

**ADVERTIMENT.** La consulta d'aquesta tesi queda condicionada a l'acceptació de les següents condicions d'ús: La difusió d'aquesta tesi per mitjà del servei TDX ([www.tesisenxarxa.net](http://www.tesisenxarxa.net)) ha estat autoritzada pels titulars dels drets de propietat intel·lectual únicament per a usos privats emmarcats en activitats d'investigació i docència. No s'autoritza la seva reproducció amb finalitats de lucre ni la seva difusió i posada a disposició des d'un lloc aliè al servei TDX. No s'autoritza la presentació del seu contingut en una finestra o marc aliè a TDX (framing). Aquesta reserva de drets afecta tant al resum de presentació de la tesi com als seus continguts. En la utilització o cita de parts de la tesi és obligat indicar el nom de la persona autora.

**ADVERTENCIA.** La consulta de esta tesis queda condicionada a la aceptación de las siguientes condiciones de uso: La difusión de esta tesis por medio del servicio TDR ([www.tesisenred.net](http://www.tesisenred.net)) ha sido autorizada por los titulares de los derechos de propiedad intelectual únicamente para usos privados enmarcados en actividades de investigación y docencia. No se autoriza su reproducción con finalidades de lucro ni su difusión y puesta a disposición desde un sitio ajeno al servicio TDR. No se autoriza la presentación de su contenido en una ventana o marco ajeno a TDR (framing). Esta reserva de derechos afecta tanto al resumen de presentación de la tesis como a sus contenidos. En la utilización o cita de partes de la tesis es obligado indicar el nombre de la persona autora.

**WARNING.** On having consulted this thesis you're accepting the following use conditions: Spreading this thesis by the TDX ([www.tesisenxarxa.net](http://www.tesisenxarxa.net)) service has been authorized by the titular of the intellectual property rights only for private uses placed in investigation and teaching activities. Reproduction with lucrative aims is not authorized neither its spreading and availability from a site foreign to the TDX service. Introducing its content in a window or frame foreign to the TDX service is not authorized (framing). This rights affect to the presentation summary of the thesis as well as to its contents. In the using or citation of parts of the thesis it's obliged to indicate the name of the author

UNIVERSIDAD POLITÉCNICA DE CATALUÑA

Programa de Doctorado:

AUTOMÁTICA, ROBÓTICA Y VISIÓN

Tesis Doctoral

**Programación de actividades en celdas  
robotizadas de tipo flowshop con buffers  
finitos y piezas distintas**

Juan Pablo Pendones Stoessel

**Directores:**

Dr. Albert Corominas Subias

Dr. Rafael Pastor Moreno

ENERO DE 2016



## **Agradecimientos**

A Ana Lucía y Emilia Belén, mis amores, mis soles, por su amor, su infinita paciencia y el apoyo en todos los momentos. A mis papás, Omar y Lilia, por su apoyo infinito y todo el amor. Muy especialmente a mis directores, los doctores Albert Corominas y Rafael Pastor por su guía permanente, su infinita paciencia y todo lo aprendido. Sin su ayuda constante este doctorado no hubiera sido posible.

Imposible olvidarme de Gustavo Arévalo. A él, por su invaluable colaboración. Una mano amiga a quien debo especial gratitud.

En el sur del mundo, a grandes amigos y compañeros de la UNICEN, Carlos Veruchi, Guillermo Santillán, Fernando Bengier, Oscar Pesci, Roberto Leegstra, Gustavo Kazlauskas, Dante Galván, Gerardo Acosta, Silvano Rossi, Raúl Rikal, Luis Arrién, Oscar Quiroga y Nora Blet, mi mayor reconocimiento a su apoyo permanente. Muy especialmente a los profesores Marcelo Spina y Roberto de la Vega, pues gracias ellos se pudo comenzar este doctorado.

Desde luego, a los amigos de toda la vida: Claudio Faizal, Marcelo Cabri, Gabriel Bulfón, Fabián Mantovani, Claudio Álvarez, Eduardo Gularte y Mauricio Falagán. Incondicionales que nos brindaron refugio cuando en 2002 llegamos con Emilia en un vientre. De ellos guardamos los más gratos recuerdos.

Aquí, donde escribo, en la latitud cero, muy especialmente a la familia Pacurucu, mi familia adoptiva: Saúl, Lucía, Fernanda y Saúl Andrés, por su paciencia, comprensión y ayuda permanente. Mi mayor gratitud para ellos, incondicionales siempre.

También a amigos: Rodrigo Sempértégui, Diego Ponce y María Toral, Juan Carlos Lazo y Rocío Vivar, Martha Cobos y Adrián Leon, Pablo Arce y Carmen Abril, Alfredo Mora y Patricia Chica, Juan Pablo Carballo y Marcela Andrade y Mónica López, por hacerme sobrellevar gratamente estos años de exilio voluntario.

Un poco más arriba, en el hemisferio norte, un gran agradecimiento Carlos Gaviria, gran compañero y amigo.

Al otro lado del mar, a todos los docentes y no docentes del IOC, quienes me brindaron la mayor colaboración en todo momento.



# Índice General

<b>Glosario de símbolos y términos</b>	<b>1</b>
<b>1 Marco teórico</b>	<b>9</b>
1.1 Introducción . . . . .	9
1.2 Las celdas robotizadas . . . . .	10
1.3 Función objetivo . . . . .	11
1.4 Número de máquinas . . . . .	14
1.5 Número de piezas y número de tipos de piezas . . . . .	14
1.6 Los tiempos del robot . . . . .	17
1.7 Patrón de flujo . . . . .	20
1.8 Restricciones por los tiempos del proceso . . . . .	20
1.9 Manipulador doble . . . . .	22
1.10 Múltiples robots . . . . .	23
1.11 Flexibilidad operacional . . . . .	23
1.12 Buffers . . . . .	23
1.13 Máquinas en paralelo . . . . .	24
<b>2 Problema y objetivos</b>	<b>25</b>
2.1 Introducción . . . . .	25
2.2 Definición del problema . . . . .	26

## Índice General

---

2.2.1	Notación de la estaciones . . . . .	29
2.2.2	Terminología de las operaciones . . . . .	30
2.2.3	Notación de los tiempos del robot . . . . .	30
2.3	Modelo de espacio de estados del problema . . . . .	32
2.3.1	Estado . . . . .	32
2.3.2	Operador del espacio de estados: movimiento del robot . . . . .	34
2.4	Análisis de movimiento-estado . . . . .	35
2.4.1	Diagrama de tiempos . . . . .	36
2.4.2	Análisis de movimientos . . . . .	37
2.4.3	Infactibilidad de un movimiento . . . . .	38
2.4.4	Condición de infactibilidad de un movimiento cuando la secuencia de piezas está predefinida . . . . .	42
2.4.5	El tiempo al final de un movimiento . . . . .	42
2.4.6	El estado al final del movimiento . . . . .	45
2.5	Funciones temporales auxiliares para las secuencias de operaciones del robot . . . . .	47
2.6	Cotas auxiliares . . . . .	48
2.6.1	Validez de las estaciones . . . . .	48
2.6.2	La cota de carga $\delta^c$ . . . . .	51
2.6.3	La cota de descarga $\delta^d$ . . . . .	52
2.6.4	La cota de descarga seguida de carga $\delta^e$ . . . . .	53
<b>3</b>	<b>Cota inferior basada en los tiempos de las máquinas cuando la secuencia de piezas es dada</b>	<b>55</b>
3.1	Introducción . . . . .	55
3.2	Grafo de acotación . . . . .	57
3.2.1	Cota del instante de salida $S_{i,k}$ . . . . .	58

## Índice General

---

3.2.2	Arcos tipo $I_k$ . . . . .	59
3.2.3	Arcos tipo $L_{i,k}$ . . . . .	60
3.2.4	Arcos tipo $T_{i,k}$ . . . . .	63
3.2.5	Arco tipo $U$ . . . . .	65
3.2.6	Procedimiento de acotación . . . . .	66
<b>4</b>	<b>Cota inferior basada en los tiempos de las máquinas cuando la secuencia de piezas no es dada</b>	<b>67</b>
4.1	Introducción . . . . .	67
4.2	Cota para las piezas que en un estado dado no ocupan la estación de entrada . . . . .	68
4.3	Cota para las piezas que en un estado dado ocupan la estación de entrada	68
4.3.1	Los tiempos de $\lambda_{i,g}^1$ . . . . .	69
4.3.2	Los tiempos de $\lambda_i^2$ . . . . .	71
4.3.3	Los tiempos de $\lambda_{i,h}^3$ . . . . .	73
4.4	Procedimiento . . . . .	75
<b>5</b>	<b>Cota inferior basada en los tiempos del robot</b>	<b>77</b>
5.1	Introducción . . . . .	77
5.2	Definición del problema relajado . . . . .	77
5.2.1	Condiciones de factibilidad de una secuencia de movimientos del robot . . . . .	77
5.3	Secuencia piramidal . . . . .	80
5.4	Procedimiento de cálculo de la cota inferior basada en los tiempos del robot . . . . .	81
5.5	Cota asociada a las piezas basada en rutas piramidales . . . . .	81
5.5.1	Caso de pieza alojada en una máquina. . . . .	81
5.5.2	Caso de pieza alojada en un buffer o en la estación de entrada. . .	83

## **Índice General**

---

5.6	Cifra de corrección . . . . .	84
5.7	Expresión final de la cota . . . . .	85
<b>6</b>	<b>Modelo lineal para la determinación de una secuencia de piezas</b>	<b>87</b>
6.1	Introducción . . . . .	87
6.2	Variables del modelo . . . . .	87
6.2.1	Instante posible de inicio de salida de las piezas . . . . .	87
6.2.2	Matriz de asignación . . . . .	88
6.3	Restricciones del modelo . . . . .	88
6.3.1	Asignación de una pieza única a cada posición de la secuencia de piezas . . . . .	88
6.3.2	Asignación de una posición única de la secuencia de piezas a cada pieza . . . . .	88
6.3.3	Instante posible de inicio de salida de la primera pieza de la primera máquina . . . . .	89
6.3.4	Precedencia entre piezas . . . . .	90
6.3.5	Precedencia entre máquinas . . . . .	91
6.3.6	Restricciones por el llenado de los buffers de capacidad no nula .	93
6.3.7	Restricciones por los buffers de capacidad nula . . . . .	95
6.4	Función objetivo . . . . .	97
6.5	Determinación de la secuencia de piezas . . . . .	97
6.6	Resumen del modelo . . . . .	97
<b>7</b>	<b>Procedimientos exactos y heurísticos de resolución del problema</b>	<b>101</b>
7.1	Introducción . . . . .	101
7.2	Procedimiento branch and bound . . . . .	101
7.2.1	Procedimiento <b>CrearDescendientes</b> . . . . .	103

## **Índice General**

---

7.3	Procedimiento branch and bound con poda heurística . . . . .	105
7.4	Procedimiento por separación . . . . .	107
7.4.1	Determinación de la secuencia de piezas . . . . .	107
7.4.2	Determinación de la secuencia de movimientos del robot . . . . .	109
7.5	Procedimiento por separación aleatorizado . . . . .	110
7.5.1	Determinación de la secuencia de piezas aleatorizada . . . . .	111
7.5.2	Determinación de la secuencia de movimientos del robot . . . . .	112
<b>8</b>	<b>Experiencia computacional</b>	<b>121</b>
8.1	Introducción . . . . .	121
8.2	Variantes de ejemplares . . . . .	121
8.2.1	Número de máquinas . . . . .	122
8.2.2	Número de piezas . . . . .	122
8.2.3	Capacidad de los buffers . . . . .	122
8.2.4	Tiempos de viaje . . . . .	123
8.2.5	Tiempos de carga y descarga . . . . .	127
8.2.6	Tiempos de proceso . . . . .	128
8.3	Ejemplares para la experiencia computacional . . . . .	129
8.4	Procedimientos . . . . .	129
8.5	Parámetros de los procedimientos . . . . .	130
8.5.1	Parámetros del procedimiento <b>BB</b> . . . . .	131
8.5.2	Parámetros del procedimiento <b>BBP</b> . . . . .	131
8.5.3	Parámetros del procedimiento <b>SEP</b> . . . . .	131
8.5.4	Parámetros del procedimiento <b>SEPA</b> . . . . .	132
8.6	Eficiencia de los procedimientos . . . . .	135

## **Índice General**

---

8.7	Análisis de los procedimientos de resolución . . . . .	136
8.7.1	Análisis de resultados para la variante de buffers nulos (variante <b>B1</b> ) . . . . .	137
8.7.2	Análisis de resultados de las variantes <b>B2</b> y <b>B3</b> . . . . .	138
8.7.3	Conclusiones generales . . . . .	140
8.8	Análisis de los óptimos. . . . .	142
8.9	Análisis de los procedimientos de acotación . . . . .	142
8.9.1	Análisis de resultados . . . . .	143
<b>9</b>	<b>Conclusiones</b>	<b>181</b>
9.1	Introducción . . . . .	181
9.2	Propuestas de procedimientos de resolución y acotación . . . . .	182
9.3	Resultados alcanzados . . . . .	183
9.3.1	Resultados de los procedimientos de acotación . . . . .	183
9.3.2	Resultados de los procedimientos de resolución . . . . .	183
9.4	Líneas futuras de investigación . . . . .	183

# Glosario de símbolos y términos

A continuación se muestran los términos y símbolos introducidos a lo largo del trabajo desarrollado en esta tesis. No es incluida la notación introducida en el capítulo 1 (marco teórico) donde los términos y símbolos de aquella notación es, en general, extraída de otras publicaciones.

- $n$  **Número de piezas.** Número de piezas a ser procesados por el sistema productivo.
- $m$  **Número de máquinas.** Número de máquinas que conforman el sistema productivo.
- $C_{max}$  **Makespan.** El instante de terminación del procesamiento de un conjunto de piezas (apartado 1.3).
- $P_j$  **Pieza  $j$ -ésima.** La pieza que ocupa la posición  $j$  del conjunto de piezas a ser procesadas en el sistema productivo.
- $M_i$  **Máquina  $i$ -ésima.** La máquina que ocupa la posición  $i$  del conjunto de máquinas del sistema productivo.
- $I$  **Estación de entrada de piezas.** Estación donde se alojan todas las piezas no procesadas.
- $O$  **Estación de salida de piezas.** Estación donde son depositadas todas las piezas terminadas.
- $B_i$  **Buffer  $i$ -ésimo.** El buffer que ocupa la posición  $i$  de un conjunto de  $m - 1$  buffers.
- $b_i$  **Capacidad del buffer  $i$ -ésimo.** Número de piezas que (como máximo) puede contener el buffer  $B_i$ .
- $E_k$  **Estación  $k$ -ésima.** La estación que ocupa la posición  $k$  del conjunto de  $2m + 1$  estaciones.
- entmed( $q$ ) Valor del cociente entero  $q/2$  (apartado 2.2.1).
- resmed( $q$ ) Valor del resto del cociente entero  $q/2$  (apartado 2.2.1).
- $t_{i,j}^c$  **Tiempo de carga.** Tiempo empleado en cargar la pieza  $P_j$  en la estación  $E_i$ .
- $t_{i,j}^d$  **Tiempo de descarga.** Tiempo empleado en descargar la pieza  $P_j$  de la estación  $E_i$ .

- $t_{h,i,j}^v$  **Tiempo de viaje.** Tiempo empleado por el robot transportando la pieza  $P_j$  desde la estación  $E_h$  hasta la estación  $E_i$ .
- $t_{h,i,0}^v$  **Tiempo de viaje sin pieza.** Tiempo empleado por el robot en un desplazamiento sin pieza desde la estación  $E_h$  hasta la estación  $E_i$ .
- $t_{i,j}^p$  **Tiempo de proceso.** Tiempo empleado en el procesamiento de la pieza  $P_j$  en la máquina  $M_i$ .
- $e, e_w$  **Estado.** Conjunto mínimo de variables que identifican el estado del sistema productivo de tipo celda robotizada analizado en esta tesis (apartado 2.3.1). El añadido de un subíndice (éstos pueden ser letras o números, por ejemplo  $w$ ) permite distinguir estados diferentes del sistema. Cuando se analiza un único estado se suele prescindir del subíndice por dado que su función se vuelve obsoleta.
- $r, r^w$  En la definición de estado, la variable cuyo valor indica el número de estación donde está ubicado el robot (apartado 2.3.1). Se añade un superíndice cuando se hace referencia a un estado concreto (por ejemplo se denota  $r^w$  cuando se hace referencia al estado  $e_w$ ).
- $\bar{l}, \bar{l}^w$  En la definición de estado, un vector formado por elementos  $l_j$ . El elemento  $l_j$  indica el número de estación donde está alojada la pieza  $P_j$  (apartado 2.3.1). Se añade un superíndice cuando se hace referencia a un estado concreto (por ejemplo se denotan  $\bar{l}^w$  y  $l_j^w$  cuando se hace referencia al estado  $e_w$ ).
- $\rho, \rho^w$  En la definición de estado, la variable que indica el instante de terminación del movimiento del robot que lleva al estado considerado (apartado 2.3.1). Se añade un superíndice cuando se hace referencia a un estado concreto (por ejemplo se denota  $\rho^w$  cuando se hace referencia al estado  $e_w$ ).
- $\bar{a}, \bar{a}^w$  En la definición de estado, un vector de elementos  $a_i$ . El elemento  $a_i$  indica el instante de terminación de la máquina  $M_i$  (apartado 2.3.1). Se añade un superíndice cuando se hace referencia a un estado concreto (por ejemplo se denotan  $\bar{a}^w$  y  $a_i^w$  cuando se hace referencia al estado  $e_w$ ).
- $\bar{\pi}, \bar{\pi}^w$  En la definición de estado, un vector formado por elementos  $\pi_k$ . El elemento  $\pi_k$  indica que la pieza  $P_{\pi_k}$  está ocupando la posición  $k$  en la secuencia  $\bar{\pi}$  (apartado 2.3.1). Se añade un superíndice cuando se hace referencia a un estado concreto (por ejemplo se denotan  $\bar{\pi}^w$  y  $\pi_k^w$  cuando se hace referencia al estado  $e_w$ ).
- $\mu, \mu_w$  **Movimiento.** Secuencia de operaciones del robot que modifica un cierto estado a partir del avance de una pieza dada un número de estaciones determinado (apartado 2.3.2). El añadido de un subíndice (éstos pueden ser letras o números, por ejemplo  $w$ ) permite distinguir movimientos diferentes. Cuando se analizan los efectos de un único movimiento se suele prescindir del subíndice dado que su función se vuelve obsoleta.

- $\epsilon, \epsilon_w$  Índice de la pieza que es transportada en un movimiento (apartado 2.3.2). Se añade un superíndice cuando se hace referencia a un movimiento concreto (por ejemplo se denota  $\epsilon_w$  cuando se hace referencia al movimiento  $\mu_w$ ).
- $\delta, \delta_w$  Número de estaciones en que una pieza dada es transportada en un movimiento (apartado 2.3.2). Se añade un superíndice cuando se hace referencia a un movimiento concreto (por ejemplo se denota  $\delta_w$  cuando se hace referencia al movimiento  $\mu_w$ ).
- $\bar{\mu}$  **Secuencia de movimientos del robot.** La solución del problema de esta tesis se expresa como una secuencia de movimientos del robot, donde el elemento  $\mu_k$  es el movimiento que ocupa la posición  $k$  en la secuencia  $\bar{\mu}$ .
- $N_\mu$  Número de movimientos de la secuencia de movimientos del robot  $\bar{\mu}$  (apartado 2.3.2).
- $F_{i,j}^w$  Determina si en el estado  $e_w$  la pieza  $P_j$  está ocupando el buffer  $B_i$  (ecuación 2.13, apartado 2.4.3).
- $f_i^w$  Determina el número de piezas que en el estado  $e_w$  están ocupando el buffer  $B_i$  (ecuación 2.14, apartado 2.4.3).
- $Z_{i,j}^w$  Determina si en el estado  $e_w$  la pieza  $P_j$  ya ha ingresado a la máquina  $M_i$  (ecuación 2.15, apartado 2.4.3).
- $z_i^w$  Determina el número de piezas que en el estado  $e_w$  ya han ingresado a la máquina  $M_i$  (ecuación 2.16, apartado 2.4.3).
- $\bar{\Pi}$  **Secuencia de piezas predefinida.** Secuencia de piezas que, en el procedimiento aleatorizado **SEP**, debe ser respetada por todas las soluciones factibles (apartado 2.4.4). Está formada por elementos  $\Pi_k$ , el cual indica que la pieza  $P_{\Pi_k}$  ocupa la posición  $k$  en la secuencia  $\bar{\Pi}$ .
- $\theta_c(i, j, q)$  **Función tiempo de secuencia de carga.** Tiempo empleado en la ejecución de la secuencia de operaciones del robot a cuyo término se produce la ocupación de la máquina  $M_i$  con la pieza  $P_j$ , cuando ésta procede de  $q$  estaciones anteriores a  $M_i$  (ecuación 2.34, apartado 2.5).
- $\theta_d(i, j, q)$  **Función tiempo de secuencia de descarga.** Tiempo empleado en la ejecución de la secuencia de operaciones del robot a cuyo término es alcanzada la disponibilidad de la máquina  $M_i$  (quedando ésta vacía), cuando la pieza  $P_j$  es adelantada  $q$  estaciones (ecuación 2.35, apartado 2.5).
- $h_{i,p}^-$  Expresa la validez de la estación anterior a la máquina  $M_i$ , distante de ésta  $p$  estaciones (ecuación 2.40, apartado 2.6.1).
- $h_{i,q}^+$  Expresa la validez de la estación siguiente a la máquina  $M_i$ , distante de ésta  $q$  estaciones (ecuación 2.41, apartado 2.6.1).
- $\delta_{i,j}^c$  **Cota de carga.** Cota del tiempo de fin de ingreso de la pieza  $P_j$  en la máquina  $M_i$  proveniente de una estación anterior (ecuación 2.44, apartado 2.6.2).

- $\delta_{i,j}^d$  **Cota de descarga.** Cota del tiempo de fin de la salida de la pieza  $P_j$  de la máquina  $M_i$  proveniente de una estación anterior (ecuación 2.47, apartado 2.6.3).
- $\delta_{i,j,k}^e$  **Cota de descarga seguida de carga.** Cota del tiempo de fin de ingreso de la pieza  $P_k$  en la máquina  $M_i$  cuando dicha máquina está previamente cargada con la pieza  $P_j$  (ecuación 2.52, apartado 2.6.4).
- $\mathcal{P}_{i,j}$  Tiempo de procesamiento de la pieza secuenciada en la  $j$ -ésima posición en la máquina  $M_i$  (apartado 3.1).
- $S$  Valor de la cota inferior del instante de terminación del programa, basada en los tiempos de las máquinas (ecuación 3.9 cuando la secuencia de piezas es dada y ecuación 4.19 cuando la secuencia de piezas no es dada).
- $S_{i,k}$  **Cota del instante de salida.** Cota inferior del instante en que es posible el inicio del movimiento que produce la salida de  $P_{\pi_k}$  (la pieza que ocupa la posición  $k$  en la secuencia de piezas  $\bar{\pi}$ ) de la máquina  $M_i$  (apartado 3.2.1).
- $I_k$  En un estado dado, la cota del instante de terminación de la pieza  $P_{\pi_k}$  en la siguiente máquina en la que debe ser procesada (ecuación 3.13, apartado 3.2.2).
- $L_{i,k}$  Cota asociada al tiempo transcurrido entre el fin del procesamiento de una pieza  $P_{\pi_{k-1}}$  en la máquina  $M_i$ , y el fin de procesamiento de la pieza siguiente  $P_{\pi_k}$  en la misma máquina (ecuación 3.24, apartado 3.2.3).
- $T_{i,k}$  Cota inferior del tiempo transcurrido entre el instante de terminación del procesamiento de una pieza  $P_{\pi_k}$  en la máquina  $M_{i-1}$  y el fin de procesamiento de la misma pieza en la máquina siguiente (ecuación 3.33, apartado 3.2.4).
- $U$  Tiempo empleado en las operaciones que producen la extracción en la última máquina  $M_m$  de la pieza secuenciada en la última posición  $n$  y su ingreso en la estación de salida (ecuación 3.34, apartado 3.2.5).
- $\tau$  Conjunto de piezas que en un estado dado ocupan la estación de entrada  $I$ .
- $\lambda_{i,g}^1$  Instante de inicio de procesamiento de las piezas de  $\tau$  en  $M_i$ , cuando la pieza  $P_g$  es la primera pieza del conjunto de piezas en  $\tau$  en salir de la estación  $I$  (apartado 4.3.1).
- $\lambda_i^2$  Tiempos de las máquinas y del robot asociados al procesamiento de las piezas de  $\tau$  en  $M_i$  (apartado 4.3.2).
- $\lambda_{i,h}^3$  Tiempos de las máquinas y del robot asociados al tiempo de procesamiento de la última pieza  $P_h$  en las máquinas siguientes a  $M_i$  (apartado 4.3.3).
- $\Lambda$  Cota basada en los tiempos del robot.
- $\Lambda_j$  Tiempo de la ruta piramidal asociada a la pieza  $P_j$ .

- $\omega$  **Cifra de corrección.** Corrige el supuesto de que al inicio de la primera secuencia de movimientos piramidal, el robot está situado en la última estación (apartado 5.6).
- $\bar{\Phi}$  Secuencia de piezas obtenida a partir de un modelo de programación lineal entera mixta (apartado 6.1).
- $\mathbb{S}$  **Matriz de instantes posibles de inicio de salida de las piezas.** Está formada por elementos  $s_{i,k}$  (apartado 6.2.1).
- $s_{i,k}$  Elemento de la matriz de instantes posibles de inicio de salida de las piezas  $\mathbb{S}$ . El elemento  $s_{i,k}$  representa el instante en que la pieza  $P_{\pi_k}$  (pieza secuenciada en la posición  $k$  de la secuencia de piezas) puede iniciar su salida de la máquina  $M_i$  (apartado 6.2.1).
- $\mathbb{X}$  **Matriz de asignación.** Está formada por elementos  $x_{j,k}$  (apartado 6.2.2).
- $x_{j,k}$  Elemento de la matriz de asignación  $\mathbb{X}$ . El elemento  $x_{j,k}$  vale 1 cuando la pieza  $P_j$  está signada en la posición  $k$  de la secuencia de piezas y vale 0 en el caso contrario (apartado 6.2.2).
- $\bar{\nabla}$  **Secuencia de piezas aleatorizada.** Secuencia de piezas que, en el procedimiento aleatorizado **SEPA**, debe ser respetada por todas las soluciones factibles. Se determina a partir de la secuencia de piezas predefinida  $\bar{\Pi}$  mediante un procedimiento aleatorizado (apartado 7.5.1). Está formada por elementos  $\nabla_k$ . El elemento  $\nabla_k$  indica que la pieza  $P_{\nabla_k}$  ocupa la posición  $k$  en la secuencia  $\bar{\nabla}$ .
- $\varrho_k$  **Índice de prioridad.** Índice asociados a la posición  $k$  de la secuencia predefinida  $\bar{\Pi}$  (ecuación 7.1, apartado 7.5.1).
- $p_k$  Probabilidad asociada a la posición  $k$  de la secuencia predefinida  $\bar{\Pi}$  (ecuación 7.2, apartado 7.5.1).
- $\gamma$  Parámetro que regula las distintas probabilidades  $p_k$  acercando o alejando sus valores (apartado 7.5.1).
- $x$  **Indicador de progreso.** Asociado a un estado dado, representa el grado de progreso en la determinación de una solución alcanzado por dicho estado. Se emplea en el contexto de un procedimiento de construcción iterativa de la solución (apartado 7.5.2).
- $X$  **Indicador de progreso máximo.** Define el valor de máximo admitido para el indicador de progreso para los estados enumerados en una iteración dada de un procedimiento de construcción iterativa de la solución (apartado 7.5.2).
- M2** Codificación de la variante de ejemplares de 2 máquinas (apartado 8.2.1).
- M3** Codificación de la variante de ejemplares de 3 máquinas (apartado 8.2.1).
- M4** Codificación de la variante de ejemplares de 4 máquinas (apartado 8.2.1).

- M5** Codificación de la variante de ejemplares de 5 máquinas (apartado 8.2.1).
- P05** Codificación de la variante de ejemplares de 5 piezas (apartado 8.2.2).
- P10** Codificación de la variante de ejemplares de 10 piezas (apartado 8.2.2).
- P15** Codificación de la variante de ejemplares de 15 piezas (apartado 8.2.2).
- P20** Codificación de la variante de ejemplares de 20 piezas (apartado 8.2.2).
- B1** Codificación de la variante de ejemplares de buffers de capacidad nula (apartado 8.2.3).
- B2** Codificación de la variante de ejemplares de buffers de capacidad  $\text{entmed}(m)$  (apartado 8.2.3).
- B3** Codificación de la variante de ejemplares de buffers de capacidad  $m$  (apartado 8.2.3).
- CD1** Codificación de la variante de ejemplares donde los tiempos de carga y descarga son pequeños o del mismo orden de magnitud que los tiempos de viaje (apartado 8.2.5).
- CD2** Codificación de la variante de ejemplares donde los tiempos de carga y descarga son grandes con respecto de los tiempos de viaje (apartado 8.2.5).
- PR1** Codificación de la variante de ejemplares donde los tiempos de proceso son pequeños o del mismo orden de magnitud frente a los tiempos de carga y descarga (apartado 8.2.6).
- PR2** Codificación de la variante de ejemplares con tiempos de proceso grandes frente a los tiempos de carga y descarga (apartado 8.2.6).
- BB** Codificación del procedimiento basado en branch and bound (apartado 7.2).
- BBP** Codificación del procedimiento basado en branch and bound con poda heurística (apartado 7.3).
- SEP** Codificación del procedimiento basado en la separación de la secuencia de piezas y la secuencia de movimientos del robot (apartado 7.4).
- SEPA** Codificación del procedimiento basado en la separación de la secuencia de piezas y la secuencia de movimientos del robot (apartado 7.5) donde la secuencia de piezas se determina por un procedimiento aleatorizado.
- $r_1^k$  **Error relativo porcentual de las soluciones referido a la mejor cota.** Para un procedimiento de resolución dado aplicado sobre un conjunto de ejemplares, consiste en la media aritmética de los errores relativos porcentuales calculados para la solución obtenida con respecto a la mejor cota de cada ejemplar (ecuación 8.28, apartado 8.6).

- $r_2^k$  **Error relativo porcentual de las soluciones referido a la mejor solución.** Para un procedimiento de resolución dado aplicado sobre un conjunto de ejemplares, es la media aritmética de los errores relativos porcentuales calculados para la solución obtenida con respecto a la mejor solución hallada para cada ejemplar (ecuación 8.29, apartado 8.6).
- $\eta_s^k$  **Porcentual de casos de mejor solución hallada.** Para un procedimiento de resolución dado aplicado sobre un conjunto de ejemplares, el número porcentual de veces que se obtiene la mejor solución con relación al número total de ejemplares. (ecuación 8.30, apartado 8.6).
- $\eta_o^k$  **Porcentual de casos de solución óptima hallada.** Para un procedimiento de resolución dado aplicado sobre un conjunto de ejemplares, consiste en el número porcentual de veces que se obtiene una solución óptima con relación al número total de ejemplares resueltos. (ecuación 8.31, apartado 8.6).
- $r_3^k$  **Error relativo porcentual de la cota referido a la mejor cota.** Para un procedimiento de acotación dado aplicado sobre un conjunto de ejemplares, consiste en la media aritmética de los errores relativos porcentuales calculados por dicho procedimiento con respecto a la mejor cota (ecuación 8.32, apartado 8.6).
- $\eta_c^k$  **Porcentual de casos de mejor cota hallada.** Para un dado procedimiento de acotación aplicado sobre un conjunto de ejemplares, consiste en el número porcentual de veces que se obtiene la mejor cota con relación al número total de ejemplares (ecuación 8.33, apartado 8.6).



# Capítulo 1

## Marco teórico

### 1.1 Introducción

Facilitado en gran medida por el desarrollo constante de los medios de software y hardware, emerge un tipo de sistemas de fabricación denominados sistemas de fabricación flexible (*flexible manufacturing systems*, FMS). Los sistemas de fabricación flexible se apoyan en dispositivos automáticos que pueden ser controlados por un computador, entre los cuales se pueden encontrar:

- ▶ **Máquinas de control numérico (*computer numerical control*, CNC):** están compuestas de una herramienta mecánica convencional para hacer las operaciones sobre las piezas y de un computador que controla la secuencia de los procesos realizados por la máquina.
- ▶ **Centros de mecanizado:** representan un nivel de automatización mayor que los CNC, dado que disponen de almacenes de herramientas y dispositivos automáticos que se pueden cambiar automáticamente.
- ▶ **Robots industriales:** son, esencialmente, máquinas programables con ciertas características similares a las del ser humano. Por lo general los robots son utilizados allí donde las condiciones de trabajo son en exceso rigurosas, peligrosas o insalubres, cuando hay que manipular elementos muy pesados, o bien cuando se requiere un alto grado de precisión, especialmente en tareas de tipo repetitivo.
- ▶ **Medios automatizados de manejo de materiales (*automated materials handling*):** mejoran la eficiencia en materia de transporte de materiales. Algunos ejemplos son los transportadores computarizados y los sistemas automatizados de almacenamiento y recuperación (*automatic storage and retrieval systems*, AS/RS) en los que las computadoras ordenan a los cargadores automáticos, recoger y colocar objetos.
- ▶ **Sistemas automatizados de vehículos guiados (*automated guided vehicles*, AGV):** utilizan diversos mecanismos para el guiado de vehículos sin conductor por distintos lugares de la planta.

Desde el punto de vista estructural, los sistemas de fabricación flexible suelen estar organizados en un conjunto de puestos de trabajo, relativamente independientes entre sí, pero vinculados mediante un sistema de control computarizado. En dichos puestos las máquinas están dispuestas según un esquema celular (generalmente en forma de U) y por ello son denominados celdas de fabricación flexible (*flexible manufacturing cell*, FMC). La manipulación de los materiales se realiza a través de medios automáticos, en la mayoría de los casos, mediante robots. Las celdas de fabricación flexible que son operadas por robots se conocen con el nombre de celdas robotizadas.

En contraste con los sistemas de producción en línea, por ejemplo, caracterizados por volúmenes de producto elevados, un tipo de pieza único, niveles básicos de automatización, una estructura rígida y maquinaria especializada y muy similar, los sistemas de fabricación flexible permiten un notable incremento de la diversidad de productos y la posibilidad de modificar los volúmenes de los productos producidos en forma ágil.

En este contexto cambiante, la tarea de programación de las actividades dentro de cada celda robotizada, es decir el problema de determinar el orden óptimo en que las piezas deben pasar por cada máquina y los instantes de inicio y terminación de procesamiento de cada pieza en cada máquina con el fin de maximizar la productividad, se convierte en un problema clave.

### 1.2 Las celdas robotizadas

Las celdas robotizadas son sistemas productivos en los que se procesan piezas y donde el traslado de piezas entre las máquinas que las procesan se realiza mediante un robot. Los problemas de celdas robotizadas se circunscriben a los problemas de programación de actividades donde:

- ▶ se asumen los tiempos de carga y descarga de las piezas en las máquinas, y los tiempos de viaje entre las máquinas, mayores que cero, y
- ▶ si el número de robots es inferior al número de máquinas, como por lo general ocurre en casos reales, el robot constituirá un recurso de transporte limitado y podrán tener lugar tiempos de espera de las piezas en las máquinas.

Se produce una variedad de problemas que derivan de diversos supuestos sobre aspectos físicos y operacionales del sistema productivo, muchos de ellos históricamente tratados en los problemas de programación clásicos donde los tiempos de transporte, carga y descarga son de magnitudes pequeñas respecto de los tiempos de proceso, tal que admiten ser despreciados.

A fin de abarcar todas las variantes observadas en la literatura en un concepto suficientemente amplio, las celdas robotizadas se pueden definir como sistemas productivos conformados por:

- ▶ un conjunto de puestos formados por una o más máquinas,
- ▶ uno o más robots, cada uno con uno o más manipuladores, quienes realizan el transporte de piezas y operaciones de carga y descarga en las máquinas,

- ▶ buffers de capacidad mayor o igual que cero, situados entre pares de puestos consecutivos que sirven de almacenamiento temporal de las piezas en proceso,
- ▶ una estación de ingreso de piezas
- ▶ y una estación de salida donde las piezas terminadas son depositadas.

Existen otros sistemas productivos en los que, al igual que en las celdas robotizadas, se tienen recursos de transporte limitados y tiempos de transporte, carga y de descarga, de magnitudes tales que no admiten ser despreciados frente a los tiempos de proceso, como los siguientes:

- ▶ Modernos sistemas de manufactura realizan el traslado de las piezas mediante vehículos autoguiados que se desplazan de modo autónomo por la planta. Diversos problemas de programación en sistemas servidos por vehículos autoguiados se analizan en Deroussi *et al.* (2008), Gnanavel Babu *et al.* (2010), Erol *et al.* (2012), Lacomme *et al.* (2013), Pan *et al.* (2013) y Bocewicz *et al.* (2014).
- ▶ El problema del puente grúa (*Hoist scheduling problem*) consiste en un sistema productivo donde diferentes baños químicos se aplican sobre un conjunto de piezas y donde se emplea un puente grúa para el transporte de las piezas entre los distintos baños. En estos problemas, las características del proceso suelen imponer restricciones de tipo temporal dado que la permanencia excesiva de las piezas en estos baños puede producir efectos indeseados. Las distintas variantes del problema del puente grúa son analizados por diversos enfoques en Mateo (2001), Mateo *et al.* (2002), Hindi y Fleszar (2004), Jégou *et al.* (2006), Zhou y Liu (2008), Amraoui *et al.* (2013a), Amraoui *et al.* (2013b) y Chtourou *et al.* (2013).
- ▶ En la manufactura de semiconductores se emplean celdas robotizadas, denominadas grupos de herramientas (*cluster tools*), formadas por uno o más robots con uno o más manipuladores que realizan la transferencia de las obleas semiconductoras entre diferentes cámaras donde las obleas reciben diversos tratamientos químicos. Son muy estudiados los problemas de grupos de herramientas múltiples (*multicluster tools*) donde múltiples celdas se asocian entre sí mediante buffers de pequeña capacidad, comúnmente de una o dos posiciones. Cada buffer es compartido por dos celdas vecinas actuando como estación de salida de piezas de una de las celdas y, simultáneamente, como estación de entrada de piezas de la otra celda. Sobre este problema puede consultarse: Tzeng *et al.* (2006), Jung y Lee (2012), Tonke y Lee (2012) y Lim *et al.* (2014).

Una diversidad de supuestos sobre características físicas y funcionales de las celdas robotizadas han sido adoptados por diferentes autores, dando lugar a una gama de variantes de problemas que serán analizadas entre los apartados 1.3 y 1.13.

### 1.3 Función objetivo

La literatura coincide en adoptar como objetivo de optimización la minimización del tiempo de producción. Este objetivo, en el contexto de sistemas productivos, asociado al procesamiento de un conjunto de piezas, es conocido con el nombre de *makespan*

y simbolizado  $C_{max}$ . En problemas de celdas robotizadas, el makespan consiste en el tiempo transcurrido desde un estado inicial caracterizado por una configuración donde todas las piezas y el robot están en la estación de entrada, y un estado final caracterizado por una configuración donde todas las piezas terminadas y el robot están en la estación de salida.

Por otro lado, una simplificación adoptada frecuentemente parte del supuesto de que el número de piezas es elevado, lo que justifica la división del conjunto total de piezas en un subconjunto de tamaño menor. Así, el objetivo consiste en la determinación de un programa de operaciones del robot para este subconjunto de piezas con la condición de que dicho programa pueda repetirse cíclicamente. El tiempo ocupado por este programa cíclico, en el contexto de problemas de celdas robotizadas, se denomina *tiempo de ciclo* y es denotado  $C_t$ .

Dado un conjunto  $C^T$  de piezas a ser procesado mediante un enfoque cíclico, son posibles dos estrategias sobre las cuales basar el diseño del programa para el procesamiento del total de piezas  $C^T$ . La primera estrategia consiste en un programa de tres fases como se muestra en la figura 1.1:

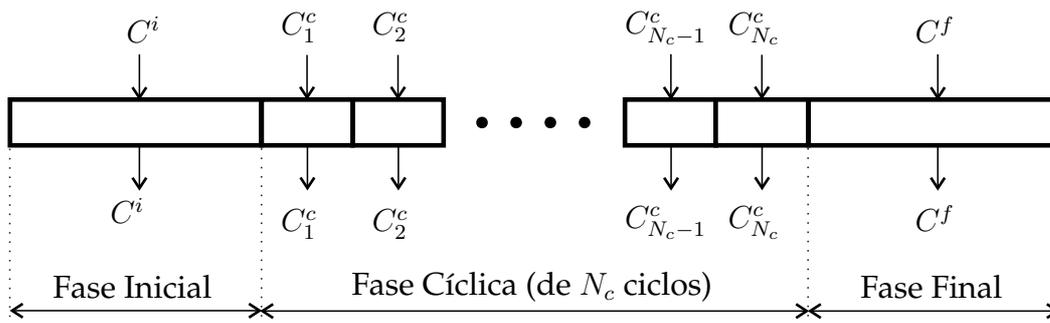


Figura 1.1: Programa de tres fases para el procesamiento de un conjunto de piezas basado en un enfoque cíclico donde se procesan  $N_c$  subconjuntos  $C^c$ .

- ▶ una fase inicial, donde, por ejecución de una determinada secuencia de operaciones del robot, ingresa al sistema un subconjunto de piezas  $C^i$  (subconjunto de piezas de la fase inicial) y el mismo subconjunto de piezas abandona el sistema (es depositado en la estación de salida tras haber sido procesado en todas las máquinas) en el transcurso de esta fase inicial;
- ▶ una fase cíclica formada por  $N_c$  ciclos idénticos, donde en cada uno de estos ciclos se procesa un subconjunto de piezas denotado  $C^c$ , de un modo tal que, si  $C^c_i$  denota el subconjunto de piezas ingresado en el  $i$ -ésimo ciclo, es el mismo subconjunto de piezas  $C^c_i$  que en dicho ciclo egresa del sistema;
- ▶ y una fase final, donde, por ejecución de una determinada secuencia de operaciones del robot, ingresa al sistema un subconjunto de piezas de la fase final denotado  $C^f$  y donde en dicha fase el mismo subconjunto de piezas abandona el sistema (es depositado en la estación de salida después de haber sido procesado en todas las máquinas) en el transcurso de esta fase final.

## Función objetivo

---

Bajo esta estrategia se verifica la igualdad de la ecuación 1.1:

$$C^T = C^i + N_c C^c + C^f \quad (1.1)$$

La segunda estrategia (también de tres fases) se muestra en la figura 1.2:

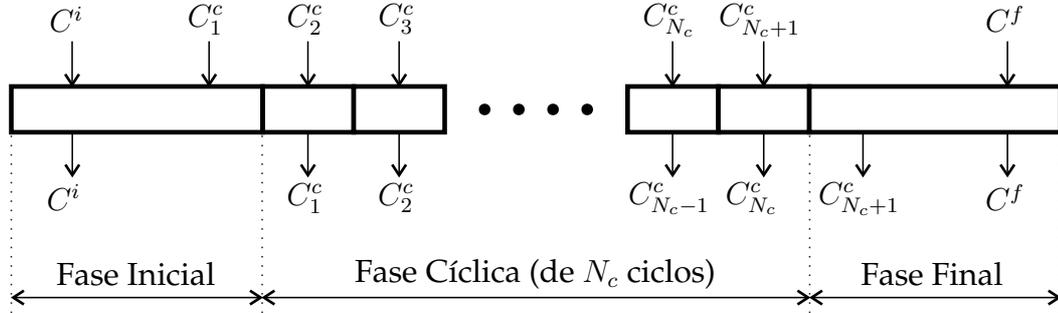


Figura 1.2: Programa de tres fases para el procesamiento de un conjunto de piezas basado en un enfoque cíclico donde se procesan  $N_c + 1$  subconjuntos  $C^c$ .

- ▶ una fase inicial, donde, mediante la ejecución de una determinada secuencia de operaciones del robot:
  - ▷ ingresan al sistema un subconjunto piezas  $C^i$  (subconjunto de piezas de la fase inicial, posiblemente igual a conjunto vacío) y el primer subconjunto de piezas de la fase cíclica  $C_1^c$ ,
  - ▷ mientras que egresa del sistema el mismo subconjunto piezas  $C^i$  ingresado en dicha fase;
- ▶ una fase cíclica formada por  $N_c$  ciclos idénticos, donde en cada uno de estos ciclos se procesa un subconjunto de piezas denotado  $C^c$ , de un modo tal que, en el  $i$ -ésimo ciclo ingresa al sistema el subconjunto de piezas  $C_i^c$  mientras que en el mismo ciclo egresa del sistema el subconjunto  $C_{i-1}^c$  (que ha ingresado en el ciclo anterior) ( $i = 1, \dots, N_c$ );
- ▶ y de una fase final, donde, por ejecución de una determinada secuencia de operaciones del robot,
  - ▷ ingresa al sistema un subconjunto piezas  $C^f$  (subconjunto de piezas de la fase final, posiblemente igual a conjunto vacío)
  - ▷ mientras que egresan del sistema el mismo subconjunto piezas  $C^f$  ingresado en dicha fase y el subconjunto de piezas  $C_{N_c}^c$  que ingresó al sistema en el último ciclo de la fase cíclica.

Este esquema verifica la igualdad de la ecuación 1.2:

$$C^T = C^i + (N_c + 1) C^c + C^f \quad (1.2)$$

No debe perderse de vista que el enfoque basado en la separación en las 3 fases explicadas es una aproximación y, por ello, no puede asegurarse la optimalidad de la solución en comparación con el enfoque no cíclico, basado en la determinación del *makespan*.

Con respecto al enfoque cíclico de problemas de celdas robotizadas, se observan abundantes aportaciones enfocadas exclusivamente en fase cíclica. Sin embargo no se encuentran propuestas para la resolución óptima de las fases inicial y final con las que se deberían completar el enfoque cíclico.

Por otro lado, con relación al objetivo de minimización del tiempo de ciclo  $C_t$ , el problema donde el objetivo consiste en la minimización del *makespan*  $C_{max}$  es escasamente tratado. Diversos casos de minimización del *makespan* en celdas robotizadas son analizados en Chu y Proth (1996), Błażewicz *et al.* (2002), Lebacque y Brauner (2005), Lee y Strusevich (2005), Soukhal y Martineau (2005), Steiner y Xue (2005), Tzeng *et al.* (2006), Werner y Kravchenko (2010), Yan *et al.* (2010c), Kim y Lee (2012), Zhang *et al.* (2012) y Elmi y Topaloglu (2013).

### 1.4 Número de máquinas

En los casos prácticos, el número de máquinas suele ser relativamente pequeño y por esto se suelen considerar problemas de sólo dos o tres máquinas. Son escasos los modelos en los que el número de máquinas es incorporado al modelo como una variable.

Diversos problemas de celdas robotizadas de dos máquinas son tratados en Agnetis (2000), Błażewicz *et al.* (2002), Drobouchevitch *et al.* (2004), Sriskandarajah *et al.* (2004), Akturk *et al.* (2005), Geismar *et al.* (2005b), Lebacque y Brauner (2005), Lee y Strusevich (2005), Gultekin *et al.* (2006a), Gultekin *et al.* (2008), Kimiagari y Mosadegh (2011), Batur *et al.* (2012), Geismar *et al.* (2012), Fathian *et al.* (2013) y Fazel Zarandi *et al.* (2013).

Problemas de celdas robotizadas de tres máquinas son tratados en Hall *et al.* (1997), Crama y van de Klundert (1999), Kamoun *et al.* (1999), Agnetis (2000), Agnetis y Pacciarelli (2000), Brauner y Finke (2001), Gultekin *et al.* (2006b), Kamalabadi *et al.* (2007), Gultekin *et al.* (2008), Geismar *et al.* (2012) y Zahrouni y Kamoun (2012).

Son escasamente tratados los problemas de celdas robotizadas de cuatro máquinas: Kamoun *et al.* (1999), Brauner y Finke (2001) y Abdulkader *et al.* (2013).

Distintos problemas de celdas robotizadas, donde el número de máquinas se incorpora como variable del modelo, son propuestos por: Crama y van de Klundert (1997), Levner *et al.* (1997), Agnetis (2000), Sethi *et al.* (2001), Sriskandarajah *et al.* (2004), Kumar *et al.* (2005), Drobouchevitch *et al.* (2006), Kats y Levner (2006), Gultekin *et al.* (2009), Werner y Kravchenko (2010), Geismar *et al.* (2011), Kats y Levner (2011a), Kats y Levner (2011b) y Yildiz *et al.* (2012).

### 1.5 Número de piezas y número de tipos de piezas

Respecto del número de tipos de piezas, los problemas se clasifican en: problemas donde todas las piezas son del mismo tipo (piezas iguales), y problemas con tipos de piezas

múltiples (piezas distintas).

**Problemas de piezas distintas.** En los problemas de piezas distintas se tienen  $N_Q$  tipos de piezas  $Q_1, Q_2, \dots, Q_j, \dots, Q_{N_Q}$ . El conjunto total piezas a ser procesado  $C^T$  se representa mediante un vector  $\bar{C}^T$  definido como

$$\bar{C}^T = (C_1^T, C_2^T, \dots, C_j^T, \dots, C_{N_Q}^T) \quad (1.3)$$

donde cada elemento  $C_j^T$  del vector  $\bar{C}^T$  representa el número de piezas de tipo  $Q_j$  a ser procesadas por el sistema<sup>1</sup>.

En el caso de piezas distintas y bajo el enfoque cíclico, a la hora de definir el conjunto de piezas a ser procesadas en cada ciclo se suele emplear el concepto de *conjunto mínimo de piezas* o MPS por sus siglas en inglés (*Minimal Part Set*). Como su nombre indica es un conjunto mínimo, tal que las cantidades de piezas de cada tipo guardan una relación constante al conjunto total de piezas a ser procesadas en la fase cíclica. Por ejemplo, sea el vector  $\bar{C}^T$  que representa el total de piezas a ser procesado en las 3 fases del enfoque cíclico definido como:

$$\bar{C}^T = (205, 307, 101) \quad (1.4)$$

Una alternativa puede consistir en expresar este vector como la siguiente suma:

$$\bar{C}^T = (205, 307, 101) = (200, 300, 100) + (5, 7, 1) \quad (1.5)$$

Por una parte el conjunto de piezas representado por el vector  $(5, 7, 1)$  puede ser procesado por completo en las fases inicial y final, es decir:

$$\bar{C}^i + \bar{C}^f = (5, 7, 1) \quad (1.6)$$

Y por otra parte el conjunto de piezas representado por el vector  $(200, 300, 100)$  puede procesarse en la fase cíclica y puede expresarse como la relación de vectores siguientes:

$$(200, 300, 100) = 100 (2, 3, 1) \quad (1.7)$$

El conjunto de piezas representado por el vector  $(2, 3, 1)$  es el vector  $\overline{\text{MPS}}$  (que representa el conjunto MPS) pues no hay factores comunes entre sus elementos.

$$\overline{\text{MPS}} = (2, 3, 1) \quad (1.8)$$

---

<sup>1</sup>Se emplea la aritmética vectorial para representar los conjuntos de piezas dado que permite realizar operaciones no disponibles en operaciones con conjuntos.

