

# MODELOS Y ALGORITMOS DE SECUENCIACIÓN EN LÍNEAS DE ENSAMBLE DE PRODUCTOS MIXTOS

Jaime Cano Belmán





UNIVERSITAT POLITÈCNICA DE CATALUNYA

Título de la tesis

MODELOS Y ALGORITMOS DE  
SECUENCIACIÓN EN LÍNEAS DE  
ENSAMBLE DE PRODUCTOS MIXTOS

Elaborada por

JAIME CANO BELMÁN

En el programa de doctorado

APLICACIONES TÈCNiques I INFORMÀTIQUES DE  
L'ESTADÍSTICA, LA INVESTIGACIÓ OPERATIVA I  
L'OPTIMITZACIÓ

Tesis dirigida por

Dr. JOAQUÍN BAUTISTA VALHONDO

Barcelona, Octubre 2006.



*A mis padres*

*Nantli, tahtli*

*Nonantzin ihcuac nimiquiz,  
motlecuilpan xinechtoca  
huan cuac tiaz titlaxcal chihuaz,  
ompa nopampa xichoca.*

*Tlazoh camati*



---

## Agradecimientos

---

Agradezco enormemente a mi familia, a mis padres por permitirme ser y hacer, a mis dos hermanos favoritos, únicos, a mi hermanita consentida. Gracias a ellos por conformar gran parte de mi mundo.

Agradezco al Dr. Joaquín Bautista Valhondo, junto a quien me he adentrado en el curioso mundo de la investigación. Gracias por todos los momentos y enseñanzas.

Agradezco al Departamento de Estadística e Investigación Operativa de la Universitat Politècnica de Catalunya, a quienes lo conforman. Agradezco especialmente a la Dra. Elena Fernández Aréizaga, una excelente persona.

También agradezco al Consejo Nacional de Ciencia y Tecnología (CONACYT, México) por su invaluable apoyo (Beca nº 150937) y por su confianza.

Agradezco el apoyo de la Càtedra Nissan Motor Ibérica d'Innovació a l'Automoció y al agradable equipo de la cátedra con quienes aprendí y conviví.

Ich möchte auch Herrn Dr. Armin Scholl von der Friedrich-Schiller-Universität Jena für seine Bereitschaft danken, uns seine Instanzen zur Verfügung zu stellen, um unsere Algorithmen zu testen (Agradezco al Dr. Armin Scholl de la Friedrich-Schiller-Universität en Jena por facilitarnos instancias para probar nuestros algoritmos).

Gracias a todos los amigos y compañeros que he conocido en esta Barcelona y en muchos otros lugares. A las familias que he sentido propias. Gracias por todos los instantes, lugares inimaginables, juegos, abundantes sensaciones y sentimientos, duros aprendizajes, frías experiencias, arenas calientes, breves espacios, íntimas poesías y pletóricos placeres que hemos devorado juntos regocijadamente una y más veces. Gracias por su amistad. Nunca esperé tantas coincidencias: muy señora mía Lui, Marta y Ricard, la reina Dulce, a Lesly, al equipo 2+2 (Carlitos y Klaus), a Cèlia Martín, a Elisa, a Todd, a Jair (no a Jair, sino el chamaco de Jair) y a todos los que han estado conmigo.

Jordi (Sns)... no tinc paraules, només la meva amistat, admiració i agraïment.

Agraeixo, enormement esta clar, a na Mari, per la seva eterna disponibilitat, pero sobre tot per la seva amistat, merci Mari.. tens raó.

Danke mein kleines Fräulein ... *ich liebe deinen Augen, ich liebe, ich liebe deinen Augen, ich bin wie der Sohn von deinen Augen, wie ein Tropfen von deinen Augen ich bin...*



Se estudia el problema de secuenciación de productos mixtos en líneas de montaje. Las líneas de montaje de productos mixtos (MMAL) permiten manufacturar productos con muchas y pequeñas variantes, evitando inventarios y tiempos de preparación importantes. Por ello, las líneas de montaje se han convertido en un componente importante en los sistemas de manufactura repetitiva moderna, en particular en el sistema de producción *Just in Time*.

La secuenciación de los productos es una decisión importante para el uso eficiente de las líneas de montaje. Se considera el nivelado de la carga como criterio de secuenciación [Monden, 1983]. Varios objetivos se han considerado en la literatura para medir el nivelado de la carga. El objetivo a estudiar depende de las políticas administrativas de la compañía y de las restricciones impuestas por el sistema productivo [Kotani y Ohno, 2004; Okamura y Yamashina, 1979; Sarker y Pan, 1998].

En esta tesis, el objetivo del problema de secuenciación es la minimización de la sobrecarga. La sobrecarga puede entenderse como el trabajo que no puede ser completado adentro de una estación cerrada debido a que el tiempo de proceso de la unidad en curso es más grande que el tiempo del que dispone el operario para completarla [Yano y Rachamadugu, 1991].

El problema se estudia asumiendo las siguientes características y políticas del sistema productivo: velocidad constante del transportador, estaciones cerradas, tiempos de proceso deterministas, introducción de productos en la línea a intervalos de tiempo constantes, dos operadores trabajan simultáneamente en cada estación y realizan su trabajo tan pronto como sea posible.

Debido a que se requiere un gran esfuerzo computacional para resolver el problema de forma óptima, en esta tesis se exploran tres diferentes procedimientos heurísticos. En la primera parte se analizan procedimientos constructivos. En la segunda se explora el desempeño de diferentes vecindarios para realizar mejora local. En la tercera parte se ensaya con la unión de dos conceptos: las reglas de prioridad y la búsqueda por entornos.

En cuanto a los procedimientos constructivos, se toman tres procedimientos de la literatura [Yano y Rachamadugu, 1991; Bolat y Yano, 1992], los cuales consideran dos tipos de productos, uno con tiempo de procesamiento mayor al tiempo de ciclo y otro con tiempo de procesamiento inferior al ciclo. Se proponen dos procedimientos para el mismo problema. El primero de ellos (Ud1) es de tipo *greedy*. Conociendo a priori los valores de sobrecarga y ocio obligados, el segundo procedimiento propuesto (Ud2) aplica, en función de la cota, uno de los tres procedimientos: dos *greedy* clásicos propuestos en la literatura y Ud1. En total se tienen 5 procedimientos que se comparan por medio de dos experiencias computacionales. En la experiencia 1, se comparan los procedimientos considerando una sola estación. Se generan 750 instancias. En la segunda experiencia, se consideran estaciones múltiples y se generan 2250 instancias. El segundo procedimiento propuesto presenta el mejor comportamiento cuando se trata una sola estación. Para estaciones múltiples, los mejores resultados se obtienen con Ud1 y con uno de los procedimientos de la literatura (YR).

Continuando con los procedimientos constructivos, se propone extender los procedimientos que produjeron mejores resultados en el caso de dos productos y múltiples estaciones (Ud1 y YR) para que consideren más de dos tipos de productos. En total se tienen tres extensiones de Ud1 y una extensión de YR. Las extensiones de Ud1 incluyen los conceptos de multi-ciclos (con desplazamientos alternativos del operario hacia el inicio y final de la estación), regeneración del sistema (recuperación de la posición inicial del operario en cada pseudo-ciclo) y regularidad (mantener una carga homogénea a lo largo del tiempo, teniendo en cuenta la regularidad de la producción). Estos procedimientos se comparan mediante una experiencia computacional. Las instancias de la experiencia se obtuvieron de la literatura (Scholl et al., 1998). El mejor resultado es obtenido con una de las extensiones de Ud1, la cual procura regularizar (repartir) la carga u ocio obligadas a lo largo de la secuencia.

En la segunda parte de esta tesis se aplican procedimientos de mejora a los resultados obtenidos con los procedimientos constructivos. Se usan dos vecindarios: intercambio de elementos e inserción de segmentos. En la experiencia computacional se analizan varias alternativas de intercambio al igual que con las inserciones. Debido al tamaño del vecindario, los óptimos locales por intercambio de elementos requieren más tiempo de búsqueda que por inserción de segmentos. La mejora local por inserción de segmentos promete mejores resultados.

En la última parte de esta tesis se propone aprovechar las características positivas de dos tipos de heurísticas: las reglas de prioridad y la generación de nuevas soluciones en el espacio de búsqueda. Para ello, se usa como bases el método llamado Búsqueda Dispersa (SS: *Scatter Search*). Las reglas de prioridad se usan para establecer una lista ordenada de productos candidatos a ser secuenciados en cada etapa de decisión. En este trabajo se usan 20 reglas de prioridad. Las reglas consideran aspectos como tiempos de procesamiento, la demanda pendiente, la diferencia absoluta entre el tiempo de proceso de cada producto en cada estación y el tiempo de ciclo, los desplazamientos de los trabajadores, la estación cuello de botella, la sobrecarga, el tiempo improductivo y la regularidad en la producción y en la sobrecarga.

Si bien, el procedimiento propuesto está inspirado en SS, el procedimiento implementado difiere del original en algunos aspectos, por ejemplo, el *RefSet* contiene cadenas de reglas en lugar de soluciones. Se prueban tres procedimientos de regeneración del *RefSet*. El tamaño del *RefSet* depende del número de reglas consideradas. La intensificación se realiza en función a la frecuencia con que cierta regla aparece en cierta posición de la secuencia en las cadenas de reglas que contiene el *RefSet*. También en función de la tabla de frecuencias se realiza el proceso de diversificación. Para probar la eficiencia de la propuesta se realiza una experiencia computacional con las mismas instancias tomadas de Scholl et al. (1998).

Tanto los procedimientos constructivos como la hiperheurística ofrecen soluciones aceptables (respecto a una cota de sobrecarga que debe mejorarse), en pocos segundos. Con la mejora local puede reducirse a la mitad la desviación con respecto a la cota, con el inconveniente de que el tiempo que ello implica aumenta considerablemente. Nuevos vecindarios pueden explorarse para mejorar soluciones en menor tiempo. Diferentes estrategias de combinación y diversificación en al hiperheurística deben identificarse y explorarse.

The object of this thesis is the Mixed-model Assembly-line Sequencing problem. Assembly lines have become a core component in modern repetitive manufacturing systems, especially in *Just in Time* production systems. In particular, mixed-model assembly lines make it possible to manufacture a product with several small variations, avoiding large inventories and setups.

Sequencing problems are important for the efficient use of mixed-model assembly lines. The sequencing for such lines entails the determination of the order in which the products are to be assembled in the line, in relation to specific criteria.

In this work load leveling is considered to be the sequencing criteria [Monden, 1983]. Various objectives have been considered by researchers to measure leveling of the load. The objective used depends on the company management policies and on the constraints imposed by the productive system [Kotani y Ohno, 2004; Okamura y Yamashina, 1979; Sarker y Pan, 1998].

The main sequencing objective treated in this thesis is to minimize the work overload. Work overload can be understood as the work which would not be completed if the assembler were to work at a normal pace and stay within the boundaries of the time window [Yano y Rachamadugu, 1991].

The sequencing problem is studied under the following hypotheses; we assume that the conveyor system is moving at constant speed (paced line), stations are closed, processing times are deterministic, products are launched into the conveyor at a fixed rate, and operator schedule has an early start.

The immense computational effort required for optimally solving real instances, has motivated the development of various heuristics. Three different methodologies are explored in this thesis. Thus, this study can be divided into three sections. Constructive procedures are handled in the first section. In the second section, various kinds of neighborhoods are used in local search. In the last section, we experiment with the combination of two concepts: priority rules and neighborhood search.

Concerning the constructive procedures, three algorithms are taken from literature [Yano y Rachamadugu, 1991; Bolat y Yano, 1992]. These procedures consider two kinds of products, one with processing time greater than cycle time and the other with processing time lesser than cycle time. Two procedures are proposed for this problem. One of them is a greedy procedure (Ud1). The other (Ud2) is based in the fact that prior values of compulsory work overload and idle time can be obtained that can be used to guide previously existing procedures. A total of five procedures are compared in two computational experiments. In experiment one, the performance of the procedures is compared on a testbed of 750 generated instances with one single station. Multiple stations are considered in experiment number two. This time 2250 instances are generated. With regards to the single-station problem, the best results are obtained with the Ud2 procedure. For the multi-station case, one of the literature procedures (YR) and the proposed procedure Ud1 produce the best results.

Since Ud1 and YR give the best results, the corresponding extensions are proposed which consider multiple products as well. Three versions of Ud1 are given and only one of YR. Ud1 extensions include concepts such as multi-cycles (alternative operator displacements from the upstream station limit to the downstream station limit, and vice versa), regeneration (the attempt to recover the original worker position after a cycle) and product load smoothing. A computational experiment is used to test the efficiency of the procedures. Experiment instances are obtained from the literature (Scholl et al., 1998). The best results are obtained with one of the Ud1 extensions, which tries to smooth out the load and idleness along the sequence.

In the second part of this thesis, local search procedures are applied on the results obtained by the constructive procedures. Two kinds of neighborhoods are used: element exchange and segment insertion. In the experiment, a variety of exchange alternatives (2 exchange, 3 exchange and two combinations of 2-3 exchange) and various segment sizes are tested. Since neighborhoods are much larger when the considered moves are element exchange, search procedures based on segment insertion were faster in general. Those procedures also performed better in terms of solution quality.

The combination of two types of heuristics is tested in the last part of this thesis: the priority rules and the generation of new solutions in the search space. The hyperheuristic proposed is based on the Scatter Search metaheuristic (SS). In greedy procedures, priority rules are used to establish an ordered list of candidate products to be sequenced at each decision stage. In this thesis, 20 priority rules are used. These rules consider a selection of problem characteristics or parameters, such as: processing times, pending demand, absolute difference between processing times and cycle time, displacements of workers in the stations, the bottleneck station, work overload, idle time, and smoothness in work overload and production.

This hyperheuristic differs from SS in various aspects. As in SS, the hyperheuristic also uses a reference set, but instead of solutions, the reference set includes chains containing priority rules. The size of the reference set depends on the number of rules considered. By combining the chains of priority rules in the reference set, new ones are produced. The chains obtained by combination are measured and the reference set is regenerated with good quality chains. Three variations of the regeneration process are tested. Intensification and diversification phases are alternatively applied. The regeneration is repeated during the intensification phase. The creation of new chains takes into account the frequency with which a priority rule appears in a certain position in the chains located in the reference set. In the diversification phase, the above mentioned frequencies are also used in an effort to move to a different region in the solutions space. The efficiency of the procedure is also tested with the instances from Scholl et al. (1998).

Constructive procedures and the hyperheuristic obtain solutions in little time. This quality in solutions can be outperformed with local search. Nevertheless, there is a big inconvenient: the search time required in the used neighborhoods is too much. Thus, new neighborhoods could be applied to improve the solutions in less time. Also new chains combination strategies in the hyperheuristic can be explored.

Agradecimientos .....	i
Resumen .....	iii
Abstract .....	v
Introducción.....	1
Capítulo 1 Estado del arte .....	7
1.1 Marco general de secuenciación en líneas de productos mixtos .....	7
1.2 Objetivos en la secuenciación.....	8
1.2.1 Cargas de trabajo .....	11
Capítulo 2 Modelo general .....	19
2.1 Características operativas del problema y supuestos .....	19
2.2 Establecimiento del problema .....	22
2.2.1 Formulación matemática .....	23
Capítulo 3 Procedimientos constructivos .....	27
3.1 Procedimientos para una estación.....	27
3.1.1 Medición de la sobrecarga .....	28
3.1.2 Procedimientos para una estación y productos con opciones .....	29
3.2 Procedimiento para múltiples estaciones .....	37
3.3 Experiencia computacional.....	39
3.3.1 Experimento 1, una estación.....	39
3.3.2 Experimento 2, múltiples estaciones .....	42
3.4 Resultados .....	46
Capítulo 4 Procedimientos para múltiples productos .....	49
4.1 Procedimiento para múltiples productos y una estación.....	49
4.1.1 Extensión del procedimiento de Yano y Rachamadugu .....	49
4.1.2 Extensiones del procedimiento updown .....	52
4.2 Procedimiento para múltiples estaciones y productos .....	59
4.3 Experiencia computacional.....	60
4.4 Resultados .....	67
Capítulo 5 Búsqueda local.....	69
5.1 Optimización combinatoria y búsqueda local.....	69
5.2 Intercambio .....	71
5.3 Inserción .....	73
5.4 Experiencia computacional.....	74
5.5 Resultados .....	81
Capítulo 6 Procedimiento constructivo por combinación de reglas .....	83
6.1 Reglas de prioridad y búsqueda por entornos .....	83
6.2 Procedimiento constructivo por combinación de reglas .....	84
6.3 Búsqueda por entornos y PCCR.....	85

6.4 Conjunto de referencia <i>RefSet</i> .....	87
6.4.1 Reglas de prioridad .....	88
6.4.2 Matriz de frecuencias.....	89
6.5 Combinación de cadenas del conjunto de referencia .....	90
6.6 Regeneración del conjunto de referencia .....	91
6.7 Diversificación.....	92
6.8 Experiencia computacional.....	93
6.9 Resultados .....	102
Capítulo 7 Conclusiones y posibles extensiones .....	105
Referencias.....	109
Anexo 1 .....	115
Anexo 2 .....	116
Anexo 3 .....	117
Glosario .....	119

---

## Índice de figuras

---

Figura 1	Tres niveles de la planificación de la producción.....	4
Figura 2	Esquema de la tesis .....	5
Figura 2.1	Esquema de una cadena de montaje .....	20
Figura 2.2	Diagrama de desplazamiento de los trabajadores en las estaciones .....	21
Figura 3.1	Puntos de cambio de dirección del trabajador para YR.....	32
Figura 3.2	Tendencia de la posición del trabajador para una secuencia de BY1 .....	33
Figura 3.3	Instantes de inicio y finalización con BY1 .....	33
Figura 3.4	Tendencia de la posición del trabajador para una secuencia de BY2 .....	34
Figura 3.5	Instantes de inicio y finalización con Ud1 .....	36
Figura 3.6	Desviación porcentual media para total de unidades.....	40
Figura 3.7	Desviación porcentual media para mixes de producción .....	41
Figura 3.8	Desviación porcentual media para número de estaciones .....	43
Figura 3.9	Desviación porcentual media para mixes de producción .....	45
Figura 3.10	Desviación porcentual media para longitudes de la estación .....	46
Figura 4.1	Puntos de cambio de dirección del trabajador con YRx.....	52
Figura 4.2	Puntos de cambio de dirección del trabajador con Udx .....	54
Figura 4.3	Puntos de cambio de dirección del trabajador con UdxC .....	56
Figura 4.4	Puntos de cambio de dirección del trabajador con UdxR .....	58
Figura 4.5	Sobrecarga acumulada para procedimientos .....	59
Figura 4.6	Sobrecarga para las instancias del grupo 9 .....	63
Figura 4.7	rel.wo1 para grupos y procedimientos .....	64
Figura 4.8	rel.wo2 para grupos y procedimientos .....	65
Figura 4.9	rel.wo1 para número de productos y procedimientos .....	66
Figura 4.10	rel.wo1 para tipo de ventana y procedimiento .....	67
Figura 5.1	2 Intercambio .....	71
Figura 5.2	3 Intercambio .....	72
Figura 5.3	3 Intercambio “reales” .....	72
Figura 5.4	Inserción de segmentos .....	73
Figura 5.5	rel.wo1, para procedimiento de BL y solución inicial.....	76
Figura 5.6	rel.wo2, para procedimiento de BL y solución inicial.....	77
Figura 5.7	rel.wo2, para intercambio de elementos y solución inicial.....	78
Figura 5.8	rel.wo2, para inserción de segmento y solución inicial.....	78
Figura 5.9	rel.wo2, para intercambio de elementos e inserción de segmentos .....	79
Figura 5.10	Óptimos conformados con valor de lbw.....	80
Figura 6.1	Procedimiento constructivo .....	85
Figura 6.2	Esquema básico SS & PCCR.....	86
Figura 6.3	Tiers del conjunto de referencia .....	87
Figura 6.4	Evaluación de las cadenas de reglas .....	88
Figura 6.5	Matriz de frecuencias.....	89
Figura 6.6	Combinación del RefSet .....	91
Figura 6.7	rel.wo2 para tipo de regeneración.....	94
Figura 6.8	rel.wo3 global para tipo de regeneración.....	95
Figura 6.9	rel.wo4 global para tipo de regeneración.....	96
Figura 6.10	rel.wo1 para grupos y tipo de regeneración.....	99
Figura 6.11	rel.wo2 para grupos y tipo de regeneración.....	99
Figura 6.12	rel.wo4 para grupos de parámetros .....	101



---

## Índice de algoritmos

---

Algoritmo 3.1	YR para una estación y productos con opción.....	31
Algoritmo 3.2	BY1 para una estación y productos con opción.....	32
Algoritmo 3.3	BY2 para una estación y productos con opción.....	34
Algoritmo 3.4	Ud1 .....	35
Algoritmo 3.5	YR para múltiples estaciones.....	38
Algoritmo 4.1	Udx .....	53
Algoritmo 4.2	UdxC.....	55
Algoritmo 4.2	UdxR.....	57
Algoritmo 4.4	Extensión para múltiples productos y estaciones .....	60
Algoritmo 5.1	Búsqueda local.....	71
Algoritmo 5.2	Búsqueda local para inserción de segmentos.....	73
Algoritmo 6.1	PCCR.....	85
Algoritmo 6.2	Pseudo código SS & PCCR.....	87
Algoritmo 6.3	Diversificación del RefSet.....	92



---

## Índice de tablas

---

Tabla 1.1	Clasificación de los problemas de secuencias regulares en MMAL .....	9
Tabla 1.2	Tabla resumen del estado del arte .....	17
Tabla 3.1	Secuencia obtenida con YR .....	31
Tabla 3.2	Secuencia obtenida con BY1 .....	33
Tabla 3.3	Secuencia obtenida con BY2 .....	34
Tabla 3.4	Secuencia obtenida con Ud1 .....	36
Tabla 3.5	Índices de calidad para el experimento 1 para total de unidades.....	39
Tabla 3.6	Promedios de sobrecarga y ocio para el experimento 1 .....	40
Tabla 3.7	Índices de calidad para el experimento 1 para ratios de mix .....	41
Tabla 3.8	Índices de calidad para el experimento 1 para longitudes de estación ....	42
Tabla 3.9	Índices de calidad para el experimento 2 para número de estaciones ....	43
Tabla 3.10	Índices de calidad para el experimento 2 para total de unidades.....	44
Tabla 3.11	Índices de calidad para el experimento 2 para ratios de mix .....	44
Tabla 3.12	Índices de calidad para el experimento 2 para longitudes de estación ....	45
Tabla 4.1	Secuencia obtenida con YRx .....	51
Tabla 4.2	Secuencia obtenida con Udx.....	53
Tabla 4.3	Secuencia obtenida con UdxC .....	55
Tabla 4.4	Secuencia obtenida con UdxR .....	58
Tabla 4.5	Suma de desviaciones absolutas para procedimientos.....	59
Tabla 4.6	Características de la batería de instancias.....	61
Tabla 4.7	Desviación relativa global .....	62
Tabla 4.8	Desviación relativa global para instancias con producto básico .....	63
Tabla 4.9	Desviación relativa global para grupos.....	64
Tabla 4.10	Desviación relativa global para parámetros.....	65
Tabla 5.1	Desviación relativa global rel.wo1 .....	75
Tabla 5.2	Desviación relativa global rel.wo2 .....	76
Tabla 5.3	rel.wo2 para intercambios.....	77
Tabla 5.4	rel.wo2 para inserciones .....	78
Tabla 5.5	Comparativo rel.wo2, para intercambios e inserciones .....	79
Tabla 5.6	Óptimos conformados con valor de lbw.....	80
Tabla 6.1	Desviación relativa global .....	94
Tabla 6.2	Óptimos alcanzados para tipo de regeneración del RefSet.....	97
Tabla 6.3	Desviación relativa para grupos.....	98
Tabla 6.4	Desviación relativa para parámetros.....	100



La industria de la automoción se ha caracterizado por pertenecer a un sector industrial altamente competitivo que tiene influencia considerable en la economía de muchos países<sup>1</sup>. Esta industria se mantiene en la vanguardia de la tecnología y gestión de los recursos para producir bienes y servicios.

La dirección de operaciones es fundamental en estas empresas (Chase y Aquilano, 2000; Ruiz et al., 1995a). El diseño y la manufactura de automóviles es un proceso complejo que involucra innovaciones científico-técnicas muy costosas y que debe adaptarse constantemente a las demandas cambiantes de los mercados. La competitividad en este sector requiere reducción de costes y mejora de precios, así como la reducción de los tiempos de espera.

Los costes y tiempos de producción dependen directamente de la forma de gestionar los recursos directos que la empresa requiere. Dentro de una organización, la función a cargo de la gestión de estos recursos se denomina dirección de operaciones. Su objetivo clave es producir un bien específico a tiempo y coste mínimos.

Alcanzar este objetivo requiere tomar decisiones complejas. La investigación de operaciones (OR: *Operational Research*) ofrece herramientas cuantitativas que se emplean no solamente en la dirección de operaciones sino en diversas disciplinas empresariales.

El trabajo de esta tesis se enfoca a los sistemas productivos físicos de manufactura continuos, con producción-distribución de artículos normalizados con alto volumen de producción.

El interés en este tipo de sistemas se concentra en la gestión de los mismos. Es necesario diferenciar dos grandes categorías que permiten establecer una clasificación del conjunto de problemas de gestión en sistemas productivos (Bautista, 1993; Companys, 2003a): decisiones estratégicas y decisiones tácticas.

Las decisiones estratégicas, también denominadas decisiones a largo plazo se relacionan con el diseño, la planificación, la construcción del sistema destinado a la producción de bienes y servicios, y con los cambios en los productos. En otras palabras, las decisiones estratégicas tienen que ver con los medios y con los fines del sistema. Estas decisiones contemplan la previsión y la planificación a largo plazo, la selección de equipos y procesos, el diseño productivo de elementos, el diseño de tareas y medición del trabajo, la localización del sistema, la distribución de planta, o la fiabilidad previsional.

Por otro lado, las decisiones tácticas o decisiones a medio plazo o dirección de operaciones, se relacionan con el funcionamiento y el control del sistema productivo. Entre estas decisiones están: la previsión de la demanda, la planificación y programación de

---

<sup>1</sup> La Unión Europea, por ejemplo, con el 34 % de la producción mundial, es la región que más automóviles produce. La industria automotriz conforma el 7.5% del total en la UE.  
<http://europa.eu.int/comm/enterprise/automotive/pagesbackground/sectoralanalysis/index.htm>

## Introducción

operaciones, lanzamiento y control de operaciones, fiabilidad estimada y operacional, control de costes y de RRHH.

Esta tesis se enmarca en la dirección de operaciones, específicamente en la planificación y programación de operaciones o planificación táctica de la producción.

Se entiende por planificación táctica de la producción (PTP), como la función que somete a un estudio científico y detallado a un conjunto de operaciones desarrollables en cierto dominio, con el fin de satisfacer la demanda de productos prevista por la empresa. Dichas operaciones implican decisiones y cálculos que concretan y distribuyen cualitativa, cuantitativa, local y temporalmente, las tareas de producción y de consumo de los medios controlables (Bautista, 1993).

Como herramientas de planificación han surgido diferentes metodologías, también llamadas filosofías de gestión, entre las que mencionamos de forma breve las más consolidadas:

- *Material Requirement Planning* (MRP), que surgió a mediados de los años setenta como una herramienta para calcular las necesidades temporizadas de materiales.
- Hacia los 80's apareció el *Manufacturing Resource Planning* (MRP II) como una extensión del MRP. El propósito inicial de esta metodología era planificar y supervisar todos los recursos de la empresa de manufactura (manufactura, finanzas, mercadotecnia, producción, recursos humanos) por medio de un sistema de ciclo cerrado que puede generar cifras financieras.
- *Optimized Production Technology* (OPT). Fue propuesto por Eliyahu M. Goldratt en 1979. OPT se enfoca a la detección del (los) recurso(s) del sistema que es (son) cuello(s) de botella. Estos recursos limitan la capacidad productiva del sistema, ya que marcan el ritmo de producción y por lo tanto la tasa a la que la empresa gana dinero (Goldratt, 1992). A partir de este sistema se desarrolló posteriormente la Teoría de Restricciones (TOC: *Theory of Constraints*).
- *Just in Time* (JIT). También conocido como sistema de producción Toyota, fue creado en la Toyota Motor Corporation y difundida por Tai-ichi Ohno. Este sistema se desarrolló después de la crisis del petróleo de 1973 en la Toyota Motor Corporation. La idea de este sistema de producción es mantener en las fábricas un flujo continuo de productos, para adaptarse flexiblemente a los cambios en la demanda. "El *Just in Time* produce sólo los artículos necesarios en la cantidad necesaria en el tiempo en que son necesarios" (Monden, 1983). Los objetivos principales del sistema de gestión de la producción son la reducción de costes y el incremento de productividad global de la empresa, coadyuvando así con otros tres objetivos: control cuantitativo, calidad asegurada y respeto por la dimensión humana. Si se concibe al JIT como una ideología, se deben distinguir al menos dos máximas:
  - a. La reducción de costes mediante la eliminación de desperdicios.
  - b. La utilización al máximo de las capacidades, ideas y opiniones de los operarios.

El JIT es el principal ámbito de aplicación del trabajo desarrollado en esta tesis.

En el dominio de la PTP (Bautista, 1993) deben tomarse muchas decisiones, con diferentes fines, objetivos, y gran variedad de medios o recursos involucrados y por supuesto, bajo limitaciones o restricciones. Al haber tantos elementos a contemplar simultáneamente para tomar una decisión, se torna muy difícil contestar de la mejor manera al ¿Qué?, ¿Cuánto?, ¿Dónde? y ¿Cuándo? producir los bienes y/o servicios que la empresa elabora, teniendo en cuenta la validez temporal de estas decisiones.

Para encontrar soluciones entre tanta información se han propuesto metodologías basadas en la jerarquización de las decisiones. La jerarquización divide el problema inicial en sub-problemas, con un orden coherente de resolución. Cada sub-problema responde de forma secuencial a decisiones con menor alcance temporal y afectando, generalmente, a menos RRHH y recursos productivos pero con mayor incidencia en los detalles del sistema productivo.

Las decisiones pueden agruparse en tres funciones básicas:

1. Planificación de operaciones.
2. Cálculo de necesidades de materiales y recursos.
3. Programación de la producción.

Dentro del tercer punto, la programación de la producción, es donde se desarrolla esta tesis, ya que, como se verá más adelante, es en esta función donde tiene lugar la secuenciación.

La programación de la producción es la función que se encarga de asignar las órdenes de producción (o las operaciones que las componen) a centros de trabajo específicos dentro de intervalos de tiempo concretos. El resultado de la programación de la producción es un programa de producción en el que se establece una asignación de tareas más un calendario.

A su vez, para realizar la función de programación de la producción, Bautista (1993) identifica tres sub-funciones:

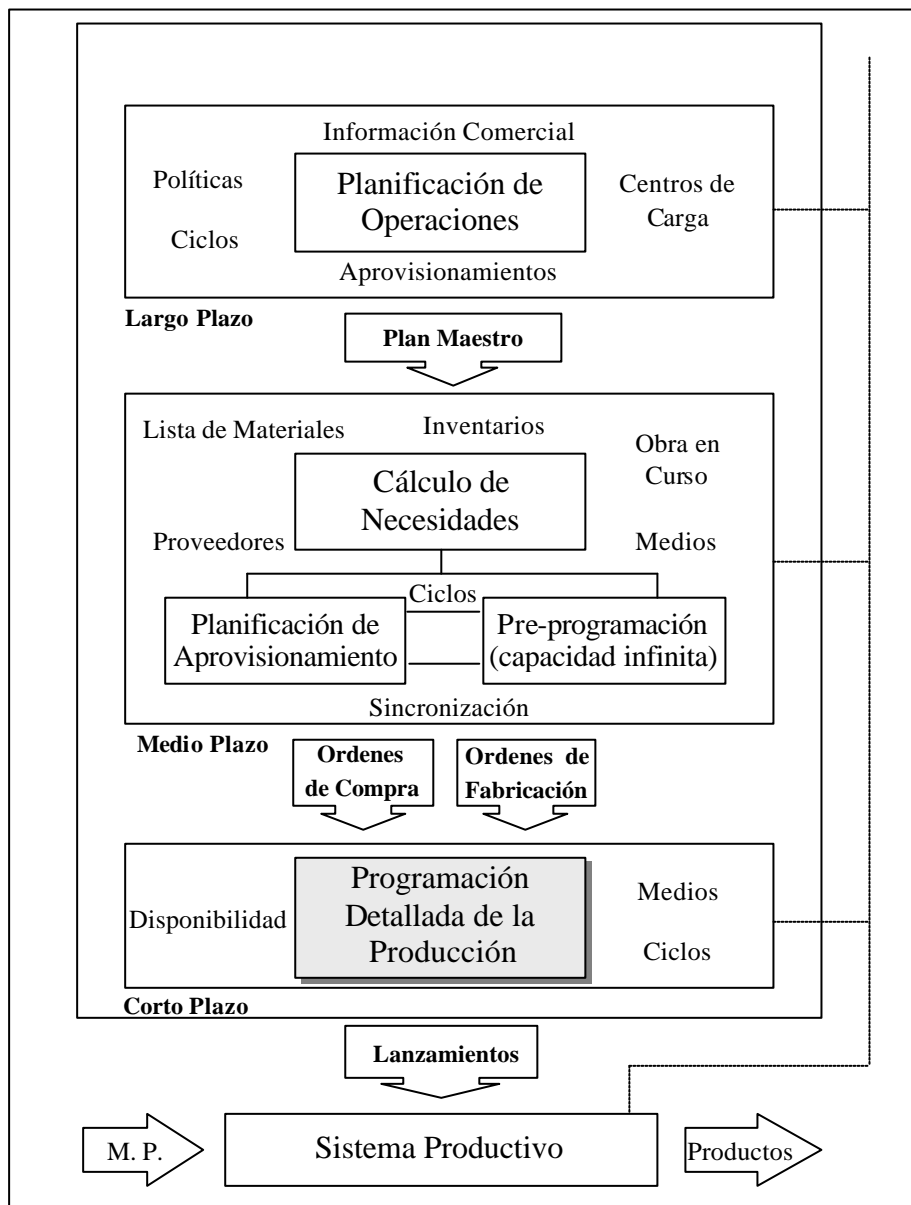
- Carga (*loading*): asignación de las operaciones a centros de trabajo, decisión que se adoptará por comparación entre la capacidad disponible del centro y la carga requerida por las operaciones ya asignadas al mismo.
- Secuenciación (*sequencing*): secuenciación de las operaciones asignadas a centros de trabajo para establecer su orden de ejecución.
- Temporización (*scheduling*): determinación de los instantes de inicio y fin (programados) de cada operación.

Esta tesis se enfoca al tercer nivel de la planificación de la producción, la programación detallada de la producción, específicamente a la secuenciación de unidades homogéneas.

La forma de establecer la secuencia depende del tipo de configuración del sistema productivo, por lo que existen diferentes opciones. Las líneas de montaje corresponden a los sistemas productivos puros con orientación al producto. Desde los primeros trabajos sobre secuenciación de unidades en líneas de montaje en los años 60's (Wester y Kilbridge, 1963; Thomopoulos, 1967), han aparecido diferentes enfoques y objetivos para

obtener la secuenciación en estos sistemas productivos. Los creadores del JIT plantearon dos objetivos en la secuenciación para adaptarse al sistema de producción: nivelar la carga y nivelar el consumo de componentes. Desde nuestro punto de vista, la secuenciación con cargas de trabajo regular ha recibido menos atención de los investigadores en los últimos años. Esta tesis explora más este objetivo, el cual también ha sido estudiado desde diferentes perspectivas debido a la amplia variedad en la configuración que puede tomar una línea de montaje y a las políticas que afectan al sistema productivo ante este criterio que debe tener en cuenta los RRHH.

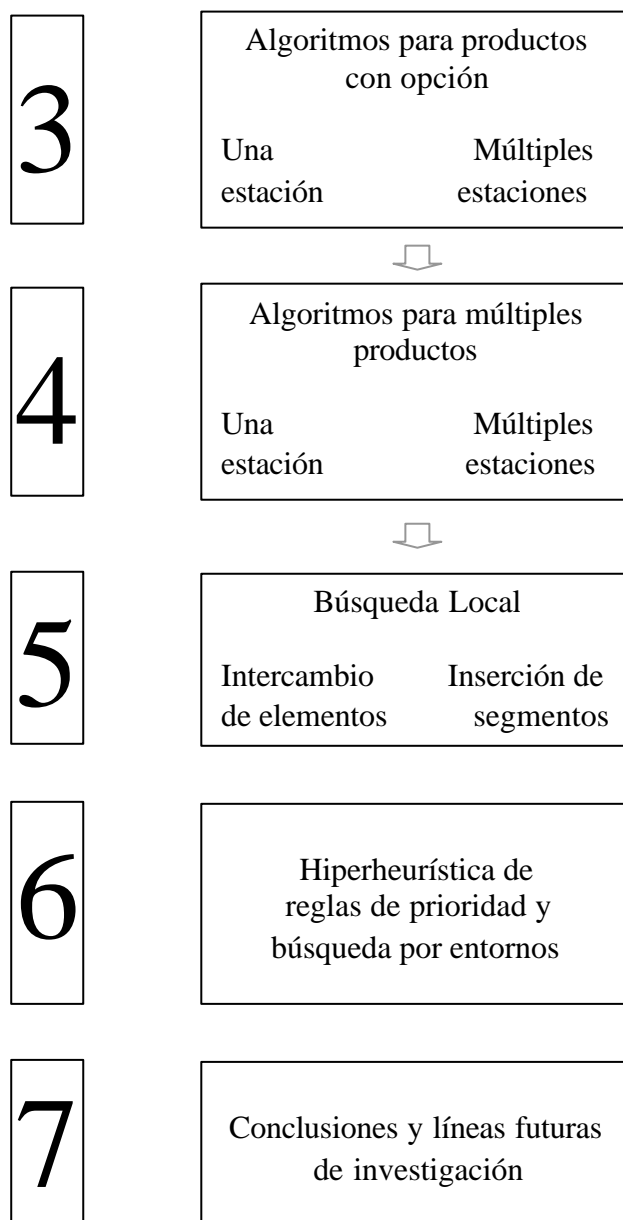
Figura 1 Tres niveles de la planificación de la producción<sup>2</sup>



<sup>2</sup> Figura tomada de Bautista, 1993.

El primer capítulo de esta tesis se centra en revisar la bibliografía sobre secuenciación en líneas de montaje, con énfasis en los trabajos dedicados a la minimización de la sobrecarga de trabajo en la línea. En el capítulo 2 se establecen los supuestos e hipótesis con los que se delimita y caracteriza el problema que se estudia en este trabajo. En el capítulo 3 se describen algunos procedimientos de la literatura y se proponen otros, considerando que en la línea se elaboran productos con y sin componentes opcionales. En el capítulo 4 se considera que en la línea se elaboran productos/modelos múltiples y se proponen procedimientos que son extensiones de los mencionados en el capítulo tres. En el capítulo 5 se describen procedimientos de búsqueda local usados para mejorar las soluciones obtenidas en los procedimientos constructivos del capítulo anterior. En el capítulo 6 se describe un procedimiento hiperheurístico que combina reglas de prioridad con búsqueda por entornos. En el capítulo 7 se establecen las conclusiones de las experiencias computacionales descritas en anteriores capítulos de la tesis y se vislumbran posibles extensiones de investigación sobre el problema.

**Figura 2** Esquema de la tesis





A continuación se hace una descripción del marco general de la secuenciación de unidades mixtas en líneas o cadenas de montaje. Se señalan diferentes criterios que se han sido considerados en la literatura para resolver este tipo de problemas y se mencionan algunas clasificaciones que se han realizado al respecto. Posteriormente, se revisa la literatura que trata el objetivo de secuenciación con el que se trabaja en esta tesis.

### 1.1 Marco general de secuenciación en líneas de productos mixtos

En los sistemas productivos continuos puros la problemática de la programación se centra prácticamente en la asignación de las operaciones a centros de trabajo, al establecimiento de la ruta y al equilibrado de la línea. Pero, si por la línea transitan productos homogéneos (productos del mismo tipo con algunas variaciones de una unidad a otra) también es necesario establecer la secuencia de dichos productos. Esto sucede por ejemplo en el montaje de automóviles.

En los sistemas de gestión de producción tipo arrastre (*pull*), concretamente, dentro del marco de la filosofía de producción JIT, la secuenciación de operaciones se convierte en el aspecto fundamental de la programación de operaciones. El tipo de problema tratado en esta tesis se puede caracterizar como un problema de ordenación con objetivos representables mediante alguna medida de eficiencia.

Así, el sistema productivo representativo en este trabajo es una línea de montaje. Una línea de montaje común consiste en un montaje progresivo que pasa por una serie de estaciones (Chase y Aquilano, 2000). El trabajo que se realiza en cada estación se compone de pequeñas porciones de trabajo llamadas tareas, elementos y unidades de trabajo. En las líneas de montaje, el orden de introducción de los productos que han de montarse será el mismo en todas las estaciones y se asume que todos los tipos de productos siguen la misma ruta.

Dos de las actividades críticas para la gestión eficiente de una línea de montaje son el equilibrado y la secuenciación (Monden, 1983; Thomopoulos, 1976; Okamura y Yamashina, 1979). Monden (1983) menciona que el diseño de una línea de montaje de productos mixtos implica al menos los siguientes pasos:

1. Determinación del ciclo de fabricación.
2. Cálculo del número mínimo de procesos.
3. Preparación de un diagrama de las relaciones integradas de precedencia entre tareas elementales.
4. Equilibrado de la línea.
5. **Determinación del programa de secuencias para introducir diversos productos en la línea.**

Xiaobo et al. (1999) también puntualizan que “la secuenciación de los modelos que serán ensamblados en un transportador es reconocida como una actividad clave para mejorar el desempeño de las líneas de montaje”<sup>3</sup>.

### 1.2 Objetivos en la secuenciación

Una vez que se ha tomado la decisión sobre la cantidad de las unidades a producir (programa de producción o mix de producción), el paso siguiente es establecer una secuencia de introducción de las unidades en la línea.

Para ello, es necesario tener alguna meta o propósito de control de la línea para decidir el orden de introducción de las unidades. Dentro del contexto JIT se contemplan básicamente dos criterios:

1. Nivelar la carga (tiempo total de operación) en cada proceso dentro de la línea.
2. Mantener un ritmo constante de consumo de cada pieza requerida en la línea.

El objetivo uno considera que hay productos que requieren tiempo de proceso mayor que el tiempo de ciclo. Esto se debe al hecho de que el equilibrado de la línea de productos mixtos se hace con la condición de que el tiempo de operación de cada proceso, que se ponderó para cada conjunto de productos mixtos, no debe ser superior al tiempo de ciclo. Dado que el tiempo de ciclo es el tiempo que transcurre entre la entrada de dos productos consecutivos a la línea (o a cualquier estación), si se introducen sucesivamente productos con tiempo de proceso mayor al ciclo se puede producirse un retraso en la terminación de alguno de esos productos. En otras palabras, habrá exceso de carga en la línea.

El segundo objetivo busca mantener una tasa constante en el consumo de componentes por periodos de tiempo. Este objetivo es el mejor camino para ajustarse al concepto de producción de Toyota. Debido al sistema de arrastre del JIT es necesario reducir al mínimo las variaciones en las cantidades de producción o los tiempos de transporte en los procesos anteriores. De la misma forma, debe reducirse al mínimo las existencias de obra en curso.

Kubiak (1993) clasificó los problemas de secuenciación en dos categorías: *Product Rate Variation* (PRV) y *Output Rate Variation* (ORV). El PRV es una visión simplificada sobre los problemas de secuencias regulares que tiene poco sentido cuando hay muchas variantes del producto. Se caracteriza la regularidad mediante una función asociada a los productos o conjuntos de ellos. La secuencia obtenida con este método debe minimizar la suma de las discrepancias (que pueden ser cuadráticas) entre las producciones reales y las ideales. El ORV surge al caracterizar la regularidad mediante una función asociada al consumo de componentes o recursos. Una de varias alternativas para medir la irregularidad en el consumo de recursos es sumar los cuadrados de las discrepancias entre los consumos reales y los consumos ideales de todos los componentes para todas las posiciones de la secuencia. El problema consiste en determinar una secuencia que contenga todas las unidades del plan y que minimice la función de No-regularidad (SQD).

---

<sup>3</sup> “Sequencing for mixed models to be assembled on the conveyor is recognized as an important work for improving the performance of an assembly line” (Xiaobo et al., 1999).

Scholl (1995) presenta diferentes objetivos de secuenciación que se han considerado en la literatura para ayudar a encontrar secuencias que cumplan ciertos requisitos:

- Objetivos relacionados con el tiempo: sobrecarga total, tiempo inhábil total, equivalencia de objetivos (sobrecarga / tiempo inhábil), objetivos similares, longitud de la línea.
- Objetivos de costes: coste laboral total, coste operativo total, equivalencia entre objetivos de costes y objetivos relacionados con el tiempo.
- Objetivos JIT: Requerimientos de partes similares, requerimientos de partes diferentes, combinación de metas relacionadas con el tiempo.

Bautista et al. (1996c) extienden la clasificación (ver Tabla 1.1) de Kubiak (1993) de los problemas de secuenciación en función de dos aspectos: el objetivo al que está enfocada la regularización (productos o recursos) y el medio para definir la regularidad.

**Tabla 1.1** Clasificación de los problemas de secuencias regulares en MMAL<sup>4</sup>

	Productos	Recursos		
		Componentes		RRHH
		Un Nivel	Multi-nivel	
Propiedad	CP	CO	CMO	CL
Función	PRV	ORV	MORV	LRV
Mixta	CPRV	CORV	CMORV	CLRV

Las columnas de la Tabla 1.1 se refieren al objetivo al que se enfoca la regularización: Productos o Recursos (Componentes o Cargas de trabajo); las filas se refieren a la forma de definir la regularidad: (1) propiedades o restricciones tales que las secuencias que las satisfacen se definen como regulares; (2) funciones para medir el nivel de regularidad o irregularidad; y (3) la combinación de ambas formas regularizar. Las letras empleadas representan las palabras en inglés: C (*Constrained*), RV (*Rate Variation*), P (*Product*), O (*Output*), L (*Load*) y M (*Multilevel*).

Xiaobo et al. (1999) agregan dos objetivos más a la clasificación hecha por Monden:

3. Mantener una tasa de alimentación constante para cada producto en la cadena de montaje. Este criterio se deriva del objetivo número dos de Monden (1983).
4. Minimizar el tiempo total de paro del transportador. Este objetivo surge del concepto de automatización del sistema de producción Toyota.

Los problemas de secuenciación han sido tratados desde hace varias décadas, desde los trabajos pioneros de Wester y Kilbridge (1963), Thomopoulos (1967), Dar-El (1978) o Okamura y Yamashina (1979), considerando ya desde los inicios diferentes criterios de secuenciación, como la minimización del coste total de ineficiencias, evitar congestiones en la línea o la reducción de la longitud total de la cadena.

<sup>4</sup> Del inglés, *Mixed-Model Assembly Line*, líneas de montaje de productos mixtos.

El conjunto de trabajos en los que el objeto de la regularidad es el consumo regular de componentes inició formalmente con Monden (1983). Kubiak (1993) le denominó ORV a este problema. Los estudios continuaron con Miltenburg y Sinnamon (1989), Companys (1989), Sumichrast y Russell (1990), Miltenburg y Goldstein (1991), Bautista (1993), Bautista et al. (1995, 1996), Duplaga et al. (1996), Aigbedo y Monden (1997), Poler et al. (1999), Xiaobo et al. (1999), entre otros.

El criterio 3 minimiza la variación de la tasa en la que los diferentes productos están presentes en cualquier segmento de la secuencia o dicho de otra forma, considera la regularidad en la introducción de productos en la línea. El problema fue presentado por Miltenburg (1989) y su estudio fue continuado, entre otros, por Miltenburg et al. (1990), Sumichrast y Russell (1990), Kubiak y Sethi (1991), Inman y Bulfin (1991), Bautista et al. (1992), Ding y Cheng (1993a, 1993b), Steiner y Yeomans (1993), Bautista et al. (1993), Kubiak (1993), Cheng y Ding (1996), o Bautista et al. (2001).

