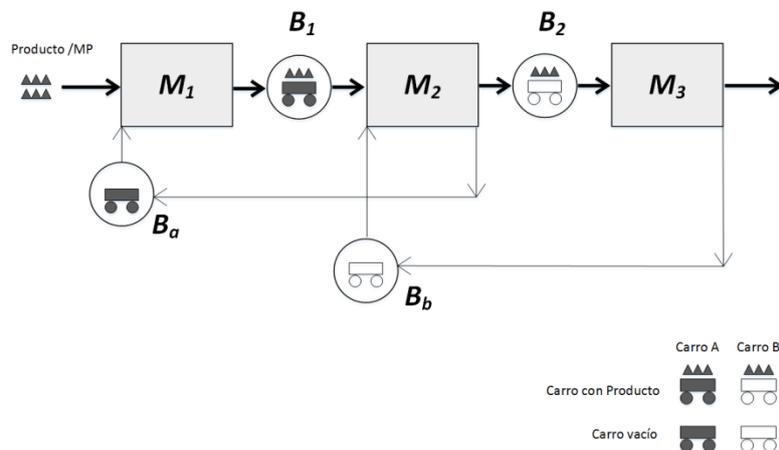


Figura 15: Línea de Producción con dos circuitos y tres máquinas.

Nazzal et al. en (28) propusieron un modelo analítico basado en teoría de colas para diseñar un sistema de manejo de materiales que utiliza un

transportador de circuito cerrado para una empresa del sector de fabricación de semiconductores donde se elaboran *wafers* (obleas de silicio). Los autores utilizan las siglas AMHS (*Automated Material Handling System*) para identificar el tipo de sistema que será una alternativa viable en la industria de fabricación de semiconductores en la siguiente generación. En los últimos años se han desarrollado programas que buscan mejorar los tiempos de ciclo en la fabricación de *wafers* eliminando actividades que limitan el proceso y la productividad del equipo. Los más conocidos son 300 mm *Prime* y *Next Generation Factory* (NFG). Una de las mejoras principales que se plantean en estos programas es producir lotes de menor tamaño. Sin embargo algunos estudios indican que las tecnologías de AMHS representan un detractor potencial en la productividad para lotes pequeños debido a una insuficiente velocidad de entrega de los AMHS y una dependencia marcada en el almacenamiento de unidades. La configuración típica de estos sistemas tiene una línea central que funciona como una columna vertebral (*interbay*) y bucles o ciclos ramificados en ambos lados de la línea (*intrabays*) que dan servicio al equipo de producción colocado entre cada ciclo. Se incluyen espacios para almacenamiento en proceso. El sistema AMHS es extremadamente dependiente de las características y mecanismos de control que tengan los transportadores. Estas características son velocidad, aceleración, desaceleración, limitantes de entrega, etc. La mayoría de los sistemas AMHS para 300 mm utilizan vehículos suspendidos en cadenas en un nivel superior al piso de fabricación conocidos como OHV (*Overhead Hoist Vehicles*). Un sistema AMHS con un número pequeño de vehículos ocasiona retrasos grandes a los lotes que se planea mover, esto en consecuencia implica tiempos de entrega más largos. Por otro lado un exceso de vehículos genera congestión del tráfico de los sistemas *interbay* e *intrabay* pues cada unidad frecuentemente bloqueará a otros transportadores que se mueven en la misma ruta. En (29) se propone el uso de transportadores de flujo continuo CFT para mover *wafers* de 450 mm dado que esta tecnología tiene una gran capacidad de transporte, tiempos de entrega cortos y predeterminados, y costes bajos. Además el CFT puede facilitar almacenamiento temporal de material en forma más cercana a las herramientas o equipos de proceso reduciendo el tamaño de estos inventarios temporales. Una mejora en la logística para estas estos sistemas NFG puede proporcionar una ventaja competitiva en comparación a los sistemas AMHS actuales. El objetivo en este artículo es calcular en forma precisa el inventario de trabajo en proceso WIP (*Work in process*) del transportador, ordenando en colas los retrasos que son ocasionados por la congestión en los puntos de intersección, además de evaluar la estabilidad del sistema transportador. Se utiliza un enfoque de cuatro etapas para estimar el WIP. Los resultados experimentales demostraron que, en el peor de los casos donde se utiliza el máximo número de cruceros y el tráfico de transportadores es alto, el modelo analítico se desempeña muy bien con un promedio relativo de errores del 4.2%. El conjunto de parámetros de entrada en el modelo analítico presentado incluye: los movimientos requeridos, la velocidad del transportador, el *layout* de la estación, retrasos en los tornamesas, el tamaño de la ventana para cada lote y la ubicación de los cruceros. El trabajo propuesto aquí se basa en trabajo previo de los mismos autores en el que había probado un enfoque analítico de dos fases para hacer la estimación del WIP de un sistema transportador con tornamesas en las esquinas de

intersección pero no con cruceros. En este trabajo se añadió una tercera fase para estimar los retrasos por cola y WIP causado por cruceros en los tornamesas y una cuarta fase para integrar todas las fases y así estimar el WIP total esperado en todo el sistema transportador.

En (30) Maggio et al. describieron un método de aproximación analítica para evaluar los valores promedio del rendimiento de los niveles de WIP para sistemas de producción de circuito cerrado con tres máquinas con inventarios temporales finitos y una población fija. El método presentado incluye un nuevo conjunto de ecuaciones de descomposición y un modelo nuevo de bloque de construcción. Las máquinas tienen tiempos de procesamiento determinísticos y probabilidades de fallo y recuperación distribuidas geométricamente. Los resultados obtenidos numéricamente son similares a aquellos obtenidos mediante simulación. El método tiene un buen desempeño dado que toma en consideración la correlación que existe entre los números de parte en el inventario temporal en proceso. En el artículo se presentan posibles extensiones al modelo básico.

Lagershausen et al. en (31) estudiaron el desempeño de líneas de ensamble de circuito cerrado con tiempos de proceso conocidos y espacios de inventario temporal finitos. Este artículo analiza líneas de flujo con recorrido de materiales que convergen o divergen, tamaños de inventario temporal limitado y tiene un número de piezas de trabajo constante que fluyen como una red de colas de ensamble o desensamble cerradas. Se modela un enfoque de descomposición en el cuál cada subsistema es un sistema de colas G/G/1/K. La restricción de población se refuerza al requerir que la suma del número esperado de clientes en el subsistema sea igual al número total de piezas de trabajo. Los resultados de los experimentos de simulación indican que la aproximación propuesta proporciona resultados precisos y que se desempeña mejor que otros enfoques.

El trabajo presentado por Ferreira et al. en (32) se describe el desarrollo de un sistema decisional basado en un modelo de simulación de eventos discretos. Dada la complejidad de los CLMS ellos utilizan la simulación para analizar y optimizar el desempeño de una línea de ensamble de circuito cerrado. El modelo está dirigido a una clase específica de líneas de producción con una configuración de red de cuatro circuitos conectados mediante transportadores en el ensamble y pre ensamble de automóviles. El número total de tarimas en los transportadores, que funcionan también como inventario temporal en proceso, permanece constante excepto en el cuarto circuito. En este último circuito el número de tarimas depende de la relación en el automóvil de cuatro puertas implementada entre las estaciones de desmontaje y de montaje de puertas del cuerpo del automóvil. Algunas ecuaciones de control de los cuatro circuitos cerrados son incompatibles con las capacidades de los inventarios temporales intermedios para ciertos valores de la relación dada para los carros de cuatro puertas. El modelo de simulación puede ser utilizado para evaluar el impacto de variación en el rendimiento de la línea a lo largo del transportador (en consecuencia también en los inventarios temporales intermedios). Adicionalmente en este artículo se presenta una propuesta para cambiar la longitud de los inventarios intermedios que puede maximizar el desempeño de la línea de producción y mejorar la disponibilidad del sistema de manufactura para cualquier valor de la relación de los automóviles de cuatro puertas.

En los artículos (30-32) se puede observar en forma similar la curva de relación inversa entre el rendimiento del sistema y el número de tarimas o población del sistema.

En el mundo real el rendimiento de los sistemas de manufactura de circuito cerrado CLMS es impactado frecuentemente por múltiples factores tales como el número total de tarimas en el sistema, la velocidad indexada de cada tarima, la confiabilidad de la máquina y el tiempo de proceso, las posiciones de carga y descarga de las partes, etc. Para obtener un diseño óptimo y el control de los CLMS se deben hacer sistemáticamente consideraciones de todos los factores mencionados y de otros que pudiesen ser establecidos. Desafortunadamente, a pesar de la complejidad de los CLMS en la realidad, la mayor parte de la literatura, en la mayor parte de los artículos, se enfoca solamente en estudiar el impacto de uno o dos factores en el desempeño del sistema y no considera o modela otros factores en un modelo integrado. De acuerdo a Li et al. (5) la mayor parte de los modelos de CLMS disponibles han sido sobre simplificados para poder hacer una manipulación matemática más sencilla pero sacrificando la precisión y la aplicabilidad a problemas actuales, incluso algunas de las técnicas de solución utilizadas son demasiado complejas para poder analizar en forma práctica sistemas CLMS de gran escala con gran cantidad de máquinas en el circuito.

En (4) los autores presentan una nueva metodología para modelar CLMS usando la técnica de acoplamiento de parámetros que se basa en el análisis tradicional de sistemas abiertos y proporcionan una técnica de solución relativamente simple pero efectiva para estimar el verdadero desempeño de los CLMS. Este proceso se realiza a través de la introducción de dos tipos de acoplamiento de parámetros: 1) el acoplamiento de máquina y 2) el acoplamiento de la capacidad del inventario temporal. La técnica de acoplamiento hace una extensión de las capacidades de modelado y de análisis de los sistemas de manufactura tradicional a los sistemas de circuito cerrado. Dado que los sistemas abiertos pueden tener diferentes modelos de tiempo de procesamiento tales como los modelos de tiempo de procesamiento exponencial, modelos de flujo continuo y modelos determinísticos, los CLMS también podrían tener varios modelos de tiempo de proceso. Esto hace a la técnica de acoplamiento de parámetros más versátil que la teorías tradicionales para modelar sistemas complejos de manufactura

3.3 SETUP

3.3.1 Introducción

Desde la primera mitad de los años 1950s se ha hecho investigación en el tema de "Programación", quizá son miles los artículos relacionados con dicho tema que se han escrito. Muchos de ellos han asumido que el tiempo y el coste de setup son despreciables como parte del proceso de un trabajo. En principio esta consideración ha parecido simplificar el análisis de los procesos y la implantación de algunas aplicaciones pero al final impacta la calidad de las soluciones de muchas aplicaciones que requieren específicamente considerarlos.

A continuación, en la Figura 16 se presentan las principales investigaciones en artículos publicados sobre el tema de *setup* que son relevantes a los problemas analizados en esta Tesis.

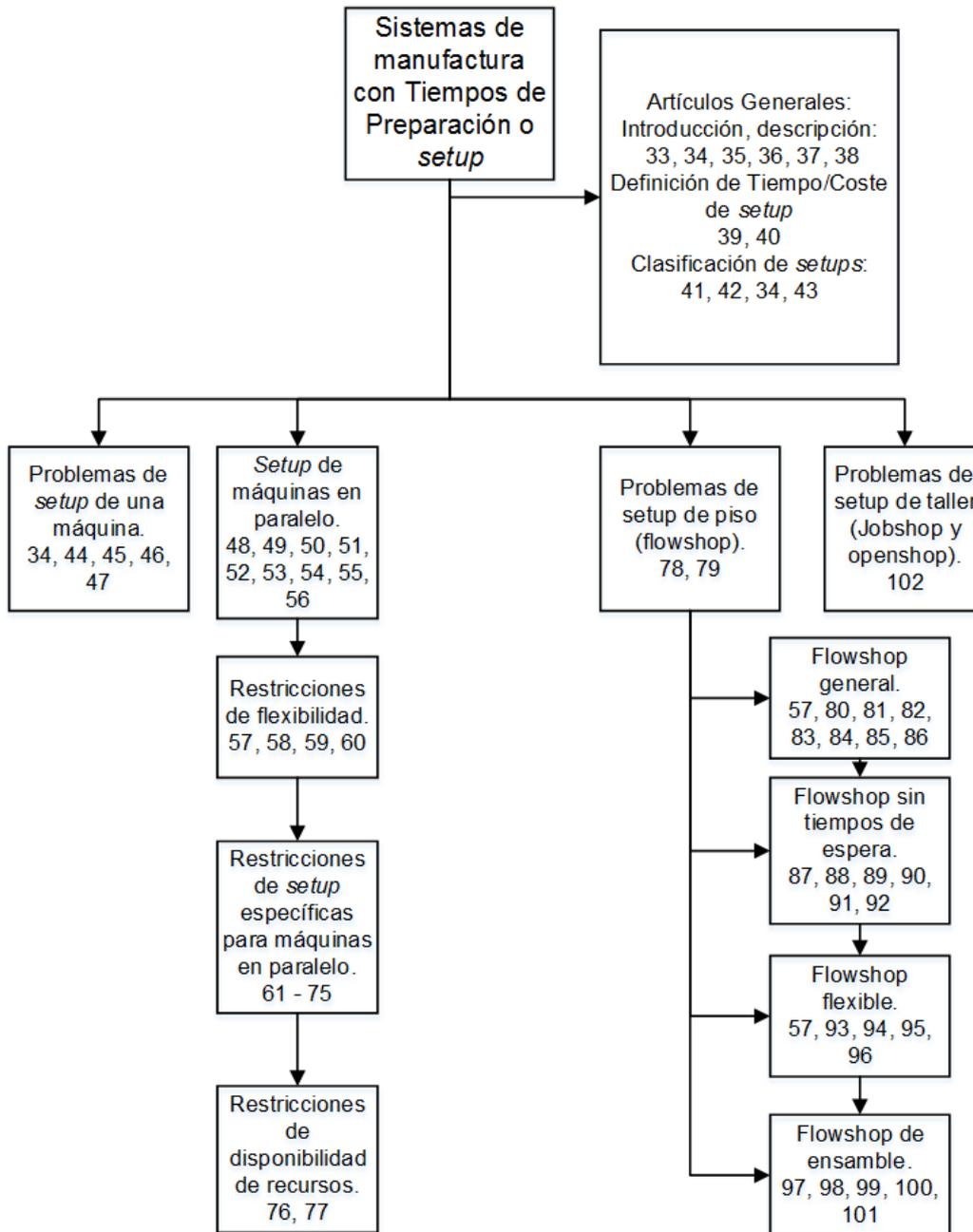


Figura 16: Revisión de Literatura en Problemas con *setup* relevantes a la Tesis

A partir de los años 1960s el tiempo de *setup* se ha tratado como parte independiente del coste total del proceso. Uno de los primeros artículos en el que se analiza el coste de *setup* es (33). La primera revisión exhaustiva sobre el tema de programación separando los costes de *setup* fue realizada en 1999 por (34), en esta se incluyeron 190 artículos sobre el tema a partir de 1965. Desde entonces se han escrito más de 40 artículos por año relacionados con los tiempos de *setup* y sus costes. Este creciente interés se debe a que existen importantes ahorros cuando dichos costes se consideran en las decisiones de programación en la práctica industrial. Una nueva revisión actualizando la

anterior fue efectuada por (35), dónde se analizan más de 300 artículos desde la revisión anterior de 1999 por el mismo primer autor.

En los problemas de programación del mundo real, sean de procesos de manufactura o de servicios es suma importancia hacer eficientemente la utilización y asignación de los recursos. Analizar los tiempos y costes de *setup* de manera separada a los tiempos y costes del proceso facilita identificar particularidades en cada caso y que ambas acciones se realicen simultáneamente y por lo tanto se haga una mejor utilización de los recursos. Esto es de particular importancia en los sistemas de administración de la producción actuales tales como el justo a tiempo (JIT), Tecnología de optimización de la producción (OPT), Grupos Tecnológicos (GT), Manufactura celular (CM) y la competencia basada en tiempos.

Los beneficios de reducir los tiempos y costes de *setup* incluyen: reducir gastos, incrementar la velocidad de la producción, incrementar la producción, reducir los tiempos de entrega, hacer cambios rápidos de producto en el proceso, aumentar la competitividad, aumentar las utilidades y la satisfacción, facilitar la manufactura esbelta, flujos más suaves en el proceso, un mayor rango de tamaño de lotes de producción, disminuir la curva del coste total, tener menos faltantes de producto, disminuir el inventario, bajar al mínimo el tamaño de las órdenes, tener mayores márgenes de ganancia en órdenes arriba del mínimo, tener una distribución más rápida y finalmente, incrementar la satisfacción del cliente.

La importancia y los beneficios de hacer investigación en tiempos y costes de *setup* ha sido revisada en numerosos artículos. Por ejemplo, (36) encontraron información cuantitativa producto de experimentos controlados en un rango representativo de plantas industriales que es útil en identificar los factores críticos para mejorar el desempeño en la manufactura. Entre otros factores destacan tanto el tamaño del lote como los tiempos de *setup*, los cuáles son los medios más efectivos para reducir los niveles de inventario y mejorar el servicio al cliente. En (37;38) se muestra que hacer un programa considerando tiempos y costes de *setup* incrementa la capacidad de producción tanto en la manufactura celular como en el taller tradicional y en los grupos tecnológicos. En (37;38) un procedimiento para lotes repetitivos (RL) capitaliza la dependencia de los *setups* en el taller. En dicho procedimiento se revisa una lista de trabajos en espera para buscar un trabajo idéntico al trabajo que se acaba de terminar en ese momento justo en la máquina, lo cuál elimina la necesidad de un nuevo *setup*. Un procedimiento de truncado (TRL) evita que los lotes sean excesivamente grandes al permitir que se ejecuten solo un número de trabajos K.

3.3.2 Definición de tiempo/coste de Setup

Los ajustes de proceso o *setups* son todas las tareas necesarias para adaptar un proceso productivo, configurado para generar un producto específico, a uno capaz de crear un producto diferente. Se considera un *setup* desde que se termina de producir la última pieza de una serie hasta la obtención de la primera pieza correcta de la siguiente serie, o de todas las actividades realizadas para obtener la primera pieza correcta al arranque de la

producción. Las tareas más comunes asociadas a los ajustes de proceso son (39) :

- *Preparación, ajustes post-proceso, revisión de materiales y herramientas.* Estas tareas aseguran que todas las partes y herramientas estén donde deben estar y que estén en buen estado. También se incluye el regresar los objetos que no serán utilizados, a su lugar de almacenaje, el acomodo, la limpieza, etc. Facilitan un ambiente propicio para ejecutar las actividades en una organización de servicios.
- Montaje y remoción de piezas. Esto incluye el desarmado y rearmado de la maquinaria para la sustitución de elementos claves como: moldes, cuchillas, llaves, etc. Son actividades relacionadas con los materiales y herramientas: acercar herramientas, acercar material al proceso, regresar herramientas, limpieza, inicializar plantillas (jigs, perchas o bastidores) y accesorios, ajustar herramientas e inspeccionar material en un sistema de manufactura. También incluye la instalación de software, transferencia de programas y archivos relacionados con un sistema de informática.
- Mediciones y calibración. Se incluyen aquí todas las actividades destinadas al correcto funcionamiento de la maquinaria ya armada y al aseguramiento de la calidad del producto. Estas actividades incluyen por ejemplo centrar, dimensionar, presurizar, calentar, etc.
- Corridas de prueba y ajustes. Se basan en lograr que los procesos sean estables y estén sincronizados antes del inicio de actividades, esto es especialmente importante cuando existen varias etapas en un sistema productivo. En este caso se consideran los ajustes realizados después de que la primera pieza fuese maquinada y evaluada. La cantidad de piezas a ser tomadas como prueba depende del tipo de producto, la calidad buscada y la variabilidad de los resultados de la configuración. Debido a esto las pruebas pueden llegar a ser innecesarias o hasta durar días enteros.

El tiempo de *setup* se define como el tiempo requerido para preparar los recursos para realizar una tarea. El coste de *setup* es el que se determina en todos los recursos antes de ser utilizados en una tarea.

En la industria automotriz la disminución de tiempos de *setup* cobra una vital importancia debido a los acelerados ritmos productivos, el alto coste y variedad de los productos fabricados.

En los casos donde el coste y el tiempo son directamente proporcionales es posible considerar en el análisis solo uno de ellos, esto ocurre normalmente cuando el objetivo es solo reducir el tiempo de inactividad. Sin embargo en otros casos el coste de *setup* será más significativo, especialmente cuando el coste del cambio entre algunas actividades sea relativamente alto, aun cuando el tiempo de *setup* sea bajo. Este caso se presenta sobre todo cuando se requiere un alto nivel de habilidad al ejecutar las actividades o cuando los

recursos implementados son relativamente de alto coste, como la fabricación de un molde o la puesta en marcha de una planta de energía, por ejemplo.

Existen dos tipos de problemas relacionados con el *setup* al separarlo del tiempo de proceso:

- 1) Cuando el *setup* depende exclusivamente del trabajo a ser procesado sin importar el que previamente se haya realizado. Se le llama "*Setup de secuencia independiente*".
- 2) El segundo caso se tiene cuando existe un efecto tanto del trabajo a ser procesado como del producto inmediato precedente en la secuencia. En este caso se le llama "*Setup de secuencia dependiente*".

En algunos casos pueden existir familias o grupos de tareas las cuales requieren un *setup* por familia. Este tipo de *setup* de familia también puede ser de secuencia dependiente o independiente.(40)

Tal como se mencionó en la introducción en el mundo real y práctico de la industria es necesario en muchos casos separar el tiempo de *setup* de los tiempos de proceso. Las aplicaciones de esto pueden encontrarse en diferentes sistemas de producción, de servicio y de proceso de información. Algunos ejemplos son:

En una aplicación de informática, un trabajo requiere un tiempo de *setup* para cargar un compilador distinto si el actual no está disponible.

En una imprenta o una editorial, un tiempo de *setup* se requiere para preparar la máquina (limpieza), el cual depende tanto del color actual como del que se deba seleccionar para el siguiente trabajo.

En una empresa textil el tiempo de *setup* para el hilado y el tejido dependen de la secuencia de los productos.

En una fábrica de envases y botellas de plástico, el tiempo de *setup* está relacionado con la forma y el tamaño de los envases y botellas que se fabrican.

En la industria del plástico diferentes modelos y colores requieren diferentes tiempos de *setup*.

Las situaciones mencionadas se pueden extender a los casos de las industrias química, farmacéutica, de procesamiento de alimentos, metalúrgica, del papel, y muchas otras, entre ellas la de los procesos de pintura en la industria automotriz.

3.3.3 Clasificación

El reciente interés en la programación con los tiempos y costes de *Setup* demuestra la importancia del problema tanto para investigadores como para la gente de la industria.

De acuerdo a (41) los problemas de *Setup* se pueden clasificar en:

- Producción con lotes y sin lotes.
- *Setup* de secuencia dependiente y de secuencia independiente

- Clasificación en piso: a) una máquina, b) máquinas en paralelo, c) flujo de taller, d) otros.

El tiempo y coste de *setup* en lote aparece cuando un trabajo es procesado en lote, por ejemplo productos en cajas, contenedores, tarimas etc., y el tiempo o coste de *setup* precede al procesamiento de cada lote. La definición de cada lote es como sigue: Los trabajos son clasificados en F familias, donde $F \geq 1$. Un lote es un grupo de trabajos a ser realizados de la misma familia. Aunque la formación de familia se debe hacer con anticipación, la formación de lotes es una parte del proceso de decisiones.

Un caso importante surge cuando se consideran los Grupos Tecnológicos pues ninguna familia puede ser dividida, solo se puede considerar un lote único para cada familia.

El tiempo/coste de *setup* puede ser dependiente de la máquina o dependiente de la secuencia de familias. Es dependiente de la secuencia si la duración (coste) depende tanto de las familias actual o en proceso y las familias del lote o lotes anteriores. Un *setup* será independiente si su duración (coste) depende solamente de la familia en lote que actualmente se está procesando y no de ningún lote precedente.

Se puede hacer un análisis con un mayor detalle si los modelos de *setup* en lote se clasifican dividiendo en modelo de disponibilidad del lote y modelo de disponibilidad del trabajo. En el primer caso todos los trabajos del primer lote están disponibles para ser procesados y para ser terminados al mismo tiempo. En el segundo caso los tiempos de inicio y de término del trabajo son independientes a los casos de otros trabajos en el mismo lote. Para efectos de nuestro estudio consideramos el segundo caso: modelo de disponibilidad de trabajo, a menos que se indique específicamente el otro caso.

Para el caso de los sistemas de proceso con etapas múltiples se deben considerar: programas con permutación y sin permutación, y programas con lotes consistentes e inconsistentes. Un programa de producción se considera con permutación cuando la secuencia de los trabajos es la misma en todas las máquinas. Los lotes son consistentes si la definición o integración del lote es la misma en todas las máquinas. Lo contrario definirá un programa sin permutación y un programa con lotes inconsistentes.

En un sistema sin lotes, el tiempo y coste de *setup* se presenta antes del procesamiento de cada trabajo. El modelo correspondiente puede ser analizado como un modelo de tiempo y coste de *setup* por lote en el que cada familia consiste en un solo trabajo.

Se puede hacer distinción entre *setup* anticipado y *setup* no anticipado. El primer caso ocurre cuando el *setup* se puede iniciar antes de que el trabajo correspondiente esté disponible en la máquina, de otra manera el trabajo será no anticipado. Se considera que los trabajos son anticipados, a menos de que se indique particularmente otra cosa o que los trabajos dependan de fechas de liberación o lanzamiento. En cualquiera de los modelos de *setup* ningún trabajo puede ser procesado en una máquina mientras se esté efectuando un *setup* en la misma.

Se utilizará la terminología de tiempo y coste de *setup* propuesta en (34). El término "familia" adoptado por (42) se utilizará para indicar la partición o

clasificación inicial de los trabajos. El término “lote” será utilizado para indicar particiones posteriores o particiones como parte de la solución. En muchos casos el término “lote” se utiliza para indicar la clasificación inicial y las particiones posteriores se indican como “sub-lote”. La terminología que utilizaremos fue adoptada por (34) y fue tomada de la notación de tres campos de (43) para describir un problema de programación.

El campo α describe el entorno del taller (piso). El campo β describe la información del *setup*, condiciones adicionales del piso y detalles de las características del proceso las cuales pueden contener varias entradas. Finalmente el campo γ contiene el objetivo a minimizar. Un ejemplo puede ser el de un problema de programación del flujo de proceso en tres máquinas donde se busca minimizar el retraso máximo con tiempos de *setup* con lotes dependientes de la secuencia. La notación sería: $F3/ST_{sd,h}/L_{max}$.

Tipo de taller (campo α)	
I	Una máquina
F	Flujo de taller
FF	Flujo de taller flexible (hibrido)
AF	Flujo de taller de montaje
P,Q,R	Máquinas en paralelo: P:máquinas idénticas Q: máquinas uniformes R: máquinas no relacionadas.
J	Taller de trabajo
O	Taller abierto

Tabla 1: Tipo de Taller

Características de taller (Campo β)	
Prec	Restricciones de precedencia
r_j	Fecha de liberación distinta a cero
Pmtn	Preferencia

Tabla 2: Características de Taller

Información del Setup (Campo β)	
ST_{si}	Tiempo de Setup independiente de la secuencia
SC_{si}	Coste de Setup independiente de la secuencia

St_{sd}	Tiempo de Setup dependiente de la secuencia
SC_{sd}	Coste de Setup dependiente de la secuencia
$ST_{si,b}$	Tiempo de Setup independiente de la secuencia por lote o familia
$SC_{si,b}$	Coste de Setup independiente de la secuencia por lote o familia
$ST_{sd,b}$	Tiempo de Setup dependiente de la secuencia por lote o familia
$SC_{sd,b}$	Coste de Setup dependiente de la secuencia por lote o familia
R_{si}	Tiempo de eliminación independiente de la secuencia
R_{sd}	Tiempo de eliminación dependiente de la secuencia
$R_{si,b}$	Tiempo de eliminación independiente de la secuencia por lote o familia
$R_{sd,b}$	Tiempo de eliminación dependiente de la secuencia por lote o familia

Tabla 3: Información General del Setup

Desempeño (campo γ)	
C_{max}	Makespan (Tiempo de ejecución)
L_{max}	Retraso máximo (lateness)
T_{max}	Retraso máximo (tardiness) el que llegó mas tarde en la carrera
D_{max}	Plazo máximo de entrega
TSC	Costo total de Setup
TST	Tiempo total de Setup
Σf_j	Tiempo de flujo total
ΣC_j	Tiempo total de ejecución
ΣE_j	Tiempo total de adelanto en la producción
ΣT_j	Tiempo total de retraso en la producción
ΣU_j	Número de trabajos con retraso
$\sigma w_j C_j$	Tiempo total ponderado de ejecución
$\Sigma w_j U_j$	Número ponderado de trabajos con retraso
$\Sigma w_j E_j$	Tiempo total ponderado de adelanto
$\Sigma w_j T_j$	Tiempo total ponderado de retraso

$\sum w_j f_j$	Tiempo total ponderado de flujo
----------------	---------------------------------

Tabla 4: Tabla de Factores de Medición del Desempeño en Setup

3.3.4 Problemas *Setup* de una máquina.

El caso de la programación de una máquina ha sido ampliamente estudiado desde mediados de los años 1950s. En este caso los trabajos que llegan son ejecutados por un solo servidor. Los trabajos son realizados uno a la vez, y cada uno tiene diferentes características, por ejemplo: tiempo de proceso, fecha de entrega, tiempo de Setup, etc. La secuencia en que es programado puede determinarse dependiendo de la característica que se busque optimizar.(34).

La programación de una máquina, siendo el caso más simple es muy útil desde el punto de vista teórico para optimizar los tiempos de Setup y aplicar de forma más eficiente las reglas de secuenciación. Un ejemplo de esto se ve en que para mantener la competitividad, las compañías que manufacturan insumos con alta tecnología deben de buscar la manera de disminuir sus costes y garantizar alta calidad al mismo tiempo. En (44) Se estudia el problema de equilibrar estos objetivos al utilizar un (APC) Proceso de Control Avanzado en la industria de semiconductores en interacción con el programa de fabricación. Se analiza el problema de ejecución de trabajos en una máquina con trabajos múltiples con restricciones por el APC que se demuestra que es del tipo *NP_hard*. Para algunos casos particulares se obtienen soluciones óptimas analíticamente. Se propone un algoritmo heurístico basado en dichos resultados estructurales y comparados con programación entera.

El artículo de (45) está orientado integralmente al análisis del tamaño del lote y al programa de fabricación considerando una máquina con una orden de fabricación cada vez y para la cual la tasa de producción es la misma para todas las órdenes del cliente. La demanda es determinística y todas las órdenes están listas para ser procesadas en el tiempo cero, además de que deben de ser entregadas en la misma fecha indicada. Cada orden puede ser entregada desde diferentes lotes. Se consideran límites superior e inferior en el tamaño del lote. El objetivo es encontrar un programa factible que minimice la suma del coste por retraso y el coste de *setup* incurrido por crear un nuevo lote. Se presentan algunas características del lote óptimo tanto para una orden sencilla como para problemas con múltiples órdenes y proponen un algoritmo de programación dinámica basado en estas características. Se presentan resultados computacionales que demuestran la eficiencia del método propuesto.

Los autores en (46) revisaron el problema de los sistemas de producción de una máquina construyendo un programa que esté protegido contra eventos inciertos. Un programa de producción es robusto cuando los cambios son pequeños al aparecer variaciones en los parámetros de entrada. Propusieron un algoritmo Branch and Bound para resolver el problema intentando asegurar la estabilidad del programa. Como resultado se obtuvo que el algoritmo sea

exacto cuando es solo uno el trabajo que es interrumpido durante la ejecución del programa.

En (47) se hace una revisión del problema de programación de un lote en una máquina con tiempos de *setup* por familia con fechas de liberación para minimizar el tiempo de ejecución. En dicho artículo se muestra que este problema es NP_hard y se propone un algoritmo de programación dinámica, una heurística y un esquema de aproximación con tiempo polinomial.

3.3.5 Problema de *Setup* de máquinas en paralelo.

Existe una considerable bibliografía sobre la programación de máquinas en paralelo y varias revisiones existen sobre el tema, ver (48-53). En dichas revisiones se pueden encontrar numerosos enfoques en la gran cantidad de estudios sobre el mismo. Una de las variables más analizadas ha sido la minimización del tiempo total de retraso $\sum T_j$, se ha demostrado que este problema es Np_hard (54). Entre los aspectos menos estudiados están: máquinas en paralelo con restricciones de flexibilidad, restricciones de *Setup* de máquinas en paralelo y restricciones por el uso de recursos secundarios(55) y (56).

3.3.5.1 Restricción de flexibilidad en las máquinas en paralelo.

Dependiendo de los autores, cuando solo un subgrupo de máquinas puede realizar un trabajo determinado se pueden clasificar en máquinas con restricción elegible (57) o máquinas multiusos. Varias heurísticas toman en cuenta esta particularidad. Pinedo (57) muestra interés en las heurísticas flexibles tales como LFJ o LFM-LFJ:

- *LFJ (Least Flexible Job)* da prioridad al trabajo que puede ser logrado en el menor número de máquinas. Encontró que es óptimo para $P_m | \rho_j=1, M_j | C_{max}$
- *LFM-LFJ (Least Flexible Machine – Least Flexible Job)* elige primero la máquina menos flexible que puede ejecutar el menor número de trabajos y la asigna al trabajo menos flexible. No se ha encontrado el óptimo en este caso.

Vairaktarakis y Cai (58) proponen dos heurísticas para minimizar el tiempo de ejecución (*makespan*) considerando la flexibilidad en el caso sin fechas de liberación: 1) *LTW-LPT* y 2) *LAW-LPT*. En el primer caso, la heurística de la menor carga de trabajo total con el mayor tiempo de procesamiento, calcula la carga total posible para cada máquina (carga de trabajos actuales más los que podrían ser asignados). Selecciona la máquina con la menor carga potencial y ejecuta el trabajo con el mayor tiempo de procesamiento. El segundo caso considera una heurística para la menor carga de trabajo promedio con el mayor tiempo de procesamiento. Es similar a la anterior pero divide la duración de cada trabajo no asignado entre el número de máquinas elegibles. Los resultados del artículo muestran que el primer caso es competitivo pero no robusto y el segundo obtiene número cercanos al óptimo.

Centeno y Armacost (59) intentan también minimizar el tiempo máximo de ejecución (*makespan*) C_{max} en máquinas en paralelo con restricciones variables. Muestran que si las máquinas no están anidadas la regla *LPT-LFM* proporciona mejores resultados que la de *LFJ-LFM*.

Considerando máquinas en paralelo no relacionadas (la duración de los trabajos depende de la máquina asignada) y criterios ponderados de anticipación o tardanza Bank y Werner (60) compararon heurísticas constructivas con metaheurísticas iterativas. Las primeras hacen una secuenciación de los trabajos en orden creciente (INC) o decreciente (DEC), o de acuerdo con fechas de liberación (RD), mayor holgura (SL) o menor holgura (SS) para las máquinas. Utilizando este orden, los trabajos son asignados a las máquinas intentando minimizar el C_{max} (*makespan*). Finalmente los trabajos en cada máquina son reprogramados para minimizar las diferencias entre trabajos adelantados y trabajos retrasados. La mejor heurística parece ser SL-DEC. Se consideran también dos metaheurísticas (recocido simulado y umbral de aceptación) con varios entornos: SH (*shift one job off machine*) cambio de un trabajo que está fuera de la máquina, PI intercambio de un par de trabajos, SHM cambio de un trabajo que está en la máquina. Los autores muestran que ambas metaheurísticas son equivalentes y que el mejor entorno es SH.

3.3.5.2 Restricciones de *Setup* en máquinas en paralelo.

Las consideraciones de tiempos y costes de *setup* son relevantes en el problema de programación de máquinas en paralelo. La programación puede ser dependiente (ST_{sd}, SC_{sd}) o independiente de la secuencia (ST_{si}, SC_{si}). Las máquinas en paralelo pueden ser P_m, R_m (Las máquinas uniformes Q_m se consideran como un caso de las máquinas idénticas) (61;62). Siendo el problema NP-hard, la mayor parte de los autores proponen metaheurísticas para obtener buenas soluciones. En (35) se hace una revisión de problemas de programación con tiempos y costes de *setup*. Se clasifica la literatura de acuerdo al entorno (una máquina, máquinas en paralelo, taller etc.), la presencia o ausencia de lotes, la existencia de *setup* dependiendo o no de la secuencia de los trabajos y las restricciones de disponibilidad. Una forma típica de minimizar el retraso total es reducir el número de *setups*. Esto lleva a agrupar las tareas en grupos de trabajo de la misma familia. La revisión muestra que hasta 2008 pocos artículos investigaban sobre las máquinas en paralelo. En (63) y en (64) se proponen procedimientos branch and bound para el caso de $R_m | p_i s_{i,j} | \sum T_i$ (máquinas en paralelo no relacionadas, tiempos de *setup* dependientes de la secuencia, y un criterio de minimización del retraso total) y para el caso $R_m | p_i s_{i,j} | C_{max} + \sum T_i$ (máquinas en paralelo no relacionadas, tiempos de *setup* dependientes de la secuencia, y un criterio de minimización del tiempo de ejecución máximo y del retraso total). La complejidad del algoritmo llevó a tiempos de uso de ordenador demasiado grandes para problemas de un tamaño realista.

En (65) se minimiza el tiempo de flujo total considerando máquinas idénticas en paralelo. Proponen dos heurísticas de programación dinámica (hacia delante y hacia atrás). En este caso la complejidad también es significativa.

En (56) se revisa el problema $R_m|p_i s_{i,j}|\Sigma w_i C_i$ (máquinas en paralelo no relacionadas, tiempos de *setup* dependientes de la secuencia, y un criterio ponderado de tiempo de flujo). Se comparan siete heurísticas. Las primeras seis secuencian los trabajos de acuerdo a la regla WSPT basada sea en la duración promedio de los trabajos en las máquinas o en la duración mínima. Enseguida los trabajos son asignados sucesivamente sea con base en el tiempo mínimo de ejecución total, la mínima duración de un trabajo o en la mínima duración de un trabajo considerando tiempos de *setup*. La séptima heurística no ordena los trabajos. Selecciona el par (trabajo-máquina) que genere el mínimo tiempo de ejecución. Esta última heurística supera en forma significativa a las demás.

Lee y Pinedo (66) también analizan el problema $R_m|p_i s_{i,j}|\Sigma T_i$. Introducen una regla de asignación ATCS como una extensión de la regla ATC dada en (67) que considera *setups*. Al parecer genera mejores resultados que WSPT u EDD. Los autores proponen adicionalmente un procedimiento de recocido simulado que administra una lista de prioridad de horarios de trabajo en piso con la regla de ATCS. EL análisis del entorno vecino consiste en el intercambio inteligente de trabajos adyacentes en la lista. Cuando un trabajo es movido a una nueva máquina, todos los trabajos pendientes en la máquina inicial también son movidos. Esto facilita el poder mantener la secuencia de máquina con buenos tiempos de *setup*. El trabajo con el mayor tiempo de *setup* se selecciona para a partir de él definir al vecino. Park et al (68) extienden este procedimiento al uso de redes neuronales para fijar parámetros de la heurística ATCS. En (69) se propone un algoritmo de tres fases: los primeros grupos de trabajo se conforman con base en los que tienen fechas de entrega similares; en segundo lugar, para cada grupo, los trabajos son secuenciados en una lista de acuerdo a la regla ATCS y el intercambio de trabajos se hace utilizando un algoritmo de búsqueda Tabú; en la tercera fase los trabajos son asignados a las máquinas. Esta heurística proporciona mejores resultados que ATCS. En (70) desarrollaron un procedimiento de recocido simulado para el mismo problema, Los autores proponen seis reglas para construir soluciones en el vecindario: intercambio e inserción de lote o artículo, fusión de lotes o división de lotes. La probabilidad de elegir un artículo o un lote depende del tiempo de retraso para el artículo o el lote. La experimentación muestra la conveniencia de combinar varias reglas (70).

En (71) Kim et al consideran el problema de programación de lotes con máquinas no relacionadas en paralelo y tiempos dependientes de la secuencia. Ellos comparan tres heurísticas (WSPT, WEDD, y TH programación de lotes en dos niveles) con un procedimiento de recocido simulado. TH da prioridad a los lotes de acuerdo a fechas de entrega ponderadas, enseguida asigna los trabajos a las máquinas intentando cumplir las fechas de entrega y minimizar los *setups*. Los experimentos muestran que TH supera a las reglas de WSPT y de WEDD. Sin embargo recocido simulado (SA) se desempeña mejor.

En (72) también se proponen varias formas de generación de entornos para el problema de programación del flujo en múltiples etapas en máquinas en paralelo no relacionadas con restricciones de *setup*. El algoritmo utiliza dos mecanismos de intercambio de trabajos en cada iteración. El primer mecanismo selecciona aleatoriamente dos trabajos que no serán procesados en la misma máquina. El segundo corresponde a la opción de dos trabajos

adyacentes procesados en la misma máquina. Los autores muestran que el algoritmo supera a un recocido simulado básico.

En (73) proponen un sistema basado en agentes que programa los trabajos en máquina múltiples no idénticas con varias restricciones y objetivos. Los agentes aplican varias heurísticas para construir o mejorar una solución. La colaboración entre agentes se basa en compartir un grupo de buenas soluciones. Los autores reportan que este sistema supera cualquier algoritmo individual.

(74) estudian el problema del caso $R_m | p_i s_{i,j} | C_{\max} + \sum T_i$ (máquinas en paralelo no relacionadas, tiempos de *setup* dependientes de la secuencia, y un criterio de minimización del tiempo de ejecución máximo y del retraso total), utilizan una Metaheurística GRASP (64;74;75). La fase de construcción del GRASP implementa un algoritmo de tratamiento de la lista: sea EDD o de la menor holgura (74). La búsqueda local se basa en el intercambio de la precedencia de los trabajos en la lista. Para una mejora en esta búsqueda local se propone el uso de una variable de búsqueda en el vecindario (VNS, variable neighborhood search). Se implementan tres estrategias (intercambiar dos trabajos en una máquina, intercambiar dos trabajos de dos diferentes máquinas, añadir un trabajo a una máquina (75)). VNS parece ser más interesante a medida que se incrementa el tamaño del problema. Finalmente para trabajos pequeños, los autores muestran que los resultados de GRASP están cerca del óptimo (64).

3.3.5.3 Restricciones de disponibilidad de recursos.

Los recursos generalmente presentan dos tipos de restricciones de disponibilidad: el primero aparece por el número de recursos lo cual limita el número de actividades que se pueden ejecutar en paralelo, la segunda es por el calendario, que limita la presencia del recurso. Cuando un *setup* requiere las actividades de montar una herramienta y desmontar otra, con un número limitado por tipo de herramienta, a esta restricción en la literatura se le llama recurso secundario (76). Se estudiado muy poco la programación de máquinas en paralelo no relacionadas con restricciones de recursos secundarios. Tamaki et al. (76) proponen una formulación matemática del problema y una representación binaria de programas admisibles. El mejor resultado lo logran con un recocido simulado binario. En (77) los autores presentan una combinación entre búsqueda Tabú y algoritmo del umbral de aceptación para la minimización del retraso. El algoritmo identifica grupos o conjuntos de operaciones que utilizan la misma herramienta en la misma máquina. Un grupo puede contener todas las operaciones, un subgrupo o una sola operación. Todos los mecanismos de búsqueda en el vecindario mueven un grupo de la máquina con el mayor tiempo de retraso a una segunda máquina. Algunos adicionalmente mueven un conjunto de la segunda máquina a una tercera. Una restricción de ventana de tiempo implica que una fecha asociada a una operación (fecha de inicio o de término) debe efectuarse dentro de dicha ventana de tiempo o dentro de un grupo de ventanas de tiempo. En el artículo de Tamaki (76) la restricción de la ventana de tiempo se refiere solo a la fecha de inicio del *setup*.

3.3.6 Problemas de Setup de piso (*Flowshop*):

En los problemas tipo flowshop todas las tareas son procesadas. En los problemas tipo flowshop todas las tareas son procesadas siguiendo un mismo orden de máquinas en la secuencia, teniendo como consecuencia que algunas tareas puedan llegar a máquinas ocupadas, lo que genera colas de espera para ser procesadas. La secuencia puede ser de producción por lotes, donde una vez que se fabrica cierta cantidad de artículos de un tipo de producto, se preparan las máquinas para fabricar el otro producto, también existen secuencias de producción continua, donde las máquinas siempre están trabajando en la producción del mismo producto (78).

Los problemas tipo *flowshop* pueden ser clasificados como; *flowshop* general, *flowshop* sin tiempos de espera, *flowshop* flexible y *flowshop* de ensamble (79). Se consideran m etapas en serie en un ambiente de *flowshop* de m máquinas. Cada trabajo debe ser procesado en cada una de las m etapas en el mismo orden, y los tiempos de procesamiento suelen diferir en cada etapa. En el problema convencional existe solo una máquina por etapa. Allahverdi hace una revisión de más de 230 artículos referentes a la solución de problemas tipo *flowshop* publicados entre el año 2006 y 2015(79)

3.3.6.1 *Flowshop* general:

En los problemas tipo flowshop todas las tareas son procesadas. En el problema de flowshop, las secuencias permutadas no son necesariamente óptimas (57). Algunos autores demuestran de distintas formas como se puede conseguir un mejor rendimiento con heurísticas no permutativas (80-82). De hecho, se ha probado que cuando se incluyen más de tres máquinas en el sistema, las soluciones obtenidas mediante a permutaciones dejan de ser dominantes (83).

Wang y Cheng proponen dos heurísticas. La primera con un acercamiento de esquema polinomial para el problema $F2/ST_{si}/C_{max}$, donde utilizan una constante de disponibilidad para la primera máquina (84). En la segunda consideran el mismo problema, aunque en esta ocasión la constante de disponibilidad se encuentra en una de las dos máquinas (85).

Yang y Liu presentan un algoritmo genético multi-objetivo con búsqueda local para el problema $F_m/ST_{si}/FI (C_{max}, T_{max})$, el cual obtiene mejores resultados que dos algoritmos genéticos multi-objetivo propuestos anteriormente (86).

3.3.6.2 *Flowshop* sin tiempos de espera:

En un sistema de *flowshop* sin tiempos de espera las tareas comienzan inmediatamente después de que la tarea anterior fue completada, ver Figura 17. Este problema forma parte de la rama de investigación de problemas de secuenciación sin buffers. Las aplicaciones más comunes se encuentran en industrias como la de moldeado de plásticos (87), la de procesamiento de alimentos (87) o la metal-mecánica (88).

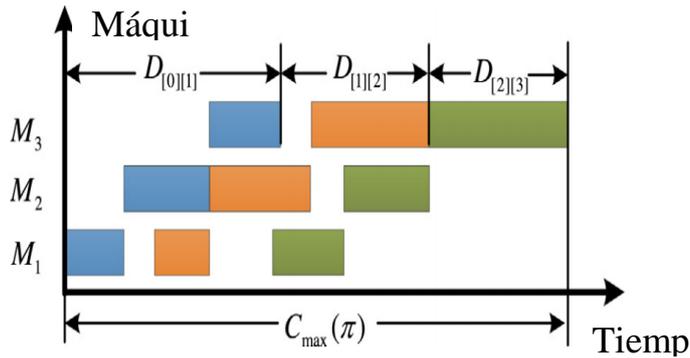


Figura 17: Gráfico de gantt para un sistema de flowshop sin tiempos de espera.

Ding y Song proponen un algoritmo con un mecanismo iterativo de búsqueda Tabú para resolver el problema de flowshop sin tiempos de espera, con criterio de tiempos de procesamiento. Este algoritmo mejora el rendimiento de sus predecesores, que no utilizan la búsqueda Tabú, debido a que escapa más fácilmente de mínimos locales cuando se incorpora una búsqueda de vecinos (89).

En una primera instancia, Nagano (90) genera una nueva heurística evolutiva con búsqueda agrupada para el problema $Fm/ST_{si}, no-wait/, no-wait/\Sigma C_j$ con buenos resultados. Más tarde, en una segunda publicación (91), se utiliza el mismo algoritmo para resolver el problema incorporando la dependencia de los tiempos de *setup*. Los resultados de ambas aplicaciones son comparados con los mejores algoritmos encontrados en la literatura.

Allahverdi y Aydilek (92) analizan el problema $Fm/ST_{si}, no-wait/\epsilon(\Sigma C_j/C_{max})$. Ellos proponen un algoritmo de inserción, dos algoritmos genéticos, tres algoritmos de recocido simulado y un algoritmo de diferenciación evolutiva. Los resultados computacionales muestran que uno de los algoritmos genéticos no solo se desempeña mejor que el resto, sino que supera al algoritmo con mejores resultados hasta el momento.

Aldowaisan y Allahverdi proponen (79) un algoritmo de recocido simulado y un algoritmo genético para el problema $Fm/ST_{si}, no-wait/\Sigma T_j$. También generan versiones mejoradas de estos algoritmos, obteniendo una mejora de desempeño del 95%. Adicionalmente, demuestran que el algoritmo genético supera los resultados del recocido simulado en un 3.5%.

3.3.6.3 Flowshop Flexible:

En un sistema de *flowshop* flexible, o híbrido, se contempla la posibilidad de utilizar más de una máquina por estación, con la finalidad de balancear el sistema productivo. Sus aplicaciones pueden ser encontradas en una gran variedad de situaciones, desde servicios de información hasta sistemas de manufactura (57). El problema de *flowshop* flexible puede ser definido de la siguiente manera. Se considera un conjunto N con n trabajos y un conjunto M con m etapas, donde en cada etapa i , $i=(1, 2, \dots, m)$ hay un conjunto de m_i máquinas idénticas. Cada trabajo j , $j=1, 2, \dots, n$) es requerido siguiendo la misma secuencia de proceso, empezando en la etapa 1 y terminando en la

etapa m . Cada trabajo puede saltar algunas etapas. Por lo tanto, cada etapa i procesa un conjunto E_i de e_i trabajos, donde $E_i \subseteq N$. Cada trabajo puede ser procesado por exactamente una máquina de las que se encuentren disponibles en cada etapa, donde $p_{i,j}$ denota el tiempo de procesamiento del trabajo j en la etapa i . Cada máquina puede procesar solo una tarea simultánea mientras que queda trabajo puede ser procesado en una máquina a la vez. Dado que las máquinas que se encuentran en la misma estación son idénticas, el tiempo de procesamiento solo depende de la etapa (93).

Comúnmente, este tipo de problemas son solucionados utilizando programación lineal entera mixta (94). Naderi y Gohari hacen una revisión de la literatura referente estos modelos, proponen cuatro modelos de programación lineal propios y llevan a cabo experimentos para comparar sus resultados y complejidad. Además, proponen un algoritmo de enjambre de partículas con un criterio de aceptación y una heurística de búsqueda local, afinado con el método de Taguchi (93).

Fattahi (95) analiza y resuelve el problema de $FF2/ST_{si}/C_{max}$ con operaciones de ensamblaje. Donde una gran cantidad de productos del mismo tipo son ordenados para ser producidos de forma en que las partes sean manufacturadas en un sistema de slowshop flexible, y más tarde sean ensambladas en un proceso posterior. Se presentan los límites inferiores y un algoritmo de ramificación y poda para el problema.

Huang presenta (96) un modelo de programación entera para el problema $FF2/ST_{si}, no-wait/\Sigma C_j$. También presenta un algoritmo de colonia de hormigas y evalúa su desempeño, probando que la heurística tiene buenos resultados.

3.3.6.4 Flowshop de ensamble:

El problema de *flowshop* de ensamble es analizado por primera vez por Lee y Cheng (97), y consiste en dos etapas. En la primera etapa se tienen m máquinas mientras que en la segunda consiste en una o más máquinas de ensamblaje. Se tienen n trabajos disponibles en el tiempo cero. Cada trabajo requiere de $m+1$ operaciones. Cada máquina puede procesar solo una tarea a la vez y no se puede anticipar. Las primeras m operaciones de un trabajo son procesadas por m máquinas diferentes en la primera etapa. Después, la siguiente operación del trabajo que corresponde al ensamble, es realizada en la segunda etapa. La última operación de un trabajo puede empezar solamente si se han terminado todas las operaciones de la primera etapa. Existen diversos casos en la industria donde se puede encontrar este problema. Uno de ellos es la producción de equipos de cómputo personales, en donde cada cliente solicita una combinación de los distintos componentes ofrecidos por el fabricante (98).

Allahverdi y Al-Anzi (99) presentan una relación de dominancia y tres heurísticas para el problema $AF2/ST_{si}/\Sigma C_j$ (100). Las tres heurísticas propuestas son una búsqueda tabú híbrida y dos algoritmos adaptativos diferenciales evolutivos. Demostraron que uno de los algoritmos adaptativos se desempeñaba mejor que las otras dos propuestas.

Tian (101) analiza el problema $AF2/ST_{s_i}/F_l (\sum C_j, C_{max})$. Propone un algoritmo de enjambre de partículas discreto y compara sus resultados con otros algoritmos como recocido simulado y colonia de hormigas, demostrando que su algoritmo tiene un buen rendimiento.

3.4 SCHEDULING

3.4.1 Introducción:

Al igual que los problemas de Job Shop, en general, cualquier problema de *scheduling* (programación) es un problema de optimización combinatoria. La función del *scheduling* es la asignación de recursos limitados a tareas a lo largo del tiempo y tiene como finalidad la optimización de uno o más objetivos (102). Los recursos pueden ser máquinas en un taller, pistas en un aeropuerto, ladrillos en una construcción, unidades de procesamiento en un ambiente computacional, etc. Como tareas se pueden tener operaciones de un proceso de producción, despegues y aterrizajes en un aeropuerto, etapas de un proyecto de ingeniería, ejecuciones de un programa computacional, etc. Cada tarea puede tener diferentes niveles de prioridad, así como tiempos de posibles inicios. Los objetivos pueden tomar varias formas: minimizar los tiempos de finalización de la última tarea, minimizar el número de tareas luego de una fecha de entrega acordada, etc.

A continuación, en la Figura 18 se presentan las principales investigaciones en artículos publicados sobre el tema de *scheduling* que son relevantes a los problemas analizados en esta Tesis. En ellos se puede observar que se hace un esfuerzo relevante en analizar el caso de la programación cíclica dado que es el tipo de situación que más frecuentemente y normalmente se presenta en los dos problemas que se revisan en esta Tesis. Por esta razón, si bien se hace una clasificación general, la importancia en la revisión de artículos se hace al caso mencionado.

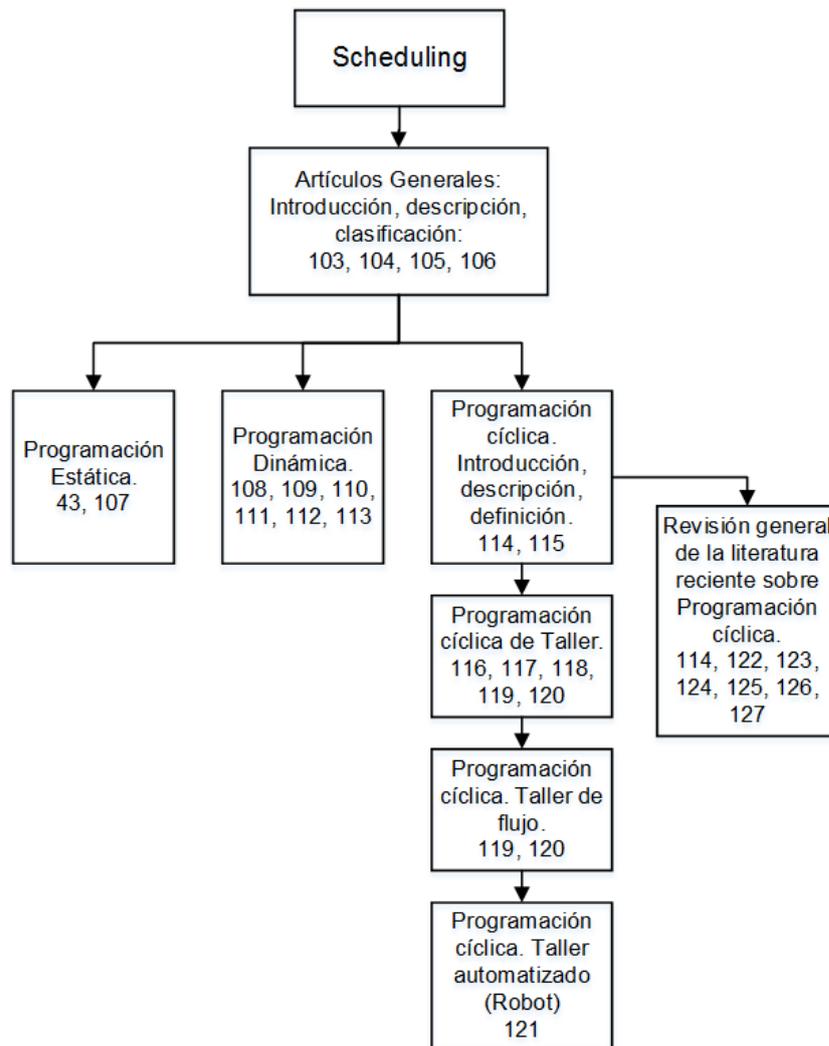


Figura 18: Revisión de Literatura en Problemas con *scheduling* relevantes a la Tesis

Scheduling se define como el proceso de asignar operaciones a recursos a lo largo del tiempo con el objetivo de optimizar un criterio (102). Entre los criterios que se mencionan en la literatura están: minimizar el tiempo de término del grupo de operaciones en evaluación, la minimización de la media del inventario en proceso WIP (*Work in process*), la minimización del tiempo de manufactura promedio (el tiempo de flujo promedio), la minimización del tiempo de retraso promedio, la minimización del costo de proceso promedio, la maximización de la productividad, etc. Algunas restricciones aplican al grupo de operaciones, por ejemplo:

- Una orden parcial
- La necesidad de efectuar algunas operaciones dentro de períodos específicos (ventanas de tiempo)
- La necesidad de asignar algunas operaciones a recursos específicos
- La necesidad de acotar los tiempos de espera de los productos entre dos operaciones sucesivas.