Una vez hallado el MPS, el conjunto de piezas a ser procesado en cada ciclo se define como:

$$\bar{C}^c = k \overline{\text{MPS}} \quad (1.9)$$

o también (expresado como relación de conjuntos):

$$C^c = k \text{ MPS} \quad (1.10)$$

donde el factor  $k$  se denomina grado de ciclo y determina el número de conjuntos MPS que son procesados a lo largo de cada ciclo.

Con respecto al grado de ciclo  $k$ , en la literatura se presentan dos planteamientos:

- ▶ monogrado: donde se asume un valor constante del grado  $k$ , por lo general de valores 1 ó 2 por ser los más simples;
- ▶ multigrado: donde el grado del ciclo  $k$  es una variable del problema, y la solución admite todos los valores de  $k$  posibles ( $k = 1, 2, \dots$ ).

Los programas de un MPS, o de grado 1 ( $k = 1$ ), son los más simples y por esta razón muchos autores suelen limitar la búsqueda de soluciones a aquellas cuyos ciclos son de grado 1, aunque estas no sean necesariamente óptimas frente a programas de grado mayor.

En el caso de piezas distintas no se observan publicaciones *multigrado* como tampoco enfoques de un solo grado en donde  $k > 1$ .

Problemas de piezas distintas son analizados en McCormick *et al.* (1989), Sriskandarajah y Sethi (1989), Sethi *et al.* (1992), Logendran y Sriskandarajah (1996), Hall *et al.* (1997), Hall *et al.* (1998), Sriskandarajah *et al.* (1998), Kamoun *et al.* (1999), Sriskandarajah *et al.* (2004), Lebacque y Brauner (2005), Kamalabadi *et al.* (2007), Kamalabadi *et al.* (2008), Batur *et al.* (2012), Geismar *et al.* (2012) y Abdulkader *et al.* (2013).

**Problemas de piezas iguales.** El caso de piezas iguales es el caso particular de piezas distintas donde el número de tipos de piezas  $N_Q$  tiene valor 1.

En los problemas de piezas iguales son escasos los trabajos que tratan problemas denominados *multigrado* donde son abarcados todos los grados posibles ( $1 \leq k \leq N_p$ ). Por el contrario es frecuente limitar las soluciones a programas de un sólo valor del grado  $k$ , principalmente los casos donde el grado es pequeño, por lo general  $k = 1$  y  $k = 2$ , por proveer programas más simples.

Problemas de grado 1 son analizados en Dawande *et al.* (2002), Akturk *et al.* (2005), Steiner y Xue (2005) y Drobouchevitch *et al.* (2006).

Problemas de grado 2 son analizados en Chu (2006), Kats y Levner (2006), Kats y Levner (2009), Levner y Kats (2009), Saez (2009), Che *et al.* (2011b), Che *et al.* (2011a), Kats y Levner (2011a), Kats y Levner (2011b) y Amraoui *et al.* (2013b).

Problemas multigrado son analizados en Geismar (2003), Che y Chu (2005), Geismar *et al.* (2005a), Brauner (2008) y Che y Chu (2009).

### 1.6 Los tiempos del robot

En los tiempos del robot se abarcan los tiempos de viaje, los tiempos de carga y los tiempos de descarga de las piezas en las distintas estaciones (máquinas, buffers y estaciones de entrada y salida de piezas).

En este apartado se analizarán distintos modelos adoptados para representar los tiempos del robot. Para ello, en este apartado se propone un *modelo de referencia* de estos tiempos intentando incorporar los factores que en problemas reales puedan incidir sobre estos tiempos, para luego, establecer en qué grado los distintos modelos de la literatura se aproximan a las condiciones incluidas en el modelo de referencia.

Se analizarán el modelo de referencia  $\mathcal{M}_0$  y los modelos de tiempos del robot de otros autores  $\mathcal{M}_i (i = 1, \dots)$ . En cada modelo se contabilizará el número de parámetros utilizados para definir los tiempos del robot. Un mayor número de estos parámetros dará al modelo una mayor capacidad de aproximar sus tiempos a los tiempos que puedan provenir de casos reales.

Para el análisis de los modelos se define la siguiente notación:

- ▶  $N_E$ : el número total de estaciones (en donde se abarcan las máquinas, los buffers, la estación de entrada y la estación de salida);
- ▶  $N_Q$ : el número de tipos de piezas a procesar;
- ▶  $\mathcal{R}_i^c$ : el número de parámetros usados por el modelo  $\mathcal{M}_i$  para modelizar los tiempos de carga en las  $N_E$  estaciones;
- ▶  $\mathcal{R}_i^d$ : el número de parámetros usados por el modelo  $\mathcal{M}_i$  para modelizar los tiempos de descarga en las  $N_E$  estaciones;
- ▶  $\mathcal{R}_i^{cd}$ : el número de parámetros usados por el modelo  $\mathcal{M}_i$  para modelizar los tiempos de carga y los tiempos de descarga en las  $N_E$  estaciones;
- ▶  $\mathcal{R}_i^v$ : el número de parámetros empleado por el modelo  $\mathcal{M}_i$  para modelizar los tiempos de viaje entre estaciones;
- ▶  $\mathcal{R}_i$ : el número total de parámetros empleado por el modelo  $\mathcal{M}_i$  para modelizar los tiempos del robot.

**Modelo de referencia**  $\mathcal{M}_0$ . En este modelo se incorporan características físicas (peso) y geométricas (formas y distancias) que tienen efecto sobre los tiempos del robot. En el modelo se asume que:

- ▶ Los tiempos de carga deben ser dependientes de las propiedades físicas y geométricas del tipo de pieza y dependientes de las propiedades físicas y geométricas de cada estación. Por esto, el modelo debe definir un tiempo de carga para cada tipo de pieza y para cada estación. Por lo tanto, un número de parámetros igual al producto:

$$\mathcal{R}_0^c = N_E N_Q \quad (1.11)$$

- ▶ También los tiempos de descarga deben ser dependientes de las propiedades físicas y geométricas del tipo de pieza y, simultáneamente, dependientes de las propiedades físicas y geométricas de cada estación. Por esto el modelo debe definir un tiempo de descarga para cada tipo de pieza y para cada estación. Por lo tanto, un número de parámetros igual al producto:

$$\mathcal{R}_0^d = N_E N_Q \quad (1.12)$$

De la suma de 1.11 y 1.12 se obtiene el número de parámetros empleados para modelizar tiempos de carga y descarga  $\mathcal{R}_0^{cd}$  del modelo  $\mathcal{M}_0$ , mediante la expresión 1.13.

$$\mathcal{R}_0^{cd} = 2 N_E N_Q \quad (1.13)$$

- ▶ Los tiempos de viaje deben ser dependientes de las propiedades físicas y geométricas del tipo de pieza y de la trayectoria a ser recorrida por el robot. Se define un tiempo de viaje para todo par de estaciones, todo tipo de pieza y el movimiento del robot sin pieza. Por lo tanto, el número de tiempos de viaje debe ser:

$$\mathcal{R}_0^v = \frac{N_E (N_E - 1) (N_Q + 1)}{2} \quad (1.14)$$

**Modelo**  $\mathcal{M}_1$ . Dentro de los modelos observados en la literatura, los modelos más completos corresponden a los trabajos de los autores de Hall *et al.* (1997) y Kamoun *et al.* (1999). Respecto de los tiempos de carga y descarga, su modelo diferencia los tiempos de carga y descarga de una máquina a otra. Sin embargo, no incorporan en su modelo la influencia de características físicas y geométricas de las piezas. De modo que, el número de parámetros que emplean para modelizar los tiempos de carga y descarga suman en total:

$$\mathcal{R}_1^{cd} = 2N_E \quad (1.15)$$

Con respecto a los tiempos de viaje, en estos modelos se consideran  $N_E - 1$  tiempos que representan los tiempos de viajes entre todo par de estaciones adyacentes. Los tiempos de viaje entre estaciones no adyacentes se determinan sumando los tiempos de viaje

asociados a los pares de estaciones adyacentes que están situadas entre la estación de partida y la estación de llegada. A continuación el tiempo hallado se ajusta mediante un tiempo  $\gamma$  de valor constante, el cual es restado tantas veces como estaciones haya entre la estación de partida y la estación de destino, para considerar el tiempo ahorrado por el robot por no haberse detenido en las estaciones intermedias y tener en cuenta los tiempos debido a las aceleraciones y desaceleraciones del robot. El tiempo  $\gamma$  es un parámetro adicional a los  $N_E - 1$  que ya fueron considerados, por lo que el número total de parámetros empleados por estos autores para modelizar los tiempos de viaje suman:

$$\mathcal{R}_1^v = N_E \quad (1.16)$$

**Modelo  $\mathcal{M}_2$ .** Los modelos propuestos en Crama y van de Klundert (1999) simplifican aún más el modelo de tiempos del robot. En efecto, definen iguales valores de los tiempos de descarga y descarga en toda estación, es decir,  $N_E$  tiempos en total para los tiempos de carga y descarga.

$$\mathcal{R}_2^{cd} = N_E \quad (1.17)$$

Por otro lado, para los tiempos de viaje definen  $N_E - 1$  tiempos para representar los tiempos de viaje entre pares de estaciones adyacentes, sin embargo no incorporan parámetros de ajuste para tener en cuenta los ahorros de tiempo en desaceleración y aceleración del robot cuando, respectivamente, se detiene y parte en cada estación. De aquí que el número total de parámetros empleados por estos autores para modelizar los tiempos de viaje sumen:

$$\mathcal{R}_2^v = N_E - 1 \quad (1.18)$$

**Modelo  $\mathcal{M}_3$ .** Una simplificación mayor se introduce en Sethi *et al.* (1992), Mangione *et al.* (2003), Gultekin *et al.* (2006a), Gultekin *et al.* (2007), Gultekin *et al.* (2008), Gultekin *et al.* (2009) y Gultekin *et al.* (2010). Estos autores suponen en toda máquina la misma magnitud del tiempo de carga y del tiempo de descarga, y que esta magnitud es la misma en todas las máquinas. De esta forma:

$$\mathcal{R}_3^{cd} = 1 \quad (1.19)$$

Suponen también que los tiempos de viaje entre todo par de estaciones adyacentes son iguales:

$$\mathcal{R}_3^v = 1 \quad (1.20)$$

**Resumen.** En la tabla 1.1 se muestra cómo el número total de parámetros empleados en los distintos modelos se va reduciendo a la hora de considerar los tiempos del robot.

Modelo	$\mathcal{R}_i$
$\mathcal{M}_0(i = 0)$	$2N_E N_Q + \frac{N_E (N_E - 1) (N_Q + 1)}{2}$
$\mathcal{M}_1(i = 1)$	$3N_E$
$\mathcal{M}_2(i = 2)$	$2N_E - 1$
$\mathcal{M}_3(i = 3)$	$2$

Tabla 1.1: Número total de parámetros empleados para modelizar los tiempos del robot en los distintos modelos.

## 1.7 Patrón de flujo

En un sistema productivo, la ruta, definida como el orden o secuencia de máquinas o puestos por donde las piezas son procesadas, se denomina *patrón de flujo*.

Los menos restrictivos son los sistemas productivos llamados de tipo *openshop*. En éstos todas las rutas son permitidas. Problemas de celdas robotizadas con patrón de flujo *openshop* son analizados en Hall y Sriskandarajah (1996), Lee y Strusevich (2005) y Lushchakova *et al.* (2009).

En el caso particular en que la secuencia de máquinas que es recorrida por las piezas es la misma para todas las piezas, el patrón de flujo de este sistema de producción es denominado *flowshop* o *patrón de flujo regular*. Cuando además se verifica que la secuencia de piezas observada por las máquinas es la misma para todas ellas, esta variante de *flowshop* se denomina *flowshop* permutativo o *flowshop tipo P*. Problemas de celdas robotizadas con patrón de flujo *flowshop* son analizados en Sriskandarajah y Sethi (1989), Levner *et al.* (1997), Crama *et al.* (2000), Sriskandarajah *et al.* (2004), Che y Chu (2005), Dawande *et al.* (2005), Kumar *et al.* (2005), Lebacque y Brauner (2005), Soukhal y Martineau (2005), Chu (2006), Gultekin *et al.* (2007), Gultekin *et al.* (2008), Che *et al.* (2011b), Che *et al.* (2012), Zahrouni y Kamoun (2012) y Zhou *et al.* (2012).

El patrón de flujo de tipo *jobshop*, o flujo general, es aquél en que para cada pieza existe definida una secuencia de máquinas propia. Problemas de celdas robotizadas con patrón de flujo *jobshop* son analizados en Kubiak (1989), Kamoun y Sriskandarajah (1993), Hall y Sriskandarajah (1996), Seo y Lee (2002), Brucker y Kampmeyer (2008), Levner *et al.* (2010), Zhang *et al.* (2012) y Lacomme *et al.* (2013).

## 1.8 Restricciones por los tiempos del proceso

Las características del proceso pueden imponer restricciones adicionales al problema de programación. Estas restricciones provienen de condiciones que deben cumplirse sobre los procesos aplicados a las piezas. Éstas tienen su origen en la naturaleza misma del proceso y por lo general están ligadas a procesos térmicos y químicos.

Estas restricciones se aplican sobre el tiempo de permanencia de la pieza en proceso  $t_p$  y sobre el tiempo transcurrido entre procesos sucesivos  $t_w$ . El tiempo de permanencia en proceso se define como el tiempo en que la pieza está en una máquina. Abarca tanto el tiempo en que la pieza es procesada en la máquina como el tiempo de espera en la misma máquina, donde ésta funciona como depósito transitorio de la pieza. El tiempo entre procesos sucesivos  $t_w$  se define como el tiempo transcurrido desde el instante de salida de la pieza de una máquina dada y el instante de ingreso de la misma pieza en la máquina siguiente.

Las restricciones definidas sobre el tiempo  $t_w$  buscan representar condiciones que tienen su origen en una o más propiedades que la pieza ha adquirido en el proceso anterior, como temperatura, adherencia de un pegamento, etc., que se reduce a través del tiempo, y que debe asegurarse al inicio del proceso en curso.

Por otra parte, las restricciones definidas sobre el tiempo de permanencia en proceso  $t_p$  buscan representar condiciones que tienen su origen en una o más propiedades que la pieza ha adquirido en la máquina en curso, como en ciertos procesos químicos o térmicos.

A continuación se analizan algunos casos que se presentan en ambientes industriales modelizados en la literatura.

**Ventanas temporales.** En los casos modelizados por ventanas temporales, a cada máquina le corresponde un tiempo de permanencia en proceso acotado inferiormente por un tiempo de proceso mínimo  $\mathcal{P}_{min}$  y superiormente por un tiempo de proceso máximo  $\mathcal{P}_{max}$ . En resumen, el tiempo de permanencia de la pieza en un proceso dado cumple la restricción:  $\mathcal{P}_{min} \leq t_p \leq \mathcal{P}_{max}$ . El par de valores  $\mathcal{P}_{min}$  y  $\mathcal{P}_{max}$  define una ventana temporal.

Las restricciones de tipo ventanas temporales reciben, según quien las cite, distintas denominaciones, como: ventanas de tiempo de procesamiento (*processing time windows*) según Yan *et al.* (2010b) y Feng *et al.* (2014), tiempo de procesamiento flexible (*flexible processing time*) según Yan *et al.* (2008) y Yan *et al.* (2010a), ventanas de proceso (*processing windows*) según Crama *et al.* (2000) y tiempos de procesamiento tipo intervalo (*interval processing times*) según Levner y Kats (2009), Kats y Levner (2011a) y Kats y Levner (2011b).

Adicionalmente, ciertos autores denominan a los problemas de celdas robotizadas en los que se aplican estas restricciones *celdas robotizadas tipo intervalo (interval robotic cells)*, por ejemplo en Dawande *et al.* (2007) y Dawande *et al.* (2009).

**Ventanas temporales ilimitadas.** La denominación de ventanas temporales ilimitadas (*unbounded processing windows*) procede de Crama *et al.* (2000), mientras que en Dawande *et al.* (2005) se adopta la denominación de *free-pickup*.

Corresponde al caso tradicional de manufactura de piezas formado por procesos puramente mecánicos. A cada pieza en cada máquina le corresponde un tiempo de procesamiento  $\mathcal{P}$  dado y se permite que, una vez finalizado dicho proceso, las piezas puedan

permanecer en la máquina por un tiempo ilimitado. En este caso la máquina actúa como almacén transitorio de la pieza.

Las restricciones de ventanas temporales ilimitadas pueden modelizarse mediante ventanas temporales haciendo  $\mathcal{P}_{min} = \mathcal{P}$  y  $\mathcal{P}_{max} = \infty$ .

**Sin esperas (no-wait).** Corresponde al caso donde a cada pieza, en cada máquina, le corresponde un tiempo dado de procesamiento  $y$ , una vez finalizado el proceso, no se permite la permanencia de la pieza en la máquina. Está presente en ciertos procesos térmicos o químicos, por ejemplo ciertos baños corrosivos, donde la permanencia de la pieza en estos procesos durante un tiempo adicional produce, en ella, efectos no deseados.

Las restricciones de tipo *sin esperas (no wait)* reciben, según quien las cite, distintas denominaciones. La designación *sin esperas* es la más frecuente, y es empleada en Agnetis (2000), Agnetis y Pacciarelli (2000) y Dawande *et al.* (2005). Otros autores expresan estas restricciones bajo la denominación de *tiempos de procesamiento fijo* (fixed processing time) en Che *et al.* (2010) y Leung y Levner (2006). La denominación de *ventanas de proceso de ancho cero* (zero-width processing windows), es empleada en Crama *et al.* (2000) y Mangione *et al.* (2003) para designar este tipo de restricciones temporales.

Las restricciones de tiempo de tipo *sin esperas* pueden modelizarse mediante ventanas temporales haciendo  $\mathcal{P}_{min} = \mathcal{P}_{max} = \mathcal{P}$ .

**Sin esperas en transporte.** Esta restricción se aplica sobre los tiempos de espera entre procesos sucesivos  $t_w$  (presente en el medio de transporte de la pieza) e impone que este tiempo es cero.

Las restricciones sin espera en el transporte modelizan el caso especial donde una vez finalizado un proceso sobre una pieza, ésta deba ingresar en el proceso siguiente en un tiempo nulo (o suficientemente pequeño tal que pueda considerarse nulo) para evitar la pérdida de cierta propiedad inherente al proceso (térmica, de adherencia de pegamentos, etc). Este tipo de restricción es denominada no-wait por los autores Hall y Sriskandarajah (1996) y Pinedo (2012), y se confunde con la versión sin esperas explicada en punto anterior. Ambos enunciados son, desde el punto de vista del modelo, completamente equivalentes para el caso de tiempos de procesamiento fijos donde no hay buffers en el sistema; es decir el caso  $t_p = \mathcal{P}$  y buffer nulo equivale al caso  $t_p = \mathcal{P}$  y  $t_w = 0$ . Aún así, en lo conceptual la confusión persiste, y por ello, en Crama *et al.* (2000) se propone emplear *zero-width processing windows* para designar la restricción sin esperas anterior y utilizar la expresión *no-wait robot loaded* cuando la restricción sin esperas se aplica al recurso de transporte.

## 1.9 Manipulador doble

El empleo de un manipulador doble (*dual gripper* o *double-blade*, como se denomina en el contexto de los grupos de herramientas) permite reducir los tiempos de transporte, debido a que permite el intercambio de piezas en las máquinas mediante un solo desplazamiento del robot frente a los dos desplazamientos necesarios cuando se emplea

un solo manipulador. En efecto, en este último caso, el intercambio de dos piezas en una misma máquina requiere el empleo de dos movimientos. El primer movimiento debe extraer de la máquina la pieza saliente y depositarla en la estación siguiente (máquina o buffer), para producir el vaciado de la máquina. El segundo movimiento debe tomar la pieza entrante de la estación anterior y depositarla en la máquina. En el caso de disponer de un manipulador doble, si el robot llega a la máquina con la pieza entrante tomada por uno de los manipuladores y el otro manipulador está vacío, en este desplazamiento del robot, la pieza que está ocupando la máquina puede ser tomada por el manipulador vacío y la pieza tomada por el manipulador ocupado, puede, a continuación, cargarse en la máquina. Varios autores analizan problemas donde se explota el beneficio del doble manipulador: Sethi *et al.* (2001), Drobouchevitch *et al.* (2004), Sriskandarajah *et al.* (2004), Drobouchevitch *et al.* (2006), Galante y Passannanti (2006), Geismar *et al.* (2006), Geismar *et al.* (2008), Dawande *et al.* (2009), Geismar *et al.* (2011) y Foumani *et al.* (2013).

### 1.10 Múltiples robots

Con el fin de aumentar los recursos de transporte de las piezas suele considerarse el empleo de múltiples robots. Diversos problemas de celdas robotizadas con múltiples robots son analizados en Geismar (2003), Geismar *et al.* (2004b), Che y Chu (2005), Che y Yuan (2008), Geismar *et al.* (2008), Che y Chu (2009), Che *et al.* (2011a), Che *et al.* (2012) y Elmi y Topaloglu (2013).

### 1.11 Flexibilidad operacional

La flexibilidad operacional consiste en la posibilidad de realizar diferentes operaciones en una misma máquina. Los sistemas de manufactura flexible (*flexible manufacturing systems*, FMS) consisten en máquinas tipo CNC (*computer numerical control*) provistos de almacenes de herramientas (*tool magazine*), donde cada herramienta se destina a una operación diferente, y de un sistema automático de cambio de herramientas. Diversos problemas de celdas robotizadas con flexibilidad operacional son tratados en Gultekin *et al.* (2006a), Gultekin *et al.* (2006b), Gultekin *et al.* (2007), Gultekin *et al.* (2008), Gultekin *et al.* (2009), Gultekin *et al.* (2010), Batur *et al.* (2012), Fathian *et al.* (2013) y Foumani y Jenab (2013).

### 1.12 Buffers

El caso más estudiado, sin duda, es el de celdas robotizadas sin buffers, o *bufferless robotic cells*. Sin embargo, el empleo de buffers (aun de pequeña capacidad) para el almacenamiento temporal de las piezas en proceso, puede conducir a programas más eficientes por eliminación de bloqueos, que pueden tener lugar cuando las piezas tienen impedido avanzar a la máquina siguiente cuando ésta está procesando otra pieza. Problemas con buffers de almacenamiento de piezas en proceso son analizados en McCormick *et al.* (1989), Brauner *et al.* (1997), Błażewicz *et al.* (2002), Drobouchevitch *et al.* (2006), Drobouchevitch *et al.* (2010) y Geismar *et al.* (2011).

## 1.13 Máquinas en paralelo

Consiste en el caso general de celda robotizada formado por  $M$  puestos, donde cada puesto tiene un número  $m_i$  ( $1 \leq i \leq M$ ) de máquinas en paralelo, de modo que en todo instante el puesto  $i$  puede contener un máximo de  $m_i$  piezas procesándose.

Diversos problemas de celdas robotizadas con máquinas en paralelo son tratados en Sriskandarajah y Sethi (1989), Kamoun y Sriskandarajah (1993), Arbib y Marinelli (2003), Geismar (2003), Geismar *et al.* (2004b), Geismar *et al.* (2004a), Dawande *et al.* (2005), Geismar *et al.* (2006), Geismar *et al.* (2008), Werner y Kravchenko (2010), Kim y Lee (2012) y Elmi y Topaloglu (2013).

# Capítulo 2

## Problema y objetivos

### 2.1 Introducción

De la revisión de la literatura del capítulo anterior pueden extraerse conclusiones sobre los modelos observados:

**Sobre los tiempos.** Los modelos de tiempos de robot suelen ser innecesariamente simplificadores (apartado 1.6). En particular, no se observan modelos en los que se tenga en cuenta la influencia de la pieza (su peso y geometría) en los tiempos de carga y descarga, ni en los tiempos de transporte.

En los trabajos publicados se asumen excesivas simplificaciones en los tiempos del robot, generalmente adoptadas con el fin de reducir la complejidad del modelo. Sin embargo, dado que las magnitudes de los tiempos del robot pueden ser determinadas con una elevada precisión y, al mismo tiempo, las operaciones del robot asociadas a estos tiempos admiten un elevado número de repeticiones produciendo errores de tiempo muy pequeños (por lo general del orden de fracciones de segundo en robots industriales), dichas simplificaciones parecen carecer de suficiente justificación.

Por otro lado, los modelos observados en la literatura no incorporan la capacidad del robot de producir giros completos (robots de 360 grados). Cuando se dispone de esta característica en el robot, y ésta, al mismo tiempo, es admitida en el modelo, los programas resultantes suelen ser más eficientes como producto de la reducción de los tiempos de transporte.

**Sobre los buffers.** La literatura se ha concentrado en tratar problemas sin buffers (apartado 1.12). Sin embargo, el uso de buffers, aun cuando éstos fueran de capacidad pequeña, tiende a producir programas de mejor calidad que cuando no se los emplea, debido a la reducción de condiciones de bloqueo que normalmente se producen durante la ejecución del programa. Un modelo con buffers situados entre cada par de máquinas, de capacidad introducida como un dato del problema, permitiría abarcar la totalidad de los problemas reales donde hubiese buffers, incluido, en particular, el problema de celdas robotizadas sin buffers, en este caso, haciendo nulos los valores de

las capacidades de todos los buffers.

**Sobre el número de piezas.** Un modelo restringido a un número de piezas constante o a un número de tipos de piezas constante (o único), carece de generalidad y como consecuencia reduce su aplicabilidad en problemas reales.

**Sobre el número de máquinas.** Un modelo restringido a un número de máquinas constante carece de generalidad y como consecuencia reduce su aplicabilidad en problemas reales.

**Sobre la función objetivo.** Como se indicó en el apartado 1.3, los modelos basados en programas cíclicos no ofrecen una solución completa a los problemas de optimización de tiempos, pues nada dicen de la forma en que se llega y se sale de la fase cíclica. Por otra parte, aunque por simplicidad se suelen emplear ciclos de grado 1 (1-ciclo para piezas iguales o 1 MPS para piezas distintas), los ciclos de grado 1 no son necesariamente óptimos frente a ciclos de grado superior. Por otro lado, mientras que el enfoque cíclico ha sido intensamente tratado, el enfoque no cíclico, basado en optimizar la función  $C_{max}$ , no ha sido explotado lo suficiente.

**Sobre las restricciones en los tiempos de proceso.** Como se vio en el apartado 1.8 el modelo más completo procede de definir, además de tiempos de proceso de cada pieza en cada máquina, restricciones sobre el tiempo de permanencia de la pieza en proceso  $t_p$  y restricciones sobre el tiempo transcurrido entre procesos sucesivos  $t_w$ . Sin embargo, el caso más frecuente en entornos industriales es aquél en que las piezas reciben operaciones de tipo mecánico, donde no se suelen presentar restricciones temporales adicionales a la de los tiempos de procesamiento. Por esta razón, el caso de *ventanas temporales ilimitadas* definido en el apartado 1.8, aunque diste de ser el caso general, merece ser tratado por su frecuente aparición en entornos industriales reales.

**Sobre la configuración de la celda: robot y máquinas.** Si bien los modelos basados en múltiples robots, múltiples manipuladores y múltiples máquinas por puesto, pueden ser de interés presente y futuro, por lo observado en la literatura se concluye que la arquitectura de celda más elemental basada en robot único, manipulador único, y una máquina por puesto, no está todavía completamente resuelta; resta por explorar, al menos, las variantes que contemplan lo expresado en los puntos anteriores.

## 2.2 Definición del problema

Del apartado anterior se extraen las características de interés para definir el problema a ser resuelto en esta tesis. Éstas, con relación a la definición general establecida en el apartado 1.2, se resumen en:

- ▶ Arquitectura de robot único, manipulador único, una máquina por puesto (se prescinde por innecesaria de la noción de puesto).
- ▶ Número variable de máquinas:  $m$  máquinas ( $m$  puestos de una máquina cada uno).
- ▶ Número variable de piezas y de tipo de piezas:  $n$  piezas, en general distintas.

## Definición del problema

---

- ▶ Buffers entre máquinas de capacidad dato del problema.
- ▶ Función objetivo  $C_{max}$ .

Estas características permiten formular formalmente el problema como se expresa a continuación.

Esta tesis trata el procesamiento de un conjunto de  $n$  piezas en una celda robotizada de  $m$  máquinas. Ésta, cuyo esquema se muestra en la figura 2.1, se compone de:

- ▶ un conjunto de estaciones, formado por:
  - ▷ una estación  $I$  donde ingresan las piezas a ser procesadas;
  - ▷ una estación  $O$  donde son depositadas las piezas terminadas;
  - ▷ un conjunto de máquinas  $M_1, M_2, \dots, M_m$ , donde se realiza el procesamiento de las piezas;
  - ▷ un conjunto de buffers  $B_1, B_2, \dots, B_{m-1}$ , de capacidades respectivas  $b_1, b_2, \dots, b_{m-1}$ , cada uno situado entre un par de máquinas consecutivas;
- ▶ y un robot que realiza el transporte de las piezas entre las distintas estaciones.

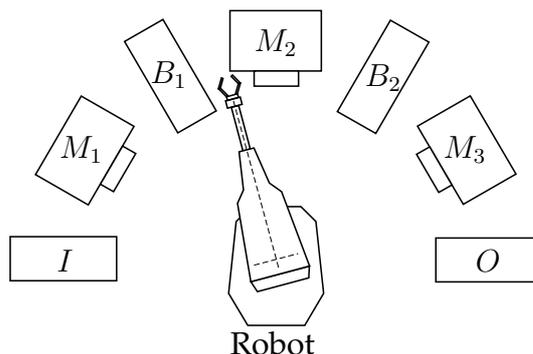


Figura 2.1: Celda robotizada con buffers.

El sistema opera bajo las siguientes condiciones:

- ▶ **Las máquinas y el robot.** Tanto las máquinas como el robot pueden contener una pieza en todo instante.
- ▶ **Las estaciones de almacenamiento del sistema.** Se conforman por cada uno de los buffers  $B_i$  ( $i = 1, \dots, m - 1$ ), la estación de entrada de piezas  $I$  y la estación de salida de piezas  $O$ .
  - ▷ Inicialmente las piezas están todas disponibles en la estación de entrada.
  - ▷ La estación de salida  $O$  dispone de capacidad suficiente para alojar todas las piezas.
  - ▷ La capacidad de los buffers es independiente del tipo de pieza alojada en el buffer.

- ▷ Se permite el valor nulo de la capacidad de los distintos buffers, es decir,  $b_i \geq 0$  ( $i = 1, \dots, m - 1$ );
- ▷ Todo buffer  $B_i$  tiene permitido recibir piezas que procedan solamente de su máquina anterior  $M_i$ , y, simultáneamente, las piezas salientes de dicho buffer pueden tener como destino sólo a la máquina siguiente  $M_{i+1}$ .

### ► Restricciones por los tiempos del proceso.

- ▷ Cada pieza tiene definido un tiempo de proceso propio en cada máquina. Se trata de un problema de piezas distintas.
- ▷ Las piezas pueden permanecer un tiempo ilimitado en las máquinas y en los buffers.

### ► El flujo del sistema.

- ▷ Las piezas son procesadas según el orden creciente de subíndices de las máquinas  $M_1, M_2, \dots, M_m$ . Tal condición implica las tres siguientes:
  - ▷ el flujo del sistema es de tipo flowshop,
  - ▷ se prohíbe el reingreso de las piezas a las máquinas o recirculación,
  - ▷ y no se permite el salto de máquinas.
- ▷ La secuencia de piezas observada por las máquinas es la misma para toda máquina. El flujo del sistema consiste en la variante flowshop denominada flowshop permutativo o flowshop tipo **P**.
- ▷ No se permiten interrupciones en los procesos sobre las piezas.
- ▷ Es permitida la transferencia directa de una máquina a la máquina siguiente cuando el buffer intermedio está vacío.

### ► Los tiempos del robot.

- ▷ Se definen tiempos de viaje para cada pieza entre cada par de estaciones. Tal condición implica que el modelo incorpora:
  - ▷ la influencia de la pieza (por su peso o geometría) en los tiempos del robot y
  - ▷ la capacidad de giro completo del robot<sup>1</sup>.
- ▷ Se definen tiempos de viaje del robot sin pieza entre cada par de estaciones.
- ▷ Se definen tiempos de carga para cada pieza en cada estación.
- ▷ Se definen tiempos de descarga para cada pieza de cada estación.
- ▷ Se desprecian las diferencias de tiempos derivadas de las distintas posiciones en los buffers.

Bajo estas condiciones se busca resolver la variante  $C_{max}$  del problema enunciado, es decir, encontrar una solución que minimice el tiempo de terminación del conjunto de piezas a procesar.

---

<sup>1</sup> Esto se consigue definiendo cada uno de los tiempos de viaje del problema (de cada pieza entre cada par de estaciones) como el menor de los tiempos que están asociados a los dos posibles desplazamientos entre todo par de estaciones, es decir, considerando los dos sentidos de giro. Esto no requiere introducir modificaciones en el modelo dado que se logra a partir de una definición adecuada de los datos del problema.

### 2.2.1 Notación de la estaciones

Las estaciones serán denotadas mediante dos conjuntos de símbolos. Se mantiene la notación tradicional, seguida por numerosos autores, en donde:

- ▶  $I$  denota la *estación de entrada*,
- ▶  $O$  denota la *estación de salida*,
- ▶ los símbolos  $M_1, M_2, \dots, M_m$  denotan las  $m$  máquinas,
- ▶ y los símbolos  $B_1, B_2, \dots, B_{m-1}$  denotan los  $m - 1$  buffers,

denominada notación por *tipo de estación*, y a ésta se añade una notación formada por la siguiente secuencia de símbolos:

$$E_1, E_2, E_3, \dots, E_{2m-1}, E_{2m}, E_{2m+1}$$

donde el subíndice de  $E$  indica la posición ocupada por la estación (estación de entrada, estación de salida, máquina o buffer) en la celda, y por esto es denominada notación por *posición*.

Una representación de las estaciones por *tipo de estación* resulta más útil para definir la notación de los tiempos de proceso, mientras que la notación por *posición* es más adecuada en la notación de los tiempos del robot. La correspondencia entre ambas es biunívoca y se muestra en la expresión siguiente:

$$\begin{array}{cccccccccccc}
 E_1 & E_2 & E_3 & \dots & E_{2i} & E_{2i+1} & \dots & E_{2m-2} & E_{2m-1} & E_{2m} & E_{2m+1} \\
 \updownarrow & \updownarrow & \updownarrow & \dots & \updownarrow & \updownarrow & \dots & \updownarrow & \updownarrow & \updownarrow & \updownarrow \\
 I & M_1 & B_1 & \dots & M_i & B_i & \dots & M_{m-1} & B_{m-1} & M_m & O
 \end{array}$$

La relación entre los subíndices de ambos conjuntos de símbolos puede expresarse mediante relaciones de las funciones  $\text{entmed}()$  y  $\text{resmed}()$ . Sea  $q$  un parámetro de tipo entero, se define  $\text{entmed}(q)$  como el cociente entero  $q/2$  y, se define  $\text{resmed}(q)$ , como el resto del cociente entero  $q/2$ . Cuando  $q$  toma exclusivamente valores enteros positivos,  $\text{resmed}(q)$  sólo puede valer 0 ó 1.

La relación entre los subíndices de ambos conjuntos de símbolos puede expresarse mediante las siguientes relaciones:

- ▶ La estación  $E_\lambda$  es  $I$  si  $\lambda = 1$ .
- ▶ La estación  $E_\lambda$  es  $O$  si  $\lambda = 2m + 1$ .
- ▶ La estación  $E_\lambda$  es la máquina  $M_i$  si:
  - ▷  $1 < \lambda < 2m + 1$ ,
  - ▷  $\text{resmed}(\lambda) = 0$ , puesto que las máquinas son estaciones pares, y

- ▷  $i = \text{entmed}(\lambda)$ .
- ▶ La estación  $E_\lambda$  es el buffer  $B_i$  si:
  - ▷  $1 < \lambda < 2m + 1$ ,
  - ▷  $\text{resmed}(\lambda) = 1$ , dado que los buffers son estaciones impares, y
  - ▷  $i = \text{entmed}(\lambda)$ .

La correspondencia en sentido inverso puede expresarse mediante las siguientes relaciones:

- ▶  $I$  es la estación  $E_1$ .
- ▶ La máquina  $M_i$  es la estación  $E_{2i}$  ( $i = 1, 2, \dots, m$ ).
- ▶ El buffer  $B_i$  es la estación  $E_{2i+1}$  ( $i = 1, 2, \dots, m - 1$ ).
- ▶  $O$  es la estación  $E_{2m+1}$ .

### 2.2.2 Terminología de las operaciones

Existen cuatro tipos de operaciones. Las operaciones de proceso, que son ejecutadas por las máquinas, y las operaciones de carga, descarga y transporte, que son realizadas por el robot. A fin de describir los procesos realizados en el sistema se requiere establecer una terminología unificada relativa a las estaciones para indicar las operaciones de carga y descarga en todas ellas.

- ▶ Las referencias a la operación de **cargar** o a la **carga** de una estación tendrá por significado la operación por la cual una pieza es ingresada en una estación: máquina, buffer o estación de salida.
- ▶ Las referencias a la operación de **descargar**, o a la **descarga** de una estación tendrá por significado la acción de **extraer** una pieza de una estación: buffer, máquina o estación de entrada.

### 2.2.3 Notación de los tiempos del robot

Ligados con las operaciones definidas en el apartado anterior, se definen los siguientes tiempos:

#### 2.2.3.1 Notación de los tiempos de carga

Se denota  $t_{i,j}^c$  el tiempo empleado en cargar la pieza  $P_j$  en la estación  $E_i$  ( $i = 2, \dots, 2m + 1$  y  $j = 1, \dots, n$ ).

#### 2.2.3.2 Notación de los tiempos de descarga

Se denota  $t_{i,j}^d$  el tiempo empleado en descargar la pieza  $P_j$  de la estación  $E_i$  ( $i = 1, \dots, 2m$  y  $j = 1, \dots, n$ ).

### 2.2.3.3 Notación de los tiempos de viaje

Se denota  $t_{h,i,j}^v$  el tiempo empleado por el robot en viajar transportando la pieza  $P_j$  desde la estación  $E_h$  hasta la estación  $E_i$  ( $h = 1, \dots, 2m + 1$ ;  $i = 1, \dots, 2m + 1$  y  $j = 1, \dots, n$ ). El tiempo empleado en el movimiento del robot sin pieza se denota  $t_{h,i,0}^v$ . Se asumen las siguientes igualdades:

$$t_{h,i,j}^v = t_{i,h,j}^v, \quad t_{h,i,0}^v = t_{i,h,0}^v \quad \text{y} \quad t_{i,i,j}^v = t_{i,i,0}^v = 0$$

**Propiedad monótona creciente.** Se asume la relación monótona creciente entre las distancias recorridas y los tiempos empleados en recorrer tales distancias. Sean  $d_1$  y  $d_2$  dos distancias, y  $t_1$  y  $t_2$ , respectivamente, los tiempos empleados en recorrer tales distancias. Se asume:

$$\begin{aligned} \text{si } d_1 = d_2 \text{ entonces } t_1 &= t_2 \\ \text{si } d_1 > d_2 \text{ entonces } t_1 &> t_2 \\ \text{si } d_1 < d_2 \text{ entonces } t_1 &< t_2 \end{aligned} \tag{2.1}$$

**Propiedad triangular.** Se asume el cumplimiento de la propiedad triangular sobre los tiempos de transporte de toda pieza  $P_j$  entre todo par de estaciones, es decir:

$$t_{f,h,j}^v \leq t_{f,g,j}^v + t_{g,h,j}^v \tag{2.2}$$

La propiedad triangular expresa que, si el desplazamiento total entre dos puntos se hace mediante un sólo desplazamiento, el tiempo empleado es menor o igual al tiempo empleado en el caso en que el desplazamiento total se divide en uno o más desplazamientos parciales. En particular, el sentido menor de la expresión 2.2 puede deberse a dos factores, los cuales pueden, posiblemente, actuar simultáneamente:

- ▶ porque el desplazamiento único emplea una trayectoria de menor tiempo que la suma de los tiempos asociados a las trayectorias en el caso de desplazamientos parciales múltiples,
- ▶ porque los desplazamientos parciales múltiples imponen detenciones que pueden introducir demoras por el frenado del robot hasta la detención completa, y, seguidamente una nueva puesta en marcha.

### 2.2.3.4 Notación de los tiempos de proceso

Se denota  $t_{i,j}^p$  el tiempo de proceso de la pieza  $P_j$  en la máquina  $M_i$  ( $i = 1, \dots, m$ ;  $j = 1, \dots, n$ ).

### 2.3 Modelo de espacio de estados del problema

En muchos casos resulta útil formular los problemas de tipo combinatorio (como el que es objeto de esta tesis) como un problema de búsqueda en un espacio de estados.

Por este enfoque, se puede definir:

- ▶ *Objeto*: son los elementos que constituyen el sistema, por ejemplo: celda robotizada, piezas, máquinas, robot, etc.
- ▶ *Estado*: es una configuración dada de los objetos relevantes del problema, por ejemplo, la posiciones ocupadas por las piezas y los estados de las máquinas (ocupada o vacía).
- ▶ *Operadores*: son las acciones que se pueden aplicar a los estados y que tienen por objetivo modificar estos, por ejemplo, en ciertos problemas de secuenciación de piezas, el cambio de la posición de las piezas en el sistema modifica el estado.
- ▶ *Estados iniciales*: son los estados asociados a la configuración de los objetos al inicio del problema, por ejemplo, en ciertos problemas de secuenciación de piezas, una configuración donde las piezas ocupan la estación de entrada.
- ▶ *Estados objetivo*: son los estados asociados a la configuración de los objetos para el cual el problema se considera resuelto, por ejemplo, en problemas de secuenciación de piezas, una configuración donde las piezas ocupan la estación de salida.
- ▶ *Espacio de estados*: es el conjunto de todos los posibles estados asociado al problema.
- ▶ *Solución*: en muchos problemas, la solución puede expresarse como la secuencia de operadores que, cuando son aplicados a un determinado estado inicial, producen un determinado estado objetivo.
- ▶ *Generación*: es la aplicación de los operadores a los estados para producir nuevos estados.
- ▶ *Árbol de enumeración de estados*: a fin de favorecer el proceso de búsqueda en el espacio de estados, éste admite una representación arborescente donde los estados corresponden a los vértices del árbol y los operadores corresponden a los arcos.

#### 2.3.1 Estado

Para el problema establecido en el apartado 2.2, se define **estado** y se denota  $e$  a la 5-tupla siguiente:

$$(r, \bar{l}, \rho, \bar{a}, \bar{\pi})$$

donde se agrupan parámetros escalares y vectoriales (éstos son denotados con una barra superior) definidos a continuación:

## Modelo de espacio de estados del problema

---

- $r$ : Es una variable escalar de valor entero positivo en  $1, 2, \dots, 2m + 1$  que registra el número de estación donde se encuentra el robot. Ejemplo:  $r$  tiene valor 2 si el robot está en la estación  $E_2$ .
- $\bar{l}$ : Es un vector  $n$ -dimensional de valor entero de la forma  $l_1, l_2, \dots, l_n$ , en el cual cada componente  $l_j$  registra el número de estación donde se encuentra alojada la pieza  $P_j$ . Ejemplo:  $l_1$  tiene valor 2 si la pieza  $P_1$  está en la estación  $E_2$ .
- $\rho$ : Es una variable de valor real positivo que registra el instante de terminación (disponibilidad) del robot. Es el instante de terminación del movimiento que lleva al estado considerado.
- $\bar{a}$ : Es un vector  $m$ -dimensional de valor real de la forma  $a_1, a_2, \dots, a_m$ , en el cual cada componente  $a_i$  registra el instante de terminación de la máquina  $M_i$ . En el caso en que  $M_i$  está vacía,  $a_i$  toma un valor arbitrario negativo.
- $\bar{\pi}$ : Es un vector  $n$ -dimensional de valor entero positivo de la forma  $\pi_1, \pi_2, \dots, \pi_n$ , en el cual cada componente  $\pi_k$  registra la pieza secuenciada en la posición  $k$  de la secuencia de piezas. En el caso en que la pieza a secuenciarse en la  $k$ -ésima posición es desconocida,  $\pi_k$  toma valor cero.

**Estado inicial.** Para el problema tratado en esta tesis, existe un estado inicial único, caracterizado por:

- ▶ todas las piezas están ocupando la estación de entrada:

$$l_j = 1; \quad j = 1 \dots n \quad (2.3)$$

- ▶ el robot está ocupando la estación de entrada:

$$r = 1 \quad (2.4)$$

- ▶ el robot está disponible en el instante inicial:

$$\rho = 0 \quad (2.5)$$

- ▶ todas las máquinas están sin pieza:

$$a_i < 0; \quad i = 1 \dots m \quad (2.6)$$

- ▶ la secuencia de piezas  $\bar{\pi}$  no está determinada:

$$\pi_k = 0; \quad k = 1 \dots n \quad (2.7)$$

**Estados objetivo.** Para el problema de esta tesis, los estados objetivo están caracterizados por:

- ▶ todas las piezas están ocupando la estación de salida:

$$l_j = 2m + 1; \quad j = 1 \dots n \quad (2.8)$$

- ▶ el robot está situado en la estación de salida:

$$r = 2m + 1 \quad (2.9)$$

- ▶ el instante de disponibilidad del robot  $\rho$  es el valor de la solución o función objetivo del problema;
- ▶ todas las máquinas están sin piezas:

$$a_i < 0; \quad i = 1 \dots m \quad (2.10)$$

- ▶ la secuencia de piezas está determinada por los índices de  $\pi_k$ , es decir que la secuencia de piezas ejecutada por el sistema es:

$$P_{\pi_1}, P_{\pi_2}, \dots, P_{\pi_{n-1}}, P_{\pi_n}$$

### 2.3.2 Operador del espacio de estados: movimiento del robot

El operador de espacio de estados definido para el problema de esta tesis se denomina **movimiento del robot**.

Un **movimiento del robot**  $\mu$  se representa mediante un par  $(\epsilon, \delta)$  y está formado por la secuencia de operaciones del robot siguiente.

- ▶ Viajar sin pieza desde la estación (máquina o buffer) donde se encuentra el robot (donde ha quedado después del último movimiento realizado), hasta la estación (máquina, buffer o estación de entrada) donde la pieza  $P_\epsilon$  se encuentra alojada.
- ▶ Esperar si es necesario a que la pieza termine de procesarse y descargar la pieza  $P_\epsilon$  de dicha estación.
- ▶ Transportar la pieza  $P_\epsilon$ , avanzando  $\delta$  estaciones, hasta la estación de destino.
- ▶ En la estación destino, cargar la pieza  $P_\epsilon$ .

**La solución: una secuencia de movimientos del robot.** La solución se expresa como la secuencia de operadores (movimientos del robot) que aplicada a un estado inicial (el cual cumple las condiciones expresadas por las relaciones 2.3 a 2.7) dan por resultado un estado objetivo (el cual cumple las condiciones expresadas por las relaciones 2.8 a 2.10).

Una secuencia  $\bar{\mu}$  formada por  $N_\mu$  movimientos del robot se representa por:

$$\bar{\mu} = \mu_1, \mu_2, \mu_3, \dots, \mu_{N_\mu} \quad (2.11)$$

equivalente a

$$\bar{\mu} = (\epsilon_1, \delta_1), (\epsilon_2, \delta_2), (\epsilon_3, \delta_3), \dots, (\epsilon_{N_\mu}, \delta_{N_\mu})$$

Donde  $N_\mu$  está acotado por:

$$n(m+1) \leq N_\mu \leq 2nm \quad (2.12)$$

En la expresión 2.12 la igualdad con la cota inferior  $N_\mu = n(m+1)$  se produce cuando la secuencia  $\bar{\mu}$  está conformada por movimientos en donde no se emplean los buffers intermedios  $B_1 \dots B_{m-1}$  por ninguna de las piezas  $P_1, \dots, P_n$ . Por otro lado, la igualdad con la cota superior  $N_\mu = 2nm$  se verifica cuando todos los buffers intermedios  $B_1 \dots B_{m-1}$  se emplean por todas las piezas  $P_1, \dots, P_n$ .

El valor de  $\delta$  sólo puede ser 1 o 2, dado que no es permitido el salto de máquinas y que cada buffer se utiliza exclusivamente como almacén de sus máquinas adyacentes, como se indicó en el apartado 2.2. El valor 2 corresponde a los movimientos hechos de una máquina a la siguiente (saltando el buffer entre ellas, lo cual sí se permite), mientras que el valor 1 corresponde a los movimientos hechos de una máquina a un buffer, o de un buffer a una máquina. Con esta notación, la ejecución de una secuencia de movimientos del robot determinada, tendrá la siguiente apariencia:

$$(2, 1), (2, 1), (1, 1), (2, 1), (2, 1), (1, 2), (1, 1)$$

Por simplicidad, la notación basada en los pares  $(\epsilon, \delta)$  será reemplazada por una representación más concisa, inspirada en la codificación tradicional de las secuencias de piezas. En efecto, puesto que, como se indicó en el párrafo anterior,  $\delta$  sólo puede valer 1 o 2, son posibles sólo los pares  $(\epsilon_w, 1)$  y  $(\epsilon_w, 2)$ . En la nueva representación, los pares  $(\epsilon_w, 1)$  (movimientos de una estación) se expresarán como  $\epsilon_w$  (el índice de la pieza), y los pares  $(\epsilon_w, 2)$  (movimientos de dos estaciones) se expresarán por  $\epsilon_w +$  (el índice de la pieza seguido de un símbolo +). De la aplicación de esta regla, la secuencia anterior se reescribe del modo siguiente:

$$2, 2, 1, 2, 2, 1+, 1$$

## 2.4 Análisis de movimiento-estado

En este apartado se analiza la transformación de un estado dado, denotado  $e_w$ , en un estado sucesor, denotado  $e_{w+1}$ , producida por la aplicación de un movimiento  $\mu_w$ . Esta transformación es denotada:

$$e_w \xrightarrow{\mu_w} e_{w+1}$$

Para un estado  $e_w$ , en la notación del estado el índice  $w$  aparecerá como subíndice, mientras que en los componentes de  $e_w$ ,  $w$  aparecerá como superíndice, de la forma:

$$e_w = (r^w, \bar{l}^w, \rho^w, \bar{a}^w, \bar{\pi}^w)$$

Para simplificar la notación, en este apartado se asumen las equivalencias para los parámetros de un movimiento  $\mu_w$ :

- ▶ el índice de la pieza transportada en el movimiento:  $j = \epsilon_w$
- ▶ y el número de estaciones avanzadas:  $\delta = \delta_w$

Con estas equivalencias, cuando, por aplicación de un movimiento  $\mu_w$  a un estado dado  $e_w$ , da origen a un estado sucesor  $e_{w+1}$ , se analizará:

- ▶ las condiciones que dan origen a un estado sucesor infactible (apartado 2.4.3),
- ▶ y los valores de las variables del estado sucesor  $e_{w+1}$  alcanzado al final del movimiento  $\mu_w$  (apartado 2.4.5).

A continuación (apartado 2.4.1) se presenta un modo de representación gráfica que será empleada recurrentemente a lo largo de todo este trabajo y sobre el cual se apoyarán los distintos análisis.

### 2.4.1 Diagrama de tiempos

Distintos autores emplean esquemas y recursos gráficos para representar los tiempos de distintos recursos en problemas de secuenciación de actividades. Estos responden, mayoritariamente, a formatos inspirados en el diagrama de Gantt tradicional. Otros emplean variantes que permiten adaptarlo al tipo de problema que se proponen analizar.