La minimización del riesgo de paro de la línea es una variante de los problemas en la que se consideran las cargas de trabajo en las estaciones de la línea como criterio de secuenciación. Algunos de los autores que lo han estudiado son Okamura y Yamashina (1979) y Tsai (1995). Básicamente, el problema consiste en minimizar la máxima distancia que el trabajador recorre hacia el final de la estación.

Un enfoque alternativo para problemas de cargas, es el CSP (*Car Sequencing Problem*) que, básicamente, busca satisfacer restricciones (Kis, 2004). Algunos trabajos sobre este tema son Coffman et al. (1985), Bolat y Yano (1992a), Bautista et al. (2001), o Kis, (2004). En este esquema, se buscan secuencias que cumplan con las restricciones de consumo de componentes bajo ciertas reglas referentes al número de veces que cada producto puede ser asignado en tramos de la secuencia.

Drexl y Kimms (2001), estudian el CSP conjuntamente con el nivelado en el consumo de componentes (*level-scheduling problem*). Ofrecen una formulación de programación entera que considera ambos criterios. Los autores relajan el problema y lo resuelven usando un esquema de generación de columnas. Considerando estos dos mismos objetivos, Drexl et al. (2006) proponen un algoritmo de ramificación y acotamiento (B&B, *Branch and Bound*) que encuentra soluciones factibles para el CSP. También presentan un algoritmo que optimiza el problema del nivelado en el consumo de partes considerando las restricciones del CSP.

Otro criterio que algunos autores como Bolat y Yano (1992a) o Sarker y Pan (1998) consideran en problemas de cargas es al tiempo improductivo, inhábil u ocioso (*Idle Time* o *Idleness*). Aunque el principal criterio de esta tesis es la minimización de la sobrecarga, en los capítulos 3 y 4 también se tiene en cuenta la reducción del tiempo.

A continuación se describe brevemente la literatura que considera entre sus criterios de secuenciación las cargas de trabajo, criterio considerado en esta tesis.

### 1.2.1 Cargas de trabajo

El primer criterio que menciona Monden (1983) como objetivo de secuenciación en un entorno JIT es nivelar la carga de trabajo en cada proceso en las líneas de trabajo de productos mixtos.

Al haber diferentes modelos de un mismo producto que deben ser ensamblados, algunos tendrán un tiempo de proceso<sup>5</sup> mayor que el tiempo de ciclo<sup>6</sup> en alguna o en varias estaciones de la línea de montaje. En consecuencia, si se introducen consecutivamente varios productos con tiempos de proceso mayores al tiempo de ciclo, llegará un momento en que no se tenga tiempo suficiente en la estación para terminar el trabajo sobre alguno de estos productos y el trabajo quedará incompleto cuando el producto salga de la estación.

Este trabajo que excede la capacidad o disponibilidad de trabajo en las estaciones recibe entre otros nombres, el de sobrecarga (*Work Overload*; Yano y Bolat, 1989; Yano y Rachamadugu, 1991), trabajo perdido (*Remaining Work*; Bolat, 1997) o trabajo utilitario<sup>7</sup> (*Utility Work*; Bolat, 1994; Tsai, 1995; Hyun et al., 1998). Cuando la filosofía de gestión de la empresa obliga a terminar el trabajo dentro de las estaciones sin recibir ayuda extra se usa el término: paro de línea (*Conveyor Stoppage*; Xiaobo y Ohno, 1997; Celano et al., 2004), ya que los trabajadores tienen la obligación de detener el avance de la línea para poder terminar el trabajo sobre los productos dentro de la estación.

#### *Procedimientos exactos*

Los problemas que consideran las cargas de trabajo como criterio de secuenciación, pueden resolverse de forma óptima con programación dinámica o B&B (Xiaobo y Ohno, 1997; Bolat, 2003). De hecho, hay trabajos que usan métodos exactos para resolver el problema considerando objetivos adicionales como lo hace Korkmazel y Meral (2001) quienes también tienen en cuenta el consumo regular de componentes.

En Yano y Rachamadugu (1991) se describe un procedimiento de programación dinámica que minimiza la sobrecarga y se indica que la complejidad computacional del procedimiento incrementa con el cuadrado del número de trabajos y exponencialmente con el número de estaciones. Los autores sólo consideran dos tipos de trabajos en las estaciones.

Cuando se usa la enumeración implícita (B&B) se han aplicado estrategias de búsqueda en profundidad para la selección de nodos para generar soluciones factibles rápidamente. Sin embargo, en el problema que se estudia en este trabajo, cualquier secuencia es factible. Bolat (1997a) reporta que se requiere un gran esfuerzo computacional y los requerimientos de memoria son enormes para un problema con 20 productos y 8 estaciones.

---

<sup>5</sup> El tiempo de proceso o tiempo de procesamiento, es el tiempo necesario para realizar una operación o tarea, sobre el producto que se desplaza sobre la línea, dentro de una estación.

<sup>6</sup> El tiempo de ciclo  $c$  de una línea de montaje, es el intervalo de tiempo constante entre el lanzamiento de dos unidades consecutivas en la línea.

<sup>7</sup> Se entiende por trabajo utilitario o trabajo en exceso al que debe realizar uno o más operarios en reserva o trabajadores auxiliares con intervención puntual cuando se produce una sobrecarga en alguna estación de la línea.

Por ello, en la literatura se encuentra una mayor cantidad de trabajos que se enfocan al uso de heurísticas y metaheurísticas que aquellos que ofrecen procedimientos exactos. Como se describe más adelante, algunos de estos procedimientos heurísticos obtienen soluciones óptimas para ciertos parámetros específicos o ciertas condiciones de sobrecarga o descarga.

### *Procedimientos heurísticos*

Yano y Bolat (1989), es uno de los primeros trabajos que tiene como objetivo la minimización de la sobrecarga. Los autores realizan una revisión sobre algoritmos para secuenciación en cadenas de velocidad constante y proponen una heurística prospectiva.

El objetivo considerado en esta tesis también fue estudiado por Yano y Rachamadugu (1991). En su trabajo muestran una formulación matemática para el problema de secuenciación, que tiene como objetivo minimizar la sobrecarga total ocasionada por la secuencia. Consideran dos tipos de productos: básicos y opcionales. Los primeros tienen tiempos de procesamiento inferior al tiempo de ciclo, mientras que los segundos tienen tiempo de procesamiento mayor.

Los autores proponen un procedimiento que utiliza cotas inferiores de la función objetivo para determinar de manera secuencial el trabajo que debe programarse en la siguiente posición de la secuencia. El algoritmo permite determinar los efectos de la longitud de la estación en la sobrecarga de trabajo, la cual, a su vez, afecta a la calidad de los productos. Para plantear la resolución del problema, los autores consideran una única estación. Los autores proponen otra heurística basada en el procedimiento de una sola estación para el caso en que se tienen  $K$  estaciones. El algoritmo requiere tiempo  $O(KN)$ .

Bolat y Yano (1992a) proponen un grupo de procedimientos de secuenciación. Un procedimiento se basa en el concepto de reglas de espacios. Se crean sub-secuencias que procuran que el trabajador finalice en la misma posición de la estación en la que inició la sub-secuencia. De este modo se establecen patrones con trabajo de refuerzo (utilitario) y tiempo inhábil igual a cero. El resto de las unidades se secuenciarán distribuyéndolas por toda la secuencia, dando preferencia a su ubicación al final de la misma. Los procedimientos brindan soluciones óptimas en determinadas circunstancias como en los programas muy descargados o muy sobrecargados.

Propone dos algoritmos *greedy*<sup>8</sup>. El primero es óptimo para el mismo objetivo, pero únicamente para determinadas proporciones de productos con opción sobre el total. Este algoritmo consiste en secuenciar un producto con opción (si éste existe) mientras no genere sobrecarga, en caso contrario secuenciar un producto básico y, si no existiera, se toma un producto con opción.

El segundo algoritmo *greedy* incorpora a la función de selección anterior la condición de no secuenciar un producto básico si genera tiempo inhábil o tiempo improductivo del trabajador. Los autores utilizan la programación dinámica para evaluar en qué condiciones es útil cada heurística.

En otro artículo del mismo año, Bolat y Yano (1992b) proponen una función objetivo distinta: minimizar el número total de unidades con opción por encima de las  $k$  permitidas

---

<sup>8</sup> Es un término del inglés que normalmente se traduce como codicioso, avaro, goloso, miope o voraz.

en cada tramo de secuencia consecutiva de tamaño  $l$ . Los autores explican que la ventaja fundamental del uso de esta función objetivo es la disminución del coste computacional requerido en su evaluación.

Con la finalidad de detectar las estaciones críticas respecto a la sobrecarga en una cadena de montaje, Rachamadugu y Yano (1994) desarrollan un método analítico para calcular un índice que caracteriza a cada una de las estaciones y permite discriminarlas para incluirlas o excluirlas en un algoritmo de secuenciación que regulariza la carga de trabajo.

Bolat (1994) considera no sólo la línea de *Trim* de la línea de montaje de automóviles, sino también la de pintura. Encuentra una secuencia en dos etapas. En la primera minimiza el trabajo de refuerzo en la línea de *Trim* con una heurística. El autor considera dos tipos de trabajos en las estaciones. En la segunda etapa se incluye la sección de pintura y se desarrollan dos heurísticas. La primera de estas dos, incorpora restricciones de tamaño de lote de color óptimo y después establece la secuencia de trabajos para cada lote mediante el uso del algoritmo de *Trim*. La segunda heurística es una extensión del procedimiento usado en *Trim* que considera costes de preparación.

Tsai (1995) analiza el problema de secuenciación de mezcla de productos con tiempo de ciclo preestablecido y tiempos de procesamiento deterministas. Minimiza dos objetivos: riesgo de paro de la línea y el trabajo en exceso total. Ambos problemas son del tipo *NP-hard*. Para una sola estación y dos tipos de productos (con o sin opción), el algoritmo propuesto por Tsai genera una secuencia en  $O(\log N)$  que es óptima para los dos objetivos. Como resultado de minimizar el trabajo incompleto, el algoritmo minimiza el tiempo inhábil total entre la llegada del primer producto y la salida del último. Los valores de la función objetivo del caso de una sola estación pueden usarse como cotas inferiores para facilitar el desarrollo del algoritmo para problemas de múltiples criterios, y múltiples estaciones. El algoritmo también puede usarse para analizar los efectos en los cambios en la longitud de la estación sobre el riesgo de paro del transportador, el trabajo incompleto y el tiempo inhábil. El autor toma en cuenta la velocidad de desplazamiento del trabajador desde un producto hasta el siguiente.

Guerre et al. (1995) proponen un planteamiento diferente del problema. Consideran tres objetivos distintos: 1) cumplir fechas de entrega, 2) equilibrar cargas de trabajo y 3) evitar que un producto esté demasiado tiempo esperando a ser secuenciado<sup>9</sup>. El equilibrio entre los tres objetivos se realiza ponderándolos mediante costes asociados. La función objetivo planteada es similar a la de Bolat y Yano (1992b), pero se considera que cada estación tiene costes diferentes. Los autores explican el problema de secuenciación dinámica.

Bolat (1997a) adopta el enfoque de los EEUU para completar el trabajo faltante en las líneas de montaje, minimizan la cantidad total de trabajo que deben realizar los operadores de refuerzo. Proponen un algoritmo estocástico que es una combinación del recosido simulado (SA: *Simulated Annealing*) y métodos de conocimiento específico. Para problemas prácticos, con frecuencia se obtienen soluciones óptimas. En (1997b) el autor aborda el problema estableciendo grupos de trabajo mínimos<sup>10</sup> que se repiten de forma cíclica. Propone un algoritmo óptimo y una heurística que usa diferentes funciones de prioridad basadas en tiempos de procesamiento, sin embargo, el procedimiento requiere un

<sup>9</sup> Guerre, et al. consideran que existen almacenes o stocks entre las estaciones de la línea.

<sup>10</sup> Minimum Job Set (MJS), es el grupo mínimo de trabajos que mantiene la misma proporción de la producción.

tiempo de búsqueda enorme y demanda gran cantidad de memoria para un problema que considera 20 trabajos y 8 estaciones.

Xiaobo y Ohno (1997) minimizan el tiempo total de paro del transportador<sup>11</sup>. Usan B&B y un algoritmo de SA. En otro trabajo, los autores retoman el problema (Xiaobo y Ohno, 2000) y proponen una formulación matemática. Caracterizan diferentes propiedades que pueden ser útiles tales como complejidad computacional, cotas inferiores y superiores de la función objetivo y condiciones suficientes de optimalidad.

Un procedimiento paramétrico es propuesto por Aigbedo y Monden (1997) que considera simultáneamente cuatro objetivos: nivelado del uso de partes, nivelado en la carga, nivelar la tasa de cada producto y nivelar las sobrecargas en las líneas de sub-ensamble. Básicamente el procedimiento es de tipo *greedy*. En cada etapa de la secuencia el procedimiento selecciona un producto de un conjunto de ellos que cumplen con un límite prescrito de valores para cada uno de los criterios, o con el máximo de ellos.

Scholl et al. (1998), proponen una formulación matemática para el problema de secuenciación que minimizan la sobrecarga en estaciones cerradas. Los autores resuelven el problema con un procedimiento de búsqueda tabu (*Tabu Search Informed*) con estrategia de construcción de vocabulario basada en patrones (*Pattern Based Vocabulary Building*). Proponen una segunda formulación del modelo con el fin de encontrar grupos de patrones adecuados que puedan combinarse para completar una secuencia.

Dos métodos ponderados que consideran múltiples objetivos de secuenciación son comparados en Ding et al., (2006). Consideran como objetivo la minimización de la suma ponderada de las deferencias porcentuales de los mejores valores solución de los respectivos objetivos. A lo que los autores han denominado como solución *jointly balance*. Uno de los métodos (*scaling*) intenta normalizar todos los términos de los diferentes objetivos, mientras que el otro (*goal*) aplica los pesos en un proceso de dos pasos. Según la experiencia computacional, el método *goal* obtiene mejores soluciones. Aplican procedimientos de mejora a los resultados de ambos métodos. Finalmente, comparan el método *goal* con un procedimiento paramétrico combinando 4 objetivos.

### ***Procedimientos de dos fases***

Zeramardini et al. (2000) consideran el consumo regular de componentes y la regularización de las cargas en las estaciones para evitar paros de línea. En la primera fase de su propuesta consideran sólo el consumo regular de partes, mientras que en la segunda se enfocan en el criterio de las cargas, estableciendo restricciones de espacios en las secuencias (*Spacing Constraints*).

Korkmazel y Meral (2001) consideran dos objetivos: equilibrar la carga en las estaciones, y regularizar el consumo de componentes. Se analizan enfoques ya conocidos para el nivelado del consumo de partes minimizando la suma de las desviaciones. Los enfoques que funcionan mejor se extienden con el otro objetivo. Formulan el problema con los dos objetivos como un problema de asignación.

---

<sup>11</sup> Política normalmente aplicada en empresas japonesas para terminar los trabajos incompletos en las estaciones de una cadena de montaje.

Kotani et al. (2004) proponen una nueva formulación considerando el nuevo problema de minimización del tiempo total de paro de línea con trabajadores auxiliares y las restricciones para mantener el consumo constante de componentes. Se explica que los trabajadores auxiliares son diferentes a los trabajadores de refuerzo considerados por ejemplo en Scholl et al. (1998). Proponen un algoritmo de aproximación de dos fases que toma el *Goal-Chasing Method* en la primera fase y en la segunda aplican SA. Se obtienen soluciones sub-óptimas para problemas reales de la Toyota.

### ***Metaheurísticas multiobjetivo***

Smith et al. (1996), además del concepto de regularización en la aparición de productos, consideran también las cargas de trabajo. Los autores usan pesos para los productos con el fin de situar dos productos iguales cercanos en el tiempo. Se compara una heurística *greedy* con otra basada en el SA y en Redes Neuronales.

Hyun et al. (1998) proponen un modelo de formulación matemática que consideran tres objetivos simultáneos para la secuenciación en la línea: 1) Minimizar el trabajo utilitario total, 2) Regularizar el consumo de componentes, y 3) Minimizar el tiempo de cambio de partida. El método se basa en algoritmos genéticos (GA: *Genetic Algorithms*) lo cual permite considerar los tres objetivos de manera coordinada y brinda soluciones casi-pareto o pareto-óptimas. Proponen una nueva evaluación genética y un mecanismo de selección que denominan *Pareto Stratumniche Cubicle*.

Kim y Cho (2003) aplican SA para obtener soluciones cercanas al óptimo considerando tres objetivos de secuenciación: mantener una tasa constante de consumo de componentes en la línea, mantener una carga de producción regular y minimizar los tiempos de preparación interdependientes debido a los cambios de un producto al siguiente.

### ***Algunas variantes del problema***

Heike et al. (2001) presentan nuevas alternativas para el uso eficiente de RRHH/maquinaria en líneas de modelos mixtos. Presentan cuatro modelos para evaluar el tiempo de ciclo y diferentes alternativas de asignación de RRHH. Los autores aplican los modelos a un proceso de ensamble de aeroplanos.

Celano et al. (2004) minimizan el tiempo total de paro en líneas de tipo U. Resuelven el problema de secuenciación con GA y prueba diferentes políticas administrativas sobre los recursos humanos para el tipo de línea que se estudia.

En Sumichrast y Russell (1990); Sumichrast, Russell y Taylor (1992); y Sumichrast y Clayton (1996), se presentan comparativas entre diferentes procedimientos de secuenciación, para diferentes objetivos. Se usa simulación para comparar los procedimientos.

Bard et al. (1992) consideran que una línea de montaje puede caracterizarse por un grupo de parámetros: Disciplina de lanzamiento, movimiento de la línea, restricciones de la estación, disciplina de trabajo del operario y criterio de eficiencia. Un marco matemático común es presentado por los autores para 6 variaciones del problema de secuenciación.

Kim et al. (1996) consideran líneas híbridas (con estaciones abiertas y cerradas). El objetivo es minimizar la longitud total de la línea de montaje necesaria para que los operarios puedan completar las tareas sin problemas. Usan GA para obtener secuencias con valor objetivo cercano al óptimo, en un breve tiempo computacional. Sarker y Pan (1998) también consideran que la línea tiene estaciones abiertas y cerradas. Minimizan el coste total del tiempo utilitario y el tiempo ocioso en el cual se incurre debido a diferentes parámetros de la cadena (ciclo, longitud de la estación, punto de inicio). Desarrollan un modelo para estaciones abiertas y otro para cerradas para determinar los parámetros de la línea.

Chakravarty y Shtub (1986) al igual que Burns y Daganzo (1987), estudian la secuenciación de trabajos en líneas de montaje desde un punto de vista estocástico. Los primeros proponen un procedimiento para minimizar el coste operativo (costes laborables, trabajo en proceso, y costes de preparación). Los segundos proponen un método basado en principios analíticos simples para asistir en la evaluación de compensaciones entre diferentes objetivos de secuenciación (diferentes costes de preparación).

Kim et al. (2000) intentan resolver el equilibrado y la secuenciación simultáneamente con un método coevolutivo en líneas tipo U. Matanachai y Yano (2001) también proponen una forma de equilibrar el trabajo en las cadenas de montaje de forma que se tengan cargas de trabajo estables en las estaciones. Xiaobo et al. (2004) proponen una heurística basada en la cota inferior de la sobrecarga para realizar el equilibrado de la línea<sup>12</sup>.

Kouvelis y Karabati (1999) estudian el problema en líneas de producción con velocidad irregular sincrónica bajo una política de programación por ciclos, y presentan una formulación matemática de programación entera del problema. Los autores tienen en cuenta el inventario en proceso. Sarin et al. (2006) determinan secuencias cíclicas para líneas que consideran colchones de material en proceso entre las estaciones. Consideran dos objetivos. El principal de ellos es maximizar el *throughput*<sup>13</sup>, y el objetivo secundario es minimizar el trabajo en proceso. Compara diferentes mecanismos de control de flujo de materiales. Otro trabajo relacionado con secuencias cíclicas es el de Bautista et al. (2000). Los autores estudian la secuenciación de unidades en líneas de montaje con el fin de atenuar las variaciones en las tasas de consumo de recursos, y se centran en el problema PRV (*Product Rate Variation*) con función de discrepancia convexa.

Hindi y Ploszajski (1994) formulan y resuelven dos problemas de planeación en la manufactura de automóviles: seleccionar un subconjunto de autos de un banco de ordenes para el siguiente periodo de planeación sin violar las restricciones de los recursos, y secuenciar dicho conjunto de automóviles en la línea de ensamble de manera que no se exceda la capacidad con exceso de cargas. Bolat (2006) retoma el problema, enfocándose en la selección de trabajos de un banco de órdenes para ser producidos en una línea bien equilibrada con velocidad constante. Proponen una formulación binaria entera para el problema con el objetivo de minimizar el coste de las fechas de expedición. El modelo intenta mantener conjuntos mínimos de trabajos (MJS). Proponen un procedimiento óptimo y una heurística.

---

<sup>12</sup> Si el lector está interesado en temas de equilibrado de líneas, en el volumen 168 (3) del *European Journal of Operational Research* puede encontrar una edición especial titulada *Balancing assembly and transfer lines*

<sup>13</sup> El precursor de la Teoría de Restricciones, Ely Goldratt, en sus libros, traduce al castellano el término “throughput” como “facturación” ya que considera costes de ventas. Sin embargo, en el sentido que usa el término Sarin et al. (2006), se refiere a volumen de producción.

Un trabajo reciente e interesante es el de Bock et al. (2006). En él se propone un enfoque para el control adaptativo en tiempo real de líneas de montaje. Además de considerar el problema de secuenciación, los autores extienden el problema al considerar aspectos del equilibrado de la línea para reaccionar adecuadamente ante diferentes posibles alteraciones en el funcionamiento programado de la línea, como la ausencia de un trabajador o el fallo del equipo/maquinaria. En la Tabla 1.2 se han resumido los trabajos considerados más relevantes en esta tesis.

**Tabla 1.2** Tabla resumen del estado del arte

Autores	Algoritmo	Objetivo(s)
Wester y Kilbridge, 1964	Procedimiento <i>greedy</i>	Tiempo ocioso y congestión de trabajo
Thomopoulos, 1967	Procedimiento <i>greedy</i>	Ineficiencia, congestión, tiempo utilitario, tiempo ocioso
Macaskill, 1973	Procedimiento <i>greedy</i>	Trabajo utilitario, ineficiencia
Okamura y Yamashina, 1979	Heurístico	Riesgo de paro de línea
Yano y Bolat, 1989	Heurística prospectiva ( <i>look ahead</i> )	Sobrecarga
Yano y Rachamadugu, 1991	Formulación matemática MIP, <i>greedy</i>	Sobrecarga
Bolat y Yano, 1992a	2 algoritmos de reglas de espacios y 2 heurísticas <i>greedy</i>	Sobrecarga y ocio
Bolat, 1994	Heurística TRIM	Trabajo utilitario
Tsai, 1995	Heurística	Riesgo de paro de línea y trabajo utilitario
Guerre et al., 1995	Secuenciación dinámica	Fechas de entrega, balancear cargas y tiempos de espera
Smith et al., 1996	Heurística <i>greedy</i> y Redes Neuronales	Regularidad en la producción y trabajo utilitario
Bolat 1997a y 1997b	Algoritmo estocástico, SA, procedimiento óptimo y heurística.	Trabajo utilitario, trabajo incompleto
Xiaobo y Ohno, 1997	B&B y SA	Paro línea
Aigbedo y Monden, 1997	Procedimiento paramétrico <i>greedy</i>	Nivelado del uso de partes, de carga, de tasa de producción y de carga de sub-ensambles
Hyun et al., 1998	Formulación matemática y GA	Trabajo utilitario, regularidad en consumo de componentes y tiempo de cambio de partida
Scholl et al., 1998	<i>Informed Tabu Search, Pattern Based Vocabulary Building</i>	Sobrecarga
Zeramardini et al. 2000	Heurística <i>Bench-mark (load)</i> seguida por enfoque de restricciones de espacio ( <i>parts</i> )	Tasa de consumo de partes, paros de línea
Xiaobo y Ohno, 2000	Heurístico	Tiempo total de paro de línea
Korkmaz y Meral, 2001	Procedimiento de dos fases	Nivelado de carga y consumo de partes
Kim y Cho, 2003	SA	Consumo de partes, carga regular y tiempos de preparación
Kotani et al. 2004	Formulación. Algoritmo de aproximación de dos fases	Consumo de partes y paro de línea
Ding et al., 2006	<i>Scaling y Goal</i>	<i>Jointly balance-</i> multiobjetivo



---

## Modelo general

---

En este capítulo se presenta un modelo general del problema y se especifica las condiciones y supuestos contemplados así como el objetivo o criterio de secuenciación. Este trabajo está enfocado a proponer herramientas efectivas para la secuenciación de productos mixtos (MSP: *Mixed-model Sequencing Problem*). A grandes rasgos, el MSP consiste en encontrar una secuencia que mezcla unidades de diferentes productos para satisfacer la demanda de cada producto de manera óptima, según algún criterio preestablecido. El criterio de secuenciación considerado es la minimización de la sobrecarga<sup>14</sup>.

Existen dos enfoques principales al trabajar con este criterio. En el primero, la sobrecarga se traduce en trabajo utilitario<sup>15</sup> el cual debe ser minimizado. El trabajo utilitario es el trabajo que no han completado los trabajadores dentro de sus estaciones y que debe ser hecho por otros trabajadores que apoyan a los primeros. De esta manera todos los trabajos son completados dentro de las estaciones y no quedan trabajos pendientes. El segundo enfoque equivale al tiempo de paro de la línea<sup>16</sup> que también se pretende minimizar. En algunas empresas Japonesas no es usual que haya trabajadores de refuerzo o utilitarios, y los trabajos deben ser terminados dentro de las estaciones. Por ello, los trabajadores pueden detener la línea si necesitan más tiempo para terminar sus trabajos dentro de sus estaciones. En este trabajo se asume el primer enfoque.

Con el fin de delimitar el problema, se describen las características de esas líneas flexibles de productos mixtos (MMAL) y se discuten varias hipótesis y objetivos. Se da una formulación matemática para una versión específica del problema. Revisiones generales o diferentes versiones del problema pueden verse, por ejemplo, en Yano y Bolat (1989) o Bard et al. (1992).

### 2.1 Características operativas del problema y supuestos

Los sistemas productivos donde se presenta el problema de secuenciación, tienen distribuciones de planta orientadas al producto. Estos sistemas finalizan en una cadena de montaje. En este trabajo, para determinar la secuenciación de productos en la línea, se considera un nivel de componentes. Se asume que las necesidades de dichos componentes han sido determinadas de acuerdo a la estructura del producto y al programa de producción.

Las hipótesis de trabajo se enumeran a continuación:

---

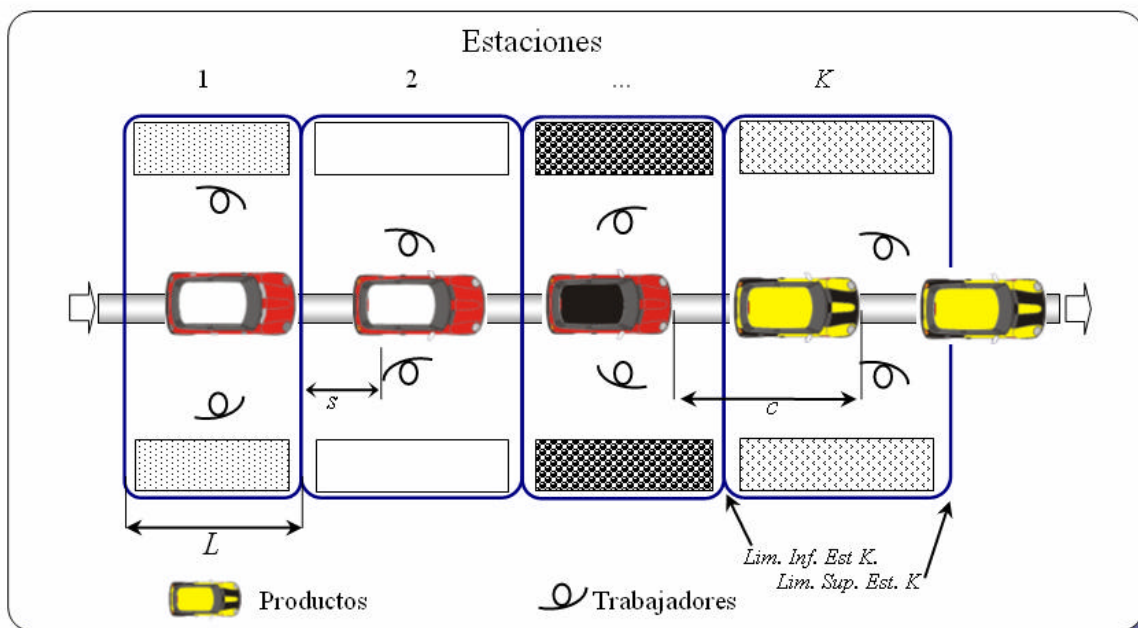
<sup>14</sup> *Work Overload*, Yano y Rachamadugu (1991) o Scholl et al. (1998).

<sup>15</sup> *Utility Work*, Bolat y Yano (1992a, 1992b), o Scholl et al. (1998).

<sup>16</sup> *Total Conveyor Stoppage Time*, criterio adoptado por Xiaobo y Ohno (1997, 2000), por ejemplo.

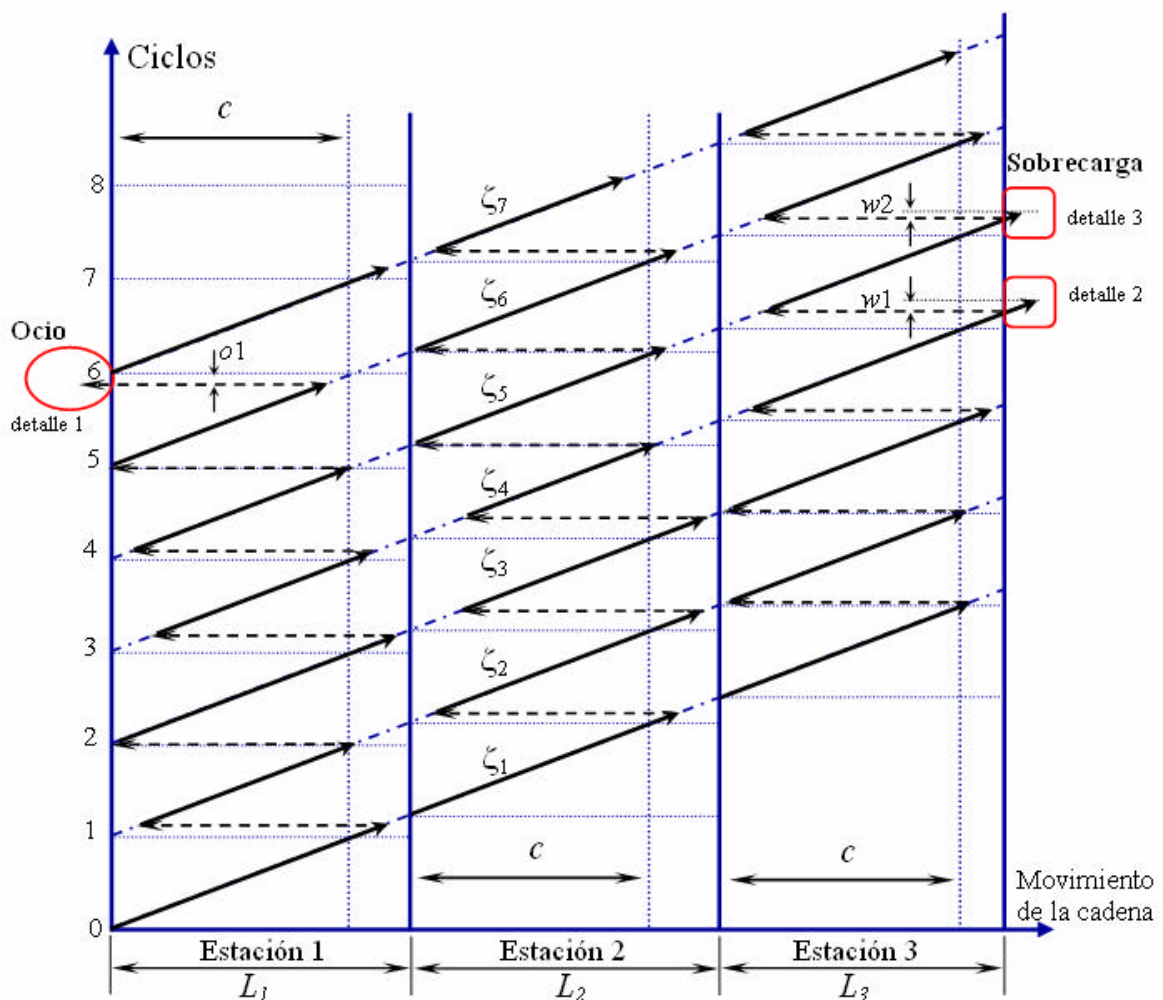
1. Las tareas necesarias para montar un producto se realizan en serie. Aunque se elaboran diferentes productos, todos siguen la misma ruta sobre la línea de montaje. Debido a la variedad en los productos, existe también gran diversidad en el consumo de componentes, recursos y servicios en las estaciones de la cadena. Bajo una filosofía de trabajo JIT, esto crea la necesidad de determinar secuencias de productos.
2. La cadena de montaje ha sido previamente balanceada o equilibrada, esto es, el trabajo necesario para fabricar todos los productos se ha dividido y ha sido repartido entre las estaciones de la línea. La distribución de la línea es fija, así como el número de estaciones y sus extensiones (Figura 2.1).
3. En cada periodo del plan, que puede variar desde un turno hasta una semana, debe satisfacerse la demanda de cada producto. Estas demandas esperadas han sido consideradas en el equilibrado de la línea. Todas las limitaciones de la cadena y las de los procesos ajenos a ella, han sido tratadas previamente y contempladas al elaborar el programa de producción.
4. La tasa de producción del sistema es determinada por la velocidad del transportador y por la frecuencia de lanzamientos. Se asume que ambos son constantes, por lo que los productos están equidistantes sobre el transportador. Al tiempo entre dos lanzamientos consecutivos se le denomina tiempo de ciclo  $c$ , y ha sido determinado durante el balanceo de la línea a partir del total de unidades a producir y del tiempo total disponible para ello. Se asume también que el tiempo medio que necesita una unidad para ser completada en la estación más cargada no supera  $c$ .
5. Existen cargas de trabajo  $p_{ik}$  (tiempos de procesamiento) para cada producto  $i$  ( $i = 1, \dots, I$ ) en cada estación  $k$  ( $k = 1, \dots, K$ ). Los tiempos de procesamiento son deterministas e incluyen los tiempos de preparación.

Figura 2.1 Esquema de una cadena de montaje



6. Las estaciones son cerradas, esto es, existe alguna restricción por la cual el trabajo sobre las unidades sólo puede realizarse dentro del espacio físico de la estación. La longitud de la ventana  $L$  es el tiempo máximo concedido para realizar el trabajo sobre un producto en la estación. Los tamaños de las ventanas pueden ser diferentes entre si, pero se fija la longitud de cada estación. No existen *stocks* intermedios de productos entre las estaciones. El tiempo que una pieza permanece dentro de una estación siempre es mayor o igual al tiempo de ciclo  $L_k = c$ .
7. Cada estación esta asistida por dos operadores que trabajan simultáneamente. Si el tiempo disponible para el trabajo regular por los trabajadores no es suficiente para completar una unidad, se agregan operarios de refuerzo a la línea para acelerar la conclusión de las operaciones. Se asume que hay un número suficiente de trabajadores de refuerzo.
8. Se asume que los operarios realizan su trabajo tan pronto como sea posible, y que el tiempo que requieren para desplazarse desde una unidad terminada hacia una nueva o hacia el inicio de la ventana, es insignificante porque la velocidad de desplazamiento del trabajador es mucho mayor a la del transportador. En cualquier instante, no puede haber más de una unidad completa de producto en cada estación.

Figura 2.2 Diagrama de desplazamiento de los trabajadores en las estaciones



Se incurre en sobrecarga ( $w$ ) cuando los operarios de una estación no pueden terminar el trabajo sobre un producto dentro de la estación (antes de que el producto salga de ella). Por otra parte, si los operarios alcanzan el límite inferior o inicio de la ventana antes de que la siguiente unidad entre a la estación se incurre en ocio<sup>17</sup> ( $o$ ), ya que el trabajador debe esperar a la entrada del nuevo automóvil en su estación para realizar su trabajo. Aunque un trabajador esté ocioso, no se desplaza a otra estación.

En la Figura 2.2 se puede apreciar el desplazamiento de los trabajadores en tres estaciones adjuntas. Se asume que el transportador se desplaza de izquierda a derecha. El eje de las ordenadas indica tiempo de ciclo. Las flechas indican el lapso de tiempo que requiere el operario para realizar su trabajo sobre la unidad. Las flechas punteadas representan el regreso del trabajador hacia el origen de la estación. Estas flechas son horizontales porque se asume que el desplazamiento de los operadores es instantáneo. Se usan dos tamaños de flechas para representar los tiempos de procesamiento, una menor al ciclo y otra mayor al ciclo. En la estación 1, al final del ciclo 6 el trabajador no puede ir más allá del límite inferior de su estación para alcanzar la unidad 7. El trabajador tiene que esperar  $o_1$  unidades de tiempo de ciclo, hasta que la unidad entre en la estación para iniciar su trabajo sobre ella (Detalle 1). El trabajador de la estación 2 no tiene tiempo improductivo ni problemas de sobrecarga. El trabajador de la estación 3 no puede terminar su trabajo sobre la cuarta y quinta unidades (Detalles 2 y 3, respectivamente) debido a que hay demasiadas unidades consecutivas que son ricas en contenido de trabajo. Se produce primero una sobrecarga o trabajo faltante de  $w_1$  unidades de tiempo de ciclo, y posteriormente se obtienen  $w_2$  unidades de tiempo.

## 2.2 Establecimiento del problema

A continuación se plantea el problema considerando las condiciones y suposiciones anteriores sobre el tipo de estaciones, ritmo de lanzamientos, velocidad del transportador, y tipo de línea. Alguna condición diferente a las descritas podría requerir un planteamiento de formulación matemática diferente al que se describe a continuación.

Como se mencionó, se asume que la velocidad del transportador es constante ( $v$  *speed*). Gracias a ello, todas las distancias se pueden medir en unidades de tiempo.

Una solución factible para el problema MSP está dada por una secuencia de longitud  $T$ , que contiene las unidades demandadas  $n_i$  de cada producto  $i$ .  $T = \sum_{i=1}^I n_i$ .

La posición inicial del trabajador en una estación sobre la  $t$ -ésima unidad, describe la distancia del trabajador al origen de la estación en unidades de tiempo. Se asume que cualquier tiempo de proceso  $p_{ik}$  no excede el tiempo equivalente de la longitud de las estaciones  $L_k$ . La solución óptima será aquella secuencia que minimice la suma de la sobrecarga de todas las  $K$  estaciones de la línea.

---

<sup>17</sup> Se asimila ocio ( $o$ ) a tiempo muerto y a tiempo improductivo.

### 2.2.1 Formulación matemática

Las secuencias se describen con las variables binarias  $x_{it}$  ( $i = 1, \dots, I$ ;  $t = 1, \dots, T$ ) que adoptan el valor de 1 si en la  $t$ -ésima posición de la secuencia se asigna un producto tipo  $i$ , y 0 si es de otro modo. Sea  $s_{kt}$  las variables que indica la posición del trabajador respecto al origen (inicio) de la estación  $k$  al iniciar una tarea en la posición  $t$  de la secuencia, y  $w_{kt}$  las variables que indican la sobrecarga de la estación  $k$  con la  $t$ -ésima unidad de la secuencia.

El problema puede modelarse de la siguiente manera:

$$\text{Minimizar } Z(x, w, s) = \sum_{k=1}^K \sum_{t=1}^T w_{kt} \quad (2.1)$$

$$\text{Sujeto a } \sum_{i=1}^I x_{it} = 1 \quad t = 1, \dots, T \quad (2.2)$$

$$\sum_{t=1}^T x_{it} = n_i \quad i = 1, \dots, I \quad (2.3)$$

$$s_{k1} \geq \sum_{i=1}^I p_{ik} \cdot x_{i1} - c \quad k = 1, \dots, K \quad (2.4)$$

$$s_{kt} \geq s_{k,t-1} + \sum_{i=1}^I p_{ik} \cdot x_{it} - c - w_{kt} \quad k = 1, \dots, K; t = 2, \dots, T \quad (2.5)$$

$$w_{k1} \geq \sum_{i=1}^I p_{ik} \cdot x_{i1} - L_k \quad k = 1, \dots, K \quad (2.6)$$

$$w_{kt} \geq w_{k,t-1} + \sum_{i=1}^I p_{ik} \cdot x_{it} - L_k \quad k = 1, \dots, K; t = 2, \dots, T \quad (2.7)$$

$$s_{k1} = 0, s_{kt} \geq 0, w_{kt} \geq 0, x_{it} \in \{1, 0\} \quad k = 1, \dots, K; t = 2, \dots, T \quad (2.8)$$

La función (2.1) minimiza la sobrecarga total. El grupo de restricciones (2.2) aseguran que en cada posición de la secuencia se ha asignado una unidad de producto. Las restricciones (2.3), garantizan que se satisfaga la demanda de cada producto. Las restricciones (2.4 y 2.5) garantizan que el trabajo sobre una unidad de producto no se iniciará antes de que se haya terminado el trabajo sobre la unidad anterior con trabajo regular o utilitario. Las restricciones (2.6 y 2.7) aseguran que los trabajos se realicen dentro de las estaciones. Las restricciones de no-negatividad e integridad se representan en (2.8).

Otra manera de representar el grupo de restricciones (2.4-2.7) es respectivamente, la siguiente:

$$\sum_{i=1}^I p_{ik} \cdot x_{i1} - s_{k1} \leq c \quad k = 1, \dots, K \quad (2.9)$$

$$\sum_{i=1}^I p_{ik} \cdot x_{it} + s_{k,t-1} - s_{kt} - w_{kt} \leq c \quad k = 1, \dots, K; t = 2, \dots, T \quad (2.10)$$

$$\sum_{i=1}^I p_{ik} \cdot x_{i1} - w_{k1} \leq L_k \quad k = 1, \dots, K \quad (2.11)$$

$$\sum_{i=1}^I p_{ik} \cdot x_{it} + s_{k,t-1} - w_{kt} \leq L_k \quad k = 1, \dots, K; t = 2, \dots, T \quad (2.12)$$

$$s_{k1} = 0, s_{kt} \geq 0, w_{kt} \geq 0, x_{it} \in \{1, 0\}$$

La variable  $s_{kt}$  también puede considerarse como el tiempo perdido respecto al instante en que idealmente debería de comenzar la siguiente tarea en la posición  $t$  en la estación  $k$  (2.13).

$$s_{kt} = \max \left\{ s_{k,t-1} + \sum_{i=1}^I p_{ik} \cdot x_{it} - c - w_{kt}, 0 \right\} \quad (2.13)$$

$s_{k1} = 0$ , y se tienen en total  $K \cdot T$  variables del tipo  $s_{kt}$ . El valor de las variables  $s_{kt}$  depende de una expresión no lineal. Para salvar este obstáculo se hace una transformación (lo cual es posible gracias a que la función objetivo es convexa, y a que se intentará minimizar):

$$s_{k1} \geq \sum_{i=1}^I p_{ik} \cdot x_{i1} - c \quad s_{k1} = 0 \quad (2.14)$$

$$s_{kt} \geq s_{k,t-1} + \sum_{i=1}^I p_{ik} \cdot x_{it} - c \quad s_{kt} \geq 0 \quad (2.15)$$

Como se mencionó,  $w_{kt}$  indica la sobrecarga de la estación  $k$  en la posición  $t$ . Hay  $K \cdot T$  de estas variables y un total de  $I \cdot T$  variables de tipo  $x_{it}$  (2.16).

$$w_{kt} = \max \left\{ s_{k,t-1} + \sum_{i=1}^I p_{ik} \cdot x_{it} - L_k, 0 \right\} \quad (2.16)$$

Debido a que las variables  $w_{kt}$  también dependen de una expresión no lineal, se hace la siguiente transformación:

$$w_{k1} \geq \sum_{i=1}^I p_{ik} \cdot x_{i1} - L_k \quad w_{k1} = 0 \quad (2.17)$$

$$w_{kt} \geq s_{k,t-1} + \sum_{i=1}^I p_{ik} \cdot x_{it} - L_k \quad w_{kt} \geq 0 \quad (2.18)$$

El objetivo es minimizar la sobrecarga total. Es posible representar la sobrecarga en términos de costes o penalizaciones por estación  $k$ , incluyendo dicho peso o coste  $cs_k$  en la función objetivo. Así, se obtiene la formulación de programación entera mixta 2.19-2.25:

$$\text{Minimizar } Z(x, w, s) = \sum_{k=1}^K \sum_{t=1}^T cs_k \cdot w_{kt} \quad (2.19)$$

$$\text{Sujeto a } \sum_{i=1}^I x_{it} = 1 \quad t = 1, \dots, T \quad (2.20)$$

$$\sum_{t=1}^T x_{it} = n_i \quad i = 1, \dots, I \quad (2.21)$$

$$\sum_{i=1}^I p_{ik} \cdot x_{i1} - s_{k1} \leq c \quad k = 1, \dots, K \quad (2.22)$$

$$\sum_{i=1}^I p_{ik} \cdot x_{it} + s_{k,t-1} - s_{kt} - w_{kt} \leq c \quad k = 1, \dots, K; t = 2, \dots, T \quad (2.23)$$

$$\sum_{i=1}^I p_{ik} \cdot x_{it} - w_{kt} \leq L_k \quad k = 1, \dots, K \quad (2.24)$$

$$\sum_{i=1}^I p_{ik} \cdot x_{it} + s_{k,t-1} - w_{kt} \leq L_k \quad k = 1, \dots, K; t = 2, \dots, T \quad (2.25)$$

$$s_{k1} = 0, s_{kt} \geq 0, w_{kt} \geq 0, x_{it} \in \{1, 0\}$$

Se minimiza la función (2.19) permite penalizar o convertir en coste la sobrecarga producida en las estaciones de la línea. El modelo cuenta con  $I \cdot T$  variables binarias,  $2 \cdot I \cdot T$  variables reales y  $T + I + (2 \cdot K \cdot T)$  restricciones. Las dimensiones del problema crecen en función del número de estaciones  $K$ , del número de productos  $I$  y del total de unidades a

secuenciar  $T$ . Como mencionan Yano y Rachamadugu (1991), en una gran compañía de automóviles pueden producirse aproximadamente 1000 unidades diarias y considerar críticas 12 estaciones de la línea respecto a la carga de trabajo. Si una instancia tiene 20 estaciones, 20 productos y demanda total de 500 unidades, el modelo tendrá 20000 variables reales, 10000 variables binarias y 20520 restricciones.

En la bibliografía pueden encontrarse diferentes formulaciones para variantes del problema. Bard et al. (1992), por ejemplo, establecen formulaciones del problema con diferentes parámetros de la línea (tipos de estaciones, ritmo de lanzamientos, actuación de operarios) para dos objetivos: 1) minimizar la longitud de la línea, y 2) minimizar el tiempo de producción (*throughput time* o *lead time*).

El trabajo en esta tesis nace de la propuesta de Yano y Rachamadugu (1991), donde las unidades del modelo matemático son unidades medidas en tiempo, y usa el tiempo de ciclo  $c$  como la unidad. Así que los tiempos de proceso de cada producto en cada estación están expresados en tiempos de ciclo. De hecho, los autores proponen un modelo donde la función objetivo maximiza el trabajo completado, lo cual equivale a minimizar la sobrecarga o trabajo perdido (Bolat, 1997).