La programación de tareas apoya a las principales áreas funcionales de una empresa:

- El departamento de producción necesita conocer la secuencia real de productos a fabricar en el día. Por ello, su personal ha de consultar la programación de tareas que resulta tras la planificación maestra de los diferentes productos a fabricar.
- El departamento de Contabilidad, a partir de la información de la programación y de las fechas de entrega a los clientes, prevé la entrada de ingresos, los costos de cada trabajo y realiza un análisis del flujo de efectivo de la empresa.
- El departamento de Marketing puede establecer medidas de la eficiencia de la programación para determinar si los tiempos de cumplimentación de los pedidos están aportando una ventaja competitiva y si las entregas se están realizando a tiempo. El conocimiento de los tiempos de entrega de cada pedido permite al departamento de Marketing ofrecer tiempos de entrega más ajustados.
- Un sistema de información se puede encargar de mantener en base de datos el tiempo de comienzo de cada tarea así como el enrutamiento de las mismas por cada uno de los recursos disponibles. También se puede encargar de realizar el seguimiento de movimiento del producto a través del proceso.

La investigación en teoría de *scheduling* se ha desarrollado en los últimos 40 años y ha sido objeto de mucha literatura con técnicas que van desde reglas de “despacho” no refinadas, hasta algoritmos paralelos de ramificación y poda altamente (103) sofisticados, heurísticas basadas en cuellos de botella y algoritmos genéticos paralelos (104;105). Además, dichas técnicas han sido formuladas desde un amplio espectro de investigadores, desde científicos de gestión, hasta expertos en producción.

3.4.2 Programación Estática.

Desde la 2ª Guerra Mundial, la administración de la producción evolucionó lentamente de una toma de decisiones local a una globalizada. El desarrollo del campo de la administración de operaciones ha sido primero limitado al ámbito de la mejora de las actividades del departamento de sistemas de producción, como son administración de inventarios, planeación y programación de las actividades de manufactura, administración de la calidad, administración del transporte, para mencionar solo algunos. Cada una de estas actividades estaba normalmente bajo la responsabilidad de un administrador particular cuyo objetivo era tomar decisiones por un período determinado. Por ejemplo, los administradores de la manufactura debían tomar las decisiones de programación durante una semana siendo el objetivo alguno de los siguientes: balancear la carga de las máquinas, alcanzar tiempo límite intermedio, o efectuar la mayoría de las operaciones durante la semana de acuerdo a prioridades particulares. Por otra parte, las decisiones se podían congelar durante el transcurso de la semana, sólo en caso de ser necesario eran ajustadas a la mitad del período.

Los primeros programas eran realizados “a mano” utilizando Gráficas de Gantt, estos quedaban fijos hasta el final del horizonte de planeación y podían ser modificados con base en la experiencia del administrador, en el caso de la aparición de eventos inesperados. En otras palabras, los ajustes globales de los diferentes departamentos era hecho al inicio de cada periodo y se hacían adaptaciones en el caso de intermitencias.

Cuando los ordenadores fueron de uso común, los programas cambiaron de los gráficos de Gantt a las pantallas de los ordenadores. Al mismo tiempo, se desarrollaron programas computacionales para calcular programas de producción que optimizaban algún criterio en el horizonte de planeación.

Los problemas de programación estática normalmente son NP hard. Existe una amplia literatura cuando los tiempos de operación son deterministas (43;106).

Existen dos razones para la debilidad histórica de los sistemas de programación estáticos: por un lado los sistemas de procesamiento de datos y de comunicación eran débiles y por otro lado existía competencia entre empresas relativamente limitada.

3.4.3 Programación Dinámica:

En la mayor parte de las situaciones del mundo real aparecen eventos inesperados que provocan que la planificación sea revisada y adaptada. Los eventos inesperados pueden estar relacionados tanto con los recursos como con las operaciones. (107) (108) y (109).

Los eventos relacionados con los recursos pueden ser:

- Descompostura de maquinaria
- Fallas en la herramienta
- No disponibilidad de herramientas o personal
- Faltantes de materiales o componentes
- Materiales o componentes defectuosos o inadecuados
- Etc.

Los eventos relacionados con la operación son por ejemplo:

- Cambios en la fecha de entrega
- Cancelación de órdenes
- Llegada tarde de órdenes
- Cambios en los procesos de manufactura debido a cambios de los recursos.

Se puede decir que frecuentemente un programa se vuelve obsoleto al mismo tiempo que se completa (110). Algunos autores insisten en la brecha que separa la teoría de la programación estática del ambiente real donde frecuentemente existen interrupciones e intermitencias provocadas por eventos inesperados (107) y (111). Estas últimas requieren la programación dinámica, un conjunto de métodos capaces de responder a eventos inesperados sea

ajustando el programa existente o haciendo la reprogramación (*Rescheduling*) de las operaciones restantes.

En los últimos años ha cambiado la estructura de los sistemas de producción debido a la presión del mercado competitivo. Dicho mercado está apoyado en nuevos sistemas de procesamiento de datos, sistemas de comunicación más poderosos y por los tratados de comercio internacional permisivos. Esta situación ha requerido:

- La integración de las actividades que cubren el espectro completo desde los requerimientos de los clientes hasta la cobranza.
- Flexibilidad en relación a los cambios en la demanda.

La respuesta a estos requerimientos es dada por el “Paradigma de la cadena de suministro” (112). En realidad existen 4 paradigmas en la cadena de suministro, que la cadena sea: 1) Agil, 2) Esbelta, 3) Verde y 4) Resiliente. En (112) se utiliza como nomenclatura “LARG” para denotarlos pues el concepto es más amplio que solo incluir el que sea esbelta, además de que existen antecedentes claros en artículos anteriores del mismo autor de integrar los demás paradigmas en la cadena de suministro al haber incluido en el pasado aspectos económicos, operacionales y de desempeño ambiental. Una cadena de suministro es una red global de organizaciones que cooperan para mejorar los flujos de materiales e información entre proveedores y clientes con el costo más bajo y la máxima velocidad. Por tanto una cadena de suministro está organizada por proyectos en lugar de departamentos. El objetivo de planificación en tales sistemas no es ya programar óptimamente una serie de tareas. El objetivo es ahora asignar tareas a los recursos en línea, en el orden en el que las demanda aparece en el sistema. Esta actividad se conoce como asignación en tiempo real (*Real Time Assignment*).

Dependiendo del tipo de problema, pueden necesitarse dos tipos diferentes de asignación de tarea en tiempo real. El primer tipo consiste en asignar tareas a recursos sin cambiar el programa previo. Se conoce a este tipo de programación como “Asignación en línea utilizando periodos de inactividad” (OAIP *Online Assignment using Idle Periods*). El segundo tipo de programación permite algunos cambios limitados en el programa existente. Este tipo de programación se conoce como “Asignación en línea con reprogramación parcial” (OAPR, *Online Assignment with Partial Rescheduling*)

La programación dinámica considera un sistema de producción como una concatenación de departamentos que están bajo la responsabilidad de distintos administradores, Se utiliza en particular en el departamento de manufactura donde un programa estático se realiza al inicio del período y este se va ajustando en la medida en que surgen eventos inesperados durante el período de vigencia. Este programa comprende un grupo de operaciones que corresponden a diferentes productos que deben ser completados en un horizonte de tiempo específico (una semana por ejemplo).

La asignación en tiempo real surge cuando el paradigma de la cadena de suministros empezó a ser más común. En ella, cada proyecto incluye todo el ciclo de producción desde los requisitos de los clientes a la cobranza, adicionalmente, la diversidad de los productos es limitada en función del número de tipos de operaciones. Un proyecto es un flujo continuo de

actividades. En particular, los sistemas de manufactura cambian progresivamente de talleres de trabajo a líneas de ensamble que, adecuadamente balanceadas, garantizan tanto productividad como flexibilidad. La asignación en tiempo real consiste en asignar un grupo de operaciones relacionadas con un producto (o con grupo pequeño de tipos de productos) a un grupo de recursos tan pronto como la orden llega al sistema de fabricación. En caso de existir intermitencias, las operaciones previamente programadas que aún no hayan sido completadas son reprogramadas (*Rescheduling*) en el orden de llegada correspondiente. La meta es reasignar las operaciones en tiempo real.

3.4.4 Programación Cíclica:

3.4.4.1 Introducción:

La programación cíclica es un tipo de programación que no puede ser restringida a los casos en la evolución global descrita anteriormente pues es de suma utilidad para los sistemas de producción masiva con productos diversificados. En las últimas décadas ha habido un interés creciente en los problemas de programación cíclicos tanto en la comunidad académica como en la práctica industrial.

Existen muchos ejemplos de programación cíclica en áreas tanto industriales como de servicios, entre otras: robótica, planeación logística y del transporte, comunicaciones, computación con procesadores múltiples y programación de cambios de turno en enfermería. Esta última conocida como "*Nurse Rostering and Scheduling Problem*" (NRP) en investigación de operaciones y que consiste en encontrar la manera óptima de asignación de enfermeras a los turnos, haciendo consideración de todas las soluciones válidas de restricciones estrictas e inevitables, y un conjunto de restricciones blandas que definen la calidad relativa de las soluciones válidas. Este problema ha sido estudiado desde los años 60's. Las soluciones al problema de programación de la enfermera se pueden aplicar también a problemas de programación en otros campos.

Las ventajas de la programación cíclica sobre la programación estática (no cíclica) en la manufactura flexible han sido ampliamente discutidas en la literatura. Las principales ventajas estudiadas son 1) La superioridad de los programas cíclicos con respecto al manejo eficiente de los materiales, 2) La mejor utilización de las estaciones de trabajo y 3) Un control de piso en la planta más simple (113).

3.4.4.2 Descripción y definiciones:

La programación cíclica o periódica es una forma efectiva de optimizar algunos procesos, entre otros: de manufactura, transporte y de computación. Resulta de mucha utilidad inclusive cuando se hacen consideraciones de *setup* y de tiempos de espera o de transporte.

Dentro de la manufactura flexible históricamente la programación cíclica se ha clasificado en dos categorías y en la primera se han considerado dos subcategorías:

- 1) Problema de programación de taller (Jobshop):
 - a. Programación Cíclica de Taller de flujo.
 - b. Programación Cíclica Automatizada (Robot)
- 2) PERT-shop

En el entorno de la programación de taller tradicional normalmente se considera despreciable tanto el tiempo de *setup* como el de transporte y en muchas ocasiones no son considerados al hacer la optimización. Esto sucede porque el equipo en los centros de maquinado están preparados para hacer los cambios de producto en forma rápida y eficiente. Sin embargo en los procesos automatizados con equipos y sistemas de manufactura flexible con robots controlados (Sistemas CAM) que incluyen sistemas de manejo de materiales tales como polipastos, montacargas y grúas los tiempos de transporte y *setup* no deben ignorarse. El sistema de manejo de materiales y los tiempos de preparación forman parte explícita de los modelos de programación. El problema se define como identificar un ciclo robotizado que maximice la productividad.

La segunda categoría de problemas de secuenciación cíclica se enfoca en la programación de proyectos PERT/CPM que periódicamente se repiten. El objetivo en este caso es coordinar los tiempos de una larga cantidad de operaciones interrelacionadas que se realizan en forma simultánea por máquinas en paralelo (procesadores u operadores).

La programación cíclica tiene una amplia aplicación en la práctica, desde sistemas de producción sin restricción de recursos, programación de proyectos que se repiten, de sistemas computacionales paralelos, CAD (*Computer Aided Design*) de sistemas digitales y programación de semáforos (113).

Por muchos años se ha observado una necesidad creciente en la demanda del mercado por la diversidad de productos (*Multiassortment*), es decir el cliente requiere que en una misma compra el proveedor entregue una amplia variedad de diseños del mismo producto. Esto es posible lograrlo, dentro de otras posibilidades por medio de la fabricación cíclica. Se fabrica en intervalos de tiempo fijos (tiempo de ciclo) un lote específico de una mezcla de productos específica. La optimización normalmente se reduce a la minimización del tiempo de ciclo. En este caso la adecuada selección de una mezcla y un tiempo de ciclo permite no solo cumplir la demanda sino también lograr la eficiencia en el uso de la maquinaria y equipo. Desde los puntos de vista teórico y práctico aun cuando la mayor parte de los problemas en el campo industrial son NP-hard se puede observar un alto interés en la investigación de este tema dada su importancia y frecuencia (114).

3.4.4.2.1 Programación cíclica de taller (Jobshop):

El sistema de producción cíclico en un taller incluye recursos, estos son máquinas o estaciones de trabajo. Se procesan múltiples productos y cada uno de ellos tiene copias a las que se llama "trabajos". Cada trabajo tiene su propio orden de procesamiento en las máquinas. El sistema puede tener o carecer de inventario en proceso entre cada estación. En caso de existir inventario en proceso este puede ser de capacidad finita o infinita. Las reglas de

programación generalmente son distintas para los problemas con inventario o sin él.

Antes de iniciar un proceso de secuenciación, el proceso de producción involucra dos procedimientos previos: 1) Determinar el *layout* del sistema de producción y la proporción en que serán fabricados los productos. 2) Asignar las operaciones de cada trabajo a las máquinas.

En forma repetitiva los trabajos son procesados en conjuntos con la misma proporción de productos en que son requeridos para cumplir con una meta de producción. Al conjunto más pequeño posible que represente la proporción de la meta se le llama: conjunto de partes mínimas (MPS, *minimal part set* en inglés). Un ejemplo se puede ver si se tiene una meta de 1000 unidades distribuidas en 400 unidades del producto A, 300 unidades del producto B, 200 unidades del producto C y 100 unidades del producto D entonces el MPS es: 4 unidades de A, 3 unidades de B, 2 unidades de C y 1 unidad de D, en total son 10 unidades que serán repetidas 100 veces para completar la meta requerida.

De acuerdo a Hall, Lee & Posner (115) dado un conjunto de máquinas M, una línea de producción estructurada como un taller con secuenciación cíclica requiere que las operaciones de cada trabajo del MPS sean asignadas a las máquinas con antelación y que el ruteo (orden de fabricación) de cada trabajo que pasa por las máquinas (no necesariamente el mismo para cada trabajo) sea fijo y conocido. El problema de programación cíclica de taller es encontrar el orden de procesamiento en el cuál son procesadas repetidamente en cada máquina, considerando que esto se realizará en un tiempo indefinido o infinito. Este orden junto con los tiempos de inicialización o *setup* de todas las operaciones se llama programa cíclico o programa periódico. Al intervalo de tiempo regular en el que todas las operaciones son repetidas se le llama periodo o ciclo de tiempo.

En el artículo de Hall, Lee & Posner (115) se asume que los tiempos de preparación o *setup* y los tiempos de transporte son despreciables y además que los tiempos de proceso son fijos. Sin embargo, en muchos casos importantes y para hacer un análisis más realista o con aplicación práctica, estas consideraciones deben evitarse como se muestra en las siguientes referencias, consideradas desde los años 90's hasta fechas recientes.

Roundy (116) analizó el problema cíclico dentro de un taller considerando que todos los trabajos son idénticos, asumiendo que un solo producto (unidades idénticas) se produce en un número finito de máquinas. La manufactura de cada parte se realiza dando una serie de operaciones en una secuencia determinada. Cada operación se puede hacer exclusivamente en una máquina específica. El problema sería trivial si se pudiera asignar exactamente una operación a una máquina, lo que ocurre en un taller de flujo. El alto coste de la maquinaria obliga al programador de producción a utilizar la capacidad al máximo lo que ocasiona que la misma unidad vuelva a pasar por la misma máquina para efectuar una operación distinta. En este artículo entre otros objetivos se buscó comparar la efectividad de un sistema como el que se describe con un taller de flujo.

En las instalaciones de manufactura automatizadas para un solo producto con niveles de productividad altos el diseño de productos y procesos

de las partes que serán producidas es sumamente cuidadoso. Se toman en cuenta dos metas importantes: 1) Hacer el tiempo de *setup* tan reducido como sea posible y 2) Hacer que los requerimientos de proceso de las diferentes partes a producir sean similares, tanto como el proceso lo permita. En muchos casos si esto se hace bien, los requerimientos de los diferentes productos son prácticamente los mismos. En muchas líneas automatizadas, los *setups* más grandes se realizan periódicamente en toda la línea de producción. En el tiempo que transcurre entre estos *setups* solamente se producen piezas similares, hay caso en los que solamente se hace un solo tipo. El modelo que se presenta aplica en situaciones en las que por un periodo específico se fabrican partes que tienen características de proceso muy similares y los costes de *setup* son despreciables. En el artículo de Roundy (116) se demuestra que el problema es NP-hard y se estudian algunas propiedades clave de la programación cíclica. Con base en estas propiedades clave se propone un algoritmo *branch & bound* y se presentan algunos resultados preliminares con el mismo.

Matsuoo, Shang & Sullivan en (117) revisan el problema de programación de una grúa que debe mover productos distintos en un ambiente de manufactura integrada por computadora CIM (*Computer Integrated Manufacturing*). La grúa debe mover todo el producto en proceso (WIP) en el sistema. El objetivo general es maximizar la tasa de rendimiento sujeta al límite de tiempo del WIP. Se muestra al formalizar el problema que la secuenciación cíclica genera una solución cercana al óptimo y es superior a las reglas de secuenciación. Inicialmente se analiza el caso de un solo producto y después se extiende el estudio al caso multi-producto. En este segundo caso se demuestra que para una secuencia dada, encontrar el tiempo de ciclo mínimo se convierte en un problema de costo máximo en una red de flujo circular. Con base en lo anterior se desarrolla una heurística de secuenciación con un algoritmo de tiempo polinómico considerando diferentes tipos de productos en un ciclo y se logra en la mayor parte de los casos minimizar el tiempo de ciclo. Al final se reportan los experimentos computacionales y se verifican empíricamente varias afirmaciones hechas en el artículo.

Claire Hanen en (118) estudia un taller cíclico con restricciones de precedencia entre las operaciones al que se conoce como taller recurrente, la clasificación de este problema se puede conceptualizar como un problema de secuenciación de proyectos cíclicos en los que se agregan máquinas. Ella demostró que el problema es *NP-hard* y que se puede modelar a través de programación entera mixta (MIP). En este artículo se probaron varias soluciones factibles. Se determinó que cualquier solución factible está asociada a una valuación llamada "altura conservativa" en el grafo de restricciones. Al resolver computacionalmente el circuito crítico se consigue la mejor solución con una "altura conservativa" específica. Con lo anterior se pueden establecer límites o fronteras en las "alturas conservativas". Finalmente se proponen una búsqueda *branch & bound* y dos heurísticas para resolver el problema y se reportan los resultados experimentales.

Hall et al. (119) revisaron una versión diferente del problema del taller cíclico que considera solamente dos máquinas, probaron que es un problema *NP-hard*, y aportaron un algoritmo en tiempo polinómico para el caso en el que cada trabajo tiene a lo sumo dos operaciones.

Este artículo considera la programación de operaciones en una célula de manufactura que en forma repetitiva produce una familia de partes similares en varias máquinas que utilizan un robot. Entre las decisiones que se necesitan hacer se incluyen las de encontrar el ciclo de movimientos del robot y la secuencia de partes que en conjunto minimizan el tiempo ciclo de producción, o en forma equivalente, maximizan la tasa de salida. En este artículo los autores se enfocan en aspectos de complejidad y de desempeño. Por ejemplo en una célula de tres máquinas produciendo partes de tipos múltiples, han probado que en dos de cada seis ciclos óptimos de movimiento del robot para producir una pieza, la versión de reconocimiento de programa de secuenciación de la parte es *NP-hard* completo en la versión unaria. El resto de ciclos, los otros cuatro, muestran también que pueden definir eficientemente problemas resolubles de secuenciación de partes. Se demuestra que el problema general de secuenciación de partes no restringido a ningún ciclo de movimientos del robot en una célula de tres máquinas es *NP-completo* en la versión unaria. Finalmente se discute la forma en que una celda robótica converge a un estado estable.

Lee & Posner (115) revisaron un caso especial del problema de taller cíclico donde el orden de procesamiento de las operaciones en las máquinas se conoce y es fijo. En este caso el problema de secuenciación se elimina. Este caso del problema cíclico de secuenciación puede resolverse utilizando los mismos algoritmos que para el problema de secuenciación de proyectos cíclicos y del problema de programación cíclica automatizada. Adicionalmente en este artículo se menciona el uso de un MPS que se repite en forma constante y se discuten tanto el comportamiento como el programa. Se plantean dos formas de hacer la medición: el tiempo de ciclo y el tiempo para hacer un trabajo, conocido como *makespan* y que es similar a la ruta crítica de un proyecto en PERT, y se demuestra que ambos están directamente relacionados. El tiempo mínimo de ciclo se identifica como la medida de un circuito en un grafo. Se establece que existe una clase en los programas que minimiza el tiempo de ciclo y que repite constantemente un patrón de tiempo cada MPS. Se desarrolla un algoritmo que construye esos programas y como segundo criterio se busca minimizar el *makespan* además de otras medidas de desempeño. Para minimizar el *makespan* se revisa el programa de inicio más temprano donde cada operación inicia lo antes posible. Se caracterizan los casos donde, después de un número finito de MPS's, el programa de inicio más temprano repite un patrón de tiempo idéntico cada número fijo de MPS's. Se hace también una exploración modificando el modelo a un programa de inicio temprano que repite un patrón de tiempo idéntico en cada MPS cuando el inicio de las operaciones se retrasa.

El problema de programación cíclica de taller puede presentar variantes, por ejemplo, considerar capacidades de inventario en proceso limitadas o ilimitadas, máquinas en paralelo, bloqueo de posiciones, condiciones de no espera etc. Existen muchos métodos de solución genéricos tales como el *Branch & Bound*, Relajación de *Lagrange* (Es un método de relajación en el que se aproxima a un problema difícil de optimización restringida mediante un problema más simple, la solución al problema relajado es una solución aproximada al problema original y proporciona información útil), programación lineal entera mixta, procedimientos de interacción humano-ordenador. A través

de ellos se puede obtener una idea más clara del amplio panorama de posibilidades que presenta el problema general de la programación cíclica de taller.

3.4.4.2.2 Programación cíclica del taller de flujo (Flowshop):

Se considera que la programación cíclica del taller de flujo es una subclase del problema de programación cíclica de taller. Algunos de estos problemas asumen que los tiempos de *setup* y de transporte son despreciables mientras existen otros que en forma explícita consideran el equipo de manejo de materiales con tiempos de operación no despreciable.

En el taller de flujo se conoce el número de máquinas. Estas máquinas están organizadas en series. Todos los trabajos (Unidades en el MPS) deben pasar a través de las máquinas en el mismo orden, el cual es definido con antelación. Para esta subclase del problema de taller se puede asumir que el número de operaciones para cada trabajo es igual al número de máquinas, y las operaciones se numeran de tal forma que la operación "*k*" del trabajo "*j*" debe ser realizada en la máquina "*k*". Cada operación para un trabajo específico en una máquina determinada tiene definido un tiempo de proceso. Este tiempo es conocido y constante pero solo como se ha indicado para la operación del trabajo que se realiza en esa máquina, si fuese un trabajo distinto, la duración también podría serlo. Existe una versión de este problema en la que los tiempos de procesamiento pueden ser desconocidos pero fluctúan entre intervalos determinados. Un sistema de transportación de cadena o algún equipo de manejo de materiales mueve los trabajos de y hacia las estaciones de trabajo que son fijas y que es el sitio donde se realizan las operaciones diseñadas para cada trabajo. Los tiempos de transporte normalmente se consideran pequeños y despreciables. El problema del flujo de taller consiste en encontrar el orden (La secuencia de cada trabajo para cada máquina) en la que los trabajos se procesan en forma repetitiva y pasan por todas las máquinas.

3.4.4.2.3 Programación cíclica automatizada (Robot):

De acuerdo a (120) este tipo de problemas fue introducido a inicios de los años 60's por un grupo de matemáticos de Bielorrusia (Aizenshtat, Suprunenko y Metelsky) quienes estudiaron procesos cíclicos en líneas de manufactura con transportadores integrados. Los transportadores se diferencian de otras máquinas en sus características y funciones. Los mencionados investigadores introdujeron el problema de taller de flujo robótico cíclico y propusieron un método de solución combinatorio llamado "*el método de intervalos prohibidos*", que más adelante fue utilizado como punto de partida para investigaciones más profundas hechas por diferentes investigadores para resolver varios problemas de programación robótica cíclica.

Se considera un sistema de producción que consiste en *m* máquinas que pueden ser estructuradas sea como taller de trabajos o taller de flujo. La diferencia entre ambos es que en el primero todos los trabajos pasan por las máquinas en el mismo orden de secuencia, mientras que en el segundo cada trabajo tiene su propia ruta tecnológica a través de las máquinas. Uno o varios robots controlados a través de ordenador u otros sistemas de manejo de

materiales transportan las partes de y hacia las máquinas fijas que realizan las tareas diseñadas para los trabajos. La duración de las operaciones de transporte y los movimientos del robot de carga o vaciado no son despreciables, de tal forma que el desempeño y la productividad del sistema depende en gran medida en la eficiencia de los sistemas de transporte. El problema de programación cíclica con robot es encontrar el orden en el que las partes son procesadas en las m máquinas, en conjunto con las rutas en las cuales los robots mueven las piezas. Cada robot puede ser considerado como una máquina de propósito especial, así que el sistema de producción con robots puede de hecho considerarse como un caso especial del taller de trabajos. Como se ha mencionado los dispositivos de transporte son distintos a otras máquinas en sus características físicas, estructura y funciones. Estas diferencias permiten realizar algunas acciones particulares en el problema de programación robotizado, estas han sido reconocidas por investigadores y por la gente que los utiliza.

De manera similar al problema de taller de trabajos no robotizado o al taller de flujo, el problema de programación cíclico robotizado es *NP-hard*, (114). Ellos demostraron que cuando se tiene una ruta de robot específica, el tiempo de ciclo mínimo puede ser encontrado a través de programación lineal. La mayor parte de los trabajos consideran el caso de un solo robot en el sistema. Un caso especial del modelo general es el de programación cíclica de una grúa con un solo producto y considerando una secuencia fija en las operaciones del robot/grúa. En este caso el aspecto de secuenciación se elimina y el problema se reduce a encontrar el tiempo de procesamiento exacto de los intervalos dados.

3.4.4.3 Revisión de la literatura:

Una revisión exhaustiva del estado del conocimiento sobre la secuenciación de tareas cíclicas se puede encontrar en el trabajo de (113) quienes analizan las cuestiones relativas a la complejidad de los algoritmos computacionales para resolver varios tipos de problemas de programación cíclica. En particular ellos consideran problemas *NP-hard* de varios tipos de ciclo e incluyen también una amplia variedad de criterios y restricciones adicionales por ejemplo: sin posibilidad de tiempo de espera o sin búfer (Sin inventario de reserva).

Los problemas de programación cíclica forman una parte relativamente pequeña de los problemas de programación. Sin embargo es un tema de investigación que se revisa con mucho interés debido a su importancia intentando resolver las dificultades que se presentan en la construcción de algoritmos eficientes. El hecho de que sean problemas *NP-hard* limita mucho la posibilidad de aplicación de algoritmos específicamente diseñados para casos con una variedad de instancias muy pequeña.

Después del trabajo desarrollado por (113) no se ha realizado una nueva revisión profunda, pero si se ha extendido la investigación en varias instancias y casos de la programación cíclica. Entre otros se pueden citar los trabajos de Tadeusz Sawik (121). Presenta modelos de programación entera mixta utilizando en unos casos lotes determinísticos y en otros programación cíclica en el taller de flujo considerando máquinas en paralelo y con inventarios en

proceso finitos. Se hacen comparativos entre modelos con todas las máquinas disponibles para el procesamiento con disponibilidad de todo el horizonte de programación y con casos para programar un patrón arbitrario de disponibilidad de las máquinas debido a posibles eventos de inactividad pre-programados. Sawik presenta ejemplos numéricos tomados de la programación flexible y real de plantas de manufactura electrónica. Al final se presentan los resultados de experimentos computacionales tanto con disponibilidad continua como parcial de las máquinas. En las conclusiones se presenta que los resultados obtenidos para la programación cíclica son significativamente superiores a los obtenidos en la modelación por lotes para ambos casos: tanto para la disponibilidad continua como para la disponibilidad limitada. En la práctica la programación cíclica o la programación con lotes cíclicos frecuentemente se simplifica descomponiéndola en dos niveles: primero se encuentra un óptimo general que se encuentra utilizando un MPS (Programa maestro de producción) simple y enseguida se construye un programa cíclico óptimo o un programa de lote cíclico óptimo que consiste en solo repetir el programa encontrado con el MPS simple hasta completar la demanda, (120) y (113).

El trabajo de Brucker et al. (122) incluye consideraciones de transporte que puede ser utilizado para describir problemas de optimización en sistemas de manufactura completamente automatizados incluyendo líneas de ensamble. En este estudio se eliminan los inventarios en proceso, lo cual reduce en forma relevante el número de soluciones factibles, pero al mismo tiempo incrementa la dificultad, haciendo más complejo el que estas se puedan encontrar. En este estudio se da relevancia al hecho de que en el entorno físico de los procesos industriales la programación no puede modelarse como el problema clásico del flujo de taller sino que es de suma importancia tomar en consideración las restricciones del mundo real tales como el manejo de materiales, los espacios de almacenamiento y los tiempos de preparación o *setup* de las máquinas. El enfoque principal en este artículo fue caracterizar los ciclos del robot que son viables en forma eficiente. Esto se hizo utilizando un procedimiento *branch & bound* vinculado a un modelo de programación entera mixta resuelto con CPLEX. Los resultados fueron competitivos aunque todavía queda margen para la mejora. Especialmente en el caso de los límites inferiores, la estrategia de búsqueda es subóptima aun cuando se encuentran soluciones factibles no triviales. En las conclusiones de este estudio se indica que queda pendiente hacer un cambio de estrategia generando una mezcla entre encontrar buenas soluciones factibles y mejorar los límites inferiores, utilizando el mismo marco de construcción del PRC y verificar la factibilidad del mismo. Otro problema es que para valores grandes de los datos se obtienen peores resultados que con valores pequeños utilizando las mismas instancias. Obviamente el espacio de búsqueda crece a medida que se incrementan los valores y hace falta desarrollar alguna heurística que mejore el resultado. Se observa la posibilidad de mejorar la formulación MIP incluyendo restricciones más específicas.

M. Rocha et al. (123) trabajan en la optimización de tres modelos de la vida real que requieren secuenciación cíclica en la fuerza de trabajo: una fábrica de vidrio, una unidad de cuidado y también en una colección de casos de *benchmark* disponibles en la literatura. El énfasis se puso en una novedosa forma de utilizar las restricciones de secuenciación y en el balanceo de la carga de trabajo que es abordada a través de secuenciación cíclica. Los modelos

utilizan CPLEX para resolver los casos. Los resultados demuestran que es posible encontrar soluciones óptimas en un tiempo razonable. En principio se desarrolló un modelo general para el problema de secuenciación de un grupo de trabajo en una organización que trabaje en forma continua 24 horas. El modelo considera variables como: número de turnos en el día, número de equipos de empleados con características similares que deben ser asignados a un trabajo o a una parte de un turno en cada uno de los días laborables planeados. La demanda diaria del turno debe ser alcanzada, considerando que el modelo debe garantizar que exista un determinado número de equipos trabajando en cada turno de cada día. Las reglas laborales incluyen un mínimo y un máximo de días consecutivos laborables para cada equipo de trabajo y una secuencia preferida de turnos que se debe respetar. Cada turno debe considerar un día de corte (break) o día no laborable. El objetivo es minimizar y nivelar el número de días que cada equipo trabaja en cada turno para obtener una carga laboral balanceada. Se realizó un modelo matemático general que después fue adaptado para cada uno de los tres casos mencionados debido a consideraciones particulares de cada uno. Por ejemplo, en la fábrica de vidrio se tenían que contratar equipos adicionales a los grupos normales para conservar el nivel de producción requerido durante el verano. Estos equipos adicionales necesitaban entrenamiento previo durante algunas semanas pero la compañía no podía evitar tener un mayor número de problemas de calidad en esa temporada. Adicionalmente el programa para los equipos existentes no era conveniente. Por estas razones los directivos requerían hacer un estudio con un mayor número de grupos operando en forma normal. En el caso de la unidad de cuidado fue necesario adaptar el modelo general al ambiente de un servicio. Se tenían 4 turnos distribuidos a lo largo del día con número de horas heterogéneo y con actividades cíclicas distintas en cada uno. La cantidad de trabajadores de tiempo completo especializado en un turno específico ahora forma parte de un grupo general con habilidades flexibles. El objetivo del modelo es aquí minimizar los requerimientos de trabajadores de tiempo parcial asumiendo que la carga de los trabajadores de tiempo completo se asigne en su totalidad. Adicionalmente se adaptó el modelo a problemas de secuenciación disponibles en la literatura para probar la efectividad del mismo. Después de probar con un modelo MIP la novedad de la formulación propuesta es tanto la flexibilidad y el amplio rango en que se abordaron las condiciones de secuenciación y el enfoque cíclico en un problema que inicialmente no lo es para asegurar la equidad y la previsibilidad de la secuencia. El sistema que se propone asegura que no solo se asegura la reducción de costes dada la adecuada gestión de los recursos sino también un incremento en la productividad al mejorar la motivación y los niveles de satisfacción de los empleados.

T. Sawik en (124) presenta una nueva formulación entera mixta para la secuenciación cíclica en líneas de flujo con máquinas en paralelo y con restricciones finitas de inventario en proceso. Define en este artículo el uso de un grupo mínimo de productos al que llama MPS (*Minimal Part Set*). Este grupo representa en las mismas proporciones pero en una cantidad mínima, los objetivos de producción. El ciclo para el MPS no se determina a priori sino con el programa óptimo para todas las partes. Adicionalmente al programa cíclico se introduce un modelo de programación cíclico por lote, donde dentro del MPS las partes del mismo tipo se procesan en forma consecutiva. Se presentan

ejemplos numéricos y se reportan algunos resultados de los experimentos computacionales. Tomando como base las definiciones dadas en (121) en este artículo el autor presenta una extensión del mismo, en este caso no hace un comparativo con el comportamiento de la programación por lote sino que profundiza en el caso cíclico. La literatura en la secuenciación cíclica de un MPS en las líneas de flujo se concentra en el estudio de configuraciones sincrónicas o asincrónicas. Esto significa que tienen o no relación con un límite externo en el tiempo de ciclo. En contraste con un enfoque común de dos niveles en el que se repite indefinidamente el programa óptimo para un MPS en este trabajo, ese programa no es determinado a priori sino que se determina con el programa óptimo para todas las partes, adicionalmente el modo restringido por lote se modela al interior de cada MPS. Los resultados experimentales indican que el enfoque propuesto minimiza el tiempo de terminación de los trabajos comparado con la simple repetición del MPS. En general el tamaño del MIP se incrementa rápidamente en forma proporcional al tamaño de la orden de producción, es decir, se incrementa con el número de veces que se debe programar un MPS y esto es igualmente proporcional al tiempo requerido para encontrar una secuencia óptima.

El sector industrial alimentario también ha sido explorado en la literatura para revisar la aplicación de modelos de secuenciación cíclicos. En (125) los autores abordan un problema de secuenciación cíclica realista en un ambiente industrial de alimentos. En este problema máquinas en paralelo procesan productos perecederos con límites en variables tales como: fechas de lanzamiento, fechas de vencimiento y plazos. El producto está sujeto a un límite en el tiempo de vida en anaquel después de haber sido producido, y debe ser entregado a los distribuidores dentro del tiempo de ventana dado por la fecha de vencimiento del producto y el plazo límite de entrega. En el control de los procesos de alimentos la seguridad es uno de los principales objetivos por lo que se debe asegurar que el tiempo de sobrevida para el producto sea mayor al planeado para el anaquel. Los trabajos entregados tanto con retraso como con antelación son penalizados. El objetivo en este problema es encontrar un programa cíclico que minimice los costes totales. Shirvani et al. (125) diseñaron un modelo de programación mixta entera e incluyen un algoritmo heurístico para generar soluciones factibles. Adicionalmente desarrollaron un algoritmo iterativo *greedy* para mejorar la calidad de las soluciones. Un algoritmo *greedy* busca el óptimo local en una serie de etapas esperando encontrar el óptimo global. En muchos casos el algoritmo *greedy* no encuentra la solución óptima pero puede encontrar soluciones locales que se aproximan a la óptima global en un tiempo razonable. Este trabajo es una extensión del que hicieron Kaplan y Rabadí (126). Ambos analizan un sistema con fechas de vencimiento y plazos tope. En el de Kaplan y Rabai se analiza un sistema de recarga de turbosina en aviones que permite la entrega anticipada, cosa que no sucede con el artículo de la industria alimentaria pues esto se penaliza con un coste por la pérdida de tiempo de vida del producto en anaquel así mismo se muestra que el problema es *NP-hard*. Shirvani et al. generaron varias instancias en forma aleatoria en las que el tamaño del ciclo, el número de trabajo, y el número de máquina son los principales parámetros que determinan el tamaño de las mismas. Los resultados del algoritmo heurístico se contrastaron con los del algoritmo iterativo *greedy*. Este último supera al primero. Se probaron varias versiones para evaluar el efecto de la búsqueda

local. El algoritmo *greedy* genera resultados con costes menores de entrega tardía o anticipada. Consideran como futuras extensiones al problema el análisis de tiempos y costes de *setup*.

3.5 Programación Multi-Objetivo:

3.5.1 Introducción

Se pueden considerar varios criterios en un problema de programación Multi-objetivo pero, en este caso, existen varias técnicas disponibles para convertir el problema multi-criterio en uno de criterio único. Entre estas técnicas están la suma ponderada de los criterios, la conversión del problema multi-criterio en uno de criterio simple considerando todas las restricciones pero una a la vez en forma iterativa, la jerarquía entre los criterios, la aplicación del método de programación de metas "*goal programming method*" que consiste en minimizar una suma ponderada de desviaciones del criterio respecto a valores predeterminados, la aplicación de la llamada "*compromise programming*" que consiste en acercarse a una solución "ideal" tan cerca como sea posible.

En el caso de esta tesis la situación Multi-objetivo, específicamente bi-objetivo es relevante en el primer problema que se analiza. Es decir en el Problema Sin Huecos. La razón por la que se presenta una breve revisión de la literatura es porque en el caso del segundo problema, el Problema Con Huecos, aunque en origen se trata de una situación bi-objetivo, el efecto que se analiza es mono-objetivo. Es decir aunque existe los cambios de geometría o color estos en cualquier caso son la causa de un solo efecto: la generación de un espacio vacío en la secuencia de producción. Por tanto en el segundo problema la conversión de un problema multi-criterio a uno de criterio único se realiza en el mismo proceso y es medible en forma independiente al origen de los cambios que los generaron. Se puede medir el número de espacios vacíos en lugar del número de cambios de modelos o colores. Dado que en esta Tesis la aportación principal se enfoca al segundo problema que es mono-objetivo, la revisión del caso bi-objetivo es general.

3.5.2 Optimización Multi-Objetivo.

La optimización busca los valores para las variables de un sistema que se deriven en el mejor desempeño en un objetivo dado. En la mayoría de las áreas de la ciencia se busca la optimización de algún sistema y los métodos para este fin son muy variados.

El primer paso para aplicar cualquier herramienta de optimización es definir el o los objetivos. Algunos ejemplos de estos objetivos pueden ser: minimizar los costos de algún proceso, maximizar las ganancias, obtener la mayor resistencia de un material, etc. Matemáticamente se le conoce como *función objetivo* a la función vectorial que determina el valor de la variable cuantitativa que queremos maximizar o minimizar.

$$\vec{f}(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), f_3(\vec{x}), \dots, f_k(\vec{x})]^T \quad (3.1)$$

La función objetivo describe cómo se comporta el sistema a ser optimizado bajo una elección particular de los valores de las variables de decisión. Cada uno de los componentes del vector resultante de evaluar la función objetivo en un vector de variables de decisión representa una medida de los criterios a optimizarse. Para que un problema sea considerado multiobjetivo se debe de contar con al menos dos objetivos contrapuestos, por ejemplo maximizar la potencia de un motor y minimizar su consumo de energía (127).

El siguiente paso es definir las variables de decisión y los parámetros del problema. Los parámetros permitirán ajustar las características del problema mientras que las variables serán los elementos que cuyo valor será determinado por la herramienta de optimización elegida para lograr el objetivo. Las variables de decisión se denotan por el vector:

$$\vec{x} = [x_1, x_2, x_3, \dots, x_n]^T \quad x_i \in \mathbb{R} \quad (3.2)$$

A una selección particular de los elementos de un vector de variables de decisión se le conoce como *solución* del problema.

Finalmente, se deben definir las restricciones que deben cumplir las variables. Las restricciones son un conjunto de ecuaciones o inecuaciones que establecen la relación entre distintas variables y parámetros. Por ejemplo, si una variable es una distancia, entonces ésta no puede tomar valores negativos. Las variables de decisión deben de satisfacer todas las restricciones del problema; cuando un vector de variables cumple con esta condición se le conoce como una *solución factible*, en caso de que no se cumpla con al menos una restricción se tiene una *solución no factible* (128).

Se dice que en un problema de optimización multiobjetivo busca el vector de variables de decisión $\vec{x}^* \in \mathbb{R}^n$ que optimice la función vectorial:

$$\vec{f}(\vec{x}) \quad f: \mathcal{F} \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^k \quad (3.3)$$

Donde \mathcal{F} representa la región factible y esta delimitada por l restricciones de desigualdad y m restricciones de igualdad:

$$g_i(\vec{x}) \leq 0 \quad , i = 1, \dots, l \quad (3.4)$$

$$h_j(\vec{x}) = 0 \quad , i = 1, \dots, m \quad (3.5)$$

Esta definición aplica para problemas de maximización y/o minimización y es posible transformar uno en otro cambiando la función objetivo (f_i) por ($-f_i$) (128). Ver Figura 19.

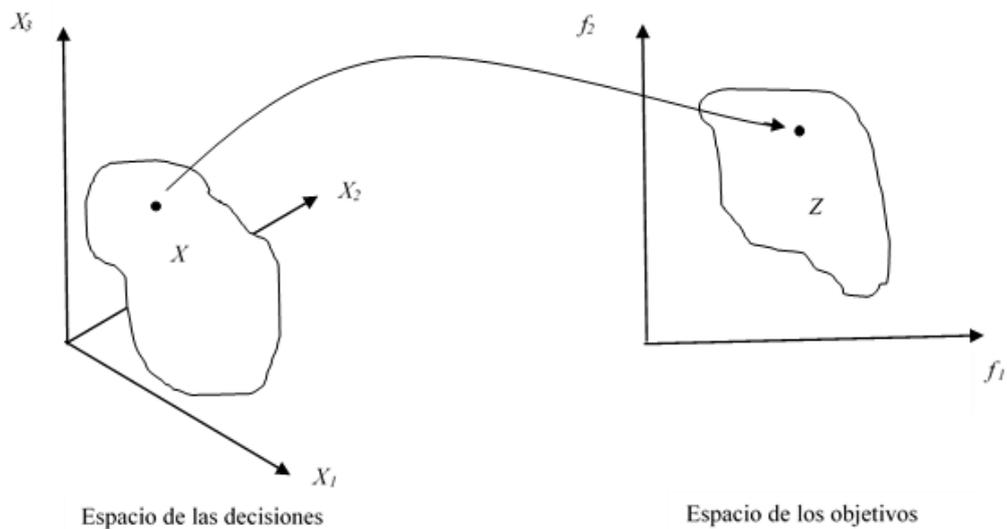


Figura 19: Representación del espacio de soluciones y objetivos

3.5.3 Frontera de Pareto

Para poder establecer de forma satisfactoria la solución óptima de un problema primero se debe poder determinar si una solución dada es mejor que otra. En los problemas enfocados a un único objetivo se utiliza la relación de “igual o menor” (\leq) para comparar los valores de las funciones objetivo escalares. En este caso se pueden dar varias soluciones factibles y ya que todas pueden ser comparadas y ordenadas de menor a mayor, solo una tendrá el valor óptimo (mínimo o máximo) para la función. De forma contrastante, los problemas multiobjetivo no poseen un orden canónico en \mathbb{R}^k y por consecuencia se requiere una relación comparativa más flexible.

La noción más aceptada de “óptimo” en el entorno de problemas multiobjetivo es la propuesta originalmente por Francis Ysidro Edgeworth en 1881 (129) y generalizada posteriormente por Vilfredo Pareto en 1896 (130).

El economista italiano Wilfredo Pareto formuló una serie de principios llamado Óptimo de Pareto. De acuerdo a Miller (131), Pareto señala “que cualquier cambio de situación afectaría a una economía sin perjudicar a otra. Esto implica que, las situaciones sean eficientes, si al haber un cambio en esa situación, se beneficia a alguno, sin perjudicar a otro”. Es decir, al efectuar una asignación de recursos y se comparan los resultados, las partes involucradas están por lo menos en igualdad de condiciones en relación a las que previamente estaban y al menos una de ellas está en una mejor.

Matemáticamente esta relación establece una comparación entre dos puntos del espacio vectorial, verificando una dominancia o mejor posición con respecto no solo a una, sino a todas las funciones objetivo en el espacio vectorial de los criterios. Decimos que un punto $\vec{x}^* \in \Omega$ es un punto óptimo de Pareto si para toda $\vec{x} \in \Omega$ e $I = \{1, 2, \dots, k\}$ ya sea,

$$\forall_{i \in I} (f_i(\vec{x}) = f_i(\vec{x}^*)) \tag{3.6}$$

O hay al menos una $i \in I$ tal que

$$f_i(\vec{x}) > f_i(\vec{x}^*) \quad (3.7)$$

Bajo este concepto los vectores solución son clasificados como dominados o no dominados; un vector $\vec{u} = (u_1, \dots, u_k)$ es dominado por otro $\vec{v} = (v_1, \dots, v_k)$ (denotado por $\vec{u} \preceq \vec{v}$) si y solo si u es parcialmente menor a v . Esta dominancia puede ser fuerte, en el caso de que el vector dominante sea mejor evaluado en todos los objetivos; o débil, al obtener el mismo valor en por lo menos un objetivo. Ver Figura 20. Para un problema multi-objetivo dado $\vec{f}(x)$, el conjunto de óptimos de Pareto (P^*) se define en (130) y (132) como:

$$P^* := \{x \in \Omega \mid \neg \exists x' \in \Omega \vec{f}(x') \preceq \vec{f}(x)\} \quad (3.8)$$

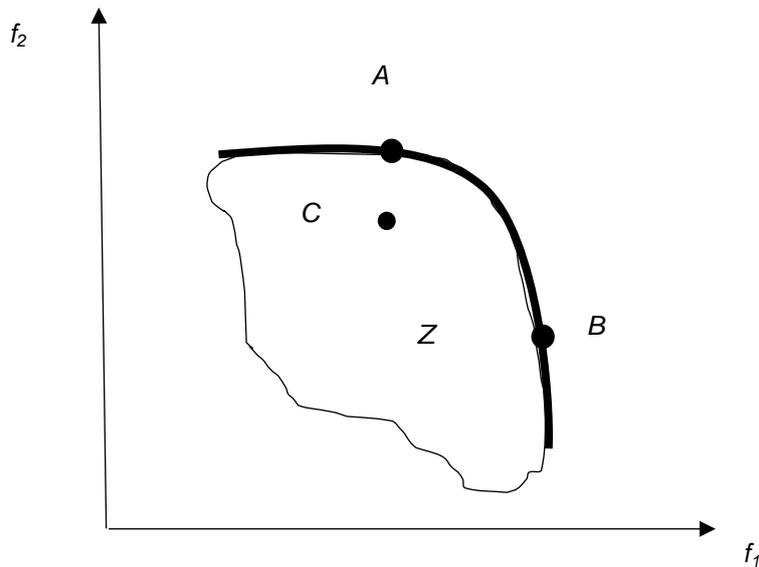


Figura 20: Frontera de Pareto

Al conjunto de puntos no dominados se le conoce como el frente de Pareto, ya que siempre se encuentran en la frontera del área de solución factible, por lo que en un desplazamiento por esta frontera siempre habrá una mejora respecto a uno de los criterios mientras se deteriora otro. En Figura 20 se puede observar como los puntos A y B son parte del frente de Pareto mientras que el punto C no lo es. El frente de Pareto (PF^*) se define como:

$$PF^* := \{\vec{u} = \vec{f} = (f_1(x), \dots, f_k(x)) \mid x \in P^*\} \quad (3.9)$$

3.5.4 Métodos de solución de problemas Multi-Objetivo.

3.5.4.1 Métodos de optimización no-lineales.

Al generar en una primera instancia un modelo no-lineal, es una alternativa obvia a considerar. Una de las características que hace que los problemas de optimización no lineal sean mucho más difíciles de resolver que los problemas lineales, es que la solución óptima no se encuentra en un punto extremo de la región de factibilidad.

La gran desventaja de los métodos de optimización no lineales, es que, generalmente encuentran un óptimo relativo o local, más no el óptimo local o absoluto, además se presentan de muchas formas distintas y no se dispone de un algoritmo que resuelva todos estos tipos especiales de problemas (133). En su lugar se han desarrollado algoritmos para algunas clases (tipos especiales) de problemas de programación no lineal. Adicionalmente, los métodos de optimización no-lineales suelen ser poco prácticos para solucionar algunos problemas multi_objetivo y/o discretos..

3.5.4.2 Métodos basados en la preferencia

Este tipo de métodos determinan la solución que mejor satisface la preferencia de quien toma las decisiones. Reduce el tiempo y el número de alternativas, pero sufren de subjetividad y falta de información. Para que los métodos basados en preferencia sean eficaces se deben conocer de forma precisa los requerimientos del cliente o dueño del sistema, por lo que pueden facilitar la solución de un problema particular pero de poca utilidad si no se cuenta con la información necesaria.

Uno de los métodos basados en preferencia más comunes es el de programación de metas (*Goal Programming*). Se basa en asignar una meta específica a alcanzar para cada una de los objetivos y generar una sola función objetivo donde se minimice la desviación total de los valores obtenidos con respecto a las metas definidas (134) y (135).

3.5.4.3 Métodos generadores

Estos métodos se basan en la generación formal de la frontera de Pareto. Las denominadas técnicas generadoras no incorporan información a priori en el proceso y, por tanto, es el analista el que plantea y resuelve el problema. Dentro de estas técnicas existen dos métodos de resolución, el Método de la ponderación y el Método de la restricción. Ambos métodos intentarán determinar las soluciones eficientes asociadas al problema y, dependiendo de los casos, podremos asegurar la eficiencia y la eficiencia propia.

3.5.4.4 Método de ponderación

El método de las ponderaciones consiste en construir un problema mono-objetivo asociado al problema original (también llamadas funciones agregativas), "mediante una escalarización de la función vectorial objetivo, de forma que a cada función Z_i se le asocia un valor λ_i , dicho valor tendrá una triple misión dentro del proceso, la primera es simplemente de cálculo u operativa, la segunda sería la información sobre la importancia relativa que se

le asigna al correspondiente objetivo Z_i , y por último, la eliminación de unidades” (136).

3.5.4.5 Método de restricciones

El método de la restricción es el más antiguo dentro de la programación multi-objetivo, pero es a su vez uno de los más pobres en cuanto a resultados importantes por la propia construcción y naturaleza de su establecimiento (136). Esto se debe a que el método se basa en construir un modelo mono-objetivo a partir del modelo original, donde el resto de las funciones objetivo se transforman en restricciones para hacer la función de cotas en el espacio de solución factible; lo cual genera la dificultad de contar con un espacio de búsqueda prácticamente infinito.

En este método la clave es la asignación de cotas, con lo que el conjunto de oportunidades cambia, y esto implica un problema importante: unas cotas grandes o demasiado optimistas implican que no se modifique el conjunto y unas cotas pequeñas o pesimistas podrían generar un conjunto vacío. La terminología utilizada, en términos de grande y pequeño es relativa a la naturaleza del problema que se esté estudiando.

Esto sucede para los problemas continuos, sin embargo, los problemas discretos con pocos objetivos cuentan con un espacio de cotas posibles finito y, en algunos casos, bien delimitado (137).

3.6 CONCLUSIONES

En este capítulo se ha buscado contextualizar el problema, presentando un estado del arte general sobre los modelos, métodos y algoritmos de resolución que se han utilizado en investigaciones en el problema de secuenciación en sistemas de circuito cerrado. Dado que las investigaciones sobre el problema con las características completas que se presentan en esta tesis no se han estudiado ampliamente, se ha establecido el estado del arte con base en investigaciones que aborden algunos de los aspectos en forma independiente. Es importante mencionar de acuerdo a la descripción del problema que se señaló en el capítulo anterior que se trata de un problema que crece en complejidad con el incremento en las variables independientes. En este caso la mejor solución se encontraría a través de procedimientos *branch and bound* que utilizan métodos exactos y que implicarían un tiempo computacional demasiado alto. En el análisis de cada una de las variantes del problema se establecerán los procedimientos exactos utilizando una modelación matemática para casos sencillos. Para estas variantes del problema se establecerán heurísticas de resolución que proporcionen una solución factible en un tiempo significativamente inferior. Con esto se logra presentar una solución eficiente que presente una buena solución en un tiempo relativamente corto.

4 PROBLEMA CBBSSP SIN HUECOS

4.1 MODELADO MATEMATICO PARA EL PROBLEMA CBBSSP SIN HUECOS

El problema puede ser formulado como un modelo mixto-entero. En esta sección se presenta una formulación y una variante de la misma para las cuales se presenta a continuación la nomenclatura y dos opciones de modelado, así como limitantes al modelo

4.1.1 NOMENCLATURA

Para poder modelar adecuadamente el problema del CBBSSP-SH, presentar su código y algunas condicionantes se definen índices, parámetros y variables de decisión que corresponden al caso general del problema.

4.1.1.1 Índices:

- i : Número del tipo de producto que se asigna (un producto está definido por un combinación de geometría y color).
- j : Número de la posición que se asigna.
- g : Número de la geometría que se revisa.
- c : Número del color que se revisa

4.1.1.2 Parámetros:

- n : Número de productos (un producto está definido por un combinación de geometría y color).
- n_c : Número de colores.
- n_g : Número de geometrías.
- d_i : Demanda para el producto i , para $i = 1, \dots, n$
- c_i : Color del producto i , para $i = 1, \dots, n$
- g_i : Geometría del producto i para $i = 1, \dots, n$
- h : Coste de setup horizontal (e.g., coste de setup cuando cambia el color)
- v : Coste de setup vertical (e.g., coste de setup debido a cambio de geometría)
- N : Número de posiciones, donde
$$N = \sum_{i=1}^n d_i$$
- PPL : Elementos por circuito o bucle
- LB_c : Límite inferior en el número de cambios por color, valor por defecto 0
- LB_g : Límite inferior en el número de cambios por geometría, valor por defecto 0
- LB : Límite inferior en el valor e la función objetivo

UB_c : Límite superior en el número de cambios por color, valor por defecto 0

UB_g : Límite superior en el número de cambios por geometría, valor por defecto 0

UB : Límite superior en el valor de la función objetivo

4.1.1.3 Variables de Decisión:

Se consideran los índices y variables presentados para definir adicionalmente el siguiente grupo de variables binarias de decisión, las cuáles permiten obtener información sobre la asignación de un producto específico a una posición determinada, si existe un cambio de color entre dos posiciones secuenciales o si existe un cambio de modelo o geometría en la misma posición al ejecutar un ciclo subsecuente en el proceso. Las variables se indican a continuación:

$x_{ij} = \begin{cases} 1 \\ 0 \end{cases}$ 1 si el producto i es colocado en la posición j , 0 en cualquier otro caso.

$y_j = \begin{cases} 1 \\ 0 \end{cases}$ 1 si existe un cambio de color entre las posiciones $j - 1$ y j , 0 en cualquier otro caso.

$z_j = \begin{cases} 1 \\ 0 \end{cases}$ 1 si existe un cambio de jig entre las posiciones $j - PPL$ y j , 0 en cualquier otro caso.

Tomando en consideración los índices y parámetros indicados, así como las variables de decisión se procede a presentar los modelos matemáticos en dos variantes para el problema básico CBBSSP-SH.

4.1.2 MODELADO DEL PROBLEMA CBBSSP-SH

El problema CBBSSP-SH puede ser referenciado fácilmente en los sistemas de pintura de partes automotrices, sin embargo en la modelación se busca la generalidad que tenga aplicabilidad en sistemas similares donde existe un flujo cíclico donde se tienen al menos dos tipos de tiempos de preparación de carga y descarga de las partes. El modelo con dos variantes se formula como sigue:

4.1.2.1 Función Objetivo del CBBSSP-SH.

En los sistemas cíclicos de pintura se presentan los tipos de tiempos de preparación mencionados en la sección anterior. En esta situación el objetivo del sistema es minimizar el coste total de los mismos, es decir minimizar los costes relacionados con el número de cambios de color y los que dependen del número de cambios de bastidor o jig.

Sin perder generalidad, el modelo asume que cualquier cambio en color genera un coste h (Coste de tiempo de preparación horizontal) y cualquier cambio en la geometría del modelo genera el coste v (Coste de tiempo de preparación vertical).

De acuerdo a lo indicado la expresión matemática de la función objetivo para el CBBSSP-SH es:

$$\text{Minimizar} \quad h \sum_{j>1} y_j + v \sum_{j>PPL} z_j \quad (0.1)$$

4.1.2.2 Restricciones básicas del problema CBBSSP-SH Opción 1.

Se debe optimizar la función objetivo cumpliendo una serie de restricciones

Restricciones de satisfacción de la demanda.

La demanda para cada tipo de producto i debe ser completada si se revisan todas las posiciones existentes en el sistema. Esto se asegura mediante el siguiente grupo de restricciones:

$$h \sum_j X_{ij} = d_i \quad i = 1, \dots, n \quad (0.2)$$

Restricciones de asignación de un solo producto en cada posición disponible.