En este apartado se introduce el diagrama de tiempos (figura 2.2) que se empleará a lo largo de este trabajo y en el que se apoyarán las distintas explicaciones. Está inspirado en distintas propuestas, como las que proceden de Agnetis (2000) o de Kise *et al.* (1991).

Éste consiste en un diagrama de Gantt, donde en el eje horizontal sigue representándose el tiempo (como en el diagrama de Gantt tradicional), y en el eje vertical, donde tradicionalmente van los recursos (máquinas, buffers y robot), estos se representan siguiendo un símil con la distribución que éstos presentan en el espacio físico.

Cada recurso es representado mediante un símbolo gráfico diferente:

- ▶ las actividades desarrolladas en las máquinas se grafican en posiciones fijas en el eje vertical mediante rectángulos,
- ▶ las operaciones ejecutadas por el robot se representan en línea de trazo continuo, imitando la trayectoria descrita por el robot dentro de la celda, coincidiendo en ciertos instantes con la representación de alguna de las estaciones, cuando el robot ejecuta una operación en dicha estación.

La figura 2.2 muestra el diagrama de tiempos de una celda de dos máquinas con buffer intermedio, donde puede verse la configuración de las piezas, las máquinas y el robot en cada estado, al final de cada movimiento, de la secuencia de movimientos del robot presentada en el apartado 2.3.2.

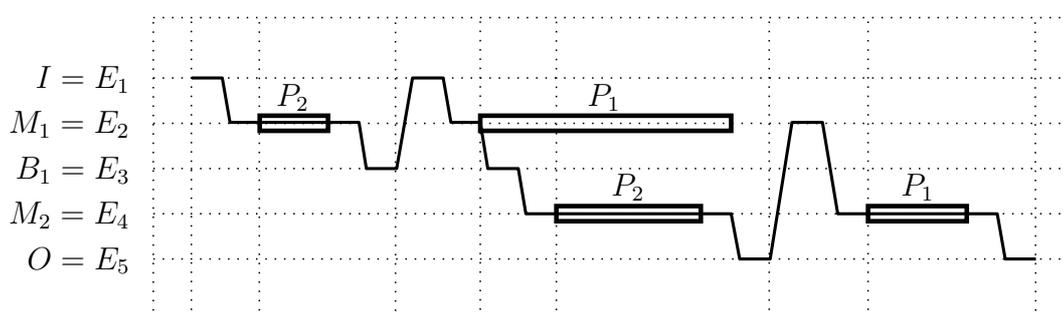


Figura 2.2: Diagrama de tiempos que representa el procesamiento de dos piezas en una celda robotizada de dos máquinas con un buffer intermedio cuando se realiza la secuencia de movimientos 2, 2, 1, 2, 2, 1+, 1.

### 2.4.2 Análisis de movimientos

Se repite el diagrama de tiempos de la secuencia de movimientos 2, 2, 1, 2, 2, 1+, 1 mostrada antes en la figura 2.2, ahora en la figura 2.3, donde son agregadas representaciones gráficas de los estados alcanzados al inicio y al final de cada movimiento. Por simplicidad, se muestran sólo las configuraciones de los objetos que resultan relevantes a un análisis cualitativo que se hará a continuación, como son las configuraciones de piezas, las estaciones y el robot.

La figura 2.3 se analiza a continuación.

- ▶  $e_1 \xrightarrow{\mu_1=2} e_2$ . Se parte del estado inicial  $e_1$  donde todas las piezas ( $P_1$  y  $P_2$ ) y el robot están en  $I$ . En el movimiento  $\mu_1 = 2$  se mueve la pieza  $P_2$  a  $M_1$  avanzando una estación y alcanzándose el estado  $e_2$  al final del movimiento.
- ▶  $e_2 \xrightarrow{\mu_2=2} e_3$ . El movimiento  $\mu_2 = 2$  se inicia en el estado  $e_2$ , donde  $P_1$  permanece en  $I$ , y tanto  $P_2$  como el robot se encuentran en  $M_1$ . En este caso  $P_2$  no está disponible, por lo que el robot debe esperar en  $M_1$  hasta que  $P_2$  termine de procesarse en dicha máquina. A continuación,  $P_2$  se avanza una estación hasta el buffer  $B_1$ , alcanzando el estado  $e_3$ .

- ▶  $e_3 \xrightarrow{\mu_3=1} e_4$ . El movimiento  $\mu_3 = 1$  se inicia en el estado  $e_3$ , donde  $P_1$  continúa en  $I$  y tanto  $P_2$  como el robot se encuentran en  $B_1$ . El robot parte de  $B_1$  en un desplazamiento sin pieza hasta  $I$ . Allí  $P_1$  está disponible (las piezas están siempre disponibles en  $I$  y  $B_1$ ) por lo que no se produce espera del robot. A continuación el robot transporta  $P_1$  desde  $I$  hasta  $M_1$  avanzando una estación, alcanzando el estado  $e_4$  al final del movimiento.
- ▶  $e_4 \xrightarrow{\mu_4=2} e_5$ . El movimiento  $\mu_4 = 2$  se inicia en el estado  $e_4$  donde  $P_2$  está ocupando el buffer  $B_1$  y tanto  $P_1$  como el robot se encuentran en  $M_1$ . El robot parte de  $M_1$  en un desplazamiento sin pieza hasta  $B_1$ . Allí  $P_2$  está disponible por lo que no se produce espera del robot. A continuación el robot toma la pieza  $P_2$  de  $B_1$ , donde se encuentra alojada desde el final del movimiento  $\mu_2$ , y la transporta a  $M_2$  avanzando una estación, donde la carga para su procesamiento. Al final del movimiento se alcanza el estado  $e_5$ .
- ▶  $e_5 \xrightarrow{\mu_5=2} e_6$ . El movimiento  $\mu_5 = 2$  se inicia en el estado  $e_5$  donde  $P_1$  está ocupando la máquina  $M_1$  y tanto  $P_2$  como el robot se encuentran en  $M_2$ . El robot espera en  $M_2$  hasta que  $P_2$  termine de procesarse allí. A continuación el robot toma la pieza  $P_2$  de  $M_2$  y la transporta a la estación de salida  $O$ , avanzando una estación y alcanzando el estado  $e_6$  al final del movimiento.
- ▶  $e_6 \xrightarrow{\mu_6=1+} e_7$ . El movimiento  $\mu_6 = 1+$  se inicia en el estado  $e_6$  donde  $P_1$  está ocupando la máquina  $M_1$  y tanto  $P_2$  como el robot se encuentran en la estación de salida  $O$ . El robot parte de  $O$  en un desplazamiento sin pieza hasta  $M_1$ . Allí  $P_1$  está disponible porque cuando el robot llega a la máquina  $M_1$ , la pieza  $P_1$  ya ha finalizado su procesamiento. A continuación el robot toma la pieza  $P_1$  de  $M_1$  y la transporta a la máquina siguiente  $M_2$  en un movimiento de dos estaciones (saltando el buffer intermedio  $B_1$ ) y alcanzando el estado  $e_7$  al final del movimiento.
- ▶  $e_7 \xrightarrow{\mu_7=1} e_8$ . El movimiento  $\mu_7 = 1$  se inicia en el estado  $e_7$  donde  $P_2$  está ocupando la estación de salida  $O$  y tanto  $P_1$  como el robot se encuentran en  $M_2$ . El robot espera en  $M_2$  hasta que  $P_1$  termina de procesarse allí. A continuación el robot toma la pieza  $P_1$  de  $M_2$  y la transporta a la estación de salida  $O$ , avanzando una estación, alcanzando el estado  $e_8$  al final del movimiento. El estado  $e_8$  es un estado final caracterizado por una configuración donde todas las piezas ( $P_1$  y  $P_2$ ) y el robot están situados la estación de salida  $O$ .

### 2.4.3 Infactibilidad de un movimiento

En el capítulo 7 se proponen procedimientos de tipo de exploración arborescente donde se aplica poda por infactibilidad. Por ello, es requisito analizar los distintos casos en que un movimiento puede ser infactible.

Previamente a introducir los casos de infactibilidad, se definen 4 variables útiles para el análisis posterior y las expresiones que permiten determinarlas:

- ▶ La variable  $F_{i,j}^w$  es una variable binaria cuyo valor es 1 cuando en el estado  $e_w$  la pieza  $P_j$  está ocupando el buffer  $B_i$  y su valor es 0 el caso contrario, y se determina como:

$$F_{i,j}^w = \begin{cases} 1 & \text{si } l_j^w = 2i + 1 \\ 0 & \text{en otro caso.} \end{cases} \quad (2.13)$$

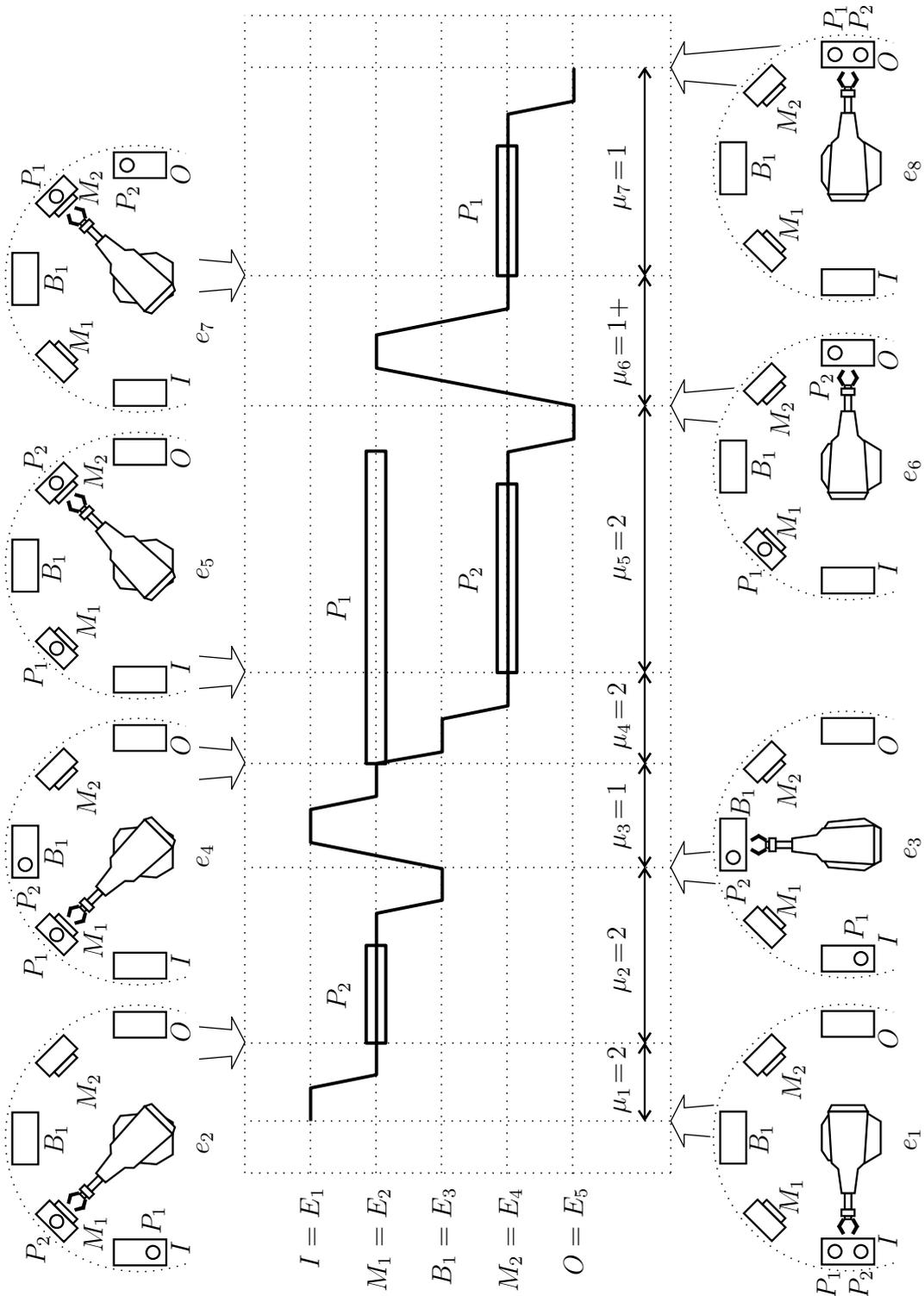


Figura 2.3: Estados en una celda robotizada de dos máquinas con un buffer intermedio cuando se realiza la secuencia de movimientos 2, 2, 1, 2, 2, 1+, 1.

- ▶ La variable  $f_i^w$  es una variable entera cuyo valor es el número de piezas que en el estado  $e_w$  están ocupando el buffer  $B_i$  y se determina como:

$$f_i^w = \sum_{j=1}^n F_{i,j}^w \quad (2.14)$$

- ▶ La variable  $Z_{i,j}^w$  es una variable binaria cuyo valor es 1 cuando en el estado  $e_w$  la pieza  $P_j$  ya ha ingresado a la máquina  $M_i$  (es decir que está ocupando  $M_i$  o una estación posterior a  $M_i$ ) y su valor es 0 en el caso contrario. Se determina como:

$$Z_{i,j}^w = \begin{cases} 1 & \text{si } l_j^w \geq 2i \\ 0 & \text{en otro caso.} \end{cases} \quad (2.15)$$

- ▶ La variable  $z_i^w$  es una variable entera cuyo valor es el número de piezas que en el estado  $e_w$  ya han ingresado a la máquina  $M_i$  (es decir, que están ocupando  $M_i$  o una estación posterior a  $M_i$ ). Se determina como:

$$z_i^w = \sum_{j=1}^n Z_{i,j}^w \quad (2.16)$$

Cuando un movimiento es aplicado a un determinado estado, el estado alcanzado puede incumplir una o más de las restricciones definidas en el apartado 2.2, en tal caso, el movimiento se dice infactible. La infactibilidad de un movimiento se determina por la aplicación de las siguientes pruebas de infactibilidad.

**La pieza  $P_j$  intenta ser movida a una estación inexistente.** Si se cumple que  $l_j^w + \delta > 2m + 1$  entonces el movimiento  $(j, \delta)$  es infactible.

**La pieza  $P_j$  intenta ser movida a una estación que no dispone de capacidad para contenerla.** Sean:

- ▶  $\lambda = l_j^w + \delta$  el índice de la estación  $E_\lambda$  a donde la pieza será movida,
- ▶  $i = \text{entmed}(\lambda)$  el índice del buffer o máquina ( $B_i$  o  $M_i$  en notación por tipo de estación) a donde la pieza será transportada.
- ▶  $f_i^w$  el número de piezas que ocupan el buffer  $B_i$  (determinada mediante las expresiones 2.13 y 2.14).

Se presentan dos casos:

- ▶ Si la pieza  $P_j$  será transportada a una máquina, comprobar si está ocupada por otra pieza.

Si:

- ▷  $\text{resmed}(\lambda) = 0$  (es una máquina) y
- ▷  $a_i^w \geq 0$  (está con pieza)

entonces el movimiento  $(j, \delta)$  es infactible.

- ▶ Si la pieza  $P_j$  será transportada a un buffer, comprobar si la capacidad de dicho buffer ha sido cubierta.

Si:

- ▷  $\text{resmed}(\lambda) = 1$  (es un buffer)
- ▷  $\lambda < 2m + 1$  (no es la última estación) y
- ▷  $f_i^w = b_i$  (no hay capacidad disponible)

entonces el movimiento  $(j, \delta)$  es infactible.

**La pieza  $P_j$  es movida a una máquina y no cumple con la secuencia de piezas.**

Sean:

- ▶  $\lambda = l_j^w + \delta$  el índice de la estación  $E_\lambda$  a donde la pieza será movida,
- ▶  $i = \text{entmed}(\lambda)$  el índice del buffer o máquina ( $B_i$  o  $M_i$  en notación por tipo de estación) a donde la pieza será transportada,
- ▶  $z_i^w$  el número de piezas que ya han ingresado en la máquina  $M_i$  (hallado mediante las ecuaciones 2.15 y 2.16),
- ▶ y  $\eta = z_i^w + 1$  la posición en la secuencia  $\bar{\pi}$  de la próxima pieza a ingresar en la máquina  $M_i$ .

Si:

- ▶  $\text{resmed}(\lambda) = 0$  (es una máquina),
- ▶  $i > 1$  (no es la primera máquina) y
- ▶  $j \neq \pi_\eta^w$  (la pieza no es la próxima que debe ser ingresada)

entonces el movimiento  $(j, \delta)$  es infactible.

**La pieza  $P_j$  es movida de un buffer a otro saltando la máquina intermedia.** Sea  $\lambda = l_j^w$  la estación desde donde parte la pieza. Si:

- ▶  $\text{resmed}(\lambda) = 1$  (es un buffer) y
- ▶  $\delta = 2$  (es un movimiento de 2 estaciones)

entonces el movimiento  $(j, \delta)$  es infactible.

### 2.4.4 Condición de infactibilidad de un movimiento cuando la secuencia de piezas está predefinida

En el capítulo 7 se presentan dos enfoques de resolución que emplean una secuencia de piezas predefinida, que debe ser respetada por todas las soluciones factibles. Bajo esta premisa, a las condiciones enunciadas en el apartado 2.4.3 debe agregarse una condición de infactibilidad que aparece cuando el problema exige la ejecución de una *secuencia de piezas predefinida*.

Una secuencia predefinida  $\bar{\Pi}$  es un vector  $n$ -dimensional de valores enteros positivos de la forma:

$$\bar{\Pi} = \Pi_1, \Pi_2, \dots, \Pi_n \quad (2.17)$$

donde cada componente  $\Pi_k$  es el índice de la pieza que debe ser secuenciada en la posición  $k$ . Es decir, cuando se impone que la pieza  $P_j$  debe ser secuenciada en la posición  $k$  de la secuencia predefinida  $\bar{\Pi}$ , entonces  $\Pi_k = j$ .

Cuando la secuencia de piezas está predefinida, debe cumplirse para todo estado objetivo que:

$$\bar{\pi} = \bar{\Pi} \quad (2.18)$$

Para que la igualdad 2.18 sea cumplida en todo estado objetivo, debe ser descartado por infactible todo estado en el que se cumple la condición siguiente. Sea  $z_1^w$  el número de piezas que ya han ingresado en la primera máquina  $M_1$  y  $\eta = z_1^w + 1$  la posición en la secuencia de piezas predefinida  $\bar{\Pi}$  de la próxima pieza a ingresar en dicha máquina; entonces si:

- ▶  $l_j^w = 1$  (la pieza  $P_j$  ocupa la estación de entrada),
- ▶  $\delta = 1$  (se realiza un movimiento de una estación) y
- ▶  $j \neq \Pi_\eta$  (la pieza  $P_j$  a ingresar en  $M_1$  que ocupará la posición  $\eta$  en  $\pi^w$ , no es coincidente con la pieza que ocupa la misma posición en la secuencia de piezas  $\bar{\Pi}$ ).

entonces el movimiento  $(j, \delta)$  es infactible.

### 2.4.5 El tiempo al final de un movimiento

Son posibles tres tipos de movimientos:

- ▶ de una máquina al buffer siguiente (o la estación de salida),
- ▶ de un buffer (o la estación de entrada) a la máquina siguiente y

- de una máquina a la siguiente.

En la figura 2.4 se muestran los tiempos de un movimiento del robot en el que una pieza  $P_j$  se mueve de una máquina  $M_i$  a la siguiente  $M_{i+1}$ , mediante el movimiento  $(j, 2)$ .

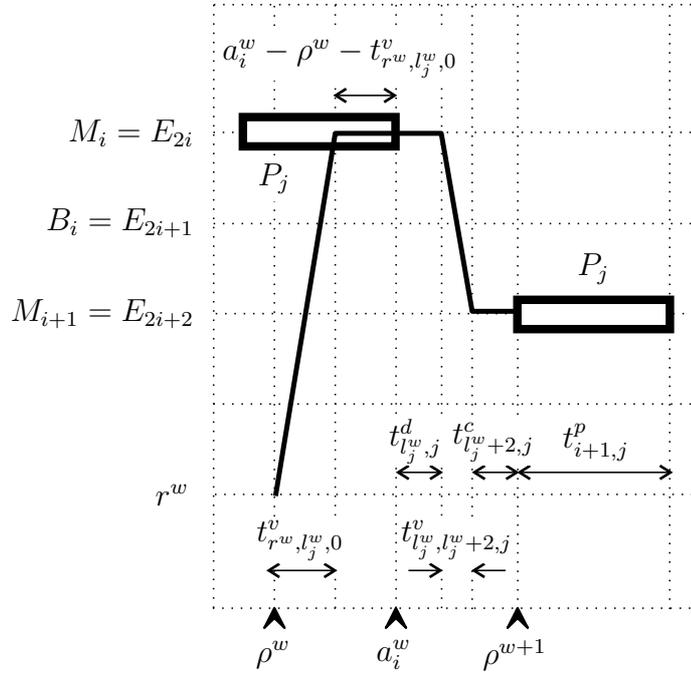


Figura 2.4: Tiempos de un movimiento del robot en el que una pieza  $P_j$  se mueve de una máquina  $M_i$  a la siguiente  $M_{i+1}$ , mediante el movimiento  $(j, 2)$ .

En la figura 2.5 se muestran los tiempos de un movimiento del robot en el que una pieza  $P_j$  se mueve de un buffer  $B_{i-1}$  a la máquina siguiente  $M_i$ , mediante el movimiento  $(j, 1)$ .

En la figura 2.6 se muestran los tiempos de un movimiento del robot en el que una pieza  $P_j$  se mueve de una máquina  $M_i$  al buffer siguiente  $B_i$ , mediante el movimiento  $(j, 1)$ .

A continuación se analizarán los tiempos en cada caso.

- El instante de inicio del movimiento es el instante de disponibilidad del robot en el estado en curso:

$$\rho^w \tag{2.19}$$

- El tiempo de desplazamiento del robot desde su estación actual  $r^w$  hasta la estación  $l_j^w$  ocupada por la pieza  $P_j$ :

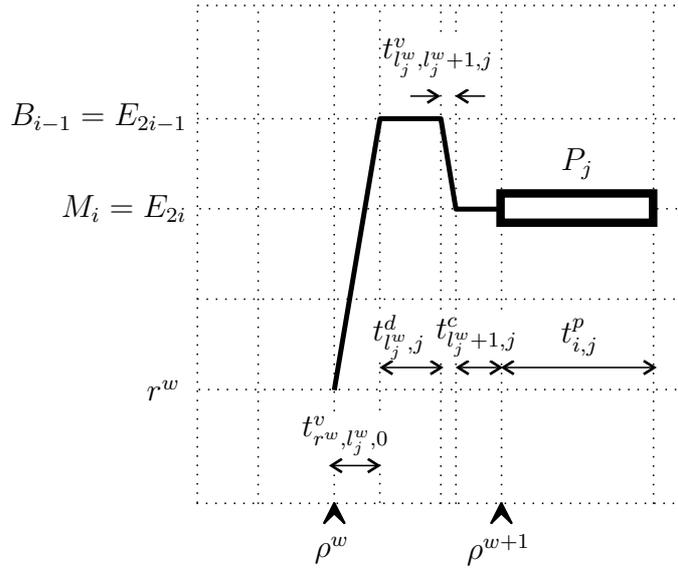


Figura 2.5: Tiempos de un movimiento del robot en el que una pieza  $P_j$  se mueve de un buffer  $B_{i-1}$  a la máquina siguiente  $M_i$ , mediante el movimiento  $(j, 1)$ .

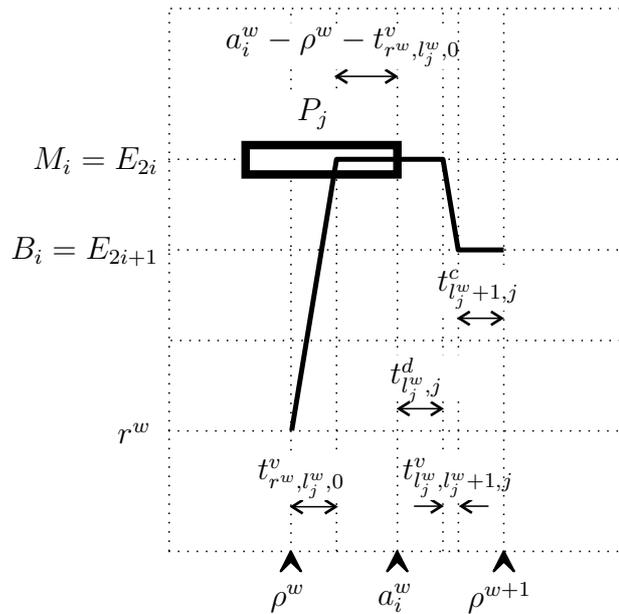


Figura 2.6: Tiempos de un movimiento del robot en el que una pieza  $P_j$  se mueve de una máquina  $M_i$  al buffer siguiente  $B_i$ , mediante el movimiento  $(j, 1)$ .

$$t_{r^w, l_j^w, 0}^v \tag{2.20}$$

- El tiempo de espera hasta la disponibilidad de la pieza.
  - ▷ Si  $P_j$  está en un buffer, su disponibilidad está asegurada por tanto el tiempo de espera es cero.
  - ▷ Si  $P_j$  está en una máquina  $M_i$  se producen dos casos:

- ▷ Si en el instante de llegada del robot a  $M_i$ , máquina ocupada por la pieza, ésta ha terminado de procesarse, el tiempo de espera es cero.
- ▷ En el otro caso, el robot deberá esperar en  $M_i$  hasta el instante de terminación  $a_i^w$ , un tiempo dado por:

$$a_i^w - \rho^w - t_{r^w, l_j^w, 0}^v \quad (2.21)$$

- ▶ El tiempo de descarga de la pieza  $P_j$  de la estación  $E_{l_j^w}$  que dicha pieza ocupa:

$$t_{l_j^w, j}^d \quad (2.22)$$

- ▶ El tiempo de transporte desde la estación de  $P_j$  hasta la estación de destino situada  $\delta$  estaciones hacia adelante:

$$t_{l_j^w, l_j^w + \delta, j}^v \quad (2.23)$$

donde  $\delta$  toma valor 1 cuando  $P_j$  se avanza una estación y toma valor 2 cuando se avanza dos estaciones.

- ▶ El tiempo requerido en cargar la pieza  $P_j$  en la estación destino:

$$t_{l_j^w + \delta, j}^c \quad (2.24)$$

Los tiempos de las ecuaciones 2.19 a 2.24 se suman en:

$$\begin{aligned} \rho^{w+1} = & \rho^w + t_{r^w, l_j^w, 0}^v + t_{l_j^w, j}^d + t_{l_j^w, l_j^w + \delta, j}^v + t_{l_j^w + \delta, j}^c + \\ & + \begin{cases} \max(0, a_i^w - \rho^w - t_{r^w, l_j^w, 0}^v) & \text{si } \text{resmed}(l_j^w) = 0; \\ 0 & \text{en otro caso.} \end{cases} \end{aligned} \quad (2.25)$$

O bien, moviendo términos dentro de 2.25:

$$\begin{aligned} \rho^{w+1} = & t_{l_j^w, j}^d + t_{l_j^w, l_j^w + \delta, j}^v + t_{l_j^w + \delta, j}^c + \\ & + \begin{cases} \max(\rho^w + t_{r^w, l_j^w, 0}^v, a_i^w) & \text{si } \text{resmed}(l_j^w) = 0; \\ \rho^w + t_{r^w, l_j^w, 0}^v & \text{en otro caso.} \end{cases} \end{aligned} \quad (2.26)$$

## 2.4.6 El estado al final del movimiento

Para todo movimiento factible, dada la información del estado  $e_w$  al comienzo del movimiento, y los tiempos del propio movimiento  $\mu_w$ , es posible encontrar el estado  $e_{w+1}$  alcanzado en el instante de finalización de dicho movimiento.

**Instante de terminación del robot.** La expresión del tiempo  $\rho^{w+1}$  al final del movimiento se determina mediante la expresión 2.25 o 2.26

**Instante de terminación de las máquinas.** Sea  $P_j$  una pieza que es movida a la máquina  $M_i$ . Por tratarse de una máquina se cumple que  $l_j^w + \delta$  es par, y el índice de la máquina  $M_i$  se determina por  $i = \text{entmed}(l_j^w + \delta)$ . El tiempo de terminación en  $M_i$  se modifica según el tiempo de proceso de  $P_j$  en la máquina donde ésta se incorpora, tomando como nuevo valor:

$$a_i^{w+1} = \rho^{w+1} + t_{i,j}^p \quad (2.27)$$

donde  $i = \text{entmed}(l_j^w + \delta)$

Si la estación donde  $P_j$  estaba originalmente es una máquina (y por tanto el tiempo de terminación de dicha máquina era algún valor mayor o igual a cero), dicha máquina pasa a estar vacía y por tanto al tiempo de terminación se le asigna un valor arbitrario negativo (por ejemplo  $-1$ ) para indicar que ya no hay piezas en  $M_i$ :

$$a_i^{w+1} = -1 \quad (2.28)$$

donde  $i = \text{entmed}(l_j^w)$

Agrupando las condiciones expresadas por las ecuaciones 2.27 y 2.28 extendida para toda máquina  $M_i$ :

$$a_i^{w+1} = \begin{cases} \rho^{w+1} + t_{i,j}^p & \text{si } l_j^w + \delta = 2i; \\ -1 & \text{si } l_j^w = 2i; \\ a_i^w & \text{en otro caso.} \end{cases} \quad (2.29)$$

$i = 1, \dots, m$

**Posición de las piezas.** La pieza movida  $P_j$  se hace avanzar  $\delta$  estaciones. Por lo tanto:

$$l_j^{w+1} = l_j^w + \delta \quad (2.30)$$

Las condición de la ecuación 2.30, extendida para toda pieza  $P_h$ :

$$l_h^{w+1} = \begin{cases} l_h^w + \delta & \text{si } h = j; \\ l_h^w & \text{en otro caso.} \end{cases} \quad (2.31)$$

$h = 1, \dots, n$

**Posición del robot.** Dado que el movimiento siempre finaliza con la carga de una pieza en una estación, la posición del robot debe coincidir con la posición de la pieza.

$$r^{w+1} = l_j^w + \delta \quad (2.32)$$

**Secuencia de piezas.** La secuencia de piezas determinada hasta el momento se modifica sólo cuando se produce un movimiento desde la estación de entrada a la primera máquina.

Sean

- ▶  $z_1^w$  el número de piezas que ya han ingresado en la máquina  $M_1$  (ecuaciones 2.15 y 2.16),
- ▶ y  $\eta = z_1^w + 1$  la posición en la secuencia  $\bar{\pi}$  de la próxima pieza a ingresar en la máquina  $M_1$ .

$$\pi_h^{w+1} = \begin{cases} \pi_h^w, & \text{si } h < \eta \\ j, & \text{si } h = \eta \text{ y } l_j^w = 1 \\ 0, & \text{si } h > \eta \end{cases} \quad (2.33)$$

$h = 1, \dots, n$

## 2.5 Funciones temporales auxiliares para las secuencias de operaciones del robot

En éstas se agrupan los tiempos de secuencias de operaciones del robot que, por la estructura de los modelos de acotación presentados en los capítulos 3, 4 y 5, aparecen frecuentemente. De aquí proceden las dos funciones auxiliares *tiempo de secuencia de carga* y *tiempo de secuencia de descarga*, denotadas respectivamente  $\theta_c$  y  $\theta_d$ .

Se define la función auxiliar *tiempo de secuencia de carga*  $\theta_c$  como el tiempo requerido por el robot en la ejecución de tres operaciones a cuyo término se produce la ocupación de la máquina  $M_i$  con la pieza  $P_j$ , cuando ésta procede de  $q$  estaciones anteriores a  $M_i$ . La expresión 2.34 agrupa los términos: tiempo de descarga de la estación de origen (estación del robot al inicio de la primera operación de la secuencia:  $E_{2i-q}$ )  $t_{2i-q,j}^d$ , tiempo de transporte entre la estación de origen y la máquina destino (estación del robot al final de la última operación de la secuencia:  $E_{2i}$ )  $t_{2i-q,2i,j}^v$  y tiempo de carga en la máquina destino  $t_{2i,j}^c$ .

$$\theta_c(i, j, q) = \begin{cases} t_{1,j}^d + t_{1,2,j}^v + t_{2,j}^c & \text{si } i = 1 \\ t_{2i-q,j}^d + t_{2i-q,2i,j}^v + t_{2i,j}^c & \text{en otro caso.} \end{cases} \quad (2.34)$$

$q \in (1, 2) ; 1 \leq i \leq m ; 1 \leq j \leq n$

Se define la función auxiliar *tiempo de secuencia de descarga*  $\theta_d$  como el tiempo requerido por el robot en la ejecución de tres operaciones a cuyo término es alcanzada la disponibilidad de la máquina  $M_i$  (quedando ésta vacía), cuando la pieza  $P_j$  es adelantada  $q$

estaciones. La expresión 2.35 agrupa los términos: tiempo de descarga de la máquina origen  $t_{2i,j}^d$ , tiempo de transporte desde la máquina origen a la estación destino ( $E_{2i+q}$ )  $t_{2i,2i+q,j}^v$  y tiempo de carga en la estación destino  $t_{2i+q,j}^c$ .

$$\theta_d(i, j, q) = \begin{cases} t_{2m,j}^d + t_{2m,2m+1,j}^v + t_{2m+1,j}^c & \text{si } i = m \\ t_{2i,j}^d + t_{2i,2i+q,j}^v + t_{2i+q,j}^c & \text{en otro caso.} \end{cases} \quad (2.35)$$

$q \in (1, 2) ; 1 \leq i \leq m ; 1 \leq j \leq n$

## 2.6 Cotas auxiliares

En este apartado se definen tres funciones de acotación auxiliares que, igual que las funciones de tiempo de secuencia de carga y descarga del apartado 2.5, también aparecen frecuentemente en las distintas expresiones principales de acotación que se definirán en los capítulos 3, 4 y 5.

Primeramente se definen dos funciones auxiliares para la determinación de la validez de las distintas estaciones, necesarias para las expresiones de las cotas auxiliares. Posteriormente se definen “cota de carga”, “cota de descarga” y “cota de descarga seguida de carga”.

### 2.6.1 Validez de las estaciones

Para simplificar la notación de las expresiones de acotación de los apartados 2.6.2, 2.6.3 y 2.6.4, en este apartado se introducen los parámetros  $h_{i,p}^-$  y  $h_{i,q}^+$  para determinar la validez de las estaciones dentro de un vecindario cercano (de un máximo de dos estaciones) de una determinada máquina  $M_i$ .

Una estación se dice inválida (y por ello no se puede realizar operaciones del robot que impliquen extraer o depositar piezas en dicha estación) cuando son cumplidas las siguientes condiciones:

- ▶ Cuando una estación no existe (por ejemplo no existen la máquina anterior a  $M_1$  y la máquina siguiente a  $M_m$ ),
- ▶ y cuando un buffer tiene capacidad nula (recuérdese que se incorpora la ausencia de buffers mediante la asignación nula al valor de su capacidad, como se indicó en el apartado 2.2, página 28).

Mientras que una estación se dice válida cuando no se cumplen dichas condiciones.

**Parámetro  $h_{i,p}^-$ .** El parámetro  $h_{i,p}^-$  expresa la validez de aquella estación anterior a la máquina  $M_i$  ( $i = 1, \dots, m$ ), distante de ésta  $p$  estaciones, siendo  $p \in \{1, 2\}$ . El valor  $h_{i,p}^- = 1$  indica que la estación  $E_{2i-p}$  es una estación válida, es decir que puede contener (al menos) una pieza. En el caso contrario  $h_{i,p}^- = 0$ .

En el caso en que  $p = 1$  son posibles los siguientes casos.

- ▶ **Caso en que**  $i = 1$ . En este caso se trata de la estación de entrada de piezas (estación anterior a  $M_1$ ). Dado que, como se mencionó en el apartado 2.2 (página 27), la estación de entrada tiene capacidad suficiente para contener todas las piezas, entonces  $h_{i,1}^- = 1$ .
- ▶ **Caso en que**  $i > 1$ . En este caso se analiza la validez del buffer  $B_{i-1}$ . En este caso  $h_{i,1}^-$  tiene el valor 1 cuando  $b_{i-1} > 0$  y el valor 0 en el caso contrario.

Estos casos son expresados por la ecuación 2.36.

$$h_{i,1}^- = \begin{cases} 1 & \text{si } i = 1 \text{ ó } (i > 1 \text{ y } b_{i-1} > 0) \\ 0 & \text{en caso contrario} \end{cases} \quad (2.36)$$

$$1 \leq i \leq m$$

Por otra parte, en el caso en que  $p = 2$ , es decir, cuando se expresa la validez de la máquina  $M_{i-1}$ , son posibles los siguientes casos.

- ▶ **Caso en que**  $i = 1$ . Dado que la máquina anterior a la primera máquina  $M_1$  no existe, entonces  $h_{i,2}^- = 0$ .
- ▶ **Caso en que**  $i > 1$ . En este caso  $h_{i,2}^- = 1$  dado que las máquinas pueden siempre almacenar al menos una pieza.

Estos casos son expresados por la ecuación 2.37.

$$h_{i,2}^- = \begin{cases} 1 & \text{si } i > 1 \\ 0 & \text{en caso contrario} \end{cases} \quad (2.37)$$

$$1 \leq i \leq m$$

**Parámetro**  $h_{i,q}^+$ . El parámetro  $h_{i,q}^+$  expresa la validez de aquella estación siguiente a la máquina  $M_i$  ( $i = 1, \dots, m$ ), distante de ésta  $q$  estaciones, siendo  $q \in \{1, 2\}$ . El valor  $h_{i,q}^+ = 1$  indica que la estación  $E_{2i+q}$  es una estación válida, es decir que puede contener (al menos) una pieza. En el caso contrario  $h_{i,q}^+ = 0$ .

En el caso en que  $q = 1$  son posibles los siguientes casos.

- ▶ **Caso en que**  $i = m$ . En este caso se analiza la validez de la estación de salida  $O$ . Dado que, como se mencionó en el apartado 2.2 (página 27), la estación de salida tiene capacidad suficiente para contener todas las piezas, entonces  $h_{i,1}^+ = 1$ .
- ▶ **Caso en que**  $i < m$ . En este caso se analiza la validez del buffer  $B_i$  (buffer siguiente a  $M_i$ ). En este caso  $h_{i,1}^+$  tiene el valor 1 cuando  $b_i > 0$  y el valor 0 en el caso contrario.

Estos casos son expresados por la ecuación 2.38.

$$h_{i,1}^+ = \begin{cases} 1 & \text{si } i = m \text{ ó } (i < m \text{ y } b_i > 0) \\ 0 & \text{en caso contrario} \end{cases} \quad (2.38)$$

$$1 \leq i \leq m$$

En el caso en que  $q = 2$ , es decir, cuando se expresa la validez de la máquina  $M_{i+1}$ , son posibles los siguientes casos.

- **Caso en que**  $i = m$ . Dado que no existe la máquina siguiente a  $M_m$ , entonces  $h_{i,2}^- = 0$ .
- **Caso en que**  $i < m$ . En este caso  $h_{i,2}^+ = 1$ , dado que las máquinas siempre pueden almacenar al menos una pieza.

Estos casos son expresados por la ecuación 2.39.

$$h_{i,2}^+ = \begin{cases} 1 & \text{si } i < m \\ 0 & \text{en caso contrario} \end{cases} \quad (2.39)$$

$$1 \leq i \leq m$$

Las ecuaciones 2.36 y 2.37 pueden agruparse en la ecuación 2.40.

$$h_{i,p}^- = \begin{cases} 1 & \text{si } (i = 1 \text{ y } p = 1) \text{ ó } (i > 1 \text{ y } b_{i-1} > 0 \text{ y } p = 1) \text{ ó } (i > 1 \text{ y } p = 2) \\ 0 & \text{en caso contrario} \end{cases} \quad (2.40)$$

$$p \in \{1, 2\}$$

$$1 \leq i \leq m$$

Y las ecuaciones 2.38 y 2.39 pueden agruparse en la ecuación 2.41.

$$h_{i,q}^+ = \begin{cases} 1 & \text{si } (i = m \text{ y } q = 1) \text{ ó } (i < m \text{ y } b_i > 0 \text{ y } q = 1) \text{ ó } (i < m \text{ y } q = 2) \\ 0 & \text{en caso contrario} \end{cases} \quad (2.41)$$

$$q \in \{1, 2\}$$

$$1 \leq i \leq m$$

### 2.6.2 La cota de carga $\delta^c$

La función “cota de carga” de parámetros  $i$  y  $j$  y notación  $\delta_{i,j}^c$  (figura 2.7) provee la cota del tiempo de fin de ingreso de la pieza  $P_j$  en la máquina  $M_i$  proveniente de una estación anterior.

Por supuestos que serán asumidos en el capítulo 4 (donde esta cota será empleada), se supondrá que en el instante inicial:

- ▶ la pieza  $P_j$  está disponible de modo que no se consideran esperas del robot, y
- ▶ el robot está situado en la estación válida siguiente a la máquina  $M_i$ , es decir, la estación  $E_{2i+2-h_{i,1}^+}$ .

Son posibles dos secuencias de operaciones mediante las cuales es posible el ingreso de la pieza  $P_j$  en la máquina  $M_i$ :

- ▶ aquella en la que  $P_j$  procede de la estación anterior a  $M_i$  (del buffer  $B_{i-1}$  o de la estación de entrada  $I$ ), en donde se emplean los tiempos:

$$t_{2i+2-h_{i,1}^+, 2i-1, 0}^v + \theta_c(i, j, 1) \quad \text{si } h_{i,1}^- = 1 \quad (2.42)$$

- ▶ y, aquella en la que  $P_j$  procede de la máquina anterior a  $M_i$  (máquina  $M_{i-1}$ ), en donde se emplean los tiempos:

$$t_{2i+2-h_{i,1}^+, 2i-2, 0}^v + \theta_c(i, j, 2) \quad \text{si } h_{i,2}^- = 1 \quad (2.43)$$

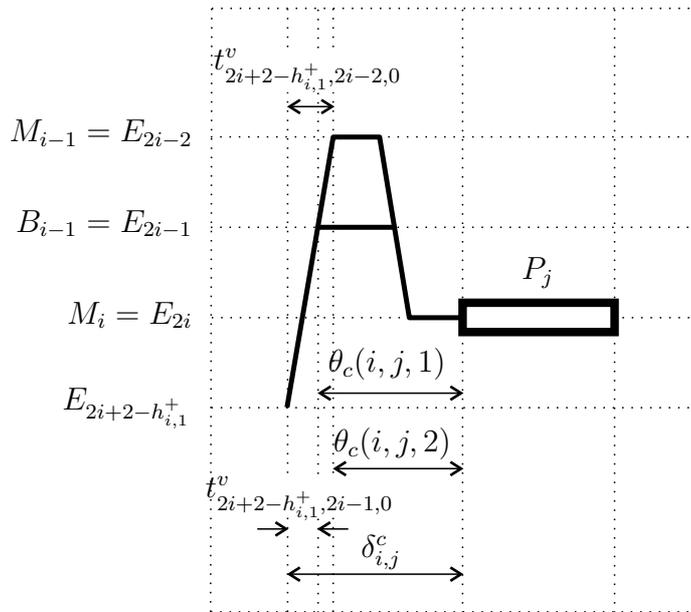


Figura 2.7: Los tiempos de la cota de carga  $\delta^c$ .

El tiempo asociado a la cota de carga  $\delta^c$ , expresado por la ecuación 2.44, resulta de tomar el valor mínimo dado por de las expresiones 2.42 y 2.43.

$$\delta_{i,j}^c = \min_{p \in \{1,2\} | h_{i,p}^- = 1} (t_{2i+2-h_{i,1}^+, 2i-p, 0}^v + \theta_c(i, j, p)) \quad (2.44)$$

### 2.6.3 La cota de descarga $\delta^d$

La función “cota de descarga” de parámetros  $i$  y  $j$  y notación  $\delta_{i,j}^d$  (figura 2.8) provee la cota del tiempo de fin de la extracción de la pieza  $P_j$  de la máquina  $M_i$  llevándola a una estación siguiente.

Son posibles dos secuencias de operaciones mediante las cuales es posible la extracción de la pieza  $P_j$  de la máquina  $M_i$  llevándola a una estación siguiente:

- aquella en la que  $P_j$  es cargada en la estación siguiente a  $M_i$  (en el buffer  $B_i$  o en la estación de salida  $O$ ), en donde se emplean los tiempos:

$$\theta_d(i, j, 1) \quad \text{si } h_{i,1}^+ = 1 \quad (2.45)$$

- y, aquella en la que  $P_j$  es cargada en la máquina siguiente a  $M_i$  (máquina  $M_{i+1}$ ), en donde se emplean los tiempos:

$$\theta_d(i, j, 2) \quad \text{si } h_{i,2}^+ = 1 \quad (2.46)$$

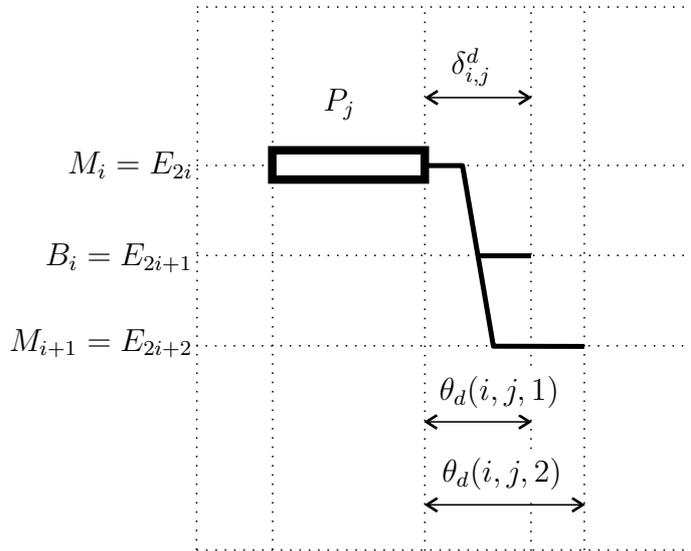


Figura 2.8: Los tiempos de la cota de descarga  $\delta^d$ .

El tiempo asociado a la cota de descarga  $\delta^d$ , expresado por la ecuación 2.47, resulta de tomar el valor mínimo dado por de las expresiones 2.45 y 2.46.

$$\delta_{i,j}^d = \min_{q \in \{1,2\} | h_{i,q}^+ = 1} \theta_d(i, k, q) \quad (2.47)$$

### 2.6.4 La cota de descarga seguida de carga $\delta^e$

La función “cota de descarga seguida de carga” de parámetros  $i, j$  y  $k$  y notación  $\delta_{i,j,k}^e$  (figura 2.9) provee la cota del tiempo de fin de ingreso de la pieza  $P_k$  en la máquina  $M_i$  cuando dicha máquina está previamente cargada con la pieza  $P_j$ . Es decir, una cota de los tiempos asociados a las operaciones necesarias para descargar la pieza  $P_j$  de la máquina  $M_i$  primero y, a continuación, cargar la pieza  $P_k$  en la misma máquina  $M_i$ .

Son posibles dos secuencias de operaciones que permiten la extracción de  $P_j$  de  $M_i$ :

- ▶ aquella en la que  $P_j$  es cargada en la estación siguiente a  $M_i$  (en el buffer  $B_i$  o en la estación de salida  $O$ ),
- ▶ y, aquella en la que  $P_j$  es cargada en la máquina siguiente a  $M_i$  (máquina  $M_{i+1}$ ).

Por otro lado, son posibles dos secuencias de operaciones que permiten el ingreso de  $P_k$  en la máquina  $M_i$ :

- ▶ aquella en la que  $P_k$  procede de la estación anterior a  $M_i$  (del buffer  $B_{i-1}$  o de la estación de entrada  $I$ ),
- ▶ y, aquella en la que  $P_k$  procede de la máquina anterior a  $M_i$  (máquina  $M_{i-1}$ ).

Las dos secuencias de operaciones que permiten la extracción de la pieza  $P_j$  de la máquina  $M_i$  y las dos secuencias que permiten el ingreso de la pieza  $P_k$  en la máquina  $M_i$  forman las siguientes cuatro opciones:

- ▶ aquella secuencia de operaciones en la que  $P_j$  es cargada la estación siguiente a  $M_i$  (en el buffer  $B_i$  o en la estación de salida  $O$ ) seguida de aquella en la que  $P_k$  se extrae de la estación anterior a  $M_i$  (del buffer  $B_{i-1}$  o de la estación de entrada  $I$ ), en donde se emplean los tiempos:

$$\theta_d(i, j, 1) + t_{2i+1, 2i-1, 0}^v + \theta_c(i, k, 1) \quad \text{si } h_{i,1}^+ = 1 \text{ y } h_{i,1}^- = 1 \quad (2.48)$$

- ▶ aquella secuencia de operaciones en la que  $P_j$  es cargada la estación siguiente a  $M_i$  (en el buffer  $B_i$  o en la estación de salida  $O$ ) seguida de aquella en la que  $P_k$  se extrae de la máquina anterior a  $M_i$  (máquina  $M_{i-1}$ ), en donde se emplean los tiempos:

$$\theta_d(i, j, 1) + t_{2i+1, 2i-2, 0}^v + \theta_c(i, k, 2) \quad \text{si } h_{i,1}^+ = 1 \text{ y } h_{i,2}^- = 1 \quad (2.49)$$

- ▶ aquella secuencia de operaciones en la que  $P_j$  es cargada en la máquina siguiente a  $M_i$  (máquina  $M_{i+1}$ ) seguida de aquella en la que  $P_k$  procede de la estación anterior a  $M_i$  (del buffer  $B_{i-1}$  o en la estación de entrada  $I$ ), en donde se emplean los tiempos:

$$\theta_d(i, j, 2) + t_{2i+2, 2i-1, 0}^v + \theta_c(i, k, 1) \quad \text{si } h_{i,2}^+ = 1 \text{ y } h_{i,1}^- = 1 \quad (2.50)$$

- y, aquella secuencia de operaciones en la que  $P_j$  es cargada en la máquina siguiente a  $M_i$  (máquina  $M_{i+1}$ ) seguida de aquella en la que  $P_k$  procede de la máquina anterior a  $M_i$  (máquina  $M_{i-1}$ ), en donde se emplean los tiempos:

$$\theta_d(i, j, 2) + t_{2i+2, 2i-2, 0}^v + \theta_c(i, k, 2) \quad \text{si } h_{i,2}^+ = 1 \text{ y } h_{i,2}^- = 1 \quad (2.51)$$

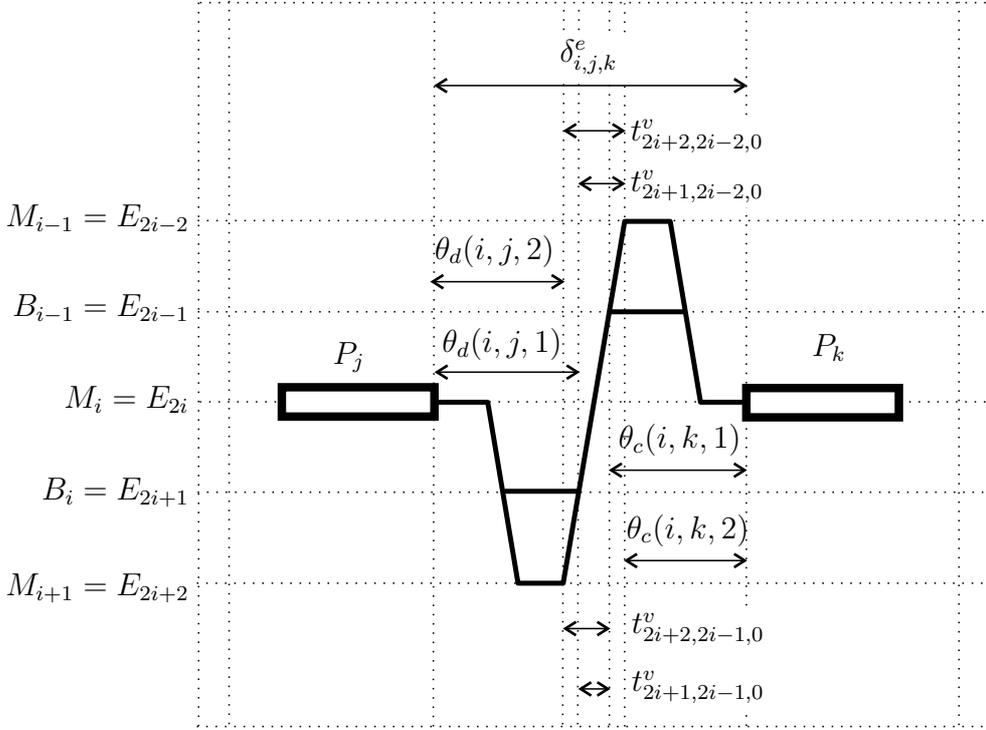


Figura 2.9: Los tiempos de la cota de descarga seguida de carga  $\delta^e$ .