En Scholl et al. (1998) y en Scholl (1999), el problema se modela haciendo una simplificación del tiempo de procesamiento necesario para procesar el producto  $i$  en la estación  $k$  (2.26).

$$\mathbf{r}_{kt} = \sum_{i=1}^I p_{ik} \cdot x_{it} \quad (2.26)$$

Los autores formulan el problema con programación lineal entera mixta para describir el problema de secuenciación de productos<sup>18</sup>, minimizando la sobrecarga, así:

$$\text{Minimizar} \quad Z(x, w, s) = \sum_{k=1}^K \sum_{t=1}^T w_{kt} \quad (2.27)$$

$$\text{Sujeto a} \quad \sum_{i=1}^I x_{it} = 1 \quad t = 1, \dots, T \quad (2.28)$$

$$\sum_{t=1}^T x_{it} = n_i \quad i = 1, \dots, I \quad (2.29)$$

$$s_{k,t-1} \geq s_{kt} + \mathbf{r}_{kt} - w_{kt} - c \quad k = 1, \dots, K; t = 1, \dots, T \quad (2.30)$$

$$s_{kt} + \mathbf{r}_{kt} - w_{kt} \leq L_k \quad k = 1, \dots, K; t = 1, \dots, T \quad (2.31)$$

$$s_{kt} \geq 0, w_{kt} \geq 0, s_{k1} = 0 \quad k = 1, \dots, K; t = 1, \dots, T \quad (2.32)$$

$$x_{it} \in \{0, 1\} \quad k = 1, \dots, K; t = 1, \dots, T \quad (2.33)$$

La función objetivo (2.27), minimiza la sobrecarga de trabajo. Las restricciones (2.28) y (2.29) aseguran que sólo se asigna una unidad en cada posición de la secuencia. La restricción (2.30) garantiza que, en la estación  $k$ , un trabajo no puede iniciarse hasta que el trabajo anterior ha sido terminado usando trabajo regular o trabajo utilitario. El trabajo se restringe a ser realizado dentro de la ventana de la estación con la restricción (2.31). La no

<sup>18</sup> En sus diferentes trabajos, Scholl denomina a este problema MSP-W, Mixed-model Sequencing Problem.

## Capítulo 2

negatividad la aseguran los dos primeros grupos de restricciones (2.32), el tercer grupo establece que, al iniciar la secuencia, el trabajador se encuentra en el límite inferior de la estación. La función (2.34) traduce la sobrecarga en coste.

$$\text{Minimizar } Z(x, w, s) = \sum_{k=1}^K \sum_{t=1}^T cs_k \cdot w_{kt} \quad (2.34)$$

Aunque el problema parece un problema de asignación, su resolución no es sencilla debido a las restricciones adicionales sobre las posiciones de los operarios y las variables para medir la sobrecarga, las cuales son variables continuas si se asume al tiempo de ciclo como la unidad de tiempo (Yano y Rachamadugu, 1991). El modelo puede verse también como una generalización del TSP (*Traveling Salesman Problem*) con distancias interdependientes (Scholl et al., 1998; Gouveia y Voss, 1995). En definitiva, el problema es de tipo *NP-hard* (Tsai, 1995; Xiaobo y Ohno, 1997; Xiaobo y Ohno, 2000; Kotani et al., 2004).

En el anexo 1 se muestra la nomenclatura principal, índices y parámetros del problema tratado en esta tesis. El cuadro A.2 del apéndice 1 contiene algunas de las relaciones entre variables y parámetros. A lo largo de cada capítulo se describirá nueva nomenclatura.

---

## Procedimientos constructivos

---

Debido a la dificultad para resolver de forma exacta problemas combinatorios con dimensiones industriales, se han desarrollado algoritmos que producen soluciones factibles. Aunque estos procedimientos no garantizan la optimización se supone que ofrecen soluciones aceptables, en breve tiempo. A estos procedimientos se les ha llamado “heurísticos”. A cambio de que no siempre obtienen un óptimo, los métodos heurísticos ofrecen una solución “satisfactoria” en un tiempo de cálculo razonable.

Un tipo de métodos heurísticos (Díaz et al., 1996) son los denominados constructivos. Este tipo de procedimientos construyen una solución añadiendo paulatinamente componentes de la misma hasta que se obtiene una solución completa. Un método constructivo popular es el denominado *greedy*, el cual construye paso a paso la solución buscando la mejor aportación a la función objetivo en cada uno de ellos.

Una técnica matemática que ha sido explorada para resolver el problema de secuenciación es la programación dinámica. Esta técnica es útil cuando se requiere tomar una serie de decisiones interrelacionadas y proporciona un procedimiento sistemático para determinar la combinación de decisiones que maximiza la efectividad total.

Dado que el problema tratado en esta tesis es difícil de resolver de forma óptima, se ha decidido explorar métodos heurísticos. En este capítulo se exponen distintos métodos constructivos que se han propuesto en la literatura para el problema de secuenciación que consideran productos con opción y sin opción. También se describe un procedimiento de programación dinámica de la literatura para el problema con múltiples estaciones.

En la primera sección se describe el procedimiento de Yano y Rachamadugu (1991) para una estación. Este procedimiento es el punto de partida de esta tesis, junto con los trabajos de Bolat y Yano (1992a, 1992b). Se describe la manera en que Yano y Rachamadugu (1991) miden la sobrecarga, así como el modelo de programación matemática que proponen para maximizar el trabajo completado. Se describen tres procedimientos que consideran una estación y productos con y sin opción especial. Se propone el procedimiento equivalente que se ha denominado Updown1. Se propone un segundo procedimiento denominado *updown* que combina tres procedimientos de acuerdo al valor de cotas de sobrecarga y ocio. Se describe el procedimiento propuesto por Yano y Rachamadugu para múltiples estaciones. La sección 3.3 contiene la experiencia computacional que prueba el desempeño de los cinco procedimientos para una estación y se experimenta también con el procedimiento para múltiples estaciones.

### 3.1 Procedimientos para una estación

Considérese una sola estación y dos tipos de producto distinguibles por sus tiempos de procesamiento: los productos de tipo básico (que notaremos B) tienen un tiempo de procesamiento inferior al tiempo de ciclo  $c$ , y los productos especiales (o con opción, que

notaremos A) tienen carga superior a  $c$ . En las líneas de montaje de automóviles un automóvil puede o no recibir un componente opcional como el techo solar, respaldo ajustable eléctrico, ordenador de abordo, o el aire acondicionado. Debido a que las estaciones son cerradas el trabajo sobre los productos debe realizarse dentro de la estación. El trabajador dispone de un tiempo máxima ( $L$ ) para realizar su trabajo. Por tanto, la longitud de la ventana y el tiempo concedido para realizar los trabajos sobre los productos en la estación son proporcionales. La longitud de la ventana también puede entenderse como el máximo número de unidades que puede haber dentro de la estación en un instante dado.

### 3.1.1 Medición de la sobrecarga

Yano y Rachamadugu (1991) proponen una forma general de medir el trabajo perdido considerando una sola estación, una plantilla fija y la posibilidad de que haya más de un operario en la estación. Los autores miden la sobrecarga en unidades de tiempo. La unidad de tiempo corresponde al tiempo de ciclo de fabricación. Sean:

- $p_t$  tiempo de procesamiento del producto en la posición  $t$  de la secuencia,
- $s_t$  instante de inicio de la tarea en la posición  $t$  de la secuencia,
- $f_t$  instante de finalización de la tarea en la posición  $t$ , y
- $w_t$  sobrecarga debida a la tarea en la posición  $t$ .

Si el trabajo sobre la unidad en la posición  $t$  no ha sido completado en el instante que la unidad abandona la estación, el valor de  $f_t$  toma el valor de dicho instante. Dada una secuencia, y  $s_1 = 0$ , la sobrecarga  $w_t$  se puede expresar de la siguiente manera:

$$s_t = \max((t-1) \cdot c, f_{t-1}) \quad (3.1)$$

$$f_t = \min(s_t + p_t, (t-1) \cdot c + L) \quad (3.2)$$

$$w_t = [p_t + s_t - ((t-1) \cdot c + L)]^+ \quad \text{donde } [x]^+ = \max(0, x) \quad (3.3)$$

La ecuación (3.1) indica el instante de inicio del trabajo en la etapa  $t$  de la secuencia. Este instante equivale al momento que el producto de la etapa  $t$ -ésima posición entra en la estación, siempre y cuando el trabajo sobre el producto anterior ya haya sido terminado. De lo contrario, dicho instante será igual al momento que el trabajo sobre el producto en la etapa  $t-1$  haya sido terminado. Se asume que el tiempo de desplazamiento del trabajador de un producto al siguiente es despreciable. La ecuación (3.2) asegura que el instante de parada del trabajo sobre un producto es el mínimo entre el instante de finalización de la tarea y el instante en que el producto deja la estación. La ecuación (3.3) representa el trabajo incompleto o faltante sobre un producto cuando éste sale de la estación. La sobrecarga total ocasionada por una secuencia de tareas dada es  $Z = \sum_{t=1}^T w_t$ .

Dado que minimizar la sobrecarga en una estación es equivalente a maximizar el trabajo total completado (Yano y Rachamadugu, 1991), los autores proponen una formulación de programación lineal entera mixta para una sola estación. Sean:  $x_{it}$  la variable binaria que vale 1 si la tarea  $i$  se realiza en la posición  $t$  de la secuencia (y vale 0 en caso contrario),  $u_t$  la cantidad de tiempo asignado para realizar el trabajo sobre el producto en la posición  $t$ , y  $T$  el total de productos a secuenciar ( $t = 1, \dots, T$ ). Como unidad de tiempo se asume el

tiempo de ciclo  $c$  y se asume también que al inicio de la secuencia, la posición de los trabajadores es sobre el origen de la estación ( $s_1 = 0$ ).

$$\text{Maximizar } Z(x, u) = \sum_{t=1}^T u_t \quad (3.4)$$

$$\text{Sujeto a } \sum_i x_{it} = 1 \quad i = 1, \dots, I \quad (3.5)$$

$$\sum_i x_{it} = 1 \quad t = 1, \dots, T \quad (3.6)$$

$$u_t \leq \sum_i p_i x_{it} \quad i = 1, \dots, I \quad (3.7)$$

$$s_t \geq t - 1 \quad t = 1, \dots, T \quad (3.8)$$

$$s_t \geq s_{t-1} + u_{t-1} \quad t = 1, \dots, T \quad (3.9)$$

$$s_t + u_t \leq t - 1 + L \quad t = 1, \dots, T \quad (3.10)$$

$$u_t \geq 0, s_t \geq 0, x_{it} \in \{0,1\} \quad (3.11)$$

La función objetivo (3.4) maximiza el trabajo completado. Las restricciones (3.5) y (3.6) aseguran que cada trabajo sea asignado y que en cada etapa de la secuencia sea asignado un producto. (3.7) indican que el tiempo de procesamiento asignado en la etapa  $t$  será como máximo el tiempo de procesamiento del producto asignado en dicha etapa. (3.8) y (3.9) aseguran que un trabajo inicia sólo cuando el producto está dentro de la estación y el trabajo anterior ha sido terminado. Las restricciones (3.10) obligan a terminar un trabajo cuando el producto abandona la estación. Las restricciones de no-negatividad e integridad se representan con (3.11).

### 3.1.2 Procedimientos para una estación y productos con opciones

En esta sección se describen algunos procedimientos tomados de la bibliografía. Se describe también una nueva propuesta para este tipo de problema. Se considera necesario describir aquellos tomados de la bibliografía ya que se realizó una experiencia computacional para comparar todos los procedimientos.

Debido a que la velocidad del transportador es constante, la longitud de la ventana y el tiempo concedido son proporcionales. Puede añadirse también un coste por incurrir en sobrecarga. Este coste puede ser proporcional al número de trabajadores, a la energía que se consume en la estación o a los materiales que se utilizan. Sean:

- $p_a$  el tiempo de procesamiento de una unidad especial (tipo A),
- $p_b$  el tiempo de procesamiento de una unidad básica (tipo B),
- $n_a$  número de unidades a procesar del tipo A, y
- $n_b$  numero de unidades a procesar del tipo B.

Dado que  $c$  es la unidad de tiempo,  $p_b < c < p_a$ . La longitud de la ventana ( $L$ ) también está expresada en ciclos,  $p_a < L$ .

***Procedimientos de Yano y Rachamadugu***

El método consiste en establecer esquemas (sub-secuencias) formados por  $m_a$  unidades especiales y  $m_b$  unidades básicas. Estos esquemas se repiten mientras existan unidades para completar un esquema más. Cuando ya no es posible completar un esquema más, el resto de los productos se acomodan de forma que se minimiza la sobrecarga que se pueda ocasionar. Dado que  $p_a > c$ , el número máximo de unidades especiales que se pueden secuenciar de forma consecutiva sin incurrir en sobrecarga está limitado a una cantidad  $X \leq (L - c) / (p_a - c)$ .

Después de secuenciar  $X$  unidades especiales, se asignan unidades básicas para lograr la *regeneración* del sistema. Es decir, mediante la asignación de unidades básicas, se desplaza al trabajador hacia el origen de la estación, donde estaba al iniciar la sub-secuencia.

De esta forma un esquema regenerativo está compuesto por  $m_a$  unidades A y  $m_b$  unidades B, cuya determinación es:

$$m_a p_a + m_b p_b = m_a + m_b \quad (m_a \leq X, m_b \text{ enteros}) \quad (3.12)$$

Nótese que en esta sección se ha tomado el ciclo como unidad de tiempo y por tanto tenemos  $p_b < 1 < p_a$ . Dicho valor del tiempo de ciclo se asume en el ejemplo usado a lo largo del capítulo. La máxima utilización (sin sobrecarga) se alcanza al resolver el siguiente problema de programación entera:

$$\text{Maximizar } (p_a m_a + p_b m_b) / (m_a + m_b) \quad (3.13)$$

$$\begin{aligned} \text{Sujeto a} \quad & m_a \leq X \\ & p_a m_a + p_b m_b \leq m_a + m_b \\ & m_a \geq 0, m_b \geq 0, \text{ enteros} \end{aligned}$$

Para un valor dado de  $m_a$ , la máxima utilización se alcanza cuando se usa el valor más pequeño de  $m_b$  que satisface la desigualdad siguiente:

$$p_a m_a + p_b m_b \leq m_a + m_b \quad (3.14)$$

En aquellos casos en que (3.12) no puede ser satisfecha con un valor entero de  $m_b$ , Bolat (1988) obtiene cotas al hacer  $m_a = X$  y obtener  $m_b$  a partir de (3.13). Así, asumiendo que se cumple la condición (3.12), Yano y Rachamadugu (1991) prueban una proposición que brinda la secuencia óptima para una secuencia en una estación. En tales condiciones, la secuencia se obtiene como se muestra en el Algoritmo 3.1.

Ejemplo: Supóngase que se tienen 5 unidades especiales (A) y 10 unidades básicas (B). Los tiempos de procesamiento son 1.15 y 0.95 unidades de tiempo de ciclo para los trabajos A y B respectivamente. El tiempo de ciclo equivale a la unidad. La longitud de la estación, también expresada en tiempos de ciclo es de 1.24. El número de unidades A y B

que conformarán un patrón se obtienen con el modelo (3.13). El mayor número de unidades especiales que pueden secuenciarse consecutivamente sin ocasionar sobrecarga es  $X = 1$ . Así, el número de unidades que conforman un patrón es  $m_a = 1$  y  $m_b = 3$ , lo cual satisface (3.12) y (3.14). Por lo cual, la solución que brinda el procedimiento es óptima.

**Algoritmo 3.1** YR para una estación y productos con opción

Dada una demanda inicial de unidades con opción especial  $n_a$  y unidades básicas  $n_b$ , y un esquema repetitivo conformado de  $m_a$  unidades especiales y  $m_b$  unidades básicas, la secuencia óptima tendrá:

1.  $nc = \min (\lfloor n_a / m_a \rfloor , \lfloor n_b / m_b \rfloor )$  ciclos de  $m_a$  unidades tipo A seguidas por  $m_b$  unidades tipo B, donde  $\lfloor x \rfloor$  es el entero inferior a  $x$ , seguido por
2.  $x_a = \min (n_a - (nc \cdot m_a), m_a)$  unidades especiales, seguidas por
3.  $x_b = n_b - (nc \cdot m_b)$  unidades básicas, seguidas por
4.  $[n_a - (nc + 1) \cdot m_a]^+$  unidades especiales si procede.

Siguiendo el Algoritmo 3.1 y asumiendo que la posición inicial del trabajador es en el instante cero, se obtiene la siguiente secuencia conformada por  $nc = 3$  patrones “ABBB” (ver Tabla 3.1).

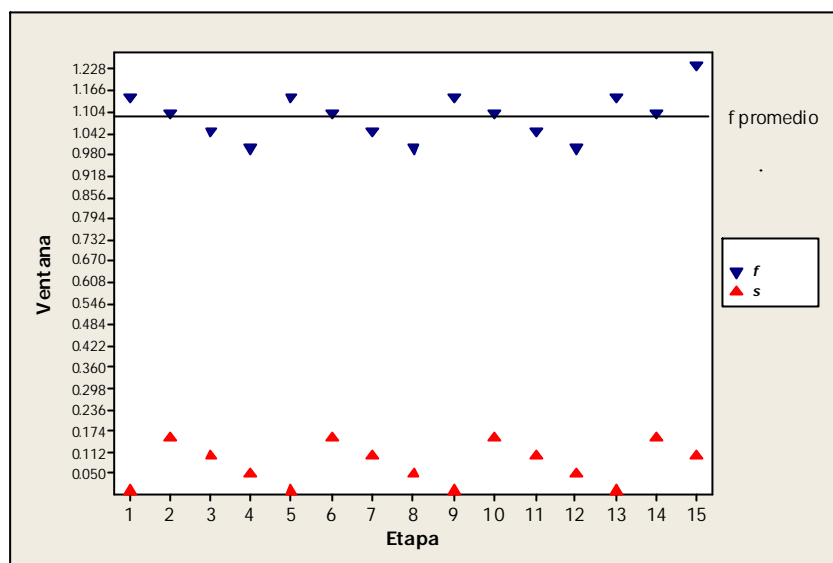
**Tabla 3.1** Secuencia obtenida con YR

Etapa $t$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Secuencia	A	B	B	B	A	B	B	B	A	B	B	B	A	B	A
$f_i$	1.15	2.10	3.05	4.00	5.15	6.10	7.05	8.00	9.15	10.10	11.05	12.00	13.15	14.10	15.24
$w_i$															0.01

Después de repetir el patrón tres veces,  $x_a = 1$  unidades tipo A se asignan a partir de la posición 13, a partir de la posición 14 se asignan  $x_b = 1$  unidades básicas y finalmente, se asigna la unidad especial pendiente. Sólo en la posición 15, el trabajador no puede terminar su tarea antes de que la unidad salga de la estación, obteniendo una sobrecarga de 0.01 unidades de tiempo de ciclo. Dado que el tiempo mínimo requerido para procesar todas las unidades demandadas es  $(n_a \cdot p_a) + (n_b \cdot p_b) = 15.25$ , el trabajo perdido obtenido con esta secuencia es el menor posible.

La Figura 3.1 muestra los cambios de dirección del trabajador dentro de la estación, es decir, cuando inicia ( $s$ ) y cuando termina ( $f$ ) un trabajo. En esta figura y en las siguientes de este tipo, el inicio y la terminación de los trabajos ( $s$  y  $f$ ) están representados en tiempo respecto al origen de la estación. El instante promedio de finalización de las tareas es de 1.09. Excepto en la última etapa, la posición del trabajador en el momento de terminar sus tareas sobre los productos tiende a mantenerse alejada del límite superior de la estación. Como se ve en la Figura 3.1, cada cuatro unidades el sistema regenera la posición del trabajador en la estación.

**Figura 3.1** Puntos de cambio de dirección del trabajador para YR



**Procedimientos de Bolat y Yano**

Bolat y Yano (1992a) proponen un grupo de procedimientos para determinar secuencias que minimizan el trabajo utilitario en una estación con dos tipos de trabajos. En uno de los procedimientos busca una sub-secuencia perfecta, en la que no hay sobrecarga, ni ocio y además el sistema regenera. Otro procedimiento es *greedy* y evita producir sobrecarga mientras ello sea posible. Similarmente, un tercero evita incurrir en tiempo improductivo mientras ello sea posible. Según las características de la instancia, los algoritmos pueden obtener una solución óptima o cercana al óptimo. De ellos se toman los dos procedimientos *greedy* que serán distinguidos como BY1 y BY2. No se consideran los procedimientos basados en reglas de espacios porque se asumen que existe un ciclo de regeneración “perfecto”.

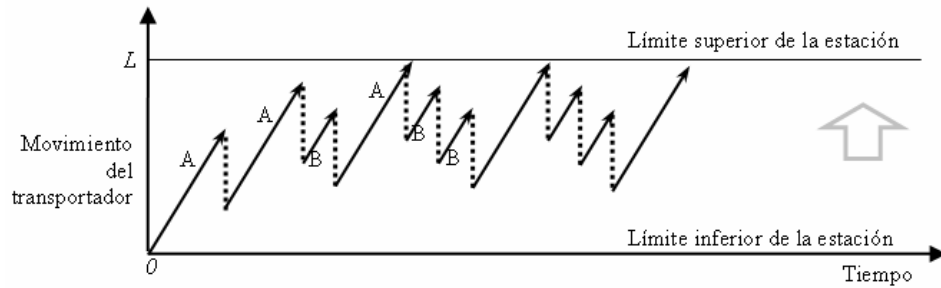
**Algoritmo 3.2** BY1 para una estación y productos con opción

Dada una posición de la secuencia y el instante en el que la unidad anterior deja libre la estación:

1. Si hay tiempo suficiente para completar un trabajo con opción antes de que salga de la estación, asignarlo, asumiendo que aun hay unidades pendientes. Si ya no hay tales unidades pendientes, asignar un trabajo sin opción.
2. Si no hay tiempo suficiente para completar un trabajo con opción antes de que salga de la estación, asignar un trabajo básico, suponiendo que hay unidades básicas pendientes, de lo contrario, asignar un trabajo con opción.

En cada etapa, el procedimiento BY1 asigna un producto especial si al hacerlo no se produce sobrecarga. Esto propicia el desplazamiento del trabajador hacia el límite superior de la estación, tal como se ejemplifica en la Figura 3.2. BY1 procede de la forma indicada en el Algoritmo 3.2.

Figura 3.2 Tendencia de la posición del trabajador para una secuencia de BY1



Este procedimiento da prioridad a la asignación de unidades ricas en carga de trabajo a menos que se ocasione sobrecarga. Un inconveniente de este procedimiento es que puede ocasionar tiempo inhábil innecesario al asignar un trabajo básico, cuando podría asignarse un trabajo con opción que ocasionaría sobrecarga inevitable.

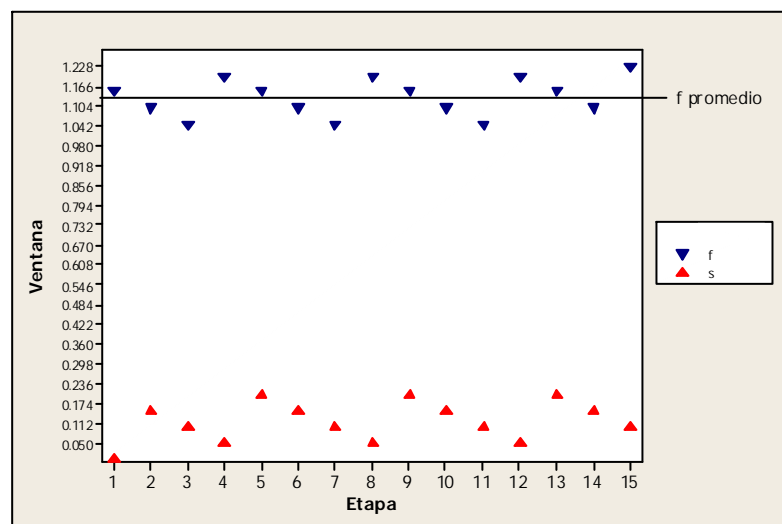
Así, dada una instancia  $(n_a, n_b, p_a, p_b, L)$ , mientras exista producción pendiente de las unidades especiales, se asigna una de ellas si no produce sobrecarga, de lo contrario, se asigna una unidad básica si hay unidades por producir.

Tabla 3.2 Secuencia obtenida con BY1

Etapa $t$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Secuencia	A	B	B	A	B	B	B	A	B	B	B	A	B	B	A
$f_t$	1.15	2.10	3.05	4.20	5.15	6.10	7.05	8.20	9.15	10.10	11.05	12.20	13.15	14.10	15.24
$w_t$															0.01

Para el ejemplo anterior, con el procedimiento BY1 se obtiene la solución de la Tabla 3.2. Esta ocasión, el instante de terminación promedio es de 1.13 unidades de tiempo respecto al origen de la estación.

Figura 3.3 Instantes de inicio y finalización con BY1



El algoritmo BY2 es una heurística que ofrece buenos resultados para instancias de cargas intermedias (Bolat y Yano, 1992a). Mientras existen unidades básicas pendientes de producir se asignan éstas si no se incurre en tiempo muerto para el trabajador, de lo contrario se asigna una unidad especial siempre que exista producción pendiente. El Algoritmo 3.3 describe el procedimiento BY2.

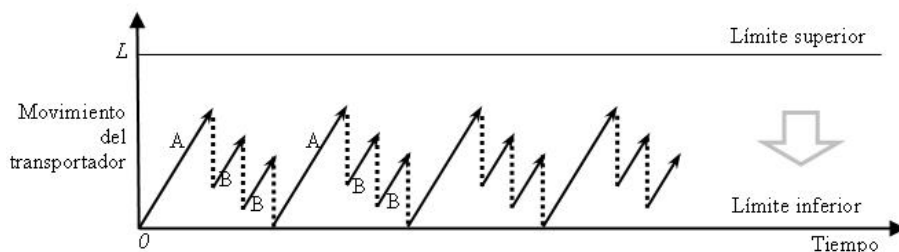
**Algoritmo 3.3** BY2 para una estación y productos con opción

Dada una posición de la secuencia y el instante en el que la unidad anterior deja libre la estación:

1. Si es posible asignar un trabajo con opción evitando sobrecarga, asignarlo si aun hay unidades pendientes.
2. Si es posible asignar una unidad básica evitando tiempo ocioso, asignar un trabajo básico, suponiendo que hay unidades pendientes. Si tal unidad no está disponible o si de asignar un trabajo básico se obtiene tiempo improductivo, asignar una unidad especial.

Contrariamente a BY1, una secuencia obtenida con BY2 propicia el desplazamiento del trabajador hacia el límite inferior de la estación, tal como se representa en la Figura 3.4.

**Figura 3.4** Tendencia de la posición del trabajador para una secuencia de BY2



La Tabla 3.3 muestra la secuencia que se obtiene para el ejemplo de este capítulo al aplicar el procedimiento BY2. El instante de terminación promedio con BY2 es de 1.092 unidades de tiempo respecto al origen de la estación.

**Tabla 3.3** Secuencia obtenida con BY2

Etapas $t$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Secuencia	A	B	B	B	A	B	B	B	A	B	B	B	A	B	A
$f_t$	1.15	2.10	3.05	4.00	5.15	6.10	7.05	8.00	9.15	10.10	11.05	12.00	13.15	14.10	15.24
$w_t$															0.01

**Procedimiento pseudo-regenerativo updown**

La propuesta *updown* (Ud1) mantiene el concepto de la regeneración en la secuencia (Yano y Rachamadugu, 1991; Bolat y Yano, 1992a, 1992b). Este procedimiento *greedy* procura la construcción de pseudo-ciclos, evitando la prioridad para secuenciar determinado tipo de producto y procurando con ello la regularidad en la aparición de productos en la secuencia. Dado que para obtener sub-secuencias o patrones perfectos es necesario que los parámetros de las instancias cumplan ciertas condiciones técnicas, se intenta generalizar al construir pseudo-ciclos, adaptándose así a las condiciones de las instancias.

El procedimiento constructivo Ud1 propicia que el trabajador se desplace de manera cíclica desde el inicio de la estación hasta el final, para después retornar al origen de la ventana, en ciclos hacia arriba y hacia abajo (updown).

Aunque el procedimiento busca pseudo-ciclos, no repite un único patrón constantemente, y los productos pueden secuenciarse teniendo en cuenta la producción pendiente (usando pesos si es necesario) para procurar la regularidad en la producción (Monden, 1983; Miltenburg, 1989; Bautista et al., 1996). Si durante la selección de un producto para ser secuenciado, tienen más prioridad los productos con alto contenido de trabajo, aquellos con bajo contenido serán retrasados hasta que no haya más unidades ricas en carga, y se incurra en tiempo de ocio innecesario, el cual a la vez induce sobrecarga.

Con  $d_a$  y  $d_b$  se denota la demanda pendiente de asignación en la etapa  $t$  de la secuencia, para los productos A y B, respectivamente. Ud1 procede como se indica en el Algoritmo 3.4.

**Algoritmo 3.4 Ud1**


---

```

0. Iniciar  $t \leftarrow 1, s \leftarrow 0$ .
while ( producción pendiente de A y B ) do
    1.  $aux \leftarrow \text{trae}$ .
    while ( no se incurra en sobrecarga ) do
        2.  $secuencia(t) \leftarrow A, aux \leftarrow \text{false}$ .
        3. Actualizar  $d_a$  y  $t$ .
    end while
    while ( no se incurra en ocio ) do
        4.  $secuencia(t) \leftarrow B, aux \leftarrow \text{false}$ .
        5. Actualizar  $d_b$  y  $t$ .
    end while
    if (  $aux$  ) then
        6. Asignar en  $t$  el producto que produce menor magnitud (carga/ocio).
        7. Actualizar  $t$  y producción pendiente.
    end if
end do

```

---

Si la preferencia en la asignación la tienen las unidades pobres en carga de trabajo, las tareas con carga de trabajo mayor al tiempo de ciclo se retrasarán hasta causar más sobrecarga de la necesaria (Yano y Rachamadugu, 1991).

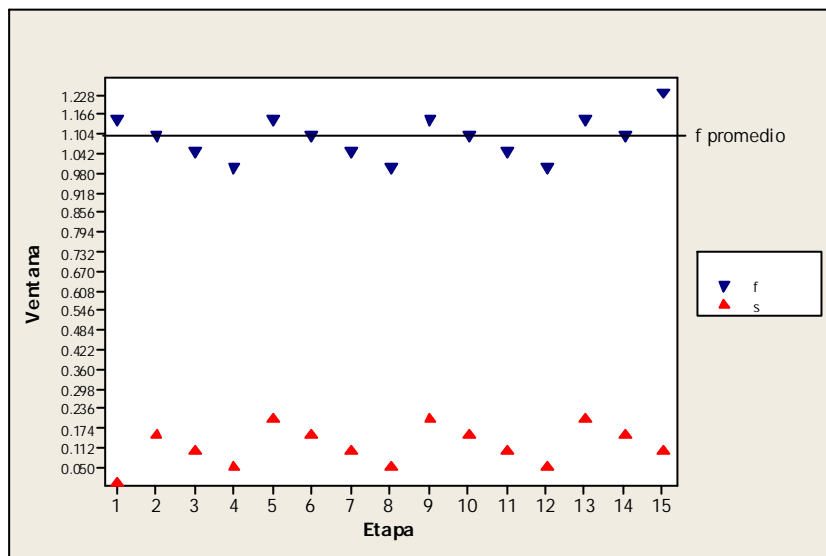
El procedimiento Ud1 también tiene en cuenta el ocio, el cual juega un papel importante en el desempeño de los sistemas con estaciones cerradas (Sarker y Pan, 2001). Dicho procedimiento evita incurrir en sobrecarga y en ocio. Puede suceder que, dada una posición de la secuencia, asignar un producto especial produzca sobrecarga y asignar un producto básico produzca ocio; en tal caso, se opta por asignar aquel producto que ocasiona un menor impacto. También puede considerarse la ponderación de las magnitudes.

**Tabla 3.4** Secuencia obtenida con Ud1

Etapa $t$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Secuencia	A	B	B	B	A	B	B	B	A	B	B	B	A	B	A
$f_i$	1.15	2.10	3.05	4.00	5.15	6.10	7.05	8.00	9.15	10.10	11.05	12.00	13.15	14.10	<b>15.24</b>
$w_i$															0.01

La Tabla 3.4 contiene la solución obtenida con el procedimiento Ud1 para el ejemplo anterior [5, 10, 1.15, 0.95, 1.24], que también es óptima. La posición final promedio del operador es 1.09 (ver Figura 3.5).

**Figura 3.5** Instantes de inicio y finalización con Ud1



**Cotas de sobrecarga y ocio**

El tiempo necesario para producir la demanda de todos los productos puede calcularse de manera sencilla. Si el tiempo máximo asignado para fabricar los productos es menor que el tiempo necesario de producción, se producirá sobrecarga inevitable. Por otra parte, si el tiempo necesario para fabricar la demanda es menor que el tiempo mínimo concedido existirá tiempo de ocio inevitable. Si el tiempo de fabricación está entre el tiempo mínimo y el tiempo máximo concedidos, podría suponerse que no debería haber sobrecarga o

tiempo de ocio. Entonces se puede tener una cota inferior del trabajo perdido  $lbw$  o del tiempo improductivo  $lbo$  iguales a cero.

Para el caso de una sola estación y dado que en esta sección  $c$  es la unidad de tiempo las cotas de sobrecarga y ocio se obtienen como indican 3.15 y 3.16 respectivamente.

$$lbw = [(n_a \cdot p_a + n_b \cdot p_b - (c \cdot (T - 1) + L)) \cdot cs]^+ \quad (3.15)$$

$$lbo = [((c \cdot T) - (n_a \cdot p_a + n_b \cdot p_b)) \cdot cs]^+ \quad (3.16)$$

En (3.15) y (3.16),  $cs$  es un valor opcional para indicar el coste de la sobrecarga u ocio en la estación.

Conociendo a priori los valores de  $lbw$  y  $lbo$ , se propone usar una u otra de las tres heurísticas *greedy* que se han descrito anteriormente (BY1, BY2 o Ud1). Puede inducirse que si BY1 es un algoritmo que evita producir sobrecarga en todo momento, puede funcionar bien para instancias con mucho tiempo improductivo. Es decir, si se sabe a priori que el sistema tiene poca carga o que es posible construir una secuencia sin sobrecarga, debería tener mejor desempeño un algoritmo que evita producir sobrecarga. Para YR2 también se intuye que si el procedimiento evita producir tiempo de ocio, debería producir buenos resultados para sistemas con mucha carga. Dado que Ud1 evita en todo momento incurrir tanto en sobrecarga como en ocio, debería tener un buen desempeño para sistemas con carga y ocio bajos.

Así, los valores de  $lbw$  y  $lbo$  proporcionan una regla para elegir entre los algoritmos BY1, BY2 o Ud1.

- (i) Si  $lbw > 0$ , aplicar BY2.
- (ii) Si  $lbo > 0$ , aplicar BY1.
- (iii) Si  $lbw = lbo = 0$ , aplicar Ud1.

En el Experimento 1 de este capítulo se aplican la regla de decisión (i)-(iii). Los resultados se muestran en la experiencia bajo el nombre de Ud2.

### 3.2 Procedimiento para múltiples estaciones

En esta sección se describe el procedimiento propuesto por Yano y Rachamadugu (1991) considerando estaciones múltiples. Este algoritmo se usa en secciones posteriores como punto de partida para proponer extensiones para casos donde un producto puede tener más de una opción, o se consideran más de dos productos. El procedimiento para múltiples estaciones se basa en procedimientos para una estación, que son útiles para definir heurísticas basados en reglas de prioridad. La secuencia se construye progresivamente de tal manera que para cada periodo  $t$  de la secuencia, se selecciona de entre los productos con producción pendiente, aquel con el mejor valor de predicción de sobrecarga.

Sea  $K$  el número de estaciones de la línea,  $L_k$  la longitud de la estación  $k$ ,  $p_{ik}$  el tiempo de proceso que requiere el producto  $i$  en la estación  $k$ ,  $w_i(k,t)$  la sobrecarga ocasionada en la etapa  $t$  en la estación  $k$  por asignar el producto  $i$  en dicha posición de la secuencia,  $wp_i(k)$  es la predicción de sobrecarga que se obtendrá en la estación  $k$  a partir del periodo siguiente al actual, asumiendo que se ha asignado un producto tipo  $i$  en la etapa actual.

$sc_i(t)$  denota el valor del posible trabajo perdido ocasionado a partir de la etapa  $t$  por asignar en ella un producto de tipo  $i$ ,  $i \in \{A, B\}$ . El predictor de sobrecarga  $wp_i(k)$  se obtiene mediante procedimientos para una sola estación.

Llegados a una posición  $t$  de la secuencia, supóngase que restan por secuenciar  $d_a$  y  $d_b$  unidades A y B, respectivamente, se procede entonces de acuerdo al Algoritmo 3.5.

**Algoritmo 3.5** YR para múltiples estaciones

---

```

0. Iniciar  $t \leftarrow 0$ ,  $s_k \leftarrow 0$ , y  $secuencia(t) = \emptyset$ .
for ( $t = 1$  to  $T$ )
    1. calcular  $sc_A(t)$  y  $sc_B(t)$ .
    if ( $sc_B(t) < sc_A(t)$ )
        2.  $secuencia(t) \leftarrow B$ , actualizar  $d_b$  y  $s_k$ .
    else
        3.  $secuencia(t) \leftarrow A$ , actualizar  $d_a$  y  $s_k$ .
    end if
end for

```

---

Donde el valor del posible trabajo perdido se determina sumando la sobrecarga ocasionada en la posición actual y la predicción de sobrecarga, ambos como resultado de asignar el producto tipo  $i$  en la etapa actual (3.17).

$$sc_i(t) = \sum_{k=1}^K [wp_i(k) + w_i(k,t)] \cdot cs_k \quad \text{para } i = A, B \quad (3.17)$$

El predictor de sobrecarga  $wp_i(k)$  se determina asumiendo que se ha asignado una unidad de tipo  $i$  en la etapa  $t$ . Entonces restarán momentáneamente  $n_i-1$  unidades tipo  $i$ . La sobrecarga en la etapa actual se determina como se indica en (3.18).

$$w_i(k,t) = [s_{kt} + p_{ik} - ((t-1) \cdot c) - L_k]^+ \quad \text{para } i = A, B \quad (3.18)$$

En esta tesis, las variantes incorporadas al procedimiento anterior, se basan en la sustitución del cálculo de las cotas del trabajo perdido por las unidades pendientes de secuenciar ( $wp_A(k)$  y  $wp_B(k)$ ), por el valor de la predicción del trabajo perdido (más el tiempo ocioso) que resulta al aplicar los procedimientos para una estación, a las unidades pendientes de secuenciar.

El algoritmo YR calcula cotas del trabajo perdido por las unidades pendientes de secuenciar ( $wp_A(k)$  y  $wp_B(k)$ ). La variante que se proponen en esta tesis consiste en sustituir estas cotas por las predicciones que resultan al aplicar los procedimientos para una estación a las unidades pendientes de secuenciar.

### 3.3 Experiencia computacional

Se ha realizado una experiencia computacional compuesta por dos experimentos. El primero se compone de ejemplares con una sola estación. El segundo tiene en cuenta varias estaciones. En ambos casos se consideran productos básicos (tipo B) y con opción especial (tipo A). El tiempo de ciclo  $c = 1$ . El objetivo de ambos experimentos es comparar los resultados que ofrece cada procedimiento.

#### 3.3.1 Experimento 1, una estación

Para probar los procedimientos para una sola estación se han generado instancias de acuerdo a las características siguientes:

- a) Tamaño de ventana  $L$ : 1.10, 1.20, 1.30, 1.40 y 1.50 (ciclos).
- b) Número total de unidades a secuenciar  $T$ : 36, 110 y 216.
- c) Ratios del mix A/B: 3/2, 5/4, 1/1, 3/4, y 1/2.
- d) Tiempo de procesamiento: distribuciones uniformes  $p_a [1, L]$  y  $p_b [2 - L, 1]$ .

Para cada terna (a, b, c) se han generado 10 instancias con tiempos de procesamiento según d), lo que supone un total de 750 instancias que han sido resueltos con los procedimientos descritos en la sección 3.1.2.

La Tabla 3.5 contiene los resultados del experimento con: la desviación porcentual media ( $d_l^T$ ) del tiempo de operaciones respecto a la cota inferior  $lbw$ , el porcentaje de óptimos confirmados ( $O_l^T$ ) y el porcentaje de ocasiones que el algoritmo produjo la mejor solución ( $G_l^T$ ), por procedimiento  $l$  y por número total de unidades a secuenciar  $T$ , así como los resultados globales de estas magnitudes por procedimiento. En el Anexo 2 se describe la obtención del índice de desviación porcentual media  $d$ .

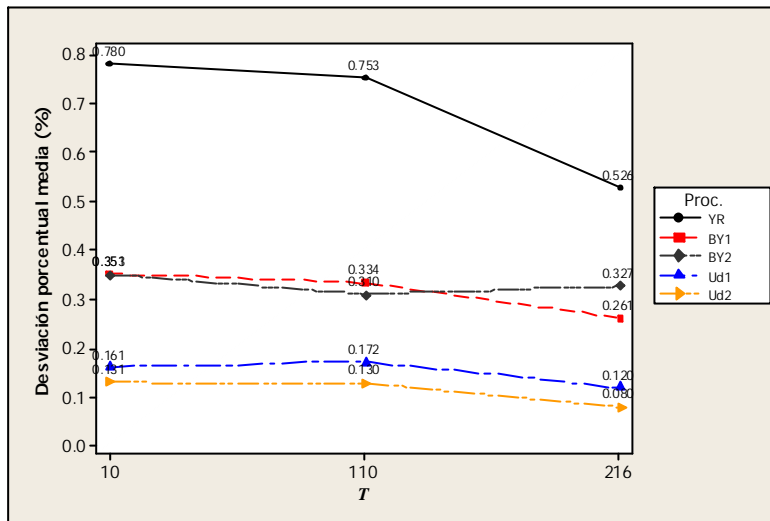
**Tabla 3.5** Índices de calidad para el experimento 1 para total de unidades

Proced. $l$	$d_l^T$ (%)				$O_l^T$ (%)				$G_l^T$ (%)			
	$T$			$d_l$	$T$			$O_l$	$T$			$G_l$
	36	110	216		36	110	216		36	110	216	
YR	0.783	0.753	0.526	0.688	76.4	81.6	86.8	81.6	80.0	82.8	88.4	83.7
BY1	0.353	0.334	0.261	0.316	74.4	71.6	81.2	75.7	78.0	74.4	82.4	78.3
BY2	0.351	0.310	0.327	0.329	72.4	74.8	69.6	72.3	76.8	76.8	71.6	75.1
Ud1	0.161	0.172	0.120	0.151	72.8	73.6	76.8	74.4	80.0	79.2	80.4	79.9
Ud2	0.131	0.130	0.080	0.114	90.0	93.6	96.0	93.2	95.2	96.0	97.6	96.3
	0.356	0.340	0.263		77.2	79.0	82.1		82.0	81.8	84.1	

En general, las desviaciones porcentuales medias por procedimiento respecto a la  $lbw$  son inferiores al 1%. A pesar de que el procedimiento YR tiene la mayor desviación porcentual media, el resultado es aceptable para los tres totales de unidades considerados, de hecho, dicho índice se reduce conforme el total de unidades aumenta. Los procedimientos *greedy* BY1 y BY2 ofrecen las siguientes mejores desviaciones con valores de 0.31% y 0.32% respectivamente. BY1 muestra también una reducción en el valor  $d$  conforme aumenta el total de unidades a secuenciar. El procedimiento Ud1 ofrece una desviación porcentual

media muy baja (0.151%), la cual mejora si se combina con BY1 y BY2 (Ud2). Al promediar los valores  $d$  de todos los procedimientos por total de unidades se encuentra una tendencia a la disminución de dicho valor conforme el valor de  $T$  aumenta (ver Figura 3.6), excepto en el procedimiento BY2 que sube ligeramente el valor de  $d$  al cambiar de  $T = 110$  a  $T = 216$ .

**Figura 3.6** Desviación porcentual media para total de unidades



Aunque YR tiene la desviación porcentual media más alta, el procedimiento alcanza el 81.6% de óptimos confirmados con valor de  $lbw$  y el 83.7% de las veces proporciona la mejor solución encontrada. Sólo el procedimiento Ud2, que combina YR1, YR2 y Ud1, obtiene mejores valores para el índice  $O$  (93.2%) y el índice  $G$  (96.3%). El porcentaje de óptimos confirmados por total de unidades ( $O^T$ ) también tiene pequeñas variaciones favorables cuando el valor de  $T$  aumenta.

Al promediar los resultados de las 750 instancias con cada procedimiento se obtienen los valores de la Tabla 3.6. Los promedios más pequeños de sobrecarga ( $w$ ) y ocio ( $\phi$ ) se obtienen con los procedimientos Ud1 y Ud2.

**Tabla 3.6** Promedios de sobrecarga y ocio para el experimento 1

Cota	YR		BY1		BY2		Ud1		Ud2			
	$w$	$\phi$	$w$	$\phi$	$w$	$\phi$	$w$	$\phi$	$w$	$\phi$		
promedio	2.945		3.698	4.416	3.313	4.045	3.334	4.046	3.116	3.899	3.066	3.785

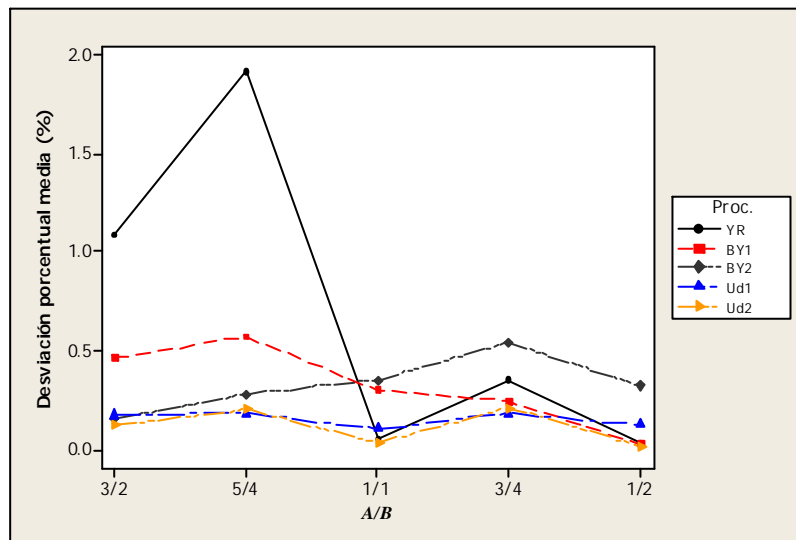
Como puede apreciarse en la Tabla 3.7, los valores de desviación porcentual media por mix de producción tienen más variación que al considerar el total de unidades a secuenciar. Algunos de los valores superan el 1% de variación. Específicamente, para el procedimiento YR con mixes de producción 3/2 y 5/4. Estos mixes de producción implican que las instancias contienen una proporción 3/2 y 5/4 unidades de productos especiales sobre unidades básicas, por lo tanto las instancias tienen exceso de carga considerable y ligera, respectivamente.

**Tabla 3.7** Índices de calidad para el experimento 1 para ratios de mix

Proced <i>l</i>	$d_l^{A/B}$ (%)					$d_l$	$O_l^{A/B}$ (%)					$O_l$	$G_l^{A/B}$ (%)					$G_l$
	A/B						A/B						A/B					
	3/2	5/4	1/1	3/4	1/2		3/2	5/4	1/1	3/4	1/2		3/2	5/4	1/1	3/4	1/2	
YR	1.09	1.92	0.05	0.35	0.03	0.69	86.7	75.3	79.3	78.7	88.0	81.6	86.7	75.3	86.7	81.3	88.7	83.7
BY1	0.46	0.57	0.30	0.24	0.02	0.32	65.3	58.0	80.0	82.0	93.3	75.7	68.0	64.0	80.0	86.0	93.3	78.3
BY1	0.15	0.28	0.35	0.54	0.32	0.33	90.0	80.7	63.3	65.3	62.0	72.3	91.3	82.7	70.7	68.0	62.7	75.1
Ud1	0.17	0.18	0.10	0.18	0.12	0.15	80.7	66.0	68.0	77.3	80.0	74.4	84.7	75.3	74.0	85.3	80.0	79.9
Ud2	0.12	0.20	0.03	0.21	0.01	0.11	95.3	88.7	90.7	92.0	98.7	93.2	96.7	94.0	98.0	93.3	99.3	96.3
	0.40	0.63	0.17	0.30	0.10		83.6	73.7	76.3	79.1	84.4		85.5	78.3	81.9	82.8	84.8	

En la figura (Figura 3.7) se puede percibir una tendencia a la disminución en el valor de  $d^{A/B}$  cuando las instancias son menos sobrecargadas. Esto es, que las instancias con mix 3/2 tienen la mayor carga, mientras que en las instancias con mix 1/2 el número de unidades con tiempo de proceso mayor al ciclo es sólo la mitad de unidades con tiempo de proceso menor al ciclo. El procedimiento BY2 es el único que no tiene este comportamiento, ya que, como se explicó anteriormente, en cada etapa  $t$  del procedimiento da más prioridad a la secuenciación de unidades con poca carga de trabajo, dejando las unidades ricas en trabajo hasta el final de la secuencia, con lo cual se obtiene más sobrecarga de la necesaria a pesar de que reduce el tiempo de ocio.