Cada posición j disponible en el sistema debe tener asignado un solo producto, lo cual se restringe asegurando que la sumatoria de todos los tipos de producto i en una posición específica j (variable de decisión X_{ij}) sea igual a 1.

$$\sum_i X_{ij} = 1 \quad j = 1, \dots, N \quad (0.3)$$

Restricciones para contabilizar los cambios de color Opción 1.

Mediante estas restricciones se asegura que la variable de decisión Y_j tenga el valor de 1 cuando ocurre un cambio de color entre la posición j y la posición $j - 1$. Se hace la evaluación a partir de la posición $j = 2$, para poder comparar con la posición $j = 1$. Se revisan todas las posiciones hasta N , el número total de posiciones.

$$n_c y_j \geq \sum_i c_i x_{ij} - \sum_i c_i x_{i,j-1} \quad j = 2, \dots, N \quad (0.4)$$

$$n_c y_j \geq \sum_i c_i x_{i,j-1} - \sum_i c_i x_{ij} \quad j = 2, \dots, N \quad (0.5)$$

Estas dos restricciones operan en conjunto. La restricción (0.4) funciona de la siguiente manera:

- 1) Se evalúa para todos los productos i si han sido asignados en la posición j . con lo que el valor de X_{ij} en dicha variable de decisión será 1 en alguno de los casos de i .
- 2) Multiplica el valor para cada caso de X_{ij} por el valor de C_i correspondiente. En un solo caso, aquel en el que el valor de X_{ij} es 1 se obtendrá un valor positivo.
- 3) Se hace la sumatoria de todos los casos i en la posición j
- 4) Se evalúa para todos los productos i si han sido asignados en la posición $j-1$. con lo que el valor de $X_{i,j-1}$ en dicha variable de decisión será 1 en alguno de los casos de i .
- 5) Multiplica el valor para cada caso de $X_{i,j-1}$ por el valor de C_i correspondiente. En un solo caso, aquel en el que el valor de $X_{i,j-1}$ es 1 se obtendrá un valor positivo.
- 6) Se hace la sumatoria de todos los casos i en la posición $j-1$
- 7) Se hace la diferencia entre la sumatoria de los casos en j menos la sumatoria de los casos en $j - 1$
- 8) Se divide el resultado de la diferencia entre n_c con lo que se obtiene un valor igual a cero o distinto a cero.
- 9) Se restringe el valor de Y_j con una desigualdad que limita a dicha variable de decisión a que sea mayor o igual al valor obtenido en el punto anterior. Es decir Y_j será mayor o igual que cero o que el valor obtenido.

La restricción (0.5) funciona como complemento de la restricción (0.4). El cálculo del valor es en la práctica igual con los términos de la diferencia invertidos. Es decir, la diferencia se calcula restando a la sumatoria de los casos en $j - 1$ el valor de la sumatoria de los casos en j .

Como ambas restricciones se deben de cumplir en el modelo, el resultado en la variable binaria Y_j solo podrá ser con valor 0 en el caso de que el color sea el mismo en las posiciones j y $j - 1$ y en el caso de que el color sea distinto entre las dos posiciones mencionadas el valor será 1.

A continuación se indican los pasos para la restricción (0.5) :

- 1) Se evalúa para todos los productos i si han sido asignados en la posición $j - 1$. con lo que el valor de X_{ij-1} en dicha variable de decisión será 1 en alguno de los casos de i .
- 2) Multiplica el valor para cada caso de X_{ij-1} por el valor de C_i correspondiente. En un solo caso, aquel en el que el valor de X_{ij-1} es 1 se obtendrá un valor positivo.
- 3) Se hace la sumatoria de todos los casos i en la posición $j - 1$
- 4) Se evalúa para todos los productos i si han sido asignados en la posición j . con lo que el valor de X_{ij} en dicha variable de decisión será 1 en alguno de los casos de i .
- 5) Multiplica el valor para cada caso de X_{ij} por el valor de C_i correspondiente. En un solo caso, aquel en el que el valor de X_{ij} es 1 se obtendrá un valor positivo.
- 6) Se hace la sumatoria de todos los casos i en la posición j
- 7) Se hace la diferencia entre la sumatoria de los casos en $j - 1$ menos la sumatoria de los casos en j
- 8) Se divide el resultado de la diferencia entre n_c con lo que se obtiene un valor igual a cero o distinto a cero.
- 9) Se restringe el valor de Y_j con una desigualdad que limita a dicha variable de decisión a que sea mayor o igual al valor obtenido en el punto anterior. Es decir Y_j será mayor o igual que cero o que el valor obtenido.

Restricciones para contabilizar los cambio de modelo Opción 1.

Mediante estas restricciones se asegura que la variable de decisión Z_j tenga el valor de 1 cuando ocurre un cambio de modelo entre la posición j y la posición $j - PPL$, es decir entre la posición actual con el producto asignado y la misma posición pero con el producto asignado en el ciclo inmediato anterior. Se hace la evaluación a partir de la posición $j = PPL + 1$, para poder comparar con la posición $j = 1$. Se revisan todas las posiciones hasta N , el número total de posiciones.

$$n_g z_j \geq \sum_i g_i x_{ij} - \sum_i g_i x_{i,j-PPL} \quad j = PPL + 1, \dots, N \quad (0.6)$$

$$n_g z_j \geq \sum_i g_i x_{i,j-PPL} - \sum_i g_i x_{ij} \quad j = PPL + 1, \dots, N \quad (0.7)$$

De manera similar a las restricciones para contabilizar los cambios de color, las restricciones por cambio de modelo operan en conjunto. La restricción (0.6) funciona de la siguiente manera:

- 1) Se evalúa para todos los productos i si han sido asignados en la posición j . con lo que el valor de X_{ij} en dicha variable de decisión será 1 en alguno de los casos de i .
- 2) Multiplica el valor para cada caso de X_{ij} por el valor de g_i correspondiente. En un solo caso, aquel en el que el valor de X_{ij} es 1 se obtendrá un valor positivo.
- 3) Se hace la sumatoria de todos los casos i en la posición j
- 4) Se evalúa para todos los productos i si han sido asignados en la posición $j - PPL$. con lo que el valor de X_{ij-1} en dicha variable de decisión será 1 en alguno de los casos de i .
- 5) Multiplica el valor para cada caso de X_{ij-1} por el valor de g_i correspondiente. En un solo caso, aquel en el que el valor de X_{ij-1} es 1 se obtendrá un valor positivo.
- 6) Se hace la sumatoria de todos los casos i en la posición $j - PPL$
- 7) Se hace la diferencia entre la sumatoria de los casos en j menos la sumatoria de los casos en $j - PPL$
- 8) Se divide el resultado de la diferencia entre n_g con lo que se obtiene un valor igual a cero o distinto a cero.
- 9) Se restringe el valor de Z_j con una desigualdad que limita a dicha variable de decisión a que sea mayor o igual al valor obtenido en el punto anterior. Es decir Z_j será mayor o igual que cero o que el valor obtenido.

La restricción (0.6) funciona como complemento de la restricción (0.7). El procedimiento de cálculo del valor es en la práctica igual solo que con los términos de la diferencia invertidos. Es decir, la diferencia se calcula restando a la sumatoria de los casos en $j - PPL$ el valor de la sumatoria de los casos en j .

Como ambas restricciones se deben de cumplir en el modelo matemático, el resultado en la variable binaria Z_j solo podrá ser con valor 0 en el caso de que el modelo de producto sea el mismo en las posiciones j y $j - PPL$ y en el caso de que el modelo sea distinto entre las dos posiciones mencionadas el valor será 1.

A continuación se indican los pasos para la restricción (0.7) :

- 1) Se evalúa para todos los productos i si han sido asignados en la posición $j - PPL$. con lo que el valor de X_{ij-1} en dicha variable de decisión será 1 en alguno de los casos de i .
- 2) Multiplica el valor para cada caso de X_{ij-1} por el valor de g_i correspondiente. En un solo caso, aquel en el que el valor de X_{ij-1} es 1 se obtendrá un valor positivo.
- 3) Se hace la sumatoria de todos los casos i en la posición $j - PPL$
- 4) Se evalúa para todos los productos i si han sido asignados en la posición j . con lo que el valor de X_{ij} en dicha variable de decisión será 1 en alguno de los casos de i .

- 5) Multiplica el valor para cada caso de X_{ij} por el valor de g_i correspondiente. En un solo caso, aquel en el que el valor de X_{ij} es 1 se obtendrá un valor positivo.
- 6) Se hace la sumatoria de todos los casos i en la posición j
- 7) Se hace la diferencia entre la sumatoria de los casos en j - PPL menos la sumatoria de los casos en j
- 8) Se divide el resultado de la diferencia entre n_g con lo que se obtiene un valor igual a cero o distinto a cero.
- 9) Se restringe el valor de Z_j con una desigualdad que limita a dicha variable de decisión a que sea mayor o igual al valor obtenido en el punto anterior. Es decir Z_j será mayor o igual que cero o que el valor obtenido.

Restricciones de integralidad

$$x_{ij}, y_j, z_j \in \{0,1\} \quad i = 1, \dots, n \quad , \quad j = 1, \dots, N \quad (0.8)$$

Este grupo de restricciones aseguran que el valor de las variables binarias: X_{ij} , Y_j , Z_j para todos los tipos de producto y para todas las posiciones sea 0 ó 1.

4.1.2.3 Restricciones básicas del problema CBBSSP-SH Opción 2.

Restricción de satisfacción de la demanda.

La demanda para cada tipo de producto i debe ser completada si se revisan todas las posiciones existentes en el sistema. Esto se asegura mediante el siguiente grupo de restricciones:

$$h \sum_j X_{ij} = d_i \quad i = 1, \dots, n \quad (4.2)$$

Restricción de asignación de un solo producto en cada posición disponible

Cada posición j disponible en el sistema debe tener asignado un solo producto, lo cual se restringe asegurando que la sumatoria de todos los tipos de producto i en una posición específica j (variable de decisión X_{ij}) sea igual a 1.

$$\sum_i X_{ij} = 1 \quad j = 1, \dots, N \quad (4.3)$$

Restricciones para contabilizar los cambios de color Opción 2.

Mediante estas restricciones se asegura que la variable de decisión Y_j tenga el valor de 1 cuando ocurre un cambio de color entre la posición j y la

posición $j - 1$. Se hace la evaluación a partir de la posición $j = 2$, para poder comparar con la posición $j = 1$. Se revisan todas las posiciones hasta N , el número total de posiciones.

$$Y_j \geq X_{ij} + \sum_{\substack{i \neq j \\ c_i \neq c_j}} X_{i,j-1} - 1 \quad i = 1, \dots, n, \quad j = 2, \dots, N \quad (0.9)$$

La restricción (4.9) evalúa en su primer término cual es el producto que es asignado a la posición j , y en el segundo término el que es asignado a la posición $j-1$. No considera en el cálculo el producto idéntico al del primer término o el que tiene el mismo color que el de dicho término, solo el que tiene un color distinto. Los pasos para esta restricción funcionan de la siguiente manera:

- 1) En un primer término se evalúa para todos los productos i si han sido asignados en la posición j , se sabe que el valor de X_{ij} en dicha variable de decisión será 1 en alguno de los casos de i .
- 2) En un segundo término se evalúa cada caso de i en la posición $j - 1$. sujeto a dos restricciones: 1) El producto i que se evalúa debe ser diferente al que se evalúa en el punto anterior. 2) Solo se consideran los productos que tienen un color distinto C_i al que tiene el producto i en la posición j (C_j) del primer término.
- 3) Se hace una sumatoria de todos los casos i que cumplan las restricciones señaladas en el punto anterior.
- 4) Se suman los términos primero y segundo y se les resta el valor 1.
- 5) Se restringe el valor de Y_j con una desigualdad que limita a dicha variable de decisión a que sea mayor o igual al valor obtenido en el punto anterior. Es decir Y_j será mayor o igual que cero o que el valor obtenido.
- 6) Como se indica en el primer punto, se evalúan todos los casos de producto i . Desde $i=1$ hasta n .
- 7) EL valor en Y_j debe cumplir las desigualdades para cada i . y solo puede ser cero o uno.
- 8) Se repiten los puntos anteriores para cada posición j , desde $j=2$ hasta N .

Restricciones para contabilizar los cambio de modelo Opción 2.

A través de este conjunto de restricciones se asegura que la variable de decisión Z_j tenga el valor de 1 cuando ocurre un cambio de color entre la posición j y la posición $j - PPL$. Se hace la evaluación a partir de la posición $j = PPL+1$, para poder comparar con la posición $j = 1$. Se revisan todas las posiciones hasta N , el número total de posiciones

$$Z_j \geq X_{ij} + \sum_{\substack{i \neq j \\ g_i \neq g_j}} X_{i,j-PPL} - 1 \quad i = 1, \dots, n, j = PPL+1, \dots, N \quad (0.10)$$

La restricción (4.10) funciona de la siguiente manera:

- 1) En un primer término se evalúa para todos los productos i si han sido asignados en la posición j , se sabe que el valor de X_{ij} en dicha variable de decisión será 1 en alguno de los casos de i .
- 2) En un segundo término se evalúa cada caso de i en la posición $j - PPL$. sujeto a dos restricciones: 1) El producto i que se evalúa debe ser diferente al que se evalúa en el punto anterior. 2) Solo se consideran los productos que tienen un modelo g_i distinto al que tiene el producto i en la posición $j (g_j)$ del primer término.
- 3) Se hace una sumatoria de todos los casos i que cumplan las restricciones señaladas en el punto anterior.
- 4) Se suman los términos primero y segundo y se les resta el valor 1.
- 5) Se restringe el valor de Z_j con una desigualdad que limita a dicha variable de decisión a que sea mayor o igual al valor obtenido en el punto anterior. Es decir Z_j será mayor o igual que cero o que el valor obtenido.
- 6) Como se indica en el primer punto, se evalúan todos los casos de producto i . Desde $i=1$ hasta n .
- 7) EL valor en Z_j debe cumplir las desigualdades para cada i . y solo puede ser cero o uno.
- 8) Se repiten los puntos anteriores para cada posición j , desde $j=PPL+1$ hasta N .

Restricciones de integralidad Opción 2

$$x_{ij} \in \{0,1\}, Y_j, Z_j \geq 0 \quad i = 1, \dots, n, j = 1, \dots, N \quad (0.11)$$

Este grupo de restricciones aseguran que el valor de las variables binarias: X_{ij} , para todos los tipos de producto y para todas las posiciones sea cero o uno. Adicionalmente para las variables binarias Y_j y Z_j se asegura que sea mayor o igual a cero.

4.1.2.4 Modelo general del problema CBBSSP-SH

Modelo general del problema CBBSSP Opción 1

Minimizar
$$h \sum_{j>1} y_j + v \sum_{j>PPL} z_j \quad (4.1)$$

Sujeta a:

$$h \sum_j X_{ij} = d_i \quad i = 1, \dots, n \quad (4.2)$$

$$\sum_i X_{ij} = 1 \quad j = 1, \dots, N \quad (4.3)$$

$$n_c y_j \geq \sum_i c_i x_{ij} - \sum_i c_i x_{i,j-1} \quad j = 2, \dots, N \quad (4.4)$$

$$n_c y_j \geq \sum_i c_i x_{i,j-1} - \sum_i c_i x_{ij} \quad j = 2, \dots, N \quad (4.5)$$

$$n_g z_j \geq \sum_i g_i x_{ij} - \sum_i g_i x_{i,j-PPL} \quad j = PPL + 1, \dots, N \quad (4.6)$$

$$n_g z_j \geq \sum_i g_i x_{i,j-PPL} - \sum_i g_i x_{ij} \quad j = PPL + 1, \dots, N \quad (4.7)$$

$$x_{ij}, y_j, z_j \in \{0, 1\} \quad i = 1, \dots, n, \quad j = 1, \dots, N \quad (4.8)$$

Modelo general del problema CBBSSP-SH Opción 2

Minimizar
$$h \sum_{j>1} y_j + v \sum_{j>PPL} z_j \quad (4.1)$$

Sujeta a:

$$h \sum_j X_{ij} = d_i \quad i = 1, \dots, n \quad (4.2)$$

$$\sum_i X_{ij} = 1 \quad j = 1, \dots, N \quad (4.3)$$

$$Y_j \geq X_{ij} + \sum_{\substack{i \neq j \\ c_i \neq c_j}} X_{i,j-1} - 1 \quad i = 1, \dots, n, \quad j = 2, \dots, N \quad (4.9)$$

$$Z_j \geq X_{ij} + \sum_{\substack{i \neq j \\ g_i \neq g_j}} X_{i,j-PPL} - 1 \quad i = 1, \dots, n, \quad j = 2, \dots, N \quad (4.10)$$

$$x_{ij} \in \{0, 1\}, Y_j, Z_j \geq 0 \quad i = 1, \dots, n, \quad j = 1, \dots, N \quad (4.11)$$

4.1.2.5 Límites al modelo y análisis del entorno en el Problema CBBSSP-SH.

Antes de revisar los posibles métodos de solución se analiza la estructura del problema con el objetivo de explorar nuevas ideas y determinar

los límites al número de cambios por tiempo de preparación por los dos tipos (horizontal y vertical) mencionados en la definición del problema.

Si se considera que el costo del tiempo de preparación es constante y no depende del cambio específico de colores o de geometrías involucradas en cada caso, entonces los límites inferiores en el número de cambios por tiempo de preparación para cada tipo de cambio (LB_c, LB_g) pueden ser utilizados para calcular un límite inferior en el costo total (LB), referido como la sumatoria de los costos obtenidos por cada uno de los dos tipo de cambio multiplicado por sendos límites inferiores:

$$LB = hLB_c + vLB_g \quad (0.12)$$

Para los problemas donde se requiera considerar necesariamente costos variables de tiempo de preparación, el límite inferior se puede calcular utilizando el mínimo coste horizontal y el mínimo vertical.

Límite inferior en el número de cambios por tiempo de preparación horizontal.

Si los tiempos de preparación se ignoran, se obtiene un límite inferior por color dado por la siguiente ecuación:

$$LB_c = [n_c] - 1 \quad (0.13)$$

En este caso, en la ausencia de tiempos de preparación vertical, la forma óptima de organizar los elementos se obtiene al agrupar todos los productos del mismo color en cada caso y asignarlos secuencialmente al proceso, siempre y cuando todas las partes con el mismo color sean procesadas antes de continuar con otro color.

Límite inferior en el número de cambios por tiempo de preparación vertical

Si se deja de lado el costo de tiempo de preparación horizontal por cambio de color se tiene solamente el costo por tiempo de preparación vertical por cambio de modelo. Una forma de conceptualizar la solución puede ser al considerar una matriz de dos dimensiones donde el número de columnas equivale al número de posiciones por ciclo (PPL) y el número de filas es igual a N/PPL .

La demanda relacionada con cada geometría se puede calcular haciendo la sumatoria de la demanda para cada producto i restringida por tipo de geometría. Lo cuál se puede conceptualizar agregando un subíndice al parámetro d_i de tal manera que se define un nuevo parámetro: $d_{(g_i)}$ como demanda para la geometría el producto " g " (la sumatoria e todos los productos i que tengan la geometría g . La demanda asociada con cada geometría g es dada por la ecuación:

$$d_{(g^*)} = \sum_i d_{(g_i)} \quad g = 1, \dots, n_g \quad (0.14)$$

Se puede agregar un parámetro nuevo: m_g el cual consiste en el cálculo del residuo resultante de dividir la demanda por geometría: $d_{(g_i)}$ entre el número de filas dado por N/PPL .

$$m_g = \text{mod}\left(\frac{d_{g_i}}{N/PPL}\right) \quad g = 1, \dots, n_g \quad (0.15)$$

Se puede demostrar que en el caso de que $m_g = 0$ para toda g entonces el $LB_g = 0$. Esto significa que cada columna incluye solamente una geometría y el número de veces que ocurre un tiempo de preparación es cero. Sin embargo al menos en una ocasión ocurre un tiempo de preparación cuando $m_g \neq 0$ para al menos una $g \in n_g$. Esto significa que cuando los costos de preparación horizontal se ignoran, el problema de minimizar los costos de preparación vertical se puede reducir en otro más simple de la siguiente manera:

Para cada geometría $g \in n_g$, $\left\lceil \frac{d_{g_i}}{N/PPL} \right\rceil$ indica el número de columnas que son asignadas totalmente a cada geometría sin incurrir en cambios por tiempos de preparación verticales. El problema reducido consiste entonces en asignar m_g elementos para cada $g \in n_g$ en una matriz con el mismo número de renglones como en el problema original (N/PPL) y un número reducido de columnas. El número reducido de columnas, o número reducido de espacios por ciclo (RPPL) es dado por:

$$RPPL = PPL - \sum_g \left\lceil \frac{d_{g_i}}{N/PPL} \right\rceil = \frac{\sum_g m_g}{N/PPL} \quad (0.16)$$

Dado que en este caso no se está considerando el coste de tiempos de preparación por cambios horizontales, las variables y restricciones relacionadas con el cambio de color pueden ser omitidas de la formulación (Restricciones 4.4 y 4.5 para la Opción 1 y 4.9 en la Opción 2) De esta manera ignorando los costes horizontales y considerando solamente los costes por cambio de geometría el problema de optimización se reduce tanto en escala como en complejidad. El número mínimo de cambios por geometría se obtiene resolviendo este modelo reducido y es un límite inferior realista en el número de cambios por geometría (LB_g). De acuerdo al n_g y a la ecuación (4.16), se sabe que $LB_g \leq n_g - 1$

Límite superior en el número de cambios por tiempo de preparación horizontal.

El límite superior en el número de cambios por tiempo de preparación horizontal ocurre en el peor de los escenarios cuando entre cada posición j ocurre un cambio lo cual incrementa el valor del número de cambios en cada ocasión. De acuerdo a esto el máximo dado por el número de cambios de color se obtiene con la ecuación:

$$h \sum_{j>1} y_j = N - 1 \quad (0.17)$$

En el caso de que el valor del coste por cambio sea $h=1$ el coste por cambios por tiempo de preparación horizontal es solo el número de cambios máximo igual al número de posiciones menos uno.

Límite superior en el número de cambios por tiempo de preparación vertical.

El límite superior en el número de cambios por tiempo de preparación vertical se tiene también en el peor de los casos cuando entre cada posición j y $j+1$ ocurre un cambio vertical lo cual hace necesario hacer un cambio de j en cada ocasión. De acuerdo a esto el máximo dado por el número de cambios de geometría se obtiene con la ecuación:

$$v \sum_{j>1} z_j = N - PPL \quad (0.18)$$

En el caso de que el valor del coste por cambio sea $v=1$ el coste por cambios por tiempo de preparación vertical es solo el número de cambios máximo igual al número de posiciones menos el número de elementos por circuito o bucle (PPL).

Espacio de solución del modelo general

Es posible obtener un límite inferior en el costo total utilizando para el cálculo los valores obtenidos de número de tiempos de preparación por cambio de color o cambio de modelo por separado, sin embargo ejemplos simples pueden demostrar que un método de solución que se enfoque en cada tipo de tiempo de preparación en forma independiente puede resultar en soluciones que no son solo subóptimas sino que son significativamente inferiores que aquellas que se pueden encontrar al considerar ambos tiempos de preparación en conjunto. Esto significa que un procedimiento que minimice el número de cambios de un tipo de tiempo de preparación y que a continuación minimice el número de cambios del segundo tipo no necesariamente va a optimizar el objetivo, incluso si se consideran solamente aquellas soluciones que no incrementen el número de cambios del primer tipo al intentar minimizar los del segundo.

Una manera simple de representar una solución al problema consiste en definir un vector Q de tamaño N (Número de posiciones), de tal manera que $Q(i)$ es el índice del producto en la posición i . La demanda del producto i es dado por el parámetro d_i . Si al revisar una instancia específica los valores de d_i son de 1 ($d_i = 1$ para toda i , entonces $N = [n]$), es decir, el número de posiciones del problema es igual al número de productos, y el espacio de solución consiste en $N!$ soluciones. La fórmula en general para determinar el espacio de soluciones se representa por la siguiente multiplicatoria:

$$\prod_{i=1}^{i=[n]} C_{d_i}^{N - \sum_{i'=0}^{i'-i-1} d_{i'}} \quad (0.19)$$

Considerando los costes de cambio horizontal y cambio vertical como unitarios, esto es ($h=v=1$) entonces el coste total asociado con una solución dada es el mismo que se obtiene como el número total de cambios necesarios al establecer una solución.

El límite superior al número de cambios total se obtiene al sumar los límites superiores de cada tipo de tiempo de preparación y se indica en la siguiente ecuación:

$$h \sum_{j>1} y_j + v \sum_{j>PPL} z_j = 2N - PPL - 1 \quad (0.20)$$

4.1.3 CONCLUSION DEL MODELADO

El primer modelo para el CBBSSP-SH (Opción1) contiene $n(N+2)$ variables binarias y $n+5N-2(PPL+1)$ restricciones. En la Opción 2 para el CBBSSP se observa un modelo ligeramente más compacto en relación al número de variables binarias. En este caso las variables binarias se reducen a nN con las restantes $2n$ variables declaradas como continuas. Sin embargo el número total de restricciones se modifica a un total de $n(2N - PPL - 1)$.

El proceso branch and bound debe considerar simetrías. Es decir, soluciones que en principio son equivalentes y que se obtienen al rotar la asignación de una pieza del tipo de producto i en una posición disponible j . Cada producto i está relacionado con una demanda d_i al cambiar en la secuenciación una pieza por otra idéntica se generaliza la simetría.

4.2 PROCEDMIENTOS EXACTOS PARA LA RESOLUCION DEL PROBLEMA CBBSSP SIN HUECOS

En el apartado anterior se desarrollaron dos opciones de modelo de programación matemática de acuerdo a la problemática de las líneas de pintura en la industria automotriz descrita en el capítulo 3, considerando el Problema Sin Huecos. Fue necesario definir las hipótesis básicas de trabajo establecer la función objetivo y justificar cada una de las restricciones presentadas. Con lo anterior se concluye que el objetivo del Problema Sin Huecos, uno de los dos estudiados en esta Tesis, consiste en definir la secuenciación de tareas en la industria automotriz con lotes cíclicos satisfaciendo el criterio de minimización bi-objetivo de los tiempos de preparación dependientes e independientes de la secuencia. Es decir minimizar el número de cambios de modelo y número de cambios de color. Se presentaron dos opciones de modelado matemático para la resolución del problema. Dado que la segunda opción es más compacta en el número de restricciones se ha seleccionado para hacer el código de programación en GLPK mediante el software GUSEK y utilizar para resolverlo

el software GUROBI que tiene un *solver* más eficaz en tiempo y capacidad que el que utiliza GUSEK.

En este capítulo se desarrolla la implementación de los modelos matemáticos utilizando herramientas de solución exacta.

4.2.1 METODOLOGIA DE LOS PROCEDIMIENTOS EXACTOS DEL PROBLEMA CBBSSP-SH.

En el siguiente apartado se hace la implementación de los modelos matemáticos presentados en el apartado anterior. Para llevarlo a cabo, se realiza primero una operacionalización de ambos modelos a través de un ejemplo básico que muestra el funcionamiento de la totalidad de las restricciones. En el apartado que sigue se desarrolla el modelo en Gusek utilizando el lenguaje que utiliza por defecto GLPK, el cual permite generar archivos con extensiones tipo .dat, .mod, .out, .mps, .lp. La ventaja de esto es poder utilizar como herramienta de resolución a distintos tipos de software como CPLEX, GUROBI o el mismo GUSEK. En este caso como se ha mencionado se utilizará GUROBI

4.2.2 IMPLEMENTACION DE LOS MODELOS PARA RESOLVER CON GUSEK-GUROBI

4.2.2.1 Operacionalización de un ejemplo básico.

Se presenta un ejemplo en el que se pueden observar las características generales del problema. Se muestra cada restricción desarrollada con los parámetros correspondientes para el caso particular del ejemplo básico y se muestra el resultado que se obtiene en cada una de ellas.

Ejemplo Básico:

Minimizar el número de cambios de modelo y el número de cambio de color en un sistema productivo encontrando la mejor secuencia de producción considerando los siguientes parámetros:

- n : Número de productos (un producto está definido por un combinación de geometría y color). → 3
- n_c : Número de colores. → 2
- n_g : Número de geometrías. → 2
- h : Coste de Setup horizontal → 1
- v : Coste de Setup vertical → 1

- c_i : Color del producto i , para $i = 1, \dots, n$. Colores disponibles: → 1,2
- g_i : Geometría del producto i para $i = 1, \dots, n$. Geometrías disponibles: → 1,2
- d_i : Demanda para el producto i , para $i = 1, \dots, n$

i	1	2	3
c_i	1	1	2
g_i	1	2	1
d_i	3	3	4

Tabla 5: Ejemplo implementado. Demanda por tipo de Producto

N : Número de posiciones, donde $N = \sum_{i=1}^n d_i \rightarrow 10$

PPL : Elementos por circuito o bucle $\rightarrow 5$

c_i	1	1	1	1	1
g_i	1	1	1	2	2
c_i	1	2	2	2	2
g_i	2	1	1	1	1

Tabla 6: Ejemplo implementado. Secuenciación en un sistema con 5 bloques

Modelo Opción 1

Función Objetivo:

$$\text{Minimizar } h \sum_{j>1} y_j + v \sum_{j>PPL} z_j$$

Minimizar

$$1*(y_1 + y_2 + y_3 + y_4 + y_5 + y_6 + y_7 + y_8 + y_9 + y_{10}) + 1*(z_6 + z_7 + z_8 + z_9 + z_{10})$$

Restricciones:

Restricciones (4.2):

$$x_{1,1} + x_{1,2} + x_{1,3} + x_{1,4} + x_{1,5} + x_{1,6} + x_{1,7} + x_{1,8} + x_{1,9} + x_{1,10} = 3$$

$$x_{2,1} + x_{2,2} + x_{2,3} + x_{2,4} + x_{2,5} + x_{2,6} + x_{2,7} + x_{2,8} + x_{2,9} + x_{2,10} = 3$$

$$x_{3,1} + x_{3,2} + x_{3,3} + x_{3,4} + x_{3,5} + x_{3,6} + x_{3,7} + x_{3,8} + x_{3,9} + x_{3,10} = 4$$

Restricciones (4.3):

$$x_{1,1} + x_{2,1} + x_{3,1} = 1$$

$$x_{1,2} + x_{2,2} + x_{3,2} = 1$$

$$x_{1,3} + x_{2,3} + x_{3,3} = 1$$

$$x_{1,4} + x_{2,4} + x_{3,4} = 1$$

$$x_{1,5} + x_{2,5} + x_{3,5} = 1$$

$$x_{1,6} + x_{2,6} + x_{3,6} = 1$$

$$x_{1,7} + x_{2,7} + x_{3,7} = 1$$

$$x_{1,8} + x_{2,8} + x_{3,8} = 1$$

$$x_{1,9} + x_{2,9} + x_{3,9} = 1$$

$$x_{1,10} + x_{2,10} + x_{3,10} = 1$$

Restricciones (4.4):

$$2y_2 \geq c_1x_{1,2} + c_2x_{2,2} + c_3x_{3,2} - (c_1x_{1,1} + c_2x_{2,1} + c_3x_{3,1})$$

$$2y_3 \geq c_1x_{1,3} + c_2x_{2,3} + c_3x_{3,3} - (c_1x_{1,2} + c_2x_{2,2} + c_3x_{3,2})$$

$$2y_4 \geq c_1x_{1,4} + c_2x_{2,4} + c_3x_{3,4} - (c_1x_{1,3} + c_2x_{2,3} + c_3x_{3,3})$$

$$2y_5 \geq c_1x_{1,5} + c_2x_{2,5} + c_3x_{3,5} - (c_1x_{1,4} + c_2x_{2,4} + c_3x_{3,4})$$

$$2y_6 \geq c_1x_{1,6} + c_2x_{2,6} + c_3x_{3,6} - (c_1x_{1,5} + c_2x_{2,5} + c_3x_{3,5})$$

$$2y_7 \geq c_1x_{1,7} + c_2x_{2,7} + c_3x_{3,7} - (c_1x_{1,6} + c_2x_{2,6} + c_3x_{3,6})$$

$$2y_8 \geq c_1x_{1,8} + c_2x_{2,8} + c_3x_{3,8} - (c_1x_{1,7} + c_2x_{2,7} + c_3x_{3,7})$$

$$2y_9 \geq c_1x_{1,9} + c_2x_{2,9} + c_3x_{3,9} - (c_1x_{1,8} + c_2x_{2,8} + c_3x_{3,8})$$

$$2y_{10} \geq c_1x_{1,10} + c_2x_{2,10} + c_3x_{3,10} - (c_1x_{1,9} + c_2x_{2,9} + c_3x_{3,9})$$

Restricciones (4.5):

$$2y_2 \geq c_1x_{1,1} + c_2x_{2,1} + c_3x_{3,1} - (c_1x_{1,2} + c_2x_{2,2} + c_3x_{3,2})$$

$$2y_3 \geq c_1x_{1,2} + c_2x_{2,2} + c_3x_{3,2} - (c_1x_{1,3} + c_2x_{2,3} + c_3x_{3,3})$$

$$2y_4 \geq c_1x_{1,3} + c_2x_{2,3} + c_3x_{3,3} - (c_1x_{1,4} + c_2x_{2,4} + c_3x_{3,4})$$

$$2y_5 \geq c_1x_{1,4} + c_2x_{2,4} + c_3x_{3,4} - (c_1x_{1,5} + c_2x_{2,5} + c_3x_{3,5})$$

$$2y_6 \geq c_1x_{1,5} + c_2x_{2,5} + c_3x_{3,5} - (c_1x_{1,6} + c_2x_{2,6} + c_3x_{3,6})$$

$$2y_7 \geq c_1x_{1,6} + c_2x_{2,6} + c_3x_{3,6} - (c_1x_{1,7} + c_2x_{2,7} + c_3x_{3,7})$$

$$2y_8 \geq c_1x_{1,7} + c_2x_{2,7} + c_3x_{3,7} - (c_1x_{1,8} + c_2x_{2,8} + c_3x_{3,8})$$

$$2y_9 \geq c_1x_{1,8} + c_2x_{2,8} + c_3x_{3,8} - (c_1x_{1,9} + c_2x_{2,9} + c_3x_{3,9})$$

$$2y_{10} \geq c_1x_{1,9} + c_2x_{2,9} + c_3x_{3,9} - (c_1x_{1,10} + c_2x_{2,10} + c_3x_{3,10})$$

Restricciones (4.6):

$$2z_6 \geq g_1x_{1,6} + g_2x_{2,6} + g_3x_{3,6} - (g_1x_{1,1} + g_2x_{2,1} + g_3x_{3,1})$$

$$2z_7 \geq g_1x_{1,7} + g_2x_{2,7} + g_3x_{3,7} - (g_1x_{1,2} + g_2x_{2,2} + g_3x_{3,2})$$

$$2z_8 \geq g_1x_{1,8} + g_2x_{2,8} + g_3x_{3,8} - (g_1x_{1,3} + g_2x_{2,3} + g_3x_{3,3})$$

$$2z_9 \geq g_1x_{1,9} + g_2x_{2,9} + g_3x_{3,9} - (g_1x_{1,4} + g_2x_{2,4} + g_3x_{3,4})$$

$$2z_{10} \geq g_1x_{1,10} + g_2x_{2,10} + g_3x_{3,10} - (g_1x_{1,5} + g_2x_{2,5} + g_3x_{3,5})$$

Restricciones (4.7):

$$2z_6 \geq g_1x_{1,1} + g_2x_{2,1} + g_3x_{3,1} - (g_1x_{1,6} + g_2x_{2,6} + g_3x_{3,6})$$

$$2z_7 \geq g_1x_{1,2} + g_2x_{2,2} + g_3x_{3,2} - (g_1x_{1,7} + g_2x_{2,7} + g_3x_{3,7})$$

$$2z_8 \geq g_1x_{1,3} + g_2x_{2,3} + g_3x_{3,3} - (g_1x_{1,8} + g_2x_{2,8} + g_3x_{3,8})$$

$$2z_9 \geq g_1x_{1,4} + g_2x_{2,4} + g_3x_{3,4} - (g_1x_{1,9} + g_2x_{2,9} + g_3x_{3,9})$$

$$2z_{10} \geq g_1x_{1,5} + g_2x_{2,5} + g_3x_{3,5} - (g_1x_{1,10} + g_2x_{2,10} + g_3x_{3,10})$$

Restricciones (4.8):

$$(x_{1,1}, x_{1,2}, x_{1,3}, x_{1,4}, x_{1,5}, x_{1,6}, x_{1,7}, x_{1,8}, x_{1,9}, x_{1,10}) \in \{0,1\}$$

$$(x_{2,1}, x_{2,2}, x_{2,3}, x_{2,4}, x_{2,5}, x_{2,6}, x_{2,7}, x_{2,8}, x_{2,9}, x_{2,10}) \in \{0,1\}$$

$$(x_{3,1}, x_{3,2}, x_{3,3}, x_{3,4}, x_{3,5}, x_{3,6}, x_{3,7}, x_{3,8}, x_{3,9}, x_{3,10}) \in \{0,1\}$$

$$(y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10}) \in \{0,1\}$$

$$(z_1, z_2, z_3, z_4, z_5, z_6, z_7, z_8, z_9, z_{10}) \in \{0,1\}$$

Modelo Opción 2

Función Objetivo:

$$\text{Minimizar } h \sum_{j>1} y_j + v \sum_{j>PPL} z_j$$

Minimizar

$$1*(y_1 + y_2 + y_3 + y_4 + y_5 + y_6 + y_7 + y_8 + y_9 + y_{10}) + 1*(z_6 + z_7 + z_8 + z_9 + z_{10})$$

Restricciones:

Restricciones (4.2):

$$x_{1,1} + x_{1,2} + x_{1,3} + x_{1,4} + x_{1,5} + x_{1,6} + x_{1,7} + x_{1,8} + x_{1,9} + x_{1,10} = 3$$

$$x_{2,1} + x_{2,2} + x_{2,3} + x_{2,4} + x_{2,5} + x_{2,6} + x_{2,7} + x_{2,8} + x_{2,9} + x_{2,10} = 3$$

$$x_{3,1} + x_{3,2} + x_{3,3} + x_{3,4} + x_{3,5} + x_{3,6} + x_{3,7} + x_{3,8} + x_{3,9} + x_{3,10} = 4$$

Restricciones (4.3):

$$x_{1,1} + x_{2,1} + x_{3,1} = 1$$

$$x_{1,2} + x_{2,2} + x_{3,2} = 1$$

$$x_{1,3} + x_{2,3} + x_{3,3} = 1$$

$$x_{1,4} + x_{2,4} + x_{3,4} = 1$$

$$x_{1,5} + x_{2,5} + x_{3,5} = 1$$

$$x_{1,6} + x_{2,6} + x_{3,6} = 1$$

$$x_{1,7} + x_{2,7} + x_{3,7} = 1$$

$$x_{1,8} + x_{2,8} + x_{3,8} = 1$$

$$x_{1,9} + x_{2,9} + x_{3,9} = 1$$

$$x_{1,10} + x_{2,10} + x_{3,10} = 1$$

Restricciones (4.9):

$$Y_j \geq X_{ij} + \sum_{\substack{i \neq j \\ c_i \neq c_j}} X_{i,j-1} - 1$$

$$y_2 \geq x_{1,2} + (x_{1,1} + x_{2,1} + x_{3,1}) - 1$$

$$y_2 \geq x_{2,2} + (x_{1,1} + x_{2,1} + x_{3,1}) - 1$$

$$y_2 \geq x_{3,2} + (x_{1,1} + x_{2,1} + x_{3,1}) - 1$$

$$y_3 \geq x_{1,3} + (x_{1,2} + x_{2,2} + x_{3,2}) - 1$$

$$y_3 \geq x_{2,3} + (x_{1,2} + x_{2,2} + x_{3,2}) - 1$$

$$y_3 \geq x_{3,3} + (x_{1,2} + x_{2,2} + x_{3,2}) - 1$$

$$y_4 \geq x_{1,4} + (x_{1,3} + x_{2,3} + x_{3,3}) - 1$$

$$y_4 \geq x_{2,4} + (x_{1,3} + x_{2,3} + x_{3,3}) - 1$$

$$y_4 \geq x_{3,4} + (x_{1,3} + x_{2,3} + x_{3,3}) - 1$$

$$y_5 \geq x_{1,5} + (x_{1,4} + x_{2,4} + x_{3,4}) - 1$$

$$y_5 \geq x_{2,5} + (x_{1,4} + x_{2,4} + x_{3,4}) - 1$$

$$y_5 \geq x_{3,5} + (x_{1,4} + x_{2,4} + x_{3,4}) - 1$$

$$y_6 \geq x_{1,6} + (x_{1,5} + x_{2,5} + x_{3,5}) - 1$$

$$y_6 \geq x_{2,6} + (x_{1,5} + x_{2,5} + x_{3,5}) - 1$$

$$y_6 \geq x_{3,6} + (x_{1,5} + x_{2,5} + x_{3,5}) - 1$$

$$y_7 \geq x_{1,7} + (x_{1,6} + x_{2,6} + x_{3,6}) - 1$$

$$y_7 \geq x_{2,7} + (x_{1,6} + x_{2,6} + x_{3,6}) - 1$$

$$y_7 \geq x_{3,7} + (x_{1,6} + x_{2,6} + x_{3,6}) - 1$$

$$y_8 \geq x_{1,8} + (x_{1,7} + x_{2,7} + x_{3,7}) - 1$$

$$y_8 \geq x_{2,8} + (x_{1,7} + x_{2,7} + x_{3,7}) - 1$$

$$y_8 \geq x_{3,8} + (x_{1,7} + x_{2,7} + x_{3,7}) - 1$$

$$y_9 \geq x_{1,9} + (x_{1,8} + x_{2,8} + x_{3,8}) - 1$$

$$y_9 \geq x_{2,9} + (x_{1,8} + x_{2,8} + x_{3,8}) - 1$$

$$y_9 \geq x_{3,9} + (x_{1,8} + x_{2,8} + x_{3,8}) - 1$$

$$y_{10} \geq x_{1,10} + (x_{1,9} + x_{2,9} + x_{3,9}) - 1$$

$$y_{10} \geq x_{2,10} + (x_{1,9} + x_{2,9} + x_{3,9}) - 1$$

$$y_{10} \geq x_{3,10} + (x_{1,9} + x_{2,9} + x_{3,9}) - 1$$

Restricciones (4.10):

$$z_6 \geq x_{1,6} + (x_{1,1} + x_{2,1} + x_{3,1}) - 1$$

$$z_6 \geq x_{2,6} + (x_{1,1} + x_{2,1} + x_{3,1}) - 1$$

$$z_6 \geq x_{3,6} + (x_{1,1} + x_{2,1} + x_{3,1}) - 1$$

$$z_7 \geq x_{1,7} + (x_{1,2} + x_{2,2} + x_{3,2}) - 1$$

$$z_7 \geq x_{2,7} + (x_{1,2} + x_{2,2} + x_{3,2}) - 1$$

$$z_7 \geq x_{3,7} + (x_{1,2} + x_{2,2} + x_{3,2}) - 1$$

$$z_8 \geq x_{1,8} + (x_{1,3} + x_{2,3} + x_{3,3}) - 1$$

$$z_8 \geq x_{2,8} + (x_{1,3} + x_{2,3} + x_{3,3}) - 1$$

$$z_8 \geq x_{3,8} + (x_{1,3} + x_{2,3} + x_{3,3}) - 1$$

$$z_9 \geq x_{1,9} + (x_{1,4} + x_{2,4} + x_{3,4}) - 1$$

$$z_9 \geq x_{2,9} + (x_{1,4} + x_{2,4} + x_{3,4}) - 1$$

$$z_9 \geq x_{3,9} + (x_{1,4} + x_{2,4} + x_{3,4}) - 1$$

$$z_{10} \geq x_{1,10} + (x_{1,5} + x_{2,5} + x_{3,5}) - 1$$

$$z_{10} \geq x_{2,10} + (x_{1,5} + x_{2,5} + x_{3,5}) - 1$$

$$z_{10} \geq x_{3,10} + (x_{1,5} + x_{2,5} + x_{3,5}) - 1$$

Restricciones (4.11):

$$(x_{1,1}, x_{1,2}, x_{1,3}, x_{1,4}, x_{1,5}, x_{1,6}, x_{1,7}, x_{1,8}, x_{1,9}, x_{1,10}) \in \{0,1\}$$

$$(x_{2,1}, x_{2,2}, x_{2,3}, x_{2,4}, x_{2,5}, x_{2,6}, x_{2,7}, x_{2,8}, x_{2,9}, x_{2,10}) \in \{0,1\}$$

$$(x_{3,1}, x_{3,2}, x_{3,3}, x_{3,4}, x_{3,5}, x_{3,6}, x_{3,7}, x_{3,8}, x_{3,9}, x_{3,10}) \in \{0,1\}$$

$$(y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10}) \in \{0,1\}$$

$$(z_1, z_2, z_3, z_4, z_5, z_6, z_7, z_8, z_9, z_{10}) \in \{0,1\}$$

4.2.2.2 Implementación del código de programación generalizado para resolver el problema CBBSSP-SH con GUSEK-GUROBI

A continuación se presenta el código de programación desarrollado para un caso simple del problema CBBSSP-SH desarrollado en GLPK. Este código es convertido a lenguaje MPS para poder ser ejecutado utilizando el *solver* de GUROBI.

En la siguiente sección se presentan los resultados de la implementación de los códigos.

4.2.3 RESULTADOS DE LA EXPERIMENTACION DEL PROBLEMA CBBSSP SIN HUECOS EN GUSEK-GUROBI.

Para definir el grupo de experimentos utilizando el modelo matemático del problema CBBSSP-CH fue necesario hacer varias pruebas. Inicialmente se intentó realizar la experimentación con base en valores altos en las variables independientes. Para el tamaño de la demanda se asignaron inicialmente valores de 100, 1000 y 10000. Para el tamaño de bloques 20, 50 y 100 bloques. Para el caso de modelos y colores valores de 5, 10 y 20. Sin embargo estos valores resultaban demasiado grandes. Esto ocasionaba que el tiempo de resolución no fuese razonable y en la mayor parte de los casos no se pudo concluir el tiempo de ejecución. Para poder encontrar la forma en que la complejidad de los casos empieza a incrementarse se decidió tomar los valores más pequeño posibles. A cada una de las variables independientes mencionadas se les asignó un valor muy bajo, y al generar experimentos con ellos se pudo observar cómo se incrementa el tiempo de resolución.

4.2.3.1 Definición de los casos analizados en los experimentos.

Con base en el punto anterior se definieron como experimentos la combinatoria de las variables independientes con los siguientes valores:

- 1) Demanda: 10, 20, 40
- 2) Número de Bloques: 5, 10
- 3) Número de Modelos: 2, 3, 4, 5
- 4) Número de Colores: 2, 3, 4, 5

De esta manera se generaron 5 Tablas de datos al combinar los casos de Demanda y Bloques:

- 1) Demanda 10 Bloques 5
- 2) Demanda 20 Bloques 5
- 3) Demanda 20 Bloques 10
- 4) Demanda 40 Bloques 5
- 5) Demanda 40 Bloques 10

Es importante aclarar que el caso de Demanda 10 Bloques 10, que sería la única combinatoria faltante para poder combinar todas las demandas y bloques, no tiene sentido, pues al generar los experimentos para todos los casos de combinación de modelos y colores se tendría solo un ciclo, es decir no se podría evaluar ningún cambio en vertical.

El número de experimentos totales para los procedimientos exactos se general al multiplicar las 5 tablas mencionadas por la combinatoria de casos de modelos y colores. Es decir 4 tipos de modelo multiplicado por 4 tipos de colores generan 16 casos para cada tabla. Con esto el número total de experimentos es de 16×5 . En total se hicieron 80 lanzamientos de experimentos.

Adicionalmente se definió como límite el tiempo de 600 segundos (10 minutos) de esta manera si no se obtiene el óptimo en ese tiempo, el experimento se detiene y se genera un informe en el que se indica el caso.

Cada uno de los experimentos se realizó mediante el siguiente procedimiento:

Para cada uno de los 80 casos se generó un código de programación en GLPK tal como se muestra en la sección anterior. GUSEK permite en el módulo "Tools" la construcción o traducción del modelo que se ha escrito en GLPK al lenguaje MPS. El código en MPS puede ser interpretado sin problema por el *solver* de GUROBI. Se ha elegido el *solver* de GUROBI dado que es más eficiente y potente que el *solver* que está integrado en el GUSEK. El tiempo de solución es menor para los casos simples y con esto se puede encontrar la forma en la que crece el nivel de complejidad en la solución del problema a medida que los valores de las variables independientes crecen en las instancias. Se busca determinar si el problema es *NP_hard*. En la siguiente sección se muestran los resultados de los experimentos.

4.2.3.2 Resultados de los experimentos.

Demanda:		10									
Bloques	Mod	Col	Tpo. (Seg)	F. Obj	C MOD	C COL	Best Bound	Nodos	Iteraciones	Procesadores	GAP
5	2	2	0.07	2	1	1	2	5	540	4	0.00%
5	2	3	0.16	2	0	2	2	19	2698	4	0.00%
5	2	4	0.29	3	0	3	3	52	3631	4	0.00%
5	2	5	0.2	4	0	4	4	32	3829	4	0.00%
5	3	2	0.06	3	2	1	3	80	1303	4	0.00%
5	3	3	0.14	4	2	2	4	195	4767	4	0.00%
5	3	4	0.38	4	1	3	4	93	7617	4	0.00%
5	3	5	0.15	5	2	3	5	232	5295	4	0.00%
5	4	2	0.09	2	1	1	2	86	1556	4	0.00%
5	4	3	0.12	3	1	2	3	72	3789	4	0.00%
5	4	4	0.53	4	1	3	4	15	6863	4	0.00%
5	4	5	0.19	5	0	5	5	187	8268	4	0.00%
5	5	2	0.36	3	2	1	3	41	5411	4	0.00%
5	5	3	0.19	3	1	2	3	15	2845	4	0.00%
5	5	4	0.19	5	2	3	5	139	5824	4	0.00%
5	5	5	0.48	5	2	3	5	18	5697	4	0.00%

Tabla 7: Resultados Procedimientos Exactos Sin Huecos con Demanda 10 Bloques 5

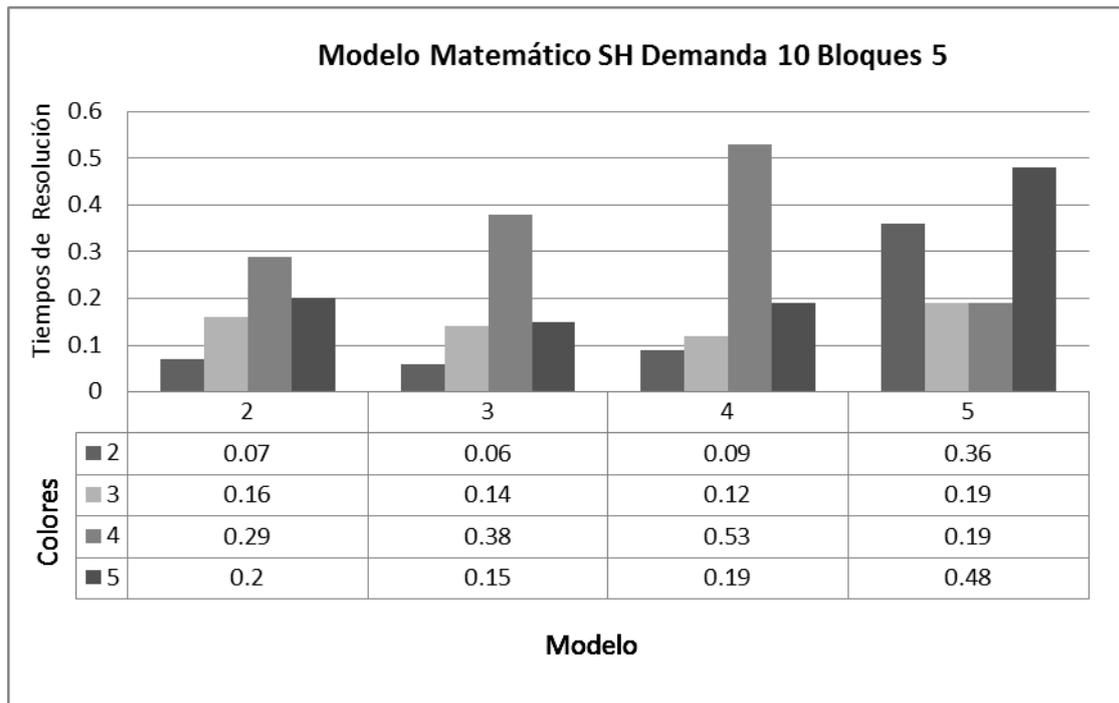


Figura 21: Comparativo del Tiempo de Optimización CBBSSP-SH Dem 10 Bloques 5

Demanda:											
Bloques	Mod	Col	Tpo. (Seg)	F. Obj	C MOD	C COL	Best Bound	Nodos	Iteraciones	Procesadores	GAP
5	2	2	0.02	4	1	3	4	124	3906	4	0.00%
5	2	3	0.33	3	1	2	3	114	7593	4	0.00%
5	2	4	0.97	5	2	3	5	2243	68529	4	0.00%
5	2	5	38.58	6	0	6	6	48320	3810936	4	0.00%
5	3	2	0.28	1	0	1	1	5	2122	4	0.00%
5	3	3	0.95	5	3	2	5	1722	62809	4	0.00%
5	3	4	295.26	6	3	3	6	101525	16587565	4	0.00%
5	3	5	203.42	7	3	4	7	87378	12860108	4	0.00%
5	4	2	0.33	3	2	1	3	307	12881	4	0.00%
5	4	3	1.36	6	4	2	6	2059	79463	4	0.00%
5	4	4	2.61	4	1	3	4	1739	134386	4	0.00%
5	4	5	600	8	2	6	6	163784	33391906	4	25.00%
5	5	2	0.83	4	3	1	4	1436	56790	4	0.00%
5	5	3	8.1	5	3	2	5	7698	543544	4	0.00%
5	5	4	6.26	5	2	3	5	3686	336755	4	0.00%
5	5	5	98.62	7	3	4	7	12182	3524381	4	0.00%

Tabla 8: Resultados Procedimientos Exactos Sin Huecos con Demanda 20 Bloques 5

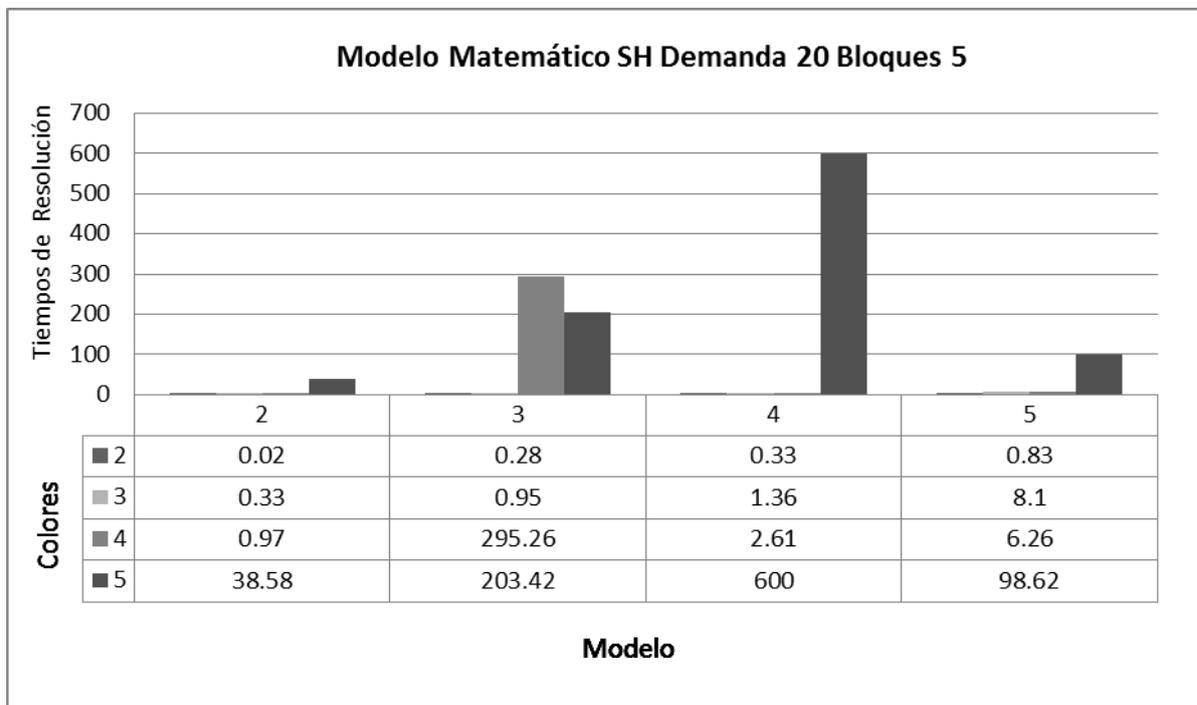


Figura 22: Comparativo del Tiempo de Optimización CBBSSP-SH Dem 20 Bloques 5

Demanda:		20									
Bloques	Mod	Col	Tpo. (Seg)	F. Obj	C MOD	C COL	Best Bound	Nodos	Iteraciones	Procesadores	GAP
10	2	2	0.14	1	0	1	1	3	1309	4	0.00%
10	2	3	0.3	3	1	2	3	751	19945	4	0.00%
10	2	4	1.53	4	1	3	4	3093	95086	4	0.00%
10	2	5	1.15	4	0	4	4	1340	67060	4	0.00%
10	3	2	0.11	1	0	1	1	0	1458	4	0.00%
10	3	3	0.86	4	1	3	4	2200	58828	4	0.00%
10	3	4	5.4	4	1	3	4	8195	329975	4	0.00%
10	3	5	70.75	5	1	4	5	41844	5164458	4	0.00%
10	4	2	12.98	4	1	3	4	34323	1082257	4	0.00%
10	4	3	0.42	4	1	3	4	411	19647	4	0.00%
10	4	4	19.16	4	1	3	4	15181	1302237	4	0.00%
10	4	5	600	7	1	6	6	299913	43277317	4	14.29%
10	5	2	9.64	4	3	1	4	29218	977100	4	0.00%
10	5	3	28.08	4	2	2	4	26757	2120226	4	0.00%
10	5	4	529.91	5	2	3	5	156010	28973416	4	0.00%
10	5	5	64.15	6	2	4	6	16952	3194566	4	0.00%