El tiempo asociado a la cota de descarga seguida de carga  $\delta^e$ , expresado por la ecuación 2.52, resulta de tomar el valor mínimo dado por de las expresiones 2.48, 2.49, 2.50 y 2.51.

$$\delta_{i,j,k}^e = \min_{\substack{p \in \{1,2\} | h_{i,p}^- = 1 \\ q \in \{1,2\} | h_{i,q}^+ = 1}} (\theta_d(i, j, q) + t_{2i+q, 2i-p, 0}^v + \theta_c(i, k, p)) \quad (2.52)$$

# Capítulo 3

## Cota inferior basada en los tiempos de las máquinas cuando la secuencia de piezas es dada

### 3.1 Introducción

En este capítulo se presenta un procedimiento de acotación basado en grafos para el caso especial en que la secuencia de piezas es conocida. Este procedimiento será empleado en los procedimientos de resolución basados en separación de los procesos de resolución de la secuencia de piezas y la secuencia de movimientos del robot, (codificados **SEP** y **SEPA**) presentados en el capítulo 7.

En el caso especial de problemas de programación de tipo flowshop permutativos con tiempos de viaje nulos, tiempos de carga y descarga también nulos y buffers de capacidad ilimitada, es posible definir un sistema de precedencias entre los instantes de terminación de las piezas en las máquinas. El grafo orientado de la figura 3.1 es una representación de este sistema de precedencias aplicado a un flowshop permutativo de  $n$  piezas y  $m$  máquinas.

En dicho grafo, los arcos se etiquetan con los tiempos de procesamiento  $\mathcal{P}_{i,j}$  que denotan el tiempo de procesamiento de la pieza secuenciada en la  $j$ -ésima posición en la máquina  $M_i$ . Los vértices se etiquetan con los tiempos de terminación  $S_{i,j}$  y representan el instante de terminación de la pieza secuenciada en la  $j$ -ésima posición en la máquina  $M_i$ .

Mediante el empleo de dicho grafo es posible determinar el instante de terminación del programa hallando la longitud del camino más largo, desde el vértice inicial  $I$  hasta el vértice  $S_{m,n}$  (tiempo de terminación en la última máquina de la pieza secuenciada en último lugar) <sup>1</sup> cuando la secuencia de piezas es conocida.

---

<sup>1</sup>El camino más largo entre el primero y el último de los vértices de un grafo de tareas, en el contexto de gestión de proyectos, suele ser denominado camino crítico por una buena parte de la literatura.

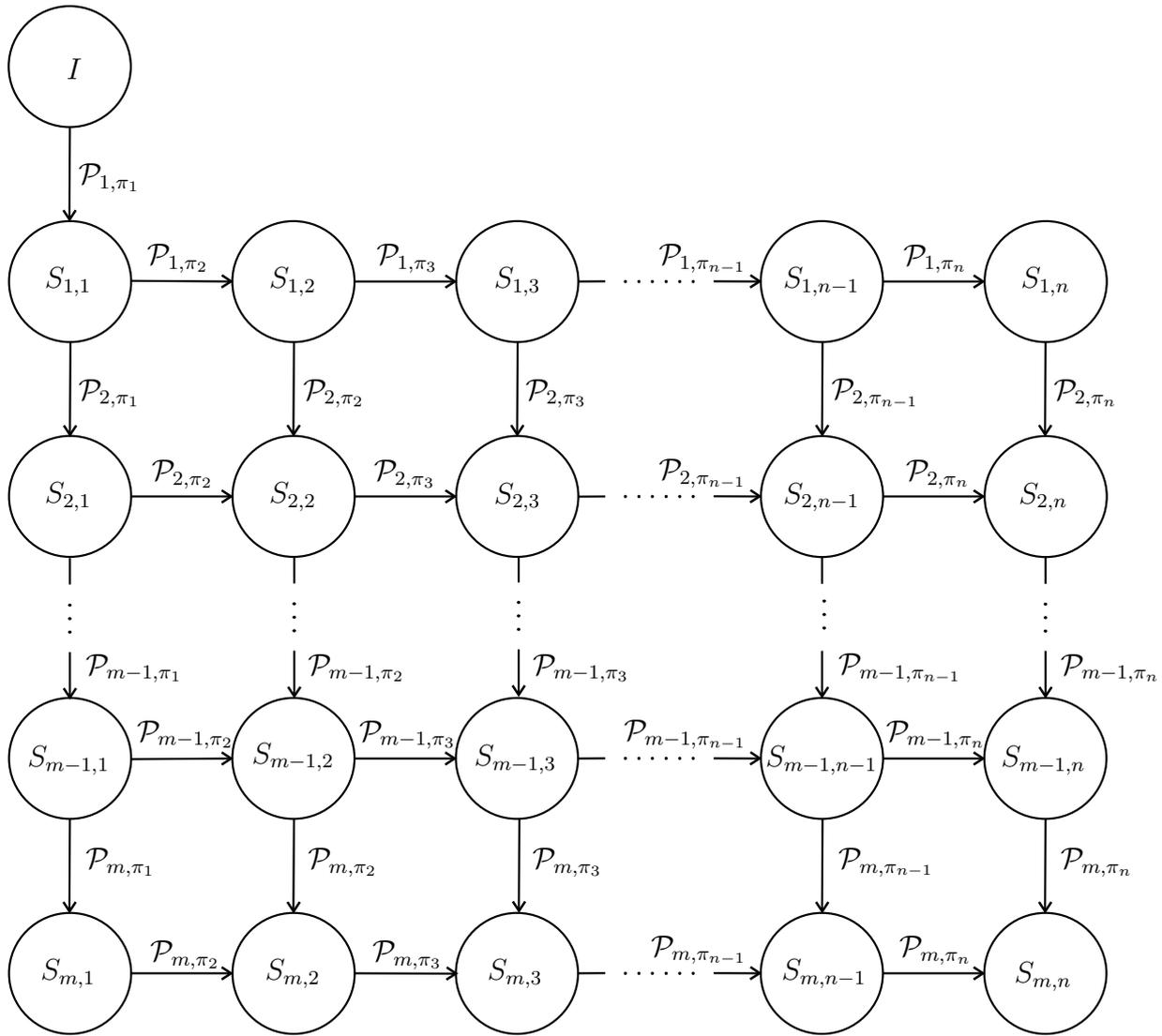


Figura 3.1: Grafo para hallar el tiempo de terminación en flowshops permutativos con tiempos de viaje, tiempos de carga y tiempos de descarga nulos y buffers de capacidad ilimitada cuando la secuencia de piezas es dada.

El conjunto de ecuaciones 3.1 a 3.4 proveen un procedimiento equivalente a determinar la longitud del camino más largo del grafo de la figura 3.1 (pues las ecuaciones proceden del grafo) tal que resueltas iterativamente, permiten hallar el tiempo de terminación del conjunto de piezas  $S_{m,n}$ .

$$S_{1,1} = P_{1,\pi_1} \tag{3.1}$$

$$S_{1,k} = S_{1,k-1} + P_{1,\pi_k} \quad k = 2, \dots, n \tag{3.2}$$

$$S_{i,1} = S_{i-1,1} + P_{i,\pi_1} \quad i = 2, \dots, m \tag{3.3}$$

$$S_{i,k} = \max (S_{i-1,k}, S_{i,k-1}) + P_{i,\pi_k} \quad i = 2, \dots, m, \quad k = 2, \dots, n \tag{3.4}$$

## 3.2 Grafo de acotación

En este apartado se presenta un método de acotación basado en el grafo de la figura 3.1 para la determinación de una cota del tiempo de terminación del conjunto de  $n$  piezas en una celda robotizada de  $m$  máquinas. Para esto se definirán los elementos del grafo: su topología, los vértices y los arcos, para adaptarlo al problema de esta tesis. De aquí que las ecuaciones 3.1 a 3.4 también deberán cambiar para satisfacer las condiciones de este problema. Las condiciones que deben ser incorporadas al modelo son las siguientes.

- ▶ El grafo (o sus ecuaciones) se resuelven para diferentes estados posibles del sistema, por tanto se deberá introducir en el grafo las distintas magnitudes que dependan del estado que se está analizando.
- ▶ Los tiempos en los arcos deben incorporar cotas basadas en los tiempos del robot: viaje, carga y descarga.
- ▶ Como los tiempos de los arcos son cotas de tiempos, los vértices  $S_{i,k}$  representarán cotas de los instantes de salida (o instantes de terminación) de las piezas de las distintas máquinas.

Cuando en el grafo se introducen las condiciones anteriores, se obtiene el grafo de la figura 3.2, donde se incorporan nuevos arcos etiquetados  $I_k$  y  $U$ ; y se han cambiado las etiquetas de otros por  $L_{i,k}$  y  $T_{i,k}$ . Los vértices  $S_{i,k}$  mantienen su etiquetado, si embargo, en el contexto del problema tratado aquí, cambia su significado. Con este etiquetado las ecuaciones iterativas que proveen el valor del camino más largo, para el problema de esta tesis, deben ser reescritas como:

$$S_{1,1} = I_1 \tag{3.5}$$

$$S_{1,k} = \max (I_k, S_{1,k-1} + L_{1,k}) \quad k = 2, \dots, n \tag{3.6}$$

$$S_{i,1} = S_{i-1,1} + T_{i,1} \quad i = 2, \dots, m \tag{3.7}$$

$$S_{i,k} = \max (S_{i-1,k} + T_{i,k}, S_{i,k-1} + L_{i,k}) \quad i = 2, \dots, m, \quad k = 2, \dots, n \tag{3.8}$$

$$S = S_{m,n} + U \tag{3.9}$$

Donde  $S$  (ecuación 3.9) denota una cota inferior del instante de terminación del programa. Los vértices  $S_{i,k}$  se tratan en el apartado 3.2.1. Las funciones de etiquetado  $I_k$  se explican en el apartado 3.2.2. Las funciones  $L_{i,k}$  se tratan en el apartado 3.2.3. Las funciones de etiquetado  $T_{i,k}$  se explican en el apartado 3.2.4. La función  $U$  se explica en el apartado 3.2.5.

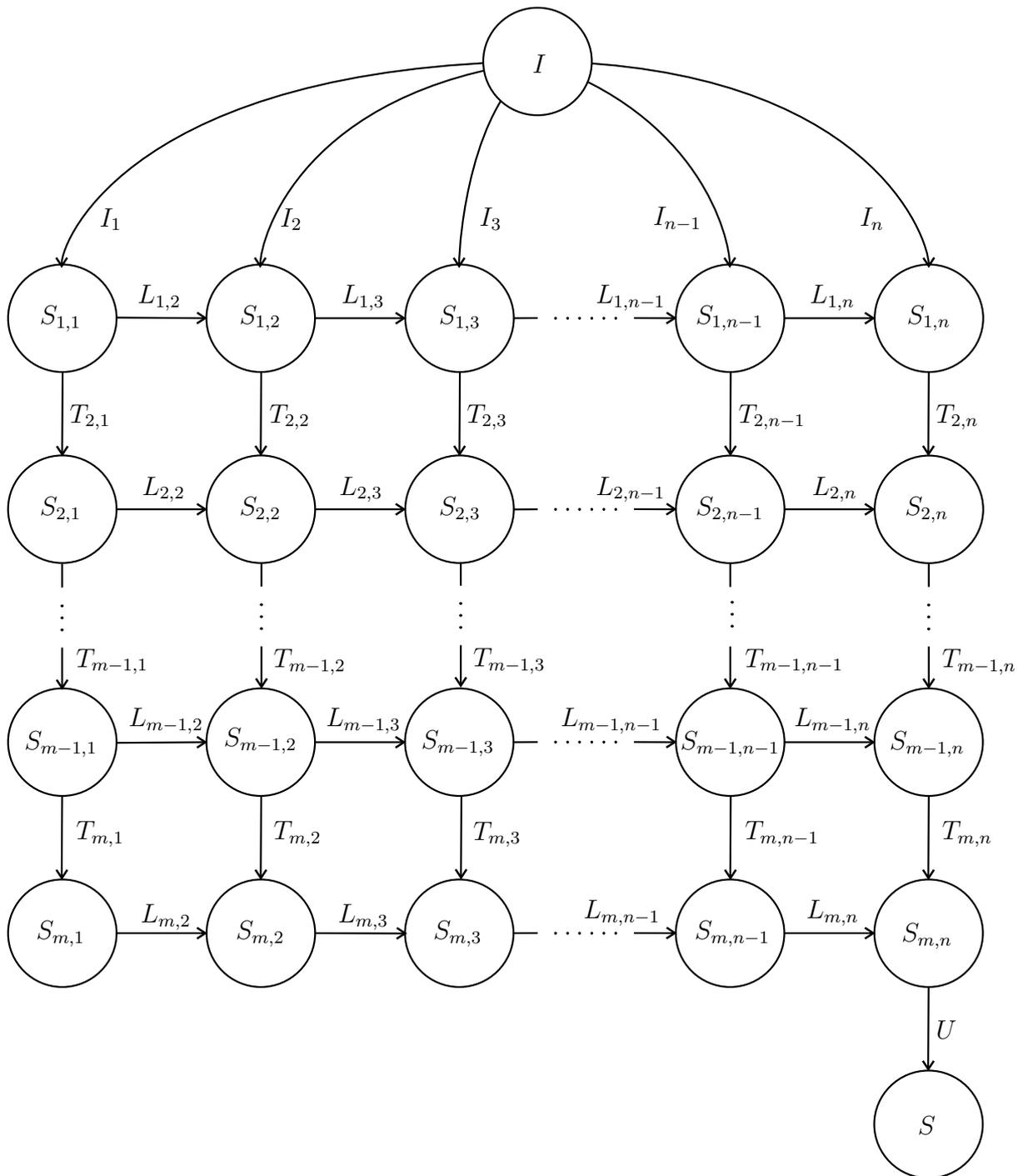


Figura 3.2: Grafo para hallar la cota del tiempo de terminación en flowshops robotizados cuando la secuencia de piezas es dada.

### 3.2.1 Cota del instante de salida $S_{i,k}$

En el contexto del problema tratado en esta tesis, la *cota del instante de salida*  $S_{i,k}$  se define como la cota inferior del instante en que es posible el inicio del movimiento que produce la salida de la pieza  $P_{\pi_k}$  (pieza que ocupa la posición  $k$  en la secuencia de piezas  $\bar{\pi}$ ) de la máquina  $M_i$ . Dicho instante verifica simultáneamente las dos condiciones siguientes:

- ▶ la pieza  $P_{\pi_k}$  ya ha terminado de procesarse en la máquina  $M_i$
- ▶ y el robot se encuentra también situado en  $M_i$ .

Estas condiciones aseguran que tanto la pieza, la máquina y el robot están disponibles para que el movimiento de salida de la pieza  $P_{\pi_k}$  de la máquina  $M_i$  pueda iniciarse.

### 3.2.2 Arcos tipo $I_k$

En los arcos  $I_k$  se tiene una cota del instante en que, a partir del estado en curso, cada pieza del sistema alcanza su disponibilidad, es decir, termina de procesarse en una máquina y simultáneamente el robot queda situado en dicha máquina. Más concretamente, en  $I_k$  se tiene una cota del instante de terminación de la pieza  $P_{\pi_k}$  en la siguiente máquina en la que debe ser procesada. Se presentan dos casos:

- ▶ si la pieza  $P_{\pi_k}$  está en una máquina, entonces  $I_k$  será la cota del instante en el que  $P_{\pi_k}$  podría empezar a ser descargada de esa misma máquina;
- ▶ si la pieza  $P_{\pi_k}$  está en un buffer o en la estación de entrada, entonces  $I_k$  será la cota del instante en el que  $P_{\pi_k}$  podría empezar a ser descargada de la máquina inmediata siguiente.

Para este análisis se define  $i = \text{entmed}(l_{\pi_k})$ , que, como fue definido en el apartado 2.2.1, define el índice de la máquina o del buffer que es ocupado por  $P_{\pi_k}$ . Es decir, por esta definición,  $P_{\pi_k}$  ocupa, bien  $M_i$ , bien  $B_i$  (buffer siguiente a  $M_i$ ).

De acuerdo con la ubicación de la pieza  $P_{\pi_k}$  son posibles tres casos:

**Caso en que  $P_{\pi_k}$  está en la estación de salida.** Se expresa mediante la condición  $l_{\pi_k} = 2m + 1$ . Dado que no hay siguiente máquina para la última estación,  $I_k$  toma el valor del tiempo de terminación del robot del estado en curso  $\rho$ .

$$\begin{aligned} I_k &= \rho \\ \text{si } l_{\pi_k} &= 2m + 1 \end{aligned} \tag{3.10}$$

**Caso en que  $P_{\pi_k}$  está en una máquina.** La pieza  $P_{\pi_k}$  está ocupando la máquina  $M_i$  y por tanto la estación  $E_{2i}$ . La pertenencia a este caso se expresa mediante las condiciones:  $l_{\pi_k} < 2m + 1$  (pues no ocupa la estación de salida) y  $l_{\pi_k} = 2i$ .

Bajo estas condiciones  $I_k$  se define como el máximo de dos tiempos:

- ▶ el tiempo de terminación de  $P_{\pi_k}$  en  $M_i$ :  $a_i$ ;
- ▶ y el instante de tiempo en el que el robot sin pieza llega a  $M_i$  (a tomar  $P_{\pi_k}$ ):  $\rho + t_{r,2i,0}^v$

como se expresa en:

$$I_k = \max(a_i, \rho + t_{r,2i,0}^v) \quad (3.11)$$

si  $l_{\pi_k} < 2m + 1$  y  $l_{\pi_k} = 2i$

**Caso en que  $P_{\pi_k}$  está en un buffer o en la estación de entrada.** La pieza  $P_{\pi_k}$  está ocupando el buffer  $B_{i-1}$  anterior a  $M_i$  (o  $I$  anterior a  $M_1$ ) y por tanto la estación  $E_{2i-1}$ . La pertenencia al caso se expresa mediante las condiciones:  $l_{\pi_k} < 2m + 1$  (pues no ocupa la estación de salida) y  $l_{\pi_k} = 2i - 1$ .

Bajo estas condiciones  $I_k$  se define como la suma de tiempos de operaciones que están implicadas en depositar la pieza en  $M_i$  para su procesamiento cuando, por la hipótesis del caso, se encuentra en el buffer anterior  $B_{i-1}$  (o  $I$  cuando  $i = 1$ ). La expresión de la cota  $I_k$  incorpora los siguientes tiempos:

- ▶ el instante de disponibilidad del robot en el estado en curso,  $\rho$ ;
- ▶ el tiempo de viaje del robot desde la estación donde se encuentra en el estado en curso hasta el buffer  $B_{i-1}$  o  $I$ ,  $t_{r,2i-1,0}^v$ ;
- ▶ el tiempo de descarga de  $B_{i-1}$  o  $I$ , el tiempo de viaje hasta  $M_i$ , el tiempo de carga en  $M_i$ , tiempos todos incluidos en  $\theta_c(i, \pi_k, 1)$ ;
- ▶ el tiempo de proceso de  $P_{\pi_k}$  en  $M_i$ ,  $t_{i,\pi_k}^p$ .

Los tiempos anteriores se suman en la expresión:

$$I_k = \rho + t_{r,2i-1,0}^v + \theta_c(i, \pi_k, 1) + t_{i,\pi_k}^p \quad (3.12)$$

si  $l_{\pi_k} < 2m + 1$  y  $l_{\pi_{k+1}} = 2i - 1$

**Expresión completa de  $I_k$ .** Los resultados de los casos expresados por las ecuaciones 3.10, 3.11 y 3.12, se reúnen en la expresión 3.13.

$$I_k = \begin{cases} \rho & \text{si } l_{\pi_k} = 2m + 1 \\ \max(a_i, \rho + t_{r,2i,0}^v) & \text{si } l_{\pi_k} < 2m + 1 \text{ y } l_{\pi_k} = 2i \\ \rho + t_{r,2i-1,0}^v + \theta_c(i, \pi_k, 1) + t_{i,\pi_k}^p & \text{si } l_{\pi_k} < 2m + 1 \text{ y } l_{\pi_k} = 2i - 1 \end{cases} \quad (3.13)$$

### 3.2.3 Arcos tipo $L_{i,k}$

$L_{i,k}$  es la cota asociada al tiempo transcurrido entre el fin del procesamiento de una pieza  $P_{\pi_{k-1}}$  en la máquina  $M_i$ , en el instante  $S_{i,k-1}$ , y el fin de procesamiento de la pieza siguiente  $P_{\pi_k}$  en la misma máquina, en el instante  $S_{i,k}$ . Se forma mediante los tiempos de operaciones siguientes:

- ▶ una secuencia de operaciones del robot que descargan la pieza  $P_{\pi_{k-1}}$  de la máquina  $M_i$ ,
- ▶ una secuencia de operaciones del robot que cargan la pieza  $P_{\pi_k}$  en la máquina  $M_i$ ,
- ▶ y el procesamiento de la pieza  $P_{\pi_k}$  en la máquina  $M_i$ , quedando el robot, al final de todas las operaciones, situado en la máquina  $M_i$ .

El análisis se divide en tres casos:

**Caso en que, en el estado en curso,  $P_{\pi_{k-1}}$  ya ha pasado por  $M_i$ .** Si al menos una de las dos piezas  $P_{\pi_{k-1}}$  o  $P_{\pi_k}$  ya ha pasado por  $M_i$ , el valor de  $L_{i,k}$  es necesariamente nulo. Dado que  $P_{\pi_{k-1}}$  es la pieza más adelantada, basta comprobar si dicha pieza ya ha pasado por  $M_i$  (con independencia de si  $P_{\pi_k}$  ha pasado o no por  $M_i$ ), entonces  $L_{i,k}$  debe tomar valor cero. Para este caso, la condición se expresa como:

$$\begin{aligned} L_{i,k} &= 0 \\ \text{si } l_{\pi_{k-1}} &> 2i \end{aligned} \quad (3.14)$$

La asignación de valor nulo a  $L_{i,k}$  se debe a que no es conocido el instante real de salida  $S_{i,k-1}$  de la pieza  $P_{\pi_{k-1}}$  de la máquina  $M_i$  (pues ya ha pasado por  $M_i$ ) y los instantes de salida pasados (anteriores al estado en curso) no son guardados. Por ello carece de sentido determinar un valor para  $L_{i,k}$  pues de todos modos, no será posible por este camino hallar  $S_{i,k}$ , debido a que el valor de  $S_{i,k-1}$  es desconocido.

De todos modos, si la pieza  $P_{\pi_k}$  no ha pasado por la máquina  $M_i$ , todavía será posible hallar un valor de  $S_{i,k}$  mediante el cálculo de los arcos tipo  $T_{i,k}$  como se explica más adelante, en el apartado 3.2.4.

**Caso en que, en el estado en curso,  $P_{\pi_k}$  está en el buffer  $B_{i-1}$  anterior a  $M_i$  o en la estación  $I$ , anterior a  $M_1$ .** Se consideran las dos opciones para la descarga de  $P_{\pi_{k-1}}$  (pieza saliente de  $M_i$ ). Estas opciones son: en el buffer siguiente  $B_i$  y en la máquina siguiente  $M_{i+1}$ .

- ▶ La descarga en el buffer siguiente ocupa los tiempos:

$$\theta_d(i, \pi_{k-1}, 1) + t_{2i+1, 2i-1, 0}^v \quad (3.15)$$

- ▶ La descarga en la máquina siguiente utiliza los tiempos:

$$\theta_d(i, \pi_{k-1}, 2) + t_{2i+2, 2i-1, 0}^v \quad (3.16)$$

El tiempo para descargar  $P_{\pi_{k-1}}$  resulta de tomar el mínimo dado por de las expresiones 3.15 y 3.16:

$$\min_{q \in \{1, 2\} | h_{i,q}^+ = 1} (\theta_d(i, \pi_{k-1}, q) + t_{2i+q, 2i-1, 0}^v) \quad (3.17)$$

### Capítulo 3: Cota inferior basada en los tiempos de las máquinas cuando la secuencia de piezas es dada

Como en el estado en curso la pieza  $P_{\pi_k}$  (pieza ingresante a  $M_i$ ) se encuentra en el buffer anterior a  $M_i$ , hay una sola estrategia para el ingreso de  $P_{\pi_k}$  en  $M_i$ . Ésta ocupa un tiempo:

$$\theta_c(i, \pi_k, 1) \quad (3.18)$$

A lo anterior se debe sumar el tiempo de procesamiento:

$$t_{i, \pi_k}^p \quad (3.19)$$

Por último, mediante la suma de los términos 3.17 a 3.19 se define  $L_{i,k}$  como:

$$L_{i,k} = \min_{q \in \{1,2\} | h_{i,q}^+ = 1} (\theta_d(i, \pi_{k-1}, q) + t_{2i+q, 2i-1, 0}^v) + \theta_c(i, \pi_k, 1) + t_{i, \pi_k}^p \quad (3.20)$$

si  $l_{\pi_k} = 2i - 1$

La figura 3.3 muestra las operaciones del robot y los tiempos de la expresión 3.20.

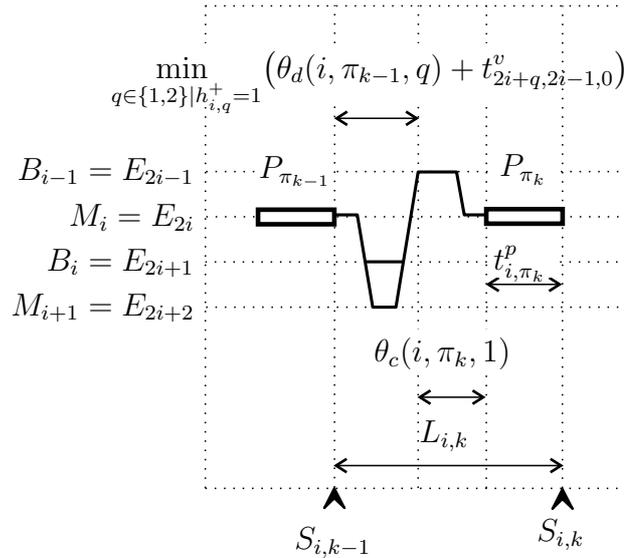


Figura 3.3: Diagrama de tiempos de la cota  $L_{i,k}$  correspondiente al caso en que, en el estado en curso,  $P_{\pi_k}$  está en el buffer  $B_{i-1}$  anterior a  $M_i$ .

**Caso en que, en el estado en curso,  $P_{\pi_k}$  está ocupando alguna de las estaciones anteriores a  $B_{i-1}$ .** Corresponde al caso en que, en el estado en curso,  $l_{\pi_k} < 2i - 1$ . Bajo esta condición deben ser consideradas como posibles las dos estrategias para la salida de  $P_{\pi_{k-1}}$  de  $M_i$  y las dos estrategias para el ingreso de  $P_{\pi_k}$  en  $M_i$ . De su combinación resultan cuatro estrategias de las cuales se tomará el menor de los tiempos mediante la cota de descarga seguida de carga (apartado 2.6.4) según la expresión:

$$\delta_{i, \pi_{k-1}, \pi_k}^e \quad (3.21)$$

Por último se debe sumar el tiempo de procesamiento:

$$t_{i,\pi_k}^p \quad (3.22)$$

La expresión final para el caso en que  $l_{\pi_k} < 2i - 1$  resulta de la suma de 3.21 y 3.22:

$$L_{i,k} = \delta_{i,\pi_{k-1},\pi_k}^e + t_{i,\pi_k}^p \quad (3.23)$$

si  $l_{\pi_k} < 2i - 1$

La figura 3.4 muestra operaciones del robot y los tiempos de la expresión 3.23.

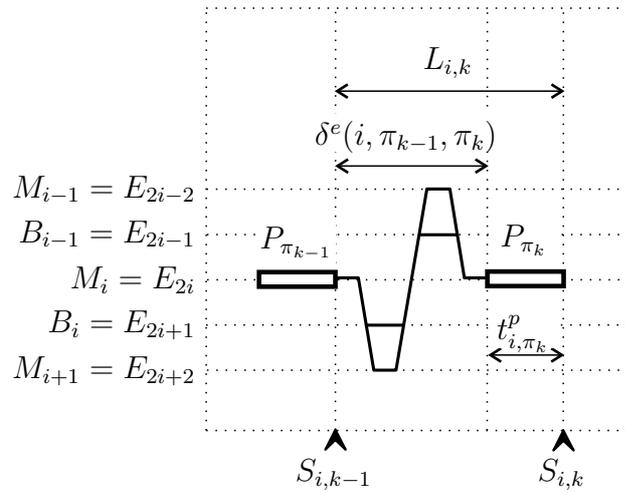


Figura 3.4: Diagrama de tiempos de la cota  $L_{i,k}$  correspondiente al caso en que, en el estado en curso,  $P_{\pi_k}$  está ocupando alguna de las estaciones anteriores a  $B_{i-1}$ .

**Expresión completa de  $L_{i,k}$ .** Los resultados de los casos expresados por las ecuaciones 3.14, 3.20, y 3.23, se reúnen en la siguiente expresión única:

$$L_{i,k} \begin{cases} 0 & \text{si } l_{\pi_{k-1}} > 2i \\ \min_{q \in \{1,2\} | h_{i,q}^+ = 1} (\theta_d(i, \pi_{k-1}, q) + t_{2i+q, 2i-1, 0}^v) + \theta_c(i, \pi_k, 1) + t_{i,\pi_k}^p & \text{si } l_{\pi_k} = 2i - 1 \\ \delta_{i,\pi_{k-1},\pi_k}^e + t_{i,\pi_k}^p & \text{si } l_{\pi_k} < 2i - 1 \end{cases} \quad (3.24)$$

### 3.2.4 Arcos tipo $T_{i,k}$

Los arcos tipo  $T_{i,k}$  representan el valor de la cota inferior asociada al tiempo transcurrido entre el instante de terminación del procesamiento de una pieza  $P_{\pi_k}$  en la máquina  $M_{i-1}$ , dado por el instante  $S_{i-1,k}$ , y el fin de procesamiento de la misma pieza en la máquina siguiente, dado por el instante  $S_{i,k}$ .

**Caso en que, en el estado en curso,  $P_{\pi_k}$  está ocupando alguna de las estaciones siguientes a  $M_{i-1}$ .** En este caso se asigna valor cero al arco  $T_{i,k}$ , pues las operaciones del robot necesarias para llevar  $P_{\pi_k}$  a  $M_i$  ya se han hecho y sus tiempos han sido incluidos en el arco tipo  $I_k$ .

$$T_{i,k} = 0 \quad \text{si } l_{\pi_k} > 2i - 2 \quad (3.25)$$

**Caso en que, en el estado en curso,  $P_{\pi_k}$  está ocupando  $M_{i-1}$  o alguna de las estaciones anteriores a  $M_{i-1}$ .** Se forma mediante los tiempos de las operaciones siguientes:

- Una secuencia de operaciones del robot que extraen  $P_{\pi_k}$  de  $M_{i-1}$  y producen su ingreso en  $M_i$ , cuyo tiempo se determina por:

$$\theta_c(i, \pi_k, 2) \quad (3.26)$$

- El procesamiento de  $P_{\pi_k}$  en  $M_i$ , quedando el robot situado en la máquina  $M_i$ , cuyo tiempo se expresa por:

$$t_{i,\pi_k}^p \quad (3.27)$$

De la suma de 3.26 y 3.27 se define la expresión que determina el valor de  $T_{i,k}$ :

$$T_{i,k} = \theta_c(i, \pi_k, 2) + t_{i,\pi_k}^p \quad \text{si } l_{\pi_k} \leq 2i - 2 \quad (3.28)$$

La figura 3.5 muestra operaciones del robot y los tiempos de la expresión 3.28.

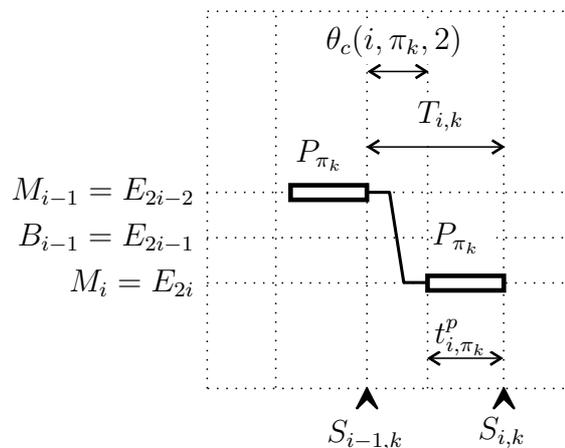


Figura 3.5: Diagrama de tiempos de la cota  $T_{i,k}$ .

No se considera la opción de paso por el buffer intermedio dado que, como se ha indicado en el apartado 2.2.3, se asume el cumplimiento de la propiedad triangular sobre

los tiempos de transporte de toda pieza  $P_j$  entre todo par de estaciones, expresada en la ecuación 2.2. Para este análisis la ecuación 2.2 puede escribirse de la forma:

$$t_{2i-2,2i,j}^v \leq t_{2i-2,2i-1,j}^v + t_{2i-1,2i,j}^v \quad (3.29)$$

si se suma  $t_{2i-2,j}^d + t_{2i,j}^c$  a ambos miembros de la desigualdad 3.29 se obtiene:

$$t_{2i-2,j}^d + t_{2i-2,2i,j}^v + t_{2i,j}^c \leq t_{2i-2,j}^d + t_{2i-2,2i-1,j}^v + t_{2i-1,2i,j}^v + t_{2i,j}^c \quad (3.30)$$

Si se suma  $t_{2i-1,j}^c + t_{2i-1,j}^d$  al miembro de la derecha de la desigualdad 3.30 y se reasocian los sumandos, dado que ambos sumandos son cantidades no negativas, el sentido de la desigualdad se mantiene (aunque el miembro de la derecha puede incrementarse). Luego de este paso se obtiene:

$$t_{2i-2,j}^d + t_{2i-2,2i,j}^v + t_{2i,j}^c \leq (t_{2i-2,j}^d + t_{2i-2,2i-1,j}^v + t_{2i-1,j}^c) + (t_{2i-1,j}^d + t_{2i-1,2i,j}^v + t_{2i,j}^c) \quad (3.31)$$

Así, la expresión 3.31 equivale a:

$$\theta_c(i, j, 2) \leq \theta_d(i-1, j, 1) + \theta_c(i, j, 1) \quad (3.32)$$

De aquí que la opción de paso por el buffer (término del lado derecho de la desigualdad 3.32) no mejora el valor expresado en la ecuación 3.26 (igual al termino del lado izquierdo de la desigualdad 3.32). Por tanto se desecha esta estrategia en el cálculo de la cota  $T_{i,k}$ .

**Expresión completa de  $T_{i,k}$ .** Los resultados de los casos expresados por las ecuaciones 3.25 y 3.28 se reúnen en la expresión 3.33.

$$T_{i,k} = \begin{cases} 0 & \text{si } l_{\pi_k} > 2i - 2 \\ \theta_c(i, \pi_k, 2) + t_{i,\pi_k}^p & \text{si } l_{\pi_k} \leq 2i - 2 \end{cases} \quad (3.33)$$

### 3.2.5 Arco tipo $U$

Consiste en el tiempo empleado en las operaciones que producen la extracción en la última máquina  $M_m$  de la pieza secuenciada en la última posición  $n$  y su ingreso en la estación de salida.

$$U = \theta_d(m, \pi_n, 1) \quad (3.34)$$

### 3.2.6 Procedimiento de acotación

Consiste en la aplicación del proceso iterativo determinado por las ecuaciones 3.5 a 3.9 del apartado 3.2, reescritas a continuación:

$$S_{1,1} = I_1 \quad (3.5)$$

$$S_{1,k} = \max (I_k, S_{1,k-1} + L_{1,k}) \quad k = 2, \dots, n \quad (3.6)$$

$$S_{i,1} = S_{i-1,1} + T_{i,1} \quad i = 2, \dots, m \quad (3.7)$$

$$S_{i,k} = \max (S_{i-1,k} + T_{i,k}, S_{i,k-1} + L_{i,k}) \quad i = 2, \dots, m, \quad k = 2, \dots, n \quad (3.8)$$

$$S = S_{m,n} + U \quad (3.9)$$

Donde:

$$I_k = \begin{cases} \rho & \text{si } l_{\pi_k} = 2m + 1 \\ \max (a_i, \rho + t_{r,2i,0}^v) & \text{si } l_{\pi_k} < 2m + 1 \text{ y } l_{\pi_k} = 2i \\ \rho + t_{r,2i-1,0}^v + \theta_c(i, \pi_k, 1) + t_{i,\pi_k}^p & \text{si } l_{\pi_k} < 2m + 1 \text{ y } l_{\pi_k} = 2i - 1 \end{cases} \quad (3.13)$$

$$L_{i,k} = \begin{cases} 0 & \text{si } l_{\pi_{k-1}} > 2i \\ \min_{q \in \{1,2\} | h_{i,q}^+ = 1} (\theta_d(i, \pi_{k-1}, q) + t_{2i+q,2i-1,0}^v) + \theta_c(i, \pi_k, 1) + t_{i,\pi_k}^p & \text{si } l_{\pi_k} = 2i - 1 \\ \delta_{i,\pi_{k-1},\pi_k}^e + t_{i,\pi_k}^p & \text{si } l_{\pi_k} < 2i - 1 \end{cases} \quad (3.24)$$

$$T_{i,k} = \begin{cases} 0 & \text{si } l_{\pi_k} > 2i - 2 \\ \theta_c(i, \pi_k, 2) + t_{i,\pi_k}^p & \text{si } l_{\pi_k} \leq 2i - 2 \end{cases} \quad (3.33)$$

$$U = \theta_d(m, \pi_n, 1) \quad (3.34)$$

# Capítulo 4

## Cota inferior basada en los tiempos de las máquinas cuando la secuencia de piezas no es dada

### 4.1 Introducción

En este capítulo se presenta un procedimiento de determinación de una cota inferior para el caso general, donde la secuencia de piezas no está totalmente preestablecida. En este caso la cota basada en los tiempos de las máquinas se compone de dos cotas:

- ▶ una cota para las piezas que, en un estado del sistema dado, no ocupan la estación de entrada  $I$ , y
- ▶ una cota para las piezas que, en el mismo estado, están ocupando la estación de entrada  $I$ .

En el capítulo 3 se estableció un procedimiento de acotación basado en un grafo (figura 3.2) y el conjunto de ecuaciones 3.5 a 3.9 que derivan de él, para el caso donde la secuencia de piezas es totalmente conocida de antemano. Este enfoque será nuevamente utilizado en este capítulo para definir una cota de los tiempos para el conjunto de piezas que no ocupan la estación de entrada  $I$ , dado que la secuencia es conocida para estas piezas, puesto que es definida por el orden con el que salen de la estación  $I$ . En el apartado 4.2 se mostrará como adaptar las anteriores ecuaciones 3.5 a 3.9 para satisfacer las condiciones de este planteo.

Por otra parte, en el apartado 4.3, para las piezas que en el estado en curso todavía están alojadas en la estación de entrada se plantea una cota basada en la cota de Ignall y Schrage (Ignall y Schrage (1965)) definida originalmente para flowshops donde son despreciados los tiempos de transporte, carga y descarga de las piezas en las distintas estaciones.

## **4.2 Cota para las piezas que en un estado dado no ocupan la estación de entrada**

Las ecuaciones 3.5 a 3.9 introducidas en el capítulo 3, en aquel caso planteadas para cuando la secuencia de piezas  $\bar{\pi}$  (determinada para cada estado) debe respetar una secuencia de piezas predefinida  $\bar{\Pi}$  de longitud  $n$ , deben ajustarse al caso en que la secuencia de piezas  $\bar{\pi}$  se va determinado gradualmente, y, por ello, en cada estado su longitud  $u$  cumple  $u \leq n$ .

El número de piezas que en un estado dado  $e_w$  no ocupan la estación de entrada,  $u$ , puede determinarse por el número de piezas que han ingresado en  $M_1$  (dado que el número de piezas que han salido de la estación de entrada  $I$  forzosamente han tenido que ingresar, por ejecución del mismo movimiento, en  $M_1$ ). De aquí se cumple que:

$$u = z_1^w \quad (4.1)$$

donde, recuérdese del capítulo 2,  $z_1^w$  es el número de piezas que en el estado  $e_w$  ya han ingresado en la máquina  $M_1$  (ecuaciones 2.15 y 2.16).

Las ecuaciones 3.5 a 3.9, reescritas en función de la longitud  $u$  de la secuencia  $\bar{\pi}$  se convierten en:

$$S_{1,1} = I_1 \quad (4.2)$$

$$S_{1,k} = \max (I_k, S_{1,k-1} + L_{1,k}) \quad k = 2, \dots, u \quad (4.3)$$

$$S_{i,1} = S_{i-1,1} + T_{i,1} \quad i = 2, \dots, m \quad (4.4)$$

$$S_{i,k} = \max (S_{i-1,k} + T_{i,k}, S_{i,k-1} + L_{i,k}) \quad i = 2, \dots, m, \quad k = 2, \dots, u \quad (4.5)$$

donde, como se observa, no se presenta un expresión equivalente a la anterior 3.9 por ser innecesaria en este análisis.

## **4.3 Cota para las piezas que en un estado dado ocupan la estación de entrada**

La cota para las piezas que ocupan la estación de entrada  $I$  se inspira en la cota de Ignall y Schrage (Ignall y Schrage (1965)) formulada originalmente para problemas de secuenciación de tipo flowshop con tiempos sólo de procesamiento. Sea  $\tau$  el conjunto de piezas que ocupan la estación de entrada  $I$ , por tanto, conformado por toda pieza  $P_j$  tal que  $l_j = 1$ . Formalmente, se define como:

$$P_j \in \tau \text{ si y solo si } l_j = 1$$

## **Cota para las piezas que en un estado dado ocupan la estación de entrada**

El siguiente análisis se basará en dos supuestos:

- ▶ se supondrá que la pieza  $P_g$  es la primera pieza del conjunto de piezas en  $\tau$  en salir de la estación  $I$ ,
- ▶ y que la pieza  $P_h$  es la última pieza del conjunto de piezas en  $\tau$  en salir de la estación  $I$ .

Estos supuestos valdrán para el caso en que  $\text{card}(\tau) > 1$ . En el caso en que  $\text{card}(\tau) = 1$  se supondrá que la única pieza alojada en la estación  $I$  es  $P_g$ .

Similarmente a la formulación original de la cota de Ignall y Schrage (Ignall y Schrage (1965)), y en base a los supuestos indicados, la cota  $S$  basada en los tiempos de las máquinas cuando la secuencia de piezas no es dada, se forma de los siguientes tiempos:

- ▶ Un tiempo  $\lambda_{i,g}^1$  que representa el instante de inicio de procesamiento de las piezas de  $\tau$  en  $M_i$ .
- ▶ Un tiempo  $\lambda_i^2$  que representa los tiempos de las máquinas y del robot asociados al procesamiento de las piezas de  $\tau$  en  $M_i$ .
- ▶ Un tiempo  $\lambda_{i,h}^3$  que representa los tiempos de las máquinas y del robot asociados al tiempo de procesamiento de la última pieza  $P_h$  en las máquinas siguientes a  $M_i$ .

### **4.3.1 Los tiempos de $\lambda_{i,g}^1$**

**Instante de inicio de procesamiento de  $\tau$  basado en los tiempos en  $M_i$ .** Sea  $P_{\pi_u}$  la pieza secuenciada en último lugar del conjunto de piezas que no están en la estación de entrada. Por definición, en el instante  $S_{i,u}$ ,  $P_{\pi_u}$  está alojada en  $M_i$  y el robot en la misma estación. Bajo esta condición, el inicio de procesamiento de  $\tau$  es la suma de siguientes tiempos:

- ▶ una cota del instante de finalización del procesamiento de  $P_{\pi_u}$  en  $M_i$ :  $S_{i,u}$ ,
- ▶ una cota del tiempo necesario para descargar  $P_{\pi_u}$  de  $M_i$  y seguidamente cargar  $P_g$ :  $\delta_{i,\pi_u,g}^e$ .

En resumen, la suma:

$$S_{i,u} + \delta_{i,\pi_u,g}^e \tag{4.6}$$

Estos tiempos pueden verse en la gráfica inferior de la figura 4.1.

**Instante de inicio de procesamiento de  $\tau$  basado en los tiempos de  $P_g$ .** El supuesto del punto anterior por el cual la pieza  $P_g$  está disponible no es válido siempre, pues, en el estado en curso  $P_g$  está en  $I$ , y por tanto serán necesarios tiempos en las máquinas y en el robot asociados al procesamiento en las máquinas anteriores a la máquina  $M_i$  hasta que  $P_g$  esté disponible para ingresar en  $M_i$ .

Consiste en la suma:

- ▶ El tiempo del robot del estado en curso:  $\rho$ ;
- ▶ El tiempo de carga de  $P_g$  en  $M_1$ ,  $\theta_c(1, g, 1)$ ;
- ▶ El tiempo que comprende los movimientos y el procesamiento de  $P_g$  en todas las máquinas anteriores a  $M_i$ : para toda máquina  $M_k$  desde la primera hasta la máquina anterior a  $M_i$ , la suma de:
  - ▷ el tiempo de procesamiento  $t_{k,g}^p$ ,
  - ▷ y el tiempo  $\theta_d(k, g, 2)$ , necesario para depositar  $P_g$  en la máquina siguiente a  $M_k$ .

Estos tiempos se suman en la expresión:

$$\rho + \theta_c(1, g, 1) + \sum_{k=1}^{i-1} (t_{k,g}^p + \theta_d(k, g, 2)) \quad (4.7)$$

Estos tiempos pueden verse en el diagrama de tiempos superior de la figura 4.1.

**Instante de inicio de procesamiento de las piezas de  $\tau$ .** Se expresa como el máximo de los tiempos dados por las ecuaciones 4.6 y 4.7:

$$\lambda_{i,g}^1 = \max \left( S_{i,u} + \delta_{i,\pi_u,g}^e, \rho + \theta_c(1, g, 1) + \sum_{k=1}^{i-1} (t_{k,g}^p + \theta_d(k, g, 2)) \right) \quad (4.8)$$

válida para el caso en que  $1 < i \leq m$ . Mientras que si  $i = 1$ , la expresión 4.8 se reduce a:

$$\lambda_{i,g}^1 = S_{i,u} + \delta_{i,\pi_u,g}^e \quad (4.9)$$

Las expresiones 4.8 y 4.9 se agrupan en 4.10.

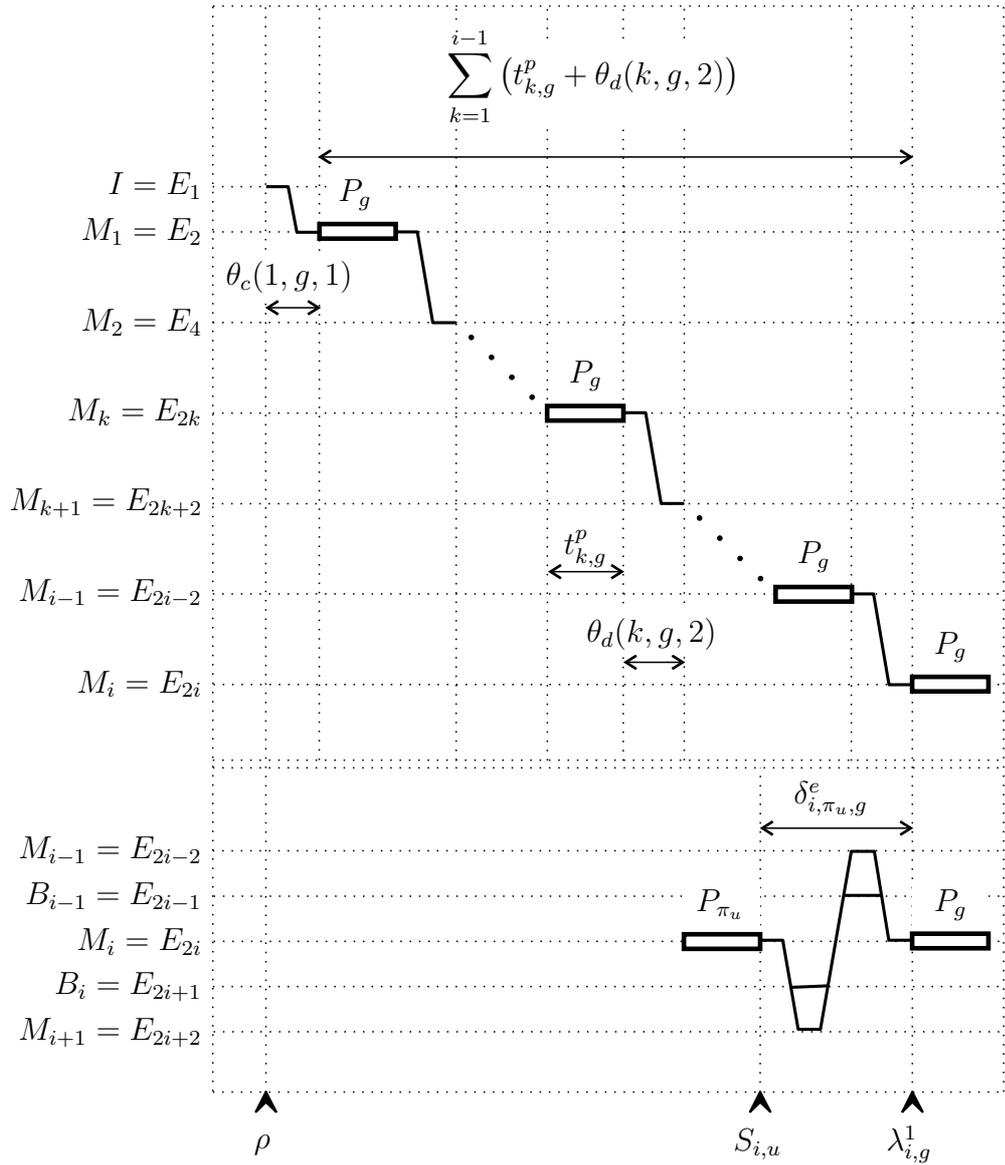


Figura 4.1: Tiempos asociados a la cota  $\lambda_{i,g}^1$ .

$$\lambda_{i,g}^1 = \begin{cases} \max \left( S_{i,u} + \delta_{i,\pi_u,g}^e, \rho + \theta_c(1,g,1) + \sum_{k=1}^{i-1} (t_{k,g}^p + \theta_d(k,g,2)) \right) & \text{si } 1 < i \leq m \\ S_{i,u} + \delta_{i,\pi_u,g}^e & \text{si } i = 1 \end{cases} \quad (4.10)$$

### 4.3.2 Los tiempos de $\lambda_i^2$

La cota  $\lambda_i^2$  consiste en una cota del tiempo asociado al procesamiento del conjunto de piezas de  $\tau$  en  $M_i$ . Se forma por los tiempos de procesamiento de las piezas y los tiempos del robot necesarios para el ingreso y egreso de las piezas en  $M_i$  entre cada inicio y fin de procesamiento. Un esquema de estos tiempos puede verse en la figura 4.2.