**Figura 3.7** Desviación porcentual media para mixes de producción



El mayor porcentaje de óptimos alcanzados se logran para los mixes 3/2 y 1/2, es decir, para las instancias mas sobrecargadas (83.6%) y menos sobrecargadas (84.4%), respectivamente. El porcentaje de óptimos confirmados por ratios de mix para el procedimiento BY1 ( $O^{A/B}$ ) tiende a aumentar conforme las instancias son menos sobrecargadas. Contrariamente, para el procedimiento BY2, el mismo índice tiende a disminuir, pues como se dijo, el primer procedimiento funciona mejor para instancias descargadas mientras que el segundo tiene mejor desempeño para instancias sobrecargadas.

**Tabla 3.8** Índices de calidad para el experimento 1 para longitudes de estación

Proced.	$d_l^L$ (%)					$d_l$	$O_l^L$ (%)					$O_l$	$G_l^L$ (%)					$G_l$
	$L$						$L$						$L$					
	1.10	1.20	1.30	1.40	1.50		1.10	1.20	1.30	1.40	1.50		1.10	1.20	1.30	1.40	1.50	
YR	0.19	0.43	0.81	0.65	1.36	0.69	85.2	86.0	78.7	82.0	76.0	81.6	86.6	87.3	80.0	88.0	76.7	83.7
BY1	0.10	0.13	0.24	0.55	0.56	0.32	79.2	78.0	78.7	72.7	70.0	75.7	83.2	81.3	80.0	74.7	72.0	78.3
BY1	0.10	0.30	0.27	0.50	0.48	0.33	73.8	74.0	74.7	65.3	73.3	72.3	75.2	75.3	76.0	72.0	76.7	75.1
Ud1	0.06	0.09	0.14	0.21	0.26	0.15	75.8	76.7	76.0	71.3	72.0	74.4	81.2	80.7	80.7	78.7	78.0	79.9
Ud2	0.03	0.08	0.10	0.20	0.16	0.11	94.6	94.7	94.0	90.7	92.0	93.2	98.0	96.7	96.0	96.0	94.7	96.3
	0.10	0.21	0.31	0.42	0.56		81.7	81.9	80.4	76.4	76.7		84.8	84.3	82.5	81.9	79.6	

Excepto para el procedimiento YR con tamaño de ventana de 1.5 unidades de tiempo de ciclo, el resto de los valores de desviación porcentual media por longitud de ventana ( $d^L$ ) son inferiores al 1%. Se nota menos efectividad en los procedimientos cuando el tamaño de la ventana aumenta. El mejor desempeño en  $d^L$  para todos los tamaños de ventana  $L$  lo muestra el procedimiento Ud2.

En definitiva, los mejores valores de los índices usados para evaluar la calidad de las soluciones en el experimento 1, se obtienen con el procedimiento Ud2 que combina Ud1, BY1 y BY2.

### 3.3.2 Experimento 2, múltiples estaciones

En el experimento 2, se prueban los procedimientos descritos en éste capítulo considerando estaciones múltiples. Las características de las instancias generadas para este experimento son las siguientes:

- Tamaño de ventana  $L$ : 1.10, 1.20, 1.30, 1.40 y 1.50 (ciclos).
- Número total de unidades a secuenciar  $T$ : 36, 110 y 216.
- Ratios del mix A/B: 3/2, 5/4, 1/1, 3/4, y 1/2.
- Número de estaciones  $K$ : 10, 50 y 100.
- Tiempo de procesamiento: distribuciones uniformes  $p_a [1, L]$  y  $p_b [2 - L, 1]$ .

Para cada cuaterna (a, b, c, d) se han generado 10 instancias con tiempos de procesamiento según e), lo que supone un total de 2250 instancias. Las instancias se resolvieron con el procedimiento para varias estaciones descrito en la sección 3.2. Como predictores de la sobrecarga se usan los procedimientos para una estación de la sección 3.1.2. Los costes por incurrir en sobrecarga se han tomado como iguales en todas las estaciones.  $\forall k, cs_k = 1$ .

La Tabla 3.9 se muestra la desviación porcentual media ( $d_l^K$ ), el porcentaje de óptimos confirmados con  $lbw$  ( $O_l^K$ ) y el porcentaje de ocasiones como algoritmo ganador ( $G_l^K$ ), por procedimiento auxiliar  $l$  y número de estaciones  $K$ , así como los resultados globales de estas magnitudes por procedimiento auxiliar.

**Tabla 3.9** Índices de calidad para el experimento 2 para número de estaciones

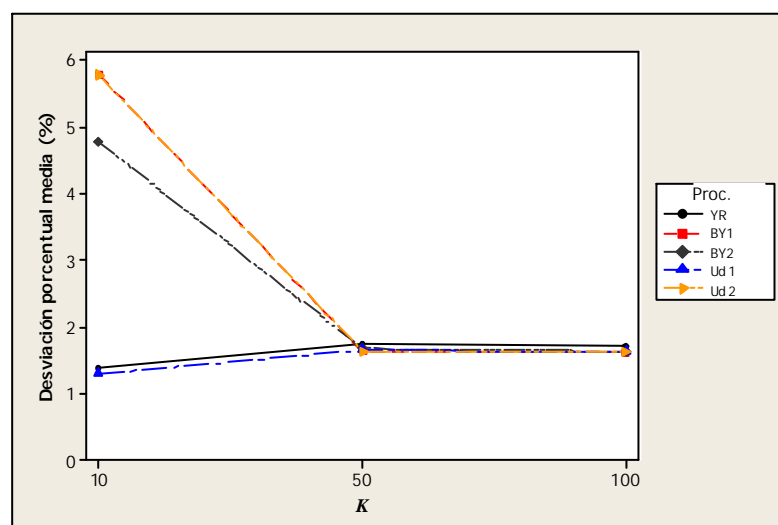
Proced. <i>L</i>	$d_l^K$ (%)				$O_l^K$ (%)				$G_l^K$ (%)			
	<i>K</i>			$d_l$	<i>K</i>			$O_l$	<i>K</i>			$G_l$
	10	50	100		10	50	100		10	50	100	
YR	1.39	1.74	1.72	1.62	9.07	0.00	0.00	3.02	44.13	21.20	21.07	28.80
BY1	5.79	1.64	1.62	3.02	0.00	0.00	0.00	0.00	0.00	20.00	21.20	13.73
BY2	4.79	1.68	1.65	3.04	0.00	0.00	0.00	0.00	0.00	33.07	34.40	22.49
Ud1	1.31	1.66	1.65	1.54	9.33	0.00	0.00	3.11	69.73	17.73	15.07	34.18
Ud2	5.79	1.64	1.63	3.02	0.00	0.00	0.00	0.00	0.00	17.73	15.07	10.93
	4.01	1.67	1.65		3.68	0.00	0.00		22.77	21.95	21.36	

La desviación porcentual media (global)  $\delta$  es de 2.44% respecto a  $lbw$ . Considerando que esta vez se consideran 10, 50 y 100 estaciones, este valor de la desviación porcentual media es aceptable. La desviación porcentual media para las instancias con 10 estaciones es la más grande, sin embargo, sólo en éste tipo de instancias se confirma el óptimo con el valor de  $lbw$  (en el 3.68 % de las instancias).

Como se puede apreciar en la Tabla 3.9 y en la Figura 3.8, para  $L = 10$ , los mejores valores  $\delta$  se obtienen con los procedimientos Ud1 (1.31%) y YR (1.39%), mientras que el resto de ellos al menos duplican este valor. Cuando  $L$  toma el valor de 50 y 100 estaciones, los valores de  $\delta$  presentan menor variación entre procedimientos. De hecho, presentan una desviación porcentual media inferior al caso en que  $L = 10$ . A pesar de ello, para las instancias con 50 y 100 estaciones no se confirma ningún valor solución con el valor  $lbw$ .

En cuanto al porcentaje de veces que un algoritmo obtuvo el mejor resultado ( $G$ ), para el caso de 10 estaciones, YR y Ud1 siempre obtienen el mejor resultado, el primero en el 44.13 % y el segundo el 69.73% de las veces. Para las instancias con 50 y 100 estaciones, los procedimientos YR y BY1 obtienen un mayor número de veces el mejor resultado de sobrecarga. BY2 es el único procedimiento que nunca obtiene una mejor solución.

**Figura 3.8** Desviación porcentual media para número de estaciones



La Tabla 3.10 se muestra la desviación porcentual media ( $d_l^T$ ), el porcentaje de óptimos confirmados con  $lbw$  ( $O_l^T$ ) y el porcentaje de ocasiones como algoritmo ganador ( $G_l^T$ ), por procedimiento auxiliar  $l$  y por total de unidades  $T$ , así como los resultados globales de estas magnitudes por procedimiento auxiliar.

**Tabla 3.10** Índices de calidad para el experimento 2 para total de unidades

Proced.	$d_l^T$ (%)					$O_l^T$ (%)				$G_l^T$ (%)			
	$T$			$d_l$	$T$			$O_l$	$T$			$G_l$	
	$L$	36	110		210	36	110		210	36	110		210
YR	1.54	1.63	1.68	1.62	3.60	2.80	2.67	3.02	29.60	28.40	28.40	28.80	
BY1	2.90	3.03	3.13	3.02	0.00	0.00	0.00	0.00	16.67	12.67	11.87	13.73	
BY2	2.92	3.05	3.15	3.04	0.00	0.00	0.00	0.00	23.47	22.80	21.20	22.49	
Ud1	1.48	1.56	1.58	1.54	4.67	2.00	2.67	3.11	37.73	33.47	31.33	34.18	
Ud2	2.91	3.03	3.12	3.02	0.00	0.00	0.00	0.00	6.13	11.60	15.07	10.93	
	2.35	2.46	2.53		1.65	0.96	1.07		22.72	21.79	21.57		

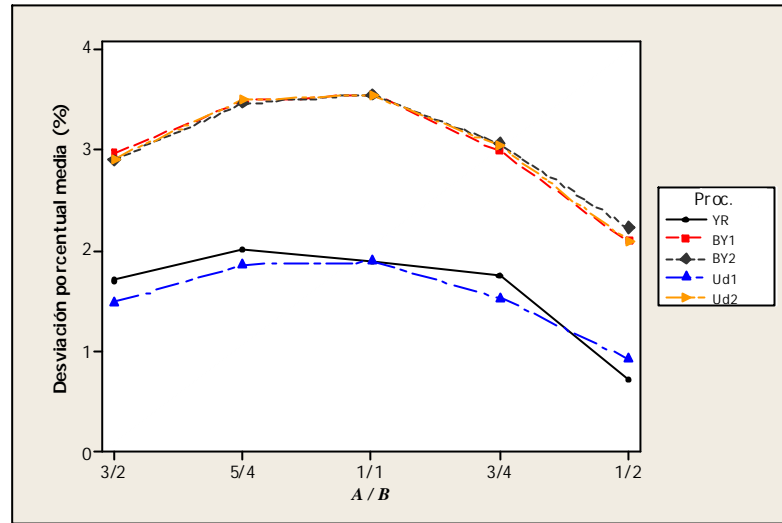
Como se aprecia en la Tabla 3.10, la desviación porcentual media para los diferentes tamaños de la secuencia ( $\delta^T$ ) presenta un leve aumento conforme el total de unidades crece. Como puede observarse en la tabla anterior, el porcentaje de veces que se alcanzó el óptimo es muy bajo, pero esta vez los procedimientos YR y Ud1 alcanzan la cota inferior de sobrecarga para algunas instancias de 36, 110 y 210 unidades a secuenciar. Aparentemente, si se aplica el procedimiento de múltiples estaciones usado en esta experiencia computacional, el número de estaciones de la instancia dificulta más alcanzar el valor de  $lbw$  que el número total de productos a secuenciar  $T$ . Excepto para los procedimientos BY2 y Ud2, el porcentaje de veces que los procedimientos obtienen la mejor solución encontrada, desciende al considerar secuencias más largas. Aunque YR2 obtiene cada vez la mejor solución encontrada conforme el total de unidades aumenta, no es suficiente para compararse con YR y Ud1 para las instancias con  $T = 210$ .

**Tabla 3.11** Índices de calidad para el experimento 2 para ratios de mix

Proced.	$d_l^{A/B}$ (%)						$O_l^{A/B}$ (%)					$G_l^{A/B}$ (%)						
	$A/B$					$d_l$	$A/B$					$O_l$	$A/B$					$G_l$
	$l$	3/2	5/4	1/1	3/4		1/2	3/2	5/4	1/1	3/4		1/2	3/2	5/4	1/1	3/4	
YR	1.70	2.01	1.89	1.74	0.73	1.62	1.56	1.56	6.22	2.00	3.78	3.02	8.44	7.78	28.89	8.67	90.22	28.80
BY1	2.97	3.48	3.55	2.99	2.09	3.02	0.00	0.00	0.00	0.00	0.00	0.00	2.44	5.33	2.89	51.33	6.67	13.73
BY2	2.90	3.46	3.55	3.06	2.23	3.04	0.00	0.00	0.00	0.00	0.00	0.00	40.22	36.00	32.00	4.22	0.00	22.49
Ud1	1.49	1.86	1.89	1.53	0.93	1.54	2.89	3.56	6.00	2.22	0.89	3.11	36.44	55.78	36.89	37.11	4.67	34.18
Ud2	2.92	3.49	3.55	3.04	2.10	3.02	0.00	0.00	0.00	0.00	0.00	0.00	17.78	2.89	30.22	3.56	0.22	10.93
	2.40	2.86	2.89	2.47	1.61		0.89	1.02	2.44	0.84	0.93		21.07	21.56	26.18	20.98	20.36	

Al ver los valores de  $\delta^{A/B}$  en la Tabla 3.11 y la Figura 3.9, puede percibirse que resulta más difícil reducir la desviación porcentual media respecto a la cota  $lbw$  cuando las instancias tienen cantidades similares de productos básicos y de productos especiales. En cambio, cuando las instancias son muy sobrecargadas (mix de 3/2) o muy descargadas (mix de 1/2), los valores que toma  $\delta$  son ligeramente inferiores. Esta misma tendencia se nota en el porcentaje de veces que se alcanzó el valor de  $lbw$  con los procedimientos YR y Ud1. En los resultados para el índice  $G$ , no se percibe una tendencia significativa para los procedimientos respecto a los mixes de producción.

Figura 3.9 Desviación porcentual media para mixes de producción



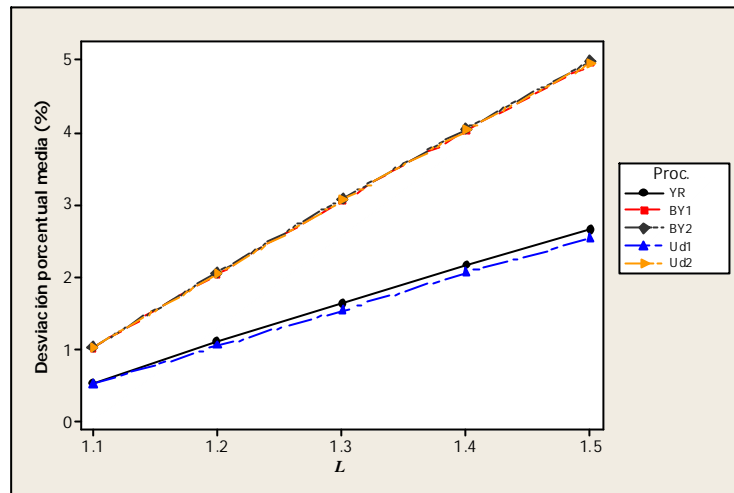
En la Tabla 3.12 se muestran los valores para los tres índices de calidad usados. Esta vez se muestran los índices por procedimiento ( $l$ ) y por la longitud de la estación ( $L$ ). En la Figura 3.10 se puede visualizar la tendencia que presenta la desviación porcentual media al tener en cuenta los diferentes valores de las ventanas. Aunque la desviación porcentual media sigue siendo aceptable, ésta aumenta al aumentar el tamaño de la estación. Los procedimientos YR y Ud1 son los únicos que alcanzan el valor de  $lbw$  en todos los casos considerados para la longitud de la estación. Cuando  $L = 1.3$ , YR y Ud1 obtienen el mismo valor para el índice  $\delta$ . Cuando la ventana de la estación es mas pequeña ( $L = 1.1$ ) YR se comporta sutilmente mejor, mientras que en las ventanas amplias ( $L = 1.5$ ) esta pequeña ventaja la obtiene Ud1.

Tabla 3.12 Índices de calidad para el experimento 2 para longitudes de estación

Proced	$d_l^L$ (%)						$O_l^L$ (%)					$G_l^L$ (%)						
	$L$					$d_l$	$L$					$O_l$	$L$					$G_l$
	1.10	1.20	1.30	1.40	1.50		1.10	1.20	1.30	1.40	1.50		1.10	1.20	1.30	1.40	1.50	
YR	0.54	1.10	1.63	2.16	2.65	1.62	3.33	2.22	3.11	3.33	3.11	3.02	29.78	28.00	30.00	27.33	28.89	28.80
BY1	1.02	2.04	3.06	4.02	4.94	3.02	0.00	0.00	0.00	0.00	0.00	0.00	14.67	12.89	13.33	13.78	14.00	13.73
BY2	1.03	2.06	3.08	4.05	4.98	3.04	0.00	0.00	0.00	0.00	0.00	0.00	22.89	21.56	22.44	22.89	22.67	22.49
Ud1	0.52	1.05	1.54	2.06	2.53	1.54	3.11	2.00	3.11	4.00	3.33	3.11	32.67	34.22	34.44	33.33	36.22	34.18
Ud2	1.02	2.04	3.06	4.03	4.95	3.02	0.00	0.00	0.00	0.00	0.00	0.00	10.89	10.44	10.00	12.00	11.33	10.93
	0.83	1.66	2.47	3.26	4.01		1.29	0.84	1.24	1.47	1.29		21.18	21.42	22.04	21.87	22.62	

En la Figura 3.10 se puede percibir la diferencia que ya se ha hecho notar en tablas y figuras anteriores del experimento 2, respecto a los resultados que ofrecen los procedimientos YR y Ud1 sobre el resto de los procedimientos para el caso de estaciones múltiples. De forma global, los procedimientos YR y Ud2 producen los mejores valores para los tres índices que miden la calidad de las soluciones. Ud1 mejora ligeramente los resultados de YR para ambos índices.

**Figura 3.10** Desviación porcentual media para longitudes de la estación



Ambas experiencias computacionales se realizaron en una SUN sparc station con 4 procesadores hyper SPARC a 100 MHz en ambiente Unix. Los procedimientos están implementados en el lenguaje de programación C.

Los tiempos de procesamiento que arrojan las experiencias son muy aceptables. En la experiencia numero 2, el procedimiento YR, con 4.62 segundos promedio por instancia, es el que más tiempo requiere para solucionar el problema. El resto de los procedimientos requieren en promedio entre 1 y 2 segundos de CPU por instancia.

### 3.4 Resultados

Se estudia una variante del problema de secuencias en una línea de producción que tiene por objetivo minimizar la sobrecarga de trabajo (trabajo perdido) que puede generarse ante la limitación del tiempo concedido en las estaciones para procesar las unidades.

Se recogen tres procedimientos hallados en la literatura (YR, BY1 y BY2) que consideran una sola estación y dos tipos de productos (básico y con opción especial) y se presentan dos propuestas (Ud1 y Ud2), inspiradas en los anteriores. Se toma también de la literatura un procedimiento para estaciones múltiples, el cuál se basa en procedimientos de una estación. Se aplica dicho procedimiento utilizando los procedimientos para una estación descritos en éste capítulo que son útiles para definir heurísticos basados en reglas de prioridad.

Se realiza una experiencia computacional formada por dos experimentos. En el primero se generan 750 instancias que han sido resueltas con los 5 procedimientos para una sola estación. La desviación porcentual media ( $\bar{d}$ ) obtenida en la experimentación es de 0.32%, lo cual muestra un buen comportamiento de todos los procedimientos.

El mejor valor de éste índice lo obtiene el procedimiento Ud2 (0.11%) gracias a que se apoya en reglas sobre los valores de  $lbw$  y  $lbo$  para elegir entre la aplicación de los algoritmos BY1, BY2 o Ud1. Le sigue el procedimiento Ud1 con 0.15% de desviación

porcentual media. El promedio de óptimos confirmados con  $lbw$  es de 79.4% en la experiencia 1. El mejor valor lo tiene Ud2 con 93.2% de óptimos confirmados con  $lbw$ . Ésta vez le sigue el procedimiento YR con 83.7%.

El procedimiento Ud2 presenta el mejor comportamiento global en el experimento 1, en la desviación porcentual media con respecto a la cota inferior  $lbw$  que se ha usado, en el número de óptimos alcanzados y en el número de ocasiones que obtuvo la mejor solución.

En el segundo experimento se generan 2250 instancias con múltiples estaciones y dos tipos de productos. Se aplica el procedimiento descrito en la sección 3.2. Para predecir la sobrecarga por asignar cierto tipo de producto en cada etapa  $t$  y en cada estación  $k$ , se usan los procedimientos para una estación descritos en éste capítulo. Las instancias fueron generadas de acuerdo a diferentes características: tamaño de la ventana ( $L$ ), total de unidades a secuenciar ( $T$ ), ratios de mix ( $A/B$ ), y número de estaciones ( $M$ ).

En el experimento 2 resalta la dificultad que implica el número de estaciones del problema para encontrar buenos valores de sobrecarga con el procedimiento utilizado. La desviación porcentual media para todos los procedimientos aumenta algunos puntos (2.45%) en comparación a la experiencia 1 (0.32%). Sin embargo, el número de óptimos confirmados con valor de  $lbw$  es muy pequeño (1.23%).

Cuando se analizan los datos de acuerdo a los diferentes tamaños de ventana, los procedimientos YR y Ud1 producen los mejores valores para la desviación porcentual media. Todos los procedimientos muestran una tendencia creciente en el valor de la desviación porcentual media conforme el tamaño de la ventana presenta más holgura. Solamente YR y Ud1 encuentran óptimos en todos los tamaños de ventana considerados. En las ventanas ajustadas el resultado de YR es mejor que el de Ud1, mientras que en las ventanas holgadas, Ud1 es un poco mejor que YR.

Los resultados de los procedimientos por longitudes de la secuencia son muy similares a los que se obtienen cuando se analizan por longitudes de ventana. Como se mencionó en el capítulo anterior, las dimensiones del problema se ven afectadas por el total de unidades y por el número de estaciones.

Respecto a los ratios del mix, se ha encontrado que los procedimientos obtienen una desviación porcentual media peor cuando las instancias tienen cargas regulares. Es decir, que para problemas muy sobrecargados o muy descargados, la dificultad para alcanzar el óptimo es menor.

En general, la propuesta Ud1 presenta el mejor comportamiento en los tres índices que miden la calidad de las soluciones para el experimento 2. Le sigue muy de cerca el procedimiento YR. En el capítulo 4 se proponen extensiones para los procedimientos YR y Ud1 en los que se consideran productos múltiples.



---

## Procedimientos para múltiples productos

---

Los procedimientos descritos en el capítulo 3 consideran dos tipos de productos/trabajos que se caracterizan por la presencia o ausencia de opciones extras (p. ej. piloto de aparcamiento o techo solar en los automóviles). Respecto al ciclo, unos productos son ricos en carga de trabajo y los otros son pobres en carga de trabajo. Sin embargo, en la industria de los automóviles los productos pueden requerir opciones que implican muchas variantes. Los asientos de los automóviles por ejemplo, pueden ser de ajuste manual o eléctrico, incluso de ajuste preestablecido, pueden tener respaldo de ajuste eléctrico, o tener calefacción integrada. Cada una de esas variantes es distinguible, entre otras características, por el su contenido de trabajo, por lo cual en el problema de esta tesis se consideran más de dos tipos de productos diferenciables por su tiempo de procesamiento.

Por lo anterior, en este capítulo se proponen extensiones para los procedimientos del capítulo 3. Las extensiones consideran más de dos tipos de productos (tareas u opciones) diferenciables por las cargas de trabajo. Se consideran sistemas con una sola estación con múltiples estaciones. Se realiza una experiencia computacional que considera múltiples estaciones y múltiples tareas. La batería de instancias usada para probar los algoritmos ha sido tomada de la literatura.

### 4.1 Procedimiento para múltiples productos y una estación

En esta sección se considera que en una estación, un producto puede requerir no sólo una sino varias opciones o atributos, tal como sucede en la industria automotriz. En ella, existe una amplia variedad en las versiones de los componentes. Cada una de esas opciones es distinguible, entre otras características, por su contenido de trabajo en cada estación. Por ello se plantea considerar más de dos tipos de trabajos a realizar en las estaciones sobre los productos. Desde este punto de vista, se asume que cada trabajo, distinguible por opción o atributo, tiene diferente tiempo de procesamiento (Scholl et al., 1998), y aun cuando un producto no requiera una opción extra en una estación, es necesaria una cantidad mínima de trabajo sobre él. Se asume que, al inicio de la secuencia, la posición inicial de los trabajadores es en el origen de la estación.

Se propone una extensión del procedimiento de YR y diferentes extensiones de Ud, descritos en la sección 3.1.

#### 4.1.1 Extensión del procedimiento de Yano y Rachamadugu

Se propone una extensión al procedimiento de Yano y Rachamadugu que denominaremos YRx, consistente en establecer esquemas (sub-secuencias) formados por  $m_i$  unidades de

tipo  $i$ . Estos esquemas se repiten mientras existan unidades para completar otra sub-secuencia completa. Cuando ya no sea posible completar otro esquema, se busca uno nuevo con los productos restantes, y así sucesivamente. Sean:

- $p_i$  tiempo de procesamiento sobre la unidad de tipo  $i$ ,
- $A$  conjunto de productos con tiempo de procesamiento mayor al ciclo ( $t_i > c$ ),
- $B$  conjunto de unidades con tiempo de procesamiento menor o igual al ciclo ( $t_i \leq c$ ), y
- $x_i$  cantidad máxima de unidades del producto  $i$  que se pueden secuenciar consecutivamente sin provocar sobrecarga o tiempo ocioso.

Para determinar los productos que conforman un esquema se resuelve el modelo (4.1). Este modelo es una extensión del (3.13).

$$\text{Maximizar } \sum_{i=1}^I p_i m_i / \sum_{i=1}^I m_i \quad (4.1)$$

$$\begin{aligned} \text{Sujeto a } \quad & m_i \leq \min\{x_i, d_i\} \\ & \sum_{i=1}^I p_i m_i \leq \sum_{i=1}^I m_i \\ & m_i \geq 0, \text{ enteros} \end{aligned}$$

Donde ( $x_i \leq L - c / p_i - c, \forall i \in A$ ), y ( $x_i \leq L - c / c - p_i, \forall i \in B$ ).  $d_i$  representa la producción pendiente tras determinar un esquema y replicarlo de forma conveniente en la secuencia en construcción. Para  $t = 0, d_i = n_i$ .

Conocido  $m_i$ , se consideran de forma alternativa unidades de la categoría  $A$  y posteriormente de la  $B$ . El orden de incorporación de unidades del esquema en la secuencia se efectúa en sintonía con el orden decreciente del índice  $r_i = m_i \cdot |c - p_i|$ .

Primero se toma el producto con mayor índice  $r_i$  de la categoría  $A$  (sea  $k$ ) y se asignan consecutivamente, las  $m_k$  unidades que le corresponden; después se toma del grupo  $B$  el producto que tiene el mayor índice  $r_i$  (sea  $l$ ) y se asignan las  $m_l$  unidades correspondientes, y así sucesivamente hasta asignar todas las unidades.

Para el caso de estaciones múltiples, se busca un esquema al menos una vez por cada producto por cada estación en cada periodo de la secuencia para el caso de estaciones múltiples. Para reducir el esfuerzo computacional se realiza el cambio de variables siguiente:

$$y_0 = \frac{1}{\sum_{i=1}^I m_i} \quad (4.2)$$

$$y_i = m_i \cdot y_0 \quad i = 1, \dots, I \quad (4.3)$$

El modelo (4.4) determina los productos que se asignarán dentro del esquema, así como la cantidad de cada uno de ellos.

$$\text{Maximizar } \sum_{i=1}^I p_i \cdot y_i \tag{4.4}$$

$$\begin{aligned} \text{Sujeto a } & y_i - x_i \cdot y_0 \leq 0 && i = 1, \dots, I \\ & \sum_{i=1}^I p_i \cdot y_i \leq c \\ & \sum_{i=1}^I y_i = 1 \\ & y_0, y_i \geq 0, \end{aligned}$$

Donde, como ya se ha mencionado antes, para  $i \in A$ ,  $x_i$  es la cantidad máxima del producto  $i$  que pueden secuenciarse consecutivamente sin causar sobrecarga. Si  $i \in B$ ,  $x_i$  es la cantidad máxima de productos del tipo  $i$  que pueden secuenciarse consecutivamente sin producir tiempo inhábil del trabajador. Si  $y_0$  es múltiplo de  $y_i$  se tiene un ciclo perfecto. La regeneración en la estación es perfecta mientras se pueda completar este patrón con la producción pendiente de los productos  $x_i$  que están incluidos en él. Si  $y_0$  no es múltiplo de  $y_i$  se redondean los valores de la solución del modelo (4.4).

Si el modelo (4.4) no proporciona una solución se asignan productos para formar un ciclo regenerativo de forma heurística. Dado que se va a iniciar un nuevo esquema, el sistema debe encontrarse regenerado o estar muy cerca de este estado. Para determinar el producto que se ha de secuenciar en cada etapa del esquema heurístico, se calcula el índice  $r_i$  y se asigna el producto con el mayor índice. En la etapa ascendente se consideran sólo productos con  $p_i$  mayor al ciclo. En la etapa descendente del ciclo se consideran productos con  $p_i \leq c$ . Para lograr la regeneración completa, se permite incurrir en ocio. La asignación de forma heurística se hace para un único ciclo. De esta forma el sistema está preparado para calcular un nuevo ciclo con el modelo (4.4) con la producción pendiente.

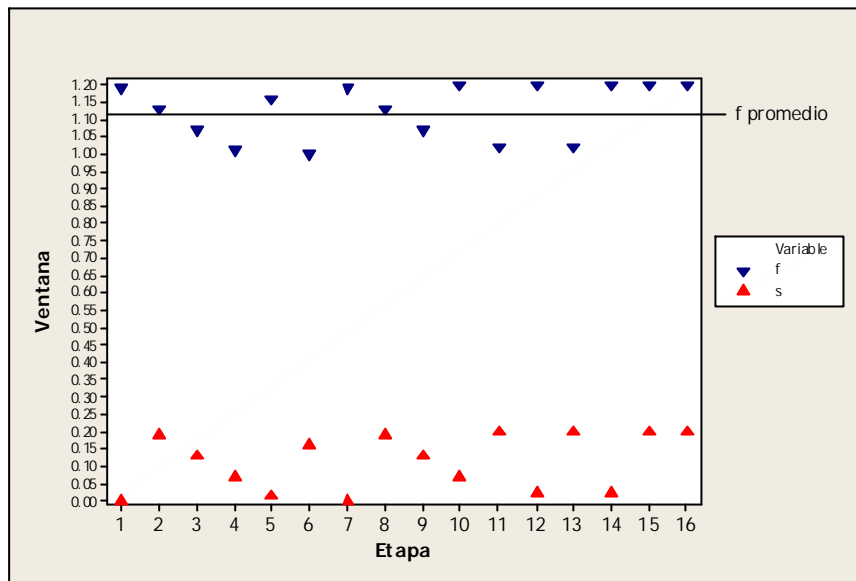
La Tabla 4.1 muestra una secuencia obtenida con la extensión del procedimiento de Yano y Rachamadugu para varios productos. Este ejemplo considera 4 productos: A, B, C y D, con tiempos de procesamiento (0.82, 0.94, 1.19, 1.15), y con demandas (3, 5, 7, 1). La longitud de la estación  $L$  es 1.2 unidades de  $c$ . En el ejemplo se mantiene el valor de  $c = 1$ .

**Tabla 4.1** Secuencia obtenida con YRx

Etapa $t$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Secuencia	C	B	B	B	D	A	C	B	B	C	A	C	A	C	C	C
$f_i$	1.19	2.13	3.07	4.01	5.16	6.00	7.19	8.13	9.07	<b>10.20</b>	11.02	<b>12.20</b>	13.02	<b>14.20</b>	<b>15.20</b>	<b>16.20</b>
$w_i$										0.06		0.01		0.01	0.19	0.19

La secuencia de la Tabla 4.1 está compuesta por tres esquemas diferentes: (1, 3, 1, 1), (0, 2, 1, 0), y (1, 0, 1, 0). La producción pendiente al final del primer ciclo, no permite repetirlo. Igual sucede con el segundo esquema. Una vez calculado el tercer esquema, la producción pendiente permite que este se asigne dos veces. Al final quedan por secuenciar tres unidades del producto C que causan gran parte de la sobrecarga. El trabajo perdido obtenido con esta secuencia es 0.46. El valor de la cota inferior de sobrecarga es de 0.44.

**Figura 4.1** Puntos de cambio de dirección del trabajador con YRx



En la Figura 4.1 se muestran los puntos de cambio de dirección del trabajador para una secuencia obtenida con el procedimiento YRx. La posición final promedio del trabajador es de 1.12 unidades de tiempo respecto al origen de la estación.

#### 4.1.2 Extensiones del procedimiento updown

A partir del procedimiento updown descrito en la sección 3.1.2 se proponen tres versiones para productos múltiples: updown extendido (Udx), updown extendido cerrado (UdxC), y updown extendido regular (UdxR).

Las tres variantes distinguen dos grupos de productos, de categoría  $A$  ( $i \mid p_i > c$ ) y de categoría  $B$  ( $i \mid p_i \leq c$ ). Los procedimientos son de tipo *greedy* que usan como regla de prioridad el índice dinámico  $r_i = d_i \mid c - p_i$  que se renueva para el producto  $i$  cada vez que una unidad es asignada. Este índice prioriza los productos con mayor potencial para ocasionar carga o inactividad, y toma en cuenta la producción pendiente  $d_i$ . Los procedimientos asignan productos en dos fases cíclicas que se identifican como fase ascendente y fase descendente. En la fase ascendente se asignan únicamente productos  $i \in A$ . En la descendente se asignan productos tipo  $B$ . Cada versión propuesta sigue la idea regenerativa de los pseudo-ciclos, pero siguen ideas diferentes para decidir la composición de cada uno. Udx regenera totalmente en cada ciclo, UdxC evita violar las ventanas de la estación, y UdxR regularizar la sobrecarga/ocio a lo largo de la secuencia.

##### *Procedimiento updown extendido*

El procedimiento evita asignar unidades que ocasionen sobrecarga en la fase ascendente del pseudo-ciclo, es decir, cuando el trabajador se acerca cada vez más al límite superior de la estación. En la etapa descendente se busca que el sistema regenere por completo. Por

ello el procedimiento permite incurrir en ocio al final de cada pseudo-ciclo (Algoritmo 4.1).

**Algoritmo 4.1** Udx

---

```

0. Inicializar  $t = 1, s = 0$ .
while ( producción pendiente de ambas categorías )
  while (  $w_t = 0$  )
    1.  $secuencia(t) \leftarrow i^* \in \arg \max_{i \in A} \{r_i\}$ .
    2. Actualizar  $t, s_t$  y  $d_i$ .
  end while
do
    3. Determinar  $i^* \in \arg \max_{i \in B} \{r_i\}$ .
    if (  $i^*$  no ocasiona ocio en  $t$  )
      4.  $secuencia(t) \leftarrow i^*$ , actualizar  $t, s_t$  y  $d_i$ .
    else if (  $i^*$  ocasiona ocio en  $t$  )
      5. Buscar  $i$  de mayor  $r_i$  que no incurra en ocio.
      if (  $i = \text{NULL}$  )
        6.  $secuencia(t) \leftarrow i^*$ , actualizar  $t, s_t$  y  $d_i$ .
      else
        7.  $secuencia(t) \leftarrow i$ , actualizar  $t, s_t$  y  $d_i$ .
      end if
    end if
  while (  $o_t = 0$  )
end while

```

---

Mientras exista producción pendiente de ambas categorías de productos se repiten las dos fases para construir pseudo-ciclos. En la fase ascendente, de las unidades con categoría A con producción pendiente, se asigna aquella con el mayor índice dinámico  $r_i$  siempre y cuando permita al trabajador terminar su tarea sobre el producto antes de que éste abandone la estación. En la fase descendente, de las unidades con categoría B con producción pendiente, se asigna el producto de mayor  $r_i$ , sólo si no ocasiona tiempo inhábil para el trabajador. Si el producto con el mayor  $r_i$  produjera ocio, se asigna otro producto de la categoría B que evite incurrir en él. Si todos los productos de categoría B incurren en ocio, se asigna el de mayor  $r_i$ .

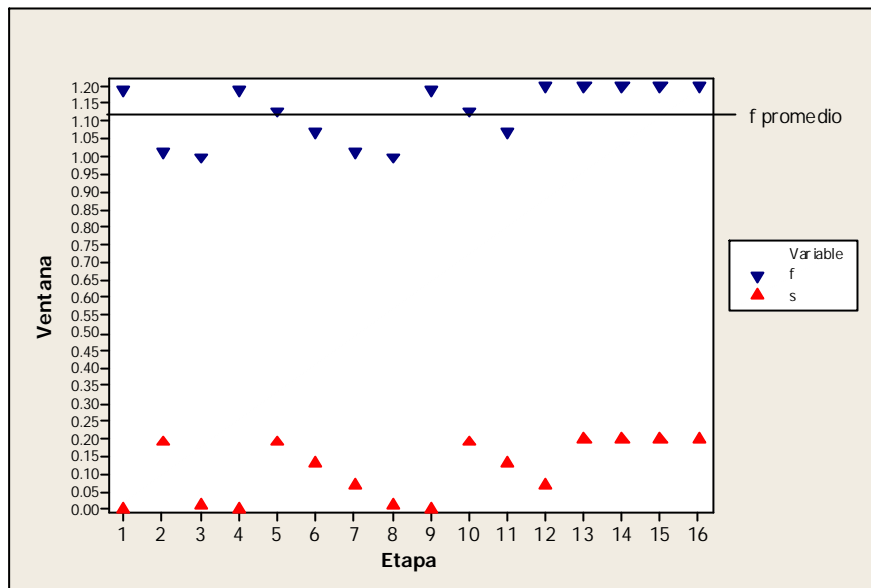
**Tabla 4.2** Secuencia obtenida con Udx

Etapa $t$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Secuencia	C	A	A	C	B	B	B	A	C	B	B	C	C	C	C	D
$f_i$	1.19	2.01	3.00	4.19	5.13	6.07	7.01	8.00	9.19	10.13	11.07	<b>12.20</b>	<b>13.20</b>	<b>14.20</b>	<b>15.20</b>	<b>16.20</b>
$w_t$												0.06	0.19	0.19	0.19	0.15

La secuencia obtenida con Udx (ver Tabla 4.2) ocasiona 0.78 unidades de tiempo de sobrecarga, mucha más de la necesaria. En dos etapas de la secuencia se intenta regenerar

por completo, lo que produce 0.34 unidades de tiempo improductivo: 0.17 unidades en la etapa 3 y otras 0.17 en la etapa 8. Como el sistema no implica ocio obligado, el incurrir en él repercute produciendo su equivalente en sobrecarga para el ejemplo. La Figura 4.2 muestra los puntos de cambio de dirección del trabajador para la secuencia obtenida con el procedimiento Udx. El instante de finalización promedio con Udx es de 1.12 unidades de tiempo de ciclo respecto al origen de la ventana.

**Figura 4.2** Puntos de cambio de dirección del trabajador con Udx



***Procedimiento updown extendido cerrado***

Mientras sea posible, el UdxC evita violar cualquiera de los límites de la estación. Cuando no es posible evitar la violación de alguno de los límites de la estación se incurre en carga u ocio (la magnitud que sea menor). Al evita incurrir tanto en sobrecarga como en ocio, no se prioriza ningún tipo de productos. Con ello se pretende obtener la sobrecarga necesaria u obligada y no más que eso.

Trasladar esta situación a una línea de montaje de automóviles, implica que las sobrecargas en al línea de montaje se presentarán en la parte final periodo de planeación (turno o día). Entonces habrá un requerimiento de relativamente pocos trabajadores de refuerzo al inicio del periodo y más de ellos conforme éste avanza. Si es preferible que las sobrecargas se presenten al inicio del turno, basta con invertir la secuencia. La sobrecarga obtenida con la secuencia invertida será la misma que la secuencia original.

El funcionamiento de UdxC se muestra en el Algoritmo 4.2. UdxC repite las dos fases del pseudo-ciclo mientras exista producción pendiente para ambas categorías de productos. Se asume que el trabajador inicia su trabajo sobre el producto asignado en la primera posición de la secuencia cuando aquel se encuentra en el origen de la estación.

**Algoritmo 4.2** UdxC

---

```

0. Inicializar  $t = 1, s = 0$ .
while ( producción pendiente de ambas categorías)
    1.  $aux = true$ .
    while (  $w_t = 0$  )
        2.  $secuencia(t) \leftarrow i^* \in \arg \max_{i \in A} \{r_i\}$ .
        3.  $aux = false$ , actualizar  $t, s_t$  y  $d_i$ .
    end while
    while (  $o_t = 0$  )
        4.  $secuencia(t) \leftarrow i^* \in \arg \max_{i \in B} \{r_i\}$ .
        5.  $aux = false$ , actualizar  $t, s_t$  y  $d_i$ .
    end while
    if (  $aux = true$  )
        6.  $secuencia(t) \leftarrow i^* \in \arg \max \{r_i\}$  y menor magnitud de
           sobrecarga u ocio,  $\min\{w_t, o_t\}$ , actualizar  $t, s_t$  y  $d_i$ .
    end if
end while

```

---

En la fase ascendente, mientras el trabajador se mantiene dentro de los límites de la ventana en la etapa  $t$ , de las unidades con categoría  $A$  con producción pendiente, se asigna la unidad que tenga el mayor valor del índice dinámico  $r_i$ . En la fase descendente, de las unidades con categoría  $B$  con producción pendiente, se asigna la unidad con el mayor valor  $r_i$  mientras el trabajador no incurra en tiempo improductivo inevitable en la etapa  $t$ . Si tanto en la etapa ascendente como en la descendente no se asignó ningún producto debido a la incidencia en sobrecarga y ocio, respectivamente, en la etapa actual ( $t$ ) de la secuencia se asigna aquel producto que produzca la menor magnitud (sobrecarga/ocio).

**Tabla 4.3** Secuencia obtenida con UdxC

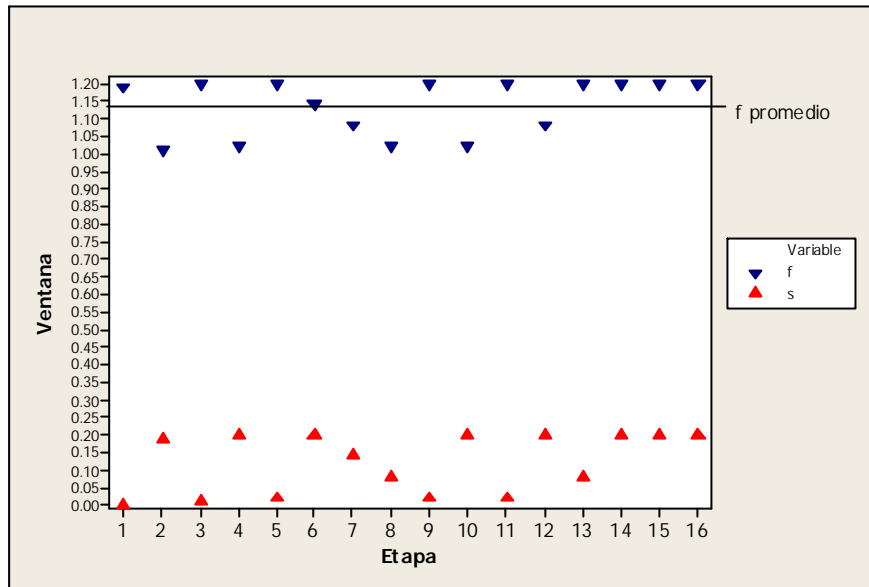
Etapa $t$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Secuencia	C	A	C	A	C	B	B	B	C	A	C	B	B	C	C	D
$f_i$	1.19	2.01	3.20	4.02	<b>5.20</b>	6.14	7.08	8.02	<b>9.20</b>	10.02	<b>11.20</b>	12.14	13.08	<b>14.20</b>	<b>15.20</b>	<b>16.20</b>
$w_t$					0.01				0.01		0.01			0.07	0.19	0.15

La Tabla 4.3 muestra la secuencia obtenida con UdxC para el ejemplo anterior. La fase ascendente y descendente del procedimiento se repiten hasta la etapa  $t = 5$ . En este punto, si el producto de categoría  $A$  con mayor  $r_i$  (producto C) se asigna, se produce sobrecarga de 0.01 unidades de tiempo. Por otra parte, si se asigna el producto de la categoría  $B$  con el mayor índice  $r_i$  (producto B) se produce 0.04 unidades de tiempo improductivo. Así que en  $t = 5$ , se asigna el producto C por ser menor la magnitud de la sobrecarga que la del ocio. La misma situación se repite en las etapas 9 y 11. En ambos casos se asigna el producto C.

En la etapa 7 hay un empate para asignar productos de categoría  $B$  de acuerdo al valor de  $r_i$ . Ambos tienen el valor de 0.18 unidades de tiempo. En tal caso se rompe el empate asignando el producto de la categoría correspondiente con mayor demanda pendiente  $d_i$ .

Después del instante 14, la sobrecarga es inevitable (ver Figura 4.3). El valor de la solución para esta secuencia es de 0.44 unidades de tiempo. El instante promedio de finalización para la secuencia producida con UdxC es de 1.13 unidades de tiempo de ciclo respecto al origen de la estación, una centésima de tiempo más que con YRx y Udx.

**Figura 4.3** Puntos de cambio de dirección del trabajador con UdxC



### ***Procedimiento updown extendido regular***

UdxR toma la idea de la regularidad, usada ampliamente en los problemas de secuenciación de unidades mixtas en líneas de montaje que tienen como objetivo la regularidad en el consumo de componentes o de la aparición de productos. A diferencia del procedimiento anterior, UdxR procura distribuir la sobrecarga a lo largo de la secuencia con lo que se reducen los picos de sobrecarga.

El procedimiento UdxR se apoya en el cálculo de cotas para sobrecarga ( $lbw$ ) y para ocio ( $lbo$ ). Aunque la calidad de las cotas no es muy alta, son una referencia de ambas magnitudes de muy poco coste computacional. La cota de sobrecarga  $lbw$  es la diferencia positiva entre el tiempo mínimo necesario para procesar todos los productos y el tiempo máximo concedido para procesarlos. La cota del trabajo perdido  $lbo$  es la diferencia positiva entre el tiempo mínimo concedido y el tiempo necesario para procesar todos los productos.

$$lbw = \left[ \sum_{i=1}^I n_i \cdot p_i - (T - 1) \cdot c - L \right]^+ \quad (4.5)$$

$$lbo = \left[ (T \cdot c) - \sum_{i=1}^I n_i \cdot p_i \right]^+ \quad (4.6)$$

El procedimiento reparte la sobrecarga/ocio sólo si existe sobrecarga/ocio inevitable. Si se sabe que alguna de las magnitudes ( $lbw$  o  $lbo$ ) es mayor a cero, entonces desde el punto de vista de la regularidad, el ideal de sobrecarga o de ocio es el que se debe incurrir por etapa,

es:  $lbw / T$ , o  $lbo / T$ , respectivamente. Hasta la etapa  $t$ , se puede permitir únicamente  $lbw / T \cdot t$  unidades de sobrecarga, o  $lbo / T \cdot t$  unidades de ocio.

Debido a que, en general,  $lbw$  y  $lbo$  subestiman considerablemente la sobrecarga y el ocio, las experiencias computacionales aconsejaron que se permita incurrir hasta en un 10% más de sobrecarga/ocio de los estimados en cada etapa.