Tabla 9: Resultados Procedimientos Exactos Sin Huecos con Demanda 20 Bloques 10

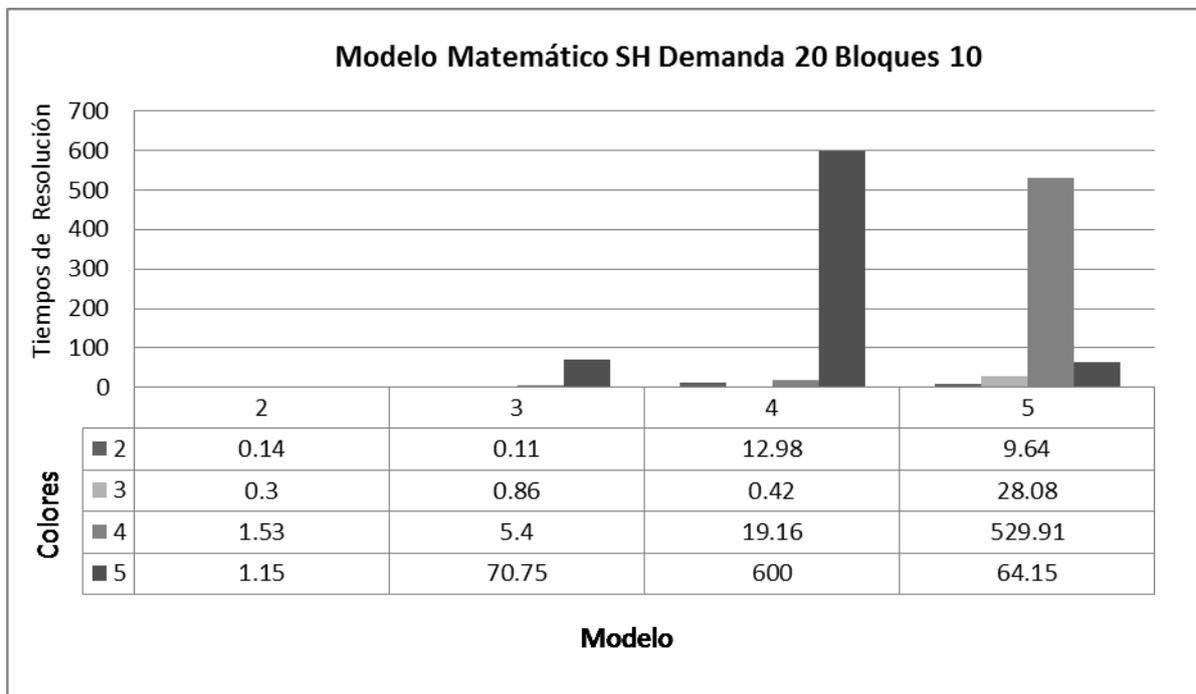


Figura 23: Comparativo del Tiempo de Optimización CBBSSP-SH Dem 10 Bloques 10

Demanda:		40									
Bloques	Mod	Col	Tpo. (Seg)	F. Obj	C MOD	C COL	Best Bound	Nodos	Iteraciones	Procesadores	GAP
5	2	2	0.39	2	1	1	2	339	15239	4	0.00%
5	2	3	5.13	4	2	2	4	4785	287531	4	0.00%
5	2	4	76.77	6	2	4	6	32037	4572711	4	0.00%
5	2	5	600	8	1	7	6	68748	19995767	4	25.00%
5	3	2	3	4	3	1	4	3019	179841	4	0.00%
5	3	3	90.01	5	3	2	5	22955	4354195	4	0.00%
5	3	4	600	7	2	5	3	29773	10668511	4	57.14%
5	3	5	600	8	4	4	4	95278	26617137	4	50.00%
5	4	2	99.14	6	5	1	6	55658	6385591	4	0.00%
5	4	3	600	7	4	3	5	86360	25303504	4	28.57%
5	4	4	600	11	4	7	3	21227	9505336	4	72.72%
5	4	5	600	12	4	8	3	49988	21187880	4	75.00%
5	5	2	7.08	4	3	1	4	3621	332088	4	0.00%
5	5	3	600	7	3	4	3	72386	23298056	4	57.14%
5	5	4	600	13	5	8	3	18637	12417940	4	76.92%
5	5	5	600	15	6	9	4	21404	11795205	4	73.33%

Tabla 10: Resultados Procedimientos Exactos Sin Huecos con Demanda 40 Bloques 5

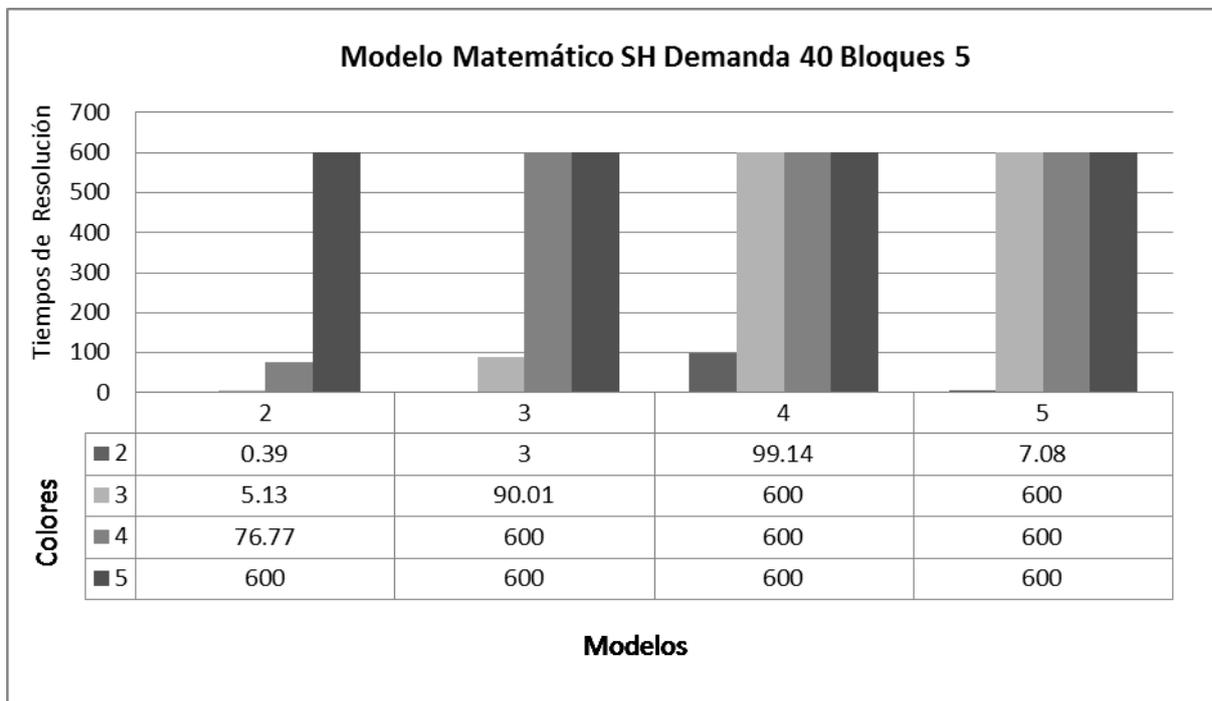


Figura 24: Comparativo del Tiempo de Optimización CBBSSP-SH Dem 40 Bloques 5

Demanda:		40									
Bloques	Mod	Col	Tpo. (Seg)	F. Obj	C MOD	C COL	Best Bound	Nodos	Iteraciones	Procesadores	GAP
10	2	2	0.81	3	0	3	3	1886	53027	4	0.00%
10	2	3	0.44	2	0	2	2	55	8445	4	0.00%
10	2	4	67.01	5	0	5	5	25941	4874513	4	0.00%
10	2	5	600	7	1	6	6	74562	22973764	4	14.29%
10	3	2	55.75	4	3	1	4	48286	4602356	4	0.00%
10	3	3	600	5	3	2	4	51223	14509728	4	20.00%
10	3	4	600	6	2	4	4	79254	25135276	4	33.33%
10	3	5	600	9	1	8	4	146014	30770862	4	55.56%
10	4	2	66.45	6	3	3	6	80353	4984181	4	0.00%
10	4	3	562.1	6	3	3	6	135322	28380605	4	0.00%
10	4	4	600	8	2	6	3	42616	16720021	4	62.50%
10	4	5	600	9	2	7	4	22741	11457966	4	55.56%
10	5	2	585.45	5	4	1	5	188948	34968492	4	0.00%
10	5	3	600	5	3	2	3	82579	27999572	4	40.00%
10	5	4	600	11	5	6	3	93015	25162358	4	72.72%
10	5	5	600	13	3	10	3	34750	17501672	4	76.92%

Tabla 11: Resultados Procedimientos Exactos Sin Huecos con Demanda 40 Bloques 10

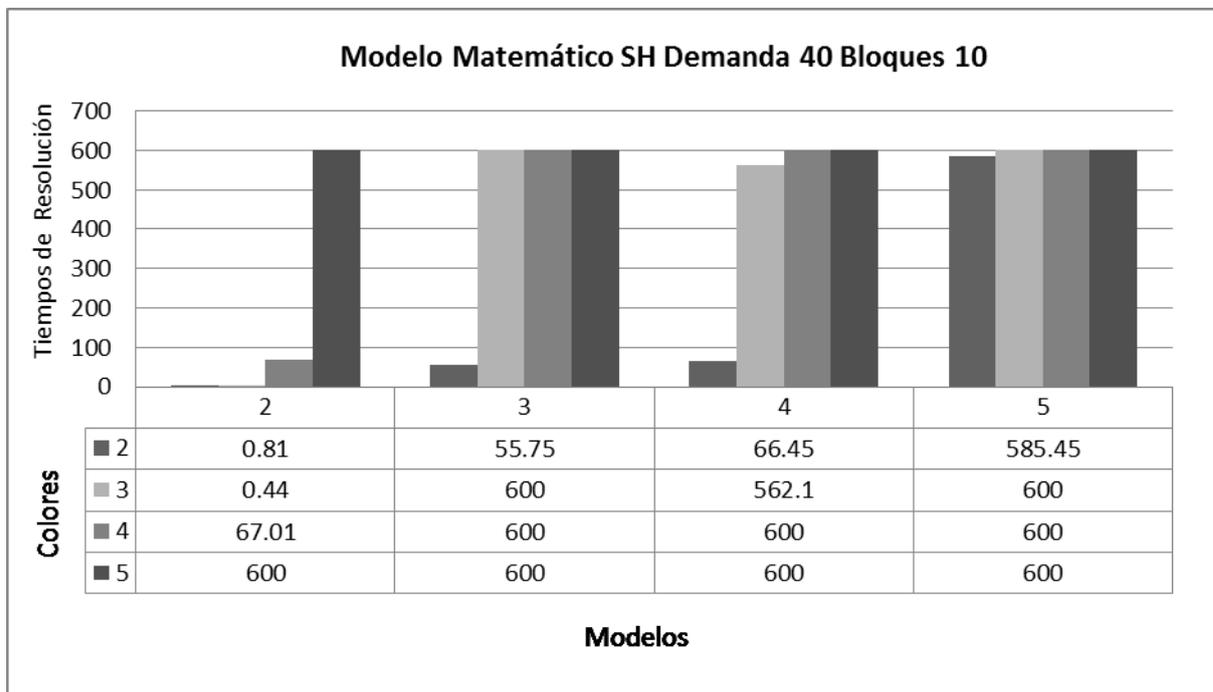


Figura 25: Comparativo del Tiempo de Optimización CBBSSP-SH Dem 40 Bloques 10

4.2.4 ANALISIS DE RESULTADOS DE LOS PROCEDIMIENTOS EXACTOS SIN HUECOS

De los experimentos con base en los modelos con procedimientos exactos en el caso del Problema CBBSSP-SH se puede concluir que rápidamente el tiempo de ejecución crece a medida que se incrementa el valor de las variables independientes. Es decir a medida que el valor del número de modelos y colores, el tamaño de la demanda y los bloques es mayor, el tiempo de ejecución aumenta. En este caso y como se ha señalado, se ha puesto

como límite superior en el tiempo 600 segundos. Los experimentos terminaban en el momento en que superaban este valor. Ese tiempo de 10 minutos indica que el tiempo de ejecución es muy largo. En las tablas (7-11) y en las Figuras (17-21) se puede observar el comportamiento señalado.

4.3 PROCEDIMIENTOS HEURÍSTICOS PARA LA RESOLUCION DEL PROBLEMA CBBSSP SIN HUECOS

INTRODUCCION

En la presente sección se describen las heurísticas que se emplean para la resolución el problema del CBBSSP-SH (*Cyclic Batch and Bi-Objective Setups Scheduling Problem Without Holes*), es decir, Problema de Programación de Lotes Cíclicos con Tiempos de Preparación Bi-Objetivo Sin Huecos. La necesidad de métodos heurísticos se tiene por el hecho de que el problema CBBSSP-SH es *NP-hard*, por lo que resolverlo de manera exacta suele implicar un tiempo computacional muy grande cuando se tiene un elevado número de variables y restricciones. Las heurísticas que se presentan se aplican en un entorno controlado restringiendo el número de variables independientes que tienen un efecto significativo sobre las variables dependientes.

Se diseñan una serie de heurísticas para el problema del CBBSSP-SH utilizando la combinatoria de una serie de categorías conceptuales:

- 1) Listas: La forma de secuenciación de las listas de entrada.
- 2) Estrategia: La forma de alimentación en el proceso, sea ésta horizontal, vertical o diagonal.
- 3) Acomodo: El acomodo libre o el acomodo utilizando alguna restricción. En el caso del acomodo con restricciones se generan una serie de combinaciones dependiendo de la existencia o no de un renglón fijo y la existencia o no de una columna fija.
- 4) Ventaja: La ventaja al modelo o la ventaja al color en caso de que las características sean las mismas (empate) al elegir el siguiente producto.

De todas las posibles combinaciones de las variables planteadas se considera necesario restringir las heurísticas a aquellas que en forma realista sean aplicables en el campo de la fabricación industrial. Sin considerar las listas, el total de heurísticas propuestas son 27.

4.3.1 METODOLOGIA DEL DESARROLLO DE HEURISTICAS SIN HUECOS.

El desarrollo de las heurísticas se hace en este capítulo. Se parte del concepto simple de alimentación del producto en una secuencia previamente establecida. El método para establecer la secuencia inicial utiliza específicamente diferentes tipos de listas. Las listas son nueve y dependen del orden ascendente o descendente de variable independientes. Las variables independientes para las listas son: color, geometría o modelo, demanda o cantidad a producir. Una vez establecida la lista será alimentada en el proceso de acuerdo a un procedimiento distinto para cada heurística. Los procedimientos son generados a partir de la combinación de las categorías mencionadas en el apartado anterior: Estrategia, Acomodo y Ventaja.

De esta manera, en este capítulo se procede en primer lugar a exponer un marco general de las heurísticas desarrolladas partiendo de las tres estrategias básicas que se han desarrollado, en segundo lugar para cada estrategia se establecen los procedimientos para las diferentes opciones generadas por los conceptos de acomodo y ventaja. Así se presentan las heurísticas básicas y los correspondientes apartados en los que son descritas con cada una de sus nueve variantes:

- 1) H-H: Heurísticas basadas en la estrategia horizontal (sección 4.3.3)
- 2) H-V: Heurísticas basadas en la estrategia vertical (sección 4.3.4)
- 3) H-D: Heurísticas basadas en la estrategia diagonal (sección 4.3.5)

Por último, se termina el apartado con una codificación completa de los procedimientos heurísticos aquí presentados. Esta codificación es la que sirve de base para realizar el estudio experimental. En el apartado siguiente se hace la experimentación y se presentan los resultados. En esta apartado se presenta al final una sección de conclusiones sobre la definición de heurísticas y codificación de los modelos diseñados.

4.3.2 MARCO GENERAL DE LAS HEURISTICAS DESARROLLADAS PARA EL PROBLEMA DEL CBBSSP-SH.

Las heurísticas planteadas en esta tesis para la resolución del problema CBBSSP-SH son de comprensión sencilla y pueden ser utilizadas en la industria dado que la implementación tampoco representa complejidad, Estas heurísticas determinan el orden de alimentación de los productos a un sistema de ciclo cerrado considerando el número de bloques disponibles, la cantidad a fabricar de cada tipo de producto y la variedad de los mismos dados por el color y la geometría. De acuerdo a los establecido en el capítulo 3 Descripción del Problema de Programación de Lotes Cíclicos con tiempos de Preparación Bi-Objetivo Dependientes e Independientes de la Secuencia en el caso del problema Sin Huecos (CBBSSP-SH) (apartado 3.2.3) el objetivo de la tesis es minimizar el coste total asociado con el número de cambios de color y de modelos en los sistemas cíclicos de pintura en la industria de autopartes automotrices lo cual se establece como función objetivo de las heurísticas desarrolladas en el presente capítulo.

4.3.2.1 Procedimiento General:

El procedimiento general de las heurísticas está basado en un algoritmo constructivo con tres elementos:

- 1) Una lista en la que los productos son agrupados y dentro de la cual son elegidos para ser secuenciados. La lista se ordena de acuerdo a criterios específicos.
- 2) Una estrategia para elegir una posición en la cuál será acomodado el producto que será seleccionado.
- 3) Una estrategia para seleccionar un producto de la lista, tomando en consideración las cantidades que se deben fabricar de cada tipo de producto, el acomodo y la ventaja.

Listas:

La lista de prioridades puede definirse utilizando nueve posibilidades de secuenciación:

- 1) Secuencia por modelo y después por color: En este caso todos los productos son agrupados de acuerdo a su geometría y ordenados lexicográficamente. A continuación cada subconjunto es ordenado internamente en forma lexicográfica por color.
- 2) Secuencia por color y después por modelo: Similar al anterior, pero en este caso todos los productos son agrupados primero de acuerdo a su color y los subconjuntos generados son ordenados lexicográficamente entre ellos. A continuación cada subconjunto es ordenado internamente en forma lexicográfica por modelo.
- 3) Por tamaño de la demanda y después por color. Los productos son secuenciados dando prioridad al orden descendente en el tamaño de la demanda por producto (Requerimiento). Si hay un empate en dicha demanda en dos o más tipos de producto se decide por la secuencia de color. En otras palabras se hace un ordenamiento de los productos por tamaño de la demanda y cuando se tiene la misma demanda en algunos productos se selecciona el siguiente producto por el orden lexicográfico del color.
- 4) Por tamaño de la demanda y después por modelo. Los productos son secuenciados dando prioridad al orden descendente en el tamaño de la demanda por producto (Requerimiento). Si hay un empate en dicha demanda en dos o más tipos de producto se decide por la secuencia de modelo. En otras palabras se hace un ordenamiento de los productos por tamaño de la demanda y cuando se tiene la misma demanda en algunos productos se selecciona el siguiente producto por el orden lexicográfico del modelo.

- 5) Secuencia por color y después por el tamaño de la demanda de cada modelo: En este caso todos los productos son agrupados primero de acuerdo a su color y secuenciados lexicográficamente. Los subconjuntos generados se ordenan a su vez al interior dependiendo del tamaño de la demanda por modelo.
- 6) Secuencia por modelo y después por el tamaño de la demanda de cada color: En este caso todos los productos son agrupados primero de acuerdo a su modelo y secuenciados lexicográficamente. Los subconjuntos generados se ordenan a su vez al interior dependiendo del tamaño de la demanda por color.
- 7) Tamaño de la demanda por color y después por el tamaño de la demanda de cada modelo: En este caso todos los productos son agrupados primero de acuerdo a su color y secuenciados por el tamaño de la demanda total de cada color. Los subconjuntos generados se ordenan a su vez al interior dependiendo del tamaño de la demanda por modelo.
- 8) Tamaño de la demanda por modelo y después por el tamaño de la demanda de cada color: En este caso primero todos los productos son agrupados de acuerdo al modelo y secuenciados por el tamaño de la demanda total de cada modelo. Los subconjuntos generados se ordenan a su vez al interior dependiendo del tamaño de la demanda por color.
- 9) Secuencia estocástica: En forma aleatoria se elige el siguiente producto en la secuencia.

Una vez que se obtienen las diferentes listas de acuerdo a los criterios mencionados en los nueve casos anteriores éstas se utilizan para ser expandidas de acuerdo a la demanda solicitada para cada tipo de producto. En otras palabras, en cada lista se obtiene el orden definitivo de cada producto individual (no solo por tipo de producto). Si la lista incluye 10 tipos de productos con demanda de 5 cada uno, la lista antes de expandirla tiene 10 elementos y la lista expandida contiene 50.

Estrategia de Acomodo.

Una estrategia para elegir una posición en la cuál será acomodado el producto que será seleccionado. La aplicación de una estrategia de secuenciación implica la elección de una posición en la que un producto individual, seleccionado de una manera específica, es asignado. La forma de elegir una posición en la matriz de acomodo puede ser en secuencia horizontal, vertical o diagonal.

Estrategia para seleccionar un Producto de la Lista:

La selección de un producto de la lista puede ser en forma libre y directa o atendiendo a algunas restricciones impuestas por las características de la heurística que en particular se ejecute. Las características en el caso de los procedimientos con huecos pueden ser por: 1) Considerar la existencia o no de

una columna cero y/o una fila cero y 2) El seguir una ventaja en color o una ventaja en modelo en el caso de existir dos productos que puedan cumplir las restricciones de selección

Filas y Columna Cero

Las columnas y filas cero sirven como marco de referencia al momento de asignar un producto. El tipo de acomodo indica el tipo de secuencia que se debe seguir y la estrategia de selección ayuda a elegir un producto de la lista que se debe colocar como el siguiente en el proceso de producción. En este caso, si existe una fila o columna cero se debe atender primero al tipo de producto que marque la referencia cero antes de colocar un producto con las mismas características que el anterior que ha sido colocado. Esta regla implica que muchas veces aunque exista un producto que cumpla horizontalmente con el producto previo y verticalmente con el que fue colocado en el ciclo anterior, se debe colocar el que cumpla con la referencia cero.

En las heurísticas desarrolladas para los procedimientos del problema sin huecos el tipo de referencia cero, sea columna o fila puede ser del tipo: Proporcional.

Las filas y/o columnas cero tipo Proporcional son asignadas calculando la proporción o porcentaje por tipo de producto del total de la demanda de productos. Para el caso de la columna cero, la proporción se calcula respecto al total de demanda por color. Así cada posición de la columna cero tendrá un color fijo que servirá de referencia. Es decir. La fila 1, en la posición de columna cero, determinará el color de referencia del inicio de esa fila 1. La fila 2 de la columna cero, el color de la segunda fila y así sucesivamente. Lo mismo sucederá en el caso de la fila cero, que determinará en este caso el modelo de referencia de toda la columna 1.

Ventaja:

La restricción de ventaja en cada heurística indica si al momento de existir un producto que cumple las restricciones por el tipo de color de referencia en la columna cero o el previo en la misma fila y al mismo tiempo otro producto cumple con la restricción por el tipo de modelo de referencia en la fila cero o el previo en la misma columna se toma una decisión dando una preferencia al color o al modelo.

4.3.2.2 Tabla General de las Heurísticas CBBSSP-CH:

Se presenta a continuación en la Tabla 12 un resumen de las heurísticas desarrolladas para resolver el problema CBBSSP-CH. En ella se asigna un número de algoritmo que sirve como identificación a lo largo de esta Tesis, se da la nomenclatura general que describe a cada algoritmo y se presentan las características que hacen distinto a cada uno.

	Algoritmo	Descripción	Características		
			Ventaja	Fila 0	Columna 0
Heurísticas Horizontales	9	HL	L	N	N
	10	HVC	VC	N	N
	11	HVM	VM	N	N
	12	HVC + F0	VC	F0	N
	13	HVC + C0	VC	N	C0
	14	HVC + F0 + C0	VC	F0	C0
	15	HVM + F0	VM	F0	N
	16	HVM + C0	VM	N	C0
	17	HVM + F0 + C0	VM	F0	C0
Heurísticas Verticales	19	VL	L	N	N
	20	VVC	VC	N	N
	21	VVM	VM	N	N
	22	VVC + F0	VC	F0	N
	23	VVC + C0	VC	N	C0
	24	VVC + F0 + C0	VC	F0	C0
	25	VVM + F0	VM	F0	N
	26	VVM + C0	VM	N	C0
	27	VVM + F0 + C0	VM	F0	C0
Heurísticas Diagonales	29	DL	L	N	N
	30	DVC	VC	N	N
	31	DVM	VM	N	N
	32	DVC + F0	VC	F0	N
	33	DVC + C0	VC	N	C0
	34	DVC + F0 + C0	VC	F0	C0
	35	DVM + F0	VM	F0	N
	36	DVM + C0	VM	N	C0
	37	DVM + F0 + C0	VM	F0	C0

Tabla 12: Marco General de Heurísticas Desarrolladas para el Problema Sin Huecos

La referencia a la tabla permite una visión global sobre los procedimientos, mismos que se procede a revisar y analizar por tipo de Estrategia en las tres secciones que a continuación se presentan.

4.3.3 HEURISTICAS HORIZONTALES

Las heurísticas horizontales desarrolladas se presentan a continuación. Se eligió como forma de referenciarlas asignar un número consecutivo a partir del 9 para las heurísticas con esta estrategia. En la lista siguiente se describe cada una de ellas y en la Tabla 13 se observa la clasificación general de las mismas.

- Algoritmo 9: Heurística Horizontal Libre.
- Algoritmo 10: Heurística Horizontal con Ventaja Color.
- Algoritmo 11: Heurística Horizontal con Ventaja Modelo.
- Algoritmo 12: Heurística Horizontal con Ventaja Color y Fila cero.
- Algoritmo 13: Heurística Horizontal con Ventaja Color y Columna cero.

- Algoritmo 14: Heurística Horizontal con Ventaja Color, Fila Cero y Columna Cero.
- Algoritmo 15: Heurística Horizontal con Ventaja Modelo y Fila cero.
- Algoritmo 16: Heurística Horizontal con Ventaja Modelo y Columna cero.
- Algoritmo 17: Heurística Horizontal con Ventaja Modelo, Fila cero y Columna cero.

HORIZONTAL	Libre	Ventaja Color		Ventaja Modelo	
	Sin Fila 0	Sin Fila 0	Con Fila 0	Sin Fila 0	Con Fila 0
Sin Columna 0	Alg. 9	Alg. 10	Alg. 12	Alg. 11	Alg. 15
Con Columna 0		Alg. 13	Alg. 14	Alg. 16	Alg. 17

Tabla 13: Resumen Heurísticas Horizontales Sin Huecos

4.3.4 HEURISTICAS VERTICALES

Las heurísticas verticales desarrolladas se presentan a continuación. Se eligió como forma de referenciarlas asignar un número consecutivo a partir del 19 para las heurísticas con esta estrategia. En la lista siguiente se describe cada una de ellas y en la Tabla 14 se observa la clasificación general de las mismas.

- Algoritmo 19: Heurística Vertical Libre.
- Algoritmo 20: Heurística Vertical con Ventaja Color.
- Algoritmo 21: Heurística Vertical con Ventaja Modelo.
- Algoritmo 22: Heurística Vertical con Ventaja Color y Fila cero.
- Algoritmo 23: Heurística Vertical con Ventaja Color y Columna cero.
- Algoritmo 24: Heurística Vertical con Ventaja Color, Fila Cero y Columna Cero.
- Algoritmo 25: Heurística Vertical con Ventaja Modelo y Fila cero.
- Algoritmo 26: Heurística Vertical con Ventaja Modelo y Columna cero.
- Algoritmo 27: Heurística Vertical con Ventaja Modelo, Fila cero y Columna cero.

VERTICAL	Libre	Ventaja Color		Ventaja Modelo	
	Sin Fila 0	Sin Fila 0	Con Fila 0	Sin Fila 0	Con Fila 0
Sin Columna 0	Alg. 19	Alg. 20	Alg. 22	Alg. 21	Alg. 25
Con Columna 0		Alg. 23	Alg. 24	Alg. 26	Alg. 27

Tabla 14: Resumen Heurísticas Verticales Sin Huecos

4.3.5 HEURISTICAS DIAGONALES

Las heurísticas diagonales desarrolladas se presentan a continuación. Se eligió como forma de referenciarlas asignar un número consecutivo a partir del 29 para las heurísticas con esta estrategia. En la lista siguiente se describe cada una de ellas y en la Tabla 15 se observa la clasificación general de las mismas.

- Algoritmo 29: Heurística Diagonal Libre.
- Algoritmo 30: Heurística Diagonal con Ventaja Color.
- Algoritmo 31: Heurística Diagonal con Ventaja Modelo.
- Algoritmo 32: Heurística Diagonal con Ventaja Color y Fila cero.
- Algoritmo 33: Heurística Diagonal con Ventaja Color y Columna cero.
- Algoritmo 34: Heurística Diagonal con Ventaja Color, Fila Cero y Columna Cero.
- Algoritmo 35: Heurística Diagonal con Ventaja Modelo y Fila cero.
- Algoritmo 36: Heurística Diagonal con Ventaja Modelo y Columna cero.
- Algoritmo 37: Heurística Diagonal con Ventaja Modelo, Fila cero y Columna cero.

DIAGONAL	Libre	Ventaja Color		Ventaja Modelo	
	Sin Fila 0	Sin Fila 0	Con Fila 0	Sin Fila 0	Con Fila 0
Sin Columna 0	Alg. 29	Alg. 30	Alg. 32	Alg. 31	Alg. 35
Con Columna 0		Alg. 33	Alg. 34	Alg. 36	Alg. 37

Tabla 15: Resumen Heurísticas Diagonales Sin Huecos

4.3.6 CODIFICACION Y DEFINICION DE LOS PROCEDIMIENTOS EXISTENTES.

En este apartado se han desarrollado 37 heurísticas para resolver el problema CBBSSP-SH. La cantidad de procedimientos permite explorar las variables dependientes e independientes del problema. Se ha desarrollado el código de programación para cada heurística asignando un algoritmo para cada una. El lenguaje utilizado es JAVA y el software es ECLIPSE

4.4 ESTUDIO EXPERIMENTAL DE LOS PROCEDIMIENTOS DESARROLLADOS

INTRODUCCION

Para definir el grupo de experimentos utilizando el modelo matemático del problema CBBSSP-SH fue necesario determinar valores realistas que exploren el caso de problemas aplicables en la industria.

Un objetivo adicional al asignar los valores a las variables de cada instancia es conocer el comportamiento de los resultados a distintos niveles y la combinación de los efectos entre variables.

El objetivo principal en este apartado es determinar cuál o cuáles son las heurísticas en las que se obtiene el resultado que minimice el número de cambios sea por modelo o por color.

4.4.1 METODOLOGIA DEL ESTUDIO EXPERIMENTAL.

Como se ha mencionado, el estudio experimental tiene como objetivo determinar las heurísticas de mejor desempeño. Para poder tener argumentos que lo demuestren se establece el método como sigue:

- 1) Se generan instancias que validen los resultados de las heurísticas en valores realistas y aplicables en la industria. Que muestren posibles valores que ocurren en las empresas. Asegurar que esas instancias permitan un análisis de sensibilidad de los valores y el efecto de distintos niveles.
- 2) Ejecutar los experimentos con los algoritmos de las heurísticas desarrollados en Eclipse JAVA.
- 3) Analizar los resultados utilizando software comercial. En este caso SPSS 17. Obtener datos de valores de resultados por heurística y por Lista. Considerar todos los casos generados.
- 4) Dado que las variables dependientes son dos: Número de Cambios de Color Horizontal y Número de Cambios de Modelo Vertical. Definir una forma de lograr hacer un análisis Mono-Objetivo.

- 5) La definición del Análisis Mono-Objetivo consiste en determinar una forma de unificar en una sola variable las dos variables dependientes. En este caso se decidió generar un Valor "P" (Entre 0.1 y 0.9) y determinar una formulación en la que el valor de "P" multiplicado por una de las dos variables y el valor de (1 - P) por el valor de la otra variable. Ambas variables divididas por el valor mínimo posible de la misma (el óptimo relativo) y sumados. Se genera un valor único.
- 6) Se define como las mejores heurísticas aquellas que mantienen el mejor lugar relativo en forma constante para distintos valores de "P".

4.4.2 GENERACION DE PROBLEMAS PARA EL ESTUDIO EXPERIMENTAL

Con base en el punto anterior se definieron como experimentos la combinatoria de las variables independientes con los valores que se presentan en la Tabla 16:

Variables	Nombre	Variantes	Valores
Instancias del problema o Variables Independientes	Color	3	5,10,20
	Modelo	3	5,10,20
	Demanda	3	100, 1000, 10000
	Bloques	3	20, 50, 100
	Tipo Dato	3	1, 2, 3
	Valor Alfa	3	1, 1.2, 1.5
Procedimentales	Algoritmo	27	9..17, 19..27, 29..37
	Lista	9	1..9
Variables Dependientes	NC Col Hor		Min y Max
	NC Mod Ver		Min y Max

Tabla 16: Instancias del Problema CBBSSP-SH

A continuación se presenta una explicación y/o justificación de los valores para cada instancia:

- 1) Color, modelo, demanda y bloques: Estas variables en el problema se debe explorar para valores pequeños y grandes.
- 2) Tipo de dato:
 1. Estocástico. Las entradas son totalmente aleatorias.
 2. Método Zipft No Aleatorio.
 3. Método Zipft Aleatorio.
- 3) Alfa: Valor que tiene alfa en el Método Zipft.
- 4) Algoritmo: Número de procedimiento heurístico específico para resolver el problema.
- 5) Lista: Número de ordenamiento específico inicial de las listas de secuenciación.
- 6) NC Col Hor: Número de *setups* horizontales por cambio de color.

7) NC Mod Ver: Número de *setups* verticales por cambio de modelo.

El número de experimentos totales para los procedimientos exactos se general al multiplicar las variantes de las instancias mostradas. Es importante aclarar que el Método Zipft solo aplica en el caso de los tipos de datos 2 y 3 por lo que en el tipo de dato 1 no se generan estas variantes. Adicionalmente no tiene sentido aplicar la combinatoria de tamaño de demanda 100 y número de bloques 100 pues no habría un segundo ciclo con el cuál evaluar los *setups* verticales.

Se generó una base de datos con 612,360 registros.

Se puede conocer la forma como se integró esta base de datos considerando los tres tipos conceptuales de variables en la tabla anterior:

1) Instancias:

(3 variantes de colores)
x (3 variantes de modelos)
x (3 variantes de demanda)
x (3 variantes de bloques)
x (3 variantes de tipo de datos (Los que aplican Método Zipft: 2 y 3))
x (3 variantes del valor Zipft)
= 486 Instancias

Adicionalmente las variantes del Tipo de datos 1:

(3 variantes de colores)
x (3 variantes de modelos)
x (3 variantes de demanda)
x (3 variantes de bloques)
= 81 Instancias

2) Procedimientos:

(27 heurísticas de solución)
x (9 Listas)
= 243 procedimientos.

3) El total de experimentos se compone finalmente así:

(486 + 81) Instancias
x 243 Procedimientos:
= 137, 781 casos

Dado que a este valor se necesita descontar los casos mencionados en que no tiene sentido evaluar una heurística con el mismo tamaño de demanda y bloques se tienen: $137, 781 - 15, 309 = 122,472$ casos.

Buscando dar validez estadística a los resultados se generaron 5 repeticiones de los experimentos. Con esto el total queda en $122,472 \times 5 = 612,360$ experimentos.

4.4.3 ESTUDIO EXPERIMENTAL DE LOS PROCEDIMIENTOS HEURISTICOS

Análisis Mono-Objetivo de los resultados.

Se presenta a continuación el análisis estadístico de los experimentos agrupando los datos para delimitar los resultados por las variables procedimentales y por los valores de las mismas.

4.4.3.1 Análisis de las Listas

El objetivo de la presente sección es revisar el comportamiento de las listas con diferentes valores de P. El valor de P representa la importancia que tiene para el usuario cada una de las variables dependientes, es decir la importancia que tiene minimizar el número de cambios de color o minimizar el número de cambios de modelos.

Comportamiento General de las Listas:

El comportamiento de las listas depende del nivel elegido del factor "P". Se puede observar que algunas de las listas son en algunos de los valores mejores que otros, es decir, minimizan el número de cambios multiplicado por la importancia relativa. Sin embargo esto no ocurre en todos los valores de "P". En general se puede hacer una clasificación de las listas de acuerdo a los resultados obtenidos (Ver Figura 26):

- 1) Listas que mantienen el comportamiento a niveles mínimos relativos con el factor "P" con valores en el rango (0.1 – 0.5) y alto con valores superiores a 0.5. Tendencia ascendente.
- 2) Listas que mantienen el comportamiento con un nivel alto relativo con el factor "P" con valores en el rango (0.1 – 0.5) y bajo con valores iguales o mayores a 0.5. Tendencia descendente.
- 3) Listas con un comportamiento medio en los valores extremos de "P" (0.1 , 0.9) y muy alto en los valores centrales de "P" (0.2 – 0.8). Tendencia cresta.
- 4) Lista con bajo desempeño manteniendo un valor constante alto en la posición relativa respecto a otras listas.

Caso 1, Tendencia Ascendente:

Tal como se puede observar en la Figura 26, en el primer caso se encuentran las listas: 7, 5 y 2.

- 1) Lista 7: Ordena primero por la Demanda de cada color y enseguida por la Demanda de cada Modelo.

- 2) Lista 5: Ordena primero lexicográficamente por Color y enseguida por la Demanda de cada Modelo.
- 3) Lista 2: Ordena primero lexicográficamente por Color y enseguida lexicográficamente por Modelo.

Del comportamiento observado en estas tres listas se concluye lo siguiente:

- a) Ordenar por Demanda (DC+DM, Lista 7) genera el mejor valor (mínimo) en el comportamiento de las listas pero sólo en valores de "P" bajos ($P < 0.5$).
- b) Ordenar lexicográficamente por Color y enseguida por Demanda de Modelo (Lista 5) resulta en un comportamiento similar al de la Lista 7 pero con menos variación en los extremos del valor "P". En valores de "P" ($P \leq 0.5$) no es mejor que la lista 7 pero tiene valores relativos de segundo lugar entre las listas y en valores superiores a ($P > 0.5$) tiene resultados de menor desempeño en relación a otras listas pero mejor que el de la lista 7.
- c) La lista 2 en todo caso tiene menor desempeño en relación a las listas 7 y 5.

Caso 2, Tendencia Descendente:

Se puede observar de igual manera en la Figura 26, que en el segundo caso se encuentran las listas: 6, 8 y 1.

- 1) Lista 6: Ordena primero lexicográficamente por Modelo y enseguida por la Demanda de cada Color.
- 2) Lista 8: Ordena primero por la Demanda de cada Modelo y enseguida por la Demanda de cada Color.
- 3) Lista 1: Ordena primero lexicográficamente por Modelo y enseguida lexicográficamente por Color.

Del comportamiento de estas tres listas se puede concluir que:

- a) Ordenar lexicográficamente por Modelo y enseguida por Demanda de Color (ML+DC, Lista 6) genera el mejor valor (mínimo) en el comportamiento de las listas pero sólo en valores de "P" altos ($P > 0.5$).
- b) Ordenar por Demanda de Modelo y enseguida por Demanda de Color (Lista 8) resulta en un comportamiento similar al de la Lista 6 pero con menor desempeño, en todo caso pierde al menos una posición respecto a la lista 6. En el valor $P=3$ son 3 posiciones las que pierde.
- c) La Lista 1 en todo caso tiene menor desempeño en relación a las listas 6 y 8 y pierde al menos una posición relativa respecto a la Lista 8 y dos respecto a la Lista 6.

Caso 3, Tendencia en Cresta:

El tercer caso se indica para las listas 3 y 4. El comportamiento es similar en ambas listas y tiene un desempeño medio en los valores extremos de "P" y de bajo rendimiento a medida que se incrementa el valor de ($P=0.1$) o se disminuye el valor de ($P=0.5$). Se tiene el peor caso en los valores centrales: Para la Lista 4 en ($P=0.5$) y para la Lista 3 en ($P=0.6$). En cualquier caso el comportamiento es el mismo para ambas listas. A partir de ($P \geq 0.6$) la Lista 4 es mejor que la 3, pero la posición que estas ocupan siempre es de menor desempeño que otras listas. Para valores de "P" ($P \leq 0.7$) las listas del Caso 1 son mejores siempre. Para valores de "P" ($P \geq 0.4$) las listas del Caso 2 en todo caso son mejores.

Caso 4, Comportamiento Constante:

La Lista 9 es estocástica y en todo caso mantiene un desempeño inferior al de las demás listas. Este comportamiento es esperado y sirve de comparativo relativo para evaluar el desempeño de las demás listas.

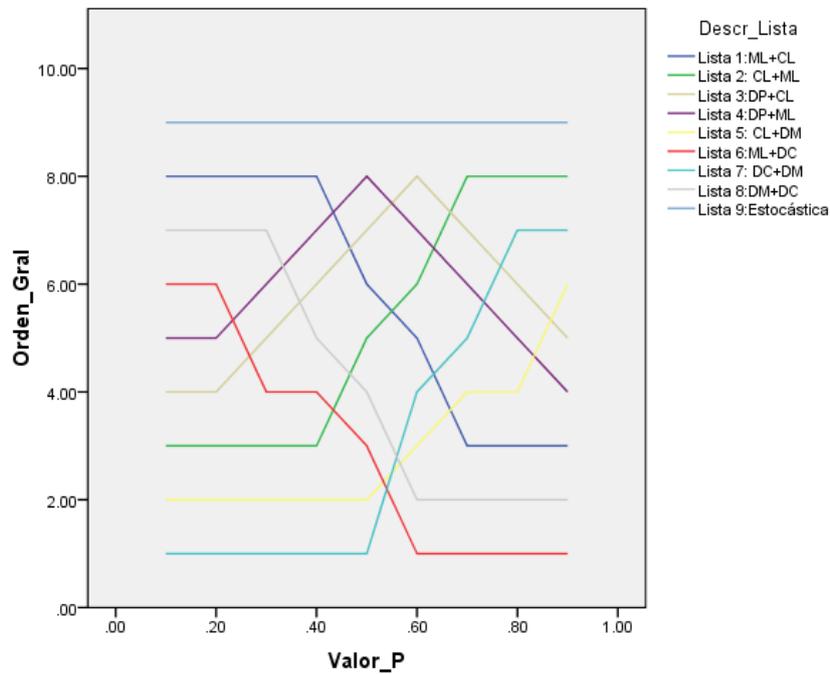


Figura 26: Comportamiento relativo de las Listas respecto al óptimo con Factor "P"

Resultados de las Listas:

De acuerdo a los casos mencionados y a la figura 1 se puede concluir que las listas de mejor desempeño son:

- 1) Lista 7, caso 1.
- 2) Lista 6, caso 2.
- 3) Lista 5, caso 1.

La Lista 7 presenta el mejor desempeño con la mejor posición hasta valores de "P" ($P \leq 0.5$). La Lista 6 muestra la mejor posición a partir de

($P \leq 0.6$). La Lista 5 aunque nunca ocupa la primera posición muestra menos variabilidad respecto a su media que la Lista 7, es decir tiene un comportamiento menos extremo y correspondería a quien elige el método la decisión de seleccionarla en caso de que se busque mayor estabilidad en los resultados. A continuación en las figuras 23, 24, y 25 se muestra el comportamiento individual de estas tres listas:

Comportamiento Lista 7:

Esta lista se ordena primero pro Demanda de color y después por demanda de modelo:

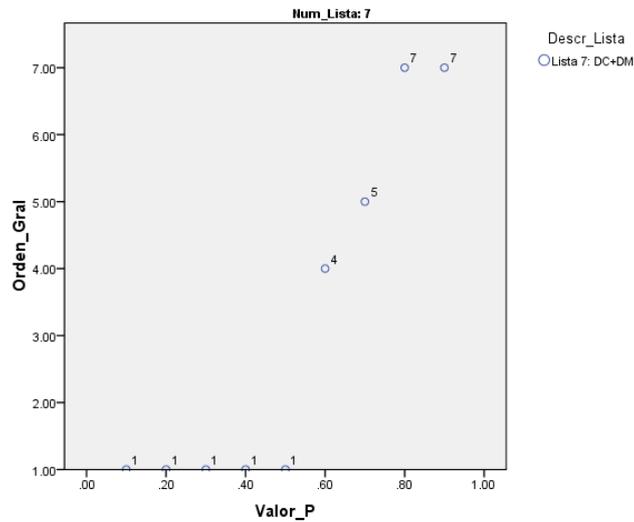


Figura 27: Comportamiento Relativo de la Lista 7 en función del valor "P"

Comportamiento Lista 6:

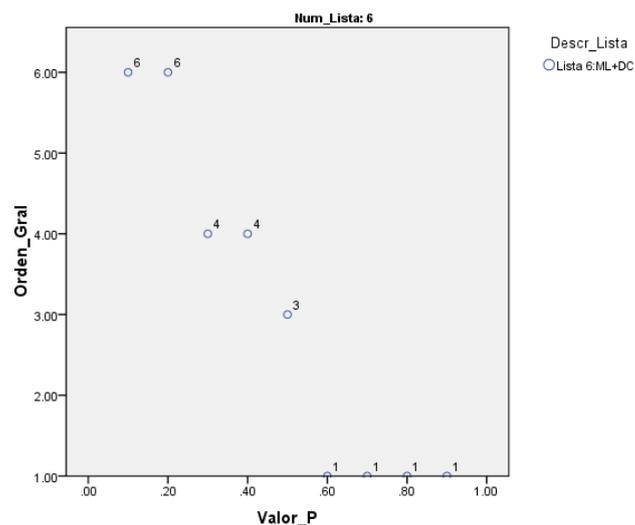


Figura 28: Comportamiento Relativo de la Lista 6 en función del valor "P"

Comportamiento Lista 5:

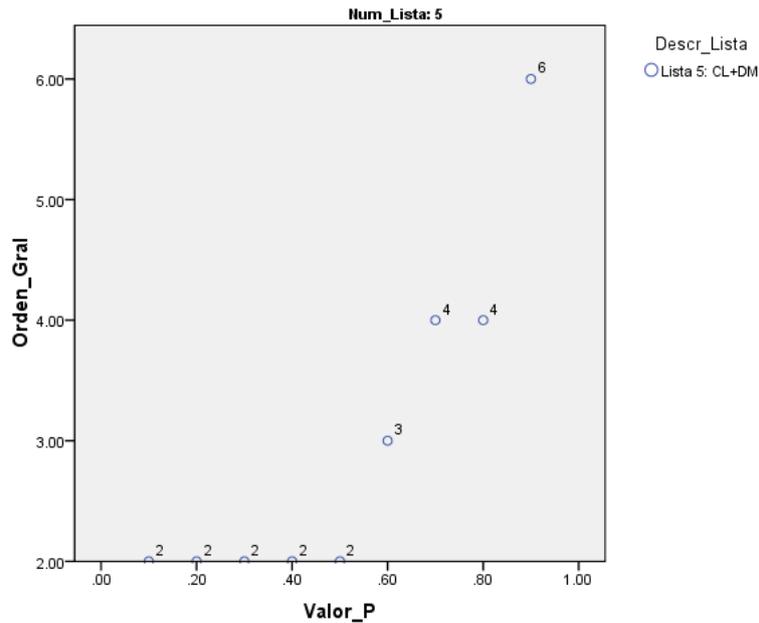


Figura 29: Comportamiento Relativo de la Lista 5 en función del valor "P"

Comportamiento Resto de Listas:

Para ver el comportamiento individual de las listas con menor desempeño relativo ver Anexo 2.

4.4.3.2 Análisis Algoritmos

El comportamiento general de cada algoritmo es distinto en cada caso. El objetivo de la presente sección es revisar el comportamiento de los algoritmos con diferentes valores de P. El valor de P representa la importancia que tiene para el usuario cada una de las variables dependientes, es decir la importancia que tiene minimizar el número de cambios de color o minimizar el número de cambios de modelos.

Comportamiento General de los Algoritmos

El comportamiento de los algoritmos está relacionado directamente con el valor elegido del factor "P". Se puede observar que en muchos de los casos los resultados de los algoritmos evaluados a distintos niveles son mejores que los resultados de otros, es decir, minimizan el número de cambios multiplicado por la importancia relativa. Es objeto de este análisis revisar y evaluar en qué casos ocurre esto y si hay alguno o varios algoritmos que en forma consistente se mantienen en una posición relativa mejor para todos o al menos la mayor parte de los valores de "P". En general se puede hacer una clasificación de los algoritmos de acuerdo a los resultados obtenidos (Ver Figura 30):

- 1) Tendencia Ascendente: Algoritmos que tienen un desempeño inferior en la posición relativa a medida que se incrementa el valor de "P".
- 2) Tendencia Descendente: Algoritmos que en valores bajos de "P" tienen un desempeño bajo en la posición relativa respecto a otros algoritmos. A medida que se incrementa el valor de "P" el desempeño mejora.

- 3) Tendencia en Valle. Algoritmos que tienen en los valores extremos de “P” un menor desempeño al tener una posición relativa inferior respecto al resto de los algoritmos.
- 4) Tendencia en Cresta. . Algoritmos que tienen en los valores extremos de “P” un mejor desempeño al tener una posición relativa menor respecto al resto de los algoritmos.
- 5) Tendencia en Valle y Cresta. Algoritmos con un comportamiento similar al de la tendencia ascendente pero que en el valor o algunos de los valores más bajos de “P” tienen un desempeño inferior al de los valores siguientes y al final en los valores más altos de “P” mejora su posición relativa.
- 6) Tendencia en Cresta y Valle. Algoritmos con un comportamiento similar a la tendencia descendente pero que en el valor o algunos de los valores más bajos de “P” tienen un mejor desempeño al de los valores siguientes y al final en los valores más altos de “P” obtiene posiciones relativamente inferiores a las inmediatamente anteriores.

Caso 1, Tendencia Ascendente:

Tal como se puede observar en la Figura 30, en el primer caso se encuentran los algoritmos: 12, 10, 9, 30 y 14.

Del comportamiento observado en estos tres algoritmos se concluye lo siguiente:

- a) Los Algoritmos 12, 10 y 9 tienen una pendiente similar y esta es mayor a la de los algoritmos 30 y 14.
- b) El Algoritmo 12 es el mejor absoluto cuando el valor de “P” es ($P=0.1$). A partir de ($P=0.3$) el algoritmo con mejor desempeño en la Tendencia Ascendente es el Algoritmo 30.
- c) El Algoritmo 30 cuando ($P=0.1$) tiene una posición absoluta de posición 7 de 27 y relativa de posición 4 de 5. A partir de ($P=0.3$) obtiene la mejor posición relativa dentro de su tendencia, con una pendiente menor a los demás algoritmos.
- d) Descripción de los mejores algoritmos dentro del grupo:
 - a. El algoritmo 12 es horizontal con ventaja color y fila cero.
 - b. El algoritmo 30 es diagonal con ventaja color.

Caso 2, Tendencia Descendente:

Al revisar de nuevo la Figura 30 se observa que en el segundo caso se encuentran los algoritmos:

15, 31, 36, 17, 35, 37, 26, 27, 21, 25 y 19.

Estos 11 algoritmos forman el grupo más numeroso. Del comportamiento observado en ellos concluye lo siguiente:

- a) El algoritmo 15 presenta el mejor desempeño relativo dentro del grupo al presentar la mejor posición de los algoritmos descendentes para el rango de "P" desde (P=0.1) hasta (P=0.8). tan solo en el valor 0.9 el desempeño es menor a otros dos algoritmos del grupo.
- b) La posición del Algoritmo 15 cuando (P=0.1) es 16 y cuando (P=0.9) es 3.
- c) Los Algoritmos 35 y 37 pasan de las posiciones 20 y 21 respectivamente, cuando (P=0.1) a las posiciones 1 y 2 cuando (P=0.9), ubicándose en ese valor de "P" en las mejores posiciones absolutas de los algoritmos. Es mejor el Algoritmo 35 que el Algoritmo 37 siempre en al menos una posición.
- d) Descripción de los mejores algoritmos dentro del grupo:
 - a. El algoritmo 15 es horizontal con ventaja modelo y fila cero
 - b. El algoritmo 35 es diagonal con ventaja modelo y fila cero

Caso 3, Tendencia en Valle:

También en la Figura 30 se puede ver que en este caso se encuentran los algoritmos:

33, 11, 32, 23, 24, 16 y 20.

Es deseable la permanencia en la posición más baja. Cuando ocurre así se aprecia la estabilidad y la robustez del algoritmo para los diferentes valores de "P". En los casos revisados se observa lo siguiente:

- a) El algoritmo 33 tiene la mejor posición relativa cuando el valor de "P" (P=0.1). Sin embargo a partir de (P>=0.2) la posición que ocupa es superada por la del Algoritmo 11. A partir del valor de "P" (P=0.4) el comportamiento que manifiesta es ascendente.
- b) El algoritmo 11 muestra el comportamiento más estable y robusto tanto en el grupo como en comparación absoluta. Solo es superado dentro del grupo cuando (P=0.1) por el algoritmos 33 y cuando (P=0.9) por el algoritmo 20. En todos los demás casos de P la posición que ocupa es 1 dentro del grupo y dentro del total de algoritmos.
- c) El algoritmo 20 cuando (P=0.1) ocupa la posición 15, el mejor desempeño lo tiene cuando (P=0.8) con la posición 2 que aunque es muy buena no supera a la del Algoritmo 11 que tiene la posición 1. Con el valor de (P=0.9) obtiene la posición 4 que es la mejor dentro del grupo para dicho valor de "P".
- d) Descripción de los mejores algoritmos dentro del grupo:
 - a. Algoritmo 33 es diagonal con ventaja color y columna cero.
 - b. Algoritmo 11 es horizontal con ventaja modelo
 - c. Algoritmo 20 es vertical con ventaja color.

Caso 4, Tendencia en Cresta:

Se determina en la Figura 30 que los algoritmos en Cresta son:

13 y 29.

En este caso el comportamiento es inverso al del caso anterior. La estabilidad se obtiene con un valor mayor al de los puntos extremos. En los algoritmos que se desarrollaron este comportamiento solo ocurre en dos algoritmos y sobre estos se concluye lo siguiente:

- a) Para el valor de "P" ($P=0.1$) los algoritmos 13 y 29 ocupan las posiciones 9 y 22 en relación al total de algoritmos. Para el valor "P" ($P=0.9$) la posición que logran es la 26 para el Algoritmo 13 y el Algoritmo 29 queda en la 22, a misma que con ($P=0.1$). En ambos casos los valores de "P" distintos a ($P=0.1$) o ($P=0.9$) son iguales o de más bajo desempeño en cuanto a la posición ocupada.
- b) No se recomienda el uso de estos algoritmos pues en ambos casos son superados por algoritmos de otras tendencias.

Caso 5, Tendencia en Valle y Cresta:

De la Figura 30 se puede determinar que el único algoritmo en este caso es el 34. De su desempeño se concluye lo siguiente:

- a) Con un valor "P" ($P=0.1$) tiene una posición 4 de 27 que es relativamente muy buena y se mueve a la posición 3 cuando ($P=0.2$) donde se mantiene estable hasta ($P=0.4$). A partir de esta valor su comportamiento es ascendente, similar al del Algoritmo 33 y logra superar a este último en una posición solamente con el valor ($P=0.9$) donde ocupa el sitio 18.
- b) El algoritmo 33 es mejor que el Algoritmo 34 en una posición para casi todos los valores de "P". Solo en ($P=0.9$) el Algoritmo 34 supera al 33.
- c) No se recomienda el uso de este algoritmo pues manifiesta estabilidad en muy pocos valores de "P" (0.2, 0.3, 0.4) y en los demás valores es superado por otros algoritmos.

Caso 6, Tendencia en Cresta y Valle:

Se observa en la Figura 30 que el algoritmo 22 inicia con un comportamiento en Cresta y después sigue uno de Valle. Sobre su desempeño se concluye como sigue:

- a) El mejor desempeño de este algoritmo es ocupar la posición 6 para los valores de "P" (0.6, 0.7 y 0.8). En todas las demás posiciones su desempeño es más bajo.
- b) Presenta un comportamiento similar al del Algoritmo 20, el cual lo supera en la posición que ocupa en al menos una posición. Solamente es mejor el Algoritmo 22 que el 20 con el valor de "P" ($P=0.1$) pero ocupando una posición 14 que relativamente es solo media.
- c) Este algoritmo presenta un comportamiento estable respecto a su media, es decir, la desviación estándar que presenta en cuanto a la posición ocupada es menor que la de otros algoritmos.

d) No se recomienda el uso de este Algoritmo pues existen al menos 5 algoritmos que lo superan en su mejor posición.