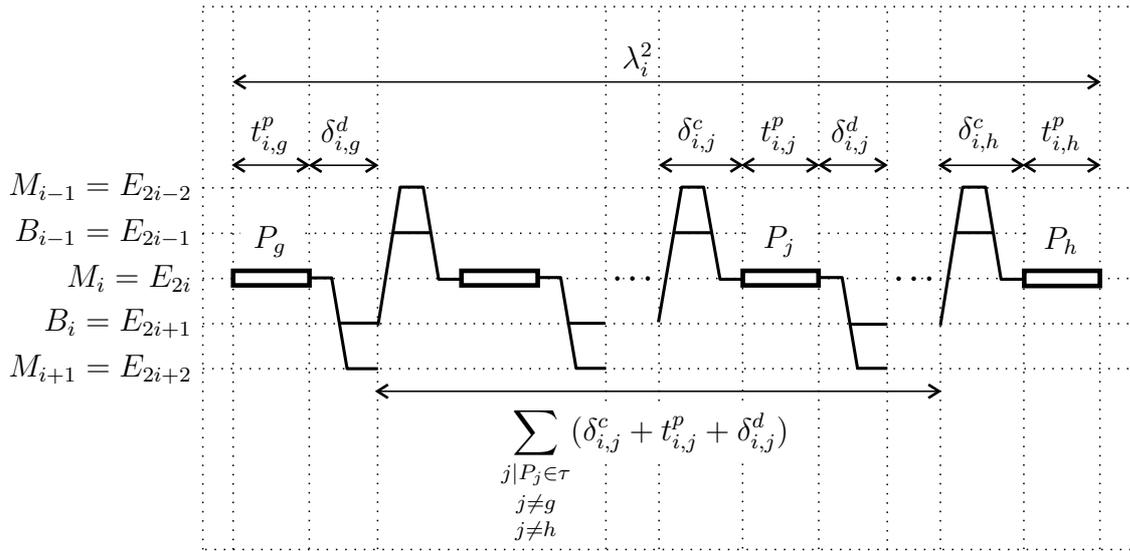


Figura 4.2: Tiempos asociados a la cota  $\lambda_i^2$ .

La cota  $\lambda_i^2$  se define como en la suma de los términos siguientes:

- ▶ el tiempo de procesamiento de  $P_g$  en  $M_i$ :  $t_{i,g}^p$ ;
- ▶ una cota del tiempo de descarga de  $P_g$  de  $M_i$ :  $\delta_{i,g}^d$ ;
- ▶ para toda pieza  $P_j$  en  $\tau$  con excepción de  $P_g$  y  $P_h$ , la suma de:
  - ▷ cota de carga<sup>1</sup> de  $P_j$  en  $M_i$ :  $\delta_{i,j}^c$ ,
  - ▷ el tiempo de procesamiento de  $P_j$  en  $M_i$ :  $t_{i,j}^p$ ,
  - ▷ y cota de descarga de  $P_j$  de  $M_i$ :  $\delta_{i,j}^d$ .
- ▶ una cota del tiempo de carga de  $P_h$  en  $M_i$ :  $\delta_{i,h}^c$ ;
- ▶ el tiempo de procesamiento de  $P_h$  en  $M_i$ :  $t_{i,h}^p$ .

Estos términos, en el caso en que  $\tau$  se compone de al menos dos piezas, es decir que se cumple la condición  $\text{card}(\tau) > 1$ , dan lugar a la siguiente expresión:

$$\lambda_i^2 = t_{i,g}^p + \delta_{i,g}^d + \sum_{\substack{j|P_j \in \tau \\ j \neq g \\ j \neq h}} (\delta_{i,j}^c + t_{i,j}^p + \delta_{i,j}^d) + \delta_{i,h}^c + t_{i,h}^p \quad (4.11)$$

mientras que en el caso en que  $\tau$  se compone de una sola pieza  $P_g$ , expresado en la condición  $\text{card}(\tau) = 1$ , la expresión 4.11 se reduce a:

$$\lambda_i^2 = t_{i,g}^p \quad (4.12)$$

<sup>1</sup>Aquí se pone de manifiesto la necesidad de suponer al robot inicialmente situado en la siguiente estación válida a la máquina  $M_i$ , adoptado en la definición de la cota de carga  $\delta_{i,j}^c$  (apartado 2.6.2).

## Cota para las piezas que en un estado dado ocupan la estación de entrada

Las expresiones 4.11 y 4.12 pueden agruparse en:

$$\lambda_i^2 = \begin{cases} t_{i,g}^p + \delta_{i,g}^d + \sum_{\substack{j|P_j \in \tau \\ j \neq g \\ j \neq h}} (\delta_{i,j}^c + t_{i,j}^p + \delta_{i,j}^d) + \delta_{i,h}^c + t_{i,h}^p & \text{si } \text{card}(\tau) > 1 \\ t_{i,g}^p & \text{si } \text{card}(\tau) = 1 \end{cases} \quad (4.13)$$

### 4.3.3 Los tiempos de $\lambda_{i,h}^3$

Consiste en la suma de tiempos asociados al procesamiento de la última pieza  $P_h$  de  $\tau$  en todas las máquinas siguientes a  $M_i$ . Se forma con los términos siguientes:

- ▶ para toda máquina  $M_k$  desde  $M_{i+1}$  hasta  $M_m$ , la suma de:
  - ▷ tiempo de carga de  $P_h$  en  $M_k$ :  $\theta_c(k, h, 2)$ ,
  - ▷ el tiempo de procesamiento de  $P_h$  en  $M_k$ ,  $t_{k,h}^p$ ,
- ▶ el tiempo de carga de  $P_h$  en  $O$ :  $\theta_d(m, h, 1)$ .

En el caso en que  $1 \leq i < m$ , la suma de los términos indicados deriva en la expresión:

$$\lambda_{i,h}^3 = \sum_{k=i+1}^m (\theta_c(k, h, 2) + t_{k,h}^p) + \theta_d(m, h, 1) \quad (4.14)$$

Mientras que para el caso en que  $i = m$  es válida la siguiente ecuación:

$$\lambda_{i,h}^3 = \theta_d(m, h, 1) \quad (4.15)$$

Las expresiones 4.14 y 4.15 pueden agruparse para dar lugar a una definición única de  $\lambda_{i,h}^3$ :

$$\lambda_{i,h}^3 = \begin{cases} \sum_{k=i+1}^m (\theta_c(k, h, 2) + t_{k,h}^p) + \theta_d(m, h, 1) & \text{si } 1 \leq i < m \\ \theta_d(m, h, 1) & \text{si } i = m \end{cases} \quad (4.16)$$

Los tiempos de  $\lambda_h^3$  pueden verse en la figura 4.3.

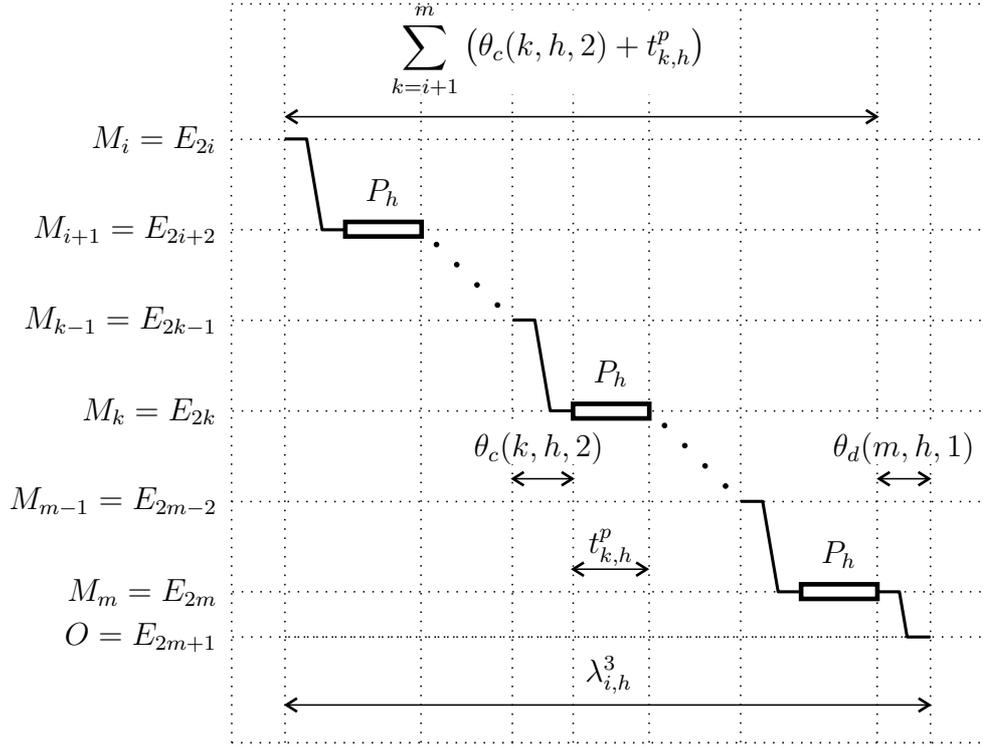


Figura 4.3: Tiempos asociados a la cota  $\lambda_{i,h}^3$ .

### 4.3.3.1 Ecuaciones finales

Para el caso  $\text{card}(\tau) > 1$  la cota del instante de terminación  $S$  se define:

$$S = \max_{i=1}^m \min_{\substack{g|P_g \in \tau \\ h|P_h \in \tau \\ g \neq h}} (\lambda_{i,g}^1 + \lambda_i^2 + \lambda_{i,h}^3) \quad (4.17)$$

Mientras que en el caso  $\text{card}(\tau) = 1$  se emplea:

$$S = \max_{i=1}^m (\lambda_{i,g}^1 + \lambda_i^2 + \lambda_{i,g}^3) \quad (4.18)$$

De agrupar 4.17 y 4.18 se obtiene una expresión única de  $S$ :

$$S = \begin{cases} \max_{i=1}^m \min_{\substack{g|P_g \in \tau \\ h|P_h \in \tau \\ g \neq h}} (\lambda_{i,g}^1 + \lambda_i^2 + \lambda_{i,h}^3) & \text{si } \text{card}(\tau) > 1 \\ \max_{i=1}^m (\lambda_{i,g}^1 + \lambda_i^2 + \lambda_{i,g}^3) & \text{si } \text{card}(\tau) = 1 \end{cases} \quad (4.19)$$

## **4.4 Procedimiento**

Para un estado dado, la determinación de la cota inferior basada en los tiempos de las máquinas cuando la secuencia de piezas no es dada,  $S$ , se resume en los siguientes pasos.

- ▶ Se determina el número de piezas que, en el estado dado, no ocupan la estación de entrada, mediante la expresión 4.1.
- ▶ Se determina, para cada máquina, la cota de los tiempos de salida del conjunto de piezas que no ocupan la estación de entrada  $S_{i,u}$  mediante las expresiones 4.2 a 4.5 (apartado 4.2).
- ▶ Se determinan los tiempos:  $\lambda_{i,g}^1$  (ecuación 4.10),  $\lambda_i^2$  (ecuación 4.13) y  $\lambda_{i,h}^3$  (ecuación 4.16). De éstos tiempos,  $\lambda_{i,g}^1$  es función de la cota  $S_{i,u}$  hallada en el paso anterior.
- ▶ En función de los tiempos:  $\lambda_{i,g}^1$ ,  $\lambda_i^2$  y  $\lambda_{i,h}^3$  mediante la expresión 4.19, se determina la cota basada en los tiempos en las máquinas cuando la secuencia de piezas no es dato del problema,  $S$ .



# Capítulo 5

## Cota inferior basada en los tiempos del robot

### 5.1 Introducción

Como en las cotas basadas en los tiempos de procesamiento en las máquinas (capítulos 3 y 4) la *cota basada en los tiempos del robot* es también el resultado de la relajación de algunas condiciones del problema original.

### 5.2 Definición del problema relajado

En este caso, procede de suponer igual a cero los tiempos de procesamiento de todas las piezas en todas las máquinas. Es decir, en este nuevo problema se asume la relajación de los tiempos de proceso siguiente:

$$t_{i,j}^p = 0 \quad i = 1, \dots, m; \quad j = 1, \dots, n$$

frente al problema original donde los tiempos de proceso cumplen:

$$t_{i,j}^p \geq 0 \quad i = 1, \dots, m; \quad j = 1, \dots, n$$

#### 5.2.1 Condiciones de factibilidad de una secuencia de movimientos del robot

Toda secuencia de movimientos del robot factible debe (con independencia del orden con que estos movimientos sean realizados) asegurar el cumplimiento de las siguientes operaciones:

- ▶ la salida de cada pieza  $P_j$  de la estación de entrada  $I$  donde se encuentra inicialmente,

- ▶ cada pieza  $P_j$  en cada máquina  $M_i$ , debe:
  - ▷ ser transportada allí,
  - ▷ ser cargada,
  - ▷ ser procesada (en un tiempo  $t_{i,j}^p$  nulo, por relajación del problema),
  - ▷ y ser descargada;
- ▶ cada pieza  $P_j$  debe ser llevada a la estación de salida  $O$  y cargada allí;
- ▶ para todo par de piezas  $P_j$  y  $P_k$  de la secuencia de piezas, es obligatorio que en algún momento, bien de forma directa (mediante un movimiento) o por tramos (mediante varios movimientos), un desplazamiento del robot sin pieza desde la estación de salida  $O$  hasta la estación de entrada  $I$ .

Como fue definido en el apartado 2.2, el patrón de flujo adoptado por el problema (ver página 28) impone que las piezas deben ser procesadas en orden creciente de los índices de las máquinas (según la secuencia  $M_1, M_2, \dots, M_m$ ). Esta condición impide el ingreso en una máquina  $M_i$  de una pieza  $P_j$ , si ésta antes no ha salido de la máquina anterior  $M_{i-1}$ .

Con el agregado de esta condición, el conjunto de movimientos (con independencia del orden con que estos movimientos se realicen) que deben formar parte de una secuencia de movimientos del robot factible puede definirse de este otro modo.

- ▶ Cada pieza  $P_j$  debe salir de la estación de entrada  $I$  (donde se encuentra inicialmente), transportada a la máquina  $M_1$ , cargada allí y procesada (en un tiempo nulo por condición del problema relajado). La realización óptima de este movimiento consiste en:
  - ▷ descargar  $P_j$  de la estación de entrada  $I$ ,
  - ▷ viajar con  $P_j$  desde la estación de entrada  $I$  hasta la máquina  $M_1$  (mediante el camino mas corto, sin desvíos ni detenciones),
  - ▷ cargar  $P_j$  en la máquina  $M_1$ ,
  - ▷ y el procesamiento de la pieza  $P_j$  en la máquina  $M_1$  (en tiempo cero por condición del problema relajado).
- ▶ Cada pieza  $P_j$  debe ser descargada de cada máquina  $M_i$  y llevada a la máquina siguiente  $M_{i+1}$  donde debe ser cargada y procesada (en un tiempo nulo por condición del problema relajado). La realización óptima de este movimiento consiste en:
  - ▷ descargar la pieza  $P_j$  en la máquina  $M_i$ ,
  - ▷ viajar desde la máquina  $M_i$  a la máquina siguiente  $M_{i+1}$  (mediante el camino mas corto, sin desvíos ni detenciones),
  - ▷ cargar la pieza  $P_j$  en la máquina  $M_{i+1}$ ,
  - ▷ y el procesamiento de la pieza  $P_j$  en la máquina  $M_{i+1}$  (en tiempo cero por condición del problema relajado).

## Definición del problema relajado

---

- ▶ Cada pieza  $P_j$  debe ser descargada de la última máquina  $M_m$ , transportada a la estación de salida  $O$  y cargada allí. La realización óptima de este movimiento consiste en:
  - ▷ descargar la pieza  $P_j$  de la máquina  $M_m$ ,
  - ▷ viajar con la pieza  $P_j$  desde la máquina  $M_m$  hasta la estación de salida  $O$  (empleando el camino mas corto, sin desvíos ni detenciones),
  - ▷ y cargar la pieza  $P_j$  en la estación de salida  $O$ .
- ▶ Para todo par de piezas  $P_j$  y  $P_k$  de la secuencia de piezas, es obligatorio que en algún momento, ya sea en forma directa (mediante un movimiento) o por tramos (mediante varios movimientos), se realice un desplazamiento del robot sin pieza desde la estación de salida  $O$  hasta la estación de entrada  $I$ . La realización óptima de este movimiento consiste en el viaje del robot sin pieza desde la estación de salida  $O$  hasta la estación de entrada  $I$  (empleando el camino mas corto, sin desvíos ni detenciones).

Hasta ahora, los distintos movimientos que garantizan que una secuencia sea factible han sido analizados de forma aislada y han demostrado ser óptimos cuando son tomados fuera de una secuencia de movimientos del robot concreta.

Sin embargo, para dar origen a una secuencia concreta será necesario introducir desplazamientos del robot a fin de conectar los distintos movimientos. Éstos añaden nuevos tiempos y la secuencia puede, aún cuando está conformada por movimientos óptimos, ya no ser óptima.

A continuación se propone una secuencia de movimientos en la que se unen los distintos movimientos óptimos indicados previamente y, dado que se cumplen las condiciones de factibilidad anteriores, será factible, aunque de momento no puede asegurarse que sea óptima. Dicha secuencia consiste en los siguientes movimientos:

Para cada pieza  $P_j$  ( $j = 1, \dots, n$ ):

- ▶ descargar  $P_j$  de la estación de entrada  $I$ , viajar con  $P_j$  desde la estación de entrada  $I$  hasta la máquina  $M_1$  (mediante el camino mas corto, sin desvíos ni detenciones), cargar  $P_j$  en la máquina  $M_1$ , y el procesamiento de la pieza  $P_j$  en la máquina  $M_1$  (en tiempo cero por condición del problema relajado);
- ▶ en cada máquina  $M_i$  ( $i = 1, \dots, m - 1$ ): descargar la pieza  $P_j$  de la máquina  $M_i$ , viajar desde la máquina  $M_i$  a la máquina siguiente  $M_{i+1}$  (mediante el camino mas corto, sin desvíos ni detenciones), cargar la pieza  $P_j$  en la máquina  $M_{i+1}$ , y el procesamiento de la pieza  $P_j$  en la máquina  $M_{i+1}$  (en tiempo cero por condición del problema relajado);
- ▶ descargar la pieza  $P_j$  de la máquina  $M_m$ , viajar con la pieza  $P_j$  desde la máquina  $M_m$  hasta la estación de salida  $O$  (empleando el camino mas corto, sin desvíos ni detenciones), cargar la pieza  $P_j$  en la estación de salida  $O$ ;
- ▶ con excepción de la pieza  $P_{\pi_n}$ , viajar sin pieza desde la estación de salida  $O$  hasta la estación de entrada  $I$  (empleando el camino mas corto, sin desvíos ni detenciones).

Como puede observarse, en la secuencia de movimientos del robot anterior basada en la ejecución de movimientos individualmente óptimos, no ha sido necesaria la introducción de movimientos del robot adicionales para conectar cada movimiento con el siguiente. Es decir, al final de cada movimiento, el robot está ocupando la misma estación que al inicio del movimiento siguiente. Por ello, dicha secuencia, además de factible, es también óptima.

### 5.3 Secuencia piramidal

Bajo el supuesto de que los tiempos de procesamiento de las piezas en las máquinas son todos igual a cero, la solución óptima de un problema de 3 piezas y tres máquinas, consiste en una secuencia de movimientos del robot como la que se muestra en la figura 5.1.

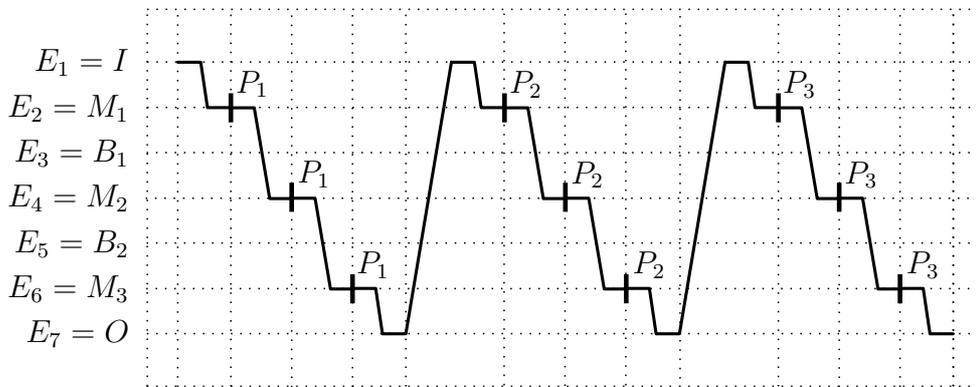


Figura 5.1: Secuencia de movimientos del robot óptima cuando los tiempos de procesamiento de las piezas en las máquinas son todos igual a cero en un problema de 3 piezas y 3 máquinas.

En este capítulo se introduce la definición de secuencia de movimientos del robot piramidal, por similitud con los tours piramidales (definidos en Lawler *et al.* (1985)) conocidos en el contexto del problema del viajante de comercio. En este caso, el robot cumple la función del viajante y las máquinas hacen las veces de ciudades.

Una secuencia de movimientos del robot piramidal asociada a una pieza  $P_j$  consiste en las siguientes operaciones del robot.

- ▶ Desplazamiento del robot sin pieza desde la estación de salida  $O$  hasta la estación  $E_k$  donde está alojada la pieza  $P_j$ ;
- ▶ Sea  $M_g$  la máquina inmediata siguiente a la estación  $E_k$ :
  - ▷ descargar la pieza  $P_j$  de la estación  $E_k$ ,
  - ▷ transportar la pieza  $P_j$  desde la estación  $E_k$  hasta a la máquina  $M_g$ ,
  - ▷ cargar la pieza  $P_j$  en la máquina  $M_g$ ,
  - ▷ y procesar la pieza  $P_j$  en la máquina  $M_g$  en un tiempo nulo por hipótesis del problema relajado,

## **Procedimiento de cálculo de la cota inferior basada en los tiempos del robot**

- ▶ En el caso en que  $g < m$ , para toda máquina  $M_i$  ( $i = g, \dots, m - 1$ )
  - ▷ descargar la pieza  $P_j$  de la máquina  $M_i$ ,
  - ▷ transportar la pieza  $P_j$  desde la máquina  $M_i$  hasta la máquina siguiente  $M_{i+1}$ ,
  - ▷ cargar la pieza  $P_j$  en la máquina  $M_{i+1}$ ,
  - ▷ y procesar la pieza  $P_j$  en la máquina  $M_{i+1}$  en un tiempo nulo por hipótesis del problema relajado.
- ▶ En la última máquina  $M_m$ :
  - ▷ descargar la pieza  $P_j$  de la máquina  $M_m$ ,
  - ▷ transportar la pieza  $P_j$  desde la máquina  $M_m$  hasta la estación de salida  $O$
  - ▷ y cargar la pieza  $P_j$  en la estación de salida  $O$ .

### **5.4 Procedimiento de cálculo de la cota inferior basada en los tiempos del robot**

El procedimiento de cálculo de la cota inferior basada en los tiempos del robot se resume en los siguientes pasos:

- ▶ Para toda pieza que no está en la última estación: determinar el tiempo de la secuencia de movimientos piramidal suponiendo, en todos los casos, que al inicio de ésta el robot está situado en la estación de salida  $O$ . Para cada secuencia de movimientos piramidal se presentan dos casos según la pieza procesada en dicha secuencia esté ocupando una máquina (apartado 5.5.1) o un buffer (apartado 5.5.2).
- ▶ Para el caso de la primera secuencia piramidal, el supuesto de que al inicio de ésta el robot está situado en la estación de salida  $O$  no siempre se cumple. En concreto el robot ocupa la estación  $r$ , pudiendo sólo eventualmente coincidir con la estación de salida  $O$ . Este supuesto es corregido mediante una cifra de corrección (apartado 5.6).

### **5.5 Cota asociada a las piezas basada en rutas piramidales**

Para toda pieza  $P_j$  se halla una cota basada en la realización de una ruta piramidal. Se distinguen dos casos según  $P_j$  esté, en un caso alojada en una máquina, o, en otro caso, esté alojada en un buffer o en la estación de entrada.

#### **5.5.1 Caso de pieza alojada en una máquina.**

Toda pieza  $P_j$  alojada en una máquina verifica la condición  $\text{resmed}(l_j) = 0$ . Sea  $M_{\Psi_j}$  la máquina que aloja a  $P_j$ , dado que la estación donde está la pieza  $l_j$  y la estación correspondiente a la máquina  $E_{2\Psi_j}$  coinciden, se cumple:

$$\Psi_j = \frac{l_j}{2} \quad (5.1)$$

Para este caso, la ruta piramidal mostrada en la figura 5.2 consiste en las siguientes operaciones:

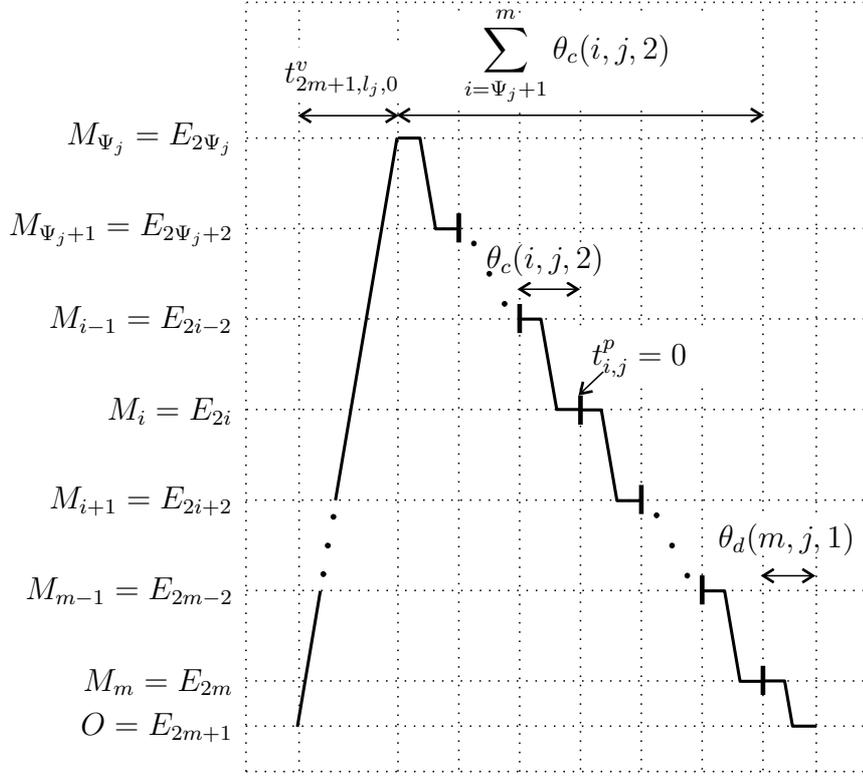


Figura 5.2: Operaciones y tiempos del robot de la ruta piramidal asociada a una pieza  $P_j$  alojada en la máquina  $M_{\Psi_j}$ .

- Desplazamiento del robot sin pieza desde la última estación hasta la estación ocupada por la pieza, empleando el tiempo:

$$t_{2m+1,l_j,0}^v \quad (5.2)$$

- Sólo en el caso en que  $\Psi_j < m$ , para toda máquina desde  $M_{\Psi_j}$  hasta la última máquina  $M_m$  la pieza será descargada y cargada en la máquina siguiente, cuyo tiempo se expresa en la suma:

$$\sum_{i=\Psi_j+1}^m \theta_c(i,j,2) \quad (5.3)$$

- Para la última máquina  $M_m$  la pieza será descargada y cargada en la estación de salida:

$$\theta_d(m,j,1) \quad (5.4)$$

### 5.5.2 Caso de pieza alojada en un buffer o en la estación de entrada.

Toda pieza  $P_j$  alojada en un buffer o en la estación de entrada verifica la condición  $\text{resmed}(l_j) = 1$ . Sea  $M_{\Psi_j}$  la máquina siguiente al buffer  $B_{\Psi_j-1}$  o estación  $I$ , donde se aloja  $P_j$ . Dado que la estación de la pieza  $l_j$  y la estación del buffer  $E_{2\Psi_j-1}$  (o estación  $I$ ) coinciden, se cumple:

$$\Psi_j = \frac{l_j + 1}{2} \quad (5.5)$$

Para este caso, la ruta piramidal mostrada en la figura 5.3 consiste en las siguientes operaciones:

- ▶ Desplazamiento del robot sin pieza desde la última estación hasta la estación ocupada por la pieza empleando el tiempo:

$$t_{2m+1, l_j, 0}^v \quad (5.6)$$

- ▶  $P_j$  debe ser extraída de  $B_{\Psi_j-1}$  (o en la estación  $I$ ) e ingresada en  $M_{\Psi_j}$  en un tiempo:

$$\theta_c(\Psi_j, j, 1) \quad (5.7)$$

- ▶ Sólo en el caso en que  $\Psi_j < m$ , para toda máquina desde  $M_{\Psi_j}$  hasta la última máquina  $M_m$  la pieza será descargada y cargada en la máquina siguiente, cuyo tiempo se expresa en la suma:

$$\sum_{i=\Psi_j+1}^m \theta_c(i, j, 2) \quad (5.8)$$

- ▶ Para la última máquina  $M_m$  la pieza será descargada y cargada en la estación de salida:

$$\theta_d(m, j, 1) \quad (5.9)$$

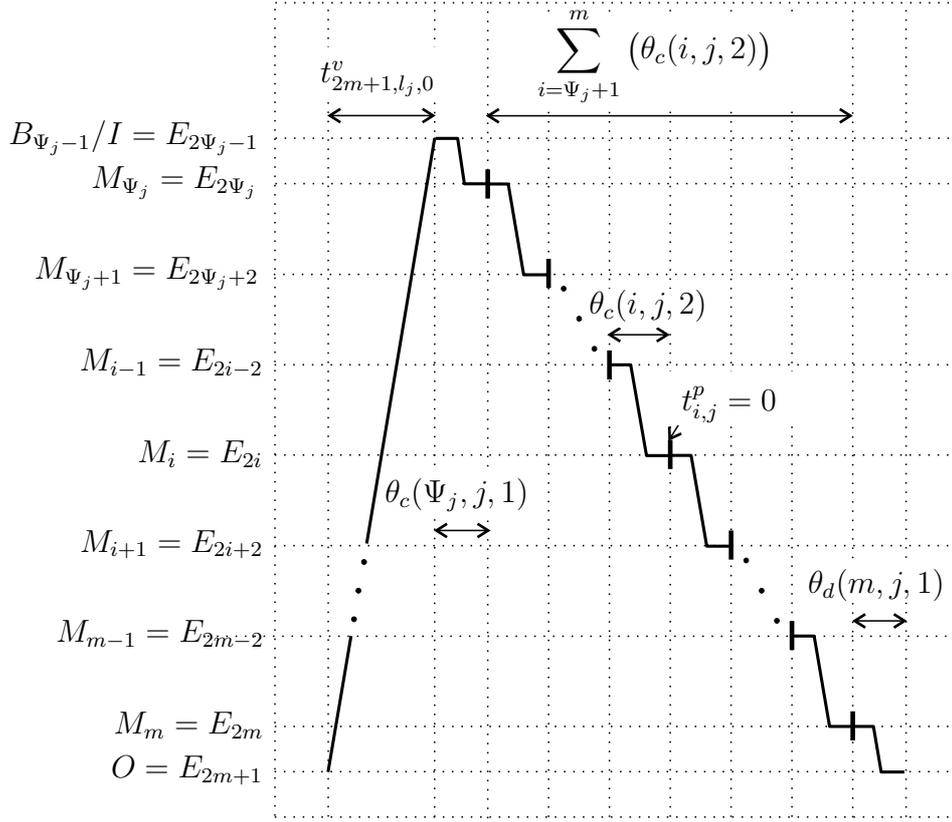


Figura 5.3: Operaciones y tiempos del robot de la ruta piramidal asociada a una pieza  $P_j$  alojada en el buffer  $B_{\Psi_j-1}$  o en la estación  $I$ , anterior a  $M_{\Psi_j}$ .

## 5.6 Cifra de corrección

El supuesto de que al inicio y al final de cada secuencia de movimientos piramidal el robot está situado en la última estación es válido para todas las rutas con excepción de la primera, donde el robot ocupa la estación  $r$ . Para esta secuencia el supuesto debe ser corregido sumando al valor de la cota un término denominado cifra de corrección  $\omega$ .

Sea  $g$  el número de piezas en la estación de salida, éste se determina por la expresión:

$$g = \max_{\substack{1 \leq k \leq n \\ l_{\pi_k} = 2m+1}} k \quad (5.10)$$

El valor  $g = 0$  indica que no hay piezas en la estación  $O$ . Un valor  $g > 0$  indica que hay  $g$  piezas en la estación  $O$ , desde la pieza  $P_{\pi_1}$  hasta la pieza  $P_{\pi_g}$ .

Es necesario distinguir entre dos casos, que se analizarán a continuación.

**Caso en que hay al menos una pieza alojada en estaciones fuera de  $I$  y  $O$ .** Corresponde al caso en que el número de piezas que en un estado  $e_w$  han salido de  $I$

(dado por  $z_1^w$ ) es mayor que el número de piezas que están ocupando la estación de salida  $O$  (dado por el valor de  $g$ ). En este caso se verifica la presencia de piezas en las estaciones que no son estación de entrada  $I$  y la estación de salida  $O$ , es decir en las máquinas o en los buffers. El número de piezas en estas estaciones es la diferencia  $z_1^w - g$ .

Dado que hay  $g$  piezas en la estación  $O$  (desde la pieza  $P_{\pi_1}$  hasta la pieza  $P_{\pi_g}$ ), la pieza correspondiente a la secuencia piramidal que se hará en primer lugar es la pieza  $P_{\pi_{g+1}}$ .

En este caso la cifra de corrección, que estará asociada a la pieza  $P_{\pi_{g+1}}$ , es:

$$\begin{aligned} \omega &= t_{r,l_{\pi_{g+1}},0}^v - t_{2m+1,l_{\pi_{g+1}},0}^v \\ &\text{si } g < z_1^w < n \end{aligned} \quad (5.11)$$

**Caso en que no hay piezas alojadas en estaciones fuera de  $I$  y  $O$ .** Corresponde al caso en que el número de piezas que en un estado  $e_w$  han salido de  $I$  (dado por  $z_1^w$ ) es igual al número de piezas que están ocupando la estación de salida  $O$  (dado por el valor de  $g$ ). Es el caso en que todas las piezas que han abandonado la estación de entrada  $I$ , han terminado depositadas en la estación de salida  $O$ . Es el caso en que no hay piezas situadas en las máquinas y en los buffers.

Dado que todas las piezas aún no procesadas están en la estación de entrada  $I$ , la cifra de corrección es la misma para cualquiera de las piezas que ocupan la estación de entrada  $I$ . El valor de la cifra de corrección  $\omega$  para este caso es:

$$\begin{aligned} \omega &= t_{r,1,0}^v - t_{2m+1,1,0}^v \\ &\text{si } g = z_1^w < n \end{aligned} \quad (5.12)$$

**Expresión completa de la cifra de corrección.** Las expresiones 5.11 y 5.12 se pueden agrupar dentro de una expresión única:

$$\omega = \begin{cases} t_{r,1,0}^v - t_{2m+1,1,0}^v & \text{si } g = z_1^w \\ t_{r,l_{\pi_{g+1}},0}^v - t_{2m+1,l_{\pi_{g+1}},0}^v & \text{en otro caso} \end{cases} \quad (5.13)$$

$g < n$

## 5.7 Expresión final de la cota

Las ecuaciones de los apartados 5.5 y 5.6 determinan las expresiones finales para el cálculo de la cota basada en los tiempos del robot  $\Lambda$ , que se determina en tres fases:

**Determinación de los tiempos  $\Lambda_j$  de las rutas piramidales asociados a cada pieza  $P_j$ .** Para toda pieza  $P_j$  se determina el tiempo  $\Lambda_j$  de la ruta piramidal correspondiente mediante las expresiones 5.14, 5.15 y 5.16.

$$\Psi_j = \begin{cases} l_j/2, & \text{si } \text{resmed}(l_j) = 0 \\ (l_j + 1)/2, & \text{si } \text{resmed}(l_j) = 1 \end{cases} \quad (5.14)$$

$$\psi_j = \begin{cases} 0, & \text{si } \text{resmed}(l_j) = 0 \\ \theta_c(\Psi_j, j, 1), & \text{si } \text{resmed}(l_j) = 1 \end{cases} \quad (5.15)$$

$$\Lambda_j = \begin{cases} 0, & \text{si } l_j = 2m + 1 \\ t_{2m+1, 2m, 0}^v + \theta_d(m, j, 1), & \text{si } l_j = 2m \\ t_{2m+1, l_j, 0}^v + \psi_j + \sum_{i=\Psi_j+1}^m (\theta_c(i, j, 2)) + \theta_d(m, j, 1), & \text{si } l_j < 2m \end{cases} \quad (5.16)$$

Donde la expresión 5.14 procede de 5.1 y 5.5. La expresión 5.15 procede de 5.7. Finalmente, en la ecuación 5.16 se agrupan las expresiones 5.2, 5.3, 5.4, 5.6, 5.8 y 5.9.

**Determinación de la cifra de corrección  $\omega$ .** Se determina:

$$g = \max_{\substack{1 \leq k \leq n \\ l_{\pi_k} = 2m+1}} k \quad (5.10)$$

y a continuación se halla:

$$\omega = \begin{cases} t_{r, 1, 0}^v - t_{2m+1, 1, 0}^v & \text{si } g = z_1^w \\ t_{r, l_{\pi_{g+1}}, 0}^v - t_{2m+1, l_{\pi_{g+1}}, 0}^v & \text{en otro caso} \end{cases} \quad (5.13)$$

$g < n$

**Determinación de la cota basada en los tiempos del robot  $\Lambda$ .** La cota basada en los tiempos del robot  $\Lambda$  consiste de la suma del instante de disponibilidad del robot  $\rho$ , de los tiempos de cada una de las rutas piramidales  $\Lambda_j$  y de la cifra de corrección  $\omega$ .

$$\Lambda = \rho + \sum_{j=1}^n \Lambda_j + \omega \quad (5.17)$$

# Capítulo 6

## Modelo lineal para la determinación de una secuencia de piezas

### 6.1 Introducción

En este capítulo se presenta un modelo de programación lineal entera mixta diseñado para la resolución de una variante simplificada del problema definido en el capítulo 2. Como resultado de la resolución de dicho modelo se obtiene una secuencia de piezas de la forma:

$$\bar{\Phi} = \Phi_1, \Phi_2, \dots, \Phi_n \quad (6.1)$$

a partir de la cual, mediante procedimientos descritos en el capítulo 7, se determinan distintas secuencias de movimientos del robot. Estos procedimientos son definidos como enfoques basados en la separación de la secuencia de piezas y la secuencia de movimientos del robot.

El modelo lineal que se presenta en este capítulo, formaliza una variante simplificada del problema a resolver en el que se relajan algunas de las restricciones de tipo disyuntivas que están presentes cuando se supone la existencia de un robot único, por tanto, es equivalente a suponer la existencia de un robot al servicio de cada máquina.

### 6.2 Variables del modelo

Las variables (incógnitas) del modelo son las matrices de tiempos de terminación  $\mathbb{S}$ , y la matriz de asignación  $\mathbb{X}$ .

#### 6.2.1 Instante posible de inicio de salida de las piezas

La matriz de *instantes posibles de inicio de salida de las piezas*,  $\mathbb{S}$ , se define como:

$$\mathbb{S} : s_{i,k} \quad (s_{i,k} \in \mathbb{R} ; i = 1, \dots, m ; k = 1, \dots, n)$$

El elemento  $s_{i,k}$  representa el instante en que la pieza  $P_{\Phi_k}$  (pieza secuenciada en la posición  $k$ ) puede iniciar su salida de la máquina  $M_i$ .

### 6.2.2 Matriz de asignación

La asignación entre las piezas del conjunto de piezas a procesar y las posiciones en la secuencia de procesamiento de las piezas, está representada mediante la matriz de asignación  $\mathbb{X}$ . Dicha matriz está formada por elementos de tipo binario  $x_{j,k}$  cuyo valor es 1 cuando la pieza  $P_j$  está signada en la posición  $k$  de la secuencia de procesamiento de las piezas, y 0 en el caso contrario.

$$\mathbb{X} : x_{j,k} \quad (x_{j,k} \in \{0, 1\} ; j = 1, \dots, n ; k = 1, \dots, n)$$

## 6.3 Restricciones del modelo

Parte esencial del diseño de un modelo de programación lineal que satisfaga los requerimientos de un problema dado, consiste en definir el conjunto de restricciones mediante el cual se expresen las diferentes condiciones que deben cumplirse en dicho problema. A continuación se presentan las distintas restricciones que definen el modelo de programación lineal entera mixta propuesto en este capítulo.

### 6.3.1 Asignación de una pieza única a cada posición de la secuencia de piezas

A cada posición en el orden de procesamiento de las piezas le corresponde una y sólo una pieza. Dicha condición se expresa mediante la restricción 6.2, donde, para toda posición  $k$ , se impone que la suma de las variables de asignación asociada a toda pieza  $P_j$  debe ser igual a uno.

$$\sum_{j=1}^n x_{j,k} = 1 \quad (1 \leq k \leq n) \quad (6.2)$$

### 6.3.2 Asignación de una posición única de la secuencia de piezas a cada pieza

Toda pieza del conjunto de piezas tiene asociada una posición única en el orden de procesamiento de las piezas. Dicha condición se expresa en la ecuación 6.3, mediante la imposición, para toda pieza  $P_j$ , de que la suma de las variables de asignación asociada a toda posición  $k$  sea igual a uno.

$$\sum_{k=1}^n x_{j,k} = 1 \quad (1 \leq j \leq n) \quad (6.3)$$

### 6.3.3 Instante posible de inicio de salida de la primera pieza de la primera máquina

El instante posible de inicio de salida de la primera máquina, de la pieza que ocupa la primera posición en la secuencia de procesamiento  $s_{1,1}$ , no puede ser inferior a la suma del tiempo de procesamiento y de los tiempos de las operaciones del robot (figura 6.1).

Esta restricción se expresa, suponiendo conocida la secuencia de piezas  $\bar{\Phi}$ :

$$s_{1,1} \geq \theta_c(1, \Phi_1, 1) + t_{1,\Phi_1}^p \quad (6.4)$$

Puesto que la secuencia de piezas  $\bar{\Phi}$  no es conocida, la misma restricción debe expresarse en función de las variables de asignación  $x_{j,1}$ :

$$s_{1,1} \geq \sum_{j=1}^n (\theta_c(1, j, 1) + t_{1,j}^p) x_{j,1} \quad (6.4.a)$$

Agrupando las variables en el primer miembro y las constantes en el segundo miembro:

$$\sum_{j=1}^n (\theta_c(1, j, 1) + t_{1,j}^p) x_{j,1} - s_{1,1} \leq 0 \quad (6.4.b)$$

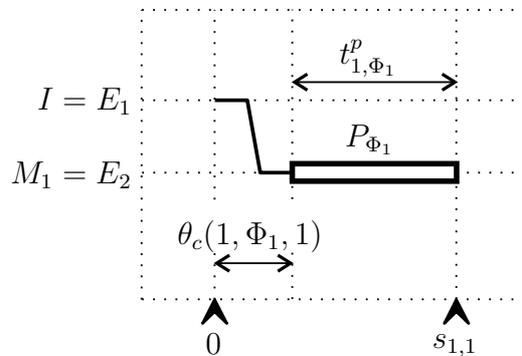


Figura 6.1: Instante posible de inicio de salida de la primera pieza de la primera máquina.

Nótese que las restricciones 6.4 a 6.4.b admiten también el sentido estrictamente igual (en lugar de menor o igual, como se ha expresado) dado que el robot no tiene más operaciones que realizar.

### 6.3.4 Precedencia entre piezas

El seguimiento de una determinada secuencia de procesamiento de las piezas, impone, en toda máquina  $M_i$ , que no puede iniciarse el procesamiento de la pieza que ocupa la posición  $k$  hasta que la pieza que ocupa la posición  $k - 1$  no ha terminado de procesarse en la misma máquina (figura 6.2).

En efecto, a partir del instante  $s_{i,k-1}$  en el que la pieza  $P_{\Phi_{k-1}}$  ha terminado de procesarse en la máquina  $M_i$  debe producirse:

- ▶ la descarga de la pieza  $P_{\Phi_{k-1}}$  de  $M_i$  y su carga en la estación  $E_{2i+2-h_{i,1}^+}$ , es decir,
  - ▷ en la estación  $E_{2i+1}$  (buffer  $B_i$  o estación de salida  $O$ ), en el caso en que la capacidad de dicha estación resultase no nula ( $h_{i,1}^+ = 1$ );
  - ▷ o en la estación estación  $E_{2i+2}$  (máquina  $M_{i+1}$ ), en el caso en que la capacidad de dicha estación resultase nula, ( $h_{i,1}^+ = 0$ );
 donde es empleado el tiempo  $\theta_d(i, \Phi_{k-1}, 2 - h_{i,1}^+)$ ;
- ▶ el retroceso sin pieza del robot hasta la estación  $E_{2i-2+h_{i,1}^-}$  (donde se supone está alojada la pieza  $P_{\Phi_k}$ ), es decir,
  - ▷ a la estación estación  $E_{2i-1}$  (buffer  $B_{i-1}$  o estación de entrada  $I$ ), en el caso en que la capacidad de dicha estación resultase no nula ( $h_{i,1}^- = 1$ );
  - ▷ o a la estación estación  $E_{2i-2}$  (máquina  $M_{i-1}$ ), en el caso en que la capacidad de dicha estación resultase nula ( $h_{i,1}^- = 0$ );
 empleando el tiempo  $t_{2i+2-h_{i,1}^+, 2i-2+h_{i,1}^-, 0}^v$ ;
- ▶ carga de la pieza  $P_{\Phi_k}$  en  $M_i$  empleando el tiempo  $\theta_c(i, \Phi_k, 2 - h_{i,1}^-)$ ,
- ▶ y el proceso de la pieza  $P_{\Phi_k}$  en la máquina  $M_i$  empleando el tiempo  $t_{i, \Phi_k}^p$ ,

alcanzándose el instante  $s_{i,k}$ , al cabo de dichos tiempos.

Suponiéndose conocida la secuencia de piezas  $\bar{\Phi}$ , la precedencia entre piezas en una máquina dada se expresa mediante la restricción:

$$s_{i,k} \geq s_{i,k-1} + \theta_d(i, \Phi_{k-1}, 2 - h_{i,1}^+) + t_{2i+2-h_{i,1}^+, 2i-2+h_{i,1}^-, 0}^v + \theta_c(i, \Phi_k, 2 - h_{i,1}^-) + t_{i, \Phi_k}^p \quad (6.5)$$

$$1 \leq i \leq m ; 2 \leq k \leq n$$

Dado que la secuencia de piezas  $\bar{\Phi}$  no es conocida, se deben introducir las variables de asignación:

$$\begin{aligned}
 s_{i,k} &\geq s_{i,k-1} + \sum_{j=1}^n \theta_d(i, j, 2 - h_{i,1}^+) x_{j,k-1} + \\
 &+ t_{2i+2-h_{i,1}^+, 2i-2+h_{i,1}^-, 0}^v + \sum_{j=1}^n (\theta_c(i, j, 2 - h_{i,1}^-) + t_{i,j}^p) x_{j,k} \\
 &1 \leq i \leq m ; 2 \leq k \leq n
 \end{aligned} \tag{6.5.a}$$

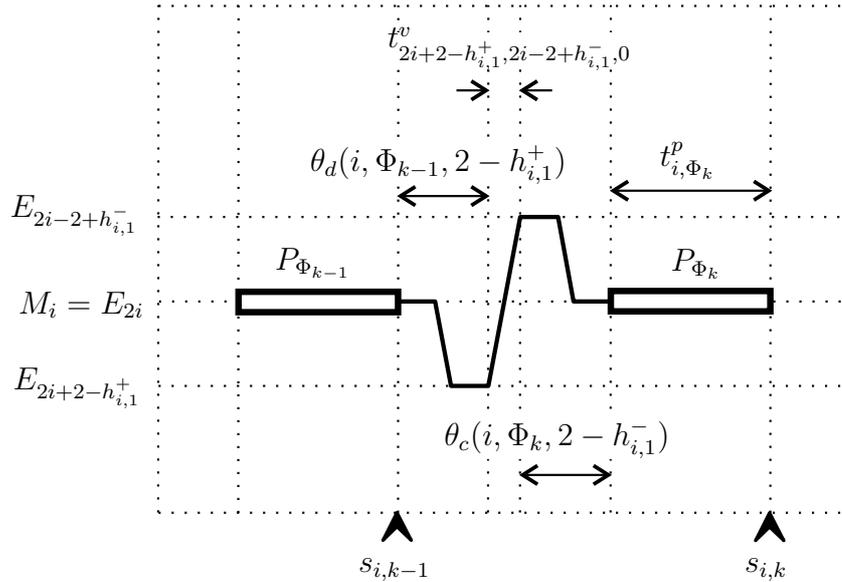


Figura 6.2: Precedencia entre piezas en toda máquina  $M_i$ .

Agrupando las variables en el primer miembro y las constantes en el segundo miembro:

$$\begin{aligned}
 s_{i,k-1} - s_{i,k} + \sum_{j=1}^n \theta_d(i, j, 2 - h_{i,1}^+) x_{j,k-1} + \\
 + \sum_{j=1}^n (\theta_c(i, j, 2 - h_{i,1}^-) + t_{i,j}^p) x_{j,k} \leq -t_{2i+2-h_{i,1}^+, 2i-2+h_{i,1}^-, 0}^v \\
 1 \leq i \leq m ; 2 \leq k \leq n
 \end{aligned} \tag{6.5.b}$$

### 6.3.5 Precedencia entre máquinas

La secuencia de procesos a ser ejecutada sobre toda pieza responde a una ruta de tipo flowshop según la secuencia de máquinas  $M_1, M_2, \dots, M_m$  (punto 2.2, página 28). El

## Capítulo 6: Modelo lineal para la determinación de una secuencia de piezas

cumplimiento de esta secuencia de operaciones queda asegurada mediante la imposición de que ninguna de las piezas puede comenzar su procesamiento en una máquina  $M_i$ , si no ha terminado de procesarse en la máquina anterior  $M_{i-1}$  (figura 6.3).

La ecuación 6.6 expresa esta condición en términos de los instantes de salida de dos máquinas consecutivas, teniendo en cuenta que entre estos instantes deben existir obligatoriamente, además del tiempo de procesamiento de la pieza, el tiempo del robot empleado en las operaciones para hacer la transferencia de la pieza de una máquina a la siguiente.