Mientras haya producción pendiente para los productos de ambas categorías, en la etapa ascendente se asigna el producto  $i$  de la categoría  $A$  con el mayor índice  $r_i$ , siempre y cuando hasta la etapa actual  $t$  no ocasione una sobrecarga mayor a  $lbw / T \cdot t$ . En la fase descendente, mientras no se incurras en más trabajo perdido que  $lbo / T \cdot t$  hasta la etapa actual  $t$  de la secuencia, se asigna el producto  $i$  de la categoría  $B$  con el mayor índice  $r_i$ .

**Algoritmo 4.2** UdxR

---

```

0. Inicializar  $t = 1, s = 0$ .
while ( producción pendiente de ambas categorías )
    1.  $aux = true$ .
    while (  $w_t \leq lbw / T \cdot t$  )
        2.  $secuencia(t) \leftarrow i^* \in \arg \max_{i \in A} \{r_i\}$ .
        3.  $aux = false$ , actualizar  $t, s_t$  y  $d_i$  .
    end while
    while (  $o_t \leq lbo / T \cdot t$  )
        4.  $secuencia(t) \leftarrow i^* \in \arg \max_{i \in B} \{r_i\}$ .
        5.  $aux = false$ , actualizar  $t, s_t$  y  $d_i$  .
    end while
    if (  $aux = true$  )
        6. De cada categoría, seleccionar  $i \in \arg \max \{r_i\}$ .
        7.  $s_t =$  carga ocasionada por  $i^* \in \arg \max_{i \in A} \{r_i\}$ .
        8.  $o_t =$  ocio ocasionado por  $i^* \in \arg \max_{i \in B} \{r_i\}$ .
        9.  $secuencia(t) \leftarrow i^* : \min \{|s_t|, |o_t|\}$ , actualizar  $t, s_t$  y  $d_i$  .
    end if
end while

```

---

Sólo para el Algoritmo 4.3 se redefine  $w_t$  y  $o_t$  como la sobrecarga acumulada y el ocio acumulado, respectivamente, hasta la etapa  $t$  de la secuencia.

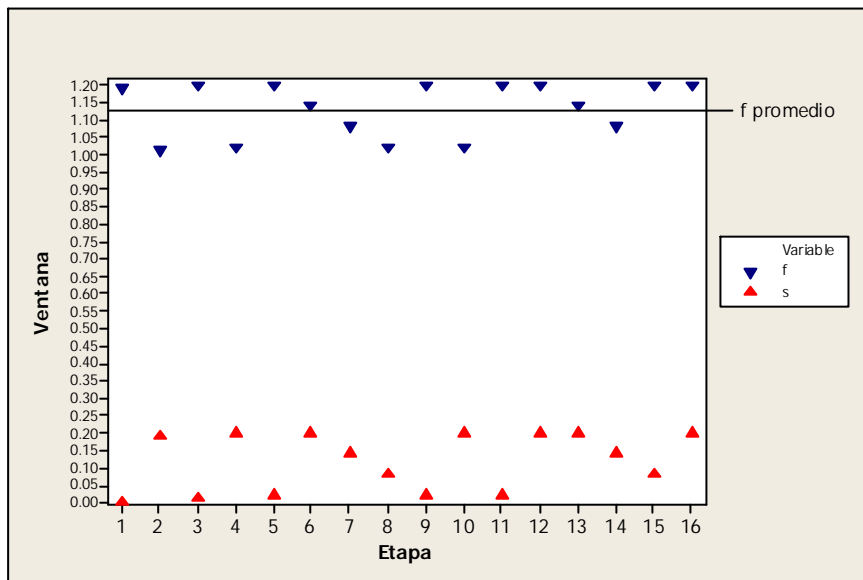
Es posible que el procedimiento busque un producto en las dos fases, ascendente y descendente, sin encontrar un candidato para ser asignado debido a que se puede incurrir en más sobrecarga o tiempo improductivo que el permitido. En tal caso se asigna en la etapa  $t$  el producto  $i$  que produzca la menor diferencia entre el ideal de sobrecarga/ocio y la sobrecarga/ocio real y que tenga el mayor valor  $r_i$  de cada categoría.

**Tabla 4.4** Secuencia obtenida con UdxR

Etapa $t$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Secuencia	C	A	C	A	C	B	B	B	C	A	C	C	B	B	C	D
$f_i$	1.19	2.01	3.20	4.02	<b>5.20</b>	6.14	7.08	8.02	<b>9.20</b>	10.02	<b>11.20</b>	<b>12.20</b>	13.14	14.08	<b>15.20</b>	<b>16.20</b>
$w_i$					0.01				0.01		0.01	0.19			0.07	0.15

La sobrecarga que se puede permitir por etapa es de 0.0275 unidades. A la etapa 5 se llega en la fase ascendente y el producto C es el de mayor  $r_i$ , pero de asignarse ocasionaría sobrecarga de 0.01. La sobrecarga que se puede permitir hasta la etapa 5 es de  $0.0275 \cdot 5 = 0.1375$ , así que este producto se asigna ya que no sobrepasa el nivel permitido. Sucede de forma similar en las etapas 9, 11, 12, 15 y 16. El instante de finalización promedio con el procedimiento UdxR es también de 1.13 unidades de tiempo (Figura 4.4). La sobrecarga producida con esta secuencia es de 0.44 unidades.

**Figura 4.4** Puntos de cambio de dirección del trabajador con UdxR



La Figura 4.5 muestra la sobrecarga acumulada ocasionada por cada uno de los procedimientos. También muestra los valores de la sobrecarga máxima acumulada en la que se debería incurrir por etapa, de acuerdo a (4.5). Los procedimientos UdxC y UdxR obtienen la sobrecarga mínima para el ejemplo con múltiples productos. YRx excede en 0.02 unidades el valor de la cota de sobrecarga. En sus intentos por regenerar por completo Udx obtiene una sobrecarga mucho mayor a la cota. La curva de sobrecarga acumulada para el procedimiento UdxR es la menos distante de la curva del ideal. Una forma de verificar esto, es sumando el valor absoluto de las diferencias entre la sobrecarga ideal acumulada y las sobrecargas reales acumuladas de cada procedimiento. Tales cargas acumuladas se muestran en la Tabla 4.5. La secuencia producida por UdxR tiene picos más pequeños de sobrecarga que el resto de las secuencias, es decir, regulariza mejor las cargas.

Figura 4.5 Sobrecarga acumulada para procedimientos

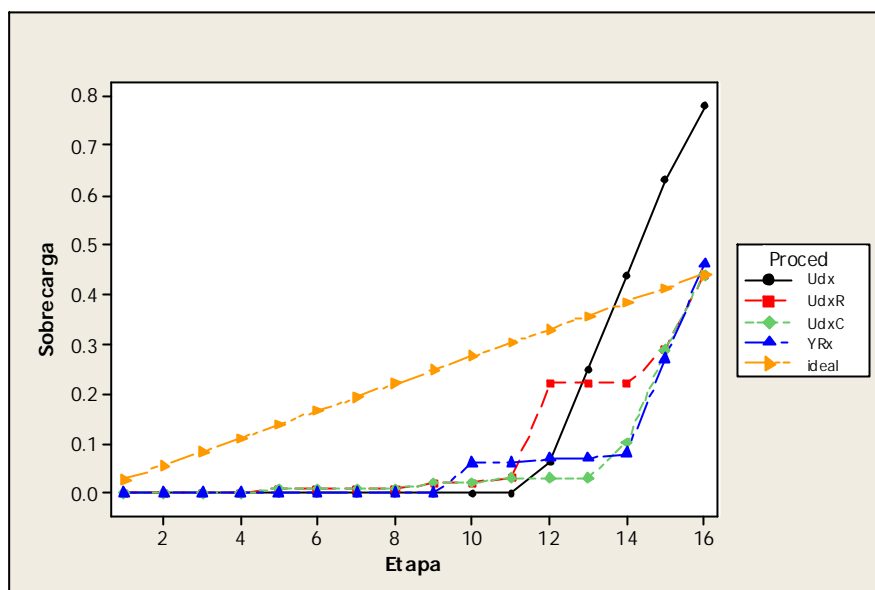


Tabla 4.5 Suma de desviaciones absolutas para procedimientos

$t$	Udx	UdxR	UdxC	YRx
1	0.0275	0.0275	0.0275	0.0275
2	0.0550	0.0550	0.0550	0.0550
3	0.0825	0.0825	0.0825	0.0825
4	0.1100	0.1100	0.1100	0.1100
5	0.1375	0.1275	0.1275	0.1375
6	0.1650	0.1550	0.1550	0.1650
7	0.1925	0.1825	0.1825	0.1925
8	0.2200	0.2100	0.2100	0.2200
9	0.2475	0.2275	0.2275	0.2475
10	0.2750	0.2550	0.2550	0.2150
11	0.3025	0.2725	0.2725	0.2425
12	0.2700	0.1100	0.3000	0.2600
13	0.1075	0.1375	0.3275	0.2875
14	0.0550	0.1650	0.2850	0.3050
15	0.2175	0.1225	0.1225	0.1425
16	0.3400	0.0000	0.0000	0.0200
Suma	<b>2.8050</b>	<b>2.2400</b>	<b>2.7400</b>	<b>2.7100</b>

## 4.2 Procedimiento para múltiples estaciones y productos

Se ha hecho una extensión del procedimiento para estaciones múltiples propuesto por Yano y Rachamadugu (1991) que permite tratar con problemas con múltiples productos. Nuevamente, la extensión se apoya en procedimientos para una estación, pero ahora considera múltiples productos. Con ellos se determina, en cada etapa de la secuencia, cuál es el producto con mejor predicción de sobrecarga por estación. La secuencia se construye progresivamente asignando en cada etapa  $t$  el producto  $i$  con producción pendiente  $d_i > 0$  con el menor valor de predicción de sobrecarga total.

El Algoritmo 4.4 expone la extensión para estaciones múltiples que considera también varios productos. Sean:  $wp_{ik}$  la predicción de sobrecarga ocasionada a partir del periodo  $t$  en la estación  $k$ ,  $w_i(k,t)$  la sobrecarga ocasionada en la estación  $k$  en la etapa  $t$  de la secuencia,  $s_{kt}$  el instante de inicio de la tarea  $k$  en la posición  $t$  (puede verse también como la posición inicial del trabajador dentro de la estación  $k$  en el periodo  $t$ ),  $cs_k$  el coste por incurrir en sobrecarga en la estación  $k$ , y  $sc_i(t)$  la sobrecarga total que el producto  $i$  ocasiona si se asigna en la etapa  $t$ .

---

**Algoritmo 4.4** Extensión para múltiples productos y estaciones

---

```

0. Inicializar  $s = 0$ .
for ( $t = 1$  to  $T$ )
  for ( $i = 1$  to  $I$ )
    1. Suponer secuencia ( $t$ )  $\leftarrow i, (d_i - 1)$ .
    2. Calcular el predictor  $sc_i(t)$  para cada estación.
    3. Reestablecer demanda pendiente  $d_i + 1$ .
  end for
  4. secuencia ( $t$ )  $\leftarrow i^* \in \arg \max_{i \in I} \{sc_i(t)\}$ .
  5. Actualizar datos.
end for

```

---

Para cada etapa  $t$  se determina la predicción de sobrecarga total que puede ocasionarse por la asignación del producto  $i$  como se indica en (4.7).

El valor  $wp_{ik}$  se obtiene con los procedimientos para una sola estación de la sección 3.1.2. Si en el periodo  $t$  de la secuencia se asigna el producto  $i$ , la sobrecarga ocasionada en ese periodo se calcula con (4.8) y (4.9).

$$sc_i(t) = \sum_{k=1}^K (wp_{ik} + w_i(k,t)) \cdot cs_k \quad (4.7)$$

$$w_i(k,t) = [s_{kt} + p_{ik} - ((t-1) \times c) - L_k]^+ \quad \forall i \in A \quad (4.8)$$

$$w_i(k,t) = [(t \times c) - s_{kt} - p_{ik}]^+ \quad \forall i \in B \quad (4.9)$$

$$s_{k1} = 0 \quad \forall k \quad (4.10)$$

Se asume que en la etapa 1 de la secuencia, los trabajadores están esperando la entrada de la primera unidad en el origen de su respectiva estación. Como una alternativa al procedimiento extendido, (4.8) y (4.9) también pueden considerar si la asignación del producto  $i$  produce tiempo improductivo en la etapa  $t$  (Sarker y Pan, 1998).

### 4.3 Experiencia computacional

En esta experiencia se prueba el procedimiento para múltiples estaciones y múltiples productos de la sección 4.2. Para predecir la sobrecarga se usan los procedimientos de una estación y varios productos de la sección 4.1. Se prueba cada procedimiento de una sola estación por separado.

La experiencia se realiza con una batería de 100 instancias diseñadas por Scholl et al. (1998). La unidad de medida continua siendo en tiempo. A diferencia de los trabajos de Yano y Rachamadugu, los autores de las instancias establecen el tiempo de ciclo igual a 90 unidades de tiempo. No considera costes o ponderaciones en las estaciones cuando se incurre en sobrecarga. Aunque el problema que resuelve Scholl es el mismo que se estudia aquí, Scholl considera que cuando los trabajadores de todas las estaciones han terminado su trabajo sobre la última unidad de la secuencia, en el siguiente instante todos ellos estarán en el origen de su respectiva estación ( $\forall k, S_{k,T+1} = 0$ ). Por lo anterior y porque en Scholl et al. (1998) no se presentan los valores de sobrecarga de las instancias (presenta índices globales que comparan sus procedimientos entre sí, debido a la dificultad para obtener el óptimo o una cota inferior de calidad), no se comparan los resultados con este autor a pesar de que la batería es la misma.

Las instancias están diseñadas con diferentes cantidades de estaciones, productos y total de unidades. Hay dos grupos de instancias pequeñas, cuatro con instancias de tamaño medio (diez estaciones y diez productos), un grupo de instancias con el doble de productos que de estaciones, uno más con el doble de estaciones que de productos y un grupo con instancias grandes (veinte estaciones y veinte productos). A excepción de dos grupos de instancias, el resto de los grupos usan la misma longitud de ventana para todas las estaciones. La longitud de la estación de dichos grupos de instancias varía entre [88, 132] unidades. 10 instancias usan longitud de estación de tamaño  $L = 150$ . El total de unidades a secuenciar varían desde 10 hasta 400. La Tabla 4.6 muestra las características generales de la batería.

**Tabla 4.6** Características de la batería de instancias

Grupo	Instancias	$I$	$K$	$L_k$	$T$ (replicas)
1	1-5	5	5	110	10(2), 20(2), 30(1)
2	6-10	5	5	[88, 132]	10(2), 20(2), 30(1)
3	11-40	10	10	110	50, 80, 110, 140, 170, 200, 250, 300, 350, 400 (3 de cada uno)
4	41-50	20	10	110	140, 200, 250, 300, 350 (2 de cada uno)
5	51-60	10	20	110	140, 200, 250, 300, 350 (2 de cada uno)
6	61-70	20	20	110	140, 200, 250, 300, 350 (2 de cada uno)
7	71-80	10	10	[88, 132]	140, 200, 250, 300, 350 (2 de cada uno)
8	81-90	10	10	150	140, 200, 250, 300, 350 (2 de cada uno)
9	91-100	10	10	110	140, 200, 250, 300, 350 (2 de cada uno)

Los tiempos de procesamiento de los productos en las estaciones se calculan de forma aleatoria de acuerdo a ciertas condiciones. Información detallada sobre la generación de los datos puede verse en Scholl et al. (1998).

Las instancias del grupo 9 consideran la existencia de un producto básico. Sus demandas para cada producto también son aleatorias. Los tiempos de procesamiento se generan de forma aleatoria siempre que  $p_{ik} \leq \min \{p_{2k}, \dots, p_{lk}\}$  sea válido para todas las estaciones de la instancia. El grupo 9 se considera especial por reflejar la situación real de las ensambladoras de automóviles, en donde un producto puede requerir o no, diferentes opciones de acuerdo al modelo del producto. Estas opciones demandan más atención de los trabajadores de las estaciones.

Para medir la calidad de las soluciones obtenidas con cada procedimiento se usa el índice de desviación relativa global usado por Scholl et al. (1998). En su trabajo, los autores de

las instancias evitan usar una cota inferior pobre para comparar los procedimientos. En vez de ello usa el valor de la función objetivo ( $wo_h^*$ ) de la mejor solución encontrada para la instancia  $h$ . Dado que el valor de la función objetivo puede tomarle valor cero, se usa el siguiente índice de desviación relativa global  $rel.wo$ .

$$rel.wo : \left( \sum_{h=1}^{100} wo_h - \sum_{h=1}^{100} wo_h^* \right) / \sum_{h=1}^{100} wo_h^* \cdot 100\% \quad (4.11)$$

En este trabajo se han usado dos variantes más de  $rel.wo$ . El índice usado por Scholl et al. (1998) equivale al índice que aquí denominamos  $rel.wo2$ . Así, en:

- $rel.wo1$   $wo_h^*$  es la cota inferior de la sobrecarga  $lbw$  (4.12),
- $rel.wo2$   $wo_h^*$  es la mejor solución encontrada con los procedimientos propuestos, y
- $rel.wo3$   $wo_h^*$  es la mejor solución encontrada, considerando los procedimientos propuestos y las soluciones encontradas con el software de optimización CPLEX durante 15 minutos de búsqueda.

$$lbw = \sum_{k=1}^K \left[ \sum_{i=1}^I n_i \cdot p_{ik} - (c \cdot (T-1) + L_k) \right]^+ \quad (4.12)$$

Las pruebas se realizaron en un ordenador Pentium 4 CPU 2.4 GHz, 512 MB RAM, con sistema operativo Microsoft Windows XP Professional 2002. Para resolver las instancias con el procedimiento YRx se uso también CPLEX 7.5. Los procedimientos se han implementado en lenguaje de programación C.

**Tabla 4.7** Desviación relativa global

	Udx	UdxC	UdxR	YRx
$rel.wo1$ (%)	45.64	41.82	40.97	69.25
$rel.wo2$ (%)	5.49	2.73	2.11	27.90
$rel.wo3$ (%)	6.00	3.22	2.59	28.51
#mejor	19	42	44	2
CPU (seg)	3.99	4.96	4.56	4267

En la Tabla 4.7 se presentan los valores para los tres índices de desviación relativa global (%), el número de veces que cada procedimiento obtuvo la mejor solución (#mejor) y el tiempo promedio (en segundos) necesario para encontrar la solución para una instancia. El procedimiento YRx requiere mucho más tiempo de CPU para construir una solución que el resto de los procedimientos ya que en cada etapa de la secuencia y para cada producto candidato debe calcular al menos una vez un esquema de productos con el modelo 4.2. Además, la extensión propuesta YRx no ofrece los mejores valores de desviación relativa global.

En 40 de las 100 instancias, CPLEX mejora las soluciones encontradas por los procedimientos propuestos. En cuatro instancias, se obtuvo la misma sobrecarga con CPLEX y con lo mejor de las versiones de Udx. En esas cuatro instancias se alcanza el óptimo. Dos de esos óptimos se logran con Udx, uno con UdxC y otro con UdxR. Los procedimientos UdxC y UdxR obtienen los mejores valores para los índices de desviación relativa global. Considerando que el índice  $rel.wo3$  toma en cuenta la solución obtenida

por CPLEX después de 15 minutos de búsqueda, puede decirse que el tiempo que requieren Udx, UdxC y UdxR para encontrar una solución es aceptable.

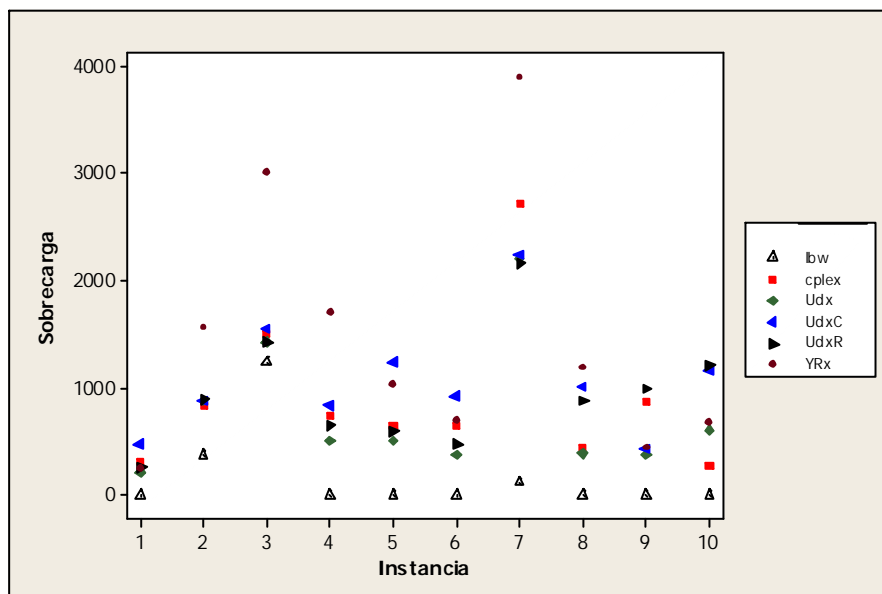
Se realiza el mismo análisis considerando sólo las instancias del grupo 9. Se analizan aparte, ya que como se mencionó, reflejan mejor la realidad de las líneas de montaje (Tabla 4.8).

**Tabla 4.8** Desviación relativa global para instancias con producto básico

	Udx	UdxC	UdxR	YRx
<i>rel.wo1</i> (%)	334.23	526.05	455.32	743.87
<i>rel.wo2</i> (%)	0.0073	0.4523	0.2882	5.8481
<i>rel.wo3</i> (%)	0.0605	0.529	0.3562	6.2096
#mejor	8	1	1	0
CPU (seg)	2.78	3.20	2.14	2430.1

El valor de *lbw* de 7 de las 10 instancias del grupo 9, es igual a cero. Con CPLEX se encontraron mejores soluciones para dos de las 10 instancias. El índice *rel.wo1* crece dramáticamente debido a que siete de las cotas obtenidas con (4.12) son iguales a cero. La Figura 4.6 expone gráficamente las sobrecargas obtenidas para cada una de las instancias del grupo 9, la cota inferior de sobrecarga *lbw* y la solución obtenida con CPLEX en 15 minutos de búsqueda.

**Figura 4.6** Sobrecarga para las instancias del grupo 9



Las instancias del grupo 9 son de carga muy baja o nula. Esta vez, Udx es el procedimiento que obtiene menor desviación relativa. En 8 de las 10 instancias Udx encuentra la mejor solución. El mejor desempeño de Udx se atribuye a que las instancias son de carga muy baja y a que el procedimiento Udx busca regenerar permitiendo incurrir en tiempo improductivo. El tiempo necesario para encontrar una solución para las instancias del

grupo 9 se reduce casi a la mitad si se compara con el tiempo global promedio por instancia mostrado en la Tabla 4.7. Nuevamente, el desempeño de la extensión propuesta YRx no es muy bueno.

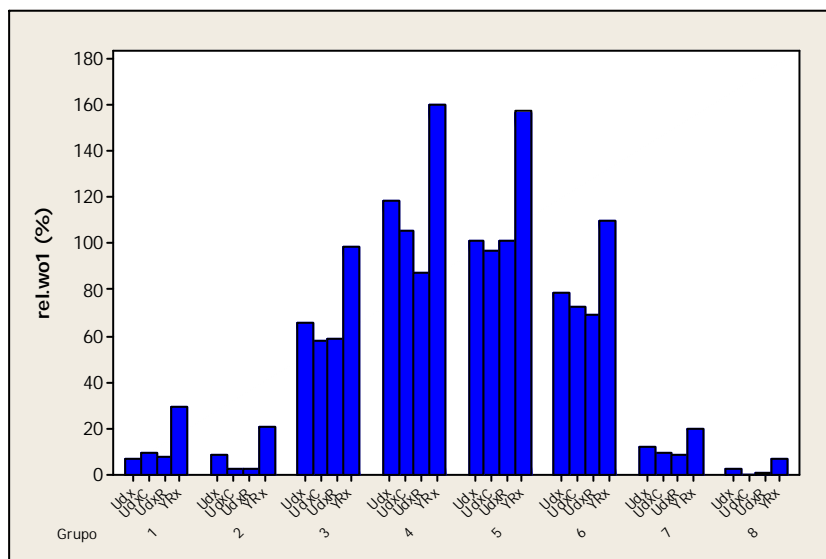
**Tabla 4.9** Desviación relativa global para grupos

Grupo	Instancias	I	K	L <sub>k</sub>	rel.wo1 (%)				rel.wo2 (%)				rel.wo3 (%)			
					Udx	UdxC	UdxR	YRx	Udx	UdxC	UdxR	YRx	Udx	UdxC	UdxR	YRx
1	1-5	5	5	110	6.885	9.932	8.126	29.57	0.014	0.043	0.026	0.229	0.040	0.069	0.052	0.026
2	6-10	5	5	[88, 132]	8.846	3.036	2.515	20.99	0.064	0.008	0.003	0.183	0.085	0.027	0.022	0.206
3	11-40	10	10	110	66.06	58.26	58.97	98.19	0.088	0.037	0.042	0.299	0.101	0.049	0.054	0.313
4	41-50	20	10	110	118.2	105.1	87.42	159.9	0.164	0.094	0.000	0.387	0.164	0.094	0.000	0.387
5	51-60	10	20	110	101.8	96.67	101.88	157.1	0.041	0.015	0.041	0.326	0.041	0.015	0.041	0.326
6	61-70	20	20	110	78.81	72.98	69.19	109.6	0.063	0.028	0.006	0.246	0.063	0.028	0.006	0.246
7	71-80	10	10	[88, 132]	12.49	9.443	8.863	19.84	0.042	0.013	0.008	0.110	0.048	0.020	0.014	0.116
8	81-90	10	10	150	2.333	0.507	1.297	7.741	0.019	0.001	0.009	0.073	0.023	0.004	0.012	0.077
9	91-100	10	10	110	334.2	526.1	455.3	743.9	0.007	0.452	0.288	0.968	0.060	0.529	0.356	1.061

La Tabla 4.9 contiene los tres diferentes índices de desviación relativa para cada grupo de instancias. En la Figura 4.7 se representan los valores de índice *rel.wo1* de la tabla anterior, excepto para el grupo 9, ya que al no estar incluido se aprecia mejor las diferencias entre los grupos restantes, además de que ya se hicieron comentarios respecto a dicho grupo anteriormente.

Los grupos 1 y 2 son los más pequeños, pues contienen el menor número de instancias (5) y los valores de los parámetros son también los más pequeños. La única diferencia entre estos dos grupos es que en el primero, las longitudes de las estaciones es el mismo para todas, mientras que en el segundo, la longitud de cada estación puede variar dentro del rango [88, 132] unidades de tiempo de ciclo. El índice *rel.wo1* tiene valores más pequeños en el grupo de las estaciones con diferentes tamaños de ventana.

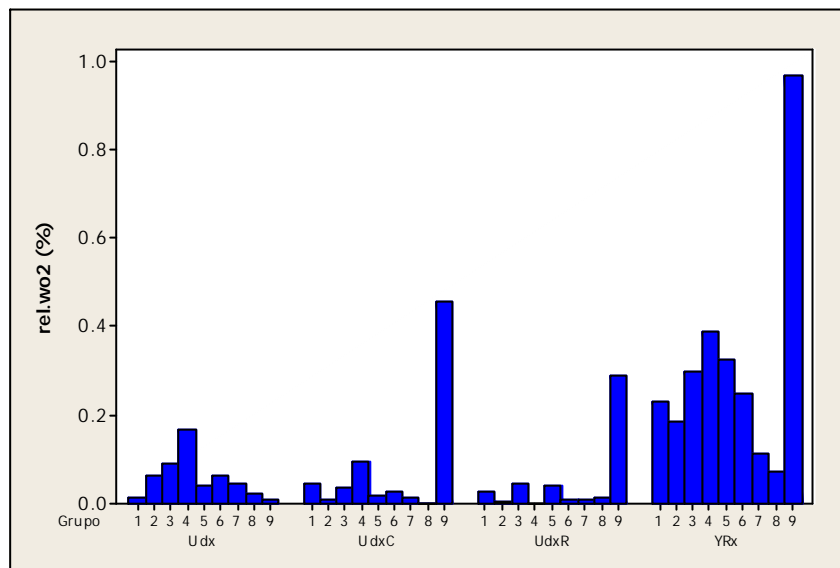
**Figura 4.7** rel.wo1 para grupos y procedimientos



Considerando ambos grupos (grupo 1 y 2), se tienen 10 instancias. Para 9 de estas 10 instancias se encontró el valor óptimo con CPLEX. De estas 9 instancias, la cota inferior de sobrecarga usada (*lbw*) coincide con 7 de ellas, lo cual permite vislumbrar la calidad de dicha cota. Para una de las instancias de cada grupo se encontró el óptimo con alguno de los procedimientos propuestos.

El grupo 3 está conformado por 30 instancias de 10 estaciones y 10 productos. El valor *T* de las instancias varía entre 50 y 400 productos. Éste grupo ofrece mejores valores para *rel.wo1* que los grupos 4, 5 y 6, los cuales tienen el doble de productos, de estaciones o de ambos parámetros.

Figura 4.8 rel.wo2 para grupos y procedimientos



El grupo 7, con ventanas variables, 10 productos y 10 estaciones, ofrece valores de *rel.wo1* aceptables si se compara con los del grupo 3 que tiene ventana fija y el mismo número de estaciones y productos. Algo similar sucede entre los grupos 1 y 2. El grupo 8, al tener ventanas más amplias, presenta *rel.wo1* mejores a cualquier otro grupo.

Tabla 4.10 Desviación relativa global para parámetros

Parámetro		<i>rel.wo1</i> (%)				<i>rel.wo2</i> (%)				<i>rel.wo3</i> (%)			
		Udx	UdxC	UdxR	YRx	Udx	UdxC	UdxR	YRx	Udx	UdxC	UdxR	YRx
<i>I</i>	5	7.994	6.032	4.953	24.72	0.042	0.023	0.013	0.204	0.065	0.045	0.035	0.230
	10	35.62	32.69	33.31	56.93	0.047	0.025	0.029	0.212	0.054	0.031	0.036	0.219
	20	87.74	80.26	73.72	121.0	0.088	0.044	0.004	0.281	0.088	0.044	0.004	0.281
<i>K</i>	5	7.994	6.032	4.953	24.72	0.042	0.023	0.013	0.204	0.065	0.045	0.035	0.230
	10	30.16	26.95	25.74	47.33	0.060	0.033	0.024	0.199	0.067	0.041	0.031	0.208
	20	88.76	83.25	83.36	130.1	0.053	0.022	0.022	0.283	0.053	0.022	0.022	0.283
<i>L</i>	110	84.99	79.74	77.85	126.2	0.074	0.043	0.032	0.313	0.079	0.048	0.037	0.319
	150	2.333	0.507	1.297	7.741	0.019	0.001	0.009	0.073	0.023	0.004	0.012	0.077
	[88, 132]	12.44	9.36	8.78	19.85	0.042	0.013	0.008	0.111	0.048	0.020	0.014	0.118

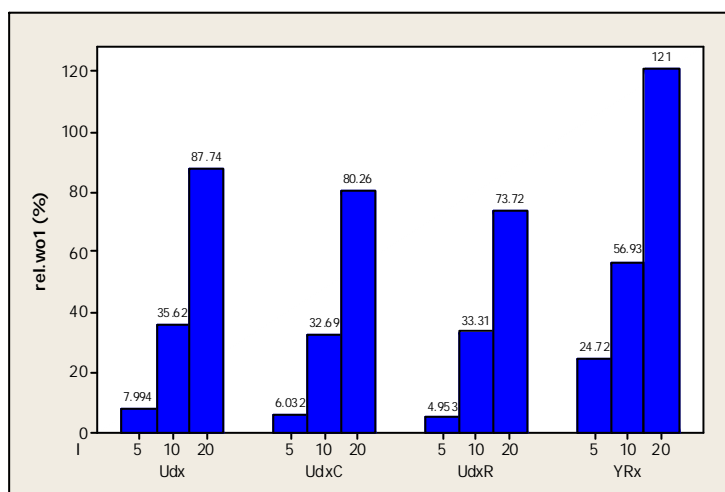
En la Figura 4.8 se puede apreciar cual es el mejor procedimiento para cada grupo. En definitiva, el procedimiento Udx funciona mejor para los grupos 1 y 9, el procedimiento UdxC es el mejor procedimiento para los grupos 3, 5, 8, el UdxR es mejor con los grupos 2, 4, 6, y 7.

El procedimiento UdxR es el mejor procedimiento para el mayor número de grupos. Como me indica en la Tabla 4.7, es el mejor procedimiento global, a pesar de que UdxC es el mejor procedimiento para 3 grupos que hacen un total de 60 instancias, 60% del total.

La Tabla 4.10 presenta otro análisis de los datos, esta vez agrupados por parámetros ( $I$ ,  $K$  y  $L$ ). Cada uno de ellos tiene tres niveles. Para el número de productos y número de estaciones, se tienen tres categorías o niveles: de 5, 10 y 20. Para el parámetro  $L$ , hay tres niveles: 110, 150 y 88-132; es decir, ventana angosta, ventana amplia y ventanas de diferentes tamaños entre el rango 88-132 unidades de tiempo de ciclo, respectivamente. Un nuevo producto entra en la línea cada 90 unidades de tiempo, así que el promedio de la longitud de las ventanas debería estar alrededor de 110. Para  $I = 5$ , UdxR ofrece un *rel.wol* ligeramente mejor que Udx y UdxC. Gráficamente (Figura 4.9), para  $I = 10$ , Udx, UdxC y UdxR, los respectivos valores del índice *rel.wol* parecen estar empatados. UdxR resuelve mejor las instancias que tienen 20 productos. El número de estacione del problema ( $K$ ) dificulta la obtención de buenas soluciones tanto como lo puede hacer el número de productos ( $I$ ). Los valores *rel.wol* para los diferentes longitudes de la estación ( $K$ ) presentan un comportamiento muy similar al de las diferentes cantidades de productos ( $I$ ).

En definitiva, el aumento en el número de estaciones y del número de productos hace más difícil la resolución del problema, en el mismo grado, con el procedimiento de resolución utilizado en la presente experiencia.

**Figura 4.9** rel.wol1 para número de productos y procedimientos



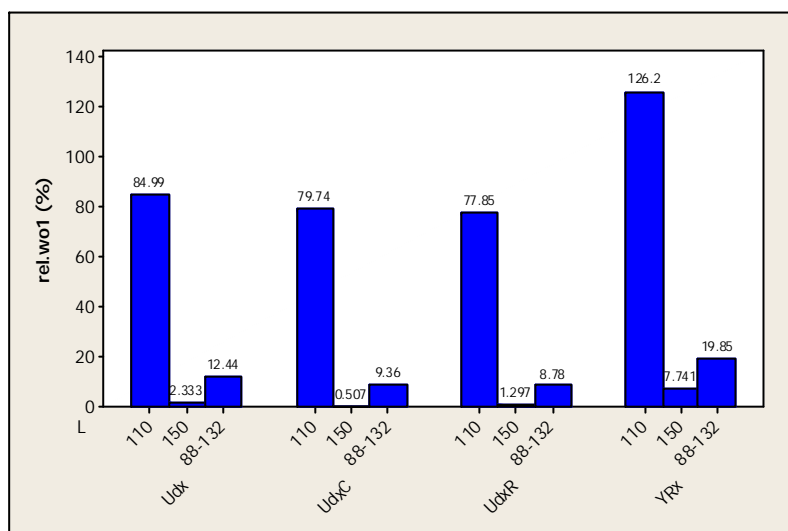
Por mucho, las estaciones angostas ( $L = 110$ ) ofrecen mayor dificultad para solucionar el problema, mientras que en los problemas con estaciones amplias ( $L = 150$ ), se obtienen valores muy cercanos a *lbw* (Figura 4.10). En éste caso, UdxC ofrece las mejores soluciones, seguido por UdxR y Udx. De hecho, al buscar el óptimo con CPLEX en un tiempo máximo de búsqueda de 15 minutos para este grupo de instancias, se encontró

dicho valor para 7 de 10 de ellas. Como se mencionó anteriormente, con CPLEX se encontró el óptimo del 16% de las instancias del total en 15 minutos de búsqueda. Ninguno de los óptimos de las instancias con estaciones amplias (o el valor de  $lbw$ ) es igual o cercano a cero, por lo que no se puede considerar que las instancias estén “descargadas”.

También se comentó que el promedio de las longitudes de las estaciones, cuando éstas son de longitud variables, debe ser cercano a 110, así que hay estaciones angostas y amplias en la misma instancia. Los valores de los índices  $rel.wo$  para éstas instancias se reducen significativamente si se comparan con las instancias angostas ( $L = 110$ ).

Para dichas instancias y para instancias con estaciones angostas, el procedimiento UdxR ofrece el mejor resultado. UdxC funciona mejor para instancias amplias.

Figura 4.10  $rel.wo1$  para tipo de ventana y procedimiento



## 4.4 Resultados

Se proponen extensiones para dos de los procedimientos descritos en el capítulo 3. Estas extensiones consideran productos múltiples y se aplican como predictores de sobrecarga en el procedimiento para múltiples estaciones.

Se proponen tres extensiones del procedimiento Ud1 del capítulo 3. Una versión permite incurrir en ocio para procurar la generación del sistema (Udx), otra versión evita incurrir en sobrecarga y ocio mientras sea posible posponerlas (UdxC), y una tercera alternativa procura repartir la sobrecarga y el ocio sobre la secuencia a partir de bs valores de las cotas usadas de sobrecarga y ocio (UdxR).

Se realiza una experiencia computacional con una batería de 100 instancias. La batería maneja 9 grupos de instancias que comparten los mismos valores para varios parámetros.

A grandes rasgos, con poca diferencia respecto a Udx y UdxC, UdxR obtiene la desviación relativa global más pequeña. La extensión propuesta YRx, requiere mayor tiempo de

## Capítulo 4

procesamiento y los resultados no son cercanos a los que se obtienen con el resto de las propuestas. El mejor resultado lo ofrece UdxR y el tiempo de procesamiento es aceptable.

Para las instancias con producto básico (grupo 9), el mejor valor de desviación relativa lo ofrece el procedimiento Udx, lo cual se atribuye a que las instancias son “descargadas” y a que el procedimiento permite regenerar el sistema aunque se incurra en tiempo improductivo.

Los grupos más pequeños son el 1 y 2, los cuales se diferencian porque el primero considera estaciones del mismo tamaño y el segundo no. Se obtienen mejores soluciones para el grupo 2, con ventanas variables en tamaño de una a otra. Esta situación se repite en si se comparan los grupos 3 y 7. Ambos tienen los mismos valores para los parámetros  $I$  y  $K$ , pero el primero tiene ventanas del mismo tamaño y el segundo no. Los procedimientos encuentran mejores soluciones para el grupo 7. De igual manera, obtener una buena solución para el problema es menos complicado cuando las ventanas son más amplias.

El procedimiento Udx ofrece las mejores soluciones para las instancias pequeñas con ventanas iguales y para instancias con producto básico. UdxC produce las mejores soluciones para el 60% de las instancias, sin embargo, los índices globales para éste procedimiento son ligeramente mayores a los de UdxR.

UdxR resuelve mejor las instancias que tienen el máximo de productos considerados (20 productos). El número de estaciones del problema dificulta obtener una buena solución tanto como lo puede hacer el número de productos. Es más difícil obtener buenas soluciones para las instancias con ventanas angostas que instancias con ventanas amplias. UdxC funciona mejor para instancias amplias. UdxR produce mejores soluciones cuando se tienen estaciones de diferentes longitudes y cuando se tienen estaciones angostas.

---

## Búsqueda local

---

Después de probar procedimientos constructivos, en el presente capítulo se aplican procedimientos de mejora. Se aplican dos tipos de vecindarios para realizar Búsqueda Local (BL): intercambios de elementos e inserción de segmentos. La experiencia computacional de este capítulo considera múltiples estaciones y múltiples productos. La batería de instancias usada en la experiencia computacional de este capítulo es la misma del capítulo 4. Como soluciones iniciales de la BL se usan las soluciones proporcionadas por los procedimientos constructivos del capítulo anterior.

### 5.1 Optimización combinatoria y búsqueda local

La optimización combinatoria (Stützle, 1998; Nemhauser y Wolsey, 1988; Papadimitriou y Steiglitz, 1982) es una disciplina de la toma de decisiones para alternativas discretas. La optimización combinatoria tiene sus raíces en la combinatoria, la investigación operativa y las ciencias computacionales (Korte y Vygen, 2000).

Formalmente, la optimización combinatoria puede definirse como el estudio matemático para encontrar un arreglo o solución óptima, agrupando, ordenando o seleccionando objetos discretos, usualmente finitos en número (Osman y Laporte, 1996). Muchos problemas combinatorios son *NP-hard* y se cree que estos problemas no pueden resolverse en tiempo polinomial de forma óptima (Aarts y Lenstra, 1997).

Un problema combinatorio puede definirse de la siguiente manera. Dados:

- Un conjunto finito  $E$ ,
- Una familia  $\mathcal{I}$  de subconjuntos de  $E$ ,  $\mathcal{I} \subseteq \mathcal{P}(E)$  (soluciones factibles) ( $\mathcal{P}(E)$  denota las partes del conjunto  $E$ ),
- Una función de coste

$$\begin{aligned} c: \mathcal{I} &\longrightarrow \mathbb{R} \\ \mathcal{I} &\longrightarrow c_{\mathcal{I}} \end{aligned}$$

Un problema de optimización combinatoria consiste en hallar  $\mathcal{I}^* \in \mathcal{I}$  tal que

$$c(\mathcal{I}^*) = \text{Min} \{c(\mathcal{I}), \mathcal{I} \in \mathcal{I}\}$$

Habitualmente, se estudian problemas de optimización combinatoria con función objetivo lineal que son aquellos en los que la función de coste viene determinada por la suma de los costes de los elementos que intervienen en la solución. Es decir:  $c(I) = \sum_{e \in I} c_e$ . Así, un problema de optimización combinatoria se suele representar por la terna  $(E, \mathcal{I}, c)$ .

Una forma de abordar estos problemas es usando algoritmos polinomiales que den soluciones aproximadas. Existen dos categorías de tales algoritmos: algoritmos de aproximación y algoritmos heurísticos. Un algoritmo de aproximación es un algoritmo que entrega una solución con una garantía teórica de cercanía al óptimo. Es frecuente que estos algoritmos posean un desempeño práctico muy superior a su garantía teórica. En las últimas décadas, los algoritmos de aproximación han sido (y continúan siendo) un tema central de investigación en computación teórica y su aplicabilidad es cada vez más evidente. En la literatura se distinguen dos grandes ramas de algoritmos de aproximación: constructivos y búsqueda local. La BL es una buena opción cuando el tamaño del problema o su estructura dificultan la aplicación de algoritmos de optimización. La BL brinda alternativas robustas para obtener soluciones de alta calidad para problemas de tamaños reales en tiempos razonables. La BL comenzó a aplicarse desde los años 50's para resolver el Traveling Salesman Problem (TSP) en Bock (1958), Croes (1958), Lin (1965), Reiter y Sherman (1965). Y en la década de los años 60's se aplicaba a problemas de Scheduling, en trabajos como Page (1965) o Nicholson (1965).

Los métodos de búsqueda local se emplean para explorar entornos construidos por soluciones vecinas. En esta tesis la solución consiste en una secuencia de automóviles, la cual establece el orden en que éstos se introducen en la línea de montaje.

Para evitar la búsqueda exhaustiva en todo el espacio de soluciones, la BL dirige la atención al vecindario local de alguna solución en particular.

Si se centra la atención sobre una región de espacio de búsqueda cercana a un punto particular del espacio, se puede describir esto como buscar en el vecindario de un punto. Gráficamente, considérese un espacio abstracto  $S$  con un punto en particular  $x \in S$ . Lo intuitivo es que el vecindario  $N(x)$  de  $x$  es el conjunto de todos los puntos del espacio de búsqueda  $S$  que están cercanos en un sentido cuantificable, al punto  $x$ . Así, un problema de búsqueda puede definirse de la siguiente manera:

Dado un espacio de búsqueda  $S$  y su parte factible  $F \subseteq S$ , encontrar  $x \in F$  tal que  $eval(x) \leq eval(y)$  para todo  $y \in F$ .

Si se define la función de distancia  $dist$  en el espacio de búsqueda  $S$ :  $dist : S \times S \rightarrow \mathbb{R}$ , entonces se puede definir el vecindario  $N(x)$  como  $N(x) = \{ y \in S : dist(x,y) < \epsilon, \text{ para } \epsilon > 0 \}$ . Otra forma de representarlo es, si  $y$  satisface la condición  $N(x)$  entonces  $y$  está en el  $\epsilon$ -vecindario de  $x$ . Luego, una solución potencial  $x \in F$  es un óptimo local con respecto al vecindario  $N$ , si y sólo si  $eval(x) \leq eval(y)$  para todo  $y \in N(x)$ . En este caso se asume que existe un criterio de minimización.

Un procedimiento de BL se puede describir como se muestra en el Algoritmo 5.1.

El desempeño del algoritmo de búsqueda local, depende del tipo de transformación aplicada a la solución actual. Al buscar dentro del vecindario de una solución se impone una guía para buscar, y cuando se ha encontrado algo mejor se puede actualizar la solución y mantener lo que se ha aprendido. Si el tamaño del vecindario es pequeño, la búsqueda en el vecindario debería ser rápida y se puede llegar a un óptimo local rápidamente. Por otra parte, si el tamaño del vecindario es grande, la posibilidad de llegar rápidamente a un óptimo local es menor, pero la eficiencia de la búsqueda puede complicarse.

**Algoritmo 5.1** Búsqueda local

1. Tomar una solución del espacio de búsqueda y evaluar el criterio que interesa de ella. Definirla como mejor solución “actual”.
2. Aplicar una transformación a la solución actual para generar una nueva solución y evaluar la solución.
3. Si la nueva solución es mejor que la solución actual, convertir la nueva solución en la solución actual, de lo contrario, descartar la nueva solución.
4. Repetir pasos 2 y 3 hasta que ninguna transformación en la solución actual mejora el resultado.

Algunas maneras de ir de una solución a otra solución vecina son las siguientes:

- **Transponer.** Cambiar dos elementos adyacentes.
- **Insertar.** Mover un elemento de su posición en la secuencia e insertarlo en otra posición.
- **Intercambiar.** Intercambiar dos elementos que pueden no ser adyacentes.
- **Insertar bloques.** Mover un grupo de elementos adjuntos de su posición en la secuencia e insertarlos en otra.

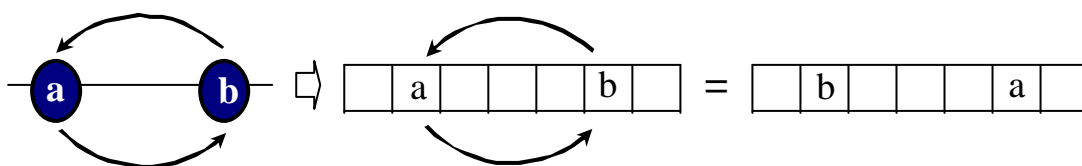
En las secciones siguientes de este capítulo se describen los vecindarios que se han usado en esta tesis para realizar la búsqueda local para el problema de secuenciación de productos mixtos. La experiencia se centra en dos clases de vecindarios: vecindarios por intercambios y vecindarios por inserción de bloques o segmentos.

## 5.2 Intercambio

La exploración de vecindarios por intercambio de elementos ha sido aplicada ampliamente para resolver el TSP. Los intercambios se realizan entre dos o más elementos de la solución. Debido al coste computacional que implica explorar este tipo de vecindarios, en esta tesis se consideran sólo los intercambios de dos y tres elementos de la secuencia.

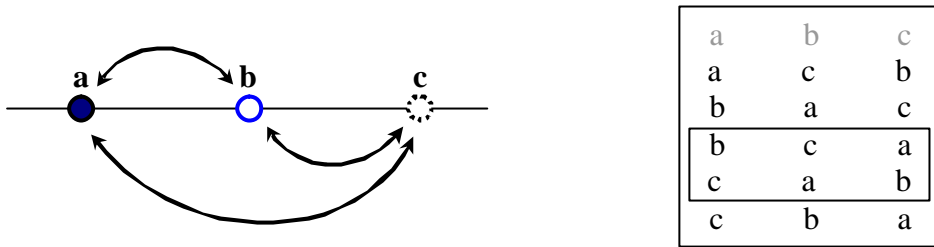
Puede decirse que una representación natural de una solución para el problema de secuenciación es una permutación de elementos. Así, para nuestro problema, un vecindario del tipo dos-intercambio consiste en seleccionar dos elementos de la solución y colocar cada uno en la posición original del otro, como se indica en la Figura 5.1.