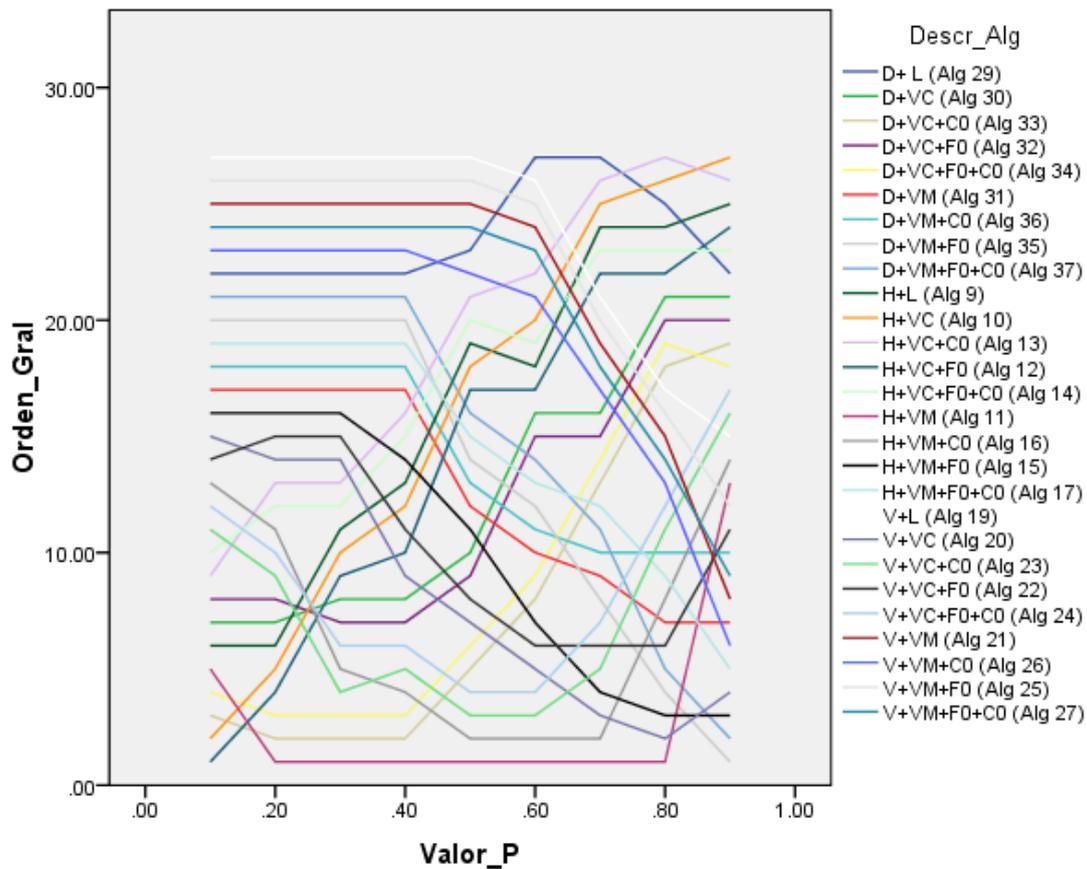


Figura 30: Comportamiento relativo de los Algoritmos respecto al óptimo con Factor "P"

Resultados de los Algoritmos:

De acuerdo a los casos mencionados y a la Figura 30 se puede concluir que los algoritmos con mejor desempeño son:

- 1) Algoritmo 12, caso 1.
- 2) Algoritmo 30, caso 1.
- 3) Algoritmo 15, caso 2.
- 4) Algoritmo 35, caso 2.
- 5) Algoritmo 33, caso 3.
- 6) Algoritmo 11, caso 3.
- 7) Algoritmo 20, caso 3.

En la Figura 31 se puede ver en forma separada a estos algoritmos para mayor claridad sobre su desempeño. De los 7 seleccionados se puede observar que 3 son ascendentes, 3 descendentes y uno estable con valores de "P" que consideran todo el espectro de posibilidades.

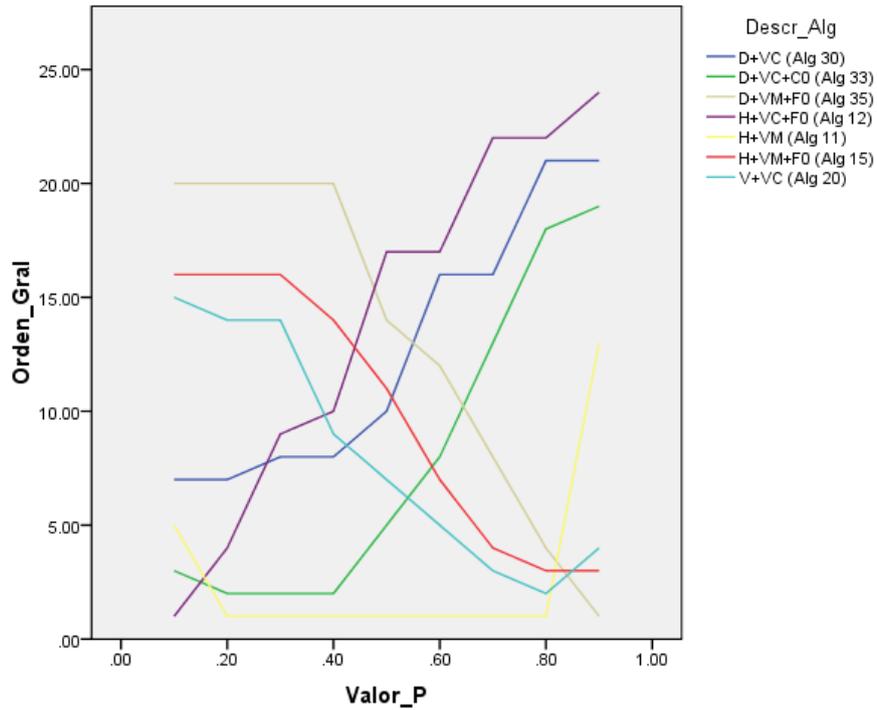


Figura 31: Comportamiento relativo de los Algoritmos con mejor desempeño respecto al óptimo con Factor "P"

El algoritmo 11 presenta el mejor desempeño con la mejor posición para valores de "P" ($0.2 \leq P \leq 0.8$). Los algoritmos 12 y 35 presentan el mejor resultado para los valores de "P" ($P=0.1$) y ($P=0.9$) respectivamente. Sin embargo los algoritmos 12 y 35 tienen una pendiente importante respecto a otros algoritmos que para esos valores de "P" no proporcionan la mejor posición, pero que tienen un desempeño mejor en otras instancias de "P". Tal es el caso del algoritmo 33 que en el valor de "P" ($P=0.2$) tiene la posición 3.

5 PROBLEMA CBBSSP CON HUECOS

5.1 MODELADO MATEMATICO PARA EL PROBLEMA CBBSSP CON HUECOS

INTRODUCCION

El problema puede ser formulado como un modelo mixto-entero. En esta sección se presentan dos opciones de modelado, incluyendo en cada una tanto la nomenclatura como el modelo matemático. El problema CBBSSP-CH puede ser identificado en los sistemas de pintura de partes automotrices. En empresas como Plastic Omnium, de la que se hizo una breve descripción en el Capítulo 3 se indica que el coste de dejar un espacio vacío es muy alto. El espacio vacío en este tipo de empresas se puede evitar algunas veces dependiendo del tipo de cambio de color. Es decir cuando el cambio se hace entre colores de intensidad similar a veces no es necesario dejar un espacio vacío. Sin embargo, en la modelación presentada se busca la generalidad y no se hacen consideraciones por cambios entre colores de intensidad parecida. Se busca que el modelo sea aplicable en todos los sistemas donde exista un flujo cíclico y se tengan al menos dos tipos de tiempos de preparación, con una estación de carga de las partes y otra de descarga. En cada tipo de tiempo de preparación se hace la consideración de dejar un espacio vacío. El espacio vacío o hueco representa un coste que debe ser minimizado.

5.1.1 MODELADO DEL PROBLEMA CBBSSP-CH OPCION 1

La primera opción de modelado es bidimensional y al evaluar la posición de un producto determinado en la secuencia compara con base en los índices de producto y de posición. Este modelo se formula como sigue:

5.1.1.1 NOMENCLATURA

Se definen índices, parámetros y variables de decisión que corresponden al caso general del problema en la primera opción.

Indices:

- i : Número del tipo de producto que se asigna (un producto está definido por un combinación de geometría y color).
- j : Número de la posición que se asigna.
- g : Número de la geometría que se revisa.
- c : Número del color que se revisa

Parámetros:

- n : Número de productos (un producto está definido por un combinación de geometría y color).
- n_c : Número de colores.

- n_g : Número de geometrías.
 d_i : Demanda para el producto i , para $i=1,\dots,n$
 c_i : Color del producto i , para $i=1,\dots,n$
 g_i : Geometría del producto i para $i=1,\dots,n$
 N : Número mínimo de posiciones, donde $N = \sum_{i=1}^n d_i$
 N_{MAX} : Número máximo de posiciones, donde $N_{MAX} = N*2$ Si se considera el peor caso posible donde después de la asignación de un producto en una posición se asigna un espacio vacío.
 PPL : Elementos por circuito o bucle

Variables de Decisión:

Se consideran los índices y variables presentados para definir adicionalmente el siguiente grupo de variables binarias de decisión, las cuales permiten obtener información sobre la asignación de un producto específico a una posición determinada, si existe un cambio de color entre dos posiciones secuenciales, si existe un cambio de modelo o geometría en la misma posición al ejecutar un ciclo subsecuente en el proceso o si existe un espacio vacío o hueco en una posición específica. Las variables se indican a continuación:

$x_{ij} = \begin{cases} 1 \\ 0 \end{cases}$ 1 si el producto i es colocado en la posición j , 0 en cualquier otro caso.

$y_j = \begin{cases} 1 \\ 0 \end{cases}$ 1 si existe un cambio de color entre las posiciones $j-1$ y j , 0 en cualquier otro caso.

$z_j = \begin{cases} 1 \\ 0 \end{cases}$ 1 si existe un cambio de jig entre las posiciones $j - PPL$ y j , 0 en cualquier otro caso.

$V_j = \begin{cases} 1 \\ 0 \end{cases}$ 1 si existe un espacio vacío o hueco en la posición j , 0 en cualquier otro caso.

Tomando en consideración los índices y parámetros indicados, así como las variables de decisión se procede a presentar el modelo matemático para la opción 1.

5.1.1.2 MODELO DEL PROBLEMA CBBSSPCH OPCION 1

Función Objetivo del CBBSSPCH Opción 1.

En los sistemas cíclicos de pintura se presentan los tipos de tiempos de preparación mencionados en la sección anterior. Cuando en el sistema no se permiten espacios vacíos, el objetivo es minimizar el coste total de los mismos,

es decir minimizar los costes relacionados con el número de cambios de color y los que dependen del número de cambios de jig. En el caso del modelo con huecos, los costes se determinan por la cantidad de espacios sin asignación de producto y estos están relacionados en forma directamente proporcional al número de cambios, sean e color o de geometría que son generados.

Sin perder generalidad, el modelo asume que cualquier cambio sea en color o en geometría genera un espacio vacío o hueco.

De acuerdo a lo indicado la expresión matemática de la función objetivo para el CBBSSPCH es:

$$\text{Minimizar} \quad \sum_{j>1} V_j \quad (5.1)$$

Restricciones básicas del problema CBBSSPCH Opción 1.

Se debe optimizar la función objetivo cumpliendo una serie de restricciones

Restricciones de satisfacción de la demanda.

La demanda para cada tipo de producto i debe ser completada si se revisan todas las posiciones existentes en el sistema. Esto se asegura mediante el siguiente grupo de restricciones:

$$\sum_j X_{ij} = d_i \quad i = 1, \dots, n \quad (5.2)$$

Restricciones de asignación como máximo de un solo producto en cada posición disponible.

Cada posición j disponible en el sistema debe tener asignado como máximo un solo producto, lo cual se restringe asegurando que la sumatoria de todos los tipos de producto i en una posición específica j (variable de decisión X_{ij}) sea igual menor o igual a 1. A diferencia del modelo sin huecos, la restricción no puede ser igual a 1 porque también existe la posibilidad de un espacio vacío, es decir sin asignación de producto. La sumatoria para el producto i se hace para el máximo de todas las posiciones j que son posibles.

$$\sum_i X_{ij} \leq 1 \quad j = 1, \dots, N_{MAX} \quad (5.3)$$

Restricciones que limitan como máximo un solo espacio vacío entre posiciones consecutivas.

Se debe asegurar que al sumar el valor e la sumatoria X_{ij} de las posiciones j y $j+1$ se obtenga un resultado mayor a cero. Así se descarta la posibilidad de dos posiciones consecutivas iguales a cero.

$$\sum_i X_{ij} + \sum_i X_{i,j-1} > 0 \quad j = 2, \dots, N_{MAX} \quad (5.4)$$

Restricciones para contabilizar los cambios de color.

Mediante estas restricciones se asegura que la variable de decisión Y_j tenga el valor de 1 cuando ocurre un cambio de color entre la posición j y la posición $j - 1$. Se hace la evaluación a partir de la posición $j = 2$, para poder comparar con la posición $j = 1$. Se revisan todas las posiciones hasta N_{MAX} , el número máximo de posiciones.

$$n_c y_j \geq \sum_i c_i x_{ij} - \sum_i c_i x_{i,j-1} \quad j = 2, \dots, N_{MAX} \quad (5.5)$$

$$n_c y_j \geq \sum_i c_i x_{i,j-1} - \sum_i c_i x_{ij} \quad j = 2, \dots, N_{MAX} \quad (5.6)$$

Estas dos restricciones operan en conjunto. La restricción (5.5) funciona de la siguiente manera:

- 1) Se evalúa para todos los productos i si han sido asignados en la posición j . con lo que el valor de X_{ij} en dicha variable de decisión será 1 en alguno de los casos de i .
- 2) Multiplica el valor para cada caso de X_{ij} por el valor de C_i correspondiente. En un solo caso, aquel en el que el valor de X_{ij} es 1 se obtendrá un valor positivo.
- 3) Se hace la sumatoria de todos los casos i en la posición j
- 4) Se evalúa para todos los productos i si han sido asignados en la posición $j-1$. con lo que el valor de X_{ij-1} en dicha variable de decisión será 1 en alguno de los casos de i .
- 5) Multiplica el valor para cada caso de X_{ij-1} por el valor de C_i correspondiente. En un solo caso, aquel en el que el valor de X_{ij-1} es 1 se obtendrá un valor positivo.
- 6) Se hace la sumatoria de todos los casos i en la posición $j-1$
- 7) Se hace la diferencia entre la sumatoria de los casos en j menos la sumatoria de los casos en $j - 1$
- 8) Se divide el resultado de la diferencia entre n_c con lo que se obtiene un valor igual a cero o distinto a cero.
- 9) Se restringe el valor de Y_j con una desigualdad que limita a dicha variable de decisión a que sea mayor o igual al valor obtenido en el punto anterior. Es decir Y_j será mayor o igual que cero o que el valor obtenido.

La restricción (5.6) funciona como complemento de la restricción (5.5). El cálculo del valor es en la práctica igual con los términos de la diferencia invertidos. Es decir, la diferencia se calcula restando a la sumatoria de los casos en $j - 1$ el valor de la sumatoria de los casos en j .

Como ambas restricciones se deben de cumplir en el modelo, el resultado en la variable binaria Y_j solo podrá ser con valor 0 en el caso de que el color sea el mismo en las posiciones j y $j-1$ y en el caso de que el color sea distinto entre las dos posiciones mencionadas el valor será 1.

A continuación se indican los pasos para la restricción (5.6):

- 1) Se evalúa para todos los productos i si han sido asignados en la posición $j-1$. con lo que el valor de X_{ij-1} en dicha variable de decisión será 1 en alguno de los casos de i .
- 2) Multiplica el valor para cada caso de X_{ij-1} por el valor de C_i correspondiente. En un solo caso, aquel en el que el valor de X_{ij-1} es 1 se obtendrá un valor positivo.
- 3) Se hace la sumatoria de todos los casos i en la posición $j-1$
- 4) Se evalúa para todos los productos i si han sido asignados en la posición j . con lo que el valor de X_{ij} en dicha variable de decisión será 1 en alguno de los casos de i .
- 5) Multiplica el valor para cada caso de X_{ij} por el valor de C_i correspondiente. En un solo caso, aquel en el que el valor de X_{ij} es 1 se obtendrá un valor positivo.
- 6) Se hace la sumatoria de todos los casos i en la posición j
- 7) Se hace la diferencia entre la sumatoria de los casos en $j-1$ menos la sumatoria de los casos en j
- 8) Se divide el resultado de la diferencia entre n_c con lo que se obtiene un valor igual a cero o distinto a cero.
- 9) Se restringe el valor de Y_j con una desigualdad que limita a dicha variable de decisión a que sea mayor o igual al valor obtenido en el punto anterior. Es decir Y_j será mayor o igual que cero o que el valor obtenido.

Restricciones para asignar en la posición j un espacio vacío por cambio de color

El valor de Y_j indica que existe un cambio de color entre la posición j y $j-1$.

Cuando existe un espacio vacío, el valor de Y_j será igual a 1 en dos posiciones consecutivas, es decir en Y_j y en Y_{j+1} .

Al hacer la multiplicación del valor de Y_j por el valor de Y_{j+1} el valor de V_j será igual a 1 solo donde hay un espacio vacío, en cualquier otro caso será 0.

$$V_j \geq y_j * y_{j+1} \quad j = 2, \dots, N_{MAX} - 1 \quad (5.7)$$

Restricciones para asegurar que cuando hay cambio de color entre la posición $j+1$ y $j-1$ el producto sea distinto en la posición j

Cuando existe un espacio vacío se cuentan dos cambios de color: uno al cambiar del color anterior al espacio vacío y otro del vacío al color nuevo. Que haya un solo cambio se debe restringir pues implica que no se asignó el espacio vacío.

El valor de V_j dado por la restricción (5.7) se suma al producto de multiplicar Y_j por Y_{j-1} y se le suma el valor de V_j . El resultado obtenido solamente puede ser "0,1, 2".

El valor de "1" se obtiene cuando solo hay un cambio de color y se debe restringir.

Para eliminar el valor de 1 al término explicado se le resta 1.

El término anterior se eleva al cuadrado para tener solo los valores positivos y restringir al valor "1" toda la ecuación.

$$\left[(y_j * y_{j+1}) + ((y_j * y_{j-1}) + y_j - 1) \right]^2 = 1 \quad j = 3, \dots, N_{MAX} - 1 \quad (5.8)$$

Restricciones para asegurar que en la posición j exista un espacio vacío y no un producto distinto cuando entre las posiciones j+1 y j-1 el producto sea distinto.

La ecuación (5.8) asegura que cuando haya un cambio de color entre los dos productos involucrados haya un espacio o un producto con color distinto.

Esta restricción asegura que cuando haya un cambio de color de color se inserte un espacio vacío y no un producto con color distinto.

La forma de operar de esta restricción es comparar el color del producto j-1 con el color del producto j+1, y al mismo tiempo el color del producto j-1 con el del producto j.

Demostración de la restricción:

$$\text{Sea } A = \sum_i C_i X_{i,j-1} * \sum_i C_i X_{i,j+1}$$

$$\text{Sea } B = \sum_i C_i X_{i,j-1} * \sum_i C_i X_{i,j}$$

La ecuación siguiente asegura que la posición j sea igual a "0":

$$0 \leq [A - (A + B) + 1] V_j \leq 1$$

Simplificando se obtiene:

$$0 \leq [1 - B] V_j \leq 1$$

Por tanto:

$$0 \leq \left[1 - \left(\sum_i c_i x_{ij-1} * \sum_i c_i x_{i,j} \right) \right] * V_j \leq 1 \quad j = 2, \dots, N_{MAX} - 1 \quad (5.9)$$

Restricciones para contabilizar los cambio de modelo Opción 1.

Mediante estas restricciones se asegura que la variable de decisión Z_j tenga el valor de 1 cuando ocurre un cambio de modelo entre la posición j y la posición $j - PPL$, es decir entre la posición actual con el producto asignado y la misma posición pero con el producto asignado en el ciclo inmediato anterior. Se hace la evaluación a partir de la posición $j = PPL + 1$, para poder comparar con la posición $j = 1$. Se revisan todas las posiciones hasta N_{MAX} , el número máximo de posiciones.

$$n_g z_j \geq \sum_i g_i x_{ij} - \sum_i g_i x_{i,j-PPL} \quad j = PPL + 1, \dots, N_{MAX} \quad (5.10)$$

$$n_g z_j \geq \sum_i g_i x_{i,j-PPL} - \sum_i g_i x_{ij} \quad j = PPL + 1, \dots, N_{MAX} \quad (5.11)$$

De manera similar a las restricciones para contabilizar los cambios de color, las restricciones por cambio de modelo operan en conjunto. La restricción (5.10) funciona de la siguiente manera:

- 1) Se evalúa para todos los productos i si han sido asignados en la posición j . con lo que el valor de X_{ij} en dicha variable de decisión será 1 en alguno de los casos de i .
- 2) Multiplica el valor para cada caso de X_{ij} por el valor de g_i correspondiente. En un solo caso, aquel en el que el valor de X_{ij} es 1 se obtendrá un valor positivo.
- 3) Se hace la sumatoria de todos los casos i en la posición j
- 4) Se evalúa para todos los productos i si han sido asignados en la posición $j - PPL$. con lo que el valor de $X_{i,j-PPL}$ en dicha variable de decisión será 1 en alguno de los casos de i .
- 5) Multiplica el valor para cada caso de $X_{i,j-PPL}$ por el valor de g_i correspondiente. En un solo caso, aquel en el que el valor de $X_{i,j-PPL}$ es 1 se obtendrá un valor positivo.
- 6) Se hace la sumatoria de todos los casos i en la posición $j - PPL$
- 7) Se hace la diferencia entre la sumatoria de los casos en j menos la sumatoria de los casos en $j - PPL$
- 8) Se divide el resultado de la diferencia entre n_g con lo que se obtiene un valor igual a cero o distinto a cero.
- 9) Se restringe el valor de Z_j con una desigualdad que limita a dicha variable de decisión a que sea mayor o igual al valor obtenido en el punto anterior. Es decir Z_j será mayor o igual que cero o que el valor obtenido.

La restricción (5.11) funciona como complemento de la restricción (5.10). El procedimiento de cálculo del valor es en la práctica igual solo que con los términos de la diferencia invertidos. Es decir, la diferencia se calcula restando a la sumatoria de los casos en $j - PPL$ el valor de la sumatoria de los casos en j .

Como ambas restricciones se deben de cumplir en el modelo matemático, el resultado en la variable binaria Z_j solo podrá ser con valor 0 en el caso de que el modelo de producto sea el mismo en las posiciones j y $j - PPL$ y en el caso de que el modelo sea distinto entre las dos posiciones mencionadas el valor será 1.

A continuación se indican los pasos para la restricción (5.11) :

- 1) Se evalúa para todos los productos i si han sido asignados en la posición $j - PPL$. con lo que el valor de X_{ij-1} en dicha variable de decisión será 1 en alguno de los casos de i .
- 2) Multiplica el valor para cada caso de X_{ij-1} por el valor de g_i correspondiente. En un solo caso, aquel en el que el valor de X_{ij-1} es 1 se obtendrá un valor positivo.
- 3) Se hace la sumatoria de todos los casos i en la posición $j - PPL$
- 4) Se evalúa para todos los productos i si han sido asignados en la posición j . con lo que el valor de X_{ij} en dicha variable de decisión será 1 en alguno de los casos de i .
- 5) Multiplica el valor para cada caso de X_{ij} por el valor de g_i correspondiente. En un solo caso, aquel en el que el valor de X_{ij} es 1 se obtendrá un valor positivo.
- 6) Se hace la sumatoria de todos los casos i en la posición j
- 7) Se hace la diferencia entre la sumatoria de los casos en $j - PPL$ menos la sumatoria de los casos en j
- 8) Se divide el resultado de la diferencia entre n_g con lo que se obtiene un valor igual a cero o distinto a cero.
- 9) Se restringe el valor de Z_j con una desigualdad que limita a dicha variable de decisión a que sea mayor o igual al valor obtenido en el punto anterior. Es decir Z_j será mayor o igual que cero o que el valor obtenido.

Restricciones para asignar en la posición j un espacio vacío por cambio de modelo

El valor de Z_j indica que existe un cambio de geometría entre la posición j y $j-PPL$.

Cuando existe un espacio vacío, el valor de Z_j será igual a 1 en dos posiciones consecutivas, es decir en Z_j y en Z_{j+PPL} .

Al hacer la multiplicación del valor de Z_j por el valor de Z_{j+PPL} el valor de V_j será igual a 1 solo donde hay un espacio vacío, en cualquier otro caso será 0.

$$V_j \geq Z_j * Z_{j+PPL} \quad j = PPL+1, \dots, N_{MAX} - PPL \quad (5.12)$$

Restricciones para asegurar que cuando hay cambio de modelo entre la posición $j+PPL$ y $j-PPL$ el producto sea distinto en la posición j

Cuando existe un espacio vacío se cuentan dos cambios de modelo: uno al cambiar del modelo anterior al espacio vacío y otro del vacío al modelo nuevo. Que haya un solo cambio se debe restringir pues implica que no se asignó el espacio como vacío.

El valor de V_j dado por la restricción (5.12) se suma al producto de multiplicar Z_j por Z_{j-PPL} y se le suma el valor de V_j . El resultado obtenido solamente puede ser "0,1, 2".

El valor de "1" se obtiene cuando solo hay un cambio de modelo y se debe restringir.

Para eliminar el valor de 1 al término explicado se le resta 1.

El término anterior se eleva al cuadrado para tener solo los valores positivos y restringir al valor "1" toda la ecuación.

$$\left[(Z_j * Z_{j+PPL}) + ((Z_j * Z_{j-PPL}) + y_j - 1) \right]^2 = 1 \quad j = PPL+1, \dots, N_{MAX} - PPL \quad (5.13)$$

Restricciones para asegurar que en la posición j exista un espacio vacío y no un producto distinto cuando entre las posiciones j+PPL y j-PPL el producto sea distinto.

La ecuación (5.13) asegura que cuando haya un cambio de modelo entre los productos asignados en las posiciones j-PPL y j+PPL haya un espacio o un producto con modelo distinto en la posición j.

Esta restricción asegura que cuando haya un cambio de modelo se inserte un espacio vacío y no un producto con modelo distinto.

La forma de operar de esta restricción es comparar el color del producto j-PPL con el color del producto j+PPL, y al mismo tiempo el modelo del producto j-PPL con el del producto j.

Demostración de la restricción:

$$\text{Sea } A = \sum_i C_i X_{i,j-PPL} * \sum_i C_i X_{i,j+PPL}$$

$$\text{Sea } B = \sum_i C_i X_{i,j-PPL} * \sum_i C_i X_{i,j}$$

La ecuación siguiente asegura que la posición j sea igual a "0":

$$0 \leq [A - (A + B) + 1] V_j \leq 1$$

Simplificando se obtiene:

$$0 \leq [1 - B] V_j \leq 1$$

Por tanto:

$$0 \leq \left[1 - \left(\sum_i C_i X_{i,j-PPL} * \sum_i C_i X_{i,j} \right) \right] * V_j \leq 1 \quad j = PPL+1, \dots, N_{MAX} - 1 \quad (5.14)$$

Restricciones de integralidad

$$V_j, X_{ij}, y_j, z_j \in \{0,1\} \quad i = 1, \dots, n, \quad j = 1, \dots, N_{MAX} \quad (5.15)$$

Este grupo de restricciones aseguran que el valor de las variables binarias: V_j , X_{ij} , Y_j , Z_j para todos los tipos de producto y para todas las posiciones sea 0 ó 1.

5.1.2 MODELADO DEL PROBLEMA CBBSSP-CH OPCION 2

Se presenta una segunda opción de modelado. La modelación matemática es tridimensional. Al evaluar la posición de un producto determinado en la secuencia este se compara con el siguiente con base en los índices de: color, modelo y posición. Este modelo se formula como sigue:

5.1.2.1 NOMENCLATURA

Para poder modelar en forma tridimensional el problema del CBBSSPCH, presentar su código y algunas condicionantes se definen índices, parámetros y variables de decisión que son distintos a los de la Opción 1 presentada en la sección previa. Esta nomenclatura es consistente con la presentada por Ganguly y Laguna en (3). La definición se presenta a continuación:

Indices:

- i : Número del color del producto que se asigna (un producto está definido por un combinación de geometría y color).
- j : Número del modelo del producto que se asigna (un producto está definido por un combinación de geometría y color).
- k : Número de la posición que se asigna.
- g : Número de la geometría que se revisa.
- c : Número del color que se revisa

Parámetros:

- n : Número de productos (un producto está definido por un combinación de geometría y color).
- n_c : Número de colores.
- n_g : Número de geometrías.
- d_i : Demanda para el producto i , para $i = 1, \dots, n$
- N : Número de posiciones requeridas sin considerar huecos, donde

$$N = \sum_{i=1}^n d_i$$

- N_{MAX} : Número máximo de posiciones con huecos, donde $N_{MAX} = N * 2$ Si se considera el peor caso posible donde después de la asignación de un producto en una posición se asigna un espacio vacío.

PPL : Elementos por circuito o bucle

Variables de Decisión:

Se consideran los índices y variables presentados para definir adicionalmente el siguiente grupo de variables binarias de decisión, las cuáles permiten obtener información sobre la asignación de un producto específico a una posición determinada, si existe un cambio de color entre dos posiciones secuenciales o si existe un cambio de modelo o geometría en la misma posición al ejecutar un ciclo subsecuente en el proceso. Las variables se indican a continuación:

$x_{ijk} = \begin{cases} 1 \\ 0 \end{cases}$ 1 si el producto i es colocado en la posición j , 0 en cualquier otro caso.

$y_{i,k} = \begin{cases} 1 \\ 0 \end{cases}$ 1 si existe un cambio de color para el color " i " entre las posiciones $k-1$ y k , 0 en cualquier otro caso.

$z_{j,k} = \begin{cases} 1 \\ 0 \end{cases}$ 1 si existe un cambio de modelo para el modelo " j " entre las posiciones $k-1$ y k , 0 en cualquier otro caso.

Tomando en consideración los índices y parámetros indicados, así como las variables de decisión se procede a presentar la modelación matemática de la Opción 2 del problema básico CBBSSPCH.

5.1.2.2 MODELO DEL PROBLEMA CBBSSPCH OPCION 2

Función Objetivo del CBBSSPCH Opción 2.

Al igual que en la opción ya presentada, en este modelo los costes se determinan por la cantidad de espacios sin asignación de producto y estos están relacionados en forma directamente proporcional al número de cambios, sean de color o de geometría que son generados.

Sin perder generalidad, el modelo asume que cualquier cambio sea en color o en geometría genera un espacio vacío o hueco.

De acuerdo a lo indicado la expresión matemática de la función objetivo para el CBBSSPCH es:

Minimizar
$$\sum_i \sum_j \sum_k x_{i,j,k}$$
 Para toda $k > N$ (5.16)

Restricciones básicas del problema CBBSSPCH Opción 2.

Se debe optimizar la función objetivo cumpliendo las restricciones siguientes:

Restricciones de satisfacción de la demanda.

La demanda para cada tipo de producto (i,j) debe ser completada si se revisan todas las posiciones (k) existentes en el sistema. Esto se asegura mediante el siguiente grupo de restricciones:

$$\sum_k X_{i,j,k} = d_{i,j} \quad \text{Para } i = 1, \dots, n_c, \quad j = 1, \dots, n_g \quad (5.17)$$

Restricciones de asignación como máximo de un solo producto en cada posición disponible.

Cada posición k disponible en el sistema debe tener asignado como máximo un solo producto, lo cual se restringe asegurando que la sumatoria de todos los tipos de producto (i,j) en una posición específica k (variable de decisión $X_{i,j,k}$) sea menor o igual a 1. A diferencia del modelo sin huecos, la restricción no puede ser igual a 1 porque también existe la posibilidad de un espacio vacío, es decir sin asignación de producto. En cada posición k se hace la sumatoria para todos los productos (i,j).

$$\sum_i \sum_j x_{i,j,k} \leq 1 \quad j = 1, \dots, N_{MAX} \quad \text{Para } k=1..N_{MAX} \quad (5.18)$$

Restricciones para contabilizar los cambios de color.

Mediante estas restricciones se asegura que la variable de decisión $Y_{j,k}$ tenga el valor de 1 cuando ocurre un cambio de color entre la posición k y la posición k -1. Se hace la evaluación a partir de la posición k = 2, para poder comparar con la posición k = 1. Se revisan todas las posiciones hasta N_{MAX} , el número máximo de posiciones.

$$y_{i,k} \geq \sum_j x_{i,j,k} - \sum_j x_{i,j,k-1} \quad k = 2, \dots, N_{MAX}, \quad i=1, \dots, n_c. \quad (5.19)$$

$$y_{i,k} \geq \sum_j x_{i,j,k-1} - \sum_j x_{i,j,k} \quad k = 2, \dots, N_{MAX}, \quad i=1, \dots, n_c. \quad (5.20)$$

Estas dos restricciones operan en conjunto. La restricción (5.19) funciona de la siguiente manera:

- 1) Se evalúa para todos los productos (i,j) si han sido asignados en la posición k con lo que el valor de $X_{i,j,k}$ en dicha variable de decisión será 1 si alguno de los productos existe en esa posición k, o de cero en el caso de que no haya ninguna asignación.
- 2) Se hace la sumatoria en j de todos los los productos $X_{i,j,k}$ en la posición k.
- 3) Se evalúa para todos los productos (i,j) si han sido asignados en la posición k-1. con lo que el valor de $X_{i,j,k-1}$ en dicha variable de decisión será "1" en alguno de los casos o de "0".
- 4) Se hace la sumatoria en j de todos los casos (i,j) en la posición k-1
- 5) Se hace la diferencia entre la sumatoria de los casos en k menos la sumatoria de los casos en k -1. El resultado para cada color i de los n_c existentes puede ser igual a (-1, 0, 1)
- 6) Se restringe el valor de $Y_{i,k}$ al ser una variable binomial. En este caso con una desigualdad que limita a dicha variable de decisión a que

sea mayor o igual al valor obtenido en el punto anterior. Es decir $Y_{i,k}$ será mayor o igual que el valor obtenido.

- 7) La restricción (5.20) funciona como complemento de la restricción (5.19). El cálculo del valor es en la práctica igual con los términos de la diferencia invertidos. Es decir, la diferencia se calcula restando a la sumatoria de los casos en $k-1$ el valor de la sumatoria de los casos en k .
- 8) Como ambas restricciones se deben de cumplir en el modelo, el resultado en la variable binaria $Y_{i,k}$ solo podrá ser con valor 0 en el caso de que el color sea el mismo en las posiciones k y $k-1$ y en el caso de que el color sea distinto entre las dos posiciones mencionadas el valor será 1. Esto se hace para cada valor de i , es decir para cada tipo de color.

A continuación se indican los pasos para la restricción (5.20) :

- 1) Se evalúa para todos los productos (i,j) si han sido asignados en la posición $k-1$ con lo que el valor de $X_{i,j,k}$ en dicha variable de decisión será 1 si alguno de los productos existe en esa posición k , o de cero en el caso de que no haya ninguna asignación.
- 2) Se hace la sumatoria en j de todos los los productos $X_{i,j,k}$ en la posición $k-1$.
- 3) Se evalúa para todos los productos (i,j) si han sido asignados en la posición k . con lo que el valor de $X_{i,j,k-1}$ en dicha variable de decisión será 1 en alguno de los casos, o de cero.
- 4) Se hace la sumatoria en j de todos los casos (i,j) en la posición k .
- 5) Se hace la diferencia entre la sumatoria de los casos en $k-1$ menos la sumatoria de los casos en k . El resultado para cada color i de los n_c existentes puede ser igual a $(-1, 0, 1)$
- 6) Se restringe el valor de $Y_{i,k}$ al ser una variable binomial. En este caso con una desigualdad que limita a dicha variable de decisión a que sea mayor o igual al valor obtenido en el punto anterior. Es decir $Y_{i,k}$ será mayor o igual que el valor obtenido.

Restricciones para asignar en la posición k un espacio vacío por cambio de color

El valor de $Y_{j,k}$ indica que existe un cambio de color para el color “ i ” entre las posiciones k y $k-1$.

Si entre las posiciones k y $k-1$ se cambia de un color a otro, es decir en ambas posiciones el valor de $X_{i,j,k}$ es 1 para ambas posiciones pero el color i es distinto. Se registrarían dos cambios: el color que deja de ser vigente cambia a “0” y el color que se asigna cambia de “0” a “1”.

Se busca que entre dos colores distintos en posiciones secuenciales exista un espacio vacío. Esto se logra limitando la suma de los valores de $Y_{i,k}$ haciéndola menor o igual a 1. Un valor de “0” indica que el entre las posiciones k y $k-1$ el no existe cambio de colore. Un valor de “1” indica que una de las posiciones tiene un espacio vacío. Si la sumatoria fuese “2” indicaría que

ambas posiciones tienen un producto asignado con color "i" distinto, lo cual se requiere limitar en este modelo.

$$\sum_i y_{i,k} \leq 1 \quad \text{Para } k = 2, \dots, N_{MAX} \quad (5.21)$$

Restricciones para contabilizar los cambios de modelo.

Mediante estas restricciones se asegura que la variable de decisión $Z_{j,k}$ tenga el valor de 1 cuando ocurre un cambio de color entre la posición k y la posición $k - PPL$. Se hace la evaluación a partir de la posición $k = PPL + 1$, para poder comparar con la posición $k = 1$. Se revisan todas las posiciones hasta N_{MAX} , el número máximo de posiciones.

$$z_{j,k} \geq \sum_i x_{i,j,k} - \sum_i x_{i,j,k-PPL} \quad \text{Para } k = PPL + 1, \dots, N_{MAX}, j = 1, \dots, n_g. \quad (5.22)$$

$$z_{i,k} \geq \sum_i x_{i,j,k-PPL} - \sum_i x_{i,j,k} \quad \text{Para } k = PPL + 1, \dots, N_{MAX}, i = 1, \dots, n_g. \quad (5.23)$$

Estas dos restricciones operan en conjunto. La restricción (5.22) funciona de la siguiente manera:

- 1) Se evalúa para todos los productos (i,j) si han sido asignados en la posición k con lo que el valor de $X_{i,j,k}$ en dicha variable de decisión será 1 si alguno de los productos existe en esa posición k , o de cero en el caso de que no haya ninguna asignación.
- 2) Se hace la sumatoria en i de todos los productos $X_{i,j,k}$ en la posición k .
- 3) Se evalúa para todos los productos (i,j) si han sido asignados en la posición $k - PPL$, con lo que el valor de $X_{i,j,k-PPL}$ en dicha variable de decisión será "1" en alguno de los casos o de "0".
- 4) Se hace la sumatoria en i de todos los casos (i,j) en la posición $k - PPL$.
- 5) Se hace la diferencia entre la sumatoria de los casos en k menos la sumatoria de los casos en $k - PPL$. El resultado para cada modelo j de los n_g existentes puede ser igual a (-1, 0, 1)
- 6) Se restringe el valor de $Z_{j,k}$ al ser una variable binomial. En este caso con una desigualdad que limita a dicha variable de decisión a que sea mayor o igual al valor obtenido en el punto anterior. Es decir $Z_{j,k}$ será mayor o igual que el valor obtenido.
- 7) La restricción (5.23) funciona como complemento de la restricción (5.22). El cálculo del valor es en la práctica igual con los términos de la diferencia invertidos. Es decir, la diferencia se calcula restando a la sumatoria de los casos en $k - PPL$ el valor de la sumatoria de los casos en k .
- 8) Como ambas restricciones se deben de cumplir en el modelo, el resultado en la variable binaria $Z_{j,k}$ solo podrá ser con valor 0 en el caso de que el color sea el mismo en las posiciones k y $k - PPL$ y en el caso de que el modelo sea distinto entre las dos posiciones

mencionadas el valor será 1. Esto se hace para cada valor de j, es decir para cada tipo de modelo.

A continuación se indican los pasos para la restricción (5.23):

- 1) Se evalúa para todos los productos (i,j) si han sido asignados en la posición k-PPL con lo que el valor de $X_{i,j,k-PPL}$ en dicha variable de decisión será 1 si alguno de los productos existe en esa posición k-PPL, o de cero en el caso de que no haya ninguna asignación.
- 2) Se hace la sumatoria en i de todos los los productos $X_{i,j,k}$ en la posición k-PPL.
- 3) Se evalúa para todos los productos (i,j) si han sido asignados en la posición k. con lo que el valor de $X_{i,j,k-1}$ en dicha variable de decisión será 1 en alguno de los casos, o de cero.
- 4) Se hace la sumatoria en i de todos los casos (i,j) en la posición k.
- 5) Se hace la diferencia entre la sumatoria de los casos en k-PPL menos la sumatoria de los casos en k. El resultado para cada modelo j de los n_g existentes puede ser igual a (-1, 0 , 1)
- 6) Se restringe el valor de $Z_{j,k}$ al ser una variable binomial. En este caso con una desigualdad que limita a dicha variable de decisión a que sea mayor o igual al valor obtenido en el punto anterior. Es decir $Z_{j,k}$ será mayor o igual que el valor obtenido.

Restricciones para asignar en la posición k un espacio vacío por cambio de modelo

El valor de $Z_{j,k}$ indica que existe un cambio de modelo para el modelo "j" entre las posiciones k y k-PPL.

Si entre las posiciones k y k-PPL se cambia de un modelo a otro, es decir, en ambas posiciones el valor de $X_{i,j,k}$ es 1 pero el modelo "j" es distinto entonces se registrarían dos cambios: el modelo que deja de ser vigente cambia a "0" y el modelo que se asigna cambia de "0" a "1".

Se busca que entre dos modelos distintos en la misma posición del siguiente ciclo exista un espacio vacío. Esto se logra limitando la suma de los valores de $Z_{j,k}$ haciéndola menor o igual a 1. Un valor de "0" indica que el entre las posiciones k y k-PPL no existe cambio de modelo. Un valor de "1" indica que una de las posiciones tiene un espacio vacío. Si la sumatoria fuese "2" indicaría que ambas posiciones tienen un producto asignado con modelo "j" distinto, lo cual se requiere limitar en este modelo matemático.

$$\sum_j z_{j,k} \leq 1 \quad \text{Para } k = PPL+1, \dots, N_{MAX} \quad (5.24)$$

Restricciones de integralidad

$$x_{i,j,k}, y_{i,k}, z_{j,k} \in \{0,1\} \quad i = 1, \dots, n, \quad j = 1, \dots, N_{MAX} \quad (5.25)$$

Este grupo de restricciones aseguran que el valor de las variables binarias: $X_{i,j,k}$, $Y_{i,k}$, $Z_{j,k}$ para todos los tipos de producto y para todas las posiciones sea 0 ó 1.

5.1.3 MODELO GENERAL DEL PROBLEMA CBBSSP-CH

5.1.3.1 Modelo general del problema CBBSSP Opción 1

$$\text{Minimizar } \sum_{j>1} V_j \quad (5.1)$$

Sujeta a:

$$\sum_j X_{ij} = d_i \quad i = 1, \dots, n \quad (5.2)$$

$$\sum_i X_{ij} \leq 1 \quad j = 1, \dots, N_{MAX} \quad (5.3)$$

$$\sum_i X_{ij} + \sum_i X_{i,j-1} > 0 \quad j = 2, \dots, N_{MAX} \quad (5.4)$$

$$n_c y_j \geq \sum_i c_i x_{ij} - \sum_i c_i x_{i,j-1} \quad j = 2, \dots, N_{MAX} \quad (5.5)$$

$$n_c y_j \geq \sum_i c_i x_{i,j-1} - \sum_i c_i x_{ij} \quad j = 2, \dots, N_{MAX} \quad (5.6)$$

$$V_j \geq y_j * y_{j+1} \quad j = 2, \dots, N_{MAX} - 1 \quad (5.7)$$

$$\left[(y_j * y_{j+1}) + ((y_j * y_{j-1}) + y_j - 1) \right]^2 = 1 \quad j = 3, \dots, N_{MAX} - 1 \quad (5.8)$$

$$0 \leq \left[1 - \left(\sum_i c_i x_{ij-1} * \sum_i c_i x_{i,j} \right) \right] * V_j \leq 1 \quad j = 2, \dots, N_{MAX} - 1 \quad (5.9)$$

$$n_g z_j \geq \sum_i g_i x_{ij} - \sum_i g_i x_{i,j-PPL} \quad j = PPL + 1, \dots, N_{MAX} \quad (5.10)$$

$$n_g z_j \geq \sum_i g_i x_{i,j-PPL} - \sum_i g_i x_{ij} \quad j = PPL + 1, \dots, N_{MAX} \quad (5.11)$$

$$V_j \geq Z_j * Z_{j+PPL} \quad j = PPL+1, \dots, N_{MAX} - PPL \quad (5.12)$$

$$\left[(Z_j * Z_{j+PPL}) + ((Z_j * Z_{j-PPL}) + y_j - 1) \right]^2 = 1 \quad j = PPL+1, \dots, N_{MAX} - PPL \quad (5.13)$$

$$0 \leq \left[1 - \left(\sum_i c_i x_{ij-PPL} * \sum_i c_i x_{i,j} \right) \right] * V_j \leq 1 \quad j = PPL+1, \dots, N_{MAX} - 1 \quad (5.14)$$

$$V_j, X_{ij}, y_j, z_j \in \{0,1\} \quad i = 1, \dots, n, \quad j = 1, \dots, N_{MAX} \quad (5.15)$$

5.1.3.2 Modelo general del problema CBBSSP Opción 2

$$\text{Minimizar } \sum_i \sum_j \sum_k x_{i,j,k} \quad \text{Para toda } k > N \quad (5.16)$$

Sujeta a:

$$\sum_k X_{i,j,k} = d_{i,j} \quad \text{Para } i = 1, \dots, n_c, j = 1, \dots, n_g \quad (5.17)$$

$$\sum_i \sum_j x_{i,j,k} \leq 1 \quad j = 1, \dots, N_{MAX} \quad \text{Para } k=1..N_{MAX} \quad (5.18)$$

$$y_{i,k} \geq \sum_j x_{i,j,k} - \sum_j x_{i,j,k-1} \quad k = 2, \dots, N_{MAX}, i=1, \dots, n_c. \quad (5.19)$$

$$y_{i,k} \geq \sum_j x_{i,j,k-1} - \sum_j x_{i,j,k} \quad k = 2, \dots, N_{MAX}, i=1, \dots, n_c. \quad (5.20)$$

$$\sum_i y_{i,k} \leq 1 \quad \text{Para } k = 2, \dots, N_{MAX} \quad (5.21)$$

$$z_{j,k} \geq \sum_i x_{i,j,k} - \sum_i x_{i,j,k-PPL} \quad \text{Para } k = PPL+1, \dots, N_{MAX}, j=1, \dots, n_g. \quad (5.22)$$

$$z_{i,k} \geq \sum_i x_{i,j,k-PPL} - \sum_i x_{i,j,k} \quad \text{Para } k = PPL+1, \dots, N_{MAX}, i=1, \dots, n_g. \quad (5.23)$$

$$\sum_j z_{j,k} \leq 1 \quad \text{Para } k = PPL+1, \dots, N_{MAX} \quad (5.24)$$

$$x_{i,j,k}, y_{i,k}, z_{j,k} \in \{0,1\} \quad i = 1, \dots, n, j = 1, \dots, N_{MAX} \quad (5.25)$$

5.1.4 LIMITES DEL MODELO Y ANALISIS DEL ENTORNO.

Antes de revisar los posibles métodos de solución se analiza la estructura del problema con el objetivo de explorar nuevas ideas y determinar los límites al número de cambios por tiempo de preparación por los dos tipos (horizontal y vertical) mencionados en la definición del problema.

Si se considera que el costo del tiempo de preparación es constante y no depende del cambio específico de colores o de geometrías involucradas en cada caso, entonces los límites inferiores en el número de cambios por tiempo de preparación para cada tipo de cambio (LB_c, LB_g) pueden ser utilizados para calcular un límite inferior en el costo total (LB), referido como la sumatoria de los costos obtenidos por cada uno de los dos tipo de cambio multiplicado por sendos límites inferiores:

$$LB = hLB_c + vLB_g \quad (5.26)$$

Para los problemas donde se requiera considerar necesariamente costos variables de tiempo de preparación, el límite inferior se puede calcular utilizando el mínimo coste horizontal y el mínimo vertical.

Parámetros:

LB_c : Límite inferior en el número de cambios por color, valor por defecto 0

LB_g : Límite inferior en el número de cambios por geometría, valor por defecto 0

LB : Límite inferior en el valor de la función objetivo

UB_c : Límite superior en el número de cambios por color, valor por defecto 0

UB_g : Límite superior en el número de cambios por geometría, valor por defecto 0

UB : Límite superior en el valor de la función objetivo

Límite inferior en el número de cambios por tiempo de preparación horizontal

Si los tiempos de preparación se ignoran, se obtiene un límite inferior por color dado por la siguiente ecuación:

$$LB_c = [n_c] - 1 \quad (5.27)$$

En este caso, en la ausencia de tiempos de preparación vertical, la forma óptima de organizar los elementos se obtiene al agrupar todos los productos del mismo color en cada caso y asignarlos secuencialmente al proceso, siempre y cuando todas las partes con el mismo color sean procesadas antes de continuar con otro color.

Límite inferior en el número de cambios por tiempo de preparación vertical

Si se deja de lado el costo de tiempo de preparación horizontal por cambio de color se tiene solamente el costo por tiempo de preparación vertical por cambio de modelo. Una forma de conceptualizar la solución puede ser al considerar una matriz de dos dimensiones donde el número de columnas equivale al número de posiciones por ciclo (PPL) y el número de filas es igual a N/PPL .

La demanda relacionada con cada geometría se puede calcular haciendo la sumatoria de la demanda para cada producto i restringida por tipo de geometría. Lo cuál se puede conceptualizar agregando un subíndice al parámetro d_i de tal manera que se define un nuevo parámetro: $d_{(g_i)}$ como demanda para la geometría el producto " g " (la sumatoria e todos los productos i que tengan la geometría g . La demanda asociada con cada geometría g es dada por la ecuación:

$$d_{(g_*)} = \sum_i d_{(g_i)} \quad g = 1, \dots, n_g \quad (5.28)$$

Se puede agregar un parámetro nuevo: m_g el cual consiste en el cálculo del residuo resultante de dividir la demanda por geometría: $d_{(g_i)}$ entre el número de filas dado por N/PPL .

$$m_g = \text{mod}\left(\frac{d_{g_i}}{N/PPL}\right) \quad g = 1, \dots, n_g \quad (5.29)$$

Se puede demostrar que en el caso de que $m_g = 0$ para toda g entonces el $LB_g = 0$. Esto significa que cada columna incluye solamente una geometría y el número de veces que ocurre un tiempo de preparación es cero. Sin embargo al menos en una ocasión ocurre un tiempo de preparación cuando $m_g \neq 0$ para al menos una $g \in n_g$. Esto significa que cuando los costos de preparación horizontal se ignoran, el problema de minimizar los costos de preparación vertical se puede reducir en otro más simple de la siguiente manera:

Para cada geometría $g \in n_g$, $\left\lceil \frac{d_{g_i}}{N/PPL} \right\rceil$ indica el número de columnas que son asignadas totalmente a cada geometría sin incurrir en cambios por tiempos de preparación verticales. El problema reducido consiste entonces en asignar m_g elementos para cada $g \in n_g$ en una matriz con el mismo número de renglones como en el problema original (N/PPL) y un número reducido de columnas. El número reducido de columnas, o número reducido de espacios por ciclo (RPPL) es dado por:

$$RPPL = PPL - \sum_g \left\lceil \frac{d_{g_i}}{N/PPL} \right\rceil = \frac{\sum_g m_g}{N/PPL} \quad (5.30)$$

Dado que en este caso no se está considerando el coste de tiempos de preparación por cambios horizontales, las variables y restricciones relacionadas con el cambio de color pueden ser omitidas de la formulación (Restricciones 4.4 y 4.5 para la Opción 1 y 4.9 en la Opción 2) De esta manera ignorando los costes horizontales y considerando solamente los costes por cambio de geometría el problema de optimización se reduce tanto en escala como en complejidad. El número mínimo de cambios por geometría se obtiene resolviendo este modelo reducido y es un límite inferior realista en el número de cambios por geometría (LB_g). De acuerdo al n_g y a la ecuación (5.16), se sabe que $LB_g \leq n_g - 1$

Límite superior en el número de cambios por tiempo de preparación horizontal.

El límite superior en el número de cambios por tiempo de preparación horizontal ocurre en el peor de los escenarios cuando entre cada posición j ocurre un cambio lo cual incrementa el valor del número de cambios en cada ocasión. De acuerdo a esto el máximo dado por el número de cambios de color se obtiene con la ecuación:

$$h \sum_{j>1} y_j = N - 1 \quad (5.31)$$

En el caso de que el valor del coste por cambio sea $h=1$ el coste por cambios por tiempo de preparación horizontal es solo el número de cambios máximo igual al número de posiciones menos uno.

Límite superior en el número de cambios por tiempo de preparación vertical.

El límite superior en el número de cambios por tiempo de preparación vertical se tiene también en el peor de los casos cuando entre cada posición j y $j+1$ ocurre un cambio vertical lo cual hace necesario hacer un cambio de jig en cada ocasión. De acuerdo a esto el máximo dado por el número de cambios de geometría se obtiene con la ecuación:

$$v \sum_{j>1} z_j = N - PPL \quad (5.32)$$

En el caso de que el valor del coste por cambio sea $v=1$ el coste por cambios por tiempo de preparación vertical es solo el número de cambios máximo igual al número de posiciones menos el número de elementos por circuito o bucle (PPL).

Espacio de solución del modelo general

Es posible obtener un límite inferior en el costo total utilizando para el cálculo los valores obtenidos de número de tiempos de preparación por cambio de color o cambio de modelo por separado, sin embargo ejemplos simples pueden demostrar que un método de solución que se enfoque en cada tipo de tiempo de preparación en forma independiente puede resultar en soluciones que no son solo subóptimas sino que son significativamente inferiores que aquellas que se pueden encontrar al considerar ambos tiempos de preparación en conjunto. Esto significa que un procedimiento que minimice el número de cambios de un tipo de tiempo de preparación y que a continuación minimice el número de cambios del segundo tipo no necesariamente va a optimizar el objetivo, incluso si se consideran solamente aquellas soluciones que no incrementen el número de cambios del primer tipo al intentar minimizar los del segundo.

Una manera simple de representar una solución al problema consiste en definir un vector Q de tamaño N (Número de posiciones), de tal manera que $Q(i)$ es el índice del producto en la posición i . La demanda del producto i es dado por el parámetro d_i . Si al revisar una instancia específica los valores de d_i son de 1 ($d_i = 1$ para toda i , entonces $N = [n]$), es decir, el número de posiciones del problema es igual al número de productos, y el espacio de solución consiste en $N!$ soluciones. La fórmula en general para determinar el espacio de soluciones se representa por la siguiente multiplicatoria:

$$\prod_{i=1}^{i=[n]} C_{d_i}^{N - \sum_{i'=0}^{i'-i-1} d_{i'}} \quad (5.33)$$

Considerando los costes de cambio horizontal y cambio vertical como unitarios, esto es ($h = v = 1$) entonces el coste total asociado con una solución dada es el mismo que se obtiene como el número total de cambios necesarios al establecer una solución.

El límite superior al número de cambios total se obtiene al sumar los límites superiores de cada tipo de tiempo de preparación y se indica en la siguiente ecuación:

$$h \sum_{j>1} y_j + v \sum_{j>PPL} z_j = 2N - PPL - 1 \quad (5.34)$$

CONCLUSIONES Y COMENTARIOS DEL MODELO MATEMATICO:

El primer modelo (Opción 1) presenta mayor complejidad que el segundo modelo (Opción 2) debido a que no es lineal (Programación No Lineal) y contiene en una de las restricciones términos de segundo orden. El segundo modelo (Opción 2) es del tipo MIP (Programación Lineal Entera Mixta), la complejidad es menor dado que se genera mayor diversidad en los índices iniciales dentro de la variable de decisión $X_{i,j,k}$ esto hace que la implementación sea más sencilla que en el primer modelo. Además el número de variables binarias es menor, en este caso son 3 respecto a las 4 de la Opción 1. Lo mismo sucede con el número de restricciones pues en el primer modelo son 14 mientras que en el segundo son 9. Existen múltiples programas de software a través de los cuáles se puede implementar. Para efecto de este modelo, de forma consecuente con el procedimiento utilizado en el Capítulo 4, se ha elegido también utilizar tanto el software Gusek como el Gurobi.

5.2 PROCEDIMIENTOS EXACTOS PARA LA RESOLUCION DEL PROBLEMA CBBSSP CON HUECOS

INTRODUCCION

Los modelos de programación matemática con huecos se desarrollaron en el apartado anterior de acuerdo a la problemática de las líneas de pintura en la industria automotriz descrita en el capítulo 3. Fue necesario definir las hipótesis básicas de trabajo establecer la función objetivo y justificar cada una de las restricciones presentadas. Con lo anterior se concluye que el objetivo del problema con huecos estudiado en este capítulo de la Tesis consiste en definir la secuenciación de tareas en la industria automotriz con lotes cíclicos satisfaciendo el criterio de minimización bi-objetivo del número de veces que ocurren los *setups* con tiempos de preparación dependientes e independientes de la secuencia asumiendo que cuando existen estos cambios se debe dejar un espacio vacío en la secuenciación. Se han presentado dos opciones de modelado matemático para la resolución del problema.

En este apartado se desarrolla la implementación de los modelos matemáticos utilizando herramientas de solución exacta.

5.2.1 METODOLOGIA PROCEDIMIENTOS EXACTOS PROBLEMA CON HUECOS.

En la siguiente sección se hace la implementación de uno de los modelos matemáticos presentados en el apartado anterior. Dado que en el capítulo 4 se ha mostrado para el caso del problema sin huecos la manera de hacer una operacionalización de este tipo de modelos matemáticos en este caso se obviará y se asume que al ejecutar los experimentos esta se hace en forma intrínseca al operar las restricciones sobre la función objetivo. En la siguiente sección se explica el desarrollo del modelo en el software Gusek utilizando el lenguaje que maneja por defecto: GLPK, este permite generar archivos con distintas extensiones tipo .dat, .mod, .out, .mps, .lp. La ventaja de esto es poder utilizar como herramienta de resolución a distintos tipos de software. Para el caso de esta tesis se exportará el código de programación a través de la extensión (.mps) para utilizar la potencia del *solver* de Gurobi.

5.2.2 IMPLEMENTACION DE LOS MODELOS PARA RESOLVER CON GUSEK – GUROBI.

En la sección de ANEXOS se encuentra un ejemplo del desarrollo del código en GUSEK-GUROBI.

5.2.3 RESULTADOS DE LA EXPERIMENTACION DEL PROBLEMA CBBSSP CON HUECOS EN GUSEK-GUROBI.

Para definir el grupo de experimentos utilizando el modelo matemático del problema CBBSSP-CH fue necesario hacer varias pruebas. Inicialmente se intentó realizar la experimentación con base en los mismos casos que se revisaron las heurísticas desarrolladas en el capítulo 4 para resolver el problema CBBSSP sin huecos. Sin embargo los valores de las variables independientes como: número de modelos, número de colores, tamaño de la demanda, número de bloques eran demasiado grandes. Esto ocasionaba que el tiempo de resolución fuese muy grande y no se lograra encontrar el óptimo en un tiempo razonable. Para encontrarla forma en que la complejidad de los casos empieza a incrementarse se decidió tomar los valores más pequeño posibles. A cada una de las variables independientes mencionadas se les asignó un valor muy bajo, y al generar experimentos con ellos se pudo observar cómo se incrementa el tiempo de resolución.