Suponiendo conocida la secuencia de piezas  $\bar{\Phi}$ , la precedencia entre máquinas de una pieza dada se expresa mediante la ecuación:

$$s_{i,k} \geq s_{i-1,k} + \theta_c(i, \Phi_k, 2) + t_{i,\Phi_k}^p \quad (6.6)$$

$$2 \leq i \leq m ; 1 \leq k \leq n$$

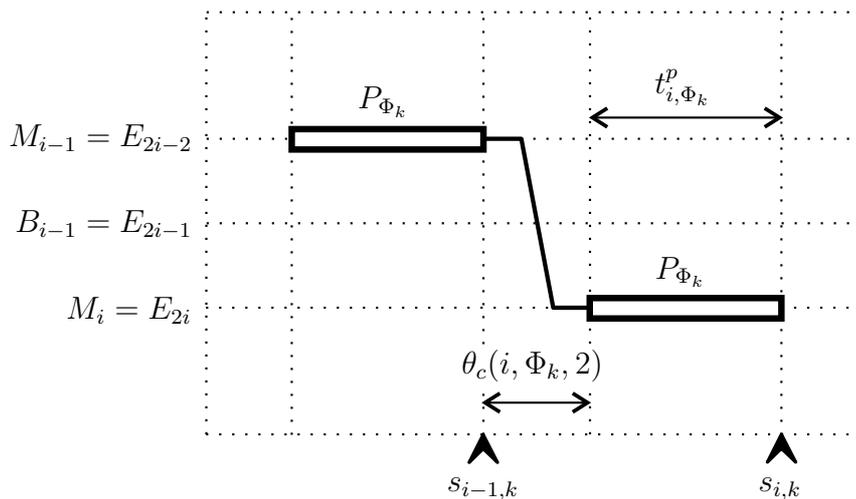


Figura 6.3: Precedencia entre máquinas para toda pieza  $P_j$ .

Dado que la secuencia de piezas  $\bar{\Phi}$  no es conocida, se deben introducir las variables de asignación:

$$s_{i,k} \geq s_{i-1,k} + \sum_{j=1}^n (\theta_c(i, j, 2) + t_{i,j}^p) x_{j,k} \quad (6.6.a)$$

$$2 \leq i \leq m ; 1 \leq k \leq n$$

Agrupando las variables en el primer miembro y las constantes en el segundo miembro:

$$s_{i-1,k} - s_{i,k} + \sum_{j=1}^n (\theta_c(i, j, 2) + t_{i,j}^p) x_{j,k} \leq 0 \quad (6.6.b)$$

$$2 \leq i \leq m ; 1 \leq k \leq n$$

### 6.3.6 Restricciones por el llenado de los buffers de capacidad no nula

En este apartado se analizan las restricciones que tienen lugar cuando se está en presencia de buffers de capacidad no nula (figura 6.4). Es decir, estas restricciones se asocian a aquellos buffers  $B_i$  que cumplen la condición:

$$0 < b_i < \infty$$

Cuando se produce el llenado de un cierto buffer  $B_i$  de capacidad  $b_i$ , una pieza  $P_{\Phi_k}$  dada no puede ingresar en dicho buffer hasta que no se haya producido espacio a partir de la salida de la pieza  $P_{\Phi_{k-b_i}}$  del mismo.

Por otro lado, la salida de la pieza  $P_{\Phi_{k-b_i}}$  del buffer  $B_i$  requiere que la máquina  $M_{i+1}$  esté sin pieza (para alojar la pieza  $P_{\Phi_{k-b_i}}$  saliente del buffer  $B_i$ ) y, para que esto sea posible, se requiere que, previamente, la pieza  $P_{\Phi_{k-b_i-1}}$  haya salido de la máquina  $M_{i+1}$ .

A partir del instante  $s_{i+1,k-b_i-1}$ , en que la pieza  $P_{\Phi_{k-b_i-1}}$  puede empezar a salir de la máquina  $M_{i+1}$ , debe producirse las operaciones del robot siguientes:

- ▶ la descarga de la pieza  $P_{\Phi_{k-b_i-1}}$  de  $M_{i+1}$  y su carga en la estación válida siguiente  $E_{2i+4-h_{i+1,1}^+}$ , es decir,
  - ▷ en la estación  $E_{2i+3}$  (buffer  $B_{i+1}$  o estación de salida  $O$ ), en el caso en que la capacidad de dicha estación resultase no nula ( $h_{i+1,1}^+ = 1$ );
  - ▷ o en la estación  $E_{2i+4}$  (máquina  $M_{i+2}$ ), en el caso en que la capacidad de dicha estación resultase nula, ( $h_{i+1,1}^+ = 0$ ),
 donde es empleado el tiempo  $\theta_d(i+1, \Phi_{k-b_i-1}, 2 - h_{i+1,1}^+)$ ;
- ▶ el desplazamiento del robot hasta el buffer  $B_i$  empleando el tiempo  $t_{2i+4-h_{i+1,1}^+, 2i+1, 0}^v$ ;
- ▶ la descarga de la pieza  $P_{\Phi_{k-b_i}}$  del buffer  $B_i$  y su carga en la máquina siguiente  $M_{i+1}$ , empleando el tiempo  $\theta_c(i+1, \Phi_{k-b_i}, 1)$ ;
- ▶ el desplazamiento del robot hasta la máquina  $M_i$  donde, en el instante  $s_{i,k}$ , puede iniciar la salida de la pieza  $P_{\Phi_k}$  con el fin de ser ingresada en el buffer  $B_i$ , empleando el tiempo  $t_{2i+2, 2i, 0}^v$ .

Suponiendo conocida la secuencia de piezas  $\bar{\Phi}$ , la precedencia entre máquinas de una pieza dada se expresa mediante la ecuación:

$$\begin{aligned}
 & s_{i+1,k-b_i-1} + \theta_d(i+1, \Phi_{k-b_i-1}, 2 - h_{i+1,1}^+) + \\
 & + t_{2i+4-h_{i+1,1}^+, 2i+1,0}^v + \theta_c(i+1, \Phi_{k-b_i}, 1) + t_{2i+2, 2i,0}^v \leq s_{i,k} \\
 & 1 \leq i \leq m-1; b_i + 2 \leq k \leq n
 \end{aligned} \tag{6.7}$$

Dado que la secuencia de piezas  $\bar{\Phi}$  no es conocida, se deben introducir las variables de asignación:

$$\begin{aligned}
 & s_{i+1,k-b_i-1} + \sum_{j=1}^n \theta_d(i+1, j, 2 - h_{i+1,1}^+) x_{j,k-b_i-1} + \\
 & + t_{2i+4-h_{i+1,1}^+, 2i+1,0}^v + \sum_{j=1}^n \theta_c(i+1, j, 1) x_{j,k-b_i} + t_{2i+2, 2i,0}^v \leq s_{i,k} \\
 & 1 \leq i \leq m-1; b_i + 2 \leq k \leq n
 \end{aligned} \tag{6.7.a}$$

Agrupando las variables en el primer miembro y las constantes en el segundo miembro:

$$\begin{aligned}
 & s_{i+1,k-b_i-1} - s_{i,k} + \sum_{j=1}^n \theta_d(i+1, j, 2 - h_{i+1,1}^+) x_{j,k-b_i-1} + \\
 & + \sum_{j=1}^n \theta_c(i+1, j, 1) x_{j,k-b_i} \leq -t_{2i+2, 2i,0}^v - t_{2i+4-h_{i+1,1}^+, 2i+1,0}^v \\
 & 1 \leq i \leq m-1; b_i + 2 \leq k \leq n
 \end{aligned} \tag{6.7.b}$$

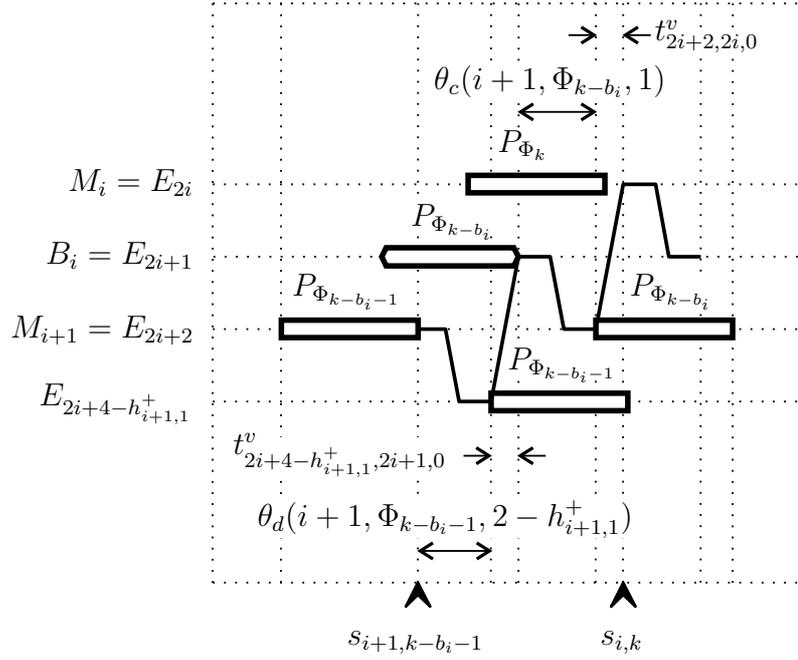


Figura 6.4: Precedencias por buffers de capacidad no nula.

### 6.3.7 Restricciones por los buffers de capacidad nula

En este apartado se analizan las restricciones que tienen lugar cuando se está en presencia de buffers de capacidad nula (figura 6.5). Es decir, las restricciones asociadas a aquellos buffers  $B_i$  que cumplen la condición:

$$b_i = 0$$

Cuando un buffer  $B_i$  es inválido ( $b_i = 0$ ), una pieza  $P_{\Phi_k}$  saliente de una máquina  $M_i$  está obligada a ingresar en la siguiente máquina  $M_{i+1}$ . Para que esto sea posible se requiere que, previamente, se haya producido espacio en dicha máquina a partir de la salida de la pieza  $P_{\Phi_{k-1}}$  de dicha máquina.

A partir del instante  $s_{i+1,k-1}$ , en que la pieza  $P_{\Phi_{k-1}}$  puede empezar a salir de la máquina  $M_{i+1}$ , deben producirse las operaciones del robot siguientes:

- la descarga de la pieza  $P_{\Phi_{k-1}}$  de  $M_{i+1}$  y su carga en la estación válida siguiente  $E_{2i+4-h_{i+1,1}^+}$ , es decir,
  - ▷ en la estación  $E_{2i+3}$  (buffer  $B_{i+1}$  o estación de salida  $O$ ), en el caso en que la capacidad de dicha estación resultase no nula ( $h_{i+1,1}^+ = 1$ );
  - ▷ o en la estación  $E_{2i+4}$  (máquina  $M_{i+2}$ ), en el caso en que la capacidad de dicha estación resultase nula ( $h_{i+1,1}^+ = 0$ ),
 donde es empleado el tiempo  $\theta_d(i+1, \Phi_{k-1}, 2-h_{i+1,1}^+)$ ;

## Capítulo 6: Modelo lineal para la determinación de una secuencia de piezas

- el desplazamiento del robot hasta la máquina  $M_i$  empleando el tiempo  $t_{2i+4-h_{i+1,1}^+, 2i, 0}^v$  alcanzando el instante  $s_{i,k}$ , donde se puede iniciar la salida de la pieza  $P_{\Phi_k}$  con el fin de ingresarla en la máquina  $M_{i+1}$ .

Suponiendo conocida la secuencia de piezas  $\bar{\Phi}$ , la precedencia entre máquinas de una pieza dada se expresa mediante la ecuación:

$$s_{i+1,k-1} + \theta_d(i+1, \Phi_{k-1}, 2 - h_{i+1,1}^+) + t_{2i+4-h_{i+1,1}^+, 2i+1, 0}^v \leq s_{i,k} \quad (6.8)$$

$$1 \leq i \leq m-1; 2 \leq k \leq n$$

Dado que la secuencia de piezas  $\bar{\Phi}$  no es conocida, deben ser introducidas las variables de asignación:

$$s_{i+1,k-1} + \sum_{j=1}^n \theta_d(i+1, j, 2 - h_{i+1,1}^+) x_{j,k-1} + t_{2i+4-h_{i+1,1}^+, 2i+1, 0}^v \leq s_{i,k} \quad (6.8.a)$$

$$1 \leq i \leq m-1; 2 \leq k \leq n$$

Agrupando las variables en el primer miembro y las constantes en el segundo miembro:

$$s_{i+1,k-1} - s_{i,k} + \sum_{j=1}^n \theta_d(i+1, j, 2 - h_{i+1,1}^+) x_{j,k-1} \leq -t_{2i+4-h_{i+1,1}^+, 2i+1, 0}^v \quad (6.8.b)$$

$$1 \leq i \leq m-1; 2 \leq k \leq n$$

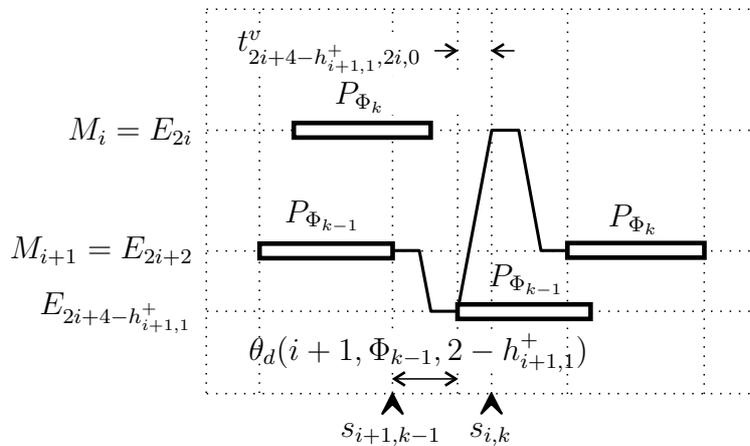


Figura 6.5: Precedencias por buffers de capacidad nula.

### 6.4 Función objetivo

La función objetivo, expresada también como una ecuación lineal, consiste en minimizar el instante de terminación del conjunto de piezas. Dicho instante coincide con el instante en que la pieza procesada en último lugar de la secuencia, termina de cargarse en la estación de salida. Consiste, por tanto, en el instante posible de inicio de salida de la última pieza procesada en la última máquina, más los tiempos empleados en las operaciones de transporte hasta la estación de salida.

Suponiendo conocida la secuencia de piezas  $\bar{\Phi}$ , la función objetivo se expresa mediante la ecuación:

$$\text{Minimizar: } s_{m,n} + \theta_d(m, \Phi_n, 1) \quad (6.9)$$

Dado que la secuencia de piezas  $\bar{\Phi}$  no es conocida, se deben introducir las variables de asignación:

$$\text{Minimizar: } s_{m,n} + \sum_{j=1}^n \theta_d(m, j, 1) x_{j,n} \quad (6.9.a)$$

### 6.5 Determinación de la secuencia de piezas

La determinación de una secuencia de piezas consta de dos pasos.

- ▶ Resolución del modelo de programación lineal entera mixta (que se resume en el apartado 6.6).
- ▶ A partir de los valores de las variables de asignación  $x_{j,k}$  hallados, se determina la secuencia de piezas  $\bar{\Phi}$  mediante la ecuación 6.10.

$$\Phi_k = j \text{ si } x_{j,k} = 1 \quad (6.10)$$

### 6.6 Resumen del modelo

El modelo de programación lineal entera mixta, conformado por su función objetivo y sus restricciones, se resume como:

#### Función objetivo

$$\text{Minimizar: } s_{m,n} + \sum_{j=1}^n \theta_d(m, j, 1) x_{j,n} \quad (6.9.a)$$

## Capítulo 6: Modelo lineal para la determinación de una secuencia de piezas

### Asignación de una pieza única a cada posición de la secuencia de piezas

$$\sum_{j=1}^n x_{j,k} = 1 \quad (1 \leq k \leq n) \quad (6.2)$$

### Asignación de una posición única de la secuencia de piezas a cada pieza

$$\sum_{k=1}^n x_{j,k} = 1 \quad (1 \leq j \leq n) \quad (6.3)$$

### Instante posible de inicio de salida de la primera pieza en la primera máquina

$$\sum_{j=1}^n (\theta_c(1, j, 1) + t_{1,j}^p) x_{j,1} - s_{1,1} \leq 0 \quad (6.4.b)$$

### Precedencia entre piezas

$$s_{i,k-1} - s_{i,k} + \sum_{j=1}^n \theta_d(i, j, 2 - h_{i,1}^+) x_{j,k-1} + \\ + \sum_{j=1}^n (\theta_c(i, j, 2 - h_{i,1}^-) + t_{i,j}^p) x_{j,k} \leq -t_{2i+2-h_{i,1}^+, 2i-2+h_{i,1}^-, 0}^v \quad (6.5.b)$$

$$1 \leq i \leq m ; 2 \leq k \leq n$$

### Precedencia entre máquinas

$$s_{i-1,k} - s_{i,k} + \sum_{j=1}^n (\theta_c(i, j, 2) + t_{i,j}^p) x_{j,k} \leq 0 \quad (6.6.b)$$

$$2 \leq i \leq m ; 1 \leq k \leq n$$

### Restricciones por el llenado de los buffers de capacidad no nula

$$s_{i+1,k-b_i-1} - s_{i,k} + \sum_{j=1}^n \theta_d(i+1, j, 2 - h_{i+1,1}^+) x_{j,k-b_i-1} + \\ + \sum_{j=1}^n \theta_c(i+1, j, 1) x_{j,k-b_i} \leq -t_{2i+2, 2i, 0}^v - t_{2i+4-h_{i+1,1}^+, 2i+1, 0}^v \quad (6.7.b)$$

$$1 \leq i \leq m - 1 ; b_i + 2 \leq k \leq n$$

### Restricciones por buffers de capacidad nula

$$s_{i+1,k-1} - s_{i,k} + \sum_{j=1}^n \theta_d(i+1, j, 2 - h_{i+1,1}^+) x_{j,k-1} \leq -t_{2i+4-h_{i+1,1}^+, 2i+1, 0}^v \quad (6.8.b)$$
$$1 \leq i \leq m-1; 2 \leq k \leq n$$



# Capítulo 7

## Procedimientos exactos y heurísticos de resolución del problema

### 7.1 Introducción

En este capítulo se presentan los procedimientos exactos y heurísticos diseñados para resolver el problema definido en el capítulo 2.

Se presentan dos enfoques exactos basados en branch and bound, y dos variantes heurísticas basadas en la separación de los procesos de determinación de la secuencia de piezas y de determinación de la secuencia de movimientos del robot.

### 7.2 Procedimiento branch and bound

En este apartado se presenta un enfoque del problema basado en la técnica branch and bound.

La factibilidad de aplicar un proceso de enumeración y exploración arborescente de tipo branch and bound para un problema dado requiere que sea posible:

- ▶ asociar a los vértices del árbol los estados por los que evoluciona el problema a medida que se determina la solución,
- ▶ asociar a los arcos operaciones de transformación que, aplicadas a los estados existentes, permitan la creación de nuevos estados,
- ▶ y procedimientos de poda por cota e infactibilidad.

Para el problema tratado en esta tesis, el algoritmo se define como:

- ▶ cada vértice del árbol de enumeración representa un *estado del sistema* tal como fue definido en el apartado 2.3.1;

## Capítulo 7: Procedimientos exactos y heurísticos de resolución del problema

---

- ▶ cada arco del árbol de enumeración (operación de transformación) representa un *movimiento del robot* como fue definido en el apartado 2.3.2.

La forma general de un procedimiento de enumeración y exploración arborescente de tipo branch and bound, para un objetivo de minimización como el tratado en esta tesis, se muestra en el Procedimiento 1.

---

### Procedimiento 1: Branch and bound

---

**Variable:** grafo  $G = (V, A)$   
**Variable:** estado  $e$   
**Variable:** conjunto de estados  $\xi$   
**Variable global:** mejor solución hallada  $s$   
**Variable global:** valor de la mejor solución hallada  $v$

- 1  $v = \infty$
- 2  $e = \text{CrearEstadoInicial} ()$
- 3  $\text{InsertarEstado} (G, e)$
- 4 **Repetir**
- 5 |  $\xi = \text{CrearDescendientes} (e)$
- 6 |  $\text{Eliminar} (G, e)$
- 7 |  $\text{InsertarEstados} (G, \xi)$
- 8 |  $e = \text{SiguientePadreNoExplorado} (G)$
- 9 **Hasta**  $e = \emptyset$

---

El Procedimiento 1 se explica a continuación.

- ▶ Un grafo  $G(V, A)$ , formado por un conjunto de vértices  $V$  y un conjunto de arcos  $A$ .
- ▶ Mejor solución hallada  $s$ . Su definición depende del problema a resolver. Para el problema de esta tesis, consiste en una secuencia de movimientos del robot como fue definida en el apartado 2.3.2.
- ▶ Valor de la mejor solución hallada  $v$ . Para el problema de esta tesis, consiste en el tiempo  $\rho$  al final del movimiento del estado objetivo (apartado 2.3.1, página 34) alcanzado cuando, a partir del estado inicial (apartado 2.3.1, página 33), se ejecuta la secuencia de movimientos  $s$ .
- ▶ La función **CrearEstadoInicial** (línea 2) genera el estado inicial  $e$ . Para el problema de esta tesis, es el estado donde las piezas y el robot se sitúan en la estación de entrada, y los tiempos de disponibilidad del robot y las máquinas son cero (apartado 2.3.1, página 33).
- ▶ La función **InsertarEstado** (línea 3) agrega el estado  $e$  en el grafo  $G$ .
- ▶ La función **CrearDescendientes** (línea 5) crea el conjunto de estados  $\xi$  formado por los estados descendientes de  $e$ . Su implementación varía con el problema a resolver, sin embargo, dentro de sus tareas incluye:
  - ▷ poda por infactibilidad,
  - ▷ poda por cota,

- ▷ y, cuando dentro del conjunto de estados descendientes se encuentre un estado objetivo, la determinación de la solución hallada  $s$  y del valor de la mejor solución hallada  $v$ ; y cuando dicho valor mejora el valor actual, la actualización de de este valor y de la solución hallada  $s$ .

La implementación de la función **CrearDescendientes** para el problema de esta tesis se explica en el apartado 7.2.1

- ▶ La función **Eliminar** (línea 6) elimina el estado  $e$  del grafo  $G$ .
- ▶ La función **InsertarEstados** (línea 7) agrega el conjunto de estados de  $\xi$  en el grafo  $G$ .
- ▶ La función **SiguientePadreNoExplorado** (línea 8) determina el siguiente estado de la estructura de vértices de  $G$  a ser ramificado (explorado). Éste varía con la estrategia de exploración empleada en la implementación de la función **SiguientePadreNoExplorado**. Son muy utilizadas: primero en profundidad (Depth First), primero en anchura (Breadth First) y primero el mejor (Best First). La función devuelve  $\emptyset$  cuando todos los vértices de  $G$  se han ramificado (explorado).

### 7.2.1 Procedimiento CrearDescendientes

Mediante **CrearDescendientes** se genera el conjunto de estados descendientes  $\xi$  de un estado  $e$  dado. El Procedimiento 2 muestra la función **CrearDescendientes** correspondiente a la implementación del problema de esta tesis. Éste se analiza a continuación.

**Procedimiento 2: Función CrearDescendientes****Argumento:** estado padre  $e$ **Variable:** grafo  $G = (V, A)$ **Variable:** estado descendiente  $e_h$ **Variable:** movimiento  $\mu$ **Variable:** valor de una solución encontrada  $a$ **Variable:** valor de una cota calculada  $C$ **Resultado:** conjunto de estados  $\xi$ **Variable global:** mejor solución hallada  $s$ **Variable global:** valor de la mejor solución hallada  $v$ 

```

1  $\xi = \emptyset$ 
2 Para  $\delta = 1$  Hasta 2 Hacer
3   Para  $j = 1$  Hasta  $n$  Hacer
4      $\mu = (j, \delta)$ 
5      $e_h = \text{AplicarMovimiento}(e, \mu)$ 
6     Si  $e_h \neq \emptyset$  entonces
7       Si EsEstadoObjetivo( $e_h$ ) entonces
8          $a = \text{ValorSolucion}(e_h)$ 
9         Si  $a < v$  entonces
10           $v = a$ 
11           $s = \text{HallarSolucion}(G, e_h)$ 
12       Si No
13          $C = \text{HallarCota}(e_h)$ 
14         Si  $C < v$  entonces
15           $\xi = \xi + e_h$ 
16 Retornar  $\xi$ 

```

- **Generación de movimientos.** Para todo desplazamiento  $\delta \in \{1, 2\}$  (línea 2) y para toda pieza  $P_j$  (línea 3) se aplica al estado padre  $e$  el movimiento  $\mu = (j, \delta)$  mediante **AplicarMovimiento** (línea 5).

El resultado puede ser  $\emptyset$  cuando el movimiento es infactible de acuerdo a los casos de infactibilidad mostrados en el apartado 2.4.3. Se está aplicando, en ese caso, poda por infactibilidad.

- **Hallar estado objetivo.** La función **EsEstadoObjetivo** (línea 7) comprueba la aparición de un estado objetivo y por tanto de una solución.

Mediante la función **ValorSolucion** se determina su valor  $a$  (línea 8).

Si este valor mejora el valor de la mejor solución encontrada  $v$ :

- ▷ éste se actualiza recibiendo el valor de  $a$  (línea 10)
- ▷ y, en la línea 11, la función **HallarSolucion** determina la solución  $s$  de la forma de una secuencia de movimientos del robot (como se presentó en el apartado 2.3.2, página 34) asociada al estado objetivo  $e_h$ .

- **Poda por cota.** La función **HallarCota** aplicada a un estado  $e_h$  (línea 13) provee la cota asociada a dicho estado, basada en los dos procedimientos de acotación descritos en los capítulos 4 y 5.

Si el valor de la cota es mayor o igual al valor de la mejor solución encontrada hasta el momento  $v$  (cuando, como en este problema, el sentido de la función objetivo es de minimización), el estado  $e_h$  no se incorpora al conjunto de estados descendientes (es podado).

En el caso contrario el estado  $e_h$  se incorpora al conjunto de estados descendientes  $\xi$  (línea 15).

### 7.3 Procedimiento branch and bound con poda heurística

Consiste en un procedimiento exacto donde se ejecuta, iterativamente, una variante heurística del procedimiento de enumeración y exploración arborescente de tipo branch and bound presentado en el apartado 7.2.

La naturaleza heurística del procedimiento branch and bound se debe a que en dicho procedimiento, al proceso de poda "exacta" (por infactibilidad y por acotación) se incorpora, de forma forzada (heurística), la poda de aquellos estados de mayor valor de cota de entre los descendientes de un estado dado, por considerarse a estos menos prometedores para conducir a mejores soluciones.

En cada iteración, el número de estados descendientes podados de un estado dado, se va reduciendo con respecto a la iteración anterior. Comienza en un valor muy grande al inicio (en la primera ejecución del procedimiento branch and bound) y gradualmente se va reduciendo en las siguientes ejecuciones del procedimiento branch and bound, hasta hacerse nulo luego de un cierto número de iteraciones, obteniéndose, en ese momento, una solución exacta.

En aquellos casos en los que se verifique la hipótesis de que los estados de menor cota efectivamente conducen a mejores soluciones, cabría esperarse una mayor eficiencia de este procedimiento frente al branch and bound tradicional mostrado en el apartado 7.2, debido, esencialmente, a que las mejores soluciones aparecerían tempranamente (en las primeras iteraciones) del proceso exploratorio y fruto de esto, el proceso de poda sería mayor en las iteraciones sucesivas.

Su implementación se muestra en el Procedimiento 3.

**Procedimiento 3:** Procedimiento branch and bound con poda heurística**Argumento:**  $w_{ini}$ **Argumento:**  $w_{inc}$ **Variable:** grafo  $G = (V, A)$ **Variable:** estado  $e$ **Variable:** conjunto de estados  $\xi$ **Variable:** número de estados de  $\xi$  que son conservados después de una poda:  $w$ **Variable:** condición de poda heurística  $c$ **Variable global:** mejor solución hallada  $s$ **Variable global:** valor de la mejor solución hallada  $v$ 

```

1  $w = w_{ini}$ 
2 Repetir
3    $c = 0$ 
4    $v = \infty$ 
5    $e = \text{CrearEstadoInicial} ()$ 
6    $\text{InsertarEstado} (G, e)$ 
7   Repetir
8      $\xi = \text{CrearDescendientes} (e)$ 
9      $\text{Eliminar} (G, e)$ 
10    Si  $\text{EliminarPeoresDescendientes} (\xi, w)$  entonces
11       $c = 1$ 
12     $\text{InsertarEstados} (G, \xi)$ 
13     $e = \text{SiguientePadreNoExplorado} (G)$ 
14  Hasta  $e = \emptyset$ 
15   $w = w + w_{inc}$ 
16 Mientras  $c = 1$ 

```

- La fase iterativa se aplica entre las líneas 2 y 16. Un parámetro entero  $w$  establece la intensidad de la poda heurística, dado que es el número de estados descendientes de un estado dado que en cada iteración no son podados. Éste se inicializa con el valor  $w_{ini}$  (línea 1) al principio del procedimiento y se incrementa al final de cada ciclo (línea 15) en un valor incremental  $w_{inc}$ .
- En la línea 3 se inicializa con valor cero la variable  $c$  que controla el bucle de la fase iterativa (se permanece en el bucle mientras  $c = 1$ ).
- Entre las líneas 4 y 14 hay un procedimiento del tipo de exploración arborescente como el branch and bound del Procedimiento 1, con excepción de las líneas 10 y 11 que se han agregado ahora.
- La función **EliminarPeoresDescendientes** (línea 10) elimina los estados de mayor valor de cota entre el conjunto de estados descendientes de un estado dado, tal que solo queden en  $\xi$  los  $w$  estados más prometedores. No se eliminan estados cuando el número de estados en  $\xi$  es menor o igual a  $w$ . Si no se han eliminado estados de  $\xi$ , la función **EliminarPeoresDescendientes** retorna el valor **falso** indicando que no se ha producido una poda heurística. La función **EliminarPeoresDescendientes** retorna **verdadero** si se ha producido una poda heurística. Si este es el caso, en la línea 11 se hace la asignación  $c = 1$  para indicar que debe finalizarse la fase iterativa.

Puede verse que si una de las ejecuciones del branch and bound dentro del bucle principal termina sin producirse poda heurística, y por tanto  $c$  se ha mantenido en el valor nulo asignado en la línea 3, entonces la solución hallada es exacta y la condición  $c = 0$  produce la terminación del procedimiento.

## 7.4 Procedimiento por separación

Consiste en un enfoque basado en determinar, de forma separada, la secuencia de piezas y la secuencia de movimientos del robot. En un primer paso se determina una secuencia de piezas  $\bar{\Pi}$  (apartado 7.4.1) y, en un paso posterior, se halla la secuencia de movimientos del robot (apartado 7.4.2) con arreglo a que la secuencia de pieza de las soluciones halladas sigan la secuencia de piezas predefinida. En otras palabras, debe cumplirse para todo estado objetivo:  $\bar{\pi} = \bar{\Pi}$ .

El proceso completo se muestra en el Procedimiento 5.

### 7.4.1 Determinación de la secuencia de piezas

El proceso de determinación de una secuencia de piezas  $\bar{\Pi}$  se basa en la resolución del modelo de programación lineal entera mixta presentado en el capítulo 6. Debido a la complejidad de los algoritmos de programación entera mixta, no hay garantía de que el software encargado de resolver el programa lineal termine dentro de un tiempo razonable, para dar lugar al proceso siguiente donde se halla la secuencia de movimientos del robot. Para garantizar la obtención de al menos una solución en un tiempo razonable, la determinación de la secuencia de piezas se estructura en dos fases consecutivas:

- ▶ Aplicación de una variante heurística para problemas lineales mixtos, denominada *Relax And Fix*, que permite hallar una solución en un tiempo reducido debido a una reducción sustancial de variables binarias<sup>1</sup>.
- ▶ Resolución del modelo mixto completo como se enunció en el capítulo 6, pero utilizando el valor de la solución obtenida en la fase anterior para acelerar su resolución (mediante el agregado de una restricción).

Los dos puntos anteriores se implementan en el procedimiento **HallarSecuenciaPredefinida**, cuyo algoritmo se muestra en el Procedimiento 4.

---

<sup>1</sup>En efecto, se pasa de un total de  $n^2$  variables binarias, cuando se tiene el modelo de programación lineal mixto completo, a un total de  $n$  variables binarias, cuando se emplea la heurística *Relax And Fix*.

---

### Procedimiento 4: Procedimiento HallarSecuenciaPredefinida

---

```
1  $p = \text{CrearProblema} ()$ 
2  $\text{AgregarVariables} (p)$ 
3  $\text{AgregarRestricciones} (p)$ 
4  $n_1 = \text{NumeroDeRestricciones} (p)$ 
5 Para  $k = 1$  Hasta  $n$  Hacer
6   Para  $j = 1$  Hasta  $n$  Hacer
7      $\lfloor \text{HacerBinaria} (x_{j,k})$ 
8   Si  $k < n$  entonces
9      $\lfloor \text{Resolver} (p)$ 
10     $j = 1$ 
11    Mientras  $\text{HallarValor} (p, x_{j,k}) = 0$  :
12       $\lfloor j = j + 1$ 
13     $\Pi_k = j$ 
14     $\text{AgregarRestriccion} (p, "x_{j,k} = 1")$ 
15  $\text{EliminarRestricciones} (p, n_1 + 1, n_1 + n - 1)$ 
16  $v = \text{HallarValorSolucion} (p)$ 
17  $\text{DefinirValorInicial} (p, v)$ 
18  $\text{Resolver} (p)$ 
```

---

En el Procedimiento 4 se realizan las siguientes tareas.

- **Crear Problema.** En la línea 1, mediante la función **CrearProblema**, se crea un problema  $p$  vacío, es decir, sin variables ni restricciones.

Mediante **AgregarVariables** (línea 2) se agregan al modelo las variables, definidas todas de tipo real (más adelante las variables de asignación  $x$  se irán definiendo binarias a medida que progrese el procedimiento).

Finalmente, en línea 3 se incorporan las restricciones al modelo mediante la función **AgregarRestricciones**.

El número de restricciones del modelo se determina mediante la función **NumeroDeRestricciones** aplicada al modelo  $p$  en la línea 4.

- **Procedimiento Relax and Fix.** Abarca las líneas 5 a 14 del Procedimiento 4. Para toda posición  $k$  de la secuencia de piezas hacer lo siguiente.

- ▷ Definir como binaria la variable de asignación  $x_{j,k}$  de toda pieza  $P_j$  ( $1 \leq j \leq n$ ) en la posición  $k$ , mediante la aplicación de la función **HacerBinaria** (líneas 6 a 7).
- ▷ Sólo si  $k < n$  (se hacen  $n$  iteraciones pero  $n - 1$  resoluciones, pues la última asignación es obligada) resolver el problema mixto con  $n$  variables de asignación binarias  $x_{1,k}$  a  $x_{n,k}$  (línea 9).
- ▷ Para toda pieza  $P_j$ , si el valor de la variable de asignación  $x_{j,k}$  correspondiente a la posición en curso  $k$ , determinado por la función **HallarValor**, es igual a 1 entonces asignar la pieza  $P_j$  en la posición  $k$  de la secuencia de piezas predefinida  $\bar{\Pi}$ , mediante la igualación  $\bar{\Pi}_k = j$  (línea 13).

---

## Procedimiento por separación

---

- ▷ Fijar a 1 el valor de la variable de asignación  $x_{j,k}$  agregando la restricción  $x_{j,k} = 1$ .
- ▶ **Problema no relajado.** Abarca las líneas 15 a 18 del Procedimiento 4, donde se realizan las siguientes tareas.
  - ▷ Eliminar las restricciones utilizadas para fijar los valores de las variables de asignación creadas en la línea 14. Mediante **EliminarRestricciones** se eliminan estas restricciones que se hallan comprendidas entre  $n_1 + 1$  y  $n_1 + n - 1$  (línea 15).
  - ▷ Asignar a  $v$  el valor de la solución encontrada mediante la función **HallarValorSolucion** (línea 16).
  - ▷ Introducir el valor de la solución encontrada  $v$  (mediante el agregado de una restricción) como valor de una solución de inicio en la siguiente resolución del programa matemático, mediante **DefinirValorInicial** (línea 17).
  - ▷ Resolver el programa matemático (línea 18). Obsérvese que todas las variables de asignación ya son tipo binario por la aplicación de **HacerBinaria** en la línea 7. Por esto, en esta fase se resuelve el modelo completo.

### 7.4.2 Determinación de la secuencia de movimientos del robot

El proceso de determinación de la secuencia de movimientos del robot consiste en un procedimiento de exploración arborescente de tipo branch and bound, como el del Procedimiento 1 explicado en el apartado 7.2. Éste está comprendido entre las líneas 2 y 10 del Procedimiento 5. Sin embargo difiere de aquél en que la función **CrearDescendientes** (línea 6) incorpora el criterio de poda por infactibilidad del apartado 2.4.4, consistente en la eliminación de aquellos estados que violen la secuencia de piezas predefinida  $\bar{\Pi}$ .

---

#### Procedimiento 5: Procedimiento por separación

---

**Variable:** grafo  $G = (V, A)$   
**Variable:** estado  $e$   
**Variable:** conjunto de estados  $\xi$   
**Variable global:** mejor solución hallada  $s$   
**Variable global:** valor de la mejor solución hallada  $v$

- 1  $\bar{\Pi} = \text{HallarSecuenciaPredefinida} ()$
- 2  $v = \infty$
- 3  $e = \text{CrearEstadoInicial} ()$
- 4  $\text{InsertarEstado} (G, e)$
- 5 **Repetir**
- 6      $\xi = \text{CrearDescendientes} (e)$
- 7      $\text{Eliminar} (G, e)$
- 8      $\text{InsertarEstados} (G, \xi)$
- 9      $e = \text{SiguientePadreNoExplorado} (G)$
- 10 **Hasta**  $e = \emptyset$

---

## 7.5 Procedimiento por separación aleatorizado

Como en el enfoque presentado en el apartado 7.4, consiste en separar el cálculo de la secuencia de piezas y la secuencia de movimientos del robot. En este caso, el proceso de determinación de la secuencia de piezas se basa en un procedimiento aleatorio parametrizado, mientras que el proceso de determinación de la secuencia de movimientos del robot consiste en una variante heurística del procedimiento de enumeración y exploración arborescente basado en branch and bound.

Su implementación se muestra en el Procedimiento 6. Éste comprende dos procesos:

- ▶ un proceso de determinación de una *secuencia de piezas predefinida*  $\bar{\Pi}$  (función **HallarSecuenciaPredefinida** en línea 1) utilizando el Procedimiento 4,
- ▶ y un proceso cíclico formado por dos subprocesos:
  - ▷ un subproceso aleatorizado parametrizado en la variable  $\gamma$  (función **HallarSecuenciaAleatorizada** en línea 6), para la determinación de una *secuencia de piezas aleatorizada*  $\bar{\nabla}$  generada a partir de la *secuencia predefinida*  $\bar{\Pi}$ ,
  - ▷ y un subproceso heurístico basado en branch and bound para la determinación de la secuencia de movimientos del robot (función **HallarSecuenciaMovimientosRobot** en línea 7) respetando la secuencia de piezas aleatorizada  $\bar{\nabla}$  determinada en el subproceso anterior.

El parámetro  $\gamma$  actúa de modo similar a la temperatura en el algoritmo de recocido simulado (simulated annealing o SA) propuesto en Kirkpatrick *et al.* (1983). Dicho parámetro se inicializa con un valor  $\gamma_{ini}$  (línea 2) y se incrementa durante el proceso en un valor incremental  $\gamma_{inc}$  (línea 5) a intervalos iguales de tiempo de ejecución del algoritmo. La función **ActualizarGamma** (línea 4) realiza la comprobación de que un intervalo de tiempo ha transcurrido y actualiza el valor del parámetro  $\gamma$ .

---

### Procedimiento 6: Algoritmo por separación aleatorizado

---

**Variable:** secuencia de piezas predefinida  $\bar{\Pi}$   
**Variable:** secuencia de piezas aleatorizada  $\bar{\nabla}$   
**Variable:** parámetro  $\gamma$   
**Variable:** valor inicial del parámetro  $\gamma$ :  $\gamma_{ini}$   
**Variable:** valor incremental del parámetro  $\gamma$ :  $\gamma_{inc}$

- 1  $\bar{\Pi} = \text{HallarSecuenciaPredefinida} ()$
- 2  $\gamma = \gamma_{ini}$
- 3 **Hacer siempre**
- 4     **Si** **ActualizarGamma** () **entonces**
- 5          $\gamma = \gamma + \gamma_{inc}$
- 6          $\bar{\nabla} = \text{HallarSecuenciaAleatorizada} (\bar{\Pi}, \gamma)$
- 7         **HallarSecuenciaMovimientosRobot** ( $\bar{\nabla}$ )

---

Las funciones **HallarSecuenciaAleatorizada** y **HallarSecuenciaMovimientosRobot** se explican, respectivamente, en los apartados 7.5.1 y 7.5.2.

### **7.5.1 Determinación de la secuencia de piezas aleatorizada**

La función **HallarSecuenciaAleatorizada**, referida en la línea 6 del Procedimiento 6, implementa el proceso de determinación de una secuencia de piezas aleatorizada  $\bar{\nabla}$  a partir de una secuencia de piezas predefinida  $\bar{\Pi}$ , como se muestra en el Procedimiento 7.

Consiste en un procedimiento donde las piezas asignadas a las distintas posiciones de la secuencia predefinida  $\bar{\Pi}$  son asignadas a una nueva posición de la secuencia  $\bar{\nabla}$ , mediante un sorteo ponderado. En este proceso, si la pieza de la posición  $k$  de  $\bar{\Pi}$  es asignada por sorteo a la posición  $i$  de la secuencia de piezas  $\bar{\nabla}$ , se hace la asignación mediante la igualdad:

$$\nabla_i = \Pi_k$$

El proceso de sorteo se basa en unos índices de prioridad  $\varrho_k$ , asociados a cada posición  $k$  de la secuencia predefinida  $\bar{\Pi}$ , cuyo valor se calcula mediante:

$$\varrho_k = k^{-\gamma} \tag{7.1}$$

Con los índices de prioridad se determinan las probabilidades  $p_k$  con las que se hacen los sorteos, por medio de la expresión:

$$p_k = \frac{\varrho_k}{\sum_{k=1}^n \varrho_k} \tag{7.2}$$

Como puede observarse en la expresión 7.2, si  $\gamma > 0$ , las probabilidades se distribuyen decrecientemente conforme crecen las posiciones de la secuencia aleatorizada, de tal modo que si en cada sorteo la pieza seleccionada es siempre aquella que ha recibido la mayor probabilidad, entonces la secuencia de piezas aleatorizada  $\bar{\nabla}$  coincidirá con la secuencia predefinida  $\bar{\Pi}$ . Como no ocurrirá siempre que la pieza de mayor probabilidad sea favorecida por el sorteo, se producirán diversas secuencias  $\bar{\nabla}$  con cierta similitud respecto de  $\bar{\Pi}$ , pero diferente en el mayor número de casos.

En la expresión 7.1, el exponente  $\gamma$  tiene por finalidad acercar o alejar los distintos valores de  $\varrho_k$  y de las probabilidades  $p_k$ . En el caso extremo en que  $\gamma = 0$ , todas las piezas reciben la misma probabilidad  $p_k = 1/n$ . En los casos en que  $\gamma > 0$ , los valores de  $p_k$  se separan conforme  $\gamma$  aumenta.

En los sucesivos sorteos, si la pieza en la posición  $k$  de  $\bar{\Pi}$  es asignada por sorteo a la posición  $i$  de secuencia de piezas  $\bar{\nabla}$ , se hace la igualdad  $\nabla_i = \Pi_k$  y el índice de prioridad  $\varrho_i$  se hace cero para evitar que sea asignada nuevamente. Seguidamente las probabilidades se recalculan mediante 7.2.

El procedimiento **HallarSecuenciaAleatorizada** mostrado en el Procedimiento 7 implementa el proceso de sorteo introducido en los párrafos anteriores. Éste se explica a

continuación.

- ▶ En las líneas 2 a 4 se determinan los índices de prioridad. Éstos se calculan en la línea 3 y su suma  $t$ , utilizada en el denominador de 7.2, acumula los índices de prioridad en la línea 4.
- ▶ Para toda posición  $i$  de la secuencia aleatorizada (líneas 5 a 13) se hace:
  - ▷ Un sorteo (función **NumeroAleatorio**) de un número real entre 0 y 1 el cual se asigna a la variable  $q$  (línea 6).
  - ▷ Para toda posición  $k$  en la secuencia de piezas predefinida comprobar que el valor aleatorio  $q$  está comprendido en un rango de amplitud  $p_k = \varrho_k/t$  (línea 9) hasta que un valor de  $k$  es encontrado (línea 10).
  - ▷ Hacer la asignación  $\nabla_k = \Pi_i$  (línea 13). Así la pieza en la posición  $i$  de la secuencia predefinida  $\bar{\Pi}$  es asignada a la posición  $k$  de la secuencia aleatorizada  $\bar{\nabla}$ .
  - ▷ Actualizar la suma de índices de prioridad  $t$  restando el índice de prioridad  $\varrho_k$  (línea 11).
  - ▷ Asignar a  $\varrho_k$  el valor cero para anular su influencia en los siguientes sorteos (línea 12).

---

### Procedimiento 7: Función **HallarSecuenciaAleatorizada**

---

**Argumento:** secuencia de piezas predefinida  $\bar{\Pi}$

**Argumento:** parámetro  $\gamma$

**Variable:** conjunto de índices de prioridad  $\varrho_k$

**Variable:** suma de los índices de prioridad  $t$

**Variable:** suma de probabilidades  $r$

**Variable:** secuencia de piezas aleatorizada  $\bar{\nabla}$

```
1  $t = 0$ 
2 Para  $k = 1$  Hasta  $n$  Hacer
3    $\varrho_k = k^{-\gamma}$ 
4    $t = t + \varrho_k$ 
5 Para  $i = 1$  Hasta  $n$  Hacer
6    $q = \text{NumeroAleatorio}(0, 1)$ 
7    $r = 0$ 
8   Para  $k = 1$  Hasta  $n$  Hacer
9      $r = r + \varrho_k/t$ 
10    Si  $r > q$  entonces salir de Para en curso
11     $t = t - \varrho_k$ 
12     $\varrho_k = 0$ 
13     $\nabla_i = \Pi_k$ 
14 Retornar  $\bar{\nabla}$ 
```

---

### 7.5.2 Determinación de la secuencia de movimientos del robot

Dada una secuencia de piezas aleatorizada  $\bar{\nabla}$ , la función **HallarSecuenciaMovimientosRobot** (referida en la línea 7 del Procedimiento 6) permite determinar una secuencia de movimientos del robot, respetando la secuencia  $\bar{\nabla}$  dada.

Dicho procedimiento está basado en la construcción iterativa de la solución mediante un procedimiento de enumeración y exploración arborescente de tipo branch and bound. Mediante este método, en cada iteración se determina una porción de la solución (en el problema de esta tesis, un subconjunto de movimientos del robot), hasta que, tras un cierto número de iteraciones, se obtiene una expresión completa de la solución.

Para que este método pueda ser aplicado a un ejemplar dado, se requiere definir las siguientes expresiones matemáticas:

- ▶ **Indicador de progreso,  $x$ .** Asociado a un estado dado, representa el grado de progreso en la determinación de una solución alcanzado por dicho estado. Debería tener un valor mínimo cuando este indicador se determina para un estado inicial (donde el grado de determinación de la solución es nulo) y un valor máximo cuando se determina para un estado objetivo (estado donde la solución es completa).
- ▶ **Indicador de progreso máximo,  $X$ .** Define el valor de máximo admitido para el indicador de progreso para los estados enumerados en una iteración dada.
- ▶ **Criterio de mejor estado.** Mediante este criterio, cuando es aplicado a dos estados  $e_h$  y  $e_b$ , es posible determinar cual de ellos es el más prometedor para conducir el proceso exploratorio hacia las mejores soluciones.

Para el problema definido en esta tesis, estas expresiones se definen a continuación.

- ▶ El **indicador de progreso** asociado a un estado  $e_h$  se define como:

$$x_h = \sum_{j=1}^n l_j - n \quad (7.3)$$

De la expresión 7.3 puede observarse que  $x$  adquiere su valor mínimo  $x = 0$  en el estado inicial, donde todas las piezas están en la estación de entrada, mientras que  $x$  encuentra su valor máximo  $x = 2mn$  en los estados objetivo, donde todas las piezas están en la estación de salida.

- ▶ El **indicador de progreso máximo  $X$** , para una iteración  $k$  dada, se define mediante la expresión:

$$X = X_{ini} + (k - 1) X_{inc} \quad (7.4)$$

Según la expresión 7.4,  $X$  parte de un valor inicial  $X_{ini}$  y, a medida que se realizan nuevas iteraciones, se incrementa en un valor incremental  $X_{inc}$ . Tanto  $X_{ini}$  como  $X_{inc}$ , son parámetros constantes establecidos por el usuario.

- ▶ **Criterio de mejor estado.** Sean  $e_h$  y  $e_b$  dos estados tal que, respectivamente:
  - ▷  $x_h$  y  $x_b$  son los valores del indicador de progreso de dichos estados, determinados por la ecuación 7.3,
  - ▷ y,  $C_h$  y  $C_b$  son los valores de las cotas inferiores halladas para estos estados.

Entonces, el criterio de mejor estado consiste en aplicar los siguientes criterios:

- ▷ Si  $x_h > x_b$  entonces  $e_h$  es el mejor estado.
- ▷ Si  $x_h < x_b$  entonces  $e_b$  es el mejor estado.
- ▷ Si  $x_h = x_b$  entonces se aplica un criterio de desempate basado en las cotas calculadas para estos estados:
  - ▷ Si  $C_h < C_b$  entonces  $e_h$  es el mejor estado.
  - ▷ Si  $C_h \geq C_b$  entonces  $e_b$  es el mejor estado.

Una vez que estas expresiones han sido definidas, el procedimiento consiste en la realización de una secuencia de iteraciones basadas en un procedimiento de tipo exploratorio de tipo branch and bound donde, a los procesos de poda ya explicados, se añade un proceso de poda por progreso.

En cada iteración  $k$  se realizan los siguientes pasos.

- ▶ Para dicha iteración se determina el **indicador de progreso máximo**;
- ▶ Se determina el estado inicial  $e$  desde donde se va a iniciar el proceso exploratorio.
  - ▷ Si es la primera iteración ( $k = 1$ ), el estado inicial  $e$  se define como el estado inicial global, como fue definido en el apartado 2.3.1.
  - ▷ Si no es la primera iteración ( $k > 1$ ), el estado inicial  $e$  es el mejor estado  $e_b$  encontrado en la iteración anterior  $k - 1$  (hallado por la aplicación del criterio de mejor estado a cada uno de los estados generados durante el proceso de enumeración de estados).
- ▶ Se realiza un procedimiento exploratorio iniciado desde un estado  $e$  donde, para cada estado  $e_h$  hallado durante el proceso de enumeración de estados se realizan las siguientes tareas.
  - ▷ Se aplica poda por infactibilidad y poda por cota.
  - ▷ Se halla el **indicador de progreso**  $x_h$  y, si este valor sobrepasa el valor de **indicador de progreso máximo**  $X$  definido para la iteración en curso, el estado  $e_h$  es podado (podado por progreso).
  - ▷ Sólo en el caso en que el estado  $e_h$  no es podado, se determina si éste es mejor que el estado  $e_b$  (mejor estado encontrado hasta ese momento de la exploración) mediante la aplicación, a ambos estados, del **criterio de mejor estado**. Si a partir de dicho criterio se determina que  $e_h$  es mejor que  $e_b$  entonces el estado  $e_b$  toma el valor  $e_h$  (así el estado  $e_b$  mantiene siempre el mejor estado hallado hasta ese momento).
  - ▷ Para todo estado objetivo se determina la solución  $s$  y el valor de dicha solución. Si este valor mejora el valor actual de la solución  $v$ , se actualiza la solución  $s$  el valor dicha solución  $v$ .