**Figura 5.1** 2 Intercambio



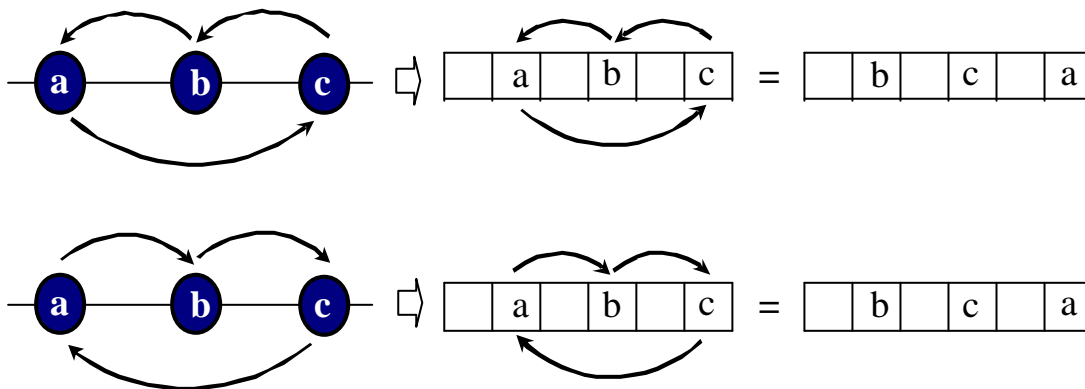
Un vecindario tres-intercambio, consiste en combinar los tres elementos de una solución a intercambiar, lo cual, a diferencia del dos-intercambio, conduce a más de una solución vecina. El intercambio de tres elementos de una solución dada, produce las combinaciones o soluciones vecinas que se muestran en la Figura 5.2. En algunas de estas cinco combinaciones puede haber alguno de los elementos que no cambian de posición, como sucede en (a, c, b), (b, a, c) y (c, b, a). En la primera, el elemento “a” no cambia de posición, en la segunda el elemento “c” y en la tercera el elemento “b”.

Figura 5.2 3 Intercambio



Por otra parte, en dos de las combinaciones se han movido de posición los tres elementos considerados para el intercambio “a”, “b” y “c”. Los cambios (a, c, b), (b, a, c) y (c, b, a) pueden considerarse como un intercambio de dos elementos, mientras que las vecinas con los cambios (b, c, a) y (c, a, b) tienen todos los elementos en posiciones diferentes a las originales. A estas últimas se les ha considerado como tres-intercambio “reales”. En la Figura 5.3 se muestran los intercambios “reales” de los 3 elementos.

Figura 5.3 3 Intercambio “reales”



En la experiencia computacional se han probado dos alternativas para el intercambio de tres elementos. Se les ha denominado 3Int(a) y 3Int(b). La alternativa 3Int(a) evalúa sólo dos de las 5 combinaciones, aquellas donde realmente cambian de posición los tres elementos a intercambiar, mientras que la alternativa 3Int(b) considera las cinco combinaciones descritas en la Figura 5.2. A diferencia de los intercambios usados para el TSP, en el problema de secuenciación se consideran también intercambios de elementos adyacentes tanto para 2-Intercambio como para 3-Intercambio.

### 5.3 Inserción

Para obtener una solución vecina a partir de una solución actual por inserción de segmentos, se debe extraer un segmento de cierto tamaño (sea  $or$ ) de una secuencia factible actual y después se inserta en otra posición diferente de la secuencia. La idea de este tipo de vecindario se representa en la Figura 5.4, y el procedimiento se describe en el Algoritmo 5.2.

---

#### Algoritmo 5.2 Búsqueda local para inserción de segmentos

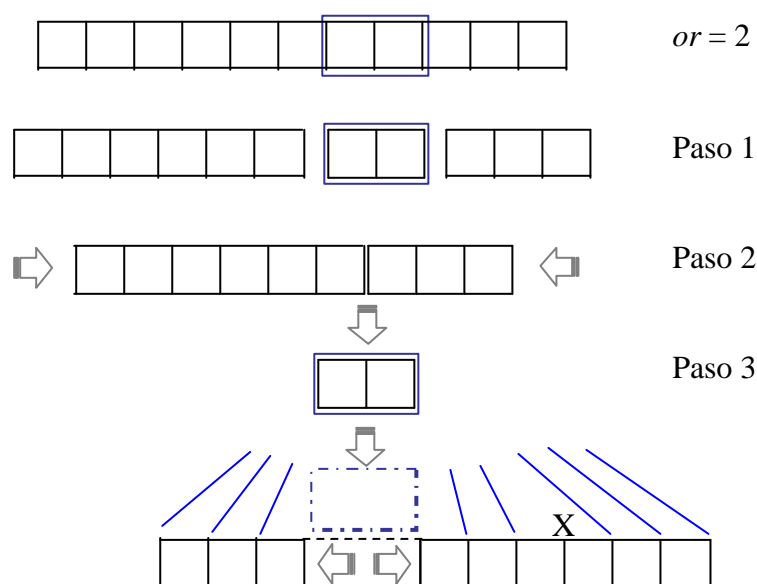
---

Dada una solución factible del problema, y un tamaño determinado de segmento  $or$ , el procedimiento itera los pasos siguientes:

1. Extraer el segmento de tamaño  $or$  de la secuencia.
  2. Unir las dos partes de la secuencia como si fueran una sola.
  3. Insertar el segmento en cada una de las posiciones de la secuencia y medir la carga,
  4. Si la función objetivo actual es mejor que la incumbente, actualizarla.
  5. Tomar la mejor secuencia del vecindario como solución inicial para buscar en otro nuevo vecindario.
  6. Parar cuando se exceda el número de iteraciones sin mejora, de lo contrario ir a 2.
- 

En este trabajo el segmento que hemos denominado  $or$  no debe asimilarse exactamente como el  $or$  que se aplican a problemas de TSP. El TSP representa de otra forma una solución del problema. Por lo tanto, la aplicación de la inserción de segmentos es diferente. De hecho, como ya se mencionó, lo mismo sucede con los intercambios.

**Figura 5.4** Inserción de segmentos



Para cualquier procedimiento de búsqueda local que se ha probado en la experiencia computacional de este capítulo, el número máximo de iteraciones sin mejora ha sido establecido en  $\sqrt{T}$ . Aunque los tamaños de segmento pueden variar desde 1 hasta  $T - 1$ , en la experiencia computacional sólo se ha considerado  $2 \leq or \leq 10$ . En cada iteración el tamaño de la vecindad es  $T - or$ , lo cual es considerablemente menor que el tamaño del vecindario de un 2 intercambio.

La evaluación de una solución vecina para el problema tiene un coste computacional considerable. También es difícil determinar de forma heurística si cierto movimiento de elementos va a mejorar la función objetivo o va a empeorarla. Desde el inicio de la secuencia hasta la posición del elemento a intercambiar que se encuentra más a la izquierda de la secuencia (asumiendo que el inicio y fin de la secuencia es la izquierda y derecha, respectivamente) la sobrecarga no varía, sin embargo a partir de este punto de la secuencia las sobrecargas obtenida en cada posición pueden cambiar. Puede ocurrir que a partir de cierto punto en la secuencia posterior a las posiciones en las que hubo cambios, la sobrecarga por etapa vuelva a ser la misma. También puede ocurrir que las cargas no modifiquen su valor entre las posiciones en que ha habido modificaciones, ya que aunque los valores de sobrecarga pueden ser diferentes, las posiciones de los trabajadores pueden ser iguales. Así, si el elemento en la segunda posición de la secuencia se intercambia con cualquier otro, será necesario evaluar la secuencia casi por completo.

### 5.4 Experiencia computacional

Se comparan los resultados de las búsquedas locales por intercambio de elementos y por inserción de segmentos descritas anteriormente. Las secuencias resultantes de los procedimientos constructivos del capítulo 4 se toman como soluciones iniciales para la BL. Por lo tanto, cada variante de la BL se prueba con cuatro soluciones iniciales diferentes.

Se prueban cuatro procedimientos de intercambio de elementos: 2 Intercambio, 3 intercambio “puro”, 3 Intercambio y 2-3 Intercambio. Se hace referencia a estos procedimientos por 2Int, 3Int(a), 3Int(b) y 2-3Int(b), respectivamente. El vecindario 3Int(a) considera para la búsqueda de soluciones vecinas aquellas en las que los 3 elementos considerados cambian da posición, mientras que el vecindario 3Int(b) toma en cuenta las cinco combinaciones que se mostraron en la Figura 5.2. Los intercambios 2-3Int(b) realiza la BL en dos fases: en la primera aplica el 2 intercambio hasta que el número de iteraciones sin mejora es alcanzado, en la segunda fase la búsqueda continua con 3Int(b) hasta llegar a un mínimo local.

En la BL por inserción de segmentos se prueban tamaños de segmentos entre 2 y 10 elementos, separadamente. El tamaño máximo del segmento se estableció en 10 porque los tamaños de secuencia de las instancias más pequeñas son de  $T = 10$ . Naturalmente, a tales instancias no se aplicó la búsqueda local con este tamaño de segmento. En total se prueban 4 alternativas para la BL por intercambio y 9 alternativas para la BL por inserción de segmentos.

Las soluciones iniciales corresponden a la batería de 100 instancias de Scholl et al. (1998), descrita brevemente en el capítulo 4. Al igual que en el capítulo anterior, la calidad de las soluciones se mide con el índice de desviación relativa global  $rel.wo$  (5.1) usado por Scholl et al. (1998).

$$rel.wo: \left( \sum_{h=1}^{100} wo_h - \sum_{h=1}^{100} wo_h^* \right) / \sum_{h=1}^{100} wo_h^* \cdot 100\% \quad (5.1)$$

Donde  $wo_h^*$  representa la cota inferior de sobrecarga  $lbw$ , o la mejor solución encontrada por los procedimientos comparados para la instancia  $h$ . Para el primer caso usamos el índice  $rel.wo1$ , y para el segundo  $rel.wo2$ .

En el capítulo 4 se aplicó el índice  $rel.wo3$  que considera la solución encontrada por CPLEX durante 15 minutos de búsqueda. Este índice no se usa en esta experimentación porque todas las soluciones encontradas con los procedimientos de BL son mejores que las que ofrece CPLEX durante dicho periodo de búsqueda. Así, se tiene que en:

$rel.wo1$   $wo_h^*$  es la cota inferior de la sobrecarga  $lbw$  (4.12),  
 $rel.wo2$   $wo_h^*$  es la mejor solución encontrada con los procedimientos de BL.

La mayoría de los procedimientos se probaron en un Netserver de 3HD 9.1 G Ultra2 SCSI, RAM 384MB. Sólo el procedimiento 3Int(b) se probó en una PC Pentium 4 CPU 2.4 GHz, 512 MB RAM, con sistema operativo Microsoft Windows XP Professional 2002. Los procedimientos se han implementado en lenguaje de programación C.

A todos los procedimientos se les impuso límite en el tiempo de búsqueda. Para los procedimientos de búsqueda por intercambio de elementos 2Int, 3Int(a) y 3Int(b) el límite de tiempo de búsqueda fue de 3600 segundos. El procedimiento 2-3Int(b), en la fase I (2Int) tiene límite de tiempo de 3600 segundos, en la fase II (3Int(b)) se limitó el tiempo de búsqueda a 1800 segundos. Para todos los procedimientos con inserción de segmentos se limitó la búsqueda a 1800 segundos.

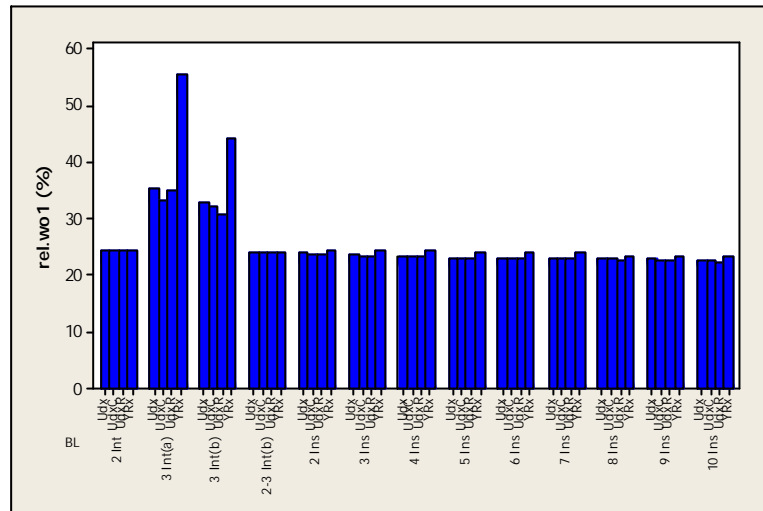
**Tabla 5.1** Desviación relativa global  $rel.wo1$

$rel.wo1$ (%)		Procedimiento Inicial			
		Udx	UdxC	UdxR	YRx
BL	Inicial	45.64	41.82	40.97	69.25
Intercambios	2Int	24.32	24.30	24.53	24.54
	3Int(a)	35.16	33.03	34.84	55.39
	3Int(b)	32.99	31.86	30.55	44.15
	2-3Int(b)	23.96	23.88	24.09	24.06
Inserciones	2 Ins	23.88	23.64	23.80	24.38
	3 Ins	23.51	23.35	23.34	24.22
	4 Ins	23.34	23.20	23.19	24.25
	5 Ins	23.15	23.00	23.00	24.02
	6 Ins	23.01	22.95	22.99	24.03
	7 Ins	22.98	22.87	22.77	23.91
	8 Ins	22.82	22.71	22.63	23.34
	9 Ins	22.72	22.61	22.60	23.35
	10 Ins	22.56	22.65	22.44	23.29

La Tabla 5.1 muestra los valores de  $rel.wo1$  tanto para los procedimientos de búsqueda por intercambio como para los procedimientos por inserción de segmentos. Las mejoras que se obtienen con la BL son sustanciales respecto a los índices de las soluciones iniciales

obtenidas con los procedimientos constructivos. El índice de desviación relativa global se reduce en general a la mitad de su valor inicial (Figura 5.5).

**Figura 5.5** rel.wo1, para procedimiento de BL y solución inicial



Como se mencionó, *rel.wo1* se obtiene teniendo como referencia el valor de la cota de sobrecarga *lbw* (4.12), sin embargo, la calidad de *lbw* no es muy buena. Se ha usado esta cota porque es fácil de calcular. En la experiencia del capítulo anterior, se usó el software CPLEX para encontrar soluciones a las 100 instancias con PLEM. El tiempo de búsqueda de la solución óptima se limitó a 15 minutos. En este tiempo se encontró el óptimo de 9 de las instancias, instancias pequeñas (con tamaños de secuencias de hasta 30 unidades). El valor de la cota inferior de sobrecarga *lbw* para estas instancias coincide con el valor del óptimo en 7 de ellas. En las dos instancias restantes el valor de la cota es inferior al óptimo.

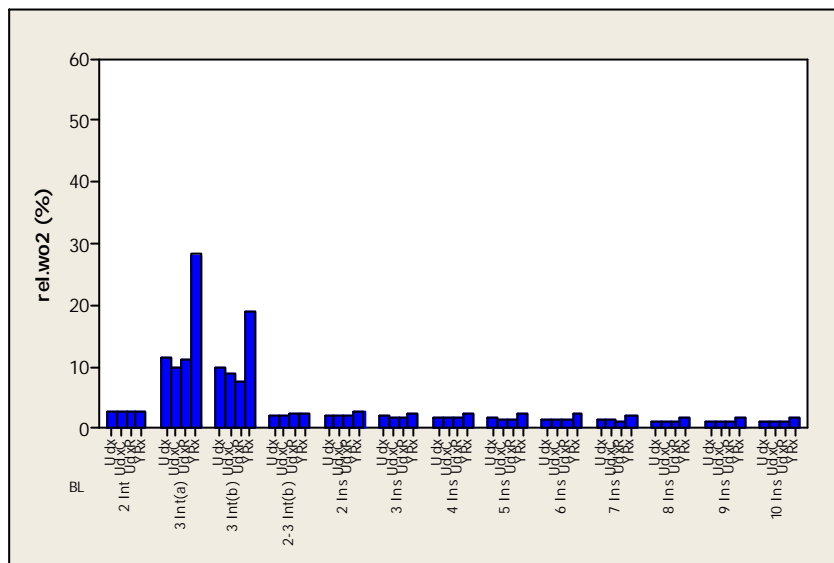
**Tabla 5.2** Desviación relativa global *rel.wo2*

<i>rel.wo2</i> (%)		Procedimiento Inicial			
		Udx	UdxC	UdxR	YRx
BL	Inicial	20.12	16.97	16.37	40.16
Intercambios	2Int	2.53	2.51	2.70	2.71
	3Int(a)	11.47	9.72	11.21	28.16
	3Int(b)	9.68	8.75	7.67	18.89
	2-3Int(b)	2.23	2.17	2.34	2.32
Inserciones	2 Ins	2.17	1.97	2.11	2.58
	3 Ins	1.86	1.74	1.72	2.45
	4 Ins	1.73	1.61	1.60	2.48
	5 Ins	1.57	1.45	1.44	2.28
	6 Ins	1.45	1.40	1.44	2.29
	7 Ins	1.43	1.34	1.26	2.20
	8 Ins	1.29	1.21	1.14	1.73
	9 Ins	1.21	1.12	1.12	1.73
	10 Ins	1.08	1.16	0.98	1.69

En la Figura 5.6 pueden verse las diferencias en la desviación relativa global obtenidas con las diferentes alternativas de BL para cada una de las cuatro soluciones iniciales Udx, UdxC, UdxR y YRx. En esta ocasión se compara con la mejor solución obtenida. En general, cuando la solución inicial es de YR se obtienen los mayores valores para *rel.wo1*.

Al igual que sucede en la experimentación del capítulo anterior, los valores de las soluciones obtenidas con YR no son tan buenos como las obtenidas con el resto de los procedimientos iniciales. En promedio, en el 25.15 % de las instancias se alcanzó el límite del tiempo de búsqueda con YRx, mientras que con Udx, UdxC y UdxR, el promedio fue de 20.38, 19.23 y 20.23 respectivamente.

**Figura 5.6** rel.wo2, para procedimiento de BL y solución inicial



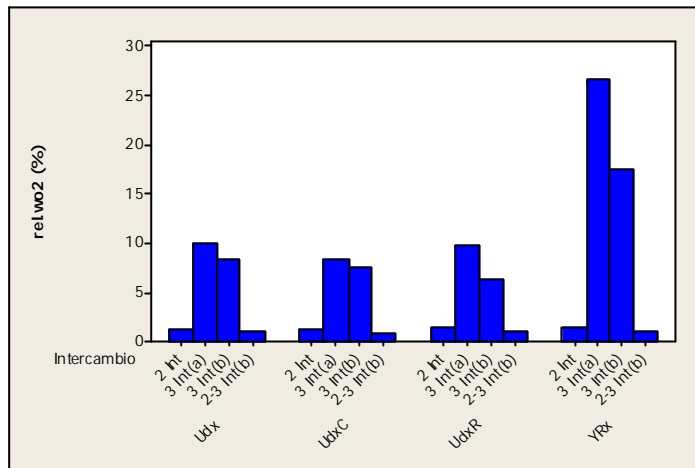
Los valores para la desviación relativa global respecto de las mejores soluciones encontradas con las alternativas de BL (*rel.wo2*) se exponen en la Tabla 5.2. La Tabla 5.3 contiene el mismo comparativo únicamente para intercambio de elementos.

**Tabla 5.3** rel.wo2 para intercambios

<i>rel.wo2</i> (%)	BL	Solución inicial				
		Inicial	Udx	UdxC	UdxR	YRx
			18.58	15.47	14.88	38.37
	Intercambios	2Int	1.22	1.2	1.39	1.4
		3Int(a)	10.04	8.31	9.78	26.52
		3Int(b)	8.28	7.36	6.29	17.37
		2-3Int(b)	0.93	0.86	1.03	1.01

En 2-3Int(b), la fase II del procedimiento mejoró el resultado 73% de las veces. Como puede verse en la Tabla 5.6, 19.75% de las veces en promedio, ya se había alcanzado el valor de la cota inferior *lbw* en la fase uno (2Int). Así, el 6.75% de las veces en promedio no se mejoró en la fase II de 2-3Int(b). Usar de forma aislada 3Int(a) y 3Int(b) produce resultados pobres. Esta diferencia puede apreciarse gráficamente en la Figura 5.7.

**Figura 5.7** rel.wo2, para intercambio de elementos y solución inicial

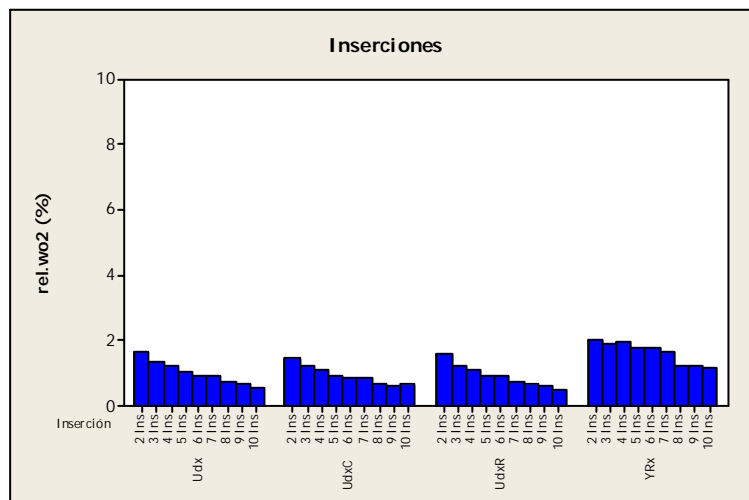


**Tabla 5.4** rel.wo2 para inserciones

BL	Rel.wo2 (%)	Procedimiento inicial			
		Udx	UdxC	UdxR	YRx
Inserciones	Inicial	19.5	16.36	15.77	39.44
	2 Ins	1.64	1.44	1.58	2.05
	3 Ins	1.33	1.21	1.20	1.92
	4 Ins	1.20	1.09	1.08	1.95
	5 Ins	1.04	0.92	0.92	1.75
	6 Ins	0.93	0.88	0.91	1.76
	7 Ins	0.91	0.82	0.73	1.67
	8 Ins	0.77	0.68	0.62	1.20
	9 Ins	0.69	0.60	0.59	1.21
	10 Ins	0.55	0.63	0.46	1.16

Por otro lado, se ha realizado otro comparativo entre los mejores resultados obtenidos con los procedimientos de BL por inserción de segmentos (Tabla 5.4 y Figura 5.8).

**Figura 5.8** rel.wo2, para inserción de segmento y solución inicial



En todos los procedimientos, los valores de *rel.wo2* por inserción de segmentos muestran una tendencia a mejorar conforme el tamaño del segmento aumenta. Sin embargo esta tendencia no se observó para las instancias más pequeñas con tamaños de secuencia  $T = 10$ .

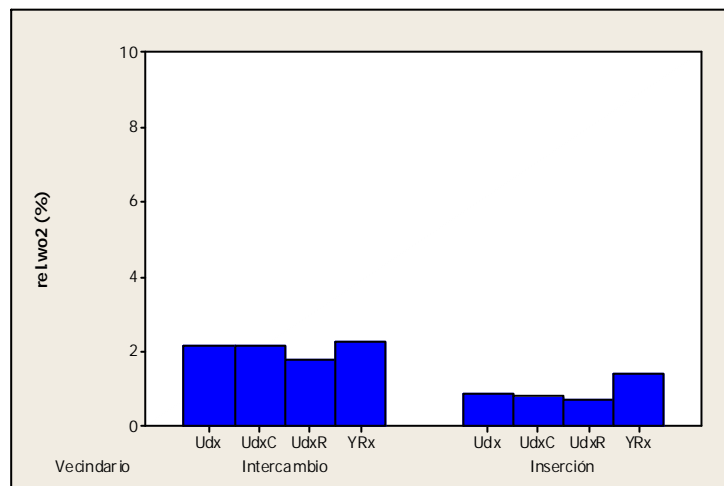
**Tabla 5.5** Comparativo *rel.wo2*, para intercambios e inserciones

<i>rel.wo2</i> (%)	Procedimiento Inicial			
	Udx	UdxC	UdxR	YRx
Inicial	20.12	16.97	16.37	40.16
Intercambio	2.15	2.12	1.74	2.26
Inserción	0.84	0.79	0.70	1.37

A partir de los tamaños de segmento de tamaño 6, se deja de alcanzar los valores de las cotas inferiores en las instancias más pequeñas. Esto sugiere que pueden obtenerse los mejores resultados para una instancia cuando se usa un tamaño de segmento cercano e inferior  $T/2$ .

La Tabla 5.5 se obtiene al tomar los mejores valores de las soluciones que se ha obtenido para cada vecindario (intercambios o inserciones) y cada procedimiento inicial. En definitiva el mejor índice de desviación relativa global *rel.wo2* se obtiene para la combinación UdxR e Inserciones (Figura 5.9).

**Figura 5.9** *rel.wo2*, para intercambio de elementos e inserción de segmentos



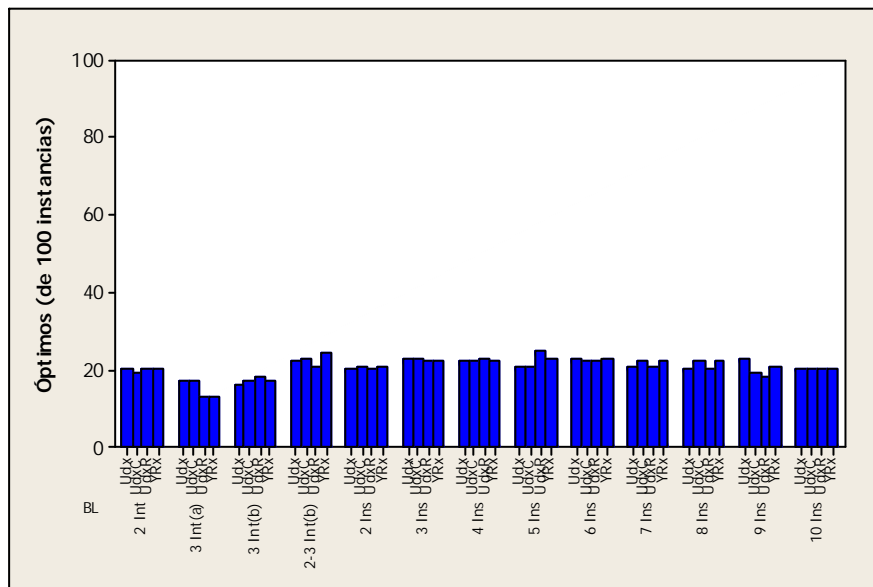
El promedio de óptimos alcanzados confirmados con el valor de *lbw* es de 20.56 instancias sobre el total. Los mejores promedios se obtienen con 2-3Int(b), 3 Ins, 4 Ins, 5 Ins y 6 Ins. Los promedios de óptimos alcanzados por procedimiento inicial no muestran diferencias significativas (entre 20% y 21%).

**Tabla 5.6** Óptimos conformados con valor de lbw

Óptimos		Solución inicial				Promedio
		Udx	UdxC	UdxR	YRx	
BL	Inicial	2	2	2	1	1.75
Intercambios	2Int	20	19	20	20	19.75
	3Int(a)	17	17	13	13	15.00
	3Int(b)	16	17	18	17	17.00
	2-3Int(b)	22	23	21	24	22.50
Inserciones	2 Ins	20	21	20	21	20.50
	3 Ins	23	23	22	22	22.75
	4 Ins	22	22	23	22	22.25
	5 Ins	21	21	25	23	22.50
	6 Ins	23	22	22	23	22.50
	7 Ins	21	22	21	22	21.50
	8 Ins	20	22	20	22	21.00
	9 Ins	23	19	18	21	20.25
	10 In	20	20	20	20	20.00

Los cantidad de óptimos alcanzados (Tabla 5.6 y Figura 5.10) por inserción de segmentos aparentemente crece hasta llegar al tamaño de segmento 5. Aquí llegan a obtenerse hasta 25 soluciones óptimas. Se cree esta detención en las mejoras conforme aumenta el tamaño del segmento se debe a que en las instancias pequeñas (las más pequeñas son 4 instancias de tamaño de secuencia  $T = 10$ ) un tamaño de segmento mayor a la mitad de del tamaño de la secuencia  $T$ , difícilmente encuentra mejores soluciones en el vecindario.

**Figura 5.10** Óptimos conformados con valor de lbw



## 5.5 Resultados

En éste capítulo se toman los resultados de los cuatro procedimientos constructivos del capítulo cuatro, Udx, UdxC, UdxR y YRx, los cuales se resuelve una variante del problema de secuencias en una línea de producción que tiene por objetivo minimizar la sobrecarga de trabajo o trabajo perdido.

Estos procedimientos se aplicaron a la batería de 100 problemas de Scholl et al. (1998). Dichos resultados se mejoran con procedimientos de BL. Se prueban dos tipos de vecindarios en la BL: intercambio de elementos e inserción de segmentos. Se consideran cuatro alternativas en el intercambio de elementos, las cuales se han denominado 2Int, 3Int(a), 3Int(b) y 2-3Int(b). En la inserción de segmentos se prueban tamaños de segmento desde 2 hasta 10. Para evaluar la calidad de las soluciones se usa el mismo índice de desviación relativa global (*rel.wo*) usado en el capítulo anterior.

Se impuso límite en el tiempo de búsqueda de los óptimos locales. El límite de tiempo para los procedimientos de BL con intercambio de elemento fue de 3600 segundos. En el vecindario 2-3Int(b) se estableció el límite en 3600 segundos para la primera fase de búsqueda y 1800 segundos para la segunda fase. La BL con inserción de segmentos estuvo limitada a 1800 segundos de búsqueda.

De los procedimientos por intercambios, se obtienen mejores resultados con intercambios de dos elementos (2Int) en tiempo relativamente bueno. El óptimo local de 2Int puede mejorarse con 3Int(b), con un esfuerzo computacional mayor debido a que el tamaño del vecindario es demasiado grande. La BL con inserción de segmentos obtiene óptimos locales tan buenos o mejores que el vecindario 2Int, pero requiere la mitad de tiempo de búsqueda.

Los vecindarios por inserción de segmentos obtienen mejores soluciones que los intercambios de elementos en un tiempo de búsqueda menor. Los tamaños de segmento que obtienen mejores resultados en la inserción son entre 2 y 6 elementos. Sea cual fuere la solución inicial, con dichos tamaños de segmentos se obtienen poco más de 20% de óptimos confirmados con valor de *lbw*.

Los mejores valores para el índice de desviación relativa global *rel.wo2* se obtienen cuando se combinan la solución inicial del procedimiento UdxR y las inserciones de segmentos de tamaño 10. Esto sugiere una relación entre el tamaño de segmento y el tamaño de la secuencia lo cual se considerada como una extensión de este trabajo.

El índice *rel.wo1* de los procedimientos constructivos se reduce a la mitad con las mejores alternativas consideradas de BL. Se contempla extender el trabajo con vecindarios más pequeños para el problema, que además mejoren las soluciones encontradas con los vecindarios presentados.



---

## Procedimiento constructivo por combinación de reglas

---

En este capítulo se explora otra alternativa del campo de las heurísticas para resolver el problema de secuenciación de productos mixtos. Esta vez se usa un procedimiento que combina soluciones representadas en el espacio de reglas heurísticas. Por una parte se intenta aprovechar algunas propiedades de las heurísticas tipo *greedy* que se basan en el uso de reglas de prioridad, y por la otra las ventajas que ofrecen las heurísticas de búsqueda por entornos. Se plantea la aplicación de un método de búsqueda por entornos conjuntamente con otro método heurístico basado con las reglas de prioridad. Se realiza una prueba computacional para probar la eficiencia del procedimiento.

### 6.1 Reglas de prioridad y búsqueda por entornos

Las reglas de prioridad son un elemento común en procedimientos heurísticos. Las reglas sirven para establecer una ordenación de tareas o productos en cada iteración con el propósito de seleccionar la mejor opción según la regla, entre un conjunto de candidatos compatibles con la parte de la solución construida.

Normalmente, este tipo de heurísticas tienen asociada una sola regla la cual determina la tarea o producto que debe lanzarse en cada etapa. Un algoritmo heurístico ofrece soluciones más aceptables cuanto mayor sean los aspectos que combina la regla. Sin embargo, no puede concluirse que una regla en particular supere a las demás ante cualquier ejemplar. Además, salvo que se incorpore el azar, el procedimiento siempre ofrecerá la misma solución.

Los procedimientos exactos para el problema de secuenciación que se tienen en cuenta la sobrecarga proporcionan buenos resultados sólo con instancias pequeñas debido a que el problema es de tipo *NP-hard*. (Tsai, 1995; Scholl et al., 1998; Kotani et al., 2004). Para la resolución de problemas con dimensiones como los que se encuentran en la industria, las heurísticas brindan soluciones aceptables. Normalmente estas heurísticas son *greedy*, las cuales se apoyan en reglas de prioridad para determinar la secuencia de productos, como las descritas en los capítulos 3 y 4 de esta tesis. Dichas reglas pueden considerar diversos aspectos del problema como la producción pendiente de un producto, su demanda inicial o los tiempos de procesamiento. También se usan reglas que relacionan diferentes aspectos del problema. Las heurísticas que usan una sola regla ya obtienen soluciones aceptables y cuando varias reglas se combinan el resultado suele ser mejor. Sin embargo, como ya se mencionó, no se puede concluir que alguna regla sea mejor que las otras para cualquier instancia.

Otro tipo de heurísticas son aquellas que realizan búsqueda por entornos. Las más populares son *Simulated Annealing* (SA), *Genetic Algorithms* (GA), *Tabu Search* (TS), *Greedy Randomized Adaptive Search Procedure* (GRASP), *Hill Climbing* (HC), *Scatter*

*Search* (SS) o *Path Relinking* (PR). Osman y Laporte (1996) presentan un amplio compendio bibliográfico de Metaheurísticas. Algunos trabajos que exploran la búsqueda por entornos y que toman en cuenta el criterio de las cargas de trabajo en la secuenciación son los siguientes: Smith et al. (1996) aplican redes neuronales al problema, Bolat (1997a) aplica SA para minimizar el trabajo incompleto, Xiaobo y Ohno (1997) tienen como objetivo la minimización del tiempo total de paro de línea y resuelven el problema con SA, Hyun et al. (1998) aplican GA, y Scholl et al. (1998) aplican TS.

Los métodos de búsqueda por entornos parten de una o varias soluciones iniciales, las cuales son alteradas iterativamente para llegar a otras soluciones de su entorno. Para ir de una solución  $s$  a otra parecida  $s'$  dentro de su entorno  $N(s)$  se realiza una operación elemental llamada “movimiento” sobre  $s$ .  $s' \in N(s)$ .

Una desventaja de este tipo de heurísticas es que no consideran el conocimiento del problema en particular como lo hacen las heurísticas *greedy*, ya que el vecindario o entorno de búsqueda es definido de forma general.

Como se explica en Laguna y Martí (2003), Martí (2006), o Martí, Laguna y Glover (2006), SS tiene su arranque en los años 60's, con la combinación de reglas en el área de programación de actividades<sup>19</sup> para obtener decisiones locales mejoradas, pero fue hasta 1977 que Fred Glover introdujo la heurística en una conferencia en Austin Texas. Combinaciones de reglas penalizadas numéricamente y estructuradas de forma conveniente para que sus evaluaciones expresaran una métrica común, generaban nuevas reglas. El enfoque nació bajo la conjetura de que la información acerca de que la ventaja relativa de diferentes alternativas puede capturarse en diferentes formas por diferentes reglas. La estrategia de combinación se ideó con la creencia de que esa información puede explotarse de forma más efectiva cuando ésta se considera en conjunto que cuando se hace de manera aislada. En general, las reglas de decisión creadas con estas estrategias de combinación producen mejores resultados empíricos que las aplicaciones estándar de reglas de decisión local.

El trabajo que se describe en este capítulo combina las reglas de prioridad aplicables en heurísticas *greedy* y las heurísticas de búsqueda por entornos. Con ello se busca aprovechar el conocimiento del problema mediante el uso de reglas de prioridad, y la generación de nuevas soluciones con la exploración de entornos.

### 6.2 Procedimiento constructivo por combinación de reglas

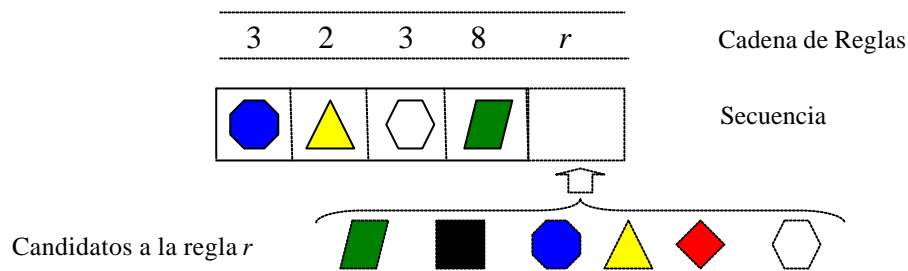
El procedimiento constructivo por combinación de reglas (PCCR) es un procedimiento *greedy* que se basa en el uso de reglas de prioridad. En el problema que se estudia en esta tesis, las reglas de prioridad condiciona la asignación de un producto en cierta etapa de la secuencia (Figura 6.1).

Las reglas que se usan en este trabajo consideran demandas pendientes, tiempos de procesamiento, regularidad en la producción, tiempos de procesamiento en la estación cuello de botella, tiempos de ocio y por supuesto, sobrecargas.

---

<sup>19</sup> Scheduling

Figura 6.1 Procedimiento constructivo



Un conjunto de reglas de prioridad o cadena de reglas  $R = \{r_1, r_2, r_3, \dots, r_R\}$  establece el orden de los productos en una secuencia. La cadena tendrá tantas reglas como posiciones tiene una secuencia solución ( $T$ ). La regla  $r$  en la posición  $t$  de la cadena, determina entre un conjunto de candidatos, el producto  $i$  que mejor satisface la regla  $r$ . El Algoritmo 6.1 indica los pasos del procedimiento constructivo por combinación de reglas.

#### Algoritmo 6.1 PCCR

0. Inicialización

**for** ( $t = 1$  to  $T$ )

1. Calcular para cada producto el valor de la aplicación de la regla  $r$ .
2. Seleccionar el mejor producto  $i$  para la regla  $r$ .
3. Asignar el producto  $i$  en al etapa  $t$  de la secuencia.
4. Actualizar demanda pendiente de  $i$ .

**end for**

### 6.3 Búsqueda por entornos y PCCR

Como ya se explicó, la idea del procedimiento propuesto es fusionar la búsqueda por entornos y las reglas de prioridad. Un algoritmo evolutivo crea nuevos elementos combinando un grupo de elementos actual, mejorando así el criterio con el que se miden los elementos. En el procedimiento propuesto (HH1), en vez de que los elementos del conjunto sean soluciones, se usan cadenas de reglas de prioridad. Para evaluar dichas cadenas se aplican el PCCR para medir la sobrecarga de las cadenas.

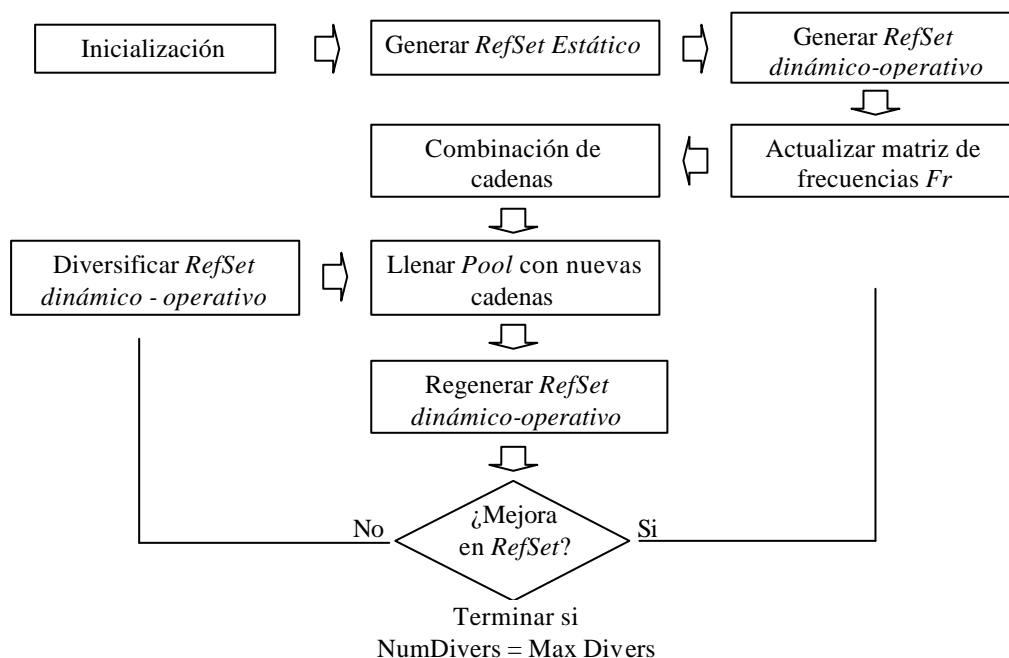
Dado que SS es toda una metodología, el procedimiento HH1 que se describe en este capítulo no pretende ser una simple sustitución de los elementos del *RefSet* por cadenas de reglas. Este trabajo es una propuesta inicial para combinar búsqueda por entornos con reglas de prioridad y visualizar su desempeño ante el problema de secuenciación en líneas de montaje bajo el criterio de minimización de sobrecarga. Es por ello que el procedimiento HH1 que se explica a continuación difiere en algunos aspectos de los

procedimientos de SS (Laguna y Martí, 2003; Laguna y Martí, 2005; González y Díaz, 2006).

En la Figura 6.2 se muestra un esquema con los elementos principales del procedimiento HH1. Así como SS opera sobre un conjunto de referencia (*RefSet*) de soluciones, HH1 opera sobre un *RefSet* de cadenas de reglas. En la fase inicial del procedimiento se crea el *RefSet* con cadenas de reglas de prioridad. Si bien, el conjunto de referencia puede crearse de forma determinista, en HH1 el conjunto de referencia se crea de forma aleatoria. Una vez que está lleno el conjunto de referencia se calcula la matriz de frecuencias del *RefSet*, y se actualiza en cada iteración del procedimiento. La matriz de frecuencias *Fr* es un elemento importante, ya que contiene información sobre la frecuencia de aparición de las reglas de prioridad en las cadenas del *RefSet*.

El siguiente paso es la combinación de las cadenas del *RefSet*. La combinación se realiza considerando la información contenida en la matriz de frecuencias (*Fr*). A diferencia de SS, HH1 no aplica un procedimiento de mejora sobre las nuevas cadenas. Al conjunto de cadenas generadas se le denomina *pool*. La regeneración del conjunto de referencia es el siguiente paso. En la regeneración se selecciona un conjunto de cadenas *elite*, considerando las cadenas existentes en el *RefSet* y en el *pool*. Ese grupo de cadenas *elite* se introducen en el *RefSet*, lo cual mejorará su “estado” hasta que después de varias iteraciones no mejore más. Entonces se procede a la diversificación, la cual también se realiza en función de la matriz de frecuencias. En la sección 6.6 se describe lo que se ha considerado como estado del *RefSet*.

Figura 6.2 Esquema básico SS & PCCR



El tamaño del *RefSet* típico en SS tiene un tamaño de 20 soluciones o menos, mientras que para PCCR el tamaño estará en función del número de reglas de prioridad (*R*) consideradas.

**Algoritmo 6.2** Pseudo código SS & PCCR

---

```

Inicializar
  0.1 Crear el RefSet estático y dinámico.
  0.2 Inicializar matriz de frecuencias Fr.
while ( no se exceda Max Diversificaciones )
  1. Combinar reglas del RefSet.
  2. Regenerar el RefSet.
  if ( no se mejora es estado del RefSet )
    3. Diversificar RefSet.
  end if
end while

```

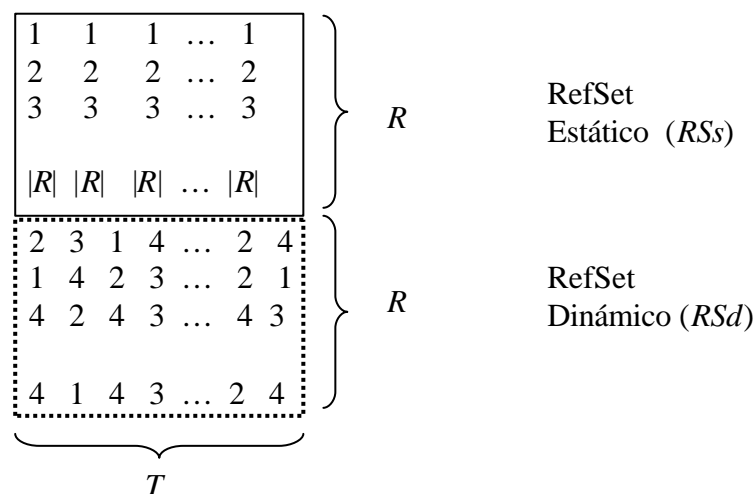
---

En las sub-secciones posteriores se describe con más detalle cada uno de los elementos del procedimiento HH1.

## 6.4 Conjunto de referencia *RefSet*

El conjunto de referencia se conforma de dos partes (*tiers*): el subconjunto estático (*RSs*) y el subconjunto dinámico u operativo (*RSd*). La Figura 6.2 representa ambos subconjuntos del *RefSet*.

**Figura 6.3** Tiers del conjunto de referencia



*RSs* contiene tantas cadenas de reglas como  $|R|$ , donde  $R$  es el número de reglas consideradas. Se le denomina “estático” porque no cambia durante el proceso de búsqueda. En el *RSs*, la cadena de reglas 1 ( $cr_1$ ) contiene sólo la regla 1, la cadena 2 ( $cr_2$ ) contiene sólo la regla 2, y así sucesivamente hasta completar el *RSs*. De esta manera se asegura que en la fase de combinación de cadenas todas las reglas sean consideradas.

$$cr_r = (r, r, r, \dots, r) \rightarrow RSs = \{cr_r : r \in R\} \quad (6.1)$$

El *RefSet dinámico* (*RSd*) cambia en cada iteración del procedimiento. Al inicio del procedimiento, las cadenas del *RSd* se generan de forma aleatoria. *RSd* contiene el grupo *elite* de cadenas de reglas, el cual se actualiza en cada iteración teniendo en cuenta las nuevas cadenas con mejores soluciones obtenidas en cada fase de combinación.

$$1 \leq RSd_{rt} \leq R \quad (6.2)$$

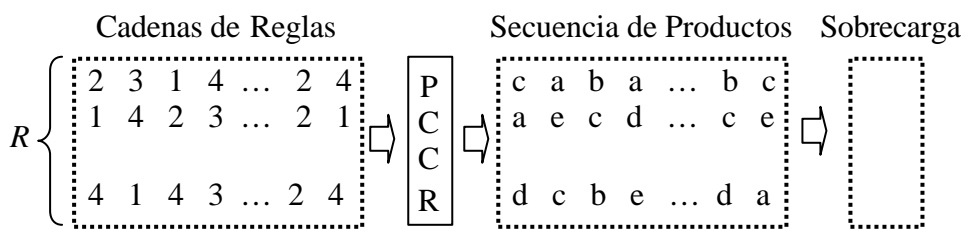
En la inicialización del procedimiento, el *RSd* es generado de manera aleatoria con las reglas de prioridad que se están considerando (*R*). La longitud de una cadena corresponde al número total de productos (*T*). El número de cadenas que conforma el *RSd* se ha establecido en *R*.

### 6.4.1 Reglas de prioridad

Como se ha mencionado, los elementos del conjunto de referencia son cadenas de reglas de prioridad. Para obtener el valor de la solución de una cadena de reglas, se aplica el PCCR de la sección 6.1. Es decir, se obtiene la secuencia de productos equivalente a tal cadena y se mide la sobrecarga de dicha secuencia (Figura 6.3).

En esta tesis se han considerado 20 reglas de prioridad. Si en la aplicación de una regla existe un empate en la selección del producto que mejor satisface la regla, (entre dos o más de los productos candidatos), se procede a desempatar de la siguiente forma: se toman únicamente los productos empatados y se aplican las reglas en orden numérico hasta romper el empate y se tiene un único producto para la asignación.