5.2.3.1 Definición de los casos analizados en los experimentos.

Con base en el punto anterior se definieron como experimentos la combinatoria de las variables independientes con los siguientes valores:

- 5) Demanda: 10, 20, 40
- 6) Número de Bloques: 5, 10
- 7) Número de Modelos: 2, 3, 4, 5

8) Número de Colores: 2, 3, 4, 5

De esta manera se generaron 5 Tablas de datos al combinar los casos de Demanda y Bloques:

6) Demanda 10 Bloques 5

7) Demanda 20 Bloques 5

8) Demanda 20 Bloques 10

9) Demanda 40 Bloques 5

10) Demanda 40 Bloques 10

Es importante aclarar que el caso de Demanda 10 Bloques 10, que sería la única combinatoria faltante para poder combinar todas las demandas y bloques, no tiene sentido, pues al generar los experimentos para todos los casos de combinación de modelos y colores se tendría solo un ciclo, es decir no se podría evaluar ningún cambio en vertical.

El número de experimentos totales para los procedimientos exactos se general al multiplicar las 5 tablas mencionadas por la combinatoria de casos de modelos y colores. Es decir 4 tipos de modelo multiplicado por 4 tipos de colores generan 16 casos para cada tabla. Con esto el número total de experimentos es de 16×5 . En total se hicieron 80 lanzamientos de experimentos.

De manera similar a la presentada en el Capítulo 4. Se definió como límite el tiempo de 600 segundos (10 minutos) de esta manera si no se obtiene el óptimo en ese tiempo, el experimento se detiene y se genera un informe en el que se indica el caso.

Cada uno de los experimentos se realizó mediante el procedimiento descrito a continuación:

Se generó un código de programación en GLPK para cada uno de los 80 casos. GUSEK permite en el módulo "Tools" la construcción o traducción del modelo que se ha escrito en GLPK al lenguaje MPS. El código en MPS puede ser interpretado sin problema por el *solver* de GUROBI. Se ha elegido el *solver* de GUROBI dado que es más eficiente y potente que el *solver* que está integrado en el GUSEK. El tiempo de solución es menor para los casos simples y con esto se puede encontrar la forma en la que crece el nivel de complejidad en la solución del problema a medida que los valores de las variables independientes crecen en las instancias. Se busca determinar si el problema es *NP_hard*. En la siguiente sección se muestran los resultados de los experimentos.

5.2.3.2 Resultados de los experimentos.

Demanda:		10							
Bloques	Mod	Col	Tpo. (Seg)	F. Obj	Best Bound	Nodos	Iteraciones	Procesadores	GAP
5	2	2	0.12	2	2	546	6389	4	0.00%
5	2	3	0.55	2	2	2651	44052	4	0.00%
5	2	4	0.5	3	3	1839	36310	4	0.00%
5	2	5	1.14	4	4	4573	92019	4	0.00%
5	3	2	0.36	2	2	2353	31860	4	0.00%
5	3	3	1.75	3	3	9763	164167	4	0.00%
5	3	4	1.06	3	3	2394	62799	4	0.00%
5	3	5	10.19	4	4	21552	829598	4	0.00%
5	4	2	0.72	2	2	3917	60257	4	0.00%
5	4	3	0.72	2	2	1468	37168	4	0.00%
5	4	4	1.73	3	3	4838	133146	4	0.00%
5	4	5	21.06	5	5	37085	1702155	4	0.00%
5	5	2	0.22	2	2	330	6317	4	0.00%
5	5	3	1.03	2	2	2703	65547	4	0.00%
5	5	4	18.91	4	4	21402	1218439	4	0.00%
5	5	5	6.44	4	4	12563	442397	4	0.00%

Tabla 17: Resultados Procedimientos Exactos Con Huecos con Demanda 10 Bloques 5

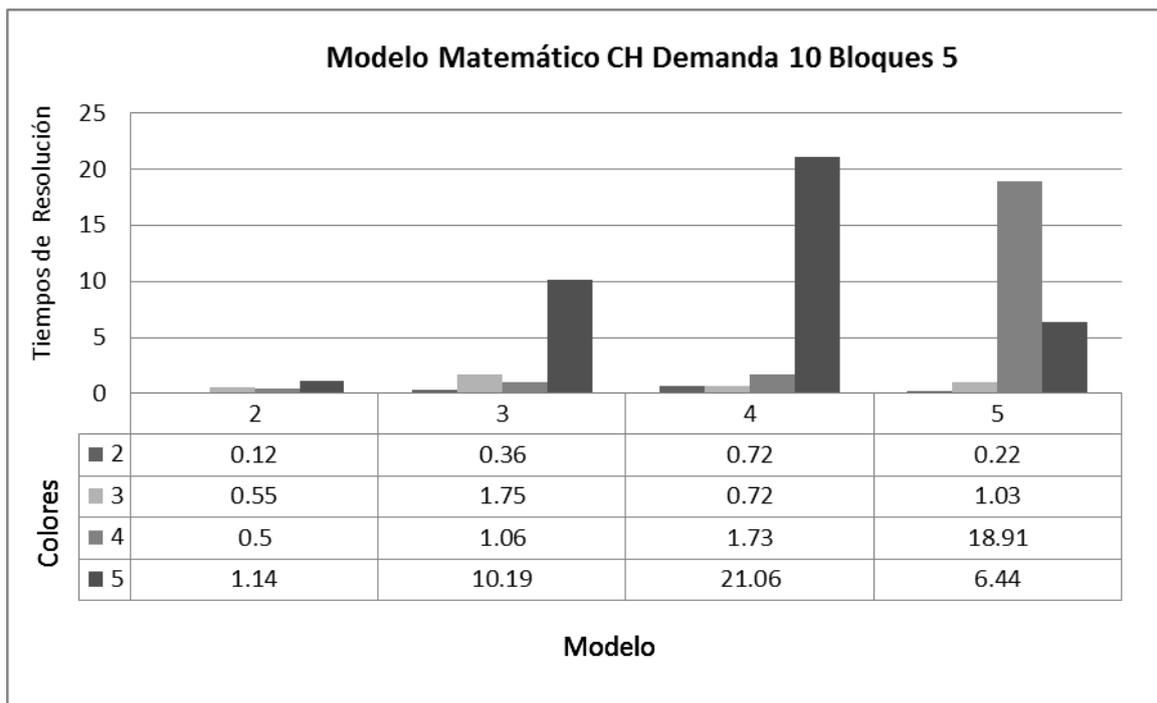


Figura 32: Comparativo del Tiempo de Optimización CBBSSP-CH Dem 10 Bloques 5

Demanda:		20							
Bloques	Mod Col	Tpo. (Seg)	F. Obj	Best Bound	Nodos	Iteraciones	Procesadores	GAP	
5	2 2	2.48	4	4	13923	256980	4	0.00%	
5	2 3	5.62	2	2	11367	409092	4	0.00%	
5	2 4	219.44	4	4	231037	17256638	4	0.00%	
5	2 5	512.95	5	5	584769	38323985	4	0.00%	
5	3 2	6.55	1	1	9777	399982	4	0.00%	
5	3 3	107.79	4	4	107884	8223361	4	0.00%	
5	3 4	600	4	3	594402	43906021	4	25.00%	
5	3 5	600	6	2	444089	36321413	4	66.67%	
5	4 2	27.54	3	3	33885	1924053	4	0.00%	
5	4 3	504.56	5	5	420824	35136369	4	0.00%	
5	4 4	328.77	3	3	187542	18234866	4	0.00%	
5	4 5	600	8	3	462424	32999067	4	62.50%	
5	5 2	59.96	3	3	60269	4563819	4	0.00%	
5	5 3	332.59	3	3	173729	19236718	4	0.00%	
5	5 4	600	5	2	372506	33272554	4	60.00%	
5	5 5	600	7	2	226006	21595739	4	71.43%	

Tabla 18: Resultados Procedimientos Exactos Con Huecos con Demanda 20 Bloques 5

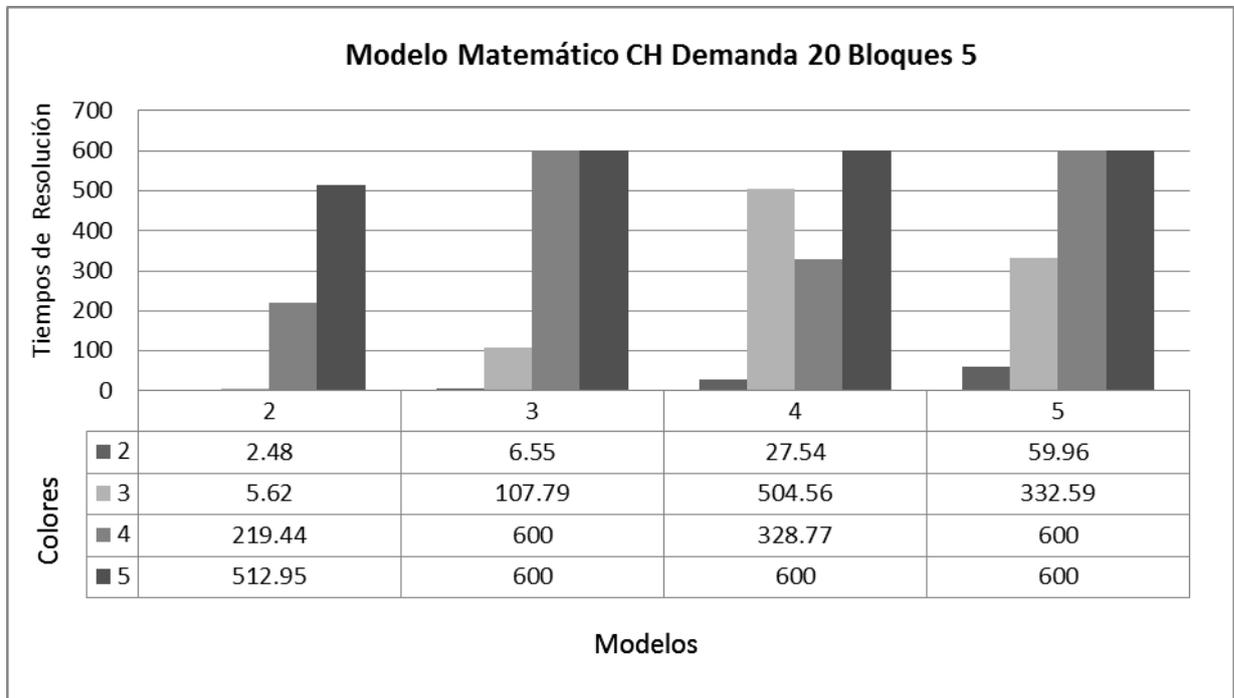


Figura 33: Comparativo del Tiempo de Optimización CBBSSP-CH Dem 20 Bloques 5

Demanda:		20						
Bloques	Mod Col	Tpo. (Seg)	F. Obj	Best Bound	Nodos	Iteraciones	Procesadores	GAP
10	2 2	0.41	2	2	1398	30151	4	0.00%
10	2 3	24.76	2	2	35604	1862164	4	0.00%
10	2 4	55.6	3	3	57450	4052581	4	0.00%
10	2 5	418.51	4	4	402587	28855318	4	0.00%
10	3 2	1.98	1	1	3921	108911	4	0.00%
10	3 3	66.52	3	3	78858	5272764	4	0.00%
10	3 4	494.2	3	3	398721	31757392	4	0.00%
10	3 5	600	4	2	415901	35630989	4	50.00%
10	4 2	115.33	3	3	274786	11347424	4	0.00%
10	4 3	144	3	3	111068	8056922	4	0.00%
10	4 4	600	3	1	466895	39226649	4	66.67%
10	4 5	600	5	2	329710	33517031	4	60.00%
10	5 2	600	3	2	1268520	48510454	4	33.33%
10	5 3	600	2	1	583127	38056154	4	50.00%
10	5 4	600	4	1	415524	31044069	4	75.00%
10	5 5	600	5	3	393368	34308830	4	40.00%

Tabla 19: Resultados Procedimientos Exactos Con Huecos con Demanda 20 Bloques 10

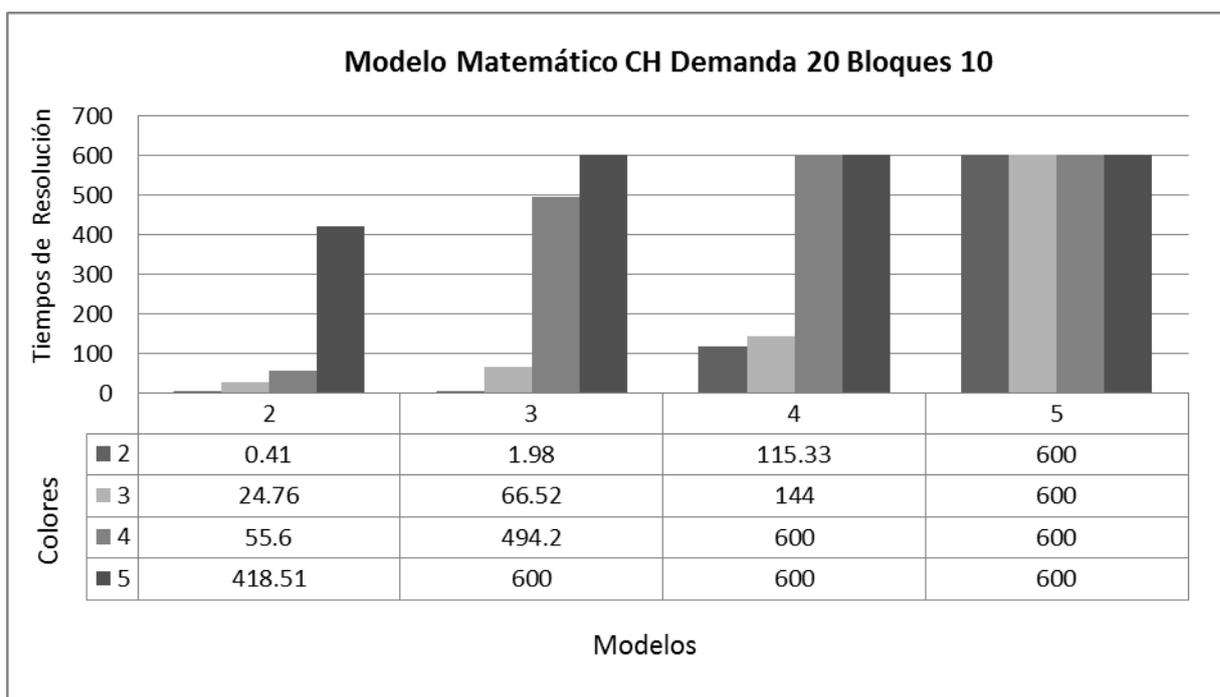


Figura 34: Comparativo del Tiempo de Optimización CBBSSP-CH Dem 20 Bloques 10

Demanda:								
Bloques	Mod Col	Tpo. (Seg)	F. Obj	Best Bound	Nodos	Iteraciones	Procesadores	GAP
5	2 2	44.3	2	2	21303	957328	4	0.00%
5	2 3	600	4	1	442674	38482617	4	75.00%
5	2 4	600	6	1	359715	29586811	4	83.33%
5	2 5	600	6	0	207299	25475587	4	100.00%
5	3 2	600	3	2	454874	41033534	4	33.33%
5	3 3	600	6	0	163735	21830615	4	100.00%
5	3 4	600	6	0	169073	20829445	4	100.00%
5	3 5	600	12	0	159317	19359036	4	100.00%
5	4 2	600	5	1	289642	31771326	4	80.00%
5	4 3	600	9	0	160583	19909691	4	100.00%
5	4 4	600	10	0	147833	17750898	4	100.00%
5	4 5	600	14	0	128744	15921137	4	100.00%
5	5 2	600	5	0	227885	24831888	4	100.00%
5	5 3	600	9	0	148726	18168547	4	100.00%
5	5 4	600	13	0	125556	15599615	4	100.00%
5	5 5	600	16	0	106712	14358872	4	100.00%

Tabla 20: Resultados Procedimientos Exactos Con Huecos con Demanda 40 Bloques 5

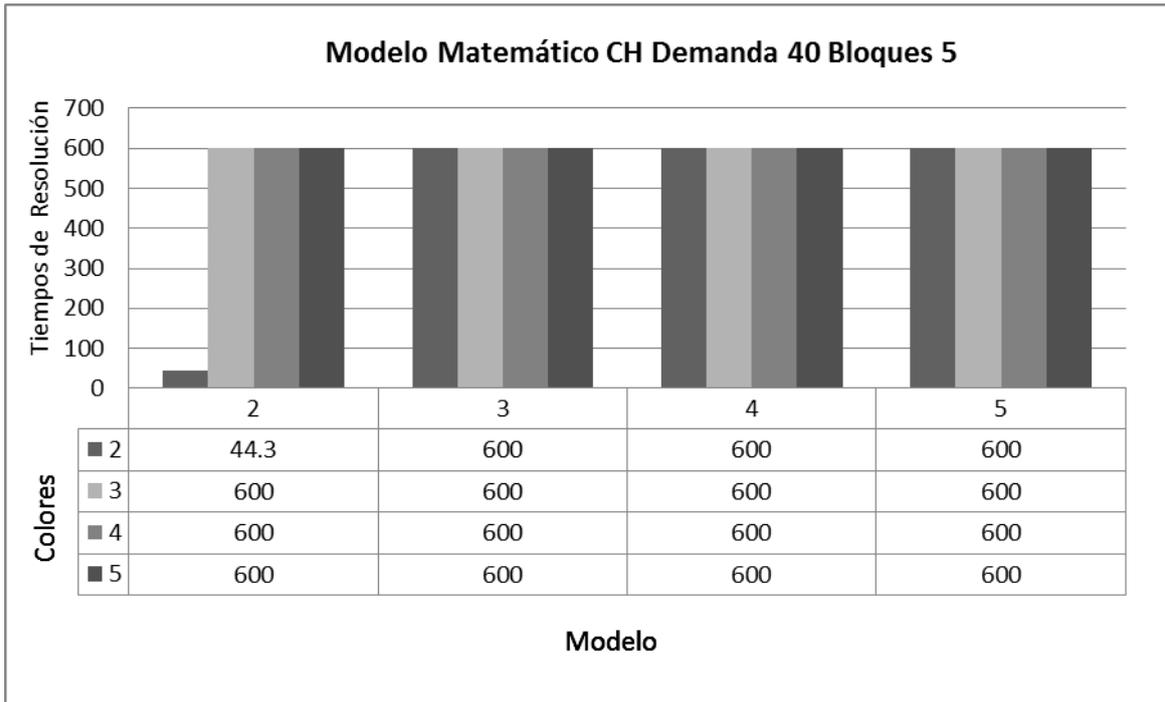


Figura 35: Comparativo del Tiempo de Optimización CBBSSP-CH Dem 40 Bloques 5

Demanda: 40									
Bloques	Mod Col	Tpo. (Seg)	F. Obj	Best Bound	Nodos	Iteraciones	Procesadores	GAP	
10	2 2	61.76	3	3	37373	2435825	4	0.00%	
10	2 3	562.86	2	2	369869	34193193	4	0.00%	
10	2 4	600	5	1	298789	30801389	4	80.00%	
10	2 5	600	8	0	255250	25395689	4	100.00%	
10	3 2	600	3	1	413486	37261723	4	66.67%	
10	3 3	600	6	0	212361	24301646	4	100.00%	
10	3 4	600	8	0	185464	19854762	4	100.00%	
10	3 5	600	10	0	171077	19131602	4	100.00%	
10	4 2	600	4	1	248457	27541268	4	75.00%	
10	4 3	600	9	0	215719	22426838	4	100.00%	
10	4 4	600	11	0	152701	18072353	4	100.00%	
10	4 5	600	14	0	147748	16310736	4	100.00%	
10	5 2	600	5	*	278609	28378413	4	100.00%	
10	5 3	600	5	0	165194	19736154	4	100.00%	
10	5 4	600	10	0	115115	15791853	4	100.00%	
10	5 5	600	13	0	109425	14817448	4	100.00%	

Tabla 21: Resultados Procedimientos Exactos Con Huecos con Demanda 40 Bloques 10

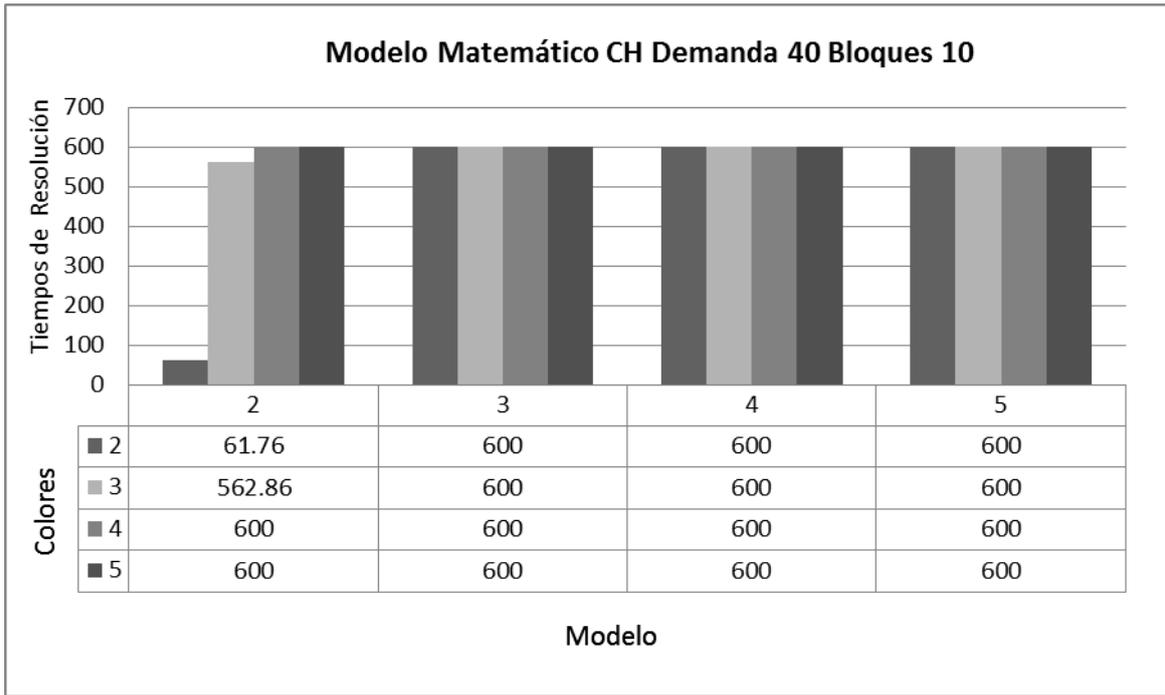


Figura 36: Comparativo del Tiempo de Optimización CBBSSP-CH Dem 40 Bloques 10

5.2.4 ANALISIS DE RESULTADOS DE LOS PROCEDIMIENTOS EXACTOS CON HUECOS

De los experimentos con base en los modelos con procedimientos exactos en el caso del Problema CBBSSP-CH se puede concluir que el tiempo de ejecución crece rápidamente a medida que se incrementa el valor de las variables independientes. Es decir a medida que el valor del número de modelos y colores, el tamaño de la demanda y los bloques se incrementan, el tiempo de ejecución aumenta. Tal como se hizo en el capítulo anterior se ha puesto como límite superior en el tiempo 600 segundos. Los experimentos terminan en el momento en que supera este valor. Si el tiempo es superior a los 10 minutos se asume que el tiempo de ejecución será muy largo y por tanto el problema se determina como tipo *NP_hard*. En las tablas (16 - 20) y en las Figuras (28 - 32) se observa el comportamiento mencionado.

5.3 PROCEDIMIENTOS HEURÍSTICOS PARA LA RESOLUCION DEL PROBLEMA CBBSSP CON HUECOS

INTRODUCCION

En el presente capítulo se describen las heurísticas que se emplean para la resolución del problema del CBBSSP-CH (*Cyclic Batch and Bi-Objective Setups Scheduling Problem With Holes*), es decir, Problema de Programación de Lotes Cíclicos con Tiempos de Preparación Bi-Objetivo Con Huecos. La necesidad de métodos heurísticos se tiene por el hecho de que el problema CBBSSP-CH es *NP-hard*, por lo que resolverlo de manera exacta suele implicar un tiempo computacional muy grande cuando se tiene un elevado número de variables y restricciones. Las heurísticas que se presentan se aplican en un entorno controlado restringiendo el número de variables independientes que tienen un efecto significativo sobre las variables dependientes.

Se diseñan una serie de heurísticas para el problema del CBBSSP utilizando la combinatoria de una serie de categorías conceptuales:

- 5) Listas: La forma de secuenciación de las listas de entrada.
- 6) Estrategia: La forma de alimentación en el proceso, sea ésta horizontal, vertical o diagonal.
- 7) Acomodo: El acomodo libre o el acomodo utilizando alguna restricción. En el caso del acomodo con restricciones se generan una serie de combinaciones dependiendo de la existencia o no de un renglón fijo y la existencia o no de una columna fija.

De todas las posibles combinaciones de las variables planteadas se considera necesario restringir las heurísticas a aquellas que en forma realista sean aplicables en el campo de la fabricación industrial. Sin considerar las listas, el total de heurísticas propuestas para el caso con huecos son 19.

5.3.1 METODOLOGIA DEL DESARROLLO DE HEURISTICAS CON HUECOS.

El desarrollo de las heurísticas se hace en esta sección. Se parte del concepto simple de alimentación del producto en una secuencia previamente establecida. El método para establecer la secuencia inicial utiliza específicamente diferentes tipos de listas. Las listas al igual que en el caso sin huecos en esta Tesis son nueve y también dependen del orden ascendente o descendente de variables independientes. Las variables independientes para las listas son: color, geometría o modelo, demanda o cantidad a producir. Una vez establecida la lista será alimentada en el proceso de acuerdo a un procedimiento distinto para cada heurística. Los procedimientos son generados a partir de la combinación de las categorías mencionadas en el apartado anterior: Estrategia y Acomodo. En este caso no existe el concepto de ventaja.

De esta manera, en este capítulo se procede en primer lugar a exponer un marco general de las heurísticas desarrolladas partiendo de las tres estrategias básicas que se han desarrollado, en segundo lugar para cada estrategia se establecen los procedimientos para las diferentes opciones generadas por el concepto de acomodo. El concepto de ventaja utilizado en las heurísticas sin huecos no se utiliza en las heurísticas con huecos. La razón es que intentar hacer el acomodo con ventaja no tiene aplicación práctica o realista al considerar huecos. El hueco aparece cuando hay un cambio y si no

es posible colocar un producto distinto no tiene sentido indicar cual tendría preferencia sobre otro (ventaja).

Así se presentan las heurísticas básicas y los correspondientes apartados en los que son descritas con cada una de sus variantes:

- 1) H-H: Heurísticas basadas en la estrategia horizontal (sección 5.3.3).
- 2) H-V: Heurísticas basadas en la estrategia vertical (sección 5.3.4).
- 3) H-D: Heurísticas basadas en la estrategia diagonal (sección 5.3.5).

Por último, se termina el apartado con una codificación completa de los procedimientos heurísticos aquí presentados. Esta codificación es la que sirve de base para realizar el estudio experimental. En el apartado siguiente se hace la experimentación y se presentan los resultados. En esta apartado se presenta al final una sección de conclusiones sobre la definición de heurísticas y codificación de los modelos diseñados.

5.3.2 MARCO GENERAL DE LAS HEURISTICAS DESARROLLADAS PARA EL PROBLEMA DEL CBBSSP-CH.

Las heurísticas planteadas en esta tesis para la resolución del problema CBBSSP-CH son de comprensión sencilla y pueden ser utilizadas en la industria dado que la implementación tampoco representa complejidad, Estas heurísticas determinan el orden de alimentación de los productos a un sistema de ciclo cerrado considerando el número de bloques disponibles, la cantidad a fabricar de cada tipo de producto y la variedad de los mismos dados por el color y la geometría. De acuerdo a los establecido en el capítulo 3 Descripción del Problema de Programación de Lotes Cíclicos con tiempos de Preparación Bi-Objetivo Dependientes e Independientes de la Secuencia en el caso del problema Con Huecos (CBBSSP-CH) (apartado 3.2.3) el objetivo de la tesis es minimizar el número de espacios vacíos y el coste asociado a los mismos. Los espacios vacíos ocurren en una secuencia específica por la existencia de cambios de color y de modelos en los sistemas cíclicos de pintura en la industria de autopartes automotrices lo cual se establece como función objetivo de las heurísticas desarrolladas en el presente capítulo.

5.3.2.1 Procedimiento General:

El procedimiento general de las heurísticas Con Huecos está basado en un algoritmo constructivo con tres elementos:

- 1) Una lista en la que los productos son agrupados y dentro de la cual son elegidos para ser secuenciados. La lista se ordena de acuerdo a criterios específicos.
- 2) Una estrategia para elegir una posición en la cuál será acomodado el producto que será seleccionado.
- 3) Una estrategia para seleccionar un producto de la lista, tomando en consideración las cantidades que se deben fabricar de cada tipo de producto.

Listas:

La lista de prioridades puede definirse utilizando nueve posibilidades de secuenciación:

- 1) Secuencia por modelo y después por color: En este caso todos los productos son agrupados de acuerdo a su geometría y ordenados lexicográficamente. A continuación cada subconjunto es ordenado internamente en forma lexicográfica por color.
- 2) Secuencia por color y después por modelo: Similar al anterior, pero en este caso todos los productos son agrupados primero de acuerdo a su color y los subconjuntos generados son ordenados lexicográficamente entre ellos. A continuación cada subconjunto es ordenado internamente en forma lexicográfica por modelo.
- 3) Por tamaño de la demanda y después por color. Los productos son secuenciados dando prioridad al orden descendente en el tamaño de la demanda por producto (Requerimiento). Si hay un empate en dicha demanda en dos o más tipos de producto se decide por la secuencia de color. En otras palabras se hace un ordenamiento de los productos por tamaño de la demanda y cuando se tiene la misma demanda en algunos productos se selecciona el siguiente producto por el orden lexicográfico del color.
- 4) Por tamaño de la demanda y después por modelo. Los productos son secuenciados dando prioridad al orden descendente en el tamaño de la demanda por producto (Requerimiento). Si hay un empate en dicha demanda en dos o más tipos de producto se decide por la secuencia de modelo. En otras palabras se hace un ordenamiento de los productos por tamaño de la demanda y cuando se tiene la misma demanda en algunos productos se selecciona el siguiente producto por el orden lexicográfico del modelo.
- 5) Secuencia por color y después por el tamaño de la demanda de cada modelo: En este caso todos los productos son agrupados primero de acuerdo a su color y secuenciados lexicográficamente. Los subconjuntos generados se ordenan a su vez al interior dependiendo del tamaño de la demanda por modelo.
- 6) Secuencia por modelo y después por el tamaño de la demanda de cada color: En este caso todos los productos son agrupados primero de acuerdo a su modelo y secuenciados lexicográficamente. Los subconjuntos generados se ordenan a su vez al interior dependiendo del tamaño de la demanda por color.
- 7) Tamaño de la demanda por color y después por el tamaño de la demanda de cada modelo: En este caso todos los productos son agrupados primero de acuerdo a su color y secuenciados por el tamaño de la demanda total de cada color. Los

subconjuntos generados se ordenan a su vez al interior dependiendo del tamaño de la demanda por modelo.

- 8) Tamaño de la demanda por modelo y después por el tamaño de la demanda de cada color: En este caso primero todos los productos son agrupados de acuerdo al modelo y secuenciados por el tamaño de la demanda total de cada modelo. Los subconjuntos generados se ordenan a su vez al interior dependiendo del tamaño de la demanda por color.
- 9) Secuencia estocástica: En forma aleatoria se elige el siguiente producto en la secuencia.

Una vez que se obtienen las diferentes listas de acuerdo a los criterios mencionados en los nueve casos anteriores éstas se utilizan para ser expandidas de acuerdo a la demanda solicitada para cada tipo de producto. En otras palabras, en cada lista se obtiene el orden definitivo de cada producto individual (no solo por tipo de producto). Si la lista incluye 10 tipos de productos con demanda de 5 cada uno, la lista antes de expandirla tiene 10 elementos y la lista expandida contiene 50.

Estrategia de Acomodo.

Una estrategia para elegir una posición en la cuál será acomodado el producto que será seleccionado. La aplicación de una estrategia de secuenciación implica la elección de una posición en la que un producto individual, seleccionado de una manera específica, es asignado. La forma de elegir una posición en la matriz de acomodo puede ser en secuencia horizontal, vertical o diagonal.

Estrategia para seleccionar un Producto de la Lista:

La selección de un producto de la lista puede ser en forma libre y directa o atendiendo a algunas restricciones impuestas por las características de la heurística que en particular se ejecute. Las características en el caso de los procedimientos con huecos pueden ser por considerar o no una columna cero y/o una fila cero.

Las columnas y filas cero sirven como marco de referencia al momento de asignar un producto. El tipo de acomodo indica el tipo de secuencia que se debe seguir y la estrategia de selección ayuda a elegir un producto de la lista que se debe colocar como el siguiente en el proceso de producción. En este caso, si existe una fila o columna cero se debe atender primero al tipo de producto que marque la referencia cero antes de colocar un producto con las mismas características que el anterior que ha sido colocado. Esta regla implica que muchas veces aunque exista un producto que cumpla horizontalmente con el producto previo y verticalmente con el que fue colocado en el ciclo anterior, se debe colocar el que cumpla con la referencia cero.

En las heurísticas desarrolladas, el tipo de referencia cero, sea columna o fila puede ser de dos tipos:

- 1) Referencia Proporcional.

2) Referencia PRV.

Las filas y/o columnas cero tipo Proporcional son asignadas calculando la proporción o porcentaje por tipo de producto del total de la demanda de productos. Para el caso de la columna cero, la proporción se calcula respecto al total de demanda por color. Así cada posición de la columna cero tendrá un color fijo que servirá de referencia. Es decir. La fila 1, en la posición de columna cero, determinará el color de referencia del inicio de esa fila 1. La fila 2 de la columna cero, el color de la segunda fila y así sucesivamente. Lo mismo sucederá en el caso de la fila cero, que determinará en este caso el modelo de referencia de toda la columna 1.

Las filas y/o columnas cero tipo PRV son asignadas de acuerdo al algoritmo desarrollado como “Problema de Rotación de Vehículos”. La asignación de la fila o columna de referencia también es fija y funciona como en el caso Proporcional. La diferencia consiste en la forma de calcular los valores de referencia. La fórmula básica es:

$$V = \text{Mayor valor de } \left(\frac{P_n}{C_n + \Delta} \right)$$

Dónde:

V = Valor a asignar en la columna o fila.

P_n = Cantidad existente del Producto n.

C_n = Cantidad asignada del Producto n.

Δ = Valor Delta.

En la fórmula presentada implica la lección del mayor valor encontrado para cada iteración. Una iteración se hace para cada posición de la columna cero o de la fila cero.

5.3.3 HEURISTICAS HORIZONTALES

Las heurísticas horizontales desarrolladas se presentan a continuación. Se eligió como forma de referenciarlas asignar un número consecutivo a partir del 10 para las heurísticas con esta estrategia. En la lista siguiente se describe cada una de ellas y en la Tabla 22 se observa la clasificación general de las mismas.

- Algoritmo 10: Heurística Horizontal Libre.
- Algoritmo 11: Heurística Horizontal con Fila Cero. Fila cero Proporcional.
- Algoritmo 12: Heurística Horizontal con Fila Cero. Fila cero PRV.
- Algoritmo 13: Heurística Horizontal con Columna Cero. Columna cero Proporcional.

- Algoritmo 14: Heurística Horizontal con Columna Cero. Columna cero PRV.
- Algoritmo 15: Heurística Horizontal con Fila Cero y Columna Cero. Ambas en Proporcional.
- Algoritmo 16: Heurística Horizontal con Fila Cero y Columna Cero. Ambas en PRV.

En el caso de los Algoritmos 13 y 15 que hacen falta en la secuencia anterior, se habría considerado la situación de tener columna cero proporcional. Debido a que se desconoce el número de filas en las que se colocaría el producto desde el momento de iniciar la ejecución de estos algoritmos, no es posible hacer un reparto proporcional de la columna para asignar una referencia al colocar los productos en forma horizontal. Por tanto se consideran solo 5 algoritmos en la Estrategia Horizontal.

	Horizontal		
	Sin Fila 0	Con Fila 0 Prop	Con Fila 0 PRV
Sin Columna 0	Alg. 10	Alg. 11	Alg. 12
Con Columna 0 Prop.			
Con Columna 0 PRV	Alg. 14		Alg. 16

Tabla 22: Resumen Heurísticas Horizontales Con Huecos

5.3.4 HEURISTICAS VERTICALES

Las heurísticas verticales desarrolladas se presentan a continuación. Se eligió como forma de referenciarlas asignar un número consecutivo a partir del 20 para las heurísticas con esta estrategia. En la Tabla 23 se puede ver un resumen de los algoritmos:

- Algoritmo 20: Heurística Vertical Libre.
- Algoritmo 21: Heurística Vertical con Fila Cero. Fila cero Proporcional.
- Algoritmo 22: Heurística Vertical con Fila Cero. Fila cero PRV.
- Algoritmo 23: Heurística Vertical con Columna Cero. Columna cero Proporcional.
- Algoritmo 24: Heurística Vertical con Columna Cero. Columna cero PRV.
- Algoritmo 25: Heurística Vertical con Fila Cero y Columna Cero. Ambas en Proporcional.
- Algoritmo 26: Heurística Vertical con Fila Cero y Columna Cero. Ambas en PRV.

	Vertical		
	Sin Fila 0	Con Fila 0 Prop	Con Fila 0 PRV
Sin Columna 0	Alg. 20	Alg. 21	Alg. 22
Con Columna 0 Prop.	Alg. 23	Alg. 25	
Con Columna 0 PRV	Alg. 24		Alg. 26

Tabla 23: Resumen Heurísticas Verticales Con Huecos

5.3.5 HEURISTICAS DIAGONALES

Las heurísticas diagonales desarrolladas se presentan a continuación. Se eligió como forma de referenciarlas asignar un número consecutivo a partir del 30 para las heurísticas con esta estrategia. En la Tabla 24 se observa un resumen de estos procedimientos:

- Algoritmo 30: Heurística Diagonal Libre.
- Algoritmo 31: Heurística Diagonal con Fila Cero. Fila cero Proporcional.
- Algoritmo 32: Heurística Diagonal con Fila Cero. Fila cero PRV.
- Algoritmo 33: Heurística Diagonal con Columna Cero. Columna cero Proporcional.
- Algoritmo 34: Heurística Diagonal con Columna Cero. Columna cero PRV.
- Algoritmo 35: Heurística Diagonal con Fila Cero y Columna Cero. Ambas en Proporcional.
- Algoritmo 36: Heurística Diagonal con Fila Cero y Columna Cero. Ambas en PRV.

	Diagonal		
	Sin Fila 0	Con Fila 0 Prop	Con Fila 0 PRV
Sin Columna 0	Alg. 30	Alg. 31	Alg. 32
Con Columna 0 Prop.	Alg. 33	Alg. 35	
Con Columna 0 PRV	Alg. 34		Alg. 36

Tabla 24: Resumen Heurísticas Diagonales Con Huecos

5.3.6 CODIFICACION Y DEFINICION DE LOS PROCEDIMIENTOS EXISTENTES

En este apartado se han desarrollado 19 heurísticas para resolver el problema CBBSSP-CH. La cantidad de procedimientos permite explorar las variables dependientes e independientes del problema. Se ha desarrollado el

código de programación para cada heurística asignando un algoritmo para cada una. El lenguaje utilizado es JAVA

Códigos de programación.

Se puede ver en el la sección de ANEXOS algunos de los códigos desarrollados.

5.4 ESTUDIO EXPERIMENTAL DE LOS PROCEDIMIENTOS DESARROLLADOS PROBLEMA CON HUECOS

5.4.1 INTRODUCCION

Para definir el grupo de experimentos utilizando el modelo matemático del problema CBBSSP-CH fue necesario determinar valores realistas que exploren el caso de problemas aplicables en la industria.

Un objetivo adicional al asignar los valores a las variables de cada instancia es conocer el comportamiento de los resultados a distintos niveles y la combinación de los efectos entre variables.

El objetivo principal en este apartado es determinar cuál o cuáles son las heurísticas que tienen el mejor resultado al buscar minimizar el número de espacios vacíos en un sistema de circuito cerrado de pintura.

5.4.2 METODOLOGIA DEL ESTUDIO EXPERIMENTAL.

Como se ha mencionado, el estudio experimental tiene como objetivo determinar las heurísticas de mejor desempeño. Para poder tener argumentos que lo demuestren se establece el método como sigue:

- 1) Se generan instancias que validen los resultados de las heurísticas en valores realistas y aplicables en la industria. Que muestren posibles valores que ocurren en las empresas. Asegurar que esas instancias permitan un análisis de sensibilidad de los valores y el efecto de distintos niveles.
- 2) Ejecutar los experimentos con los algoritmos de las heurísticas desarrollados en JAVA.
- 3) Validar los resultados utilizando software comercial. En este caso SPSS 17. Obtener datos de valores de resultados por Algoritmo y por Lista. Considerar todos los casos generados.
- 4) Dado que la variable dependiente es única: Número de Espacios Vacíos generados sea por Cambios de Color Horizontal o por Cambios de Modelo Vertical. Definir una forma de hacer un análisis Mono-Objetivo.
- 5) La definición del Análisis Mono-Objetivo para el Problema CBBSSP con Huecos es más simple que el que se hizo en el caso del CBBSSP sin Huecos. En este caso el resultado de Número de Espacios Vacíos ya contiene tanto los que son generados por el cambio de modelo y/o los generados por el cambio de color. No es

necesario definir una forma de unificar dos variables dependientes. Para hacer el Análisis Mono-Objetivo se utilizarán las pruebas Pos-Hoc de Análisis Estadístico.

- 6) Se define como las mejores heurísticas aquellas que mantienen el mejor lugar relativo en las pruebas de Análisis Estadístico.

5.4.3 GENERACION DE PROBLEMAS PARA EL ESTUDIO EXPERIMENTAL

Con base en el punto anterior se definieron como experimentos la combinatoria de las variables independientes con los valores que se presentan en la Tabla 25:

Variables	Nombre	Variantes	Valores
Instancias del problema o Variables Independientes	Color	3	5,10,20
	Modelo	3	5,10,20
	Demanda	3	100, 1000, 10000
	Bloques	3	20, 50 , 100
Procedimentales	Algoritmo	19	(10,11,12,14,16),(20..26),(30..36)
	Lista	9	1..9
Dependiente	Vacíos		Min y Max

Tabla 25: Instancias del Problema CBBSSP-CH

A continuación se presenta una explicación y/o justificación de los valores para cada instancia:

- 1) Color, modelo, demanda y bloques: Estas variables en el problema se debe explorar para valores pequeños y grandes. Se han elegido los valores de 5, 10 y 20 que representan posibles combinaciones realistas del problema.
- 2) Algoritmo: Número específico de procedimiento heurístico para resolver el problema, los cuales han sido explicados ampliamente en el apartado anterior.
- 3) Lista: Número de ordenamiento específico inicial de las listas de secuenciación.
- 4) Vacíos: Número de espacios vacíos generados sea por 1) *setups* horizontales por cambio de color y/o por 2) *setups* verticales por cambio de modelo.

El número de experimentos totales para los procedimientos exactos en el problema CBBSSP-CH se general al multiplicar las variantes de las instancias mostradas.

A través de los códigos de programación desarrollados para cada uno de los algoritmos presentados se generó una base de datos con 13,851 registros.

Se puede conocer la forma como se integró esta base de datos considerando los tipos conceptuales de variables en la tabla anterior:

- 1) Variables independientes:

(3 variantes de colores)
x (3 variantes de modelos)
x (3 variantes de demanda)
x (3 variantes de bloques)
= 81 Instancias

2) Procedimientos:

(19 heurísticas de solución)
x (9 Listas)
= 171 procedimientos.

3) El total de experimentos que generan la variable dependiente se compone finalmente así:

(81) Combinaciones de Instancias o variables independientes
x 171 Procedimientos:
= 13, 851 casos

5.4.4 ESTUDIO EXPERIMENTAL DE LOS PROCEDIMIENTOS HEURISTICOS.

5.4.4.1 Validación de los datos generados en los experimentos con las heurísticas.

Se presenta a continuación, en las siguientes tablas: Tabla 26 y Tabla 27 un resumen de los valores generados por caso tanto para Algoritmos como para Listas. Se muestra así que la distribución de datos es regular y uniforme. Es decir que los experimentos se ejecutaron para todos los casos descritos en forma completa.

ALG

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	10	729	5.3	5.3	5.3
	11	729	5.3	5.3	10.5
	12	729	5.3	5.3	15.8
	14	729	5.3	5.3	21.1
	16	729	5.3	5.3	26.3
	20	729	5.3	5.3	31.6
	21	729	5.3	5.3	36.8
	22	729	5.3	5.3	42.1
	23	729	5.3	5.3	47.4
	24	729	5.3	5.3	52.6
	25	729	5.3	5.3	57.9
	26	729	5.3	5.3	63.2
	30	729	5.3	5.3	68.4
	31	729	5.3	5.3	73.7
	32	729	5.3	5.3	78.9
	33	729	5.3	5.3	84.2
	34	729	5.3	5.3	89.5
	35	729	5.3	5.3	94.7
	36	729	5.3	5.3	100.0
	Total	13851	100.0	100.0	

Tabla 26: Frecuencias de los datos por Algoritmo

LIST

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	1	1539	11.1	11.1	11.1
	2	1539	11.1	11.1	22.2
	3	1539	11.1	11.1	33.3
	4	1539	11.1	11.1	44.4
	5	1539	11.1	11.1	55.6
	6	1539	11.1	11.1	66.7
	7	1539	11.1	11.1	77.8
	8	1539	11.1	11.1	88.9
	9	1539	11.1	11.1	100.0
	Total	13851	100.0	100.0	

Tabla 27: Frecuencia de los datos por Lista

5.4.4.2 Resultados del Análisis de Varianza

Se presenta a continuación el Análisis de Varianza ANOVA, para determinar el efecto de las variables independientes en los resultados finales

de la variable dependiente. En este caso se busca saber si estadísticamente es posible decir que un resultado es mejor que otro para cada variable.

Un análisis de la varianza permite determinar si diferentes tratamientos muestran diferencias significativas o por el contrario puede suponerse que sus medias poblacionales no difieren. El análisis de la varianza permite superar las limitaciones de hacer contrastes bilaterales por parejas (que son un mal método para determinar si un conjunto de variables con $n > 2$ difieren entre sí). Para el caso del Problema CBBSSP-CH se utiliza el nivel de significancia de 0.05 esto quiere decir que cuando en el valor obtenido en el valor de significancia de la relación de las variables analizadas con la prueba F de Fisher es inferior a 0.05 se obtiene como conclusión que la relación es significativa. En otras palabras, que hay evidencia estadística para poder decir que los distintos valores de las variables analizadas tienen efecto en los resultados. Si el valor es superior a 0.05 el resultado en la significancia quiere decir que no hay diferencia entre los valores internos de las variables analizadas. No hay evidencia estadística de que un nivel sea mejor que otro. Se muestra en la Tabla 28 donde se puede observar el resultado de los efectos inter-sujetos. Es decir los efectos entre cada una de las variables analizadas en este problema.

Pruebas de los efectos inter-sujetos

Variable dependiente:HUECOS

Origen	Suma de cuadrados tipo III	gl	Media cuadrática	F	Sig.
Modelo corregido	8.358E9	80	1.045E8	352.530	.000
Intersección	7.023E9	1	7.023E9	23697.313	.000
TDEM	7.237E9	2	3.618E9	12209.865	.000
BLOQ	2.627E7	2	1.313E7	44.323	.000
MOD	7.900E7	2	3.950E7	133.292	.000
COL	3.730E8	2	1.865E8	629.325	.000
TDEM * BLOQ	3.855E7	4	9638330.454	32.524	.000
TDEM * MOD	6.882E7	4	1.721E7	58.059	.000
TDEM * COL	4.604E8	4	1.151E8	388.414	.000
BLOQ * MOD	2.066E7	4	5165101.737	17.430	.000
BLOQ * COL	3520099.925	4	880024.981	2.970	.018
MOD * COL	1719683.923	4	429920.981	1.451	.214
TDEM * BLOQ * MOD	3.793E7	8	4741813.323	16.001	.000
TDEM * BLOQ * COL	5564984.079	8	695623.010	2.347	.016
TDEM * MOD * COL	3890637.313	8	486329.664	1.641	.108
BLOQ * MOD * COL	496485.615	8	62060.702	.209	.989
TDEM * BLOQ * MOD * COL	1135545.470	16	70971.592	.239	.999
Error	4.081E9	13770	296342.343		
Total	1.946E10	13851			
Total corregida	1.244E10	13850			

a. R cuadrado = .672 (R cuadrado corregida = .670)

Tabla 28: Efectos inter-sujetos en variables independientes del Problema CBBSSP-CH

Se puede observar que existe un nivel de significancia inferior a 0.05 en casi todas las relaciones presentadas. Solamente en las siguientes relaciones es superior:

- 1) MOD*COL
- 2) BLOQ*MOD*COL
- 3) TDEM*BLOQ*MOD*COL

Esto quiere decir que cuando se analizan los tres casos indicados el resultado interno, es decir los distintos valores que resultan en las combinaciones internas no muestran evidencia estadística de que uno sea mejor que otro. En el resto de las relaciones que muestra la Tabla 28 si hay evidencia para decir que hay variaciones entre los resultados con los diferentes valores que pueda tomar cada una de las variables.

Dado que se ha mostrado con lo explicado en el párrafo anterior que hay diferencias significativas en los efectos de las variables independientes y solo en tres casos de intersección hay un efecto no relevante el análisis de los resultados se enfoca ahora a determinar cuál Algoritmo o cual Lista es mejor para generar menos espacios vacíos, dado que lo que se busca es minimizarlos. Existe una prueba que ayuda a clasificar los valores de los resultados. Si un resultado es mejor que otro y existen 19 posibles en el caso de los Algoritmos, lo importante es asegurar que los niveles de calidad en el resultado sean consistentes. Es decir que el valor que se obtiene en el orden no sea casual sino que siempre se mantenga. Para esto existen Pruebas Pos-Hoc de los datos. Entre ellas la que se puede aplicar en este análisis de datos es la Prueba Tukey-B. En ella se indican niveles en los que se clasifican los resultados.

Se procede a presentar el comportamiento de los distintos niveles o valores de los procedimientos, es decir de los Algoritmos y de las Listas. Se aprecia en las siguientes tablas cual es el orden de efectividad y la eficiencia de cada Algoritmo y de cada Lista en la minimización de espacios vacíos, para el caso de las tablas mencionadas se utilizan todos los datos de la base generada, sin ninguna segmentación:

HUECOS

Tukey B^a

ALG	N	Subconjunto para alfa = .05	
		1	2
10	729	357.00	
11	729	383.61	
12	729	388.44	
14	729	433.29	
16	729	469.54	
20	729		752.52
22	729		761.75
21	729		761.99
32	729		789.17
36	729		804.23
31	729		816.19
30	729		829.62
23	729		843.06
24	729		847.45
26	729		850.10
33	729		851.05
25	729		854.50
34	729		856.41
35	729		878.87

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 729.000.

Tabla 29: Tukey-B. Efecto de variable Algoritmos sobre Datos Generales

HUECOS

Tukey B^a

LIST	N	Subconjunto para alfa = .05	
		1	2
4	1539	622.34	
3	1539	623.65	
9	1539	639.91	
1	1539	679.51	
8	1539	684.17	
6	1539	684.20	
2	1539		778.02
5	1539		843.91
7	1539		852.68

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 1539.000.

Tabla 30: Tukey-B. Efecto de variable Listas sobre Datos Generales

En las tablas anteriores se observa que los Algoritmos horizontales cuando se toman todos los datos de las instancias del problema funcionan de mejor manera que el resto de los Algoritmos, y en el caso de las listas, la 2, 5, y 7 funcionan de peor manera que el resto. Sin embargo lo que se ha descrito no necesariamente funciona cuando se hace un análisis segmentado de cada

variable. El análisis segmentado quiere decir separar la base de datos general por valores de las variables independientes. Por ejemplo la variable: Tamaño de Demanda. Se hace una separación de los datos por cada uno de los tres niveles: 100, 1000 y 10000 que se han definido en los experimentos. Los resultados obtenidos a nivel valor específico de demanda pueden ser distintos al general y al de otro nivel que sea analizado. Para el caso de Tamaño de Demanda se muestran en las siguientes tablas los resultados. Para el análisis de los Algoritmos se muestra una general de ANOVA y enseguida 3 con el análisis de cada nivel (100, 1000 y 10000) Para el caso de las Listas se procede de igual manera.

En la Tabla 31 se observa que a diferentes niveles de demanda existe significancia en los resultados de los datos. Es decir, el número de espacios vacíos o huecos en cada nivel asegura variación en los resultados de los Algoritmos.

ANOVA

HUECOS		Suma de cuadrados	gl	Media cuadrática	F	Sig.
100	Inter-grupos	1436281.262	18	79793.403	214.720	.000
	Intra-grupos	1708690.247	4598	371.616		
	Total	3144971.509	4616			
1000	Inter-grupos	6245217.918	18	346956.551	25.589	.000
	Intra-grupos	6.234E7	4598	13558.611		
	Total	6.859E7	4616			
10000	Inter-grupos	1.302E9	18	7.235E7	86.914	.000
	Intra-grupos	3.828E9	4598	832441.211		
	Total	5.130E9	4616			

Tabla 31: Análisis de Varianza de la variable Algoritmos a diferentes niveles de Demanda

A nivel de demanda 100, el algoritmo 10 y el 14 son los que funcionan mejor y no tienen traslape con el segundo nivel en la prueba Tukey-B. Esto se observa en la Tabla 32.

TDEM=100

0ª

ALG	N	Subconjunto para alfa = .05					
		1	2	3	4	5	6
10	243	43.03					
14	243	43.40					
12	243		58.16				
30	243	.00	59.56	59.56			
20	243		59.92	59.92			
16	243		60.25	60.25			
32	243		61.35	61.35			
33	243		62.49	62.49			
34	243		62.51	62.51			
24	243		62.75	62.75			
23	243		62.77	62.77			
36	243		63.38	63.38			
31	243		63.57	63.57			
35	243			64.83			
22	243				74.36		
26	243				75.67		
11	243					93.37	
21	243						109.14
25	243						109.74

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 243.000.

Tabla 32: Tukey-B. Efecto de variable Algoritmos sobre Datos con Demanda 100

Cuando se hace la prueba Tukey-B a nivel de Demanda 1000, el mejor resultado es el Algoritmo 10. En este caso, aunque se puede observar en la Tabla 33 que hay otros algoritmos en el primer nivel (12, 20, 14 y 16), estos también aparecen en el segundo nivel. Por esto no se puede asegurar que en forma consistente puedan obtener siempre el mejor resultado y permanecer en el primer nivel en forma normal. Solamente el Algoritmo 10.

TDEM=1000

Tukey B^a

ALG	N	Subconjunto para alfa = .05							
		1	2	3	4	5	6	7	8
10	243	288.51							
12	243	299.83	299.83						
20	243	305.56	305.56						
14	243	306.32	306.32						
16	243	314.98	314.98	314.98					
23	243		322.28	322.28	322.28				
22	243		322.91	322.91	322.91				
24	243		324.81	324.81	324.81				
11	243		325.33	325.33	325.33				
26	243			341.97	341.97	341.97			
21	243				349.60	349.60	349.60		
32	243					368.54	368.54	368.54	
25	243					369.59	369.59	369.59	
36	243						379.12	379.12	379.12
30	243						379.58	379.58	379.58
31	243							383.58	383.58
34	243							397.68	397.68
33	243							397.95	397.95
35	243								409.47

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 243.000.

Tabla 33: Tukey-B. Efecto de variable Algoritmos sobre Datos con Demanda 1000

Al continuar ahora con la Tabla 34, se puede observar en el nivel de Demanda de 10,000 que los algoritmos con los mejores resultados que no aparecen tampoco en el segundo o tercer nivel son los algoritmos 10 y 11. El 12 y el 14 aparecen en el primero y el segundo.

TDEM=10000

Tukey B^a

ALG	N	Subconjunto para alfa = .05			
		1	2	3	4
11	243	732.13			
10	243	739.47			
12	243	807.33	807.33		
14	243	950.16	950.16		
16	243		1033.39		
21	243			1827.23	
22	243			1887.99	1887.99
20	243			1892.08	1892.08
32	243			1937.61	1937.61
36	243			1970.17	1970.17
31	243			2001.41	2001.41
30	243			2049.72	2049.72
25	243			2084.19	2084.19
33	243			2092.70	2092.70
34	243				2109.05
26	243				2132.67
23	243				2144.15
24	243				2154.78
35	243				2162.30

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 243.000.

Tabla 34: Tukey-B. Efecto de variable Algoritmos sobre Datos con Demanda 10000

5.4.4.3 Resultados de los Algoritmos:

Con las tablas anteriores se puede concluir que el comportamiento de los algoritmos es necesario analizarlo a diferentes niveles de cada una de las variables independientes. Si se hace solamente a nivel general, estadísticamente se podría concluir que los algoritmos: 10, 11, 12, 14 y 16, es decir, los horizontales, son mejores que el resto. Pero un análisis mediante la Prueba Tukey-B, haciendo la segmentación de la variable Demanda nos lleva a concluir que el único algoritmo que en forma consistente y repetida, en todos los niveles de demanda siempre aparece en el primer nivel, es el Algoritmo 10.

Para corroborar el desempeño de los Algoritmos con las demás variables, se puede observar el comportamiento de los mismos con las demás variables independientes: Número de Bloques, Número de modelos, Número de colores. (ANEXO 3).