Si el número de estados podados por progreso en la iteración en curso es cero, el procedimiento termina y el resultado consiste en la mejor solución encontrada a lo largo del procedimiento.

## Procedimiento por separación aleatorizado

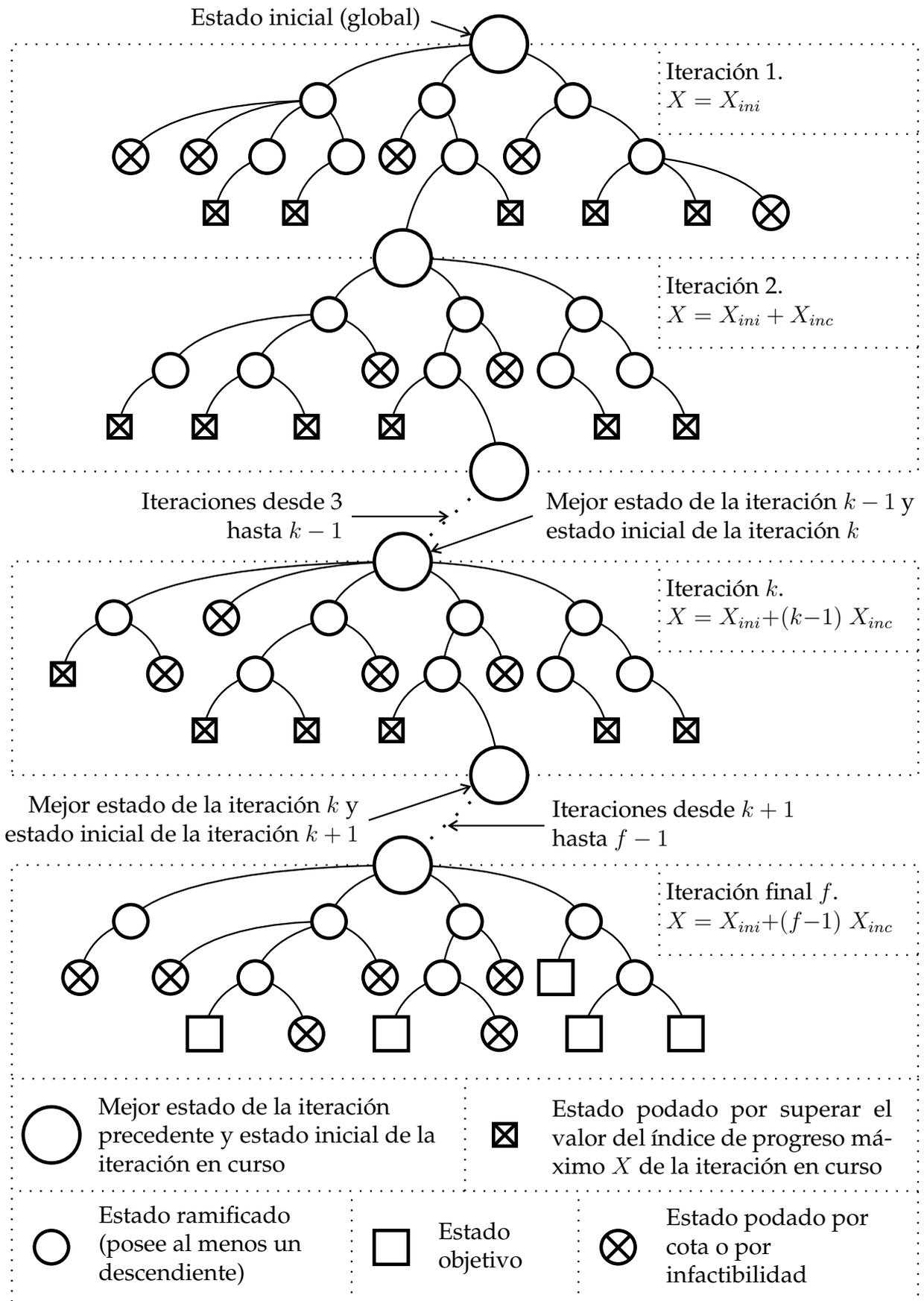


Figura 7.1: Esquema del procedimiento de determinación de la secuencia de movimientos del robot basado en un proceso de exploración con poda por progreso.

La figura 7.1 muestra los distintos procesos exploratorios que se ejecutan iterativamente desde una primera iteración, donde  $k = 1$ , hasta una iteración final, donde  $k = f$ . El número de iteraciones necesarias (el valor  $f$ ) no se conoce de antemano, sino que surge de condiciones propias del proceso exploratorio, como se explica en el apartado siguiente, donde se describe su implementación.

### 7.5.2.1 Procedimiento HallarSecuenciaMovimientosRobot

El proceso de determinación de la secuencia de movimientos del robot asociado a una secuencia de piezas aleatorizada dada  $\bar{V}$ , se implementa en el procedimiento **HallarSecuenciaMovimientosRobot**, cuyos pasos se muestran en el Procedimiento 8.

---

#### Procedimiento 8: HallarSecuenciaMovimientosRobot

---

**Argumento:** secuencia de piezas aleatorizada:  $\bar{V}$   
**Variable:** indicador de progreso máximo inicial  $X_{ini}$   
**Variable:** indicador de progreso máximo incremental  $X_{inc}$   
**Variable:** indicador de progreso máximo de la iteración en curso  $X$   
**Variable global:** condición de fin de exploración  $b$

- 1  $b = 0$
- 2  $X = X_{ini}$
- 3  $e = \text{CrearEstadoInicial}()$
- 4 **Hacer siempre**
- 5     ExploracionConPodaPorProgreso ( $e, X$ )
- 6     **Si**  $b = 1$  **entonces** Terminar
- 7      $X = X + X_{inc}$
- 8      $e = e_b$

---

El procedimiento **HallarSecuenciaMovimientosRobot** incluye la variable  $b$  para expresar la condición de fin de las iteraciones.

El procedimiento **HallarSecuenciaMovimientosRobot** se define como un núcleo iterativo, abarcado por las líneas 4 a 8, consistente en:

- ▶ La determinación del estado inicial, con el que se inicia el proceso, se realiza:
  - ▷ en la línea 3, cuando se ejecuta la primera iteración,
  - ▷ y en la línea 8, cuando se ejecutan iteraciones distintas de la primera.
- ▶ La determinación del indicador de progreso máximo, empleado en el proceso de poda por progreso, se determina en las líneas 2 y 7.
- ▶ La ejecución de un procedimiento exploratorio (invocado en la línea 5) realizado por el procedimiento **ExploracionConPodaPorProgreso**.

En dicho procedimiento se permite modificar la variable global  $b$ , dando a esta un valor 1 en el caso en que no se ha producido poda por progreso. En dicho caso, se ha llegado a la iteración final, y por ello el proceso iterativo termina (línea 6).

**Procedimiento ExploracionConPodaPorProgreso.** El proceso de enumeración y exploración de estados lo realiza el procedimiento **ExploracionConPodaPorProgreso**, cuyos pasos se muestran en el Procedimiento 9.

---

**Procedimiento 9: Función ExploracionConPodaPorProgreso**

---

**Argumento:** estado a partir del cual se inicia la enumeración:  $e$

**Argumento:** indicador de progreso máximo en curso  $X$

**Variable:** grafo  $G = (V, A)$

**Variable global:** mejor solución hallada  $s$

**Variable global:** valor de la mejor solución hallada  $v$

**Variable global:** mejor estado  $e_b$

**Variable global:** cota inferior del mejor estado  $C_b$

**Variable global:** indicador de progreso del mejor estado  $x_b$

**Variable global:** condición de fin de exploración  $b$

1  $v = \infty$

2  $b = 1$

3 InsertarEstado ( $G, e$ )

4 **Repetir**

5      $\xi = \text{CrearDescendientes2}(e, X)$

6     Eliminar ( $G, e$ )

7     InsertarEstados ( $G, \xi$ )

8      $e = \text{SiguientePadreNoExplorado}(G)$

9 **Hasta**  $e = \emptyset$

---

Se agregan las siguientes variables:

- ▶ el estado  $e_b$ , donde se mantiene el mejor estado encontrado hasta el momento del proceso de enumeración y exploración de estados;
- ▶ la cota  $C_b$ , cota inferior determinada para el estado  $e_b$ ;
- ▶ el indicador de progreso  $x_b$  determinado para dicho estado;
- ▶ y la condición de fin de iteraciones  $b$  para indicar al procedimiento invocador si deben hacerse, o no, más iteraciones.

El procedimiento **ExploracionConPodaPorProgreso** implementa los pasos de la función branch and bound (apartado 7.2) con excepción de la función **CrearDescendientes**, la que es sustituida por **CrearDescendientes2**.

**Función CrearDescendientes2.** **CrearDescendientes2** implementa los pasos de la función **CrearDescendientes** (tratada en el apartado 7.2.1) con el agregado de dos nuevos procesos:

- ▶ determinación del mejor estado (necesario para la iniciación de la iteración siguiente),
- ▶ y el proceso de poda por progreso.

## Capítulo 7: Procedimientos exactos y heurísticos de resolución del problema

Los pasos de la función **CrearDescendientes2** se muestran en el Procedimiento 10.

---

### Procedimiento 10: Función **CrearDescendientes2**

---

**Argumento:** estado padre  $e$

**Argumento:** indicador de progreso máximo  $X$

**Variable:** grafo  $G = (V, A)$

**Variable:** estado descendiente  $e_h$

**Variable:** movimiento  $\mu$

**Variable:** valor de una solución encontrada  $a$

**Variable:** valor de una cota calculada  $C$

**Resultado:** conjunto de estados  $\xi$

**Variable global:** mejor solución hallada  $s$

**Variable global:** valor de la mejor solución hallada  $v$

**Variable global:** mejor estado  $e_b$

**Variable global:** cota inferior del mejor estado  $C_b$

**Variable global:** indicador de progreso del mejor estado  $x_b$

**Variable global:** indicador de progreso del estado descendiente  $x_h$

**Variable global:** condición de fin de exploración  $b$

```
1  $\xi = \emptyset$ 
2 Para  $\delta = 1$  Hasta 2 Hacer
3   Para  $j = 1$  Hasta  $n$  Hacer
4      $\mu = (j, \delta)$ 
5      $e_h = \text{AplicarMovimiento}(e, \mu)$ 
6     Si  $e_h \neq \emptyset$  entonces
7       Si EsEstadoObjetivo ( $e_h$ ) entonces
8          $a = \text{ValorSolucion}(e_h)$ 
9         Si  $a < v$  entonces
10           $v = a$ 
11           $s = \text{HallarSolucion}(G, e_h)$ 
12       Si No
13         Para el estado  $e_h$  hallar:  $x_h = \sum_{j=1}^n l_j - n$ 
14         Si  $x_h > X$  entonces
15            $b = 0$ 
16         Si No
17            $C = \text{HallarCota}(e_h)$ 
18           Si  $C < v$  entonces
19              $\xi = \xi + e_h$ 
20             Si  $x_h > x_b$  ó  $x_h = x_b$  y  $C_h < C_b$  entonces
21                $e_b = e_h$ 
22                $C_b = C_h$ 
23                $x_b = x_h$ 
24 Retornar  $\xi$ 
```

---

Los procesos implementados en **CrearDescendientes** (Procedimiento 2) se mantienen en la función **CrearDescendientes2** (Procedimiento 10):

## Procedimiento por separación aleatorizado

---

- ▶ generación de movimientos (líneas 2 a 5),
- ▶ comprobación de que un estado objetivo ha sido encontrado (líneas 7 a 11),
- ▶ y poda por cota (líneas 17 a 19);

y se añaden:

- ▶ **Poda por progreso:** en la línea 13 se determina el indicador de progreso  $x_h$  calculado para el estado  $e_h$ . Si este valor supera al valor de indicador de progreso máximo  $X$  establecido para la iteración en curso, entonces el estado  $e_h$  es podado (ver que en ese caso ya no se ejecutará la línea 19 donde dicho estado se puede incorporar al árbol de enumeración).

Por otro lado, si se ha realizado una poda por progreso, se asigna cero a la variable  $b$  (que define la finalización del proceso iterativo) indicando que este proceso debe continuar. Si en una iteración dada, al menos un estado es podado por progreso las iteraciones continúan. En el caso en que no se ha producido poda por progreso,  $b$  mantendrá el valor 1 recibido del procedimiento invocador.

- ▶ **Determinación de mejor estado:** en la línea 20 se aplica el criterio de mejor estado comparando el estado  $e_h$  recientemente hallado con el mejor estado  $e_b$  encontrado hasta el momento. Si el estado  $e_h$  mejora a  $e_b$  (resulta, por ello, más prometedor) entonces:
  - ▷  $e_b$  toma el valor  $e_h$  (línea 21),
  - ▷  $C_b$  toma el valor  $C_h$  (línea 22)
  - ▷ y  $x_b$  toma el valor  $x_h$  (línea 23).



# Capítulo 8

## Experiencia computacional

### 8.1 Introducción

La eficiencia de los procedimientos de resolución del capítulo 7 y de los procedimientos de acotación de los capítulos 4 y 5, frente a la variación de los datos del problema, es evaluada mediante una experiencia computacional.

La experiencia computacional comprende:

- ▶ la definición de un conjunto de ejemplares del problema (apartados 8.2 y 8.3),
- ▶ la definición de los distintos procedimientos a ser ensayados (apartado 8.4) y los parámetros que éstos emplean (apartado 8.5),
- ▶ y la definición de índices de desempeño para evaluar la eficiencia de los distintos procedimientos (apartado 8.6).

El análisis de los resultados obtenidos de la resolución de los ejemplares por los distintos procedimientos se realiza en los apartados (8.7 y 8.9).

### 8.2 Variantes de ejemplares

La definición del problema en el capítulo 2 establece, para el problema, un conjunto de parámetros como:

- ▶ *número de máquinas,*
- ▶ *número de piezas,*
- ▶ *capacidad de los buffers,*
- ▶ *tiempos de viaje,*
- ▶ *tiempos de carga y descarga,*

► y *tiempos de proceso* de las piezas.

A partir de definir estos parámetros, surgen distintas variantes, las cuales serán definidas a continuación.

### 8.2.1 Número de máquinas

Con respecto al número de máquinas  $m$  se analizarán cuatro variantes codificadas **M2**, **M3**, **M4** y **M5**, las cuales se corresponden con los valores de  $m$  indicados en la tabla 8.1.

<i>Variante</i>	$m$
<b>M2</b>	2
<b>M3</b>	3
<b>M4</b>	4
<b>M5</b>	5

Tabla 8.1: Codificación de las variantes por el número de máquinas.

### 8.2.2 Número de piezas

Con respecto al número de piezas  $n$  se analizarán cuatro variantes, codificadas **P05**, **P10**, **P15** y **P20**, las cuales se corresponden con los valores de  $n$  indicados en la tabla 8.2.

<i>Variante</i>	$n$
<b>P05</b>	5
<b>P10</b>	10
<b>P15</b>	15
<b>P20</b>	20

Tabla 8.2: Codificación de las variantes por el número de piezas.

### 8.2.3 Capacidad de los buffers

Con respecto al número de piezas máximo  $b_i$  que se permite alojar en el buffer  $B_i$  ( $i = 1, \dots, m - 1$ ) se definen tres variantes, codificadas **B1**, **B2** y **B3**, las cuales se corresponden con los valores de  $b_i$  indicados en la tabla 8.3.

<i>Variante</i>	$b_i$
<b>B1</b>	0
<b>B2</b>	$\text{entmed}(m)$
<b>B3</b>	$m$

Tabla 8.3: Codificación de las variantes por la capacidad de los buffers.

### 8.2.4 Tiempos de viaje

Para la determinación de los tiempos de viaje, con el fin de definir tiempos realistas, se propone un modelo en el que son incluidos los parámetros que tienen influencia en estos tiempos. En el modelo de generación de los tiempos de viaje se consideran:

- ▶ El movimiento no uniforme del robot: éste se compone de los siguientes movimientos:
  - ▷ un movimiento acelerado cuando el robot parte de una estación,
  - ▷ un movimiento desacelerado cuando el robot llega a una estación,
  - ▷ y un movimiento de velocidad constante en medio.
- ▶ La influencia de la distancia entre las estaciones: a mayor distancia el tiempo de viaje se hace mayor.
- ▶ La influencia del peso de cada pieza en los tiempos del robot cuando éste se acelera o se desacelera.

Un modelo para los tiempos de viaje que cumple con las características indicadas en los puntos anteriores, se define a continuación.

**Velocidad máxima del robot.** La velocidad de desplazamiento del robot máxima se define:

$$v_{max} = 1 \text{ m/s}$$

**Distancia entre estaciones.** Se denota  $D_{g,h}$  la distancia entre dos estaciones  $E_g$  y  $E_h$ . La distancia entre todo par de estaciones consecutivas  $E_g$  y  $E_{g+1}$  se define como un valor aleatorio  $D_{g,g+1}$  acotado inferiormente por un valor  $D_{min}$  y superiormente por un valor  $D_{max}$ . La distancia  $D_{g,g+1}$  se determina por la función **aleatorio()** como se muestra en la expresión 8.1.

$$D_{g,g+1} = D_{g+1,g} = \text{aleatorio} (D_{min}, D_{max}) \quad (8.1)$$

donde:

$$D_{min} = 0,6 \text{ m} \quad \text{y} \quad D_{max} = 1,2 \text{ m}$$

Una vez determinadas las distancias entre estaciones consecutivas mediante 8.1, las distancias entre estaciones no consecutivas se determinan mediante:

$$D_{g,h} = D_{h,g} = \sum_{i=\min(g,h)}^{\max(g,h)-1} D_{i,i+1}$$

**Distancia de movimiento acelerado.** En el modelo empleado para la generación de los tiempos de viaje, la influencia debida al peso de las piezas en los tiempos de viaje se introduce mediante la distancia de movimiento acelerado  $d_j$ , definida para cada pieza  $P_j$ . Ésta se define como la distancia que debe recorrer el robot antes de alcanzar su velocidad máxima cuando éste se acelera partiendo del reposo, e, inversamente, la distancia que el robot debe recorrer antes de alcanzar el reposo cuando éste se desacelera partiendo de la velocidad máxima (como se ha indicado, estas distancias pueden depender del peso de la pieza).

La distancia de movimiento acelerado  $d_j$  asociado a la pieza  $P_j$  se define como un valor aleatorio acotado inferiormente por un valor  $d_{min}$  y superiormente por un valor  $d_{max}$ . La distancia  $d_j$  se determina por la función **aleatorio()** como se muestra en la expresión 8.2.

$$d_j = \text{aleatorio}(d_{min}, d_{max}) \quad (8.2)$$

donde:

$$d_{min} = 0,1 \text{ m} \quad \text{y} \quad d_{max} = 0,6 \text{ m}$$

Para el robot sin pieza se define la distancia  $d_0$ :

$$d_0 = d_{min}/2$$

**Tiempo de viaje.** El desplazamiento del robot entre dos estaciones comprenderá tres desplazamientos:

- ▶ El robot parte de un punto  $x_1$  con velocidad nula y acelera con aceleración constante hasta alcanzar una velocidad máxima  $v'$  en un punto  $x'_1$ .
- ▶ A partir de  $x'_1$  se desplaza a velocidad constante  $v'$  hasta el punto  $x'_2$  recorriendo una distancia  $x'_2 - x'_1$  (nula en la variante particular en que  $x'_2 = x'_1$ ).
- ▶ A partir de  $x'_2$  desacelera con desaceleración constante desde la velocidad  $v'$  hasta el reposo en  $x_2$ .

Se supone la igualdad de las distancias:  $x'_1 - x_1 = x_2 - x'_2$ , y por tanto la aceleración del primer desplazamiento coincide con la desaceleración del tercer desplazamiento en valor absoluto.

Sean  $D = x_2 - x_1$  la distancia a ser recorrida por el robot y  $d$  la distancia de movimiento acelerado, son posibles dos casos:

- ▶ Si  $D \geq 2d$  entonces el robot verifica la evolución de su velocidad  $v$  con respecto a su posición  $x$  como la mostrada en la figura 8.1. Allí se observa que el robot

logra alcanzar su velocidad máxima  $v_{max}$  en el punto  $x = x'_1$  y la mantiene hasta el punto  $x = x'_2$ . Verifica, además, la igualdad:

$$x'_1 - x_1 = x_2 - x'_2 = d \quad (8.3)$$

Bajo esta condición el tiempo total  $t$  ocupado en todo el recorrido  $D$ , comprendido entre los puntos  $x_1$  y  $x_2$ , vale:

$$t = \frac{D + 2d}{v_{max}} \quad (8.4)$$

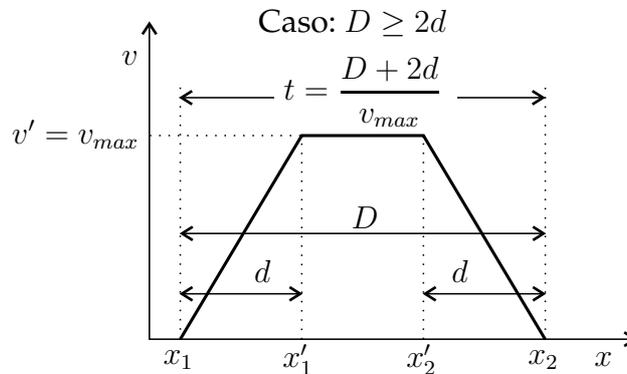


Figura 8.1: Velocidad del robot  $v$  en función de su posición  $x$ . Caso en que  $D \geq 2d$ .

- Si  $D < 2d$  entonces el robot verifica la evolución de su velocidad  $v$  con respecto a su posición  $x$ , como la mostrada en la figura 8.2. Allí se observa que el robot no logra alcanzar su velocidad máxima  $v_{max}$  y, como consecuencia, que no realiza desplazamiento a velocidad constante por lo que los puntos  $x'_1$  y  $x'_2$  son coincidentes. El desplazamiento se forma por un tramo de aceleración constante entre los puntos  $x = x_1$  y  $x = x'_1 = x'_2$  y un tramo de desaceleración constante entre el último punto y el punto  $x = x_2$ . En este caso se verifica la igualdad:

$$x'_1 - x_1 = x_2 - x'_2 = D/2 \quad (8.5)$$

Bajo esta condición el tiempo total  $t$  ocupado en todo el recorrido  $D$  comprendido entre los puntos  $x_1$  y  $x_2$  vale:

$$t = \frac{2\sqrt{2Dd}}{v_{max}} \quad (8.6)$$

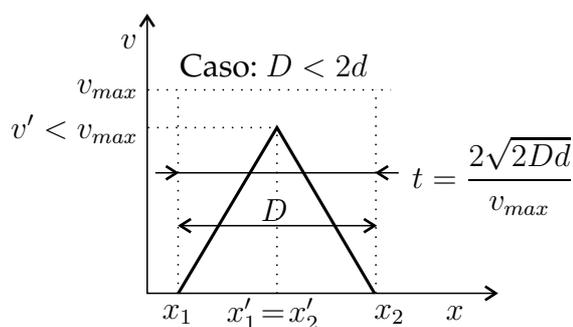


Figura 8.2: Velocidad del robot  $v$  en función de su posición  $x$ . Caso en que  $D < 2d$ .

Dadas la distancia total a recorrer por el robot  $D$  entre dos puntos y la distancia que éste recorre en movimiento acelerado y desacelerado  $d$ , la expresión final del tiempo  $t$  de desplazamiento entre dichos puntos resulta de reunir las expresiones 8.4 y 8.6 en una nueva expresión:

$$t = \begin{cases} \frac{D + 2d}{v_{max}} & \text{si } D \geq 2d \\ \frac{2\sqrt{2Dd}}{v_{max}} & \text{en otro caso} \end{cases} \quad (8.7)$$

**Tiempo de viaje de referencia  $t_r$ .** El tiempo de viaje de referencia  $t_r$  consiste en una magnitud temporal de valor constante a partir de la cual se determinarán los tiempos de carga y descarga (apartado 8.2.5) y los tiempos de proceso (apartado 8.2.6).

$$t_r = \frac{1}{2} t \quad (8.8)$$

donde el tiempo  $t$  se determina mediante la expresión 8.7. Por otro lado, la distancia de movimiento acelerado  $d$  se halla mediante la expresión 8.9, como el promedio de los valores extremos  $d_{min}$  y  $d_{max}$ :

$$d = \frac{d_{max} + d_{min}}{2} \quad (8.9)$$

y la distancia entre estaciones consecutivas  $D$  se halla como el promedio de los valores extremos  $D_{min}$  y  $D_{max}$ :

$$D = \frac{D_{max} + D_{min}}{2} \quad (8.10)$$

El factor  $1/2$  de la expresión 8.8 permite que, en los datos del ejemplar, puedan aparecer tiempos de carga, descarga y proceso menores a los tiempos de viaje del robot.

### 8.2.5 Tiempos de carga y descarga

El tiempo de carga  $t_{i,j}^c$  empleado en cargar la pieza  $P_j$  en la estación  $E_i$  se define como un valor aleatorio acotado inferiormente por un valor  $t_{min}^{cd}$  y superiormente por un valor  $t_{max}^{cd}$ . El tiempo  $t_{i,j}^c$  se determina con la función **aleatorio()** como se muestra en la expresión 8.11.

$$t_{i,j}^c = \text{aleatorio} (t_{min}^{cd}, t_{max}^{cd}) \quad (8.11)$$

y, similarmente, el tiempo de descarga  $t_{i,j}^d$  empleado en descargar la pieza  $P_j$  de la estación  $E_i$  se define como un valor aleatorio acotado inferiormente por un valor  $t_{min}^{cd}$  y superiormente por un valor  $t_{max}^{cd}$ . El tiempo  $t_{i,j}^d$  se determina con la función **aleatorio()** como se muestra en la expresión 8.12.

$$t_{i,j}^d = \text{aleatorio} (t_{min}^{cd}, t_{max}^{cd}) \quad (8.12)$$

Una vez definido el valor del tiempo de viaje de referencia  $t_r$  (definido en el apartado 8.2.4), los tiempos de carga y descarga se pueden definir respecto de este tiempo. Se propone la separación de la experiencia computacional en dos variantes:

- ▶ Ejemplares con tiempos de carga y descarga pequeños o del mismo orden de magnitud que los tiempos de viaje. Esta variante del problema se codifica **CD1** y en ella se definen:

$$t_{min}^{cd} = 0,5 t_r \quad \text{y} \quad t_{max}^{cd} = 4 t_r \quad (8.13)$$

- ▶ Ejemplares con tiempos de carga y descarga grandes con respecto de los tiempos de viaje. Esta variante del problema se codifica **CD2** y en ella se definen:

$$t_{min}^{cd} = 4 t_r \quad \text{y} \quad t_{max}^{cd} = 32 t_r \quad (8.14)$$

Las dos variantes se reúnen en la tabla 8.4.

Variante	$t_{min}^{cd}$	$t_{max}^{cd}$
<b>CD1</b>	$0,5 t_r$	$4 t_r$
<b>CD2</b>	$4 t_r$	$32 t_r$

Tabla 8.4: Variantes por los tiempos de carga y descarga en función de  $t_r$ .

### 8.2.6 Tiempos de proceso

El tiempo  $t_{i,j}^p$  empleado en procesar la pieza  $P_j$  en la máquina  $M_i$  se define como un valor aleatorio acotado inferiormente por  $t_{min}^p$  y superiormente por  $t_{max}^p$ , el cual es determinado mediante la función **aleatorio()** según la expresión 8.15.

$$t_{i,j}^p = \text{aleatorio} (t_{min}^p, t_{max}^p) \quad (8.15)$$

Se definen dos variantes, con relación a los tiempos de carga y descarga mínimo  $t_{min}^{cd}$  y máximo  $t_{max}^{cd}$ :

- ▶ Ejemplares donde los tiempos de proceso son pequeños o del mismo orden de magnitud frente a los tiempos de carga y descarga. Esta variante del problema se codifica **PR1** y en ella se definen:

$$t_{min}^p = 0,5 t_{min}^{cd} \quad \text{y} \quad t_{max}^p = 4 t_{max}^{cd}$$

- ▶ Ejemplares donde los tiempos de proceso en un rango del mismo orden de magnitud que los tiempos de carga y descarga. Esta variante del problema se codifica **PR2** y en ella se definen:

$$t_{min}^p = 4 t_{min}^{cd} \quad \text{y} \quad t_{max}^p = 32 t_{max}^{cd}$$

Las dos variantes se reúnen en la tabla 8.5.

<i>Variante</i>	$t_{min}^p$	$t_{max}^p$
<b>PR1</b>	$0,5 t_{min}^{cd}$	$4 t_{max}^{cd}$
<b>PR2</b>	$4 t_{min}^{cd}$	$32 t_{max}^{cd}$

Tabla 8.5: Variantes por los tiempos de proceso en función de  $t_{min}^{cd}$  y  $t_{max}^{cd}$ .

Por sustitución de  $t_{min}^{cd}$  y  $t_{max}^{cd}$  por las relaciones 8.13 y 8.14, éstos quedan expresados en función de  $t_r$  como se muestra en la tabla 8.6.

<i>Variante</i>	<i>Variante</i>	$t_{min}^p$	$t_{max}^p$
<b>CD1</b>	$t_{min}^{cd} = 0,5 t_r$	<b>PR1</b>	$0,25 t_r$
	$t_{max}^{cd} = 4 t_r$	<b>PR2</b>	$2 t_r$
<b>CD2</b>	$t_{min}^{cd} = 4 t_r$	<b>PR1</b>	$2 t_r$
	$t_{max}^{cd} = 32 t_r$	<b>PR2</b>	$16 t_r$

Tabla 8.6: Variantes por los tiempos de proceso en función de  $t_r$ .

### 8.3 Ejemplares para la experiencia computacional

La experiencia computacional se realiza sobre un conjunto  $\mathcal{E}$ , formado por  $N_{\mathcal{E}}$  ejemplares del problema.

Se propone la siguiente combinación de variantes:

- ▶ 4 variantes por el número de máquinas (apartado 8.2.1): **M1**, **M2**, **M3** y **M4**;
- ▶ 4 variantes por el número de piezas (apartado 8.2.2): **P05**, **P10**, **P15** y **P20**;
- ▶ 2 variantes por la capacidad de los buffers: (apartado 8.2.3): **B1**, **B2** y **B3**;
- ▶ 2 variantes por los tiempos de carga y descarga (apartado 8.2.5): **CD1** y **CD2**;
- ▶ 2 variantes por los tiempos de proceso (apartado 8.2.6): **PR1** y **PR2**;

dando lugar a un total de 192 variantes. Para cada variante se generan 10 ejemplares dando lugar a un total de  $N_{\mathcal{E}} = 1920$  ejemplares.

### 8.4 Procedimientos

Un conjunto de ejemplares  $\mathcal{E}$  será resuelto por cada uno de los procedimientos a ser evaluados. Entre estos procedimientos debe diferenciarse entre procedimientos de resolución y procedimientos de acotación.

**Procedimientos de resolución.** Se dispone de un conjunto  $R$  formado por  $n_r$  procedimientos de resolución  $R_k$  ( $k = 1, \dots, n_r$ ). Los procedimientos resuelven cada ejemplar del problema dentro de un tiempo límite establecido. Con el fin de hacer comparables los resultados, este tiempo se ha definido constante para todo procedimiento de resolución y para todo ejemplar, en un valor de 30 minutos. La tabla 8.7 muestra la correspondencia de los elementos  $R_k$  de  $R$  con los procedimientos del capítulo 7 donde  $n_r = 4$ . Así mismo, para simplificar la notación en las distintas tablas de resultados, se asocia a cada procedimiento una codificación, como previamente se ha hecho con las variantes de problema mostrados en el apartado 8.2.

$R_1$	Procedimiento branch and bound del apartado 7.2, codificado <b>BB</b>
$R_2$	Procedimiento branch and bound con poda heurística del apartado 7.3, codificado <b>BBP</b>
$R_3$	Procedimiento por separación del apartado 7.4, codificado <b>SEP</b>
$R_4$	Procedimiento por separación aleatorizado del apartado 7.5, codificado <b>SEPA</b>

Tabla 8.7: Procedimientos de resolución en  $R$ .

Como resultado de aplicar el procedimiento  $R_k$  al ejemplar  $\mathcal{E}_i$  ( $i = 1, \dots, N_{\mathcal{E}}$ ) se obtiene un resultado o valor de la solución  $\mathcal{S}_i^k$ . Se define *mejor solución*  $\mathcal{S}_i$ , la solución de menor

valor obtenida como resultado de ejecutar el conjunto de procedimientos  $R$  al ejemplar  $\mathcal{E}_i$ . Se expresa como:

$$\mathcal{S}_i = \min_{k=1}^{n_r} \mathcal{S}_i^k \quad (8.16)$$

**Procedimientos de acotación.** Se dispone de un conjunto  $A$  formado por  $n_a$  procedimientos de acotación  $A_k$  ( $k = 1, \dots, n_a$ ). Como resultado de aplicar el procedimiento de acotación  $A_k$  al ejemplar  $\mathcal{E}_i$  se obtiene la cota  $\mathcal{C}_i^k$ . La tabla 8.8 muestra la correspondencia de la notación de los elementos  $A_k$  de  $A$  con los procedimientos vistos en los capítulos 4 y 5 (de aquí que  $n_a = 2$ ).

$A_1$	Cota basada en los tiempos de las máquinas (capítulo 4)
$A_2$	Cota basada en los tiempos del robot (capítulo 5)

Tabla 8.8: Procedimientos de resolución en  $A$ .

Se define *mejor cota*  $\mathcal{C}_i$  del ejemplar  $\mathcal{E}_i$ , el mayor valor obtenido como resultado de ejecutar los procedimientos de acotación  $A_k$  para todos los valores  $k = 1, \dots, n_a$  al estado inicial  $e_0$ , caracterizado por todas las piezas y el robot en la estación de entrada, al ejemplar  $\mathcal{E}_i$ . Se expresa como:

$$\mathcal{C}_i = \max_{k=1}^{n_a} \mathcal{C}_i^k \quad (8.17)$$

## 8.5 Parámetros de los procedimientos

La ejecución de los procedimientos explicados en el capítulo 7 requiere, previamente, dar valor a los distintos parámetros empleados por ellos.

Para los distintos procedimientos de resolución, con el fin de hacer comparables los resultados, se define un tiempo de ejecución máximo  $\mathcal{T}_{max}$ . Es decir, el tiempo  $\mathcal{T}$  empleado por un procedimiento dado en resolver un ejemplar determinado estará acotado por:

$$0 \leq \mathcal{T} \leq \mathcal{T}_{max}$$

En donde  $\mathcal{T}_{max}$  se define como:

$$\mathcal{T}_{max} = 30 \text{ minutos} = 1800 \text{ segundos}$$

en donde, este tiempo, es el mismo para todos los procedimientos de resolución y para todos los ejemplares.

Además del tiempo de ejecución  $\mathcal{T}_{max}$ , como se vio en el capítulo 7, cada procedimiento dispone de parámetros propios. Estos parámetros son especificados como se explica a continuación.

### 8.5.1 Parámetros del procedimiento BB

El procedimiento **BB** no emplea parámetros adicionales.

### 8.5.2 Parámetros del procedimiento BBP

En base a pruebas preliminares, para el procedimiento **BBP**, los parámetros  $w_{ini}$  y  $w_{inc}$  reciben los valores siguientes:

$$w_{ini} = 2, \quad w_{inc} = 1$$

### 8.5.3 Parámetros del procedimiento SEP

Para el procedimiento **SEP**, los parámetros a definirse, consisten en los tiempos máximos asignados a cada una de las fases que conforman el procedimiento. Se definen los siguientes tiempos.

- El tiempo de resolución de la primera fase (resolución del método relax and fix),  $\mathcal{T}_{SEP}^1$ , está acotado por:

$$0 \leq \mathcal{T}_{SEP}^1 \leq \mathcal{T}_{SEP,max}^1$$

en donde:

$$\mathcal{T}_{SEP,max}^1 = 90 \text{ segundos}$$

Sea  $\mathcal{T}_{SEP,k}^1$  el tiempo empleado en la iteración  $k$  de las  $n - 1$  iteraciones<sup>1</sup> del procedimiento relax and fix, se cumple la igualdad 8.18.

$$\mathcal{T}_{SEP}^1 = \sum_{k=1}^{n-1} \mathcal{T}_{SEP,k}^1 \quad (8.18)$$

El tiempo de cada iteración  $\mathcal{T}_{SEP,k}^1$  está acotado por:

$$0 \leq \mathcal{T}_{SEP,k}^1 \leq \mathcal{T}_{SEP,k,max}^1$$

---

<sup>1</sup>Recuérdese que la asignación de la pieza en la posición  $n$  de la secuencia de piezas, dado que sólo queda una pieza a asignar y sólo una posición, es obligada, por lo que no se requiere realizar una iteración para determinarla.

donde  $\mathcal{T}_{SEP,k,max}^1$  se determina como:

$$\mathcal{T}_{SEP,k,max}^1 = \begin{cases} \frac{\mathcal{T}_{SEP,max}^1}{n-1} & \text{si } k = 1 \\ \frac{\mathcal{T}_{SEP,max}^1 - \sum_{i=1}^{k-1} \mathcal{T}_{SEP,i}^1}{n-k} & \text{si } 2 \leq k \leq n-1 \end{cases} \quad (8.19)$$

De la expresión 8.19 se ve que el tiempo asignado a cada iteración consiste en el tiempo que resta hasta cumplir el tiempo  $\mathcal{T}_{SEP,max}^1$ , dividido por el número de iteraciones que resta realizarse.

- El tiempo de resolución de la segunda fase (resolución de programa lineal mixto),  $\mathcal{T}_{SEP}^2$ , está acotado por:

$$0 \leq \mathcal{T}_{SEP}^2 \leq \mathcal{T}_{SEP,max}^2$$

en donde,

$$\mathcal{T}_{SEP,max}^2 = 180 \text{ segundos} - \mathcal{T}_{SEP}^1 \quad (8.20)$$

En la expresión 8.20 se observa que, a las dos primeras fases, donde se determina la secuencia de piezas predefinida  $\bar{\Pi}$ , se les asigna un tiempo máximo del 10 % del tiempo  $\mathcal{T}_{max}$  asignado a todo el procedimiento.

- El tiempo de resolución de la tercera y última fase (determinación de la secuencia de movimientos del robot),  $\mathcal{T}_{SEP}^3$ , está acotado por:

$$0 \leq \mathcal{T}_{SEP}^3 \leq \mathcal{T}_{SEP,max}^3$$

en donde,

$$\mathcal{T}_{SEP,max}^3 = \mathcal{T}_{max} - \mathcal{T}_{SEP}^1 - \mathcal{T}_{SEP}^2 \quad (8.21)$$

Es decir que, de acuerdo con la expresión 8.21,  $\mathcal{T}_{SEP,max}^3$  es el tiempo que completa el tiempo total  $\mathcal{T}_{max}$ .

### 8.5.4 Parámetros del procedimiento SEPA

El procedimiento SEPA presenta diversos parámetros, analizados a continuación.

#### Parámetros que definen los tiempos de las distintas fases del procedimiento.

Como en el procedimiento SEP, el procedimiento SEPA, emplea una secuencia predefinida basada en resolución de un modelo de programación lineal. Se definen en este procedimiento los mismos parámetros de tiempo y son establecidos los mismos valores que fueron establecidos para el procedimiento anterior. Estos parámetros son los siguientes.

## Parámetros de los procedimientos

---

- El tiempo de resolución de la primera fase (resolución del método relax and fix),  $\mathcal{T}_{SEPA}^1$ , está acotado por:

$$0 \leq \mathcal{T}_{SEPA}^1 \leq \mathcal{T}_{SEPA,max}^1$$

en donde:

$$\mathcal{T}_{SEPA,max}^1 = 90 \text{ segundos}$$

Sea  $\mathcal{T}_{SEPA,k}^1$  el tiempo empleado en la iteración  $k$  del procedimiento relax and fix, se cumple la igualdad 8.22.

$$\mathcal{T}_{SEPA}^1 = \sum_{k=1}^{n-1} \mathcal{T}_{SEPA,k}^1 \quad (8.22)$$

El tiempo de cada iteración  $\mathcal{T}_{SEPA,k}^1$  está acotado por:

$$0 \leq \mathcal{T}_{SEPA,k}^1 \leq \mathcal{T}_{SEPA,k,max}^1$$

y se determina como:

$$\mathcal{T}_{SEPA,k,max}^1 = \begin{cases} \frac{\mathcal{T}_{SEPA,max}^1}{n-1} & \text{si } k = 1 \\ \frac{\mathcal{T}_{SEPA,max}^1 - \sum_{i=1}^{k-1} \mathcal{T}_{SEPA,i}^1}{n-k} & \text{si } 2 \leq k \leq n-1 \end{cases}$$

- El tiempo de resolución de la segunda fase (resolución de programa lineal mixto),  $\mathcal{T}_{SEPA}^2$ , está acotado por:

$$0 \leq \mathcal{T}_{SEPA}^2 \leq \mathcal{T}_{SEPA,max}^2$$

en donde,

$$\mathcal{T}_{SEPA,max}^2 = 180 \text{ segundos} - \mathcal{T}_{SEPA}^1 \quad (8.23)$$

- El tiempo de resolución de la tercera y última fase (determinación de la secuencia de movimientos del robot),  $\mathcal{T}_{SEPA}^3$ , está acotado por:

$$0 \leq \mathcal{T}_{SEPA}^3 \leq \mathcal{T}_{SEPA,max}^3$$

en donde,

$$\mathcal{T}_{SEPA,max}^3 = \mathcal{T}_{max} - \mathcal{T}_{SEPA}^1 - \mathcal{T}_{SEPA}^2 \quad (8.24)$$

**Parámetros empleados en la determinación de la variable  $\gamma$ .** Durante la segunda fase del procedimiento SEPA, la variable  $\gamma$  se incrementa en la magnitud  $\gamma_{inc}$  a intervalos de tiempo regulares de magnitud  $\mathcal{T}_{inc}$ , partiendo de un cierto valor inicial  $\gamma_{ini}$ . Para los parámetros  $\gamma_{ini}$  y  $\gamma_{inc}$  se adoptan los siguientes valores.

$$\gamma_{ini} = 0, \quad \gamma_{inc} = 0,1$$

Como se explicó en el apartado 7.5.1, el valor  $\gamma_{ini} = 0$  tiene por finalidad permitir una gran diversidad de secuencias aleatorizadas  $\bar{\nabla}$  al inicio del procedimiento, mientras que ésta se va reduciendo a medida que el valor de  $\gamma$  es incrementado.

La determinación del tiempo  $\mathcal{T}_{inc}$  se facilita sensiblemente si se introduce el parámetro  $\gamma_{fin}$ , correspondiente al valor que se desea sea alcanzado por la variable  $\gamma$  al final de todo el proceso. Un valor adecuado de  $\gamma_{fin}$ , en base a pruebas preliminares, es:

$$\gamma_{fin} = 6$$

Una vez definidos estos parámetros es posible hallar el número de veces  $N_{inc}$  que, a lo largo del procedimiento, se debería realizar el incremento del valor de la variable  $\gamma$ , como:

$$N_{inc} = \frac{\gamma_{fin} - \gamma_{ini}}{\gamma_{inc}} \quad (8.25)$$

Luego, el tiempo  $\mathcal{T}_{inc}$  transcurrido entre cada incremento del valor de la variable  $\gamma$ , se determina como:

$$\mathcal{T}_{inc} = \frac{\mathcal{T}_{SEPA,max}^3}{N_{inc} + 1} \quad (8.26)$$

Ahora, sustituyendo la expresión 8.25 en la expresión 8.26, se obtiene la expresión 8.27.

$$\mathcal{T}_{inc} = \frac{\gamma_{inc}}{\gamma_{fin} - \gamma_{ini} + \gamma_{inc}} \mathcal{T}_{SEP,max}^3 \quad (8.27)$$

**Parámetros empleados en la determinación de la variable  $X$ .** Para los parámetros que determinan el valor en curso del indicador de progreso máximo empleado en la determinación de la secuencia de movimientos del robot, se definen los siguientes valores:

$$X_{ini} = 1, \quad X_{inc} = 2$$

Estos valores son pequeños para permitir que la determinación de la secuencia de movimientos del robot sea un proceso rápido y facilite que puedan ser generadas un elevado número de secuencias de piezas aleatorizadas.

## 8.6 Eficiencia de los procedimientos

La eficiencia de los procedimientos de resolución y de los procedimientos de acotación es evaluada mediante distintos índices de desempeño, que se definen a continuación.

**Error relativo porcentual de las soluciones referido a la mejor cota.** El *error relativo porcentual de las soluciones referido a la mejor cota*,  $r_1^k$ , de un procedimiento de resolución  $R_k$  aplicado a un conjunto de ejemplares  $\mathcal{E}$  consiste en la media aritmética de los errores relativos porcentuales calculados para cada solución  $S_i^k$  obtenida, con respecto a la mejor cota  $C_i$  del ejemplar  $\mathcal{E}_i$ . Ésta se determina mediante la expresión 8.28.

$$r_1^k = \frac{100}{N_{\mathcal{E}}} \sum_{i=1}^{N_{\mathcal{E}}} \frac{S_i^k - C_i}{C_i} \quad (8.28)$$

**Error relativo porcentual de las soluciones referido a la mejor solución** El *error relativo porcentual de las soluciones referido a la mejor solución*,  $r_2^k$ , de un procedimiento de resolución  $R_k$  aplicado a un conjunto de ejemplares  $\mathcal{E}$  consiste en la media aritmética de los errores relativos porcentuales calculados para cada solución  $S_i^k$  obtenida, con respecto a la mejor solución  $S_i$  del ejemplar  $\mathcal{E}_i$ . Ésta se determina mediante la expresión 8.29.

$$r_2^k = \frac{100}{N_{\mathcal{E}}} \sum_{i=1}^{N_{\mathcal{E}}} \frac{S_i^k - S_i}{S_i} \quad (8.29)$$

**Porcentual de casos de mejor solución hallada.** Sea  $n_s^k$  el número de veces que para las  $N_{\mathcal{E}}$  soluciones  $S_i^k$  obtenidas, el procedimiento de resolución  $R_k$  iguala al valor de la mejor solución  $S_i$ . Entonces, el *porcentual de casos de mejor solución hallada*,  $\eta_s^k$ , de un procedimiento de resolución  $R_k$  aplicado a un conjunto de ejemplares  $\mathcal{E}$  consiste en el número porcentual de veces que, mediante el procedimiento  $R_k$ , se obtiene la mejor solución (del conjunto de procedimientos de resolución  $R$ ) con relación al número total de ejemplares  $N_{\mathcal{E}}$ . Ésta se determina mediante la expresión 8.30.

$$\eta_s^k = 100 \frac{n_s^k}{N_{\mathcal{E}}} \quad (8.30)$$

**Porcentual de casos de solución óptima hallada.** Recuérdese que los procedimientos **BB** y **BBP** son exactos y por ello, eventualmente, pueden alcanzar soluciones óptimas. El primero, cuando el árbol de enumeración se ha explorado por completo

antes del tiempo límite dado al ejemplar, y el segundo, cuando, dentro del tiempo prefijado, una de las ejecuciones del branch and bound dentro del bucle principal termina sin producirse poda heurística. En tal caso el valor de la mejor solución  $S_i$  es, a la vez, solución óptima. Por otro lado, se puede constatar qué soluciones procedentes de los procedimientos de resolución **SEP** y **SEPA** son óptimas, sólo cuando los valores de estas soluciones coinciden con el valor óptimo obtenido y verificado con los procedimientos **BB** o **BBP**.

Sea  $n_o^k$  el número de veces que para las  $N_{\mathcal{E}}$  soluciones  $S_i^k$  obtenidas, el procedimiento de resolución  $R_k$  iguala al valor de la mejor solución  $S_i$  cuando ésta es óptima. Entonces, el *porcentaje de casos de solución óptima hallada*,  $\eta_o^k$ , de un procedimiento de resolución  $R_k$  aplicado a un conjunto de ejemplares  $\mathcal{E}$  consiste en el número porcentual de veces que, mediante el procedimiento  $R_k$ , se obtiene una solución óptima (del conjunto de procedimientos de resolución  $R$ ) con relación al número total de ejemplares  $N_{\mathcal{E}}$ . Ésta se determina mediante la expresión 8.31.

$$\eta_o^k = 100 \frac{n_o^k}{N_{\mathcal{E}}} \quad (8.31)$$

**Error relativo porcentual de la cota referido a la mejor cota** El *error relativo porcentual de la cota referido a la mejor cota*,  $r_3^k$ , de un procedimiento de acotación  $A_k$  aplicado a un conjunto de ejemplares  $\mathcal{E}$  consiste en la media aritmética de los errores relativos porcentuales calculados para cada cota  $C_i^k$  obtenida, con respecto a la mejor cota  $C_i$  del ejemplar  $\mathcal{E}_i$ . Ésta se determina mediante la expresión 8.32.

$$r_3^k = \frac{100}{N_{\mathcal{E}}} \sum_{i=1}^{N_{\mathcal{E}}} \frac{C_i - C_i^k}{C_i} \quad (8.32)$$

**Porcentaje de casos de mejor cota hallada** Sea  $n_c^k$  el número de veces que para las  $N_{\mathcal{E}}$  cotas  $C_i^k$  obtenidas, el procedimiento de acotación  $A_k$  iguala al valor de la mejor cota  $C_i$  del ejemplar  $\mathcal{E}_i$ . Entonces, el *porcentaje de casos de mejor cota hallada*,  $\eta_c^k$ , de un procedimiento de acotación  $A_k$  aplicado a un conjunto de ejemplares  $\mathcal{E}$  consiste en el número porcentual de veces que, mediante el procedimiento  $A_k$ , se obtiene la mejor cota  $C_i$  con relación al número total de ejemplares  $N_{\mathcal{E}}$ . Ésta se determina mediante la expresión 8.33.

$$\eta_c^k = 100 \frac{n_c^k}{N_{\mathcal{E}}} \quad (8.33)$$

## 8.7 Análisis de los procedimientos de resolución

En este apartado se evalúa el desempeño de los procedimientos de resolución para las distintas variantes de capacidad de los buffers (**B1**, **B2** y **B3**), para cada variante de tiempos de proceso (**PR1** y **PR2**) y cada variante de tiempos de carga y descarga (**CD1**

y **CD2**), en función del número de piezas  $n$  (**P01** a **P20**) y del número de máquinas  $m$  (**M2** a **M5**) de los distintos ejemplares.

Para medir la eficiencia de los procedimientos de resolución se introducen los índices de desempeño (definidos en el apartado 8.6) siguientes:

- ▶ *error relativo porcentual de las soluciones referido a la mejor cota*  $r_1^k$  (ecuación 8.28),
- ▶ *error relativo porcentual de las soluciones referido a la mejor solución*  $r_2^k$  (ecuación 8.29),
- ▶ *y porcentual de casos de mejor solución hallada*  $\eta_s^k$  (ecuación 8.30).