**Figura 6.4** Evaluación de las cadenas de reglas



En las reglas de prioridad se han considerado diferentes aspectos del problema que creemos que pueden aprovecharse. Algunas reglas consideran aspectos de forma individual y otras combinan varios de ellos. De hecho, otros criterios de secuenciación se consideran también en algunas de las reglas.

Las reglas 1-4 consideran los tiempos de procesamiento para seleccionar uno de los productos: el mayor, el menor, el más cercano al promedio y el más cercano al tiempo de ciclo. Las reglas 5 y 6 seleccionan el producto a partir de la demanda pendiente. Las reglas 7, 8, 14 y 15 discriminan los productos relacionando la demanda pendiente con la diferencia absoluta entre el tiempo de procesamiento de cada producto en cada estación y

el tiempo de ciclo. El desplazamiento que realizan los trabajadores de todas las estaciones debido a la asignación de un producto determinado se consideran en las reglas 9 y 10. Las reglas 11 y 12 toman en cuenta los tiempos de procesamiento sólo en la estación cuello de botella. La sobrecarga se considera en las reglas 16 y 18. Las reglas 17 y 19 priorizan los productos a partir del tiempo improductivo que puede ocasionar. Las reglas 13 y 20 toman en cuenta el concepto de regularidad, la primera respecto a la producción ideal y la segunda respecto a la sobrecarga ideal que se obtiene a partir de una cota inferior de sobrecarga ( $lbw$ ). Una lista detallada de las 20 reglas de prioridad puede encontrarse en el Anexo 3.

La estación cuello de botella es aquella que tiene más trabajo total. Al inicializar el procedimiento se determina cuál es esta estación. La carga total de una estación se ha determinado sumando el producto de los tiempos de procesamiento de cada producto en cada estación por la demanda de cada producto.

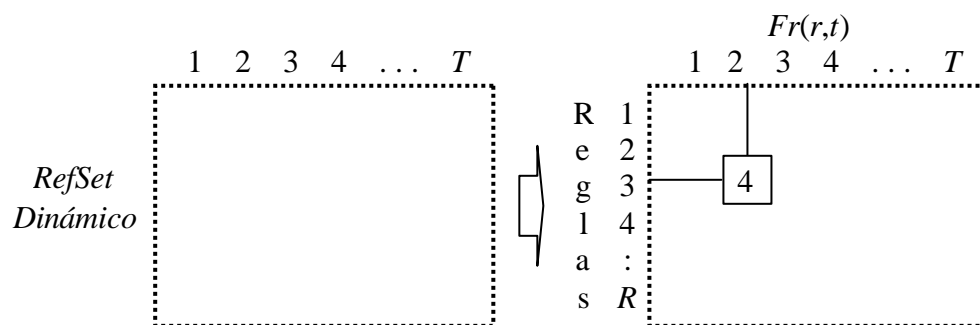
La tasa ideal de producción del producto  $i$  por etapa se determina dividiendo la demanda de  $i$  entre el total de productos a secuenciar ( $T$ ). Esta tasa también se establece al inicializar el procedimiento.

De manera similar, la sobrecarga ideal por etapa se calcula dividiendo el valor de  $lbw$  entre  $T$ . Nótese que si  $lbw = 0$ , la aplicación de la regla número 20 no permitirá la asignación de productos que ocasionen sobrecarga.

#### 6.4.2 Matriz de frecuencias

La matriz de frecuencias es una parte importante en el procedimiento. En esta matriz se basa el procedimiento de combinación de las cadenas de reglas del *RefSet*. Como se explica en posteriormente,  $Fr$  también se usa en la fase de diversificación. Es necesario actualizar la matriz de frecuencias en cada iteración.

Figura 6.5 Matriz de frecuencias



$Fr(r,t)$  contiene el número de veces que una regla determinada  $r$  aparece en la posición  $t$  de las cadenas de reglas que contiene el *RSd*. La Figura 6.5 indica que si se revisan todas las reglas que aparecen en la posición número 2 de todas las cadenas del *RSd*, se encontrará que la regla número 3 aparece un total de 4 veces. Dado que el *RSs* no se modifica, no es necesario considerarlo para calcular  $Fr$ .

## 6.5 Combinación de cadenas del conjunto de referencia

La combinación de las cadenas del conjunto de referencia se realiza mediante el re-encadenamiento de trayectorias ya que proporciona la integración de las estrategias de intensificación y diversificación (Glover, 1989; Díaz et al., 1996). Este enfoque se basa en la exploración de las trayectorias que conectan soluciones *elite* para generar nuevos elementos. Así, partiendo de soluciones “iniciales” se generan trayectorias en el espacio de entornos, lo cual conduce a otros elementos “guías”.

Desde un punto de vista espacial, la generación de combinaciones lineales entre elementos de un conjunto *elite*, puede caracterizarse como la generación de caminos entre y más allá de estas soluciones. Este enfoque se apoya en que un camino proyectado entre soluciones de un espacio de vecindario producirá elementos que mantienen características importantes de los elementos “ascendentes”.

$$\begin{array}{l} cp = [ cp(1), cp(2), \dots, cp(T) ] \\ cq = [ cq(1), cq(2), \dots, cq(T) ] \end{array} \left. \vphantom{\begin{array}{l} cp \\ cq \end{array}} \right\} \text{PCCR descendiente } cr = [ cr(1), cr(2), cr(3), \dots, cr(T) ]$$

Bajo estas ideas, HH1 genera nuevos elementos. Mediante la combinación de dos cadenas ascendente (sean  $cp$  y  $cq$ ) se obtiene una nueva cadena “descendiente” (sea  $cr$ ). La regla que contendrá la cadena descendiente en la posición  $t$  ( $cr(t)$ ), será alguna de las reglas que contienen las cadenas ascendentes ( $cp(t)$  o  $cq(t)$ ) en la misma posición.

$$cr(t) = \begin{cases} cp(t) & \text{si } cp(t) = cq(t) \\ cp(t) & \text{si } Fr(cp(t),t) > Fr(cq(t),t) \\ cq(t) & \text{en otro caso} \end{cases} \quad (6.3)$$

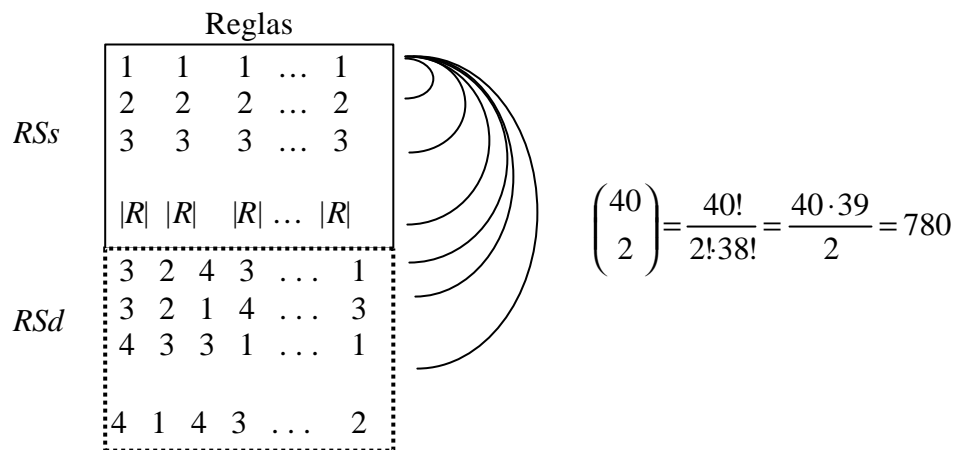
El elemento  $t$  de la cadena de reglas descendiente se genera de acuerdo a la frecuencia que esa regla  $r$  tiene en dicha posición de la matriz de frecuencias (6.3). Por ejemplo, si  $cp(t) = 4$  y  $cq(t) = 2$ , y  $Fr(4, t) = 5$  y  $Fr(2, t) = 8$ , en la posición  $t$  de la cadena descendiente se colocará la regla 2 porque en la columna  $t$  de la matriz de frecuencias  $Fr$  la regla 2 tiene mayor índice de frecuencia que la regla 4.

La idea detrás de las reglas anteriores para la creación de cadenas descendientes es que en el *RefSet* se tiene un conjunto de cadenas *elite*, y que la exploración de las reglas que aparecen con más frecuencia en cierta posición en éste conjunto de cadenas puede guiar a mejores resultados.

Dado que una cadena descendiente se obtiene a partir de dos cadenas ascendentes, se tendrán tantas cadenas nuevas como combinaciones por pares se pueden hacer de las cadenas del *RefSet*. Como en este trabajo se usan 20 reglas de prioridad, se tienen 20 cadenas del *RSs* y 20 cadenas del *RSd*, de tal forma que se obtienen 780 cadenas descendientes.

Una vez que se tiene este “*pool*” de cadenas descendientes, debe aplicarse el PCCR para obtener las secuencias de producto equivalentes y entonces calcular su valor de sobrecarga. Después de evaluar las cadenas del *pool*, se consideran aquellas cadenas con mejores valores de sobrecarga para actualizar el *RSd*. A este proceso se la ha llamado “regeneración del *RefSet*”.

**Figura 6.6** Combinación del *RefSet*



## 6.6 Regeneración del conjunto de referencia

Una vez que se han realizado todas las combinaciones y se tienen todas las cadenas descendientes se realiza la regeneración del *RefSet*, esto es, se consideran las cadenas de reglas del *pool* con mejores valores de sobrecarga para ser incorporadas en la parte operativa del conjunto de referencia (*RSd*). En este trabajo se han considerado tres alternativas de regeneración:

- 1) Considerando tanto el *RefSet* como el *pool*, se seleccionan las  $R$  mejores cadenas.
- 2) Se eliminan las cadenas del *RefSet* y se sustituyen por las  $R$  mejores cadenas del *pool*.
- 3) Se sustituyen las  $aR$  peores cadenas del *RefSet* por las  $aR$  mejores del *pool*.

Siempre se tiene guardada la mejor solución encontrada hasta el momento. En el proceso de regeneración del *RefSet* se evita duplicar las cadenas de reglas, por lo tanto, dentro del *RefSet* existirán cadenas con valor de la función objetivo diferentes.

Después de la regeneración del *RefSet*, se evalúa el estado del mismo. Para ello, en cada iteración se guardan los valores de sobrecarga de la mejor y de la peor cadena del *RefSet*. El estado del *RefSet* está en función de dichos valores. Para detectar los cambios de un *RefSet* al siguiente, únicamente se comparan dichos valores del conjunto de referencia anterior y los del actual. Se considera que el estado del *RefSet* ha mejorado si al menos en una de las cadenas de referencia (la mejor o la peor) ha habido una mejora en el valor de la sobrecarga.

## 6.7 Diversificación

Cuando la combinación de cadenas deja de producir mejoras en el estado del conjunto de referencia se pasa a la fase de diversificación. Para ello se aprovecha la información que contiene la matriz de frecuencias  $Fr(r,t)$ . La diversificación se realiza en dos fases. En la primera se generan nuevas cadenas con un criterio de combinación diferente al usado en la combinación normal. En la segunda fase se seleccionan de forma iterativa las cadenas diversificadas del *pool* que son más diferentes de las cadenas contenidas ya en el *RefSet*.

$$\begin{array}{l} cp = [ cp(1), cp(2), \dots, cp(T) ] \\ cq = [ cq(1), cq(2), \dots, cq(T) ] \end{array} \begin{array}{l} \searrow \\ \nearrow \end{array} \text{PCCR descendiente } cr = [ cr(1), cr(2), cr(3), \dots, cr(T) ]$$

La matriz de frecuencias contiene la información para saber cuales reglas aparecen con mayor frecuencia en el *RSd*, pero también se puede saber cuales aparecen menos frecuencia. Por ello la fase numero 1 de la diversificación se realiza como si de una combinación se tratase. La diferencia radica en el criterio que se usa para determinar los elementos de la cadena descendiente. Dadas dos cadenas de reglas ascendientes  $cp$  y  $cq$  se obtiene una nueva cadena descendiente diversificada.

Al diversificar, la regla  $r$  en la posición  $t$  de la cadena descendiente  $cr$  se genera de acuerdo a (6.4).

$$cr(t) = \begin{cases} cp(t) & \text{si } cp(t) = cq(t) \\ cp(t) & \text{si } Fr(cp(t),t) < Fr(cq(t),t) \\ cq(t) & \text{en otro caso} \end{cases} \quad (6.4)$$

A diferencia de la combinación normal de cadenas de reglas (sección 6.5), en la posición  $t$  de la nueva cadena  $cr$ , se asignará la regla de la cadena ascendiente ( $cp$  o  $cq$ ) que tenga menor frecuencia en la matriz de frecuencias  $Fr(r,t)$ . Con ello se pretende que las reglas que aparecen con menor frecuencia en el *RefSet* dirijan la búsqueda hacia espacios no explorados del entorno.

### Algoritmo 6.3 Diversificación del RefSet

---

**Inicializar**

0.1 Limpiar el *RefSet* dinámico.

0.2 Generar *pool* con criterio de diversificación.

0.3 Meter al *RefSet* la cadena de reglas del *pool* con mejor valor.

**for** ( $t = 2$  **to**  $t = R$ )

1. Buscar cadena  $cr$  del *pool* con menos coincidencias respecto al *RefSet*.

2.  $RefSet \leftarrow cr$ .

**end for**

---

Para llenar el *RefSet* con las nuevas cadenas diversas (segunda fase), se buscan en el *pool* “diverso” aquellas que sean diferentes entre si. El proceso de diversificación inicia eliminando todas las cadenas del *RefSet* e insertando en él la cadena del *pool* diverso con mejor valor objetivo. De forma iterativa se busca en el *pool* diverso la cadena que sea más diferente a las existentes hasta entonces en el *RefSet*, hasta completar las  $R$  cadenas de reglas del *RSd*.

El grado de diferenciación entre dos cadenas se mide con el número de coincidencias que existe entre ambas cadenas. Existe una coincidencia si en la misma posición  $t$  de ambas cadenas se encuentra la misma regla  $r$ .

En la diversificación se buscan aquellas cadenas con el menor número de coincidencias. El paso 1.1 del Algoritmo 6.3 implica que la cadena  $cr$  será comparada con cada una de las cadenas de reglas existentes en ese instante en el *RefSet*. El número total de coincidencias de una cadena  $cr$  respecto al *RefSet* será la suma de las coincidencias de  $cr$  con la primera cadena del *RefSet* mas las coincidencias de  $cr$  con la segunda cadena del *RefSet*, y así sucesivamente. Después de comparar todas las cadenas del *pool*, se introduce al *RefSet* aquella  $cr$  con el menor número de coincidencias respecto al *RefSet*.

## 6.8 Experiencia computacional

Para probar el desempeño del procedimiento descrito en este capítulo se realiza una experiencia computacional. Nuevamente, las pruebas se realizan con las 100 instancias diseñadas por Scholl et al. (1998) y se usan los mismos índices (desviación relativa global) descritos en capítulos anteriores para medir la calidad de las soluciones obtenidas con el procedimiento.

En este capítulo se usan tres variantes de *rel.wo*. El índice usado por Scholl equivale al índice *rel.wo2*. Así, en:

<i>rel.wo1</i>	$wo_h^*$ es la cota inferior de la sobrecarga,
<i>rel.wo2</i>	$wo_h^*$ es la mejor solución encontrada con HH1,
<i>rel.wo3</i>	$wo_h^*$ es la mejor solución encontrada, considerando HH1 y las soluciones encontradas con CPLEX durante 15 minutos de búsqueda,
<i>rel.wo4</i>	$wo_h^*$ es la mejor solución encontrada considerando CPLEX, HH1 y las mejores soluciones encontradas con la búsqueda local del capítulo 5.

Cuando se hace mención a los resultados de CPLEX, siempre se refiere al resultado alcanzado en 15 minutos de búsqueda.

Las pruebas se realizaron en un ordenador Pentium 4 CPU 2.4 GHz, 512 MB RAM, con sistema operativo Microsoft Windows XP Professional 2002. HH1 ha sido implementado en lenguaje de programación C.

La experiencia compara las tres alternativas de generación descritas en la sección 6.6. Se hace referencia a cada alternativa con Reg1, Reg2 y Reg3. El tipo de regeneración Reg1 selecciona las  $R$  mejores cadenas existentes en el *RefSet* y en el *pool*. Reg2 toma las  $R$  mejores cadenas del *pool*. Reg3 sustituye las  $aR$  peores cadenas del *RefSet* por las  $aR$  mejores cadenas del *pool*.

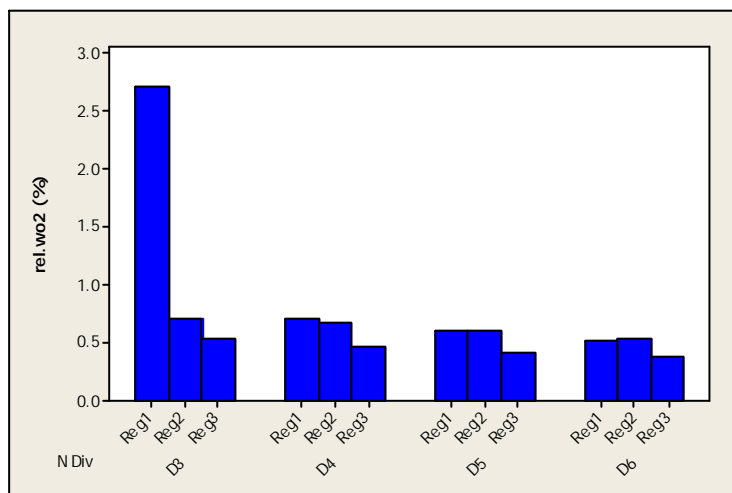
**Tabla 6.1** Desviación relativa global

N Div	<i>rel.wo1</i> (%)			<i>rel.wo2</i> (%)			<i>rel.wo3</i> (%)			<i>rel.wo4</i> (%)		
	Reg1	Reg2	Reg3	Reg1	Reg2	Reg3	Reg1	Reg2	Reg3	Reg1	Reg2	Reg3
D3	42.81	42.79	42.54	2.71	0.70	0.53	1.85	1.84	1.66	17.78	17.77	17.56
D4	42.72	42.74	42.45	0.71	0.66	0.46	1.78	1.80	1.60	17.71	17.72	17.49
D5	42.64	42.64	42.37	0.59	0.60	0.41	1.73	1.73	1.54	17.64	17.64	17.42
D6	42.53	42.55	42.33	0.52	0.53	0.38	1.65	1.66	1.51	17.33	17.57	17.39

La Tabla 6.1 contiene los diferentes índices de desviación relativa global *rel.wo1* para las diferentes alternativas de regeneración. La columna de la izquierda de esta tabla indica el número de diversificaciones (N Div) que se realizaron para obtener el valor de desviación relativa mostrado para cada tipo de regeneración. En la experiencia computacional sólo se realizó búsqueda hasta la sexta diversificación ya que por los resultados obtenidos hasta entonces, no se habría mejoras significativas en los resultados al aumentar el número de diversificaciones. El criterio de parada del procedimiento ha sido el número de diversificaciones.

En general, los resultados son muy similares dentro de cada tipo de desviación relativa global. Para el *rel.wo1* no hay diferencias significativas entre tipos de regeneración y número de diversificaciones. En promedio, las soluciones encontradas para *rel.wo1* están desviadas un 42.59% de la cota inferior de sobrecarga *lbw*. Los resultados para el índice *rel.wo1* en HH1 son ligeramente peores que los resultados para el mismo índice de los dos mejores procedimientos constructivos (UdxC y UdXR, 41.82% y 40.97% respectivamente) descritos en el capítulo 4.

**Figura 6.7** *rel.wo2* para tipo de regeneración

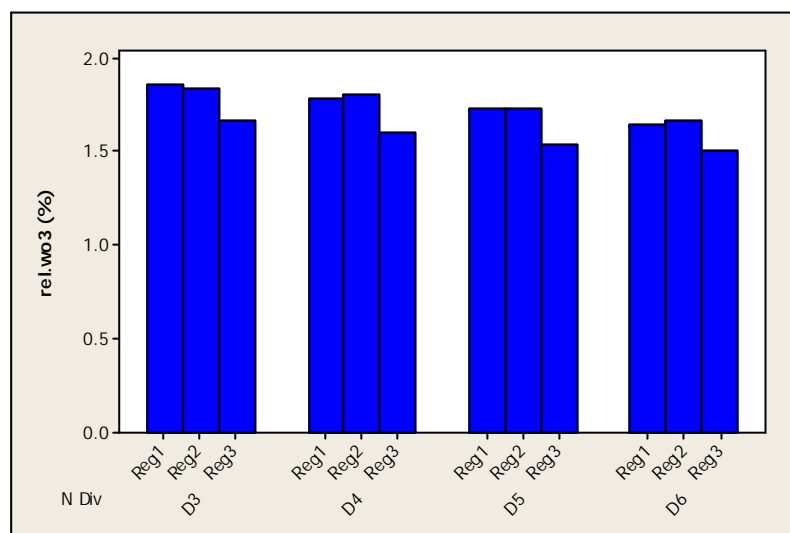


Los índices de desviación relativa global permiten detectar que la alternativa de regeneración Reg3 ofrece resultados ligeramente mejores que los otros dos procedimientos de regeneración. El tipo de regeneración Reg3 es aquella en la que cierto número ( $aR$ ) de las peores cadenas del *RefSet* son sustituidas por el mismo número de las mejores cadenas del *pool*. En la experiencia computacional el valor de  $a$  para Reg3 se estableció en 0.5.

Sin embargo, si se observan los valores de *rel.wo2*, (Figura 6.7), debido a que la mejora porcentual en cada diversificación para Reg3 disminuye (0.07, 0.05 y 0.03 entre diversificaciones), puede pensarse que si se aumenta el número de diversificaciones, el tipo de regeneración Reg1, podría igualar los resultados de Reg3 (2.00, 0.12 y 0.07), y si las mejoras en cada diversificación continúan aumentando en el tipo de regeneración Reg2 (0.04, 0.06 y 0.07), ésta también podría igualar o mejorar a Reg3. Si bien, las mejoras en Reg2 son, hasta D6, menores a 0.1%.

El índice *rel.wo3* permite comparar los resultados de HH1 con los de CPLEX después de realizar búsqueda del óptimo durante 15 minutos. HH1 se desvía en menos de dos puntos porcentuales de la soluciones obtenidas por CPLEX. HH1 ofrece soluciones casi tan buenas como las encontradas con CPLEX, pero en una quinta parte de tiempo en promedio.

**Figura 6.8** *rel.wo3* global para tipo de regeneración



En el 52% de las instancias, HH1 mejora el resultado de CPLEX. En 11 instancias HH1 obtiene el mismo resultado que CPLEX. En los grupos de instancias 1 y 2, HH1 no mejoró a CPLEX, a pesar de ser instancias pequeñas con 5 productos, 5 estaciones y longitud máxima de secuencia de 30 productos.

El grupo tres es el mayor con 30 instancias. En 16 de ellas (53.3%) HH1 mejora el resultado de CPLEX.

Los grupos 4 y 5 tienen 10 instancias cada uno. El grupo 4 tiene 10 estaciones y 20 productos, mientras que el segundo tiene 20 estaciones y 10 productos. HH1 obtiene mejores sobrecargas que CPLEX para 8 instancias en cada grupo.

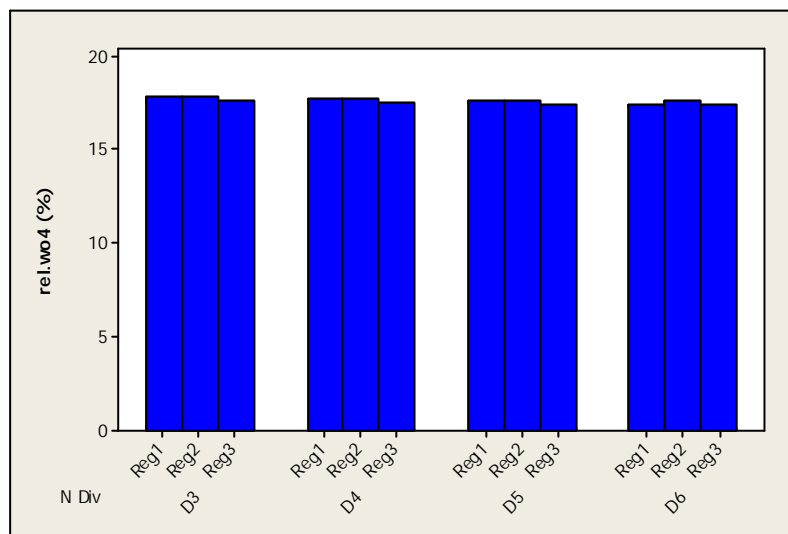
El grupo 6 considera 20 estaciones y 20 productos en las instancias. Para ambos parámetros se tienen las cantidades máximas consideradas. En las 10 instancias del grupo, HH1 encuentra menores soluciones que CPLEX, lo cual puede atribuirse a la complejidad del problema.

HH1 mejora los resultados de CPLEX únicamente para dos instancias de 10 en el grupo 7 que considera el mismo número de estaciones y productos (10) y tiene longitudes de ventanas diferentes. En el grupo 8, con ventanas amplias, CPLEX ofrece mejores resultados que HH1 en todas las instancias del grupo. El grupo 9 representa la existencia de un modelo básico con diferentes opciones especiales. Para dicho grupo HH1 consigue mejores resultados que HH1 en 8 de 10 instancias.

A grosso modo, al comparar HH1 con CPLEX, de acuerdo a la experiencia puede decirse que para las instancias pequeñas (grupos 1 y 2), las instancias de tamaño medio con longitudes de ventanas diferentes (grupo 7) y las instancias de tamaño medio con ventanas amplias (grupo 8), CPLEX encuentra mejores soluciones que HH1.

El índice *rel.wo4* de la Tabla 6.1 compara los resultados de HH1 con los mejores resultados obtenidos con los procedimientos de mejora local del capítulo anterior, con los procedimientos constructivos y con CPLEX. HH1 está a 17-18 puntos de desviación relativa global de las mejores soluciones encontradas por cualquiera de los procedimientos (Figura 6.9).

**Figura 6.9** rel.wo4 global para tipo de regeneración



En 44% de las instancias, HH1 obtiene mejores valores de sobrecarga que los procedimientos constructivos (en adelante PC) del capítulo 4.

En el grupo 1, para 4 de 5 instancias con  $I = 5$ ,  $K = 5$  y longitudes de estaciones iguales, se obtuvieron mejores resultados con HH1 que con PC. El grupo 2, con longitudes de estaciones diferentes (desde estrechas hasta amplias), HH1 mejora a PC en 4 de 5 instancias. El grupo 3 representa instancias de tamaño medio en cuanto al número de estaciones (todas estrechas) y productos, pero el tamaño de las secuencias varía entre 50 y 400 unidades. En 17 de las 30 unidades de éste grupo, HH1 mejora a PC. En el grupo 4, para 3 de las 10 instancias, HH1 mejora a PC. Igualmente sucede en el grupo 5. El grupo 6, con  $I = 20$  y  $K = 20$ , únicamente en una de 10 instancias HH1 mejora a PC. En el grupo 7,

con instancias de tamaño medio y estaciones de diferentes longitudes, sucede igual que con el grupo 4 y 5. Tanto para el grupo 8 como para el 9, el 50% de las veces HH1 encuentra mejor soluciones que PC.

Para todas las instancias de la batería, el mejor resultado considerando todos los procedimientos (PC, BL y HH1) fue encontrado con BL. HH1 coincide con el mejor valor de sobrecarga encontrado en 4 instancias del grupo 1, 4 del grupo 2, 3 del grupo 8 y 3 del grupo 9, en total, 14% de las instancias.

En lo referente a los tiempos de cómputo debe mencionarse que en la mejora local (inserción de segmentos) se ha realizado la búsqueda durante un máximo de 1800 segundos, mientras que HH1 ha requerido en promedio 220 segundos para buscar una solución en seis diversificaciones.

Cuando se usa el tipo de regeneración Reg3 se requiere en promedio 168 segundos para realizar seis diversificaciones y se obtienen resultados ligeramente mejores que Reg1 y Reg2. La Tabla 6.2 muestra el número de óptimos (según el valor de *lbw*) alcanzado después de las diversificaciones 3-6 para cada tipo de regeneración. La batería consta de 100 instancias. Hasta D6, únicamente Reg2 incrementa el número de óptimos alcanzados, lo cual aunado a la idea de que Reg2 mejora cada vez más, en cada diversificación, induce a pensar que dicha regeneración puede ser tan buena o mejor que Reg3 en cuanto a óptimos (*lbw*) alcanzados.

**Tabla 6.2** Óptimos alcanzados para tipo de regeneración del RefSet

<i>Óptimos (lbw)</i> N Div	Tipo de Regeneración		
	Reg1	Reg2	Reg3
D3	12	11	14
D4	12	11	14
D5	12	11	14
D6	12	13	14

Si se revisa la Tabla 5.6 del capítulo anterior, encontrará que con los procedimientos constructivos se alcanzó como máximo 2 resultados óptimos por procedimiento. Con la mejora local se alcanzaron hasta 25 soluciones óptimas. En una octava parte del tiempo que requiere la mejora local, HH1 puede encontrar poco más de la mitad de soluciones óptimas (confirmadas con *lbw*) alcanzadas con mejora local.

La Tabla 6.3 contiene los diferentes índices de desviaciones relativas por grupos de instancias. Para cada grupo se muestra el índice correspondiente al número de diversificaciones realizadas (D3,..., D6). En las columnas de la Tabla 6.3 se tienen los diferentes índices de desviación relativa usados en esta experiencia computacional. A su vez, se muestran los valores para cada estrategia de regeneración (Reg1,..., Reg3). Para cada grupo se ha obtenido el valor promedio de los resultados hasta cada una de las diversificaciones (Prom.). El índice respectivo de cada grupo ha sido calculado únicamente con los resultados de sobrecarga de las instancias del mismo grupo.

**Tabla 6.3** Desviación relativa para grupos

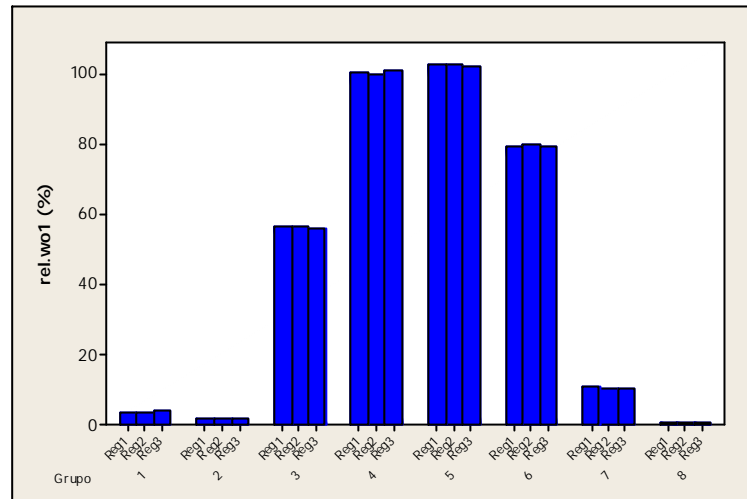
Grupo	<i>rel.wo1</i> (%)			<i>rel.wo2</i> (%)			<i>rel.wo3</i> (%)			<i>rel.wo4</i> (%)			
	Reg1	Reg2	Reg3	Reg1	Reg2	Reg3	Reg1	Reg2	Reg3	Reg1	Reg2	Reg3	
1	D3	3.16	3.16	3.61	0.00	0.00	0.44	0.33	0.33	0.77	0.33	0.33	0.77
	D4	3.16	3.16	3.61	0.00	0.00	0.44	0.33	0.33	0.77	0.33	0.33	0.77
	D5	3.16	3.16	3.61	0.00	0.00	0.44	0.33	0.33	0.77	0.33	0.33	0.77
	D6	3.16	3.16	3.61	0.00	0.00	0.44	0.33	0.33	0.77	0.33	0.33	0.77
	Prom.	3.16	3.16	3.61	0.00	0.00	0.44	0.33	0.33	0.77	0.33	0.33	0.77
2	D3	1.65	1.73	1.91	0.09	0.17	0.34	1.30	1.38	1.56	1.65	1.73	1.91
	D4	1.65	1.73	1.91	0.09	0.17	0.34	1.30	1.38	1.56	1.65	1.73	1.91
	D5	1.65	1.73	1.91	0.09	0.17	0.34	1.30	1.38	1.56	1.65	1.73	1.91
	D6	1.65	1.65	1.91	0.09	0.09	0.34	1.30	1.38	1.56	1.65	1.65	1.91
	Prom.	1.65	1.71	1.91	0.09	0.15	0.34	1.30	1.38	1.56	1.65	1.71	1.91
3	D3	56.65	56.71	55.95	1.12	1.16	0.67	2.04	2.08	1.59	23.74	23.78	23.19
	D4	56.45	56.56	55.87	0.99	1.06	0.62	1.91	1.98	1.54	23.58	23.67	23.13
	D5	56.29	56.42	55.72	0.89	0.97	0.52	1.81	1.89	1.43	23.45	23.56	23.00
	D6	56.19	56.27	55.49	0.82	0.88	0.37	1.74	1.80	1.28	23.38	23.44	22.82
	Prom.	56.395	56.490	55.757	0.955	1.017	0.545	1.875	1.937	1.460	23.537	23.612	23.035
4	D3	101.14	100.76	101.29	1.19	1.00	1.27	1.43	1.24	1.51	43.65	43.37	43.75
	D4	100.98	100.68	101.29	1.11	0.96	1.27	1.35	1.20	1.51	43.53	43.31	43.75
	D5	100.25	99.94	100.74	0.74	0.59	0.99	0.98	0.83	1.23	43.01	42.79	43.36
	D6	99.59	99.24	100.74	0.41	0.23	0.99	0.65	0.47	1.23	42.54	42.29	43.36
	Prom.	100.490	100.155	101.015	0.862	0.695	1.130	1.102	0.935	1.370	43.182	42.940	43.555
5	D3	102.87	103.11	102.35	0.84	0.96	0.59	1.74	1.86	1.48	23.05	23.19	22.73
	D4	102.67	103.03	102.35	0.74	0.92	0.58	1.64	1.82	1.48	22.93	23.14	22.73
	D5	102.67	103.03	102.26	0.74	0.92	0.54	1.64	1.82	1.44	22.93	23.14	22.68
	D6	102.65	103.03	102.26	0.74	0.92	0.54	1.63	1.82	1.44	22.92	23.14	22.68
	Prom.	102.715	103.05	102.305	0.765	0.930	0.562	1.662	1.830	1.460	22.957	23.152	22.705
6	D3	79.42	79.66	79.35	0.62	0.75	0.58	0.62	0.75	0.58	27.88	28.05	27.83
	D4	79.32	79.64	79.04	0.56	0.74	0.40	0.56	0.74	0.40	27.80	28.04	27.61
	D5	79.23	79.54	78.92	0.51	0.69	0.34	0.51	0.69	0.34	27.74	27.04	27.52
	D6	78.96	79.34	78.92	0.36	0.57	0.34	0.36	0.57	0.34	27.55	27.82	27.52
	Prom.	79.232	79.545	79.057	0.512	0.687	0.415	0.512	0.687	0.415	27.742	27.737	27.620
7	D3	10.80	10.58	10.61	0.60	0.41	0.43	2.69	2.49	2.52	7.85	7.64	7.66
	D4	10.79	10.53	10.53	0.60	0.35	0.36	2.69	2.44	2.45	7.84	7.58	7.59
	D5	10.78	10.49	10.53	0.58	0.32	0.36	2.67	2.40	2.45	7.83	7.54	7.59
	D6	10.71	10.49	10.53	0.52	0.32	0.36	2.61	2.40	2.45	7.76	7.54	7.59
	Prom.	10.770	10.525	10.550	0.575	0.350	0.377	2.665	2.432	2.467	7.820	7.575	7.067
8	D3	0.61	0.61	0.55	0.08	0.08	0.02	0.55	0.55	0.49	0.61	0.61	0.55
	D4	0.59	0.61	0.55	0.06	0.08	0.02	0.53	0.55	0.49	0.59	0.61	0.55
	D5	0.59	0.61	0.55	0.06	0.08	0.02	0.53	0.55	0.49	0.59	0.61	0.55
	D6	0.59	0.60	0.55	0.06	0.07	0.02	0.53	0.54	0.49	0.59	0.60	0.55
	Prom.	0.595	0.607	0.550	0.065	0.077	0.02	0.535	0.547	0.49	0.595	0.607	0.550
9	D3	394.98	390.71	391.65	2.70	1.82	2.01	31.24	30.11	30.35	121.95	120.04	120.46
	D4	394.86	390.71	389.54	2.68	1.82	1.58	31.21	30.11	29.80	121.90	120.04	119.51
	D5	394.63	387.44	389.54	2.63	1.14	1.58	31.14	29.24	29.80	121.79	118.57	119.51
	D6	394.63	387.44	389.54	2.63	1.14	1.58	31.14	29.24	29.80	121.79	118.57	119.51
	Prom.	394.77	389.08	390.07	2.660	1.480	1.687	31.182	29.675	29.937	121.85	119.03	119.74

La Figura 6.10 representa gráficamente los valores promedios del *rel.wo1* por grupos y por tipo de regeneración. Aunque los grupos 1 y 2 son similares en parámetros, el segundo, con ventanas diferentes, se acerca más a los valores de la cota *lbw* que el primero. En la Figura 6.11 puede apreciarse que Reg1 y Reg2 obtienen el mejor resultado de los tres tipos de regeneración para el grupo 1, y Reg1 es la mejor para el grupo 2.

Respecto a *lbw*, en el grupo 3, el promedio de desviación porcentual aumenta en promedio hasta el 56.214%. El mejor valor obtenido es 55.49%. En este sentido, HH1 ofrece resultados levemente mejores que el mejor procedimiento de PC (UdxR, 58.97%). Para los cuatro índices *rel.wo* de ésta experiencia, Reg3 ofrece los resultados más cercanos a *lbw* y Reg2 los más lejanos.

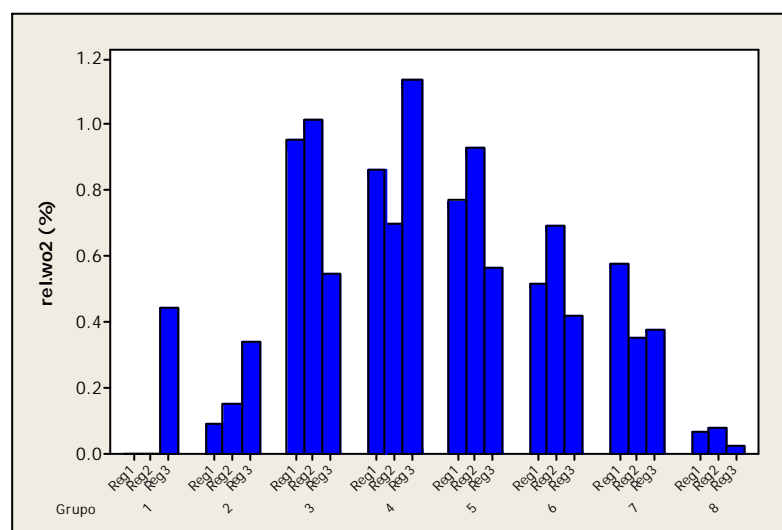
El grupo 4, está en promedio a 100.55% puntos porcentuales de desviación relativa respecto a *lbw*, y el mejor valor es 99.24%. Reg2 ofrece los mejores resultados para las cuatro desviaciones porcentuales analizadas. En ésta ocasión, PC ofrece mejores resultados (UdxR, 87.43%).

Figura 6.10 *rel.wo1* para grupos y tipo de regeneración



Similarmente, en el grupo 5 la desviación relativa respecto a *lbw* es en promedio 102.69%. UdxC en los PC ofrece 96.67% para el mismo índice mientras que Reg3 obtiene mejor resultado (102.26%) en HH1. Por otra parte, Reg3 es el mejor tipo de regeneración en HH1 en el grupo 6, no sólo para *rel.wo1* sino para los cuatro índices. En promedio obtiene 79.27%. El mejor valor en PC es 69.19% y 78.92% en HH1.

Figura 6.11 *rel.wo2* para grupos y tipo de regeneración



Reg2 obtiene los mejores valores para *rel.wo1-3* para el grupo 7. Para *rel.wo4* el mejor tipo de regeneración es Reg3. El promedio de *rel.wo1* es 10.61% y el mejor valor obtenido es

10.49%. El índice es mejorado con PC (UdxR, 8.86%). Reg3 es también la mejor regeneración para todos los índices del grupo con ventana amplia (grupo 8). El mejor *rel.wo1* del grupo es de 0.55% (promedio del grupo, 0.58%) en HH1 y el y en PC es 0.50%. Para el grupo con modelo básico, Reg3 se acerca más a *lbw*. Para *rel.wo2-4*, la mejor regeneración es Reg2. El mejor *rel.wo1* en HH1 es 389.08%, mientras que en PC es 334.2%.

En suma, HH1 ofrece mejores valores de *rel.wo1* que PC en los grupos pequeños (1 y 2) y en el grupo 3.

**Tabla 6.4** Desviación relativa para parámetros

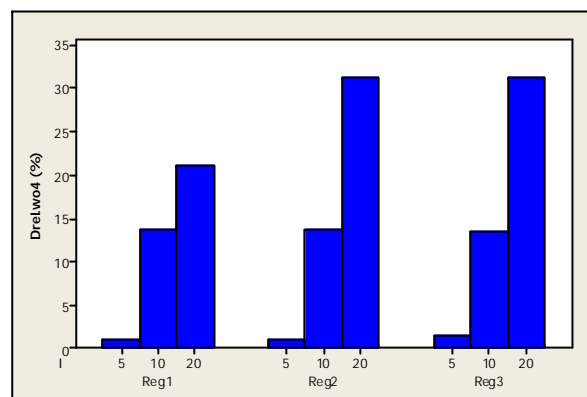
Grupo	<i>rel.wo1</i> (%)			<i>rel.wo2</i> (%)			<i>rel.wo3</i> (%)			<i>rel.wo4</i> (%)			
	Reg1	Reg2	Reg3	Reg1	Reg2	Reg3	Reg1	Reg2	Reg3	Reg1	Reg2	Reg3	
<i>I</i> = 5	D3	2.31	2.35	2.65	0.05	0.10	0.38	0.87	0.92	1.21	1.07	1.11	1.41
	D4	2.31	2.35	2.65	0.05	0.10	0.38	0.87	0.92	1.21	1.07	1.11	1.41
	D5	2.31	2.35	2.65	0.05	0.10	0.38	0.87	0.92	1.21	1.07	1.11	1.41
	D6	2.31	2.31	2.65	0.05	0.05	0.38	0.87	0.87	1.21	1.07	1.07	1.41
	Prom.	2.310	2.340	2.650	0.050	0.875	0.380	0.870	0.907	1.210	1.070	1.10	1.410
<i>I</i> = 10	D3	32.94	32.90	32.62	0.70	0.67	0.45	2.20	2.17	1.95	13.87	13.84	13.59
	D4	32.86	32.84	32.56	0.64	0.62	0.41	2.14	2.12	1.91	13.80	13.78	13.55
	D5	32.82	32.78	32.52	0.60	0.58	0.38	2.11	2.08	1.88	13.76	13.73	13.51
	D6	32.77	32.74	32.47	0.57	0.55	0.34	2.07	2.05	1.84	13.73	13.70	13.46
	Prom.	32.847	32.815	32.542	0.675	0.605	0.395	2.13	2.105	1.895	13.79	13.762	13.527
<i>I</i> = 20	D3	84.34	88.44	84.32	0.76	0.81	0.75	0.82	0.87	0.81	31.45	31.52	31.43
	D4	84.23	84.41	84.08	0.70	0.80	0.62	0.76	0.85	0.68	31.36	31.49	31.26
	D5	84.00	84.17	83.86	0.57	0.66	0.50	0.63	0.72	0.56	31.20	31.32	31.10
	D6	83.64	83.85	83.86	0.37	0.49	0.50	0.43	0.55	0.56	30.94	31.09	31.10
	Prom.	84.052	85.217	84.030	0.600	0.690	0.592	0.660	0.747	0.652	21.237	21.355	21.222
<i>K</i> = 5	D3	2.31	2.35	2.65	0.05	0.10	0.38	0.87	0.92	1.21	1.07	1.11	1.41
	D4	2.31	2.35	2.65	0.05	0.10	0.38	0.87	0.92	1.21	1.07	1.11	1.41
	D5	2.31	2.35	2.65	0.05	0.10	0.38	0.87	0.92	1.21	1.07	1.11	1.41
	D6	2.31	2.31	2.65	0.05	0.05	0.38	0.87	0.87	1.21	1.07	1.07	1.41
	Prom.	2.310	2.340	2.650	0.050	0.087	0.380	0.870	0.907	1.210	1.070	1.10	1.410
<i>K</i> = 10	D3	26.00	25.90	25.74	0.71	0.63	0.50	2.25	2.16	2.03	13.99	13.90	13.75
	D4	25.93	25.84	25.68	0.65	0.68	0.45	2.19	2.11	1.98	13.93	13.84	13.70
	D5	25.84	25.73	25.61	0.58	0.49	0.39	2.11	2.02	1.93	13.85	13.74	13.63
	D6	24.76	25.65	25.55	0.51	0.42	0.35	2.05	1.95	1.88	13.77	13.67	13.58
	Prom.	25.632	25.780	25.645	0.612	0.555	0.422	2.150	2.060	1.955	13.885	13.787	13.665
<i>K</i> = 20	D3	89.59	89.83	89.32	0.72	0.85	0.58	1.14	1.27	1.00	25.59	25.75	25.41
	D4	89.44	89.78	89.14	0.65	0.83	0.49	1.06	1.24	0.90	25.49	25.72	25.30
	D5	89.39	89.72	89.04	0.62	0.80	0.43	1.03	1.21	0.84	25.46	25.68	25.23
	D6	89.23	89.61	89.04	0.53	0.73	0.43	0.95	1.15	0.84	25.36	25.60	25.23
	Prom.	89.412	89.735	89.135	0.630	0.802	0.482	1.045	1.217	0.895	25.475	25.6875	25.292
<i>L</i> = 110	D3	81.03	81.11	80.63	0.94	0.98	0.71	2.01	2.05	1.78	27.83	27.89	27.54
	D4	80.87	81.02	80.48	0.85	0.93	0.63	1.91	2.00	1.70	27.71	27.82	27.44
	D5	80.72	80.85	80.32	0.76	0.84	0.54	1.83	1.90	1.60	27.61	27.70	27.33
	D6	80.54	80.67	80.24	0.66	0.74	0.49	1.73	1.80	1.56	27.48	27.58	27.27
	Prom.	80.790	80.912	80.417	0.8025	0.8725	0.5925	1.870	1.937	1.660	27.657	27.747	27.395
<i>L</i> = 150	D3	0.61	0.61	0.55	0.08	0.08	0.02	0.55	0.55	0.49	0.61	0.61	0.55
	D4	0.59	0.61	0.55	0.06	0.08	0.02	0.53	0.55	0.49	0.59	0.61	0.55
	D5	0.59	0.61	0.55	0.06	0.08	0.02	0.53	0.55	0.49	0.59	0.61	0.55
	D6	0.59	0.60	0.55	0.06	0.07	0.02	0.53	0.54	0.49	0.59	0.60	0.55
	Prom.	0.595	0.607	0.550	0.065	0.077	0.020	0.535	0.547	0.490	0.490	0.595	0.550
[88-132]	D3	10.69	10.47	10.50	0.59	0.40	0.43	2.67	2.48	2.51	7.77	7.57	7.59
	D4	10.68	10.42	10.43	0.59	0.35	0.36	2.67	2.43	2.44	7.77	7.51	7.52
	D5	10.67	10.38	10.43	0.58	0.32	0.36	2.66	2.39	2.44	7.75	7.47	7.52
	D6	10.60	10.38	10.43	0.52	0.32	0.36	2.60	2.39	2.44	7.69	7.47	7.52
	Prom.	10.660	10.412	10.447	0.570	0.347	0.377	2.650	2.422	2.457	7.747	7.505	7.537

La Tabla 6.4 contiene los diferentes índices de desviaciones relativas. A diferencia de la Tabla 6.3, ésta vez la información está agrupada por parámetros. El índice respectivo de

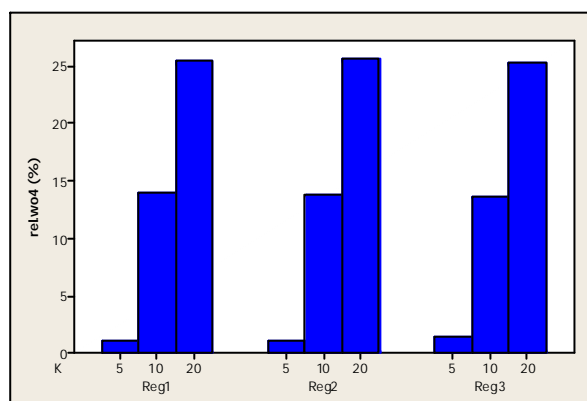
cada grupo ha sido calculado únicamente con los resultados de sobrecarga de las instancias del mismo grupo.