En la Tabla 35 se observa que a diferentes niveles de demanda existe significancia en los resultados de los datos en los casos de Demanda 1000 y 10000. En el caso de la Demanda 100 el valor de significancia es superior 0.05, lo que indica que no hay diferencias entre los resultados que proporcionan las distintas listas cuando se tiene una demanda de 100 productos.

ANOVA

HUECOS

TDEM		Suma de cuadrados	gl	Media cuadrática	F	Sig.
100	Inter-grupos	8375.240	8	1046.905	1.538	.138
	Intra-grupos	3136596.269	4608	680.685		
	Total	3144971.509	4616			
1000	Inter-grupos	3565851.298	8	445731.412	31.588	.000
	Intra-grupos	6.502E7	4608	14110.647		
	Total	6.859E7	4616			
10000	Inter-grupos	2.461E8	8	3.077E7	29.031	.000
	Intra-grupos	4.884E9	4608	1059835.879		
	Total	5.130E9	4616			

Tabla 35: Análisis de Varianza de la variable Listas a diferentes niveles de Demanda

En el caso de la variable Listas se puede observar que cuando la demanda es 100 no hay diferencias significativas entre los valores que las distintas listas generan. Se observa en la Tabla 36 que solo existe un nivel y todas las Listas están en él. No se puede concluir en este nivel si una lista es mejor que otra.

TDEM=100

Tukey B^a

LIST	N	Subconjunto para alfa = .05
		1
4	513	65.65
9	513	66.28
3	513	66.91
6	513	67.81
8	513	68.00
1	513	68.08
5	513	69.21
7	513	69.56
2	513	69.67

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 513.000.

Tabla 36: Tukey-B. Efecto de variable Listas sobre Datos con Demanda 100

En el nivel de demanda de 1000 se puede observar en la Tabla 37 que las listas que generan mejor resultado son 4, 3, y 9. Estas también están incluidas en el nivel de demanda 10000 como se observa en la Tabla 38.

TDEM=1000

Tukey B^a

LIST	N	Subconjunto para alfa = .05	
		1	2
4	513	302.82	
3	513	303.43	
9	513	322.23	
8	513		356.13
1	513		356.14
6	513		358.36
2	513		368.67
5	513		375.73
7	513		376.94

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 513.000.

Tabla 37: Tukey-B. Efecto de variable Listas sobre Datos con Demanda 1000

TDEM=10000

Tukey B^a

LIST	N	Subconjunto para alfa = .05		
		1	2	3
4	513	1498.54		
3	513	1500.61		
9	513	1531.22		
1	513	1614.31		
6	513	1626.43		
8	513	1628.38		
2	513		1895.71	
5	513			2086.80
7	513			2111.52

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 513.000.

Tabla 38: Tukey-B. Efecto de variable Listas sobre Datos con Demanda 10000

5.4.4.4 Resultados de las Listas:

Con las tablas anteriores se puede concluir que el comportamiento de los listas es posible hacerlo a diferentes niveles de cada una de las variables independientes. Si se hace solamente a nivel general, estadísticamente se podría concluir que el 66% de las listas tienen buen desempeño: 4, 3, 9, 1, 8, y 6. Pero un análisis mediante la Prueba Tukey-B, haciendo la segmentación de la variable Demanda nos lleva a concluir que son 3 listas, es decir, el 33% del total, las que en forma consistente y repetida, en todos los niveles de demanda siempre aparece en el primer nivel, las listas 4, 3 y 9. Las listas 4 y 3 fueron creadas mediante un procedimiento pero la 9 es aleatoria.

Para corroborar el desempeño de los Listas con las demás variables, se puede observar el comportamiento de las mismas con las demás variables independientes: Número de Bloques, Número de modelos, Número de colores. (ANEXO 3).

6 CONCLUSIONES Y LINEAS FUTURAS DE INVESTIGACION

6.1 CONCLUSIONES Y APORTACIONES

En esta Tesis se han desarrollado y conseguido las siguientes aportaciones:

- Diseñar un modelo matemático para el problema CBBSSP-SH que minimice el número de ajustes de proceso generados por los cambios de modelo y los cambios de modelo en forma simultánea.
- Diseñar un modelo matemático para el problema CBBSSP-CH que minimice el número de espacios vacíos generados por los ajustes requeridos en el proceso cuando existen cambios de modelo y/o cambios de modelo.
- Para cada variante:
 - Diseñar un modelo matemático que pueda determinar cuáles son las soluciones que minimizan los ajustes por cambio de color y/o de modelo considerando según sea la variante, solo los ajustes o los espacios vacíos generados por los ajustes.
 - Programar e implementar una herramienta computacional capaz de resolver el problema de forma exacta.
 - Desarrollar y evaluar heurísticas que minimicen el problema mediante experimentación.
 - Evaluar el impacto de los factores involucrados en la complejidad y valor de la solución de cada modelo propuesto.

Existen otros contextos en los cuales es posible aplicar los modelos desarrollados. Principalmente aquellos que utilicen sistemas de circuito cerrado y requieran de tiempos de preparación en los cambios de producto. Aunque también está relacionado con el sector automotriz, se puede mencionar como ejemplo la fabricación de asientos para automóviles.

6.2 LINEAS FUTURAS DE INVESTIGACION

Se propone la exploración del efecto en los inventarios por una mejor secuenciación. Es decir, cuando los inventarios temporales existen para cubrir faltantes por la secuenciación del sistema, como se pueden optimizar.

Las diferencias en el coste de cambiar de un modelo específico a otro específico como influyen en la complejidad de los modelos diseñados.

Las diferencias en el coste de cambiar de un color específico a otro específico como influyen en la complejidad de los modelos diseñados.

Si la demanda fuese dinámica y el programa cambiase antes de concluir la ejecución de una secuencia, como se puede optimizar el resultado final minimizando cambios o espacios vacíos según sea el caso.

Determinar la aplicación de los modelos desarrollados en esta Tesis a otros sistemas multi-objetivo con características similares.

7 BIBLIOGRAFIA

- (1) Staebelin T, Aoki K. Planning and scheduling in the automotive industry: A comparison of industrial practice at German and Japanese makers. *International Journal of Production Economics* 2015; 162(0):258-272.
- (2) Garcia-Sabater JP, Andres C, Miralles C. An Application Oriented Approach for Scheduling a Production Line with Two Dimensional Setups. *Proceedings of the 11th International Workshop on Project Management and Scheduling* 2008; 1(1):90-93.
- (3) Ganguly S, Laguna M. Modeling and solving a closed-loop scheduling problem with two types of setups. *IIE Transactions* 2014; 47(8):880-891.
- (4) Yang S, Riggs RJ, Hu SJ. Modeling and analysis of closed loop manufacturing systems using parameter coupling. *Journal of Manufacturing Systems* 2013; 32(4):817-824.
- (5) Li J, Blumenfeld E, Huang N, Alden M. Throughput analysis of production systems: recent advances and future topics. *Int J Prod Res* 2009; 47(14):3823-3851.
- (6) Muth EJ. Analysis of Closed-Loop Conveyor Systems, the Discrete Flow Case. *A I I E Transactions* 1974; 6(1):73-83.
- (7) Muth EJ. Analysis of Closed-Loop Conveyor Systems. *A I I E Transactions* 1972; 4(2):134-143.
- (8) Smith JS. Survey on the use of simulation for manufacturing system design and operation. *Journal of Manufacturing Systems* 2003; 22(2):157-171.
- (9) Investigating distributed simulation at the Ford Motor Company. *Distributed Simulation and Real-Time Applications, 2005. DS-RT 2005 Proceedings. Ninth IEEE International Symposium on*; 2005.
- (10) Dallery Y, Gershwin S. Manufacturing flow line systems: a review of models and analytical results. *Queueing Syst* 1992; 12(1-2):3-94.
- (11) Koren Y. *The Global Manufacturing Revolution: Product-Process-Business Integration and Reconfigurable Systems*. Wiley; 2010.
- (12) Yang S, Wu C, Hu SJ. Modeling and analysis of multi-stage transfer lines with unreliable machines and finite buffers. *Annals of Operations Research* 2000; 93(1-4):405-421.
- (13) Liu J, Yang S, Wu A, Hu SJ. Multi-state throughput analysis of a two-stage manufacturing system with parallel unreliable machines and a finite buffer. *European Journal of Operational Research* 2012; 219(2):296-304.
- (14) Kharoufeh JP, Cochran JJ, Cox LA, Keskinocak P, Kharoufeh JP, Smith JC. Level-Dependent Quasi-Birth-and-Death Processes. *Wiley Encyclopedia of Operations Research and Management Science*. John Wiley & Sons, Inc.; 2010.
- (15) Fadiloglu MM, Yeralan S. Models of production lines as quasi-birth-death processes. *Mathematical and Computer Modelling* 2002; 35(7):913-930.
- (16) Gershwin S, Burman M. A decomposition method for analyzing inhomogeneous assembly/disassembly systems. *Annals of Operations Research* 2000; 93(1-4):91-115.

- (17) Li S, Wang H, Hu SJ, Lin YT, Abell JA. Automatic generation of assembly system configuration with equipment selection for automotive battery manufacturing. *Journal of Manufacturing Systems* 2011; 30(4):188-195.
- (18) Um I, Cheon H, Lee H. The simulation design and analysis of a Flexible Manufacturing System with Automated Guided Vehicle System. *Journal of Manufacturing Systems* 2009; 28(4):115-122.
- (19) Koren Y, Shpitalni M. Design of reconfigurable manufacturing systems. *Journal of Manufacturing Systems* 2010; 29(4):130-141.
- (20) Huang N. Mean station reliabilities cause throughput overestimates in production system design. *Journal of Manufacturing Systems* 2012; 31(2):184-194.
- (21) Huang SH, Dismukes JP, Shi J, Su Q, Wang G, Razzak MA et al. Manufacturing system modeling for productivity improvement. *Journal of Manufacturing Systems* 2002; 21(4):249-259.
- (22) Bastani AS. Analytical solution of closed-loop conveyor systems with discrete and deterministic material flow. *European Journal of Operational Research* 1988; 35(2):187-192.
- (23) Narahari Y, Khan LM. Modeling reentrant manufacturing systems with inspection stations. *Journal of Manufacturing Systems* 1996; 15(6):367-378.
- (24) Bouhchouch A, Frein Y, Dallery Y. Analysis of a closed-loop manufacturing system with finite buffers. *Appl Stochastic Models Data Anal* 1993; 9(2):111-125.
- (25) Frein Y, Commault C, Dallery Y. Modeling and analysis of closed-loop production lines with unreliable machines and finite buffers. *IIE Transactions* 1996; 28(7):545-554.
- (26) Gershwin SB, Werner LM. An approximate analytical method for evaluating the performance of closed-loop flow systems with unreliable machines and finite buffers. *Int J Prod Res* 2007; 45(14):3085-3111.
- (27) Li N, Zheng L, Li QL. Performance analysis of two-loop closed production systems. *Computers & Operations Research* 2009; 36(1):119-134.
- (28) An analytical model for conveyor based AMHS in semiconductor wafer fabs. *Simulation Conference, 2008. WSC 2008. Winter; 2008.*
- (29) Pettinato JS, Pillai D. Technology decisions to minimize 450-mm wafer size transition risk. *Ieee Transactions on Semiconductor Manufacturing* 2005; 18(4):501-509.
- (30) Maggio N, Matta A, Gershwin SB, Tolio T. A decomposition approximation for three-machine closed-loop production systems with unreliable machines, finite buffers and a fixed population. *IIE Transactions* 2009; 41(6):562-574.
- (31) Lagershausen S, Manitz M, Tempelmeier H. Performance analysis of closed-loop assembly lines with general processing times and finite buffer spaces. *IIE Transactions* 2012; 45(5):502-515.
- (32) Ferreira L, Gomez E, Lourido G, Quintas JDg, Tjahjono B. Analysis and optimisation of a network of closed-loop automobile assembly line using simulation. *Int J Adv Manuf Technol* 2012; 59(1-4):351-366.
- (33) Ríos S. Some inventory problems for many items with stochastic dependent demand. *TDE* 1962; 13(3):173-181.

- (34) Allahverdi A, Gupta JND, Aldowaisan T. A review of scheduling research involving setup considerations. *Omega* 1999; 27(2):219-239.
- (35) Allahverdi A, Ng CT, Cheng TCE, Kovalyov MY. A survey of scheduling problems with setup times or costs. *European Journal of Operational Research* 2008; 187(3):985-1032.
- (36) Krajewski LJ, King BE, Ritzman LP, Wong DS. KANBAN, MRP, AND SHAPING THE MANUFACTURING ENVIRONMENT. *Management Science* 1987; 33(1):39-57.
- (37) Flynn BB. Repetitive lots: The use of a sequence-dependent set-up time scheduling procedure in group technology and traditional shops. *J Oper Manage* 1987; 7(1-2):203-216.
- (38) Flynn BB. The effects of setup time on output capacity in cellular manufacturing. *Int J Prod Res* 1987; 25(12):1761.
- (39) Shingo S. A revolution in manufacturing: the SMED system. Productivity Press; 1985.
- (40) Chen B, Ye Y, Zhang J. Lot-sizing scheduling with batch setup times. *J Scheduling* 2006; 9(3):299-310.
- (41) Allahverdi A, Soroush HM. The significance of reducing setup times/setup costs. *European Journal of Operational Research* 2008; 187(3):978-984.
- (42) Potts CN, Kovalyov Mikhail Y. Scheduling with batching a review. *European Journal of Operational Research* 120, 228-249. 2000.
- (43) Graham RL, Lawler EL, Lenstra JK, Kan AHGR. Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey. In: Hammer PL, editor. *Annals of Discrete Mathematics. Discrete Optimization II Proceedings of the Advanced Research Institute on Discrete Optimization and Systems Applications of the Systems Science Panel of NATO and of the Discrete Optimization Symposium co-sponsored by IBM Canada and SIAM Banff, Aha. and Vancouver. Volume 5 ed. Elsevier; 1979. 287-326.*
- (44) Cai Y, Kutanoglu E, Hasenbein J, Qin J. Single-machine scheduling with advanced process control constraints. *J Scheduling* 2010;1-15.
- (45) Chrétienne P, Hazır O, Kedad-Sidhoum S. Integrated batch sizing and scheduling on a single machine. *J Scheduling* 2011;1-15.
- (46) Leus R, Herroelen W. Scheduling for stability in single-machine production systems. *J Scheduling* 2007; 10(3):223-235.
- (47) Yuan J, Liu Z, Ng C, Cheng T. Single machine batch scheduling problem with family setup times and release dates to minimize makespan. *J Scheduling* 2006; 9(6):499-513.
- (48) Cheng TCE, Sin CCS. A state-of-the-art review of parallel-machine scheduling research. *European Journal of Operational Research* 1990; 47(3):271-292.
- (49) Pfund M, Fowler JW, Gupta JND. A survey of algorithms for single and multi-objective unrelated parallel-machine deterministic scheduling problems. *J Chin Inst Ind Eng* 2004; 21(3):230-241.
- (50) Eren T, G++ner E. A literature survey for multicriteria scheduling problems on single and parallel machines. *J Fac Eng Archit Gazi Uni* 2002; 17(4):37-69.

- (51) Mokotoff E. Parallel machine scheduling problems: A survey. *Asia Pac J Oper Res* 2001; 18(2):193-242.
- (52) Koulamas C. Polynomially solvable total tardiness problems: Review and extensions. *Omega* 1997; 25(2):235-239.
- (53) Koulamas C. Total tardiness problem: review and extensions. *Oper Res* 1994; 42(6):1025-1041.
- (54) Kravchenko SA, Werner F. Minimizing total tardiness on parallel machines with preemptions. *J Scheduling*. In press 2010.
- (55) Lamothe J, Marmier F, Dupuy M, Gaborit P, Dupont L. Scheduling rules to minimize total tardiness in a parallel machine problem with setup and calendar constraints. *Computers & Operations Research* 2012; 39(6):1236-1244.
- (56) Weng MX, Lu J, Ren H. Unrelated parallel machine scheduling with setup consideration and a total weighted completion time objective. *International Journal of Production Economics* 2001; 70(3):215-226.
- (57) Pinedo M. *Scheduling, theory, algorithms and systems*. 3rd Ed. ed. 2008.
- (58) Vairaktarakis GL, Cai X. The value of processing flexibility in multipurpose machines. *IIE Trans* 2003; 35(8):763-774.
- (59) Centeno G, Armacost RL. Minimizing makespan on parallel machines with release time and machine eligibility restrictions. *Int J Prod Res* 2004; 42(6):1243-1256.
- (60) Bank J, Werner F. Heuristic algorithms for unrelated parallel machine scheduling with a common due date, release dates, and linear earliness and tardiness penalties. *Mathematical and Computer Modelling* 2002; 33(4-5):363-383.
- (61) Koulamas C, Kyparisis GJ. Scheduling on uniform parallel machines to minimize maximum lateness. *Operations Research Letters* 2000; 26(4):175-179.
- (62) Koulamas C, Kyparisis GJ. A modified LPT algorithm for the two uniform parallel machine makespan minimization problem. *European Journal of Operational Research* 2009; 196(1):61-68.
- (63) Liaw CF, Lin YK, Cheng CY, Chen M. Scheduling unrelated parallel machines to minimize total weighted tardiness. *Computers & Operations Research* 2003; 30(12):1777-1789.
- (64) Rocha PL, Ravetti MnGm, Mateus GR, Pardalos PM. Exact algorithms for a scheduling problem with unrelated parallel machines and sequence and machine-dependent setup times. *Computers & Operations Research* 2008; 35(4):1250-1264.
- (65) Webster S, Azizoglu M. Dynamic programming algorithms for scheduling parallel machines with family setup times. *Computers & Operations Research* 2001; 28(2):127-137.
- (66) Lee YH, Pinedo M. Scheduling jobs on parallel machines with sequence-dependent setup times. *European Journal of Operational Research* 1997; 100(3):464-474.
- (67) Vepsalainen APJ, Morton TE. PRIORITY RULES FOR JOB SHOPS WITH WEIGHTED TARDINESS COSTS. *Management Science* 1987; 33(8):1035-1047.

- (68) Park Y, Kim S, Lee YH. Scheduling jobs on parallel machines applying neural network and heuristic rules. *Computers & Industrial Engineering* 2000; 38(1):189-202.
- (69) Eom DH, Shin HJ, Kwun IH, Shim JK, Kim SS. Scheduling Jobs on Parallel Machines with Sequence-Dependent Family Set-up Times. *The International Journal of Advanced Manufacturing Technology* 2002; 19(12):926-932.
- (70) Kim DW, Kim KH, Jang W, Frank Chen F. Unrelated parallel machine scheduling with setup times using simulated annealing. *Robotics and Computer-Integrated Manufacturing* 2006; 18(3-4):223-231.
- (71) Kim DW, Na DG, Frank Chen F. Unrelated parallel machine scheduling with setup times and a total weighted tardiness objective. *Robotics and Computer-Integrated Manufacturing* 2002; 19(1-2):173-181.
- (72) Chinyao L. Simulated annealing heuristic for flow shop scheduling problems with unrelated parallel machines. *Computers & Operations Research* 2005; 32(8):2013-2025.
- (73) Akkiraju R, Keskinocak P, Murthy S, Wu F. An Agent-Based Approach for Scheduling Multiple Machines. *Applied Intelligence* 2001; 14(2):135-144.
- (74) Ravetti MG, Mateus GR, Rocha PL, Pardalos PM. A scheduling problem with unrelated parallel machines and sequence dependent setups. *Int J Oper Res* 2007; 2(4):380-399.
- (75) De Paula MR, Ravetti MG, Mateus GR, Pardalos PM. Solving parallel machines scheduling problems with sequence-dependent setup times using variable neighbourhood search. *IMA J Manage Math* 2007; 18(2):101-115.
- (76) Tamaki H, Hasegawa Y, Kozasa J, Araki M. Application of search methods to scheduling problem in plastics forming plant: a binary representation approach. *Proc IEEE Conf Decis Control* 1993; 4(Piscataway, NJ, United States):3845-3850.
- (77) Chen JF, Wu TH. Total tardiness minimization on unrelated parallel machine scheduling with auxiliary equipment constraints. *Omega* 2006; 34(1):81-89.
- (78) Ziaee M. General flowshop scheduling problem with the sequence dependent setup times: A heuristic approach. *Information Sciences* 2013; 251:126-135.
- (79) Allahverdi A. The third comprehensive survey on scheduling problems with setup times/costs. *European Journal of Operational Research* 2015.
- (80) Potts CN, Shmoys DB, Williamson DP. Permutation vs. non-permutation flow shop schedules. *Operations Research Letters* 1991; 10(5):281-284.
- (81) Tandon M, Cummings PT, LeVan MD. Flowshop sequencing with non-permutation schedules. *Computers & Chemical Engineering* 1991; 15(8):601-607.
- (82) Pugazhendhi S, Thiagarajan S, Rajendran C, Anantharaman N. Performance enhancement by using non-permutation schedules in flowline-based manufacturing systems. 2002; 44:133-157.
- (83) Koulamas C. A new constructive heuristic for the flowshop scheduling problem. *European Journal of Operational Research* 1998; 105(1):66-71.
- (84) Wang X, Cheng TCE. An approximation scheme for two-machine flowshop scheduling with setup times and an availability constraint. *Computers & Operations Research* 2007; 34(10):2894-2901.

- (85) Wang X, Edwin Cheng TC. Heuristics for two-machine flowshop scheduling with setup times and an availability constraint. *Computers & Operations Research* 2007; 34(1):152-162.
- (86) A bi-criteria optimization model and algorithm for scheduling in a real-world flow shop with setup times. *IEEE*; 2008.
- (87) Hall NG, Sriskandarajah C. A survey of machine scheduling problems with blocking and no-wait in process. *Oper Res* 1996; 44(3):510-525.
- (88) Aldowaisan T, Allahverdi A. Minimizing total tardiness in no-wait flowshops. *Foundations of Computing and Decision Sciences* 2012; 37(3):149-162.
- (89) Ding JY, Song S, Gupta JND, Zhang R, Chiong R, Wu C. An improved iterated greedy algorithm with a Tabu-based reconstruction strategy for the no-wait flowshop scheduling problem. *Applied Soft Computing* 2015; 30:604-613.
- (90) Nagano MS, da Silva AA, Lorena LAN. A new evolutionary clustering search for a no-wait flow shop problem with set-up times. *Engineering Applications of Artificial Intelligence* 2012; 25(6):1114-1120.
- (91) Nagano MS, da Silva AA, Nogueira Lorena LA. An evolutionary clustering search for the no-wait flow shop problem with sequence dependent setup times. *Expert Systems with Applications* 2014; 41(8):3628-3633.
- (92) Allahverdi A, Aydilek H. Total completion time with makespan constraint in no-wait flowshops with setup times. *European Journal of Operational Research* 2014; 238(3):724-734.
- (93) Naderi B, Gohari S, Yazdani M. Hybrid flexible flowshop problems: Models and solution methods. *Applied Mathematical Modelling* 2014; 38(24):5767-5780.
- (94) Naderi B, Ruiz Rn. The distributed permutation flowshop scheduling problem. *Computers & Operations Research* 2010; 37(4):754-768.
- (95) Fattahi P, Hosseini SMH, Jolai F, Tavakkoli-Moghaddam R. A branch and bound algorithm for hybrid flow shop scheduling problem with setup time and assembly operations. *Applied Mathematical Modelling* 2014; 38(1):119-134.
- (96) Huang RH, Yang CL, Huang YC. No-wait two-stage multiprocessor flow shop scheduling with unit setup. *The International Journal of Advanced Manufacturing Technology* 2009; 44(9-10):921-927.
- (97) Lee CY, Cheng TCE, Lin BMT. Minimizing the Makespan in the 3-Machine Assembly-Type Flowshop Scheduling Problem. *Management Science* 1993; 39(5):616-625.
- (98) Navaei J, Fatemi Ghomi SMT, Jolai F, Mozdgir A. Heuristics for an assembly flowshop with non-identical assembly machines and sequence dependent setup times to minimize sum of holding and delay costs. *Computers & Operations Research* 2014; 44:52-65.
- (99) Allahverdi A, Al-Anzi FS. The two-stage assembly flowshop scheduling problem with bicriteria of makespan and mean completion time. *The International Journal of Advanced Manufacturing Technology* 2008; 37(1-2):166-177.
- (100) Allahverdi A, Al-Anzi FS. The two-stage assembly scheduling problem to minimize total completion time with setup times. *Computers & Operations Research* 2009; 36(10):2740-2747.

- (101) Tian Y, Liu D, Yuan D, Wang K. A discrete PSO for two-stage assembly scheduling problem. *The International Journal of Advanced Manufacturing Technology* 2013; 66(1-4):481-499.
- (102) Proth JM. Scheduling: New trends in industrial environment. *Annu Rev Control* 2007; 31(1):157-166.
- (103) Wang WL, Wang HY, Zhao YW, Zhang LP, Xu XL. Parallel machine scheduling with splitting jobs by a hybrid differential evolution algorithm. *Computers & Operations Research* 2013; 40(5):1196-1206.
- (104) Aydilek H, Allahverdi A. Heuristics for no-wait flowshops with makespan subject to mean completion time. *Applied Mathematics and Computation* 2012; 219(1):351-359.
- (105) Gu J, Gu X, Gu M. A novel parallel quantum genetic algorithm for stochastic job shop scheduling. *Journal of Mathematical Analysis and Applications* 2009; 355(1):63-81.
- (106) Wiers VCS. A review of the applicability of OR and AI scheduling techniques in practice. *Omega* 1997; 25(2):145-153.
- (107) Cowling P, Johansson M. Using real time information for effective dynamic scheduling. *European Journal of Operational Research* 2002; 139(2):230-244.
- (108) Stoop PM, Wiers VCS. The complexity of scheduling in practice. *Int Jnl of Op & Prod Mngemnt* 1996; 16(10):37-53.
- (109) Vieira G, Herrmann J, Lin E. Rescheduling Manufacturing Systems: A Framework of Strategies, Policies, and Methods. *J Scheduling* 2003; 6(1):39-62.
- (110) Wu SD, Byeon ES, Storer RH. A Graph-Theoretic Decomposition of the Job Shop Scheduling Problem to Achieve Scheduling Robustness. *Oper Res* 1999; 47(1):113-124.
- (111) MacCarthy BL, LIU JIYI. Addressing the gap in scheduling research: a review of optimization and heuristic methods in production scheduling. *Int J Prod Res* 1993; 31(1):59-79.
- (112) Azfar KRW, Khan N, Gabriel HF. Performance Measurement: A Conceptual Framework for Supply Chain Practices. *Procedia - Social and Behavioral Sciences* 2014; 150(0):803-812.
- (113) Levner E, Kats V, Alcaide L+ipez de Pablo D, Cheng TCE. Complexity of cyclic scheduling problems: A state-of-the-art survey. *Computers & Industrial Engineering* 2010; 59(2):352-361.
- (114) Bozejko W, Uchrowski M, Wodecki M. Block approach to the cyclic flow shop scheduling. *Computers & Industrial Engineering* 2015; 81(0):158-166.
- (115) Lee TE, Posner ME. Performance Measures and Schedules in Periodic Job Shops. *Oper Res* 1997; 45(1):72-91.
- (116) Roundy R. Cyclic Schedules for Job Shops with Identical Jobs. *Mathematics of OR* 1992; 17(4):842-865.
- (117) Matsuo H, Shang JS, Sullivan RS. A Crane Scheduling Problem in a Computer-Integrated Manufacturing Environment. *Management Science* 1991; 37(5):587-606.
- (118) Hanen C. Study of a NP-hard cyclic scheduling problem: The recurrent job-shop. *European Journal of Operational Research* 1994; 72(1):82-101.

- (119) Hall NG, Kamoun H, Sriskandarajah C. Scheduling in robotic cells: Complexity and steady state analysis. *European Journal of Operational Research* 1998; 109(1):43-65.
- (120) Karabati S, Kouvelis P. Cyclic scheduling in flow lines: Modeling observations, effective heuristics and a cycle time minimization procedure. *Naval Research Logistics* 1996; 43(2):211-231.
- (121) Sawik T. Batch versus cyclic scheduling of flexible flow shops by mixed-integer programming. *Int J Prod Res* 2011; 50(18):5017-5034.
- (122) Brucker P, Burke EK, Groenemeyer S. A branch and bound algorithm for the cyclic job-shop problem with transportation. *Computers & Operations Research* 2012; 39(12):3200-3214.
- (123) Rocha M, Oliveira J, Carravilla M. Cyclic staff scheduling: optimization models for some real-life problems. *J Sched* 2013; 16(2):231-242.
- (124) Sawik T. A mixed integer program for cyclic scheduling of flexible flow lines. *Bulletin of the Polish Academy of Sciences: Technical Sciences. bpasts* 62, 121. 2014. 29-5-2015.
- (125) Shirvani N, Ruiz Rn, Shadrokh S. Cyclic scheduling of perishable products in parallel machine with release dates, due dates and deadlines. *International Journal of Production Economics* 2014; 156(0):1-12.
- (126) Kaplan S, Rabadi G. Exact and heuristic algorithms for the aerial refueling parallel machine scheduling problem with due date-to-deadline window and ready times. *Computers & Industrial Engineering* 2012; 62(1):276-285.
- (127) Hennen M, Voll P, Bardow A. An Adaptive Normal Constraint Method for Bi-Objective Optimal Synthesis of Energy Systems. In: Ji+Ö+j Jarom+jr Kleme+i PSVaPYL, editor. *Computer Aided Chemical Engineering. 24th European Symposium on Computer Aided Process Engineering. Volume 33* ed. Elsevier; 2014. 1279-1284.
- (128) Gen M, Lin L. Multiobjective evolutionary algorithm for manufacturing scheduling problems: state-of-the-art survey. *Journal of Intelligent Manufacturing* 2014; 25(5):849-866.
- (129) Newman P. Edgeworth, Francis Ysidro (1845ΓÇô1926). In: Wright JD, editor. *International Encyclopedia of the Social & Behavioral Sciences (Second Edition)*. Oxford: Elsevier; 2015. 113-116.
- (130) Vilfredo Pareto on Labor: A Critical Re-examination. Taylor & Francis; 2015.
- (131) Miller RL. *Microeconom+ja Moderna*, Ed. Harla, M+@xico 1995.
- (132) Blasco X, Herrero JM, Sanchis J, Mart+jenez M. A new graphical visualization of n-dimensional Pareto front for decision-making in multiobjective optimization. *Information Sciences* 2008; 178(20):3908-3924.
- (133) Harjunkski I, P+Ârn R, Westerlund T, Skrifvars H. Different strategies for solving bilinear integer non-linear programming problems with convex transformations. *Computers & Chemical Engineering* 1997; 21, Supplement(0):S487-S492.
- (134) Jadidi O, Zolfaghari S, Cavalieri S. A new normalized goal programming model for multi-objective problems: A case of supplier selection and order allocation. *International Journal of Production Economics* 2014; 148(0):158-165.

- (135) Jadidi O, Cavalieri S, Zolfaghari S. An improved multi-choice goal programming approach for supplier selection problems. *Applied Mathematical Modelling* 2015; 39(14):4213-4222.
- (136) Marler RT, Kim CH, Arora JS. System identification of simplified crash models using multi-objective optimization. *Computer Methods in Applied Mechanics and Engineering* 2006; 195(33):4383-4395.
- (137) Alves MJ, Clímaco J. A review of interactive methods for multiobjective integer and mixed-integer programming. *European Journal of Operational Research* 2007; 180(1):99-115.

ANEXOS

ANEXO 1

ESTADISTICOS DESCRIPTIVOS DE EXPERIMENTOS CBBSSP-SH

1.1 Análisis Mono-Objetivo

Demanda = 100, Bloques = 20

Tabla de frecuencias:

Estadísticos

NCCoIHor							
Dem 100	Bloques 20	5 Mod	5 Col	N	Válidos	Perdidos	
							8505
							0
			10 Col	N	Válidos		8505
					Perdidos		0
			20 Col	N	Válidos		8505
					Perdidos		0
		10 Mod	5 Col	N	Válidos		8505
					Perdidos		0
			10 Col	N	Válidos		8505
					Perdidos		0
			20 Col	N	Válidos		8505
					Perdidos		0
		20 Mod	5 Col	N	Válidos		8505
					Perdidos		0
			10 Col	N	Válidos		8505
					Perdidos		0
			20 Col	N	Válidos		8505
					Perdidos		0

ANALISIS VARIABLE DEPENDIENTE NCCoIHor:

Estadísticos descriptivos:

Estadísticos descriptivos

DemTot	Bloques	Modelos	Colores	N	Mínimo	Máximo	Media	Desv. típ.	
Dem 100	Bloques 20	5 Mod	5 Col	NCColHor	8505	4	93	21.05	11.622
				N válido (según lista)	8505				
			10 Col	NCColHor	8505	9	99	33.15	14.289
				N válido (según lista)	8505				
			20 Col	NCColHor	8505	19	99	38.64	13.975
				N válido (según lista)	8505				
		10 Mod	5 Col	NCColHor	8505	4	98	26.54	13.448
				N válido (según lista)	8505				
			10 Col	NCColHor	8505	9	99	32.35	13.359
				N válido (según lista)	8505				
			20 Col	NCColHor	8505	19	99	38.65	14.156
				N válido (según lista)	8505				
		20 Mod	5 Col	NCColHor	8505	4	90	25.98	12.615
				N válido (según lista)	8505				
			10 Col	NCColHor	8505	9	99	32.10	13.139
				N válido (según lista)	8505				
20 Col	NCColHor		8505	19	60	32.19	10.235		
	N válido (según lista)		8505						

Análisis de Varianza de factor "Listas":

ANOVA

NCColHor				Suma de cuadrados	gl	Media cuadrática	F	Sig.			
Dem 100	Bloques 20	5 Mod	5 Col	Inter-grupos	20704.598	8	2588.075	19.495	.000		
				Intra-grupos	1127910.974	8496	132.758				
				Total	1148615.572	8504					
			10 Col	Inter-grupos	14967.943	8	1870.993			9.235	.000
				Intra-grupos	1721252.974	8496	202.596				
				Total	1736220.916	8504					
		20 Col	Inter-grupos	7533.616	8	941.702	4.839	.000			
			Intra-grupos	1653285.888	8496	194.596					
			Total	1660819.504	8504						
		10 Mod	5 Col	Inter-grupos	34342.148	8			4292.768	24.257	.000
				Intra-grupos	1503535.875	8496			176.970		
				Total	1537878.023	8504					
			10 Col	Inter-grupos	12873.947	8	1609.243	9.086	.000		
				Intra-grupos	1504708.910	8496	177.108				
				Total	1517582.857	8504					
		20 Col	Inter-grupos	3746.368	8	468.296	2.340			.017	
			Intra-grupos	1700470.408	8496	200.150					
			Total	1704216.777	8504						
		20 Mod	5 Col	Inter-grupos	40385.384	8		5048.173	32.669		.000
				Intra-grupos	1312847.913	8496		154.525			
				Total	1353233.297	8504					
			10 Col	Inter-grupos	14495.007	8	1811.876	10.590		.000	
				Intra-grupos	1453576.277	8496	171.089				
				Total	1468071.285	8504					
20 Col	Inter-grupos	564.167	8	70.521	.673	.716					
	Intra-grupos	890331.395	8496	104.794							
	Total	890895.562	8504								

Pruebas Post-Hoc, Tukey B :

A continuación se presenta el factor Listas con todos los casos para las distintas cantidades de modelos y colores:

a) 5 Modelos y 5 Colores

DemTot=Dem 100Bloques=Bloques 20Modelos=5 ModColores=5 Col

Tukey B^a

Listas	N	Subconjunto para alfa = .05					
		1	2	3	4	5	6
List 7	945	18.42					
List 5	945	19.39	19.39				
List 2	945		19.89	19.89			
List 4	945			20.88	20.88		
List 3	945			20.93	20.93		
List 8	945				21.51		
List 6	945				21.82	21.82	
List 1	945					23.06	23.06
List 9	945						23.51

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 945.000.

b) 5 Modelos y 10 Colores

DemTot=Dem 100Bloques=Bloques 20Modelos=5
ModColores=10 Col

Tukey B^a

Listas	N	Subconjunto para alfa = .05		
		1	2	3
List 7	945	31.24		
List 3	945	31.96	31.96	
List 4	945	32.30	32.30	
List 8	945	32.48	32.48	
List 5	945		33.16	
List 6	945		33.27	
List 2	945		33.30	
List 9	945			35.25
List 1	945			35.43

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 945.000.

c) 5 Modelos y 20 Colores

**DemTot=Dem 100Bloques=Bloques 20Modelos=5
ModColores=20 Col**

Tukey B^a

Listas	N	Subconjunto para alfa = .05		
		1	2	3
List 3	945	37.50		
List 7	945	37.58		
List 4	945	37.89	37.89	
List 8	945	37.90	37.90	
List 6	945	38.49	38.49	38.49
List 5	945	39.14	39.14	39.14
List 2	945	39.31	39.31	39.31
List 1	945		39.69	39.69
List 9	945			40.25

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 945.000.

d) 10 Modelos y 5 Colores

**DemTot=Dem 100Bloques=Bloques 20Modelos=10
ModColores=5 Col**

Tukey B^a

Listas	N	Subconjunto para alfa = .05			
		1	2	3	4
List 7	945	23.35			
List 5	945	24.21			
List 2	945	24.43			
List 3	945		26.31		
List 4	945		26.99	26.99	
List 8	945		27.59	27.59	
List 6	945		28.02	28.02	28.02
List 1	945			28.43	28.43
List 9	945				29.57

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 945.000.

e) 10 Modelos y 10 Colores

DemTot=Dem 100Bloques=Bloques 20Modelos=10 ModColores=10 Col

Tukey B^a

Listas	N	Subconjunto para alfa = .05				
		1	2	3	4	5
List 7	945	30.43				
List 5	945	31.09	31.09			
List 2	945	31.33	31.33	31.33		
List 3	945	31.92	31.92	31.92	31.92	
List 8	945		32.51	32.51	32.51	
List 4	945		32.66	32.66	32.66	
List 6	945			33.07	33.07	33.07
List 1	945				33.49	33.49
List 9	945					34.63

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 945.000.

f) 10 Modelos y 20 Colores

**DemTot=Dem 100Bloques=Bloques
20Modelos=10 ModColores=20 Col**

Tukey B^a

Listas	N	Subconjunto para alfa = .05	
		1	2
List 3	945	37.58	
List 7	945	37.97	
List 4	945	38.34	38.34
List 5	945	38.53	38.53
List 2	945	38.63	38.63
List 8	945	38.73	38.73
List 6	945	38.96	38.96
List 1	945	39.11	39.11
List 9	945		40.04

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 945.000.

g) 20 Modelos y 5 Colores

**DemTot=Dem 100Bloques=Bloques 20Modelos=20
ModColores=5 Col**

Tukey B^a

Listas	N	Subconjunto para alfa = .05			
		1	2	3	4
List 7	945	22.90			
List 5	945	23.05			
List 2	945	23.39			
List 3	945		25.65		
List 4	945		27.08	27.08	
List 8	945			27.43	27.43
List 1	945			27.64	27.64
List 6	945			27.74	27.74
List 9	945				28.92

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 945.000.

h) 20 Modelos y 10 Colores

**DemTot=Dem 100Bloques=Bloques 20Modelos=20
ModColores=10 Col**

Tukey B^a

Listas	N	Subconjunto para alfa = .05			
		1	2	3	4
List 7	945	30.22			
List 5	945	30.83	30.83		
List 2	945	30.91	30.91		
List 3	945	31.04	31.04		
List 4	945		32.42	32.42	
List 8	945			32.93	32.93
List 6	945			33.05	33.05
List 1	945			33.29	33.29
List 9	945				34.23

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 945.000.

i) 20 Modelos y 20 Colores

DemTot=Dem
 100Bloques=Bloques
 20Modelos=20 ModColores=20 Col

Tukey B^a

Listas	N	Subconjunto para alfa = .05
		1
List 7	945	31.74
List 3	945	31.96
List 4	945	32.06
List 8	945	32.19
List 5	945	32.23
List 6	945	32.23
List 2	945	32.30
List 1	945	32.32
List 9	945	32.73

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 945.000.

Análisis de Varianza de factor "Algoritmos":

ANOVA

NCColHor

DemTot	Bloques	Modelos	Colores		Suma de cuadrados	gl	Media cuadrática	F	Sig.
Dem 100	Bloques 20	5 Mod	5 Col	Inter-grupos	838578.537	26	32253.021	881.963	.000
				Intra-grupos	310037.035	8478	36.570		
				Total	1148615.572	8504			
		10 Col	Inter-grupos	1241810.275	26	47761.934	819.007	.000	
			Intra-grupos	494410.641	8478	58.317			
			Total	1736220.916	8504				
	20 Col	Inter-grupos	920575.561	26	35406.752	405.513	.000		
		Intra-grupos	740243.943	8478	87.314				
		Total	1660819.504	8504					
	10 Mod	5 Col	5 Col	Inter-grupos	1083873.578	26	41687.445	778.464	.000
				Intra-grupos	454004.444	8478	53.551		
				Total	1537878.023	8504			
		10 Col	Inter-grupos	1043379.994	26	40130.000	717.461	.000	
			Intra-grupos	474202.863	8478	55.933			
			Total	1517582.857	8504				
	20 Col	Inter-grupos	827881.685	26	31841.603	308.048	.000		
		Intra-grupos	876335.092	8478	103.366				
		Total	1704216.777	8504					
20 Mod	5 Col	5 Col	Inter-grupos	918271.989	26	35318.153	688.400	.000	
			Intra-grupos	434961.308	8478	51.305			
			Total	1353233.297	8504				
	10 Col	Inter-grupos	885256.529	26	34048.328	495.289	.000		
		Intra-grupos	582814.756	8478	68.744				
		Total	1468071.285	8504					
20 Col	Inter-grupos	477418.425	26	18362.247	376.502	.000			
	Intra-grupos	413477.137	8478	48.771					
	Total	890895.562	8504						

Pruebas Post-Hoc, Tukey B :

A continuación se presenta el factor Algoritmos con todos los casos para las distintas cantidades de modelos y colores:

a) 5 Modelos y 5 Colores

DemTot=Dem 100Bloques=Bloques 20Modelos=5 ModColores=5 Col

Tukey B ^a		Subconjunto para alfa = .05														
Algoritmos	N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Alg 10	315	6.49														
Alg 12	315		8.12													
Alg 33	315			10.62												
Alg 23	315			10.69												
Alg 13	315			11.18	11.18											
Alg 11	315				12.39											
Alg 34	315					14.43										
Alg 14	315					14.52										
Alg 24	315					14.65										
Alg 9	315					14.81										
Alg 32	315						17.13									
Alg 16	315						17.28	17.28								
Alg 30	315							18.59	18.59							
Alg 22	315							18.69	18.69							
Alg 15	315								19.20							
Alg 20	315									21.43						
Alg 36	315										22.89					
Alg 17	315										24.01	24.01				
Alg 31	315										24.34	24.34				
Alg 35	315											25.35	25.35			
Alg 37	315											25.36	25.36			
Alg 27	315												29.59			
Alg 26	315												30.56			
Alg 25	315												30.82			
Alg 21	315													34.18		
Alg 29	315														44.54	
Alg 19	315															46.36

Se muestran las medias para los grupos en los subconjuntos homogéneos.
a. Usa el tamaño muestral de la media armónica = 315.000.

b) 5 Modelos y 10 Colores

DemTot=Dem 100Bloques=Bloques 20Modelos=5 ModColores=10 Col

Tukey B ^a		Subconjunto para alfa = .05															
Algoritmos	N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Alg 10	315	12.00															
Alg 12	315		15.39														
Alg 13	315			18.77													
Alg 33	315				22.14												
Alg 14	315				22.51												
Alg 23	315				22.73												
Alg 9	315				23.29												
Alg 11	315					25.07											
Alg 34	315					25.71											
Alg 24	315					25.96											
Alg 32	315						28.16										
Alg 22	315						29.54	29.54									
Alg 16	315							29.54	29.54								
Alg 30	315							30.30	30.30								
Alg 20	315							30.99	30.99								
Alg 15	315								33.94	33.94							
Alg 36	315									37.50							
Alg 31	315									37.90							
Alg 37	315									39.02	39.02						
Alg 17	315									39.15	39.15						
Alg 35	315										40.04	40.04					
Alg 27	315										41.48	41.48					
Alg 25	315										43.16	43.16					
Alg 26	315												45.81				
Alg 21	315													47.64			
Alg 29	315														60.62		
Alg 19	315															62.37	

Se muestran las medias para los grupos en los subconjuntos homogéneos.
a. Usa el tamaño muestral de la media armónica = 315.000.

c) 5 Modelos y 20 Colores

DemTot=Dem 100Bloques=Bloques 20Modelos=5 ModColores=20 Col

Tukey B^a

Algoritmos	N	Subconjunto para alfa = .05												
		1	2	3	4	5	6	7	8	9	10	11	12	13
Alg 10	315	21.00												
Alg 12	315	22.58												
Alg 13	315		25.33											
Alg 9	315			29.01										
Alg 14	315			29.04										
Alg 33	315			29.63										
Alg 11	315			30.21										
Alg 23	315			30.38										
Alg 34	315				32.51									
Alg 24	315				33.15	33.15								
Alg 32	315				34.54	34.54	34.54							
Alg 16	315					35.03	35.03	35.03						
Alg 22	315						35.60	35.60						
Alg 30	315								38.70					
Alg 15	315								39.75					
Alg 20	315									42.09				
Alg 36	315									42.34	42.34			
Alg 31	315									42.77	42.77			
Alg 37	315									43.77	43.77	43.77		
Alg 35	315									44.46	44.46	44.46	44.46	
Alg 17	315									44.57	44.57	44.57	44.57	
Alg 27	315										45.23	45.23	45.23	
Alg 25	315										46.29	46.29	46.29	
Alg 26	315											49.39	49.39	
Alg 21	315											50.08	50.08	
Alg 29	315													62.27
Alg 19	315													63.49

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 315.000.

d) 10 Modelos y 5 Colores

DemTot=Dem 100Bloques=Bloques 20Modelos=10 ModColores=5 Col

Tukey B^a

Algoritmos	N	Subconjunto para alfa = .05														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Alg 10	315	7.14														
Alg 12	315		11.41													
Alg 33	315			13.36												
Alg 13	315			13.37												
Alg 23	315			14.07												
Alg 11	315				17.50											
Alg 14	315				18.64	18.64										
Alg 34	315				19.62	19.62	19.62									
Alg 24	315				19.71	19.71	19.71									
Alg 9	315				19.78	19.78	19.78									
Alg 30	315					21.22	21.22	21.22								
Alg 16	315						22.51	22.51	22.51							
Alg 32	315						22.86	22.86	22.86							
Alg 22	315							24.13	24.13							
Alg 20	315							24.32	24.32							
Alg 31	315								29.56							
Alg 36	315								30.25							
Alg 15	315								30.88							
Alg 35	315									33.53						
Alg 37	315									34.08	34.08					
Alg 17	315										35.30					
Alg 27	315											35.30				
Alg 25	315											36.37	36.37			
Alg 26	315											36.84	36.84	36.84		
Alg 21	315												37.87	37.87	37.87	
Alg 29	315													38.88	38.88	
Alg 19	315														50.97	50.97
															52.54	52.54

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 315.000.

e) 10 Modelos y 10 Colores

DemTot=Dem 100Bloques=Bloques 20Modelos=10 ModColores=10 Col

Tukey B^a

Algoritmos	N	Subconjunto para alfa = .05															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Alg 10	315	12.25															
Alg 12	315		16.46														
Alg 13	315			18.54													
Alg 33	315				21.06												
Alg 23	315				21.72	21.72											
Alg 11	315					23.23	23.23										
Alg 14	315						23.88	23.88									
Alg 9	315							24.08									
Alg 34	315								26.09								
Alg 24	315								26.73								
Alg 30	315									26.73							
Alg 32	315									27.90	27.90						
Alg 16	315									28.11	28.11						
Alg 22	315									28.37	28.37						
Alg 20	315										29.49						
Alg 31	315											31.83					
Alg 36	315												35.97				
Alg 15	315												36.50				
Alg 35	315												37.48				
Alg 37	315													37.48			
Alg 27	315														38.85		
Alg 17	315															38.85	
Alg 25	315																41.44
Alg 26	315																41.46
Alg 21	315																41.67
Alg 29	315																43.31
Alg 19	315																43.41
																	56.38
																	57.79

Se muestran las medias para los grupos en los subconjuntos homogéneos.
 a. Usa el tamaño muestral de la media armónica = 315.000.

f) 10 Modelos y 20 Colores

DemTot=Dem 100Bloques=Bloques 20Modelos=10 ModColores=20 Col

Tukey B^a

Algoritmos	N	Subconjunto para alfa = .05											
		1	2	3	4	5	6	7	8	9	10	11	12
Alg 10	315	21.26											
Alg 12	315	23.53											
Alg 13	315		23.53										
Alg 33	315			25.04									
Alg 11	315				28.54								
Alg 9	315				29.28								
Alg 23	315				29.66								
Alg 14	315				29.74								
Alg 16	315				30.54								
Alg 34	315					33.08							
Alg 24	315					33.55	33.55						
Alg 32	315					34.05	34.05						
Alg 22	315							35.65					
Alg 30	315							36.91					
Alg 20	315							37.58					
Alg 31	315								40.90				
Alg 36	315								42.21	42.21			
Alg 15	315								42.89	42.89			
Alg 37	315									44.32	44.32		
Alg 35	315									44.67	44.67		
Alg 27	315										44.84		
Alg 25	315											45.95	
Alg 21	315											46.28	
Alg 17	315											47.57	
Alg 26	315											47.79	
Alg 29	315											48.23	
Alg 19	315												59.23
													60.38

Se muestran las medias para los grupos en los subconjuntos homogéneos.
 a. Usa el tamaño muestral de la media armónica = 315.000.

g) 20 Modelos y 5 Colores

DemTot=Dem 100Bloques=Bloques 20Modelos=20 ModColores=5 Col

Tukey B^a

Algoritmos	N	Subconjunto para alfa = .05												
		1	2	3	4	5	6	7	8	9	10			
Alg 10	315	7.54												
Alg 12	315		12.04											
Alg 33	315		13.36											
Alg 13	315		13.58	13.36										
Alg 23	315			13.58	14.06									
Alg 11	315				14.06	15.55								
Alg 14	315						19.33							
Alg 34	315						19.93							
Alg 24	315						20.03							
Alg 9	315						20.06							
Alg 16	315						20.42							
Alg 30	315						20.87							
Alg 32	315							24.01						
Alg 20	315							24.85						
Alg 22	315							25.05						
Alg 31	315								27.90					
Alg 36	315								28.43					
Alg 15	315									32.16				
Alg 35	315										34.47			
Alg 37	315										34.90			
Alg 26	315										35.10			
Alg 21	315										35.37			
Alg 27	315										35.44			
Alg 25	315										35.52			
Alg 17	315										35.69			
Alg 29	315											47.22		
Alg 19	315												48.47	

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 315.000.

h) 20 Modelos y 10 Colores

DemTot=Dem 100Bloques=Bloques 20Modelos=20 ModColores=10 Col

Tukey B^a

Algoritmos	N	Subconjunto para alfa = .05													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
Alg 10	315	12.96													
Alg 12	315		17.00												
Alg 13	315			19.18											
Alg 11	315			20.77	20.77										
Alg 33	315				21.63										
Alg 23	315				22.44										
Alg 9	315					24.79									
Alg 14	315					24.92									
Alg 16	315					25.92	25.92								
Alg 34	315					26.81	26.81								
Alg 24	315						27.31	27.31							
Alg 30	315							28.83	28.83						
Alg 32	315							29.39	29.39						
Alg 22	315								29.39	30.87					
Alg 20	315									30.87	32.47				
Alg 31	315										34.89				
Alg 36	315										35.85				
Alg 15	315											38.13			
Alg 35	315											39.76		39.76	
Alg 37	315													40.18	
Alg 21	315													40.72	
Alg 27	315													40.83	
Alg 25	315													40.97	
Alg 17	315													41.65	
Alg 26	315													41.74	
Alg 29	315														52.84
Alg 19	315														53.97

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 315.000.

i) 20 Modelos y 20 Colores

DemTot=Dem 100Bloques=Bloques 20Modelos=20 ModColores=20 Col

Tukey B²

Algoritmos	N	Subconjunto para alpha = .05												
		1	2	3	4	5	6	7	8	9	10	11	12	13
Alg 10	315	19.28												
Alg 11	315	20.70	20.70											
Alg 12	315	20.96	20.96											
Alg 9	315		21.28											
Alg 13	315		21.34											
Alg 16	315			23.99										
Alg 33	315				26.45									
Alg 14	315				26.88									
Alg 23	315				27.20									
Alg 34	315					29.58								
Alg 24	315					30.32								
Alg 32	315						32.43							
Alg 22	315						33.57	33.57						
Alg 31	315							34.43	34.43					
Alg 15	315							34.88	34.88					
Alg 30	315								35.58	35.58				
Alg 36	315								35.76	35.76				
Alg 35	315								36.01	36.01	36.01			
Alg 37	315									36.72	36.72	36.72		
Alg 25	315										37.51	37.51	37.51	
Alg 27	315										37.74	37.74	37.74	
Alg 17	315											37.96	37.96	37.96
Alg 20	315												38.15	38.15
Alg 21	315													38.49
Alg 26	315													39.60
Alg 29	315													45.68
Alg 19	315													46.77

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 315.000.

ANALISIS VARIABLE DEPENDIENTE NCMoVer:

Estadísticos descriptivos:

Estadísticos descriptivos

DemTot	Bloques	Modelos	Colores	N	Mínimo	Máximo	Media	Desv. típ.
Dem 100	Bloques 20	5 Mod	5 Col NCMoVer	8505	2	79	26.53	17.306
			N válido (según lista)	8505				
			10 Col NCMoVer	8505	2	80	32.29	19.699
			N válido (según lista)	8505				
			20 Col NCMoVer	8505	2	80	30.72	19.217
			N válido (según lista)	8505				
			10 Mod 5 Col NCMoVer	8505	5	80	38.07	20.698
			N válido (según lista)	8505				
			10 Col NCMoVer	8505	5	80	36.49	20.178
			N válido (según lista)	8505				
			20 Col NCMoVer	8505	5	80	35.87	19.934
			N válido (según lista)	8505				
			20 Mod 5 Col NCMoVer	8505	7	80	42.86	20.459
			N válido (según lista)	8505				
			10 Col NCMoVer	8505	7	80	40.68	20.041
			N válido (según lista)	8505				
20 Col NCMoVer	8505	2	80	34.81	19.137			
N válido (según lista)	8505							

Análisis de Varianza de factor "Listas":

ANOVA

NCModVer

DemTot	Bloques	Modelos	Colores		Suma de cuadrados	gl	Media cuadrática	F	Sig.	
Dem 100	Bloques 20	5 Mod	5 Col	Inter-grupos	5938.782	8	742.348	2.482	.011	
				Intra-grupos	2541076.324	8496	299.091			
				Total	2547015.106	8504				
			10 Col	Inter-grupos	8860.370	8	1107.546	2.859	.004	
				Intra-grupos	3291220.470	8496	387.385			
				Total	3300080.840	8504				
			20 Col	Inter-grupos	6052.304	8	756.538	2.051	.037	
				Intra-grupos	3134522.785	8496	368.941			
				Total	3140575.089	8504				
			10 Mod	5 Col	Inter-grupos	3541.496	8	442.687	1.033	.408
					Intra-grupos	3639742.722	8496	428.407		
					Total	3643284.218	8504			
			10 Col	Inter-grupos	4379.182	8	547.398	1.345	.216	
				Intra-grupos	3458088.040	8496	407.025			
				Total	3462467.222	8504				
			20 Col	Inter-grupos	2214.095	8	276.762	.696	.695	
				Intra-grupos	3377059.926	8496	397.488			
				Total	3379274.021	8504				
			20 Mod	5 Col	Inter-grupos	1590.938	8	198.867	.475	.875
					Intra-grupos	3557915.395	8496	418.775		
					Total	3559506.332	8504			
			10 Col	Inter-grupos	1981.043	8	247.630	.616	.765	
				Intra-grupos	3413416.508	8496	401.767			
				Total	3415397.551	8504				
20 Col	Inter-grupos	1883.999	8	235.500	.643	.742				
	Intra-grupos	3112654.049	8496	366.367						
	Total	3114538.048	8504							

Pruebas Post-Hoc, Tukey B :

A continuación se presenta el factor Listas con todos los casos para las distintas cantidades de modelos y colores:

a) 5 Modelos y 5 Colores

DemTot=Dem 100Bloques=Bloques
20Modelos=5 ModColores=5 Col

Tukey B^a

Listas	N	Subconjunto para alfa = .05	
		1	2
List 8	945	25.28	
List 6	945	25.67	25.67
List 5	945	26.27	26.27
List 1	945	26.33	26.33
List 4	945	26.36	26.36
List 2	945	26.49	26.49
List 3	945	26.56	26.56
List 7	945		27.75
List 9	945		28.06

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 945.000.

b) 5 Modelos y 10 Colores

DemTot=Dem 100Bloques=Bloques
20Modelos=5 ModColores=10 Col

Tukey B^a

Listas	N	Subconjunto para alfa = .05	
		1	2
List 8	945	30.83	
List 6	945	31.06	31.06
List 1	945	31.22	31.22
List 5	945	32.09	32.09
List 2	945	32.25	32.25
List 4	945	32.70	32.70
List 3	945	33.18	33.18
List 7	945	33.55	33.55
List 9	945		33.70

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 945.000.

c) 5 Modelos y 20 Colores

DemTot=Dem
100Bloques=Bloques
20Modelos=5 ModColores=20 Col

Tukey B^a

Listas	N	Subconjunto para alfa = .05
		1
List 1	945	29.53
List 6	945	29.65
List 8	945	29.91
List 2	945	30.58
List 5	945	30.68
List 4	945	31.01
List 3	945	31.50
List 9	945	31.77
List 7	945	31.89

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 945.000.

d) 10 Modelos y 5 Colores

DemTot=Dem
100Bloques=Bloques
20Modelos=10 ModColores=5 Col

Tukey B^a

Listas	N	Subconjunto para alfa = .05
		1
List 8	945	37.38
List 5	945	37.46
List 6	945	37.70
List 1	945	37.70
List 2	945	37.71
List 4	945	37.94
List 3	945	38.48
List 7	945	39.09
List 9	945	39.19

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 945.000.

e) 10 Modelos y 10 Colores

DemTot=Dem
100Bloques=Bloques
20Modelos=10 ModColores=10 Col

Tukey B^a

Listas	N	Subconjunto para alfa = .05
		1
List 8	945	35.67
List 6	945	35.73
List 5	945	35.94
List 1	945	35.98
List 2	945	36.07
List 4	945	36.93
List 3	945	37.12
List 7	945	37.30
List 9	945	37.67

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 945.000.

f) 10 Modelos y 20 Colores

DemTot=Dem
100Bloques=Bloques
20Modelos=10 ModColores=20 Col

Tukey B^a

Listas	N	Subconjunto para alfa = .05
		1
List 8	945	35.05
List 5	945	35.45
List 6	945	35.53
List 1	945	35.55
List 2	945	35.65
List 4	945	36.16
List 3	945	36.36
List 9	945	36.54
List 7	945	36.55

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 945.000.

g) 20 Modelos y 5 Colores

DemTot=Dem
100Bloques=Bloques
20Modelos=20 ModColores=5 Col

Tukey B^a

Listas	N	Subconjunto para alfa = .05
		1
List 5	945	42.33
List 2	945	42.37
List 8	945	42.57
List 4	945	42.67
List 3	945	42.80
List 6	945	42.81
List 1	945	43.23
List 9	945	43.29
List 7	945	43.71

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 945.000.

h) 20 Modelos y 10 Colores

**DemTot=Dem
100Bloques=Bloques
20Modelos=20 ModColores=10 Col**

Tukey B^a

Listas	N	Subconjunto para alfa = .05	
		1	
List 5	945	40.10	
List 8	945	40.15	
List 6	945	40.16	
List 2	945	40.48	
List 1	945	40.67	
List 4	945	40.84	
List 3	945	41.00	
List 9	945	41.10	
List 7	945	41.61	

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 945.000.

i) 20 Modelos y 20 Colores

**DemTot=Dem
100Bloques=Bloques
20Modelos=20 ModColores=20 Col**

Tukey B^a

Listas	N	Subconjunto para alfa = .05	
		1	
List 5	945	34.13	
List 2	945	34.20	
List 6	945	34.53	
List 8	945	34.64	
List 1	945	34.69	
List 9	945	34.94	
List 4	945	35.28	
List 3	945	35.36	
List 7	945	35.52	

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 945.000.