Si se comparan los resultados obtenidos con HH1 por grupos de parámetros con la cota  $lbw$ , Reg1 es el tipo de regeneración que tiene el menor índice para las instancias con  $I = 5$ ,  $I = 20$ ,  $K = 5$  y  $K = 10$ . Reg2 obtiene el menor valor de  $rel.wo1$  únicamente para las instancias con longitudes de estación diferentes. Reg3 tiene la mejor desviación relativa para el grupo de instancias con  $I = 10$ ,  $K = 20$ ,  $L = 110$  y  $L = 150$ .

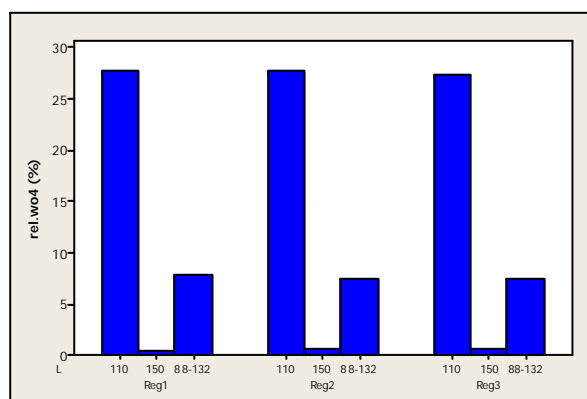
Figura 6.12  $rel.wo4$  para grupos de parámetros



a)



b)



c)

La Figura 6.12 representa los promedios de los valores obtenidos para los grupos de instancias por parámetros para el índice de desviación relativa *rel.wo4*.

HH1 obtiene desviaciones relativas mejores (*rel.wo1*) que las que obtiene PC para las instancias con  $I = 5$ ,  $I = 10$ ,  $K = 5$  y  $K = 10$ . Para las instancias con  $I = 20$ ,  $K = 20$  y para los tres tipos de ventana, PC mejora a HH1.

### 6.9 Resultados

Se propone una hiperheurística (un procedimiento heurístico que manipula heurísticas) denominada HH1 para resolver el problema de secuenciación de productos mixtos minimizando la sobrecarga. La propuesta combina soluciones representadas en el espacio de reglas heurísticas. HH1 intenta aprovechar las ventajas de las reglas de prioridad usadas normalmente en procedimientos *greedy* y la generación de nuevas soluciones de calidad dentro de un entorno. Aunque la propuesta de generación de nuevas cadenas de reglas está basada en SS, el procedimiento global implementado difiere en algunos aspectos de SS, por ejemplo, HH1 contiene cadenas de reglas en el *RefSet* en lugar de contener soluciones del problema.

La intensificación se realiza en función a la frecuencia con que cierta regla aparece en cierta posición de la secuencia en las cadenas de reglas que contiene el *RefSet*. También en función de esta tabla de frecuencias se realiza el proceso de diversificación. Se prueban tres alternativas para regenerar el *RefSet* en cada iteración.

Para probar la eficiencia de la propuesta se usa la misma batería de instancias usada en los capítulos anteriores. Se consideran sólo los resultados obtenidos entre la tercera y la sexta diversificación del procedimiento (ambas inclusive). En la experiencia se prueban tres alternativas de regeneración del conjunto de referencia. Los resultados obtenidos por cada alternativa de generación para los índices de desviación relativa son muy similares a nivel global, pero la regeneración que obtiene mejores resultados es Reg3, la cual sustituye una porción de las cadenas del conjunto de referencia (las peores) por otra cantidad igual de las mejores nuevas cadenas generadas en la presente iteración. Las mejoras entre una diversificación y la siguiente son poco prometedoras.

El alejamiento de los resultados obtenidos con HH1 respecto a la cota de sobrecarga es ligeramente mayor a la que ofrecen los 2 mejores procedimientos constructivos del capítulo 4 (UdxC y UdxR). En 44% de las instancias, HH1 obtiene mejores valores de sobrecarga que los procedimientos constructivos del capítulo 4. En este sentido, los grupos menos favorecidos por HH1 (frente a los procedimientos constructivos) son los que tienen instancias con nivel alto (20) de productos o estaciones, o tienen ventanas con diferentes longitudes. En las instancias con ventanas amplias o con producto básico, HH1 va a la par con los procedimientos constructivos.

HH1 ofrece soluciones casi tan buenas como las encontradas con CPLEX en un tiempo de búsqueda limitado a 15 minutos, pero en una quinta parte de tiempo en promedio. HH1 mejora los resultados de sobrecarga de CPLEX en el 52% de las instancias y en el 11% obtiene el mismo resultado. CPLEX ofrece mejores soluciones que HH1 en las instancias pequeñas, las instancias de tamaño medio con ventanas de diferentes longitudes, y las instancias de tamaño medio con ventanas amplias.

La búsqueda local obtiene resultados mucho mejores que HH1, pero requiere ocho veces más tiempo para obtener tales soluciones. Con la mejora local se alcanzaron hasta 25 soluciones óptimas. En una octava parte del tiempo que requiere la mejora local, HH1 puede encontrar poco más de la mitad de soluciones óptimas confirmadas con *lbw*.

Para todas las instancias de la batería, el mejor resultado considerando todos los procedimientos (PC, BL y HH1) fue encontrado con BL. HH1 coincide con el mejor valor de sobrecarga encontrado en 14% de las instancias. Tales instancias se encuentran en grupos pequeños, el grupo con ventana amplia y el grupo con modelo básico.

Con la estrategia de regeneración Reg1, HH1 está a 17.33% de desviación relativa global de la mejor solución encontrada con cualquiera de los procedimientos.



---

## Conclusiones y posibles extensiones

---

Se estudia una variante del problema de secuencias en una línea de producción que tiene por objetivo minimizar la sobrecarga de trabajo (trabajo perdido) que puede generarse ante la limitación del tiempo concedido en las estaciones para procesar las unidades. El problema tiene complejidad *NP-hard*.

El problema surge principalmente en los sistemas de manufactura repetitiva moderna (JIT). Se estudia asumiendo las siguientes características y políticas del sistema productivo: velocidad constante del transportador, estaciones cerradas, tiempos de proceso deterministas, introducción de productos en la línea a intervalos de tiempo constantes, dos operadores trabajan simultáneamente en cada estación y realizan su trabajo tan pronto como sea posible.

Para solucionar el problema se exploran procedimientos constructivos, de mejora local y una hiperheurística que combina reglas de prioridad con exploración por entornos.

Los procedimientos constructivos descritos en el capítulo 3 consideran dos tipos de productos/trabajos que se caracterizan por la presencia o ausencia de opciones extras (piloto de aparcamiento o techo solar). Respecto al ciclo, unos productos son ricos en carga de trabajo y los otros son pobres en carga de trabajo. Se propone dos procedimientos. El primero es un procedimiento *greedy* y el segundo aplica reglas sobre cotas de sobrecarga y de ocio para decidir sobre la aplicación de alguno de los procedimientos considerados. Se realiza una experiencia computacional con instancias generadas considerando diferentes características del problema como la longitud de las ventanas, el total de unidades a secuenciar, ratios de mix, número de estaciones y número de productos. El segundo algoritmo propuesto (Ud2) obtiene los mejores resultados cuando se analiza una sola estación, mientras que en el caso de estaciones múltiples tiene mayor éxito el primer algoritmo propuesto (Ud1).

Sin embargo, en la industria de los automóviles, los productos pueden requerir opciones que implican muchas variantes. Los asientos de los automóviles por ejemplo, pueden ser de ajuste manual o eléctrico, incluso de ajuste preestablecido, pueden tener respaldo de ajuste eléctrico, o tener calefacción integrada. Cada una de esas variantes es distinguible, entre otras características, por su contenido de trabajo, por lo cual en el problema de esta tesis deben considerarse más de dos tipos de productos diferenciables por su tiempo de procesamiento. Por lo anterior, en el capítulo 4 se proponen extensiones para algunos procedimientos del capítulo 3. Las extensiones consideran más de dos tipos de productos (tareas u opciones) diferenciables por las cargas de trabajo. Se consideran una y múltiples estaciones.

Se proponen tres extensiones del procedimiento Ud1 del capítulo 3. Una versión permite incurrir en ocio para procurar la regeneración del sistema (Udx), otra versión evita incurrir en sobrecarga y ocio mientras sea posible posponerla (UdxC), y una tercera alternativa procura repartir/regularizar la sobrecarga y el ocio sobre la secuencia a partir de los valores de las cotas usadas de sobrecarga y ocio (UdxR).

Se realiza una experiencia computacional que considera múltiples estaciones y múltiples tareas con una batería de 100 instancias de Scholl et al. (1998). Con poca diferencia respecto a Udx y UdxC, UdxR obtiene la desviación relativa global más pequeña. La extensión propuesta YRx, requiere mayor tiempo de procesamiento y los resultados no son cercanos a los que se obtienen con el resto de las propuestas. El mejor resultado lo ofrece UdxR y el tiempo de procesamiento es aceptable. El procedimiento UdxR usa la cota de sobrecarga  $lbw$  que aunque no es de muy buena calidad, puede obtenerse en un tiempo insignificante. Para el grupo de instancias 9, el cual considera un modelo básico en las instancias, el mejor valor de desviación relativa global lo ofrece el procedimiento Udx.

Después de probar procedimientos constructivos, en el capítulo 5 se aplican procedimientos de Búsqueda Local (BL). Se aplican dos tipos de vecindarios en BL: intercambios de elementos e inserción de segmentos. La experiencia computacional de este capítulo considera múltiples estaciones y múltiples tareas. La batería de instancias usada en la experiencia computacional de este capítulo es la misma del capítulo 4. Como soluciones iniciales de la BL se usan las soluciones proporcionadas por los procedimientos constructivos del capítulo 4.

Así, en el capítulo 5 se toman los resultados de los cuatro procedimientos constructivos del capítulo cuatro, Udx, UdxC, UdxR y YRx, sobre los cuales se aplican procedimientos de BL. Se prueban dos tipos de vecindarios en la BL: intercambio de elementos e inserción de segmentos. Se consideran cuatro alternativas en el intercambio de elementos, las cuales se han denominado 2Int, 3Int(a), 3Int(b) y 2-3Int(b). En la inserción de segmentos se prueban tamaños de segmento desde 2 hasta 10.

Se impuso límite en el tiempo de búsqueda de los óptimos locales. El límite de tiempo para los procedimientos de BL con intercambio de elemento fue de 3600 segundos. Excepto para 2-3Int(b), el cual tenía límite de 3600 para la primera fase de Búsqueda y 1800 segundos para la segunda fase. La BL con inserción de segmentos estuvo limitada a 1800 segundos de búsqueda.

De los procedimientos por intercambios, se obtienen mejores resultados con intercambios de dos elementos (2Int) en tiempo relativamente bueno. El óptimo local de 2Int puede mejorarse con 3Int(b), con un esfuerzo computacional mayor debido al tamaño del vecindario. La BL con inserción de segmentos obtiene óptimos locales tan buenos o mejores como 2Int, en la mitad de tiempo. El tiempo que requiere la mejora local por inserción de segmentos es más aceptable que el intercambio de elementos. De igual manera, la calidad de las soluciones es ligeramente mejor. Los mejores valores de desviación relativa para la inserción se obtiene en los segmentos de mayor tamaño (10 elementos), sin embargo, la mayor cantidad de óptimos confirmados con valor de  $lbw$  se obtienen con segmentos de tamaño más pequeño (3-6 elementos).

Los mejores valores para el índice de desviación relativa global  $rel.wo2$  se obtienen cuando se combinan la solución inicial del procedimiento UdxR y las inserciones de segmentos de tamaño 10. Esto sugiere una relación entre el tamaño de segmento y el tamaño de la secuencia lo cual se considera como una extensión de este trabajo. El índice  $rel.wo1$  se reduce a la mitad con las diferentes alternativas consideradas de BL.

Se propone una hiperheurística denominada HH1 para resolver el problema de secuenciación de productos mixtos minimizando la sobrecarga. La propuesta combina soluciones representadas en el espacio de reglas heurísticas. HH1 intenta aprovechar las ventajas de las reglas de prioridad usadas normalmente en procedimientos *greedy* y la generación de nuevas soluciones de calidad dentro de un entorno. Aunque la propuesta está basada en SS, el procedimiento implementado difiere en algunos aspectos, por ejemplo, HH1 contiene cadenas de reglas en el *RefSet* en lugar de contener soluciones del problema.

La intensificación se realiza en función a la frecuencia con que cierta regla aparece en cierta posición de la secuencia en las cadenas de reglas que contiene el *RefSet*. También en función de esta tabla de frecuencias se realiza el proceso de diversificación. Se prueban tres formas de regenerar el *RefSet*. Los resultados obtenidos por cada tipo de regeneración son muy similares.

Para probar la eficiencia de la propuesta se usa la misma batería de instancias usada en los capítulos anteriores. Los resultados que produce HH1 son casi tan buenos como los que se obtienen con los procedimientos constructivos del capítulo 4. La búsqueda local obtiene resultados mucho mejores que HH1, pero requiere ocho veces más tiempo que HH1 para encontrarlos.

Sin duda es necesario conseguir una cota de sobrecarga de calidad. Respecto a la cota usada, BL se desvía 20 por ciento, pero requiere, a nuestro juicio, demasiado tiempo de CPU. Por otra parte, las heurísticas constructivas y la hiperheurística propuesta ofrecen una solución en tiempo aceptable pero de menor calidad. Hay diferentes puntos en los procedimientos susceptibles de modificaciones para procurar mejores soluciones. Por ejemplo, detectar si existe una relación entre el tamaño de la secuencia y el tamaño de segmento usado en la mejora local por inserción de segmentos, buscar vecindarios alternativos, buscar otras estrategias de exploración que requieran menos tiempo y obtengan mejores soluciones como la búsqueda en vecindad variable. En HH1 se pueden buscar mejoras al seleccionar los elementos a intercambiar, en la aplicación de la aleatoriedad para seleccionar los elementos a intercambiar, en el uso de memoria sobre los cruces de cadenas, o en la aplicación de operadores de cruce como los aplicados en GA.

Hasta donde sabemos, las instancias extraídas de la literatura y que han sido usadas en varias experiencias computacionales de esta tesis son relativamente pequeñas. Las instancias con las mayores cantidades de unidades a producir, número de estaciones y versiones del producto principal, pueden representar situaciones realistas, aunque también se sabe que hay empresas que contemplan planes de producción por turno o día mayores a los usados en este estudio, pues la cantidad de estaciones de la línea es mayor (aunque algunos autores sugieren solo considerar estaciones críticas en carga para determinar una secuencia). Debe considerarse que los tiempos de procesamiento de las instancias son deterministas, mientras que en la realidad son estocásticos, a pesar de la alta estandarización que reduce la variabilidad en las tareas y sus duraciones. Considerar tiempos de procesamiento estocásticos en el problema es también una posible extensión del trabajo.



---

## Referencias

---

- Aarts E., Lenstra J.K., 1997. Local search in combinatorial optimization. Wiley. New York.
- Aigbedo H., Monden Y., 1997. A parametric procedure for multicriterion sequence scheduling for Just-In-Time mixed-model assembly lines. *International Journal of Production Research*, 35 (9), 2543-2564.
- Bard J.F., Dar-El E., Shtub A., 1992. An analytic framework for sequencing mixed model assembly lines. *International Journal of Production Research*, 30 (1), 35-48.
- Bard J.F., Shtub A., Joshi S.B., 1994. Sequencing mixed-model assembly lines to level part usage and minimize line length. *International Journal of Production Research*, 32 (10), 2431-2454.
- Bautista J., 1993. Procedimientos heurísticos y exactos para la secuenciación de sistemas productivos de unidades homogéneas (contexto JIT). Tesis doctoral. cpda-etseib.
- Bautista J., Companys R., Corominas A., 1992. Level schedules in assembly lines and the Alabama paradox. D.I.T. 92/10, DOE, ETSEIB-UPC.
- Bautista J., Companys R., Corominas A., 1993. Level schedules for mixed-model assembly lines and the apportionment problem. D.I.T. 93/11, DOE, ETSEIB-UPC.
- Bautista J., Companys R., Corominas A., 1995. Seqüenciació de unitats en context JIT. Edicions UPC, TOE, Barcelona.
- Bautista J., Companys R., Corominas A., 1996a. Heuristic and exact algorithms for solving the Monden problem. *European Journal of Operational Research*, 88 (1), 101-113.
- Bautista J., Companys R., Corominas A., 1996b. A note on the relation between the product rate variation problem and the apportionment problem. *Journal of the Operations Research Society*, 47 (11), 1410-1414.
- Bautista J., Companys R., Corominas A., 1996c. Una visión sobre secuencias regulares. *Boletín SEIO*, 12 (2), 4-6, Junio.
- Bautista J., Companys R., Corominas A., 2000. Note on cyclic sequences in the product rate variation problem. *European Journal of Operational Research*, 124, (3), 468-477.
- Bautista J., Companys R., Pereira J., Mateo M., 2001. Aplicación de la CLP al problema de secuencias regulares con restricciones en una cadena de montaje de automóviles. XXVI Congreso Nacional de Estadística de Investigación Operativa, Ubeda.
- Bock F., 1958. An algorithm for solving traveling-salesman and related network optimization. *Operations Research*, 6 (6), 897-897.
- Bock S., Rosenberg O., Brackel T., 2006. Controlling mixed-model assembly lines in real-time by using distributed systems. *European Journal of Operational Research*, 168 (3), 880-904.
- Bolat A., 1994. Sequencing jobs on an automobile assembly line: objectives and procedures. *International Journal of Production Research*, 32 (5), 1219-1236.
- Bolat A., 1997a. Efficient methods for sequencing minimum job sets on mixed model assembly lines. *Naval Research Logistics*, 44 (5), 419-437.
- Bolat A., 1997b. Stochastic procedures for scheduling minimum job sets on mixed model assembly lines. *Journal of the Operational Research Society*, 48 (5), 490-501.
- Bolat A., Yano C.A., 1992a. Scheduling algorithms to minimize utility work at a single station on paced assembly line. *Production Planning and Control*, 3 (4), 393-405.
- Bolat A., Yano C.A., 1992b. A surrogate objective for utility work in paced assembly line. *Production Planning and Control*, 3 (4), 406-412.

## Referencias

- Bolat A., 2003. A mathematical model for selecting mixed models with due dates. *International Journal of Production Research*, 41 (5), 897-918.
- Brucker P., 2001. *Scheduling Algorithms*. 3a Edición. Springer. Germany.
- Burns L.D., Daganzo C.G., 1987. Assembly line job sequencing principles. *International Journal of Production Research*, 25 (1), 71-99.
- Celano G., Costa A., Fichera S., Perrone G., 2004. Human factors policy testing in the sequencing of manual mixed-model assembly lines. *Computers & Operations Research*, 31 (1), 39-59.
- Chakravarty A.K., Shtub A., 1986. A cost minimization procedure for mixed-model production lines with normal distributed task times. *European Journal of Operations Research*, 23 (1), 25-36.
- Chase R.B., Aquilano N.J., Jacobs N.J., 2000. *Administración de producción y operaciones. Manufactura y servicios*. 8a Edición, McGraw-Hill, Santa Fe de Bogotá.
- Cheng L.P., Ding F.Y., 1996. Modifying mixed-model assembly line sequencing methods to consider weighted variations for just-in-time production systems. *IIE Transactions*, 28 (11), 919-927.
- Chul J.H., Yeongho K., y Yeo K.K., 1998. A genetic algorithm for multiple objective sequencing problems in mixed model assembly lines. *Computers & Operations Research*, 25 (7-8), 675-690.
- Coffman P.E., Hoffman S.E., Weiner S.A., 1985. *Assembly plant modeling*. TIMS/ORSA Conference in Boston, Massachusetts.
- Companys R., 1989. Secuenciación de productos en el montaje para lograr la regularidad en el consumo de recursos. *CIM* 10, 49-54.
- Companys R., 2003a. *Secuenciación. Programación de proyectos y de taller. Equilibrado y secuenciación de líneas*. Vol. I, Ed. cpda-etseib, Barcelona.
- Companys R., 2003b. *Secuenciación. Programación de proyectos y de taller. Equilibrado y secuenciación de líneas*. Vol. II, Ed. cpda-etseib, Barcelona.
- Companys R., Corominas A., 1996a. *Organización de la producción II, Dirección de operaciones 4*. Escola Tecnica Superior de Enginyers Industrials de Barcelona, ETSEIB. Universitat Politècnica de Catalunya, Ediciones UPC, España.
- Companys R., Corominas A., 1996b. *Direcció de la producció I*. Ediciones multimedia de la UOC, Barcelona, España.
- Croes G.A., 1958. A method for solving traveling-salesman problems. *Operations Research*, 6 (6), 791-812.
- Dar-El E.M., 1978. Mixed-model assembly line sequencing problems. *OMEGA*, 6, 313-323.
- Dar-El E.M., Cothor R.F., 1975, Assembly line sequencing for model mix. *International Journal of production research*. 13,(5), 463-477.
- Díaz A., Glover F., Ghaziri H.M., González J.L., Laguna M., Moscazo P., Tseng F.T., 1996. *Optimización heurística y redes neuronales*. Editorial Paraninfo S.A. Madrid.
- Ding F.Y., Cheng L., 1993a. A simple sequencing algorithm for mixed models assembly lines in JIT Production Systems. *Operations Research Letters*, 13 (1), 27-36.
- Ding F.Y., Cheng L., 1993b. An effective mixed model assembly lines sequencing heuristics for just-in-time productions systems. *Journal of Operations Management*, 11, 45-50.
- Ding F.Y., Zhu J., Sun H., 2006. Comparing two weighted approaches for sequencing mixed-model assembly lines with multiple objectives. *International Journal of Production Economics*, 102 (1), 108-131.
- Drexl A., Kimms A., 2001. Sequencing JIT mixed-model assembly lines under station-load and part-usage constraints. *Management Science*, 47 (3), 480-491.

- Drexl A., Kimms A., Matthiessen L., 2006. Algorithms for the car sequencing and the level scheduling problem. *Journal of Scheduling*, 9 (2), 153-176.
- Duplaga E.A., Hahn C.K., Hur D., 1996. Mixed-model assembly line sequencing at Hyundai Motor Company. *Production & Inventory Management Journal*, 37 (3), 20-26.
- García Sabater J.P., 2000. Modelos, métodos y algoritmos de resolución del problema de secuenciación de Unidades homogéneas en el Sector del Automóvil. Tesis Doctoral, Universidad Politécnica de Valencia.
- Glover F., 1989. Tabu search-Part I. *ORSA Journal on Computing*, 1, 190-206.
- Glover F., Laguna M., 1997. *Tabu search*. Kluwer Academic Publishers, Boston.
- Glover F., Martí R., 2000. Fundamentals of scatter search and path relinking. *Control and cybernetics*, 29 (3), 653-684.
- Goldratt E.M., 1992. *El Síndrome del Pajar*. Editorial Castillo, Monterrey, México.
- González B., Adenso-Díaz B., 2006. A scatter search approach to the optimum disassembly sequence problem. *Computers & Operations Research*, 33 (6), 1776-1793.
- Gouveia L., Voss S., 1995. A classification of formulation for the (Time-Dependent) Traveling Saleman Problem. *European Journal of Operational Research*, 83 (1), 69-82.
- Guerre F., Frein Y., Bouffard R., 1995. An efficient procedure for solving a car sequencing problem. *Proceedings ETFA, Symposium of Emerging Technologies and Factory Automation*.
- Heike G, Ramulu M, Sorenson E, Shanahan P, Moinzadeh K, 2001. Mixed model assembly alternatives for low-volume manufacturing: The case of the aerospace industry. *International Journal of Production Economics*, 72 (2), 103-120.
- Hiller F., Lieberman G.J., 1991. *Introducción a la investigación de operaciones*. Quinta edición en inglés, tercera edición en español, McGraw-Hill, México.
- Hindi K.S., Ploszajski G., 1994. Formulation and solution of a selection and sequencing problem in car manufacture. *Computers & Industrial Engineering*, 26 (1), 203-211.
- Hyun C.J., Kim Y., Kim Y.K., 1998. A genetic algorithm for multiple objective sequencing problems in mixed model assembly lines. *Computers & Operations Research*, 25 (7-8), 675-690.
- Holland J.H., 1992. *Adaptation in natural and artificial systems. An introductory analysis with applications to biology, control and artificial intelligence*. Cambridge, Mass. MIT Press.
- Inman R.R., Bulfin R.L., 1991. Sequencing JIT mixed-model assembly lines. *Management Science*, 37 (7), 901-904.
- Kim H.G., Cho H.S., 2003. Sequencing in mixed-model final assembly line with three goals: Simulated annealing approach. *International Journal of Industrial Engineering-Theory Applications and Practice*, 10 (4), 607-613.
- Kim S., Jeong B., 2001. Product sequencing problem in mixed-model assembly line to minimize unfinished works. Yonsei University, Seoul, Corea.
- Kim Y.K., Hyun C.J., Kim Y., 1996. Sequencing in mixed model assembly lines: A genetic algorithm approach. *Computers & Operations Research*, 23 (12), 1131-1145.
- Kim Y.K., Kim J.Y., Kim Y., 2000. A coevolutionary algorithm for balancing and sequencing in mixed model assembly lines. *Applied Intelligence*, 13 (3), 247-258.
- Kis T., 2004. On the complexity of the car sequencing problem. *Operations Research Letters*, 32 (4), 331-335.

- Korkmaz T., Meral S., 2001. Bicriteria sequencing methods for mixed model assembly line just-in-time production systems. *European Journal of Operational Research*, 131 (1), 188-207.
- Korte B., Vygen J., 2000. *Combinatorial optimization, theory and algorithms*. 21 Algorithms and Combinatorics. Springer, Germany.
- Kotani S., Ito T., Ohno K., 2004. Sequencing problem for a mixed-model assembly line in the Toyota production system. *International Journal of Production Research*, 42 (23), 4955-4974.
- Kouvelis P., Karabati S., 1999. Cyclic scheduling in synchronous production lines. *IIE Transactions*, 31 (8), 709-719.
- Kubiak W., 1993. Minimizing variation of production rates in just-in-time systems: a survey. *European Journal of Operational Research*, 66 (3), 259-271.
- Kubiak W., Sethi S., 1991. Level schedules for mixed-model assembly lines in just-in-time production systems. *Management Science*, 37 (1): 121-122.
- Laguna M., Matí R., 2003. *Scatter Search. Methodology and implementations in C*. Kluwer Academic Publishers. USA.
- Laguna M., Martí R., 2005. Experimental testing of advanced scatter search designs for global optimization of multimodal functions. *Journal of Global Optimization*, 33 (2): 235-255.
- Lin S., 1965. Computer solutions of the traveling salesman problem. *Bell System Technical Journal*, 44 (10), 2245-2269.
- Lin S., Kerningham B.W., 1973. An effective heuristic algorithm for the traveling salesman problem. *Operations Research*, 21, 498-516.
- Macaskill J.L.C., 1972. Production-line balances for mixed-model lines. *Management Science, Application Series*, 19 (4), 423-434.
- Macaskill J.L.C., 1973. *Computer Simulation for Mixed-Model Production Lines*. *Management Science, Theory Series*, 20 (3), 341-348.
- Mansouri S.A., 2005. A Multi-objective genetic algorithm for mixed-model sequencing on JIT assembly lines. *European Journal of Operational Research*, 167 (3), 696-716.
- Martí R., 2006. Scatter search - Wellsprings and challenges. *European Journal of Operational Research*, 169 (2), 351-358.
- Martí R., Laguna M., Glover F., 2006. Principles of scatter search. *European Journal of Operational Research*, 169 (2), 359-372.
- Matanachai S., Yano C.A., 2001. Balancing mixed-model assembly lines to reduce work overload. *IIE Transactions*, 33 (1), 29-42.
- Michalewicz Z., Fogel D.B., 2004. *How to solve it: Modern Heuristics*. Second Edition. Springer. Germany.
- Miltenburg J., 1989. Level schedules for mixed-model assembly lines in just-in-time production systems. *Management Science*, 35 (2), 192-207.
- Miltenburg J., Goldstein T., 1991. Developing production schedules with balance part usage and smooth production loads for just-in-time production systems. *Naval Research Logistics*, 38 (6), 893-910.
- Miltenburg J., Sinnamon G., 1989. Scheduling mixed-model multilevel just-in-time production systems. *International Production Research*, 27 (9), 1487-1509.
- Miltenburg J., Steiner G., Yeomans S., 1990. A dynamic programming algorithm for scheduling mixed-model, just-in-time production systems. *Mathl. Comput. Modelling*, 13 (3), 57-66.
- Monden Yasuhiro, 1983. *Toyota production system*. Institute of Industrial Engineers Press, Norcross, GA.

- Monden Yasuhiro, 1987. El sistema de producción toyota. Price Waterhouse, IESE, España.
- Nemhauser G.L., Wolsey L.A., 1988. Integer and combinatorial optimization. John Wiley & Sons, New York.
- Nicholson T.A.J., 1965. A sequential method for discrete optimization problems and its application to the assignment, traveling salesman and three scheduling problems. *Journal of the Institute of Mathematics and its Applications*, 13, 362-375.
- Nof S.Y., Wilhelm W.E., Warnecke H.J., 1997. *Industrial Assembly*. Ed. Chapman & Hall, UK.
- Okamura K., Yamashina H., 1979. A heuristic algorithm for the assembly line model-mix sequencing problem to minimize the risk of stopping the conveyor. *International Journal of Production Research*, 17 (3), 233-247.
- Osman I.H., Laporte G., 1996. Metaheuristics: A bibliography. *Annals of Operations Research*, 63, 513-623.
- Ow P.S., Morton T.E., 1988. Filtered beam search in scheduling. *International Journal of Production Research*, 26 (1), 35-62.
- Page E.S., 1965. On montecarlo methods in congestion methods: I, searching for an optimum in discrete situation. *Operations Research*, 13, 291-299.
- Papadimitriou C.H., Steiglitz K., 1982. *Combinatorial optimization. Algorithms and complexity*. Prentice Hall.
- Poler R., Rodríguez A., García J.P., Lario F.C., 1999. Genetic algorithm for assembly sequencing. *International Conference on Industrial Engineering and Production Management*. Glasgow. Proceeding book-2. INRIA.
- Rachamadugu R., Yano C.A., 1994. Analytical tools for assembly-line design and sequencing. *IIE Transactions*, 26 (2), 2-11.
- Reiter S., Sherman G., 1965. Discrete optimizing. *Journal of the Society for Industrial and Applied Mathematics*, 13 (3), 864-889.
- Ruiz Jiménez A., Domínguez Machuca J.A., Álvarez Gil M.J., Domínguez Machuca M.A., García González S., 1995a. *Dirección de Operaciones: Aspectos Estratégicos en la Producción y en los Servicios*. McGraw-Hill. Madrid, España.
- Ruiz Jiménez A., Domínguez Machuca J.A., Domínguez Machuca M.A., García González S., Álvarez Gil M.J., 1995b. *Dirección de Operaciones: Aspectos Tácticos y Operativos en la Producción y en los Servicios*. McGraw-Hill. Madrid, España.
- Sarin S.C., Greco M.P., Dar-El E.M., 2006. Sequencing and loading of products on a flowline. *European Journal of Operations Research*, 168 (3), 905-921.
- Sarker B.R., Pan H.X., 1998. Designing a mixed-model assembly line to minimize the cost of idle and utility times. *Computers & Industrial Engineering*, 34 (3), 609-628.
- Sarker B.R., Pan H.X., 2001. Design configuration for a closed-station, mixed-model assembly line: a filing cabinet manufacturing system. *International Journal of Production Research*, 39 (10), 2251-2270.
- Scholl A., 1999. *Balancing and sequencing of assembly lines*. 2nd ed. Physica, Heidelberg.
- Scholl A., Klein R., Domschke W., 1998. Pattern based vocabulary building for effectively sequencing mixed-model assembly lines. *Journal of Heuristics*, 4 (4), 359-381.
- Smith K., Palaniswami M., Krishnamoorthy M., 1996. Traditional heuristic versus Hopfield neural network approaches to a car sequencing problem. *European Journal of Operational Research*, 93 (2), 300-316.
- Steiner G., Yeomans S., 1993. Level schedules for mixed-model just-in-time processes. *Management Science*, 39 (6), 728-735.

- Steiner G., Yeomans S., 1996. Optimal level schedules in mixed-model, multilevel JIT assembly systems with pegging. *European Journal of Operational Research*, 95 (1), 38-52.
- Stützle T., 1998. Local search algorithms for combinatorial problems. Analysis, improvements and new applications. PhD thesis, Technischen Universität Darmstadt.
- Sugimori Y., Kusunoki F., Cho F., Uchikawa S., 1977. Toyota production system and kanban system. Materialization of just-in-time and respect-for-human system. *International Journal of Production Research*, 15 (6), 553.
- Sumichrast R.T., Russell R.S., 1990. Evaluating mixed model assembly line heuristics for just-in-time production systems. *Journal of Operations Management*, 9 (3), 371-390.
- Sumichrast R.T., Russell R.S., Taylor B., 1992. A comparative analysis of sequencing procedures for mixed-model assembly lines in a just-in-time production system. *International Journal of Production Research*, 30 (1), 199-214.
- Sumichrast R.T., Clayton E.R., 1996, Evaluating sequences for paced mixed-model assembly lines with JIT component fabrication, *International Journal of Production Research*, 34 (11), 3125-3143.
- Thomopoulos N.T., 1967. Line balancing – sequencing for mixed model assembly. *Management Science*, 14, B59-B75.
- Tsai L., 1995. Mixed-model sequencing to minimize utility work and the risk of conveyor stoppage. *Management Science*, 41 (3), 485-495.
- Voss S., Martello S., Osman I.H., Roucairol C., 1999. *Meta-heuristics: advances and trends in local search paradigms for optimization*. Kluwer Academic Publishers. INRIA. USA.
- Wester L., Kilbridge M., 1963. The assembly line model-mix sequencing problem. *Proceedings of the international Conference on Operations Research, Oslo 1963, Dunod, Paris* 247-260.
- Xiaobo Z., Ohno K., 1997. Algorithms for sequencing mixed models on an assembly line in a JIT production system. *Computer and Industrial Engineering*, 32 (1), 47-56.
- Xiaobo Z., Ohno K., 2000. Properties of a sequencing problem for a mixed model assembly line with conveyor stoppages. *European Journal of Operational Research*, 124 (3), 560-570.
- Xiaobo Z., Ohno K., Lau H.S., 2004. A balancing problem for mixed model assembly lines with a paced moving conveyor. *Naval Research Logistics*, 51 (3), 446-464.
- Xiaobo Z., Zhaoying Z., Ainishet A., 1999. A note on Toyota's goal of sequencing mixed models on an assembly line. *Computer and Industrial Engineering*, 36 (1), 57-65.
- Yano C.A., Bolat A., 1989. Survey, development, and application of algorithms for sequencing paced assembly lines. *Journal of Manufacturing and Operations Management*, 2 (3), 172-198.
- Yano C.A., Rachamadugu R., 1991. Sequencing to minimize work overload in assembly lines with product options. *Management Science*, 37 (5), 572-586.
- Zeramardini W., Aigbedo H., Monden Y., 2000. Bicriteria sequencing for just-in-time mixed-model assembly lines. *International Journal of Production Research*, 38 (15), 3451-3470.

**Tabla A1.1** Nomenclatura general

## Índices

- $i$  índice relativo al tipo de producto  $i$  ( $i = 1, \dots, I$ ),  
 $k$  índice relativo a las estaciones  $k$  ( $k = 1, \dots, K$ ),  
 $t$  posición en la secuencia  $t$  ( $t = 1, \dots, T$ ).

## Parámetros

- $c$  tiempo de ciclo,  
 $K$  número de estaciones en la línea,  
 $L_k$  longitud de la estación-ventana temporal  $k$ ,  
 $I$  conjunto de productos,  
 $n_i$  demanda inicial del producto  $i$ ,  
 $d_i$  demanda pendiente de fabricar del producto  $i$  en cierta etapa de la secuencia,  
 $T$  total de unidades a fabricar en un horizonte de tiempo determinado, o número de posiciones en la secuencia,  
 $p_{ik}$  tiempo de procesamiento para el producto  $i$  en la estación  $k$ ,  
 $cs_k$  coste asociado a la sobrecarga en la estación  $k$  por unidad de tiempo,  
 $f_{kt}$  instante de finalización de la tarea que ocupa la posición  $t$  en la estación  $k$ ,  
 $r_{kt}$  tiempo de procesamiento en la estación  $k$  en la posición  $t$ ,  
 $r_i$  índice de prioridad dinámico de asignación para el producto  $i$ ,  
 $lbw$  una cota inferior del trabajo perdido,  
 $lbo$  una cota inferior del tiempo ocioso o improductivo,  
 $Z_\zeta$  sobrecarga o trabajo perdido total ocasionado por la secuencia  $\zeta$ .

## Variables

- $x_{it}$  variable binaria que adopta el valor 1 si en la  $t$ -ésima posición de la secuencia se asigna el producto  $i$ ; en caso contrario adopta el valor 0,  
 $s_{kt}$  posición del trabajador respecto al origen (inicio) de su estación  $k$  al iniciar su tarea sobre el producto asignado en la posición  $t$  de la secuencia,  
 $w_{kt}$  sobrecarga en la estación  $k$  en la etapa  $t$  de la secuencia,  
 $o_{kt}$  ocio en la estación  $k$  en la etapa  $t$  de la secuencia.

**Tabla A1.2** Relaciones entre variables y parámetros

$$\begin{aligned}
 T &= \sum_{i=1}^I n_i \\
 r_{ik} &= d_i \cdot c - p_{ik} \\
 \mathbf{r}_{kt} &= \sum_{i=1}^I p_{ik} \cdot x_{it} \\
 s_{kt} &= \max((t-1) \cdot c, f_{k,t-1}) \\
 f_{kt} &= \min(s_{kt} + p_{ik}, (t-1) \cdot c + L_k) \\
 w_{kt} &= [p_{ik} + s_{kt} - ((t-1) \cdot c + L_k)]^+ \quad \text{donde } [x]^+ = \max(0, x)
 \end{aligned}$$

Se muestra la forma general para obtener la desviación relativa para instancias  $h$  ( $T, P, M, n_i, t_{ij}, cs_j, L_j$ ) con múltiples estaciones usadas en el capítulo 3. Sean:

$C_1^h$  tiempo mínimo necesario ponderado para procesar las  $T$  unidades de la instancia  $h$ .

$$C_1^h = \sum_{k=1}^K \left( \sum_{i=1}^I n_i \cdot p_{ik} \right) \cdot cs_k$$

$C_2^h$  tiempo máximo concedido para procesar las  $T$  unidades de la instancia  $h$ .

$$C_2^h = \left( \sum_{k=1}^K ((T-1) \cdot c) + L_k \right) \cdot cs_k$$

$C_\infty^{hl}$  tiempo real necesario para procesar las  $T$  unidades de la instancia  $h$ , en el orden de la secuencia obtenida con el procedimiento  $l$ .

La desviación relativa entre la cota y el tiempo total necesario para procesar todos los trabajos de la instancia  $h$  en el orden de la secuencia obtenida con el procedimiento  $l$  ( $d_{hl}$ ) se obtiene de la forma siguiente:

$$d_{hl} = \left[ \frac{C_\infty^{hl} - \max\{C_1^h, C_2^h\}}{\max\{C_1^h, C_2^h\}} \right]^+$$

La desviación media ( $d_l^H$ ) ofrecida por el procedimiento  $l$  para el grupo de problemas  $H$  ( $h = 1, \dots, |H|$ ), con cardinalidad  $|H|$ , se obtiene de la forma siguiente:

$$d_l^H = \frac{\sum_{\forall h \in H} d_{hl}}{|H|}$$

A continuación se enumeran las 20 reglas de prioridad que se han utilizado en el capítulo 6. Sean:

$i$	el tipo de producto,
$p_{ik}$	tiempo de procesamiento de la producto $i$ en la estación $k$ ,
$n_i$	demanda inicial del producto $i$ ,
$T$	producción total o tamaño de la secuencia, $T = \sum n_i$ ,
$d_i$	demanda pendiente del producto $i$ ,
$r_{ik}$	índice dinámico de asignación para el producto $i$ en estación $k$ , $r_{ik} = d_i \cdot \Delta_{ik}$ ,
$\Delta_{ik}$	desplazamientos $\Delta_{ik} =  p_{ik} - c $ ,
$L_k$	longitud de la estación $k$ ,
$u_i$	producción ideal del producto $i$ , $u_i = n_i/T$
$s_k$	posición inicial del trabajador en la estación $k$ ,
$w_t$	sobrecarga acumulada hasta la etapa $t$ , $w_t = \sum_0^{t-1} w_{tk}$ ,
$\bar{p}$	tiempo de procesamiento promedio $\bar{p} = \sum p_{ik} / (K \cdot I)$ ,
$k^b$	estación cuello de botella,
$w_i$	sobrecarga total en la etapa actual producida por $i$ , $w_i = \sum_k w_{ik}$ ,
$o_i$	ocio total en la etapa actual producida por $i$ , $o_i = \sum_k o_{ik}$ .

Regla 1. Producto con mayor tiempo de procesamiento.

$$i^* \in \arg \max \{p_{ik}\}$$

Regla 2. Producto con menor tiempo de procesamiento.

$$i^* \in \arg \min \{p_{ik}\}$$

Regla 3. Producto con tiempo de procesamiento más cercano al tiempo de procesamiento promedio.

$$i^* \hat{I} \arg \min \{|p_{ik} - \bar{p}|\}$$

Regla 4. Producto con tiempo de procesamiento más cercano al tiempo de ciclo.

$$i^* \hat{I} \arg \min \{\Delta_{ik}\}$$

Regla 5. Producto con mayor demanda pendiente.

$$i^* \hat{I} \arg \max \{d_i\}$$

Regla 6. Producto con menor demanda pendiente.

$$i^* \in \arg \min \{d_i\}$$

Regla 7. Producto con mayor índice dinámico.

$$i^* \hat{I} \arg \max \{r_i\}$$

Regla 8. Producto con menor índice dinámico.

$$i^* \hat{I} \arg \min \{r_i\}$$

Regla 9. Producto que de asignarse, ocasionaría un mayor desplazamiento total de los trabajadores en todas las estaciones.

$$i^* \hat{I} \arg \max \{\sum_{k=1}^K \Delta_{ik}\}$$

Regla 10. Producto que ocasiona un menor desplazamiento total de los trabajadores en todas las estaciones.

$$i^* \in \arg \min \left\{ \sum_{k=1}^K \Delta_{ik} \right\}$$

Regla 11. De la estación cuello de botella, el producto con mayor tiempo de procesamiento.

$$i^* \hat{=} \arg \max \{ p_{ik^b} \}$$

Regla 12. De la estación cuello de botella, el producto con menor tiempo de procesamiento.

$$i^* \in \arg \min \{ p_{ik^b} \}$$

Regla 13. El producto  $i$  que se acerca más a la producción ideal.

$$i^* \in \arg \min \{ |(n_i - d_i) - t \times u_i| \}$$

Regla 14. Producto  $i$  con mayor suma de índices dinámicos para todas las estaciones.

$$i^* \in \arg \max \left\{ \sum_{k=1}^K r_{ik} \right\}$$

Regla 15. Producto  $i$  con menor suma de índices dinámicos para todas las estaciones.

$$i^* \in \arg \min \left\{ \sum_{k=1}^K r_{ik} \right\}$$

Regla 16. El producto que ocasiona más sobrecarga en la etapa actual.

$$i^* \in \arg \max \{ w_i \}$$

Regla 17. El producto que ocasiona más ocio en la etapa actual.

$$i^* \in \arg \max \{ o_i \}$$

Regla 18. El producto que ocasiona menos sobrecarga en la etapa actual.

$$i^* \hat{=} \arg \min \{ w_i \}$$

Regla 19. El producto que ocasiona menos ocio en la etapa actual.

$$i^* \hat{=} \arg \min \{ o_i \}$$

Regla 20. Bajo el concepto de regularidad, aquel producto que ocasiona la sobrecarga mas cercana a la sobrecarga ideal, suponiendo que la cota de la sobrecarga  $lbw > 0$ .

$$i^* \hat{=} \arg \min \left\{ \left( \frac{lbw}{T} \right) \cdot t - w_i - w_i \right\} \quad \text{donde } w_i \text{ es acumulada.}$$

BL	<i>Local Search</i> o Búsqueda Local.
CSP	<i>Car Sequencing Problem</i> .
Congestión	<i>Congestion</i> . Ocurre congestión en una cadena de montaje cuando en una estación abierta por el lado derecho (final de la estación), el operador necesita cruzar el límite final de su estación para concluir su trabajo.
Gestión	Acción que consiste en hacer diligencias conducentes al logro de un negocio o de un deseo cualquiera.
Greedy	Del inglés; codicioso, avaro, voraz o miope.
GA	<i>Genetic Algorithms</i> o Algoritmos Genéticos.
Estratégico	Relativo a estrategia. Formulación de los objetivos últimos de una operación y de los medios necesarios para su éxito. Habilidad para conducir un asunto.
JIT	Sistema de producción Justo a Tiempo o <i>Just in Time</i> .
Línea de Montaje	<i>Assembly Line</i> . Se refiere a un ensamble progresivo unido por un dispositivo de manejo de materiales. (Chase, 1999). Sistema de producción orientado al flujo, consiste de un número de estaciones de trabajo ubicadas a lo largo de un sistema de manejo de materiales. Las piezas de trabajo se introducen cada cierto tiempo en la línea y se mueven de estación en estación. En cada estación se realiza repetidamente una operación.
MJS	<i>Minimum Job Set</i> , es el grupo mínimo de trabajos que mantiene la misma proporción de la producción.
MMAL	<i>Mixed Model Assembly Line</i> , o línea de montaje de productos mixtos.
MSP	<i>Mixed-model Sequencing Problem</i> .
Ocio	<i>Idle Time</i> . Inactividad del trabajador que ocurre cuando éste ha completado su tarea sobre un producto en su estación y debe esperar a que un nuevo producto entre en ella.
PR	<i>Path Relinking</i> o Re-encadenamiento de Trayectorias.
Patrón	Sub-secuencia de productos que se repite.
Sobrecarga	<i>Work Overload</i> . También, trabajo en exceso, Es el trabajo que no puede ser completado dentro de la estación cerrada. Equivale a la suma de las diferencias entre los instantes reales de compleción de las unidades en las estaciones y sus fechas de finalización comprometidas dependientes del valor de la ventana temporal.
Secuencia	Una secuencia de un conjunto de operaciones prescribe el orden en que las operaciones han de realizarse.
SA	<i>Simulated Annealing</i> o Recocido Simulado.
SS	<i>Scatter Search</i> o Búsqueda Dispersa.
Tiempo de Ciclo	Intervalo de tiempo constante entre el lanzamiento de dos unidades consecutivas en la línea.
Tiempo de Proceso	o también Tiempo de Procesamiento, es el tiempo necesario para realizar una operación o tarea, sobre el producto que se desplaza sobre la línea, dentro de una estación.

Trabajo Perdido	<i>Lost Work.</i> En una estación cerrada, es el trabajo que no ha podido completar el operario y que, debido a que la línea no puede ser detenida o no existen trabajadores de refuerzo que apoyen en estas situaciones, el trabajo debe ser completado fuera de la línea.
Trabajo Utilitario	<i>Utility Work.</i> Ocurre cuando en una estación cerrada, el operario no puede terminar su trabajo dentro de los límites de su estación, y un nuevo trabajador (utilitario o de refuerzo, <i>utility worker</i> ) se asigna para completar el trabajo. Se entiende por trabajo utilitario (trabajo en exceso o sobrecarga) al que debe realizar uno o más operarios en reserva y con intervención puntual cuando se produce una sobrecarga en alguna estación de la línea.
TS	<i>Tabu Search</i> o Búsqueda Tabu.

*And then you realize that all the things...  
...it's just all there to try to keep us away from death.  
"My life without me". Isabel Coixet, 2003.*

*"se hizo allí dios"*