Análisis de Varianza de factor “Algoritmos”:

ANOVA

NCModVer

DemTot	Bloques	Modelos	Colores		Suma de cuadrados	gl	Media cuadrática	F	Sig.		
Dem 100	Bloques 20	5 Mod	5 Col	Inter-grupos	2016549.830	26	77559.609	1239.573	.000		
				Intra-grupos	530465.276	8478	62.570				
				Total	2547015.106	8504					
			10 Col	Inter-grupos	2794965.901	26	107498.688			1804.290	
				Intra-grupos	505114.940	8478	59.579				
				Total	3300080.840	8504					
			20 Col	Inter-grupos	2618821.439	26	100723.901			1636.667	
				Intra-grupos	521753.651	8478	61.542				
				Total	3140575.089	8504					
			10 Mod	5 Col	Inter-grupos	3100939.456	26			119266.902	1864.395
					Intra-grupos	542344.762	8478			63.971	
					Total	3643284.218	8504				
			10 Col	Inter-grupos	2898139.508	26	111466.904			1674.588	
				Intra-grupos	564327.714	8478	66.564				
				Total	3462467.222	8504					
			20 Col	Inter-grupos	2874672.262	26	110564.318			1857.632	
				Intra-grupos	504601.759	8478	59.519				
				Total	3379274.021	8504					
			20 Mod	5 Col	Inter-grupos	2968045.812	26			114155.608	1636.307
					Intra-grupos	591460.521	8478			69.764	
					Total	3559506.332	8504				
			10 Col	Inter-grupos	2826675.780	26	108718.299			1565.619	
				Intra-grupos	588721.771	8478	69.441				
				Total	3415397.551	8504					
20 Col	Inter-grupos	2308675.432	26	88795.209	934.161						
	Intra-grupos	805862.616	8478	95.053							
	Total	3114538.048	8504								

Pruebas Post-Hoc, Tukey B :

A continuación se presenta el factor Algoritmos con todos los casos para las distintas cantidades de modelos y colores:

a) 5 Modelos y 5 Colores

DemTot=Dem 100Bloques=Bloques 20Modelos=5 ModColores=5 Col

Algoritmos	N	Subconjunto para alfa = .05															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Alg 21	315	4.43															
Alg 26	315	6.16	6.16														
Alg 25	315		6.44	6.44													
Alg 27	315			8.08	8.08												
Alg 35	315				9.62	9.62											
Alg 37	315					11.10	11.10										
Alg 19	315						12.24	12.24									
Alg 31	315							13.39	13.39								
Alg 15	315								16.25	16.25							
Alg 36	315								16.36	16.36							
Alg 17	315								16.71	16.71							
Alg 20	315								17.42	17.42							
Alg 22	315									23.06							
Alg 30	315										29.46						
Alg 23	315										29.68	29.68					
Alg 32	315										29.99	29.99					
Alg 24	315											31.57					
Alg 34	315												35.56				
Alg 29	315												37.15	37.15			
Alg 33	315													38.56			
Alg 11	315													38.91			
Alg 16	315													38.95			
Alg 14	315														43.49		
Alg 12	315														43.76		
Alg 13	315															47.79	
Alg 10	315															48.54	
Alg 9	315																61.61

Se muestran las medias para los grupos en los subconjuntos homogéneos.
a. Usa el tamaño muestral de la media armónica = 315.000.

b) 5 Modelos y 10 Colores

DemTot=Dem 100Bloques=Bloques 20Modelos=5 ModColores=10 Col

Tukey B ^a		Subconjunto para alfa = .05														
Algoritmos	N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Alg 21	315	5.12														
Alg 26	315	6.82														
Alg 25	315		9.23													
Alg 27	315			11.15												
Alg 35	315				15.07											
Alg 31	315				15.64											
Alg 19	315				16.21											
Alg 37	315				16.64											
Alg 36	315					18.76										
Alg 20	315						21.38									
Alg 15	315							25.62								
Alg 22	315							26.28								
Alg 17	315							26.47								
Alg 23	315								31.88							
Alg 30	315								32.27							
Alg 32	315								32.86							
Alg 24	315									34.89						
Alg 34	315										40.46					
Alg 33	315										40.72					
Alg 29	315											42.86				
Alg 16	315												47.78			
Alg 11	315												48.17			
Alg 14	315													57.76		
Alg 13	315													58.55	58.55	
Alg 12	315													58.88	58.88	
Alg 10	315														59.87	
Alg 9	315															70.62

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 315.000.

c) 5 Modelos y 20 Colores

DemTot=Dem 100Bloques=Bloques 20Modelos=5 ModColores=20 Col

Tukey B ^a		Subconjunto para alfa = .05															
Algoritmos	N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Alg 21	315	4.99															
Alg 26	315		6.79														
Alg 25	315			9.22													
Alg 27	315				11.41												
Alg 35	315					15.57											
Alg 31	315					15.82											
Alg 19	315					16.20	16.20										
Alg 37	315					17.10	17.10	17.10									
Alg 20	315						17.87	17.87	17.87								
Alg 36	315							18.85									
Alg 22	315								22.59								
Alg 15	315									26.36							
Alg 23	315									26.63	26.63						
Alg 17	315									27.23	27.23						
Alg 32	315									28.06	28.06	28.06					
Alg 30	315									28.40	28.40	28.40					
Alg 24	315										29.57						
Alg 34	315											36.42					
Alg 33	315											37.30					
Alg 29	315												39.69				
Alg 16	315													48.52			
Alg 11	315													48.63			
Alg 13	315														56.63		
Alg 14	315														56.66		
Alg 12	315														57.36		
Alg 10	315														57.81		
Alg 9	315																67.89

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 315.000.

d) 10 Modelos y 5 Colores

DemTot=Dem 100Bloques=Bloques 20Modelos=10 ModColores=5 Col

Tukey B^a

Algoritmos	N	Subconjunto para alfa = .05																
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
Alg 21	315	8.91																
Alg 26	315		11.26															
Alg 25	315		12.22															
Alg 27	315			14.43														
Alg 35	315				17.55													
Alg 19	315				19.08													
Alg 37	315				19.38													
Alg 31	315					23.47												
Alg 20	315						26.35											
Alg 36	315						26.55											
Alg 15	315							29.20										
Alg 22	315								31.08									
Alg 17	315								31.23									
Alg 32	315									39.30								
Alg 30	315									40.80								
Alg 24	315										40.80							
Alg 23	315										41.23							
Alg 34	315										42.41							
Alg 33	315											47.49						
Alg 29	315												50.13					
Alg 16	315												51.41					
Alg 11	315													55.33				
Alg 14	315													55.46				
Alg 12	315														62.53			
Alg 13	315														63.39			
Alg 10	315															65.96		
Alg 9	315															66.86		
																		74.99

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 315.000.

e) 10 Modelos y 10 Colores

DemTot=Dem 100Bloques=Bloques 20Modelos=10 ModColores=10 Col

Tukey B^a

Algoritmos	N	Subconjunto para alfa = .05																
		1	2	3	4	5	6	7	8	9	10	11	12	13	14			
Alg 21	315	8.80																
Alg 26	315		10.86															
Alg 25	315		12.35															
Alg 27	315			14.41														
Alg 35	315				17.23													
Alg 37	315					19.30												
Alg 19	315					19.31												
Alg 31	315						21.69											
Alg 36	315							24.70										
Alg 20	315							26.07										
Alg 15	315								28.92									
Alg 22	315								29.23									
Alg 17	315								30.54									
Alg 23	315									36.48								
Alg 32	315									36.65								
Alg 24	315									37.74								
Alg 30	315									38.19								
Alg 34	315										44.69							
Alg 33	315										46.33							
Alg 29	315											46.33						
Alg 16	315												46.66					
Alg 11	315													53.89				
Alg 14	315													53.94				
Alg 13	315														62.99			
Alg 12	315														63.43			
Alg 10	315														63.86			
Alg 9	315														64.56			
																		72.39

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 315.000.

f) 10 Modelos y 20 Colores

DemTot=Dem 100Bloques=Bloques 20Modelos=10 ModColores=20 Col

Tukey B^a

Algoritmos	N	Subconjunto para alfa = .05															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Alg 21	315																
Alg 26	315	9.02															
Alg 25	315		10.81														
Alg 27	315		12.52														
Alg 35	315			14.62													
Alg 19	315				18.31												
Alg 37	315				19.36	19.36											
Alg 20	315					20.33											
Alg 31	315						22.12										
Alg 22	315						23.22										
Alg 36	315							25.04									
Alg 15	315							26.24									
Alg 23	315								30.38								
Alg 17	315								31.57								
Alg 32	315								32.01								
Alg 24	315									33.84							
Alg 30	315									33.86							
Alg 34	315										36.00						
Alg 33	315											42.18					
Alg 29	315												44.77				
Alg 11	315												44.87				
Alg 16	315													55.21			
Alg 13	315													55.36			
Alg 10	315														62.85		
Alg 14	315														64.38	64.38	
Alg 12	315															64.74	65.95
Alg 9	315																68.95

Se muestran las medias para los grupos en los subconjuntos homogéneos.
 a. Usa el tamaño muestral de la media armónica = 315.000.

g) 20 Modelos y 5 Colores

DemTot=Dem 100Bloques=Bloques 20Modelos=20 ModColores=5 Col

Tukey B^a

Algoritmos	N	Subconjunto para alfa = .05															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Alg 21	315																
Alg 26	315	14.57															
Alg 25	315		17.10														
Alg 27	315		17.28														
Alg 35	315			19.54													
Alg 19	315				22.04												
Alg 37	315				22.44	22.44											
Alg 20	315					24.17											
Alg 31	315						29.93										
Alg 22	315						31.23										
Alg 36	315							33.42									
Alg 15	315							34.55	34.55								
Alg 17	315							35.88	35.88								
Alg 32	315									38.23							
Alg 30	315										42.32						
Alg 24	315											44.93					
Alg 23	315											45.34					
Alg 34	315											46.50					
Alg 29	315												51.88				
Alg 33	315												52.10				
Alg 16	315													54.88			
Alg 11	315														62.44		
Alg 14	315														62.92		
Alg 12	315															68.79	
Alg 13	315															69.54	
Alg 10	315															69.61	
Alg 9	315															70.79	
																	74.93

Se muestran las medias para los grupos en los subconjuntos homogéneos.
 a. Usa el tamaño muestral de la media armónica = 315.000.

h) 20 Modelos y 10 Colores

DemTot=Dem 100Bloques=20Modelos=20 ModColores=10 Col

Tukey B^a

Algoritmos	N	Subconjunto para alfa = .05																
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Alg 21	315	13.61																
Alg 26	315		15.99															
Alg 25	315		16.31															
Alg 27	315			18.58														
Alg 35	315				21.56													
Alg 19	315				21.81													
Alg 37	315				23.43													
Alg 20	315					27.84												
Alg 31	315					29.03												
Alg 22	315						29.03											
Alg 36	315							30.28										
Alg 15	315								30.28									
Alg 17	315									31.79								
Alg 32	315										34.86							
Alg 23	315											36.86						
Alg 24	315												39.70					
Alg 30	315												40.27					
Alg 29	315												40.70					
Alg 34	315												41.99					
Alg 33	315													48.78				
Alg 16	315													48.85				
Alg 11	315														51.16			
Alg 13	315															60.40		
Alg 10	315															60.47		
Alg 14	315																66.82	
Alg 12	315																68.20	68.20
Alg 9	315																68.21	68.21
																	69.72	69.72
																	69.72	71.11

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 315.000.

i) 20 Modelos y 20 Colores

DemTot=Dem 100Bloques=20Modelos=20 ModColores=20 Col

Tukey B^a

Algoritmos	N	Subconjunto para alfa = .05																
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		
Alg 21	315	11.74																
Alg 26	315	13.76	13.76															
Alg 19	315		15.01															
Alg 25	315		15.31	15.31														
Alg 20	315			17.43														
Alg 27	315			17.50														
Alg 22	315				20.90													
Alg 35	315				21.20													
Alg 37	315				22.67													
Alg 23	315					27.97												
Alg 31	315					28.19	28.19											
Alg 24	315					29.39	29.39	29.39										
Alg 36	315						30.45	30.45										
Alg 32	315							30.93										
Alg 30	315								33.25									
Alg 15	315									35.54								
Alg 17	315									36.17								
Alg 29	315										37.40							
Alg 34	315										37.40							
Alg 33	315											38.73						
Alg 16	315												40.99					
Alg 11	315													56.23				
Alg 13	315													57.12				
Alg 10	315														57.12			
Alg 14	315															58.59		
Alg 9	315																58.59	59.90
Alg 12	315																60.65	60.65
																	60.85	60.85
																	60.85	61.97

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 315.000.

1.2 Análisis Bi-Objetivo

Demanda = 100, Bloques = 20

Se presentan a continuación los algoritmos que son Frontera de Pareto cuando la demanda es 100 y los bloques 20. Enseguida se muestran por separado las gráficas correspondientes a los distintos niveles y combinaciones de las variables Modelo y Color.

Estadísticos

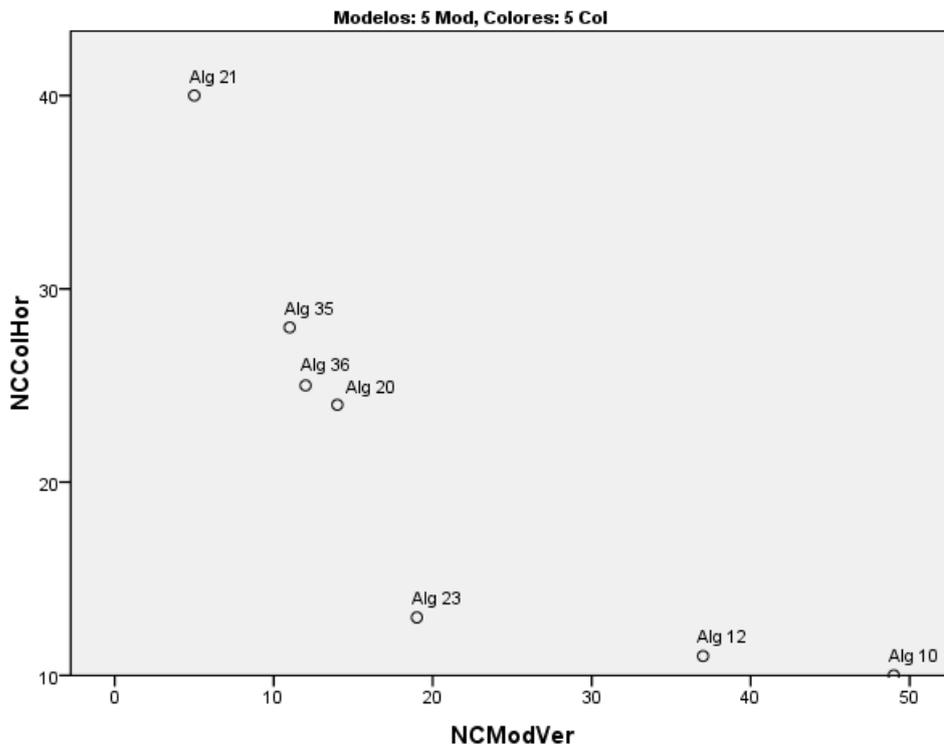
Algoritmos

N	Válidos	57
	Perdidos	0

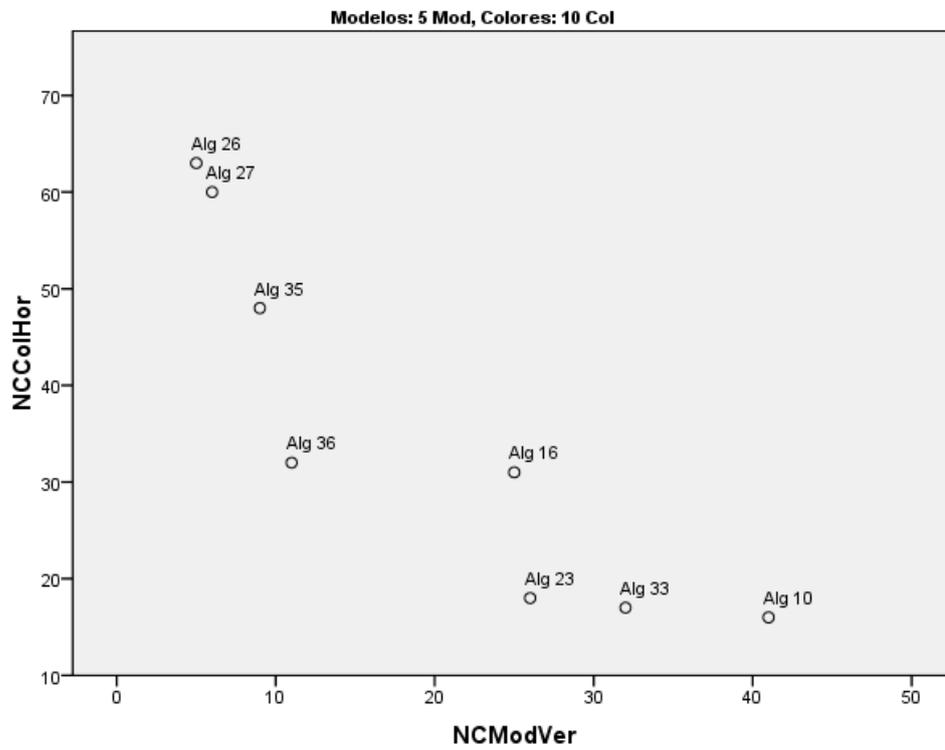
Algoritmos

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos Alg 9	1	1.8	1.8	1.8
Alg 10	2	3.5	3.5	5.3
Alg 12	3	5.3	5.3	10.5
Alg 13	1	1.8	1.8	12.3
Alg 15	1	1.8	1.8	14.0
Alg 16	1	1.8	1.8	15.8
Alg 19	3	5.3	5.3	21.1
Alg 20	3	5.3	5.3	26.3
Alg 21	5	8.8	8.8	35.1
Alg 22	1	1.8	1.8	36.8
Alg 23	9	15.8	15.8	52.6
Alg 25	4	7.0	7.0	59.6
Alg 26	3	5.3	5.3	64.9
Alg 27	1	1.8	1.8	66.7
Alg 30	3	5.3	5.3	71.9
Alg 31	6	10.5	10.5	82.5
Alg 33	3	5.3	5.3	87.7
Alg 35	2	3.5	3.5	91.2
Alg 36	5	8.8	8.8	100.0
Total	57	100.0	100.0	

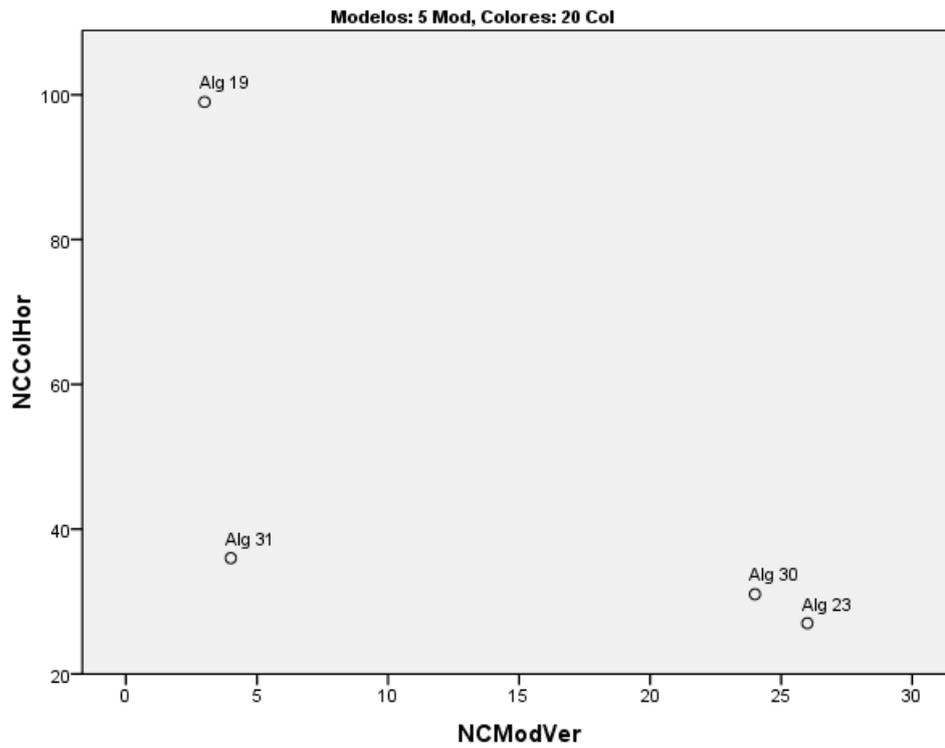
a) 5 Modelos y 5 Colores



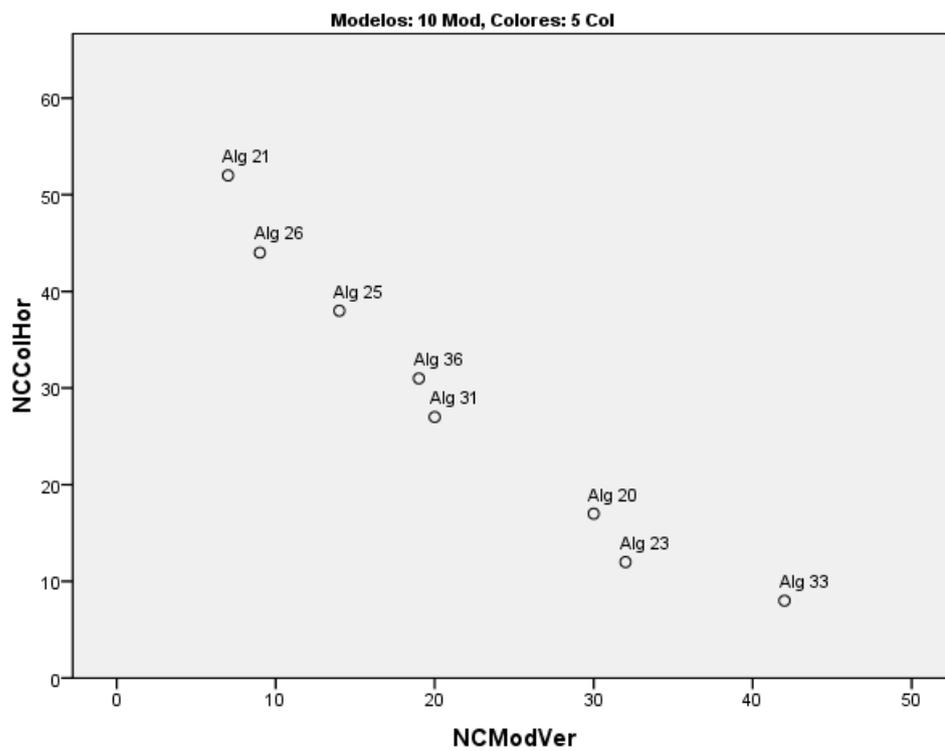
b) 5 Modelos y 10 Colores



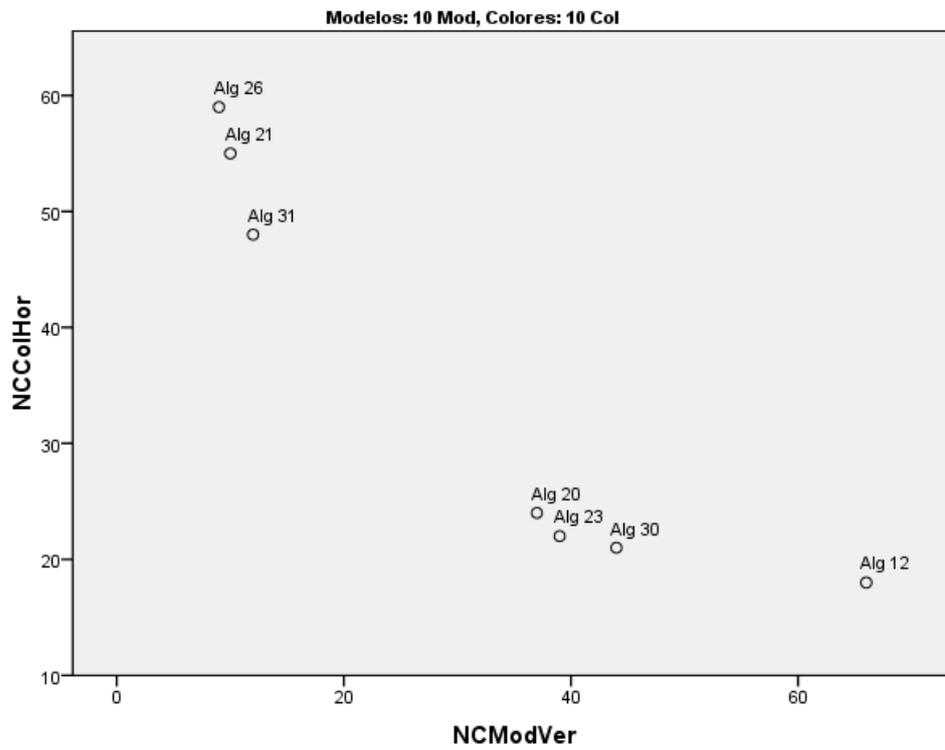
c) 5 Modelos y 20 Colores



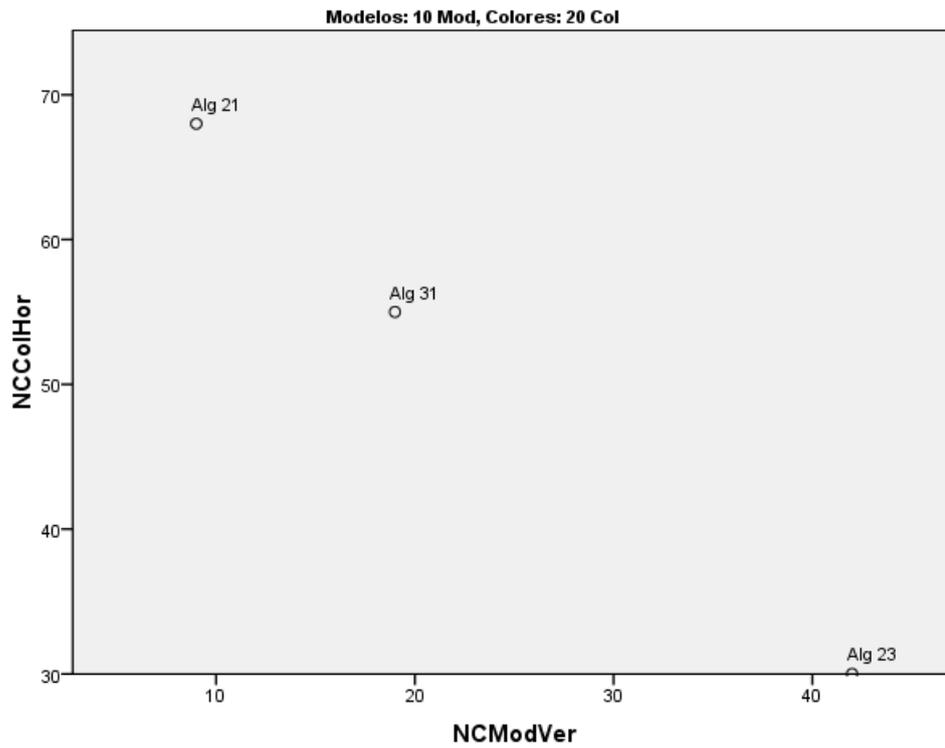
d) 10 Modelos y 5 Colores



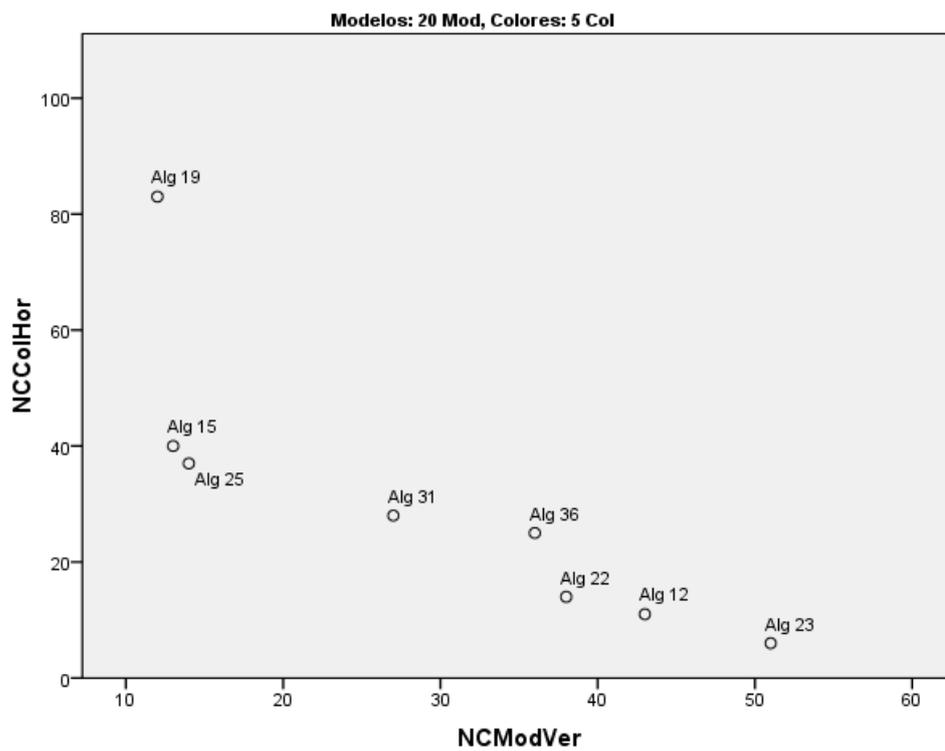
e) 10 Modelos y 10 Colores



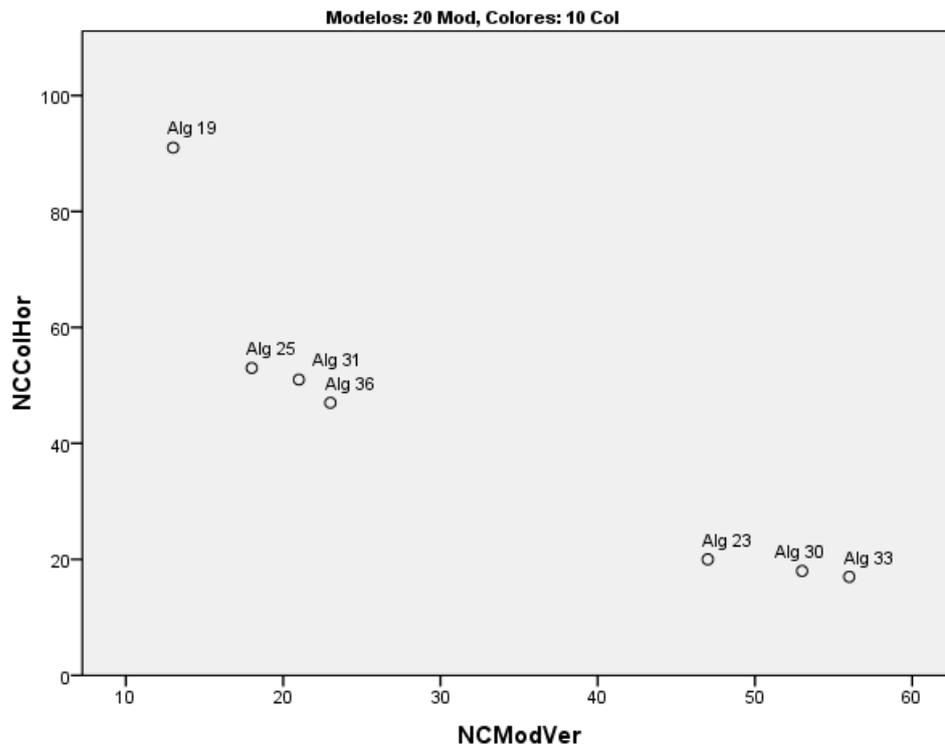
f) 10 Modelos y 20 Colores



g) 20 Modelos y 5 Colores



h) 20 Modelos y 10 Colores



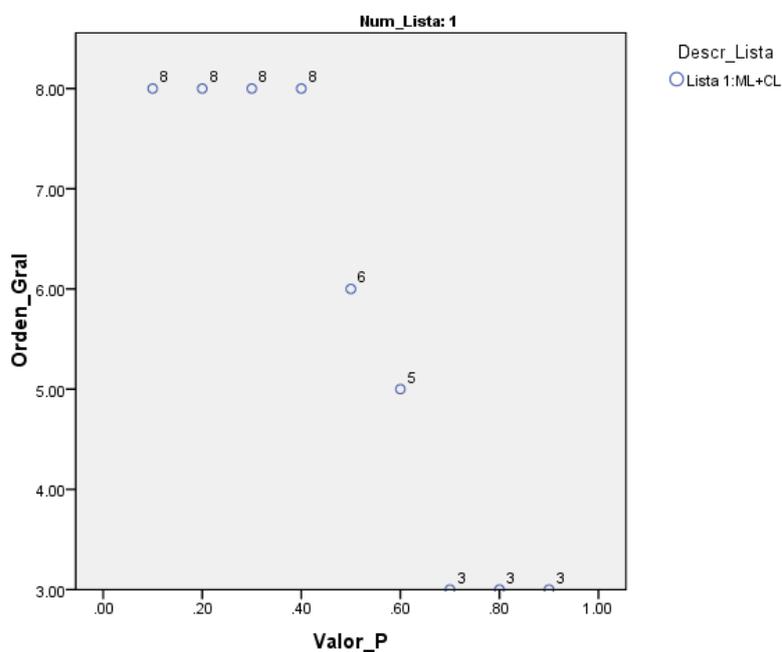
i) 20 Modelos y 20 Colores

ANEXO 2

COMPORTAMIENTO RELATIVO DE LISTAS CON FACTOR P. EN EXPERIMENTOS CBBSSP-SH

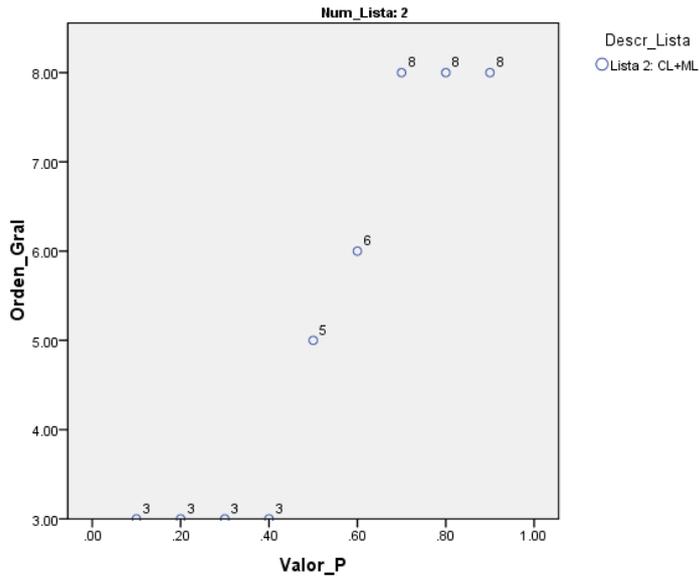
2.1 Comportamiento Individual de cada Lista:

2.1.1 Lista 1

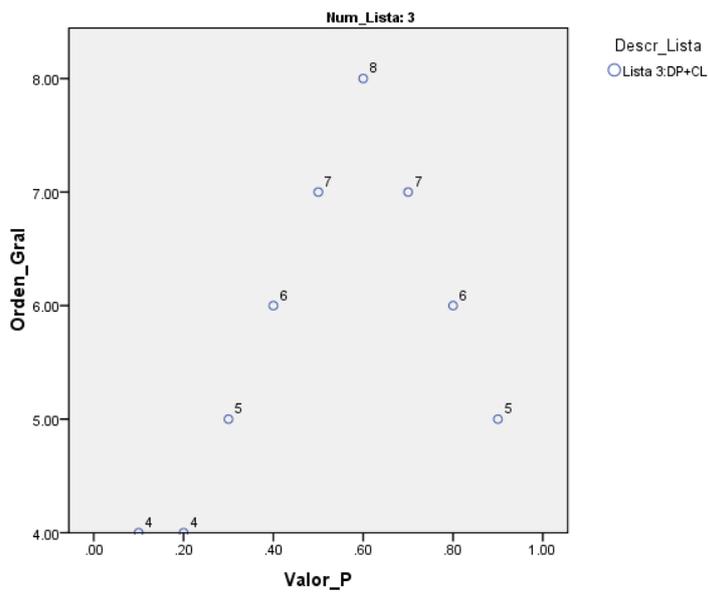


2.1.2 Lista 2

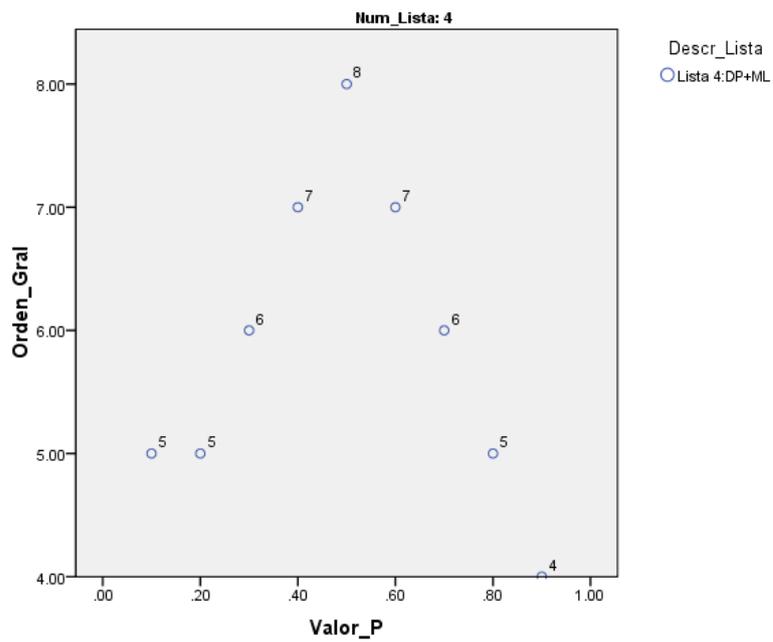
La lista 2 ordena los productos por color lexicográficamente y enseguida por modelo.



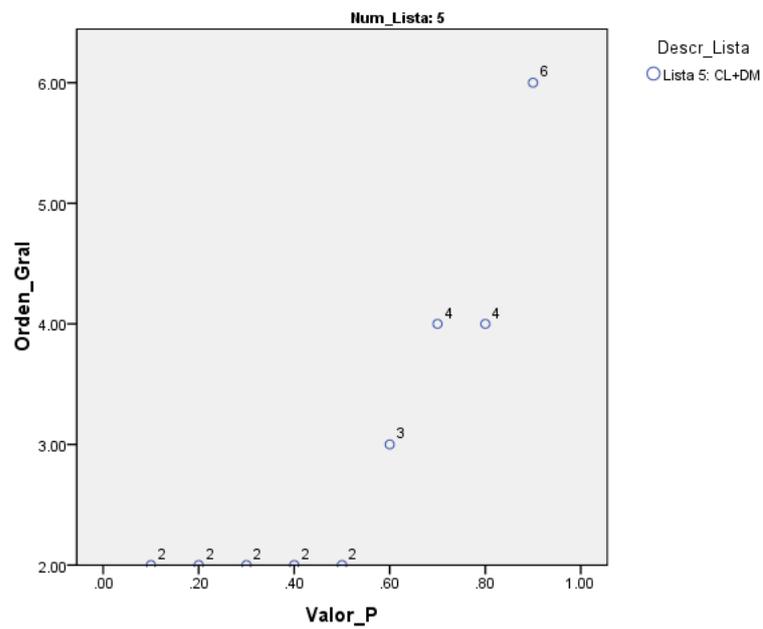
2.1.3 Lista 3



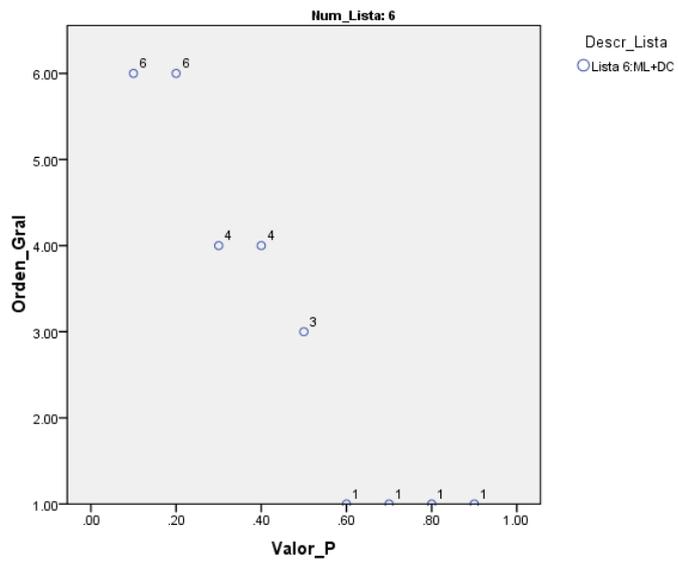
2.1.4 Lista 4



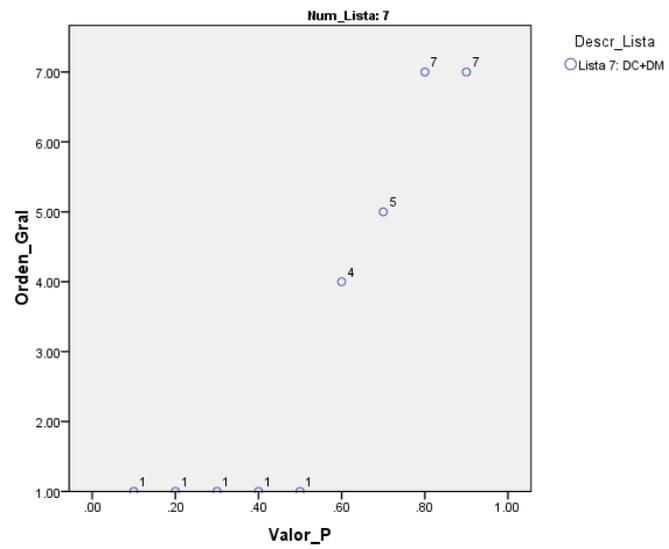
2.1.5 Lista 5



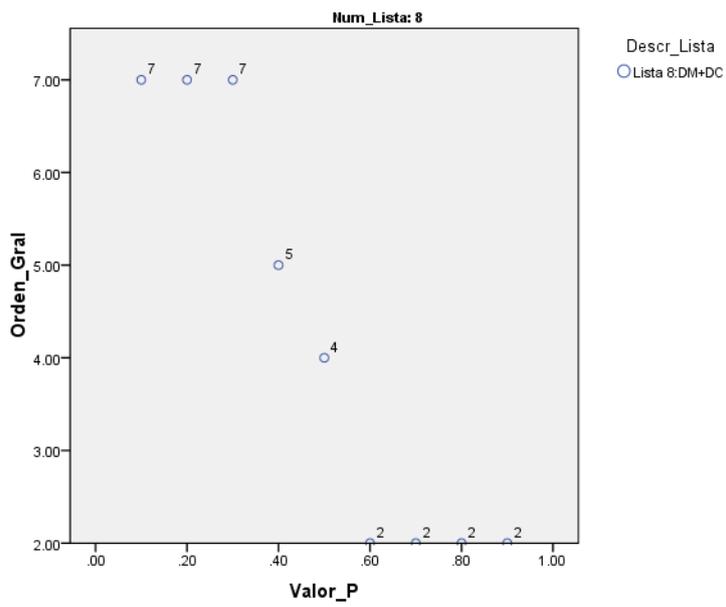
2.1.6 Lista 6



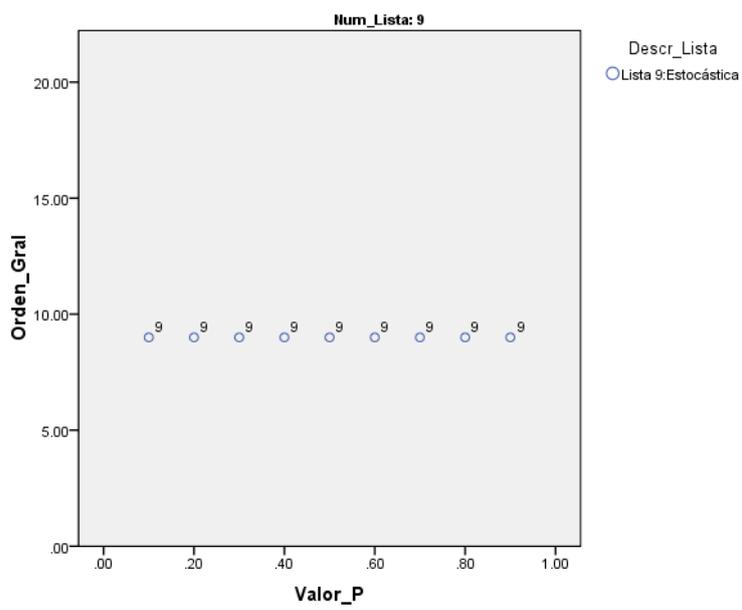
2.1.7 Lista 7



2.1.8 Lista 8



2.1.9 Lista 9



ANEXO 3

ANALISIS ESTADISTICO VARIABLES INDEPENDIENTES

3.1 Segmentación por Variable Independiente Bloques:

En la Tabla 39 se observa que a diferentes niveles de bloques existe significancia en los resultados de los datos. Es decir, el número de espacios vacíos o huecos en cada nivel asegura variación en los resultados de los Algoritmos al interior de cada nivel de bloques.

ANOVA

HUECOS

BLOQ		Suma de cuadrados	gl	Media cuadrática	F	Sig.
20	Inter-grupos	5.117E8	18	2.843E7	27.374	.000
	Intra-grupos	4.775E9	4598	1038389.730		
	Total	5.286E9	4616			
50	Inter-grupos	1.517E8	18	8428302.432	11.755	.000
	Intra-grupos	3.297E9	4598	716999.133		
	Total	3.448E9	4616			
100	Inter-grupos	2.203E8	18	1.224E7	16.275	.000
	Intra-grupos	3.457E9	4598	751857.070		
	Total	3.677E9	4616			

Tabla 39: Análisis de Varianza de la variable Algoritmos a diferentes niveles de Bloques

A nivel de bloques 20, los algoritmos 10, 11 y 12 son los que funcionan mejor y no tienen traslape con el segundo nivel en la prueba Tukey-B. Los algoritmos 14 y 16 funcionan bien, pero aparecen también en el segundo nivel. No se puede asegurar que sean parte de los mejores para este nivel de bloques. Esto se observa en la Tabla 40.

BLOQ=20

Tukey B^a

ALG	N	Subconjunto para alfa = .05				
		1	2	3	4	5
10	243	270.45				
11	243	283.99				
12	243	286.24				
14	243	418.36	418.36			
16	243	439.51	439.51			
32	243		644.45	644.45		
30	243		667.25	667.25		
31	243		676.52	676.52		
36	243		701.32	701.32		
34	243			738.00	738.00	
33	243			741.63	741.63	
35	243			748.79	748.79	
21	243				1009.25	1009.25
20	243					1040.73
22	243					1052.84
25	243					1211.64
23	243					1244.25
26	243					1258.33
24	243					1262.08

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 243.000.

Tabla 40: Tukey-B. Efecto de variable Algoritmos sobre Datos con Bloques 20

Cuando se hace la prueba Tukey-B con Bloques 50, el primer nivel contiene a los algoritmos horizontales. Es decir: algoritmos 10, 11, 12, 14 y 16. No existe ninguno que se traslape con otro nivel. Ver Tabla 41.

BLOQ=50

Tukey B^a

ALG	N	Subconjunto para alfa = .05	
		1	2
10	243	349.78	
11	243	372.06	
12	243	383.37	
14	243	405.56	
16	243	451.64	
21	243		711.70
20	243		712.98
22	243		716.66
26	243		763.69
24	243		765.08
23	243		769.44
25	243		779.57
36	243		807.60
32	243		811.27
31	243		822.77
30	243		835.18
33	243		841.52
34	243		849.35
35	243		872.89

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 243.000.

Tabla 41: Tukey-B. Efecto de variable Algoritmos sobre Datos con Bloques 50

Al continuar ahora con la Tabla 42, se puede observar en el nivel de Bloques de 100 que existen 12 algoritmos de los 19 que tienen buen desempeño. Se incluyen todos los horizontales y verticales. Los diagonales no tienen buen desempeño.

BLOQ=100

TukeyB ^a		Subconjunto para alfa = .05	
ALG	N	1	2
10	243	450.77	
14	243	475.96	
11	243	494.78	
12	243	495.71	
20	243	503.85	
24	243	515.18	
23	243	515.51	
22	243	515.76	
16	243	517.47	
26	243	528.30	
21	243	565.02	
25	243	572.31	
36	243		903.76
32	243		911.78
31	243		949.27
33	243		969.99
34	243		981.90
30	243		986.44
35	243		1014.93

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 243.000.

Tabla 42: Tukey-B. Efecto de variable Algoritmos sobre Datos con Bloques 100

Conclusiones de los Algoritmos:

Con las tablas anteriores se puede concluir que el comportamiento de los algoritmos es necesario analizarlo a diferentes niveles de cada una de las variables independientes. Si se hace solamente a nivel general, estadísticamente se podría concluir que los algoritmos: 10, 11, 12, 14 y 16, es decir, los horizontales, son mejores que el resto. Pero un análisis mediante la Prueba Tukey-B, haciendo la segmentación de la variable Bloques nos lleva a medida que se incrementa el número de Bloques, existen mayor cantidad de algoritmos con buen desempeño. Con 20 bloques funciona bien 3 de 5 algoritmos horizontales, con bloques 50 funcionan bien todos los horizontales, con bloques 100 funcionan bien los algoritmos horizontales y los verticales.

En la Tabla 35 se observa que a diferentes niveles de demanda existe no existe significancia en los resultados de los datos en todos los casos. Si se observa a detalle. En el caso de la Demanda 100 el valor de significancia es superior 0.05, lo que indica que no hay diferencias entre los resultados que proporcionan las distintas listas cuando se tiene una demanda de 100 productos.

ANOVA

HUECOS

BLOQ		Suma de cuadrados	gl	Media cuadrática	F	Sig.
20	Inter-grupos	4.490E7	8	5612225.009	4.934	.000
	Intra-grupos	5.241E9	4608	1137429.250		
	Total	5.286E9	4616			
50	Inter-grupos	3.508E7	8	4385020.739	5.920	.000
	Intra-grupos	3.413E9	4608	740753.318		
	Total	3.448E9	4616			
100	Inter-grupos	2.987E7	8	3733328.339	4.717	.000
	Intra-grupos	3.647E9	4608	791542.014		
	Total	3.677E9	4616			

Tabla 43: Análisis de Varianza de la variable Listas a diferentes niveles de Bloques

ANOVA

HUECOS

TDEM		Suma de cuadrados	gl	Media cuadrática	F	Sig.
100	Inter-grupos	8375.240	8	1046.905	1.538	.138
	Intra-grupos	3136596.269	4608	680.685		
	Total	3144971.509	4616			
1000	Inter-grupos	3565851.298	8	445731.412	31.588	.000
	Intra-grupos	6.502E7	4608	14110.647		
	Total	6.859E7	4616			
10000	Inter-grupos	2.461E8	8	3.077E7	29.031	.000
	Intra-grupos	4.884E9	4608	1059835.879		
	Total	5.130E9	4616			

Tabla 44: Análisis de Varianza de la variable Listas a diferentes niveles de Demanda

En el caso de la variable Listas se puede observar que cuando la demanda es 100 no hay diferencias significativas entre los valores que las distintas listas generan. Se observa en la Tabla 36 que solo existe un nivel y todas las Listas están en él. No se puede concluir en este nivel si una lista es mejor que otra.

BLOQ=20

Tukey B^a

LIST	N	Subconjunto para alfa = .05	
		1	2
6	513	690.98	
9	513	698.22	
8	513	704.47	
1	513	707.85	
4	513	711.70	
3	513	714.58	
2	513	875.68	875.68
7	513		928.01
5	513		929.59

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 513.000.

Tabla 45: Tukey-B. Efecto de variable Listas sobre Datos con Bloques 20

TDEM=100

Tukey B^a

LIST	N	Subconjunto para alfa = .05
		1
4	513	65.65
9	513	66.28
3	513	66.91
6	513	67.81
8	513	68.00
1	513	68.08
5	513	69.21
7	513	69.56
2	513	69.67

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 513.000.

Tabla 46: Tukey-B. Efecto de variable Listas sobre Datos con Demanda 100

En el nivel de demanda de 1000 se puede observar en la Tabla 37 que las listas que generan mejor resultado son 4, 3, y 9. Estas también están incluidas en el nivel de demanda 10000 como se observa en la Tabla 38.

BLOQ=50

Tukey B^a

LIST	N	Subconjunto para alfa = .05	
		1	2
4	513	591.96	
3	513	592.93	
9	513	614.22	
1	513	655.99	
6	513	656.46	
8	513	659.41	
2	513	742.51	742.51
5	513		822.85
7	513		832.04

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 513.000.

Tabla 47: Tukey-B. Efecto de variable Listas sobre Datos con Bloques 50

TDEM=1000

Tukey B^a

LIST	N	Subconjunto para alfa = .05	
		1	2
4	513	302.82	
3	513	303.43	
9	513	322.23	
8	513		356.13
1	513		356.14
6	513		358.36
2	513		368.67
5	513		375.73
7	513		376.94

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 513.000.

Tabla 48: Tukey-B. Efecto de variable Listas sobre Datos con Demanda 1000

BLOQ=100

Tukey B^a

LIST	N	Subconjunto para alfa = .05	
		1	2
4	513	563.34	
3	513	563.43	
9	513	607.29	
1	513	674.69	674.69
8	513	688.64	688.64
6	513	705.16	705.16
2	513	715.86	715.86
5	513		779.30
7	513		797.98

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 513.000.

Tabla 49: Tukey-B. Efecto de variable Listas sobre Datos con Bloques 100

TDEM=10000

Tukey B^a

LIST	N	Subconjunto para alfa = .05		
		1	2	3
4	513	1498.54		
3	513	1500.61		
9	513	1531.22		
1	513	1614.31		
6	513	1626.43		
8	513	1628.38		
2	513		1895.71	
5	513			2086.80
7	513			2111.52

Se muestran las medias para los grupos en los subconjuntos homogéneos.

a. Usa el tamaño muestral de la media armónica = 513.000.

Tabla 50: Tukey-B. Efecto de variable Listas sobre Datos con Demanda 10000

Conclusiones de las Listas:

Con las tablas anteriores se puede concluir que el comportamiento de los listas es posible hacerlo a diferentes niveles de cada una de las variables independientes. Si se hace solamente a nivel general, estadísticamente se podría concluir que el 66% de las listas tienen buen desempeño: 4, 3, 9, 1, 8, y 6. Pero un análisis mediante la Prueba Tukey-B, haciendo la segmentación de la variable Demanda nos lleva a concluir que son 3 listas, es decir, el 33% del total, las que en forma consistente y repetida, en todos los niveles de demanda siempre aparece en el primer nivel, las listas 4, 3 y 9. Las listas 4 y 3 fueron creadas mediante un procedimiento pero la 9 es aleatoria.

Para corroborar el desempeño de los Listas con las demás variables, se puede observar el comportamiento de las mismas con las demás variables independientes: Número de Bloques, Número de modelos, Número de colores.