### 8.7.1 Análisis de resultados para la variante de buffers nulos (variante B1)

**Variante PR1-CD1.** El análisis de esta variante emplea los valores de los índices de desempeño  $r_1^k$  (tabla 8.9),  $r_2^k$  (tabla 8.10) y  $\eta_s^k$  (tabla 8.11).

Para los ejemplares de menor tamaño es decir, cuando el número de máquinas  $m$  y el número de piezas  $n$  toman los valores más pequeños, los procedimientos **BB** y **BBP** presentan los mejores índices de desempeño. Sin embargo éstos empeoran en la medida en que crece el tamaño de los ejemplares.

Por otro lado, el procedimiento **SEP** muestra los peores valores de estos índices en los ejemplares de menor tamaño, sin embargo, en la medida en que el número de máquinas  $m$  y el número de piezas  $n$  aumentan, muestra una mejora creciente de dichos índices llegando a ser el procedimiento de mejor desempeño en los ejemplares de mayor tamaño.

El procedimiento **SEPA** presenta los mejores resultados en algunos ejemplares de tamaño menor (por ejemplo, en  $n = 5$  y  $m = 2$ , en  $n = 5$  y  $m = 4$  y en  $n = 5$  y  $m = 5$ ) igualando los mejores índices de desempeño hallados para esta variante, pero empeora muy rápidamente en la medida que crece el tamaño de los ejemplares (crecen  $n$  o  $m$ ).

**Variante PR1-CD2.** El análisis de esta variante emplea los valores de los índices de desempeño  $r_1^k$  (tabla 8.15),  $r_2^k$  (tabla 8.16) y  $\eta_s^k$  (tabla 8.17).

En esta variante, todos los procedimientos mantienen la tendencia de los índices de desempeño presentada en la variante anterior. Sin embargo, con respecto a la variante anterior, se produce un incremento del tamaño de los ejemplares (tanto en el valor de  $m$  como en el de  $n$ ) a partir del cual el procedimiento **SEP** comienza a mostrar los mejores resultados.

El procedimiento **SEPA** mantiene la tendencia de los índices de desempeño mostrada en la variante anterior. Es decir presenta los mejores resultados en algunos ejemplares de tamaño menor (por ejemplo, en  $n = 5$  y  $m = 2$  y en  $n = 5$  y  $m = 5$ ) igualando los mejores índices de desempeño hallados para esta variante, pero empeora muy rápidamente en la medida que crece el tamaño de los ejemplares.

**Variante PR2-CD1.** El análisis de esta variante emplea los valores de los índices de desempeño  $r_1^k$  (tabla 8.21),  $r_2^k$  (tabla 8.22) y  $\eta_s^k$  (tabla 8.23).

El procedimiento **BB** muestra los mejores índices de desempeño en casi la totalidad de los tamaños de ejemplares, mientras que, en los otros casos, muestra índices muy próximos a los mejores valores de estos índices.

El procedimiento **SEP** muestra índices de desempeño de baja calidad en los ejemplares de menor tamaño, sin embargo, mejora sus índices en la medida en que crece el tamaño de los ejemplares, llegando a mejorar ligeramente el desempeño del procedimiento **BB**.

Por otro lado, el procedimiento **SEPA** mantiene la tendencia mostrada en la variante anterior, es decir, empeora muy rápidamente con el tamaño de los ejemplares.

En cuanto al procedimiento **BBP**, a diferencia de las variantes anteriores donde el procedimiento **BBP** mantenía cierta paridad con el procedimiento **BB** en todos los tamaños, en esta variante, en la medida en que crece el tamaño de los ejemplares, el procedimiento **BBP** empeora sus índices de desempeño, presentando peores desempeños que el procedimiento **SEPA** en los ejemplares de mayor tamaño.

**Variante PR2-CD2.** El análisis de esta variante emplea los valores de los índices de desempeño  $r_1^k$  (tabla 8.27),  $r_2^k$  (tabla 8.28) y  $\eta_s^k$  (tabla 8.29).

En esta variante, todos los procedimientos muestran valores similares de los índices de desempeño que en la variante anterior e idénticas tendencias de estos índices, por lo que le caben las mismas conclusiones.

### 8.7.1.1 Conclusiones para la variante B1

En la variante de buffers de capacidad nula (variante **B1**) los mejores desempeños se dividen entre los procedimientos **BB** y **SEP**, como se detalla a continuación.

- ▶ En los ejemplares donde los tiempos de proceso son del mismo orden de magnitud respecto de los tiempos de carga y descarga (ejemplares de la variante **PR1**), el procedimiento **BB** muestra el mejor desempeño en los ejemplares de menor tamaño, mientras que en la medida que el tamaño de los ejemplares crece, el procedimiento **SEP** muestra el mejor desempeño, aunque seguido muy de cerca por el procedimiento **BB**.
- ▶ En los ejemplares donde los tiempos de proceso son grandes respecto de los tiempos de carga y descarga (ejemplares de la variante **PR2**), el procedimiento **BB** muestra el mejor desempeño en ejemplares de todos los tamaños.

### 8.7.2 Análisis de resultados de las variantes B2 y B3

En las variantes de buffers de capacidad no nula (variantes **B2** y **B3**) se observan valores similares de sus índices de desempeño e idénticas tendencias de estos índices, razón por la cual se los analiza conjuntamente.

**Variante PR1-CD1.** El análisis de esta variante emplea los valores de los índices de desempeño  $r_1^k$  (tablas 8.33 y 8.57),  $r_2^k$  (tablas 8.34 y 8.58) y  $\eta_s^k$  (tablas 8.35 y 8.59).

El análisis del índice de desempeño  $r_1^k$  muestra diferencias muy leves entre los distintos procedimientos y los diferentes tamaños de los ejemplares, lo que hace muy difícil poner de relieve sus diferencias y similitudes. Sin embargo, el análisis de los índices de desempeño  $r_2^k$  y  $\eta_s^k$  muestra con mayor contraste los desempeños de los distintos procedimientos. En efecto, los procedimientos **BB** y **BBP** presentan un mejor desempeño en ejemplares de menor tamaño, mientras que los procedimientos **SEP** y **SEPA** mejoran los valores de estos índices en la medida en que el tamaño de los ejemplares aumenta. Se observa también que el procedimiento **SEP** presenta los peores resultados en ejemplares de menor tamaño. Sin embargo, aún con estas tendencias, es de remarcar que los valores de los distintos índices se mantienen para todos los tamaños de ejemplares y para los distintos procedimientos en valores muy cercanos.

**Variante PR1-CD2.** El análisis de esta variante emplea los valores de los índices de desempeño  $r_1^k$  (tablas 8.39 y 8.63),  $r_2^k$  (tablas 8.40 y 8.64) y  $\eta_s^k$  (tablas 8.41 y 8.65).

En esta variante, como en la variante anterior, los desempeños medidos por los distintos índices se mantienen en valores similares para todos los procedimientos con excepción del procedimiento **BBP**. Éste muestra un desempeño similar al resto de los procedimientos cuando el tamaño de los ejemplares es pequeño, pero empeora sus índices de desempeño en la medida que el tamaño de los ejemplares crece. Se observa también que el procedimiento **SEP** presenta los peores resultados en ejemplares de menor tamaño.

**Variante PR2-CD1.** El análisis de esta variante emplea los valores de los índices de desempeño  $r_1^k$  que se muestran en las tablas 8.45 y 8.69,  $r_2^k$  que se muestran en las tablas 8.46 y 8.70 y  $\eta_s^k$  que se muestran en las tablas 8.47 y 8.71.

Para los ejemplares de menor tamaño los procedimientos **BB** y **BBP** muestran los mejores valores de los índices de desempeño, sin embargo, el procedimiento **BBP** empeora estos valores muy rápidamente en la medida que el tamaño de los ejemplares aumenta.

Por otra parte, cuando el tamaño de los ejemplares crece, los procedimientos **SEP** y **SEPA** muestran, con gran diferencia respecto de los procedimientos **BB** y **BBP**, los mejores valores de los índices de desempeño. Y entre los dos primeros, el procedimiento **SEPA** mejora notablemente los valores de sus índices respecto del procedimiento **SEP**.

Como en las variantes anteriores, se observa nuevamente que el procedimiento **SEP** presenta los peores resultados en ejemplares de menor tamaño.

**Variante PR2-CD2.** El análisis de esta variante emplea los valores de los índices de desempeño  $r_1^k$  (tablas 8.51 y 8.75),  $r_2^k$  (tablas 8.52 y 8.76) y  $\eta_s^k$  (tablas 8.53 y 8.77).

En esta variante, todos los procedimientos muestran valores similares de los índices de desempeño que en la variante anterior e idénticas tendencias de estos índices, por lo que le caben las mismas conclusiones.

### 8.7.2.1 Conclusiones para las variantes B2 y B3

Del análisis de la variante de buffers de capacidad no nula (variantes **B2** y **B3**) se desprende que los mejores desempeños se observan en los procedimientos **BB** y **SEPA**. En efecto:

- ▶ En los ejemplares donde los tiempos de proceso son del mismo orden de magnitud respecto de los tiempos de carga y descarga (ejemplares de la variante **PR1**), el procedimiento **SEP** muestra el peor desempeño en ejemplares de menor tamaño, y, por otro lado, el procedimiento **BBP** muestra el peor desempeño en ejemplares de mayor tamaño. En los procedimientos **BB** y **SEPA** se ubican los mejores desempeños con valores similares de los índices en ejemplares de todos los tamaños.
- ▶ En los ejemplares donde los tiempos de proceso son grandes respecto de los tiempos de carga y descarga (ejemplares del caso **PR2**), el procedimiento **BB** muestra el mejor desempeño en los ejemplares de menor tamaño, mientras que, en la medida que el tamaño de los ejemplares crece, el procedimiento **SEPA** muestra el mejor desempeño. De todas formas hay que considerar que, en los ejemplares de menor tamaño, el procedimiento **SEPA** muestra valores de los índices de desempeño muy próximos a los que presenta el procedimiento **BB**.

### 8.7.3 Conclusiones generales

En este apartado se analizan los desempeños de los procedimientos de resolución en el total de las variantes estudiadas.

#### 8.7.3.1 Sobre los ámbitos de aplicación de los procedimientos

A continuación se presenta una conclusión general acerca de qué procedimientos son más aptos para ser empleados según la variante de problema que se presente, a la luz de las conclusiones parciales de los apartados 8.7.1.1 y 8.7.2.1.

En la variante de ejemplares con buffers de capacidad nula se puede aconsejar el empleo del procedimiento **BB** para ejemplares de todos los tamaños. En efecto:

- ▶ en los ejemplares donde los tiempos de proceso son del mismo orden de magnitud respecto de los tiempos de carga y descarga (ejemplares de la variante **PR1**), el procedimiento **BB** muestra el mejor desempeño en los ejemplares de menor tamaño, mientras que cuando el tamaño de los ejemplares crece, muestra un desempeño, aunque inferior, muy cercano al mejor desempeño logrado por el procedimiento **SEP**;
- ▶ y, en los ejemplares donde los tiempos de proceso son grandes respecto de los tiempos de carga y descarga (ejemplares de la variante **PR2**), el procedimiento **BB** muestra el mejor desempeño en ejemplares de todos los tamaños.

En la variante de ejemplares de capacidad no nula es aconsejable el empleo de:

- ▶ el procedimiento **BB** en los ejemplares donde los tiempos de proceso son del mismo orden de magnitud respecto de los tiempos de carga y descarga (ejemplares de la variante **PR1**), donde muestra el mejor desempeño en los ejemplares de todos los tamaños;
- ▶ y, el procedimiento **SEPA** en los ejemplares donde los tiempos de proceso son grandes respecto de los tiempos de carga y descarga (ejemplares de la variante **PR2**), pues muestra el mejor desempeño (y con gran diferencia respecto del resto) en los ejemplares de mayor tamaño, y un desempeño muy cercano al mejor desempeño observado, en los ejemplares de menor tamaño.

### 8.7.3.2 Sobre los orígenes de los distintos desempeños

El peor desempeño del procedimiento **SEPA** se produce sólo en la variante de buffers nulos (variante **B1**). En las otras variantes se mantiene en desempeños del mismo orden o superior a los mostrados por los mejores procedimientos. Este peor desempeño puede deberse, quizás, a un mejoramiento del desempeño de los procedimientos basados en exploración (principalmente **BB** y **SEP**), debido, posiblemente, a que en la variante de buffers nulos se produce una poda adicional de aquellos movimientos que ingresan o sacan piezas de los buffers. Debido a ello una aceleración del proceso exploratorio, frente al procedimiento **SEPA**, donde este hecho afecta mínimamente su eficiencia.

Por otro lado, el procedimiento **SEP** sufre, respecto del procedimiento **BB**, una poda aún mayor de todos los movimientos que no se corresponden con la secuencia predefinida  $\bar{\Pi}$ , y a ello puede deberse el mejoramiento observado de los índices de desempeño de este procedimiento, cuando el tamaño de los ejemplares va en aumento.

El procedimiento **BBP** presenta buen desempeño en ejemplares de tamaños pequeños, en algunos pocos casos superando al procedimiento **BB**. Sin embargo, a medida que el tamaño de los ejemplares aumenta, este procedimiento empeora rápidamente. Este bajo desempeño muestra que no ocurre que «las mejores soluciones aparecen tempranamente», como se esperaba, quizás debido a que el método de eliminación de los peores descendientes (apartado 7.3) basado únicamente en los valores de las cotas no es suficientemente efectivo. Puede que el empleo de una estrategia diferente en el proceso de eliminación de los peores descendientes permitiera mejorar los resultados de este procedimiento.

En cuanto a los enfoques basados en la separación de la secuencia de piezas por un lado y la secuencia de movimientos del robot por el otro, se observa que la estrategia basada en una secuencia aleatorizada (múltiples secuencias) supera la estrategia de secuencia predefinida (secuencia única). Si bien la secuencia predefinida basada en el modelo de programación lineal no parece excesivamente mala, pues de hecho, el procedimiento **SEP** supera en muchos casos al procedimiento **BB** (principalmente en problemas de gran tamaño), es un hecho también que el procedimiento **SEPA** supera a ambos de forma notable. Por ejemplo, en las variantes donde los tiempos de proceso son grandes respecto de los tiempos del robot (variantes **PR2**), el procedimiento **SEPA** alcanza, para los ejemplares de mayor tamaño, errores respecto de la mejor cota, inferiores al 10%. Estos resultados muestran el buen funcionamiento del procedimiento de determinación de la secuencia de movimientos del robot basado en la construcción

iterativa de la solución, explicado en el apartado 7.5.2 y realizado en la segunda fase del procedimiento SEPA.

### 8.8 Análisis de los óptimos.

El análisis de los óptimos procede del análisis de los valores del índice  $\eta_o^k$  (tablas 8.12, 8.18, 8.24, 8.30, 8.36, 8.42, 8.48, 8.54, 8.60, 8.66, 8.72 y 8.78).

La capacidad de obtención de soluciones óptimas de los procedimientos que pueden garantizar la optimalidad de una solución (procedimientos **BBy BBP**), puede considerarse relacionado de forma inversa con el tamaño del espacio de estados a ser explorado. Por tanto, relacionado de forma inversa con el tamaño de los ejemplares (a mayor tamaño, menor es la capacidad de garantizar soluciones óptimas) y de forma directa con la capacidad de poda (a mayor eficiencia de los métodos de acotación, mayor es la capacidad de asegurar soluciones óptimas).

Esto se confirma observando los valores de los índices de desempeño, donde se ve que para todas las variantes de buffers (**B1**, **B2** y **B3**) se obtienen soluciones que se garantizan óptimas para los ejemplares de menor tamaño, mientras que la aparición de soluciones óptimas comprobadas cesa cuando los tamaños de los ejemplares se incrementan.

Por otro lado, particularmente en la variante de buffers nulos **B1**, se observa un mayor número de tamaños de ejemplares en los cuales se obtienen soluciones óptimas garantizadas, en relación con los casos de buffers no nulos (variantes **B2** y **B3**). La poda adicional de aquellos movimientos que depositan y extraen piezas de los buffers reduce el proceso de exploración y parece favorecer la terminación de forma óptima.

Finalmente se puede observar que en la medida en que crece el tiempo de proceso de las piezas (se pasa de **PR1** a **PR2**) crece el número de tamaños de ejemplares en los cuales se presentan soluciones óptimas comprobadas. Esto puede deberse a que en las variantes donde los tiempos de proceso crecen, la cota basada en los tiempos de las máquinas tiene el mejor desempeño. Sería razonable suponer que el proceso de poda por cota debería intensificarse y con ello la obtención de soluciones óptimas debería incrementarse.

### 8.9 Análisis de los procedimientos de acotación

En este apartado se evalúa el desempeño de los procedimientos de acotación para las distintas variantes de capacidad de los buffers (**B1**, **B2** y **B3**), para cada variante de tiempos de proceso (**PR1** y **PR2**) y cada variante de tiempos de carga y descarga (**CD1** y **CD2**), en función del número de piezas  $n$  (**P05** a **P20**) y del número de máquinas  $m$  (**M2** a **M5**) de los distintos ejemplares.

Para medir el desempeño de los procesos de acotación se emplean los índices de desempeño (definidos en el apartado 8.6) siguientes:

- *error relativo porcentual de la cota referido a la mejor cota*  $r_3^k$  (ecuación 8.32)

► y porcentual de casos de mejor cota hallada  $\eta_c^k$  (ecuación 8.33).

### 8.9.1 Análisis de resultados

**Variante PR1-CD1.** En el análisis de esta variante se utilizan los valores de los índices de desempeño  $r_3^k$  (tablas 8.13, 8.37 y 8.61) y los índices de desempeño  $\eta_c^k$  (tablas 8.14, 8.38 y 8.62).

El análisis de los valores de los índices de desempeño muestra que, para los ejemplares de menor tamaño la cota basada en los tiempos de las máquinas presenta un mejor desempeño que la cota basada en los tiempos del robot. Luego, cuando el tamaño de los ejemplares se incrementa, la cota basada en los tiempos del robot presenta mejor desempeño frente a la cota basada en los tiempos de las máquinas.

**Variante PR1-CD2.** En el análisis de esta variante se utilizan los valores de los índices de desempeño  $r_3^k$  (tablas 8.19, 8.43 y 8.67) y los índices de desempeño  $\eta_c^k$  (tablas 8.20, 8.44 y 8.68).

Se produce la misma división que en el la variante anterior. Para los ejemplares de menor tamaño la cota basada en los tiempos de las máquinas muestra un mejor desempeño que la cota basada en los tiempos del robot, y este resultado se invierte cuando crece el tamaño de los ejemplares. Sin embargo, se produce un incremento de los tamaños a partir de los cuales la cota basada en los tiempos del robot presenta mejor desempeño frente a la cota basada en los tiempos de las máquinas.

**Variantes PR2-CD1 y PR2-CD2.** En el análisis de esta variante se utilizan los valores de los índices de desempeño  $r_3^k$  (tablas 8.25, 8.31, 8.49, 8.55, 8.73 y 8.79) y los índices de desempeño  $\eta_c^k$  (tablas 8.26, 8.32, 8.50, 8.56, 8.74 y 8.80).

En estas variantes la cota basada en los tiempos de las máquinas muestra un mejor desempeño que la cota basada en los tiempos del robot para todos los tamaños de ejemplares, alcanzando, en todos éstos, el 100 % de efectividad ( $\eta_c^k = 100$ ).

También es notable los valores del índice  $r_3^k$  hallados para la cota basada en los tiempos del robot, en donde dicho índice alcanza, en algunos casos, valores cercanos al 85 % y nunca son inferiores al 55 %.

#### 8.9.1.1 Conclusiones

En los ejemplares en los que los tiempos de proceso son del mismo orden de magnitud respecto de los tiempos de carga y descarga (ejemplares de la variante **PR1**), la cota basada en los tiempos de las máquinas muestra el mejor desempeño en los ejemplares de menor tamaño; cuando el tamaño de los ejemplares crece, la cota basada en los tiempos del robot muestra el mejor desempeño.

Este comportamiento puede encontrar su explicación en el hecho de que la cota basada en los tiempos de las máquinas no es tan ajustada cuando los tiempos de las máquinas no son grandes (en estos ejemplares los valores de los tiempos de proceso se aproximan

a la condición de relajación de estos tiempos asumida en la cota basada en los tiempos del robot). Aún así, en ejemplares de menor tamaño muestra el mejor desempeño. Sin embargo, cuando el número de piezas  $n$  o el número de máquinas  $m$  crece, y puesto que el número de movimientos del robot crece en proporción directa al producto entre  $m$  y  $n$ , el robot se convierte en el cuello de botella y la cota basada en los tiempos del robot mejora frente a la cota basada en los tiempos de las máquinas.

Por otro lado, en los ejemplares donde los tiempos de proceso son grandes respecto de los tiempos de carga y descarga (ejemplares de la variante **PR2**), la cota basada en los tiempos de las máquinas muestra el mejor desempeño en ejemplares de todos los tamaños. Este comportamiento puede ser explicado por el hecho de que la relajación asumida en la cota basada en los tiempos del robot (tiempos de proceso nulos) está muy lejos de ser cumplida por los tiempos de proceso reales, dado que estos son, en la variante **PR2**, grandes respecto de los tiempos del robot.

## Análisis de los procedimientos de acotación

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	3.9682	14.8927	19.2177	21.4720
	<b>BBP</b>	3.9682	14.8927	19.2177	21.4720
	<b>SEP</b>	7.6348	18.3641	21.3355	24.1718
	<b>SEPA</b>	3.9682	20.3611	43.4195	56.8458
<b>M3</b> $m = 3$	<b>BB</b>	3.6497	19.5172	21.3460	22.8706
	<b>BBP</b>	3.6497	18.2333	21.9893	23.1167
	<b>SEP</b>	8.6000	20.6730	20.6804	21.3911
	<b>SEPA</b>	3.6909	26.5179	50.7449	61.6411
<b>M4</b> $m = 4$	<b>BB</b>	6.2368	19.2027	20.9869	22.5528
	<b>BBP</b>	6.2353	19.5572	21.4570	22.8435
	<b>SEP</b>	9.7599	18.5373	20.7608	22.0918
	<b>SEPA</b>	6.2180	30.9130	56.0301	69.5220
<b>M5</b> $m = 5$	<b>BB</b>	7.8587	18.8163	21.1672	22.3980
	<b>BBP</b>	7.4473	18.9879	21.4402	22.6371
	<b>SEP</b>	10.4406	18.4864	20.6188	21.7564
	<b>SEPA</b>	6.7837	39.5318	60.6686	72.3759

Tabla 8.9: Valores de *error relativo porcentual de las soluciones referido a la mejor cota  $r_1^k$* , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B1-CD1-PR1**.

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	0.0000	0.0000	0.0000	0.0000
	<b>BBP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEP</b>	3.5605	3.0125	1.7698	2.2197
	<b>SEPA</b>	0.0000	4.6258	20.3246	29.1053
<b>M3</b> $m = 3$	<b>BB</b>	0.0000	1.0901	1.1034	1.2194
	<b>BBP</b>	0.0000	0.0000	1.6376	1.4205
	<b>SEP</b>	4.8078	2.0711	0.5354	0.0000
	<b>SEPA</b>	0.0400	7.0080	25.5886	33.1736
<b>M4</b> $m = 4$	<b>BB</b>	0.3404	0.7176	0.5387	0.8729
	<b>BBP</b>	0.3462	1.0218	0.9300	1.1119
	<b>SEP</b>	3.6761	0.1547	0.3560	0.4949
	<b>SEPA</b>	0.3246	10.6163	29.6783	39.5378
<b>M5</b> $m = 5$	<b>BB</b>	1.2676	0.4971	0.6637	0.6352
	<b>BBP</b>	0.8843	0.6423	0.8893	0.8318
	<b>SEP</b>	3.7047	0.2182	0.2061	0.1088
	<b>SEPA</b>	0.2719	18.0331	33.4759	41.7515

Tabla 8.10: Valores de *error relativo porcentual de las soluciones referido a la mejor solución  $r_2^k$* , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B1-CD1-PR1**.

## Capítulo 8: Experiencia computacional

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	100.0000	100.0000	100.0000	100.0000
	<b>BBP</b>	100.0000	100.0000	100.0000	100.0000
	<b>SEP</b>	20.0000	0.0000	10.0000	0.0000
	<b>SEPA</b>	100.0000	30.0000	0.0000	0.0000
<b>M3</b> $m = 3$	<b>BB</b>	100.0000	10.0000	40.0000	0.0000
	<b>BBP</b>	100.0000	100.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	60.0000	100.0000
	<b>SEPA</b>	80.0000	0.0000	0.0000	0.0000
<b>M4</b> $m = 4$	<b>BB</b>	50.0000	20.0000	40.0000	40.0000
	<b>BBP</b>	50.0000	0.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	80.0000	60.0000	60.0000
	<b>SEPA</b>	10.0000	0.0000	0.0000	0.0000
<b>M5</b> $m = 5$	<b>BB</b>	30.0000	50.0000	30.0000	20.0000
	<b>BBP</b>	10.0000	20.0000	0.0000	10.0000
	<b>SEP</b>	0.0000	50.0000	70.0000	70.0000
	<b>SEPA</b>	60.0000	0.0000	0.0000	0.0000

Tabla 8.11: Valores de *porcentual de casos de mejor solución hallada*  $\eta_s^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B1-CD1-PR1**.

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	100.0000	100.0000	100.0000	100.0000
	<b>BBP</b>	100.0000	100.0000	100.0000	100.0000
	<b>SEP</b>	20.0000	0.0000	10.0000	0.0000
	<b>SEPA</b>	100.0000	30.0000	0.0000	0.0000
<b>M3</b> $m = 3$	<b>BB</b>	100.0000	0.0000	0.0000	0.0000
	<b>BBP</b>	100.0000	0.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	80.0000	0.0000	0.0000	0.0000
<b>M4</b> $m = 4$	<b>BB</b>	0.0000	0.0000	0.0000	0.0000
	<b>BBP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	0.0000	0.0000	0.0000	0.0000
<b>M5</b> $m = 5$	<b>BB</b>	0.0000	0.0000	0.0000	0.0000
	<b>BBP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	0.0000	0.0000	0.0000	0.0000

Tabla 8.12: Valores de *porcentual de casos de solución óptima hallada*  $\eta_o^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B1-CD1-PR1**.

## Análisis de los procedimientos de acotación

$m$	Cota	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	Máquinas	0.0000	1.5262	15.3328	17.8925
	Robot	20.3347	5.5449	0.0000	0.0000
<b>M3</b> $m = 3$	Máquinas	0.0000	7.5862	21.4472	32.8863
	Robot	16.8239	0.0000	0.0000	0.0000
<b>M4</b> $m = 4$	Máquinas	0.0000	9.6907	26.3607	35.5062
	Robot	14.2176	0.0000	0.0000	0.0000
<b>M5</b> $m = 5$	Máquinas	0.0000	10.3167	26.9458	38.1764
	Robot	13.2392	0.0000	0.0000	0.0000

Tabla 8.13: Valores de *error relativo porcentual de la cota referido a la mejor cota*  $r_3^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B1-CD1-PR1**.

$m$	Cota	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	Máquinas	100.0000	60.0000	0.0000	0.0000
	Robot	0.0000	40.0000	100.0000	100.0000
<b>M3</b> $m = 3$	Máquinas	100.0000	0.0000	0.0000	0.0000
	Robot	0.0000	100.0000	100.0000	100.0000
<b>M4</b> $m = 4$	Máquinas	100.0000	0.0000	0.0000	0.0000
	Robot	0.0000	100.0000	100.0000	100.0000
<b>M5</b> $m = 5$	Máquinas	100.0000	0.0000	0.0000	0.0000
	Robot	0.0000	100.0000	100.0000	100.0000

Tabla 8.14: Valores de *porcentual de casos de mejor cota hallada*  $\eta_c^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B1-CD1-PR1**.

## Capítulo 8: Experiencia computacional

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	2.0644	7.4242	14.6073	11.6708
	<b>BBP</b>	2.0644	7.4242	14.6073	11.6708
	<b>SEP</b>	4.3576	11.8161	20.9766	15.1568
	<b>SEPA</b>	2.0644	8.1035	25.5808	36.8112
<b>M3</b> $m = 3$	<b>BB</b>	1.8168	7.2945	10.6400	11.8560
	<b>BBP</b>	1.8168	7.2945	9.1014	12.5398
	<b>SEP</b>	6.7908	12.9602	14.2329	10.7548
	<b>SEPA</b>	1.8239	9.4009	22.8439	50.4288
<b>M4</b> $m = 4$	<b>BB</b>	2.4313	10.4068	9.5667	9.6223
	<b>BBP</b>	2.2371	10.1396	10.5333	10.0956
	<b>SEP</b>	9.1215	14.7159	9.9434	8.3809
	<b>SEPA</b>	2.8414	13.3496	38.0186	68.1176
<b>M5</b> $m = 5$	<b>BB</b>	4.2192	11.6853	9.3986	7.9625
	<b>BBP</b>	3.2752	18.0495	10.0097	8.2226
	<b>SEP</b>	8.6873	14.7803	9.8871	7.8080
	<b>SEPA</b>	4.2058	21.5942	50.4582	74.6421

Tabla 8.15: Valores de *error relativo porcentual de las soluciones referido a la mejor cota  $r_1^k$* , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B1-CD2-PR1**.

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	0.0000	0.0000	0.0000	0.0000
	<b>BBP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEP</b>	2.2410	4.1254	5.5812	3.1314
	<b>SEPA</b>	0.0000	0.6191	9.6114	22.5521
<b>M3</b> $m = 3$	<b>BB</b>	0.0000	0.0000	1.8071	1.3435
	<b>BBP</b>	0.0000	0.0000	0.3908	1.9719
	<b>SEP</b>	4.8899	5.3118	5.1062	0.3522
	<b>SEPA</b>	0.0069	1.9362	13.0452	36.3166
<b>M4</b> $m = 4$	<b>BB</b>	0.2465	0.7276	0.6102	1.3694
	<b>BBP</b>	0.0558	0.4716	1.4984	1.8066
	<b>SEP</b>	6.8068	4.6866	0.9664	0.2225
	<b>SEPA</b>	0.6460	3.3946	26.7421	55.4410
<b>M5</b> $m = 5$	<b>BB</b>	0.9456	0.0888	0.4782	0.6110
	<b>BBP</b>	0.0310	5.8076	1.0374	0.8584
	<b>SEP</b>	5.2923	2.8802	0.9258	0.4732
	<b>SEPA</b>	0.9298	8.9467	38.1944	62.7881

Tabla 8.16: Valores de *error relativo porcentual de las soluciones referido a la mejor solución  $r_2^k$* , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B1-CD2-PR1**.

## Análisis de los procedimientos de acotación

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	100.0000	100.0000	100.0000	100.0000
	<b>BBP</b>	100.0000	100.0000	100.0000	100.0000
	<b>SEP</b>	30.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	100.0000	70.0000	0.0000	0.0000
<b>M3</b> $m = 3$	<b>BB</b>	100.0000	100.0000	40.0000	30.0000
	<b>BBP</b>	100.0000	100.0000	60.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	70.0000
	<b>SEPA</b>	80.0000	10.0000	10.0000	0.0000
<b>M4</b> $m = 4$	<b>BB</b>	60.0000	50.0000	30.0000	40.0000
	<b>BBP</b>	70.0000	50.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	70.0000	60.0000
	<b>SEPA</b>	20.0000	0.0000	0.0000	0.0000
<b>M5</b> $m = 5$	<b>BB</b>	30.0000	90.0000	60.0000	30.0000
	<b>BBP</b>	60.0000	0.0000	0.0000	10.0000
	<b>SEP</b>	0.0000	10.0000	40.0000	60.0000
	<b>SEPA</b>	10.0000	0.0000	0.0000	0.0000

Tabla 8.17: Valores de *porcentual de casos de mejor solución hallada*  $\eta_s^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B1-CD2-PR1**.

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	100.0000	100.0000	100.0000	100.0000
	<b>BBP</b>	100.0000	100.0000	100.0000	100.0000
	<b>SEP</b>	30.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	100.0000	70.0000	0.0000	0.0000
<b>M3</b> $m = 3$	<b>BB</b>	100.0000	100.0000	20.0000	0.0000
	<b>BBP</b>	100.0000	100.0000	10.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	80.0000	10.0000	0.0000	0.0000
<b>M4</b> $m = 4$	<b>BB</b>	40.0000	0.0000	0.0000	0.0000
	<b>BBP</b>	40.0000	0.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	0.0000	0.0000	0.0000	0.0000
<b>M5</b> $m = 5$	<b>BB</b>	0.0000	0.0000	0.0000	0.0000
	<b>BBP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	0.0000	0.0000	0.0000	0.0000

Tabla 8.18: Valores de *porcentual de casos de solución óptima hallada*  $\eta_o^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B1-CD2-PR1**.

## Capítulo 8: Experiencia computacional

$m$	Cota	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	Máquinas	0.0000	0.0000	1.6547	7.5969
	Robot	29.2396	16.2878	0.9209	1.7337
<b>M3</b> $m = 3$	Máquinas	0.0000	1.4951	7.0794	22.0763
	Robot	24.1149	4.3185	0.0000	0.0000
<b>M4</b> $m = 4$	Máquinas	0.0000	1.0203	15.2038	25.3990
	Robot	23.0985	4.2380	0.0000	0.0000
<b>M5</b> $m = 5$	Máquinas	0.0000	0.8370	15.3341	27.3449
	Robot	22.5005	2.3322	0.0000	0.0000

Tabla 8.19: Valores de *error relativo porcentual de la cota referido a la mejor cota*  $r_3^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B1-CD2-PR1**.

$m$	Cota	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	Máquinas	100.0000	100.0000	40.0000	30.0000
	Robot	0.0000	0.0000	60.0000	70.0000
<b>M3</b> $m = 3$	Máquinas	100.0000	60.0000	0.0000	0.0000
	Robot	0.0000	40.0000	100.0000	100.0000
<b>M4</b> $m = 4$	Máquinas	100.0000	70.0000	0.0000	0.0000
	Robot	0.0000	30.0000	100.0000	100.0000
<b>M5</b> $m = 5$	Máquinas	100.0000	50.0000	0.0000	0.0000
	Robot	0.0000	50.0000	100.0000	100.0000

Tabla 8.20: Valores de *porcentual de casos de mejor cota hallada*  $\eta_c^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B1-CD2-PR1**.

## Análisis de los procedimientos de acotación

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	4.9187	5.7718	6.5614	15.8423
	<b>BBP</b>	4.9187	5.7718	6.5614	15.8423
	<b>SEP</b>	10.4834	11.4995	14.2304	23.5930
	<b>SEPA</b>	4.9187	6.4609	7.7721	17.0786
<b>M3</b> $m = 3$	<b>BB</b>	1.4691	3.8118	6.3687	12.5286
	<b>BBP</b>	1.4691	3.8118	6.3687	26.2093
	<b>SEP</b>	5.7197	12.8095	15.0507	19.2088
	<b>SEPA</b>	1.4691	4.4541	7.9575	10.6209
<b>M4</b> $m = 4$	<b>BB</b>	2.3631	5.9411	9.7046	17.1078
	<b>BBP</b>	2.3061	6.9600	33.2498	92.2053
	<b>SEP</b>	12.1761	16.0210	18.4988	22.0386
	<b>SEPA</b>	2.9494	9.3063	13.1282	20.5185
<b>M5</b> $m = 5$	<b>BB</b>	4.2213	12.1687	16.8318	21.3107
	<b>BBP</b>	2.8641	19.6171	60.8773	108.3059
	<b>SEP</b>	11.9873	21.7402	23.4941	25.1826
	<b>SEPA</b>	4.4432	20.4296	32.1991	58.2959

Tabla 8.21: Valores de *error relativo porcentual de las soluciones referido a la mejor cota  $r_1^k$* , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B1-CD1-PR2**.

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	0.0000	0.0000	0.0000	0.0000
	<b>BBP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEP</b>	5.2370	5.1719	7.2697	6.5375
	<b>SEPA</b>	0.0000	0.6288	1.1557	1.0730
<b>M3</b> $m = 3$	<b>BB</b>	0.0000	0.0000	0.0000	2.0795
	<b>BBP</b>	0.0000	0.0000	0.0000	14.4017
	<b>SEP</b>	4.1598	8.6479	8.1742	8.1829
	<b>SEPA</b>	0.0000	0.6134	1.4822	0.3573
<b>M4</b> $m = 4$	<b>BB</b>	0.0558	0.2797	0.3985	0.4028
	<b>BBP</b>	0.0000	1.2060	21.7479	64.7535
	<b>SEP</b>	9.6275	9.8574	8.4522	4.6750
	<b>SEPA</b>	0.6164	3.4414	3.5002	3.3530
<b>M5</b> $m = 5$	<b>BB</b>	1.3887	1.0025	0.0000	0.2436
	<b>BBP</b>	0.0818	7.4077	38.3695	72.2952
	<b>SEP</b>	8.9730	9.6138	5.6943	3.4461
	<b>SEPA</b>	1.6213	8.1766	13.0636	30.4431

Tabla 8.22: Valores de *error relativo porcentual de las soluciones referido a la mejor solución  $r_2^k$* , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B1-CD1-PR2**.

## Capítulo 8: Experiencia computacional

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	100.0000	100.0000	100.0000	100.0000
	<b>BBP</b>	100.0000	100.0000	100.0000	100.0000
	<b>SEP</b>	30.0000	30.0000	20.0000	0.0000
	<b>SEPA</b>	100.0000	80.0000	60.0000	40.0000
<b>M3</b> $m = 3$	<b>BB</b>	100.0000	100.0000	100.0000	50.0000
	<b>BBP</b>	100.0000	100.0000	100.0000	20.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	100.0000	30.0000	10.0000	50.0000
<b>M4</b> $m = 4$	<b>BB</b>	90.0000	80.0000	80.0000	80.0000
	<b>BBP</b>	100.0000	40.0000	10.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	10.0000
	<b>SEPA</b>	40.0000	0.0000	10.0000	10.0000
<b>M5</b> $m = 5$	<b>BB</b>	30.0000	80.0000	100.0000	80.0000
	<b>BBP</b>	80.0000	20.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	20.0000
	<b>SEPA</b>	10.0000	0.0000	0.0000	0.0000

Tabla 8.23: Valores de *porcentual de casos de mejor solución hallada*  $\eta_s^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B1-CD1-PR2**.

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	100.0000	100.0000	100.0000	100.0000
	<b>BBP</b>	100.0000	100.0000	100.0000	100.0000
	<b>SEP</b>	30.0000	30.0000	20.0000	0.0000
	<b>SEPA</b>	100.0000	80.0000	60.0000	40.0000
<b>M3</b> $m = 3$	<b>BB</b>	100.0000	100.0000	100.0000	20.0000
	<b>BBP</b>	100.0000	100.0000	100.0000	10.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	100.0000	30.0000	10.0000	0.0000
<b>M4</b> $m = 4$	<b>BB</b>	90.0000	20.0000	0.0000	0.0000
	<b>BBP</b>	90.0000	20.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	40.0000	0.0000	0.0000	0.0000
<b>M5</b> $m = 5$	<b>BB</b>	10.0000	0.0000	0.0000	0.0000
	<b>BBP</b>	10.0000	0.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	0.0000	0.0000	0.0000	0.0000

Tabla 8.24: Valores de *porcentual de casos de solución óptima hallada*  $\eta_o^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B1-CD1-PR2**.

## Análisis de los procedimientos de acotación

$m$	Cota	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	Máquinas	0.0000	0.0000	0.0000	0.0000
	Robot	77.5270	74.6309	74.7988	68.0126
<b>M3</b> $m = 3$	Máquinas	0.0000	0.0000	0.0000	0.0000
	Robot	78.9222	71.3597	66.9671	63.6743
<b>M4</b> $m = 4$	Máquinas	0.0000	0.0000	0.0000	0.0000
	Robot	77.4140	71.3698	65.7295	58.9899
<b>M5</b> $m = 5$	Máquinas	0.0000	0.0000	0.0000	0.0000
	Robot	76.7905	68.8516	63.3814	57.8184

Tabla 8.25: Valores de *error relativo porcentual de la cota referido a la mejor cota*  $r_3^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B1-CD1-PR2**.

$m$	Cota	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	Máquinas	100.0000	100.0000	100.0000	100.0000
	Robot	0.0000	0.0000	0.0000	0.0000
<b>M3</b> $m = 3$	Máquinas	100.0000	100.0000	100.0000	100.0000
	Robot	0.0000	0.0000	0.0000	0.0000
<b>M4</b> $m = 4$	Máquinas	100.0000	100.0000	100.0000	100.0000
	Robot	0.0000	0.0000	0.0000	0.0000
<b>M5</b> $m = 5$	Máquinas	100.0000	100.0000	100.0000	100.0000
	Robot	0.0000	0.0000	0.0000	0.0000

Tabla 8.26: Valores de *porcentual de casos de mejor cota hallada*  $\eta_c^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B1-CD1-PR2**.

## Capítulo 8: Experiencia computacional

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	4.0139	6.8784	8.1870	13.8456
	<b>BBP</b>	4.0139	6.8784	8.1870	13.8456
	<b>SEP</b>	10.0716	9.0765	11.6828	21.6577
	<b>SEPA</b>	4.0139	7.0459	8.6223	15.5689
<b>M3</b> $m = 3$	<b>BB</b>	2.8190	5.6017	5.9941	12.5456
	<b>BBP</b>	2.8190	5.6017	6.8763	37.7536
	<b>SEP</b>	10.7381	13.7525	15.5036	19.6539
	<b>SEPA</b>	2.8200	6.4099	6.7767	10.8356
<b>M4</b> $m = 4$	<b>BB</b>	3.3890	6.4797	12.6631	15.7752
	<b>BBP</b>	3.3890	9.2589	31.7115	73.8824
	<b>SEP</b>	11.9306	15.7846	21.9608	23.6625
	<b>SEPA</b>	4.3419	8.5433	14.8178	16.9761
<b>M5</b> $m = 5$	<b>BB</b>	6.8122	9.9387	17.1760	24.6405
	<b>BBP</b>	4.5774	12.7050	62.4618	115.6764
	<b>SEP</b>	15.3083	17.8933	23.3965	28.1458
	<b>SEPA</b>	5.7698	12.0987	29.9378	49.1244

Tabla 8.27: Valores de *error relativo porcentual de las soluciones referido a la mejor cota  $r_1^k$* , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B1-CD2-PR2**.

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	0.0000	0.0000	0.0000	0.0000
	<b>BBP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEP</b>	5.7848	2.0014	3.2635	6.6285
	<b>SEPA</b>	0.0000	0.1518	0.3933	1.5043
<b>M3</b> $m = 3$	<b>BB</b>	0.0000	0.0000	0.0000	2.2033
	<b>BBP</b>	0.0000	0.0000	0.7367	24.4863
	<b>SEP</b>	7.7481	7.7255	9.0396	8.6910
	<b>SEPA</b>	0.0009	0.7624	0.7306	0.7034
<b>M4</b> $m = 4$	<b>BB</b>	0.0000	0.6404	1.0358	0.9165
	<b>BBP</b>	0.0000	3.1756	17.7944	50.3069
	<b>SEP</b>	8.2237	9.4572	9.4811	7.8364
	<b>SEPA</b>	0.9234	2.5563	2.9374	1.8945
<b>M5</b> $m = 5$	<b>BB</b>	2.2499	1.0671	0.2072	0.8686
	<b>BBP</b>	0.0932	3.5554	39.3031	75.2096
	<b>SEP</b>	10.3892	8.3944	5.5298	3.8577
	<b>SEPA</b>	1.2306	3.0252	11.0264	20.7077

Tabla 8.28: Valores de *error relativo porcentual de las soluciones referido a la mejor solución  $r_2^k$* , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B1-CD2-PR2**.

## Análisis de los procedimientos de acotación

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	100.0000	100.0000	100.0000	100.0000
	<b>BBP</b>	100.0000	100.0000	100.0000	100.0000
	<b>SEP</b>	20.0000	20.0000	30.0000	10.0000
	<b>SEPA</b>	100.0000	90.0000	80.0000	50.0000
<b>M3</b> $m = 3$	<b>BB</b>	100.0000	100.0000	100.0000	40.0000
	<b>BBP</b>	100.0000	100.0000	90.0000	20.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	90.0000	50.0000	30.0000	50.0000
<b>M4</b> $m = 4$	<b>BB</b>	100.0000	70.0000	70.0000	70.0000
	<b>BBP</b>	100.0000	30.0000	20.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	0.0000	10.0000	10.0000	30.0000
<b>M5</b> $m = 5$	<b>BB</b>	20.0000	50.0000	80.0000	80.0000
	<b>BBP</b>	60.0000	20.0000	10.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	10.0000	20.0000
	<b>SEPA</b>	20.0000	30.0000	0.0000	0.0000

Tabla 8.29: Valores de *porcentual de casos de mejor solución hallada*  $\eta_s^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B1-CD2-PR2**.

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	100.0000	100.0000	100.0000	100.0000
	<b>BBP</b>	100.0000	100.0000	100.0000	100.0000
	<b>SEP</b>	20.0000	20.0000	30.0000	10.0000
	<b>SEPA</b>	100.0000	90.0000	80.0000	50.0000
<b>M3</b> $m = 3$	<b>BB</b>	100.0000	100.0000	100.0000	10.0000
	<b>BBP</b>	100.0000	100.0000	90.0000	10.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	90.0000	50.0000	30.0000	0.0000
<b>M4</b> $m = 4$	<b>BB</b>	90.0000	10.0000	0.0000	0.0000
	<b>BBP</b>	90.0000	10.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	0.0000	0.0000	0.0000	0.0000
<b>M5</b> $m = 5$	<b>BB</b>	0.0000	0.0000	0.0000	0.0000
	<b>BBP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	0.0000	0.0000	0.0000	0.0000

Tabla 8.30: Valores de *porcentual de casos de solución óptima hallada*  $\eta_o^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B1-CD2-PR2**.

## Capítulo 8: Experiencia computacional

$m$	Cota	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	Máquinas	0.0000	0.0000	0.0000	0.0000
	Robot	84.0677	81.2118	81.0790	77.8280
<b>M3</b> $m = 3$	Máquinas	0.0000	0.0000	0.0000	0.0000
	Robot	82.0635	77.3146	75.7782	72.5213
<b>M4</b> $m = 4$	Máquinas	0.0000	0.0000	0.0000	0.0000
	Robot	82.0750	78.2745	73.7502	70.0154
<b>M5</b> $m = 5$	Máquinas	0.0000	0.0000	0.0000	0.0000
	Robot	81.9894	79.0880	73.2228	68.5477

Tabla 8.31: Valores de *error relativo porcentual de la cota referido a la mejor cota*  $r_3^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B1-CD2-PR2**.

$m$	Cota	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	Máquinas	100.0000	100.0000	100.0000	100.0000
	Robot	0.0000	0.0000	0.0000	0.0000
<b>M3</b> $m = 3$	Máquinas	100.0000	100.0000	100.0000	100.0000
	Robot	0.0000	0.0000	0.0000	0.0000
<b>M4</b> $m = 4$	Máquinas	100.0000	100.0000	100.0000	100.0000
	Robot	0.0000	0.0000	0.0000	0.0000
<b>M5</b> $m = 5$	Máquinas	100.0000	100.0000	100.0000	100.0000
	Robot	0.0000	0.0000	0.0000	0.0000

Tabla 8.32: Valores de *porcentual de casos de mejor cota hallada*  $\eta_c^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B1-CD2-PR2**.

## Análisis de los procedimientos de acotación

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	8.6054	16.3405	21.6008	27.3580
	<b>BBP</b>	8.6054	16.3405	21.3538	32.6463
	<b>SEP</b>	11.5522	22.2005	25.3384	24.5448
	<b>SEPA</b>	9.4714	30.2769	31.7686	32.7824
<b>M3</b> $m = 3$	<b>BB</b>	11.1276	20.2930	27.2479	33.4415
	<b>BBP</b>	11.0845	22.4531	28.8293	33.1554
	<b>SEP</b>	15.1407	21.4260	24.7806	28.7831
	<b>SEPA</b>	11.6514	31.2771	26.6481	29.8781
<b>M4</b> $m = 4$	<b>BB</b>	15.6167	20.4298	27.1905	33.1789
	<b>BBP</b>	15.0797	21.6755	29.4909	33.9454
	<b>SEP</b>	17.8061	20.2447	25.3460	33.3612
	<b>SEPA</b>	13.7459	32.2260	26.6433	29.6661
<b>M5</b> $m = 5$	<b>BB</b>	16.4052	22.2453	25.3495	33.5191
	<b>BBP</b>	16.5628	23.1590	28.7529	33.9056
	<b>SEP</b>	17.2422	20.9486	24.4073	33.1199
	<b>SEPA</b>	14.3127	34.7057	26.8422	29.9052

Tabla 8.33: Valores de *error relativo porcentual de las soluciones referido a la mejor cota  $r_1^k$* , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B2-CD1-PR1**.

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	0.0000	0.0000	0.3081	2.9961
	<b>BBP</b>	0.0000	0.0000	0.1056	7.2357
	<b>SEP</b>	2.7354	5.0739	3.4430	0.6667
	<b>SEPA</b>	0.8222	12.0201	8.6969	7.3746
<b>M3</b> $m = 3$	<b>BB</b>	0.0372	0.4615	2.7464	4.2984
	<b>BBP</b>	0.0000	2.2741	4.0537	4.1069
	<b>SEP</b>	3.6419	1.4293	0.7700	0.6694
	<b>SEPA</b>	0.5262	9.6675	2.3209	1.5423
<b>M4</b> $m = 4$	<b>BB</b>	2.0182	0.7130	2.0866	3.1338
	<b>BBP</b>	1.5440	1.7590	3.9350	3.7359
	<b>SEP</b>	3.9914	0.5600	0.5874	3.2617
	<b>SEPA</b>	0.3909	10.5867	1.6445	0.4102
<b>M5</b> $m = 5$	<b>BB</b>	1.9157	1.3354	1.4410	2.9419
	<b>BBP</b>	2.0889	2.0983	4.2094	3.2401
	<b>SEP</b>	2.6861	0.2595	0.6880	2.6311
	<b>SEPA</b>	0.1283	11.6891	2.6641	0.1485

Tabla 8.34: Valores de *error relativo porcentual de las soluciones referido a la mejor solución  $r_2^k$* , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B2-CD1-PR1**.

**Capítulo 8: Experiencia computacional**

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	100.0000	100.0000	90.0000	30.0000
	<b>BBP</b>	100.0000	100.0000	90.0000	0.0000
	<b>SEP</b>	20.0000	0.0000	10.0000	70.0000
	<b>SEPA</b>	60.0000	0.0000	0.0000	0.0000
<b>M3</b> $m = 3$	<b>BB</b>	80.0000	70.0000	30.0000	20.0000
	<b>BBP</b>	100.0000	0.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	30.0000	50.0000	60.0000
	<b>SEPA</b>	10.0000	0.0000	20.0000	20.0000
<b>M4</b> $m = 4$	<b>BB</b>	10.0000	40.0000	20.0000	20.0000
	<b>BBP</b>	30.0000	0.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	60.0000	60.0000	20.0000
	<b>SEPA</b>	60.0000	0.0000	20.0000	60.0000
<b>M5</b> $m = 5$	<b>BB</b>	30.0000	40.0000	40.0000	10.0000
	<b>BBP</b>	10.0000	0.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	60.0000	60.0000	10.0000
	<b>SEPA</b>	60.0000	0.0000	0.0000	80.0000

Tabla 8.35: Valores de *porcentual de casos de mejor solución hallada*  $\eta_s^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B2-CD1-PR1**.

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	100.0000	100.0000	70.0000	0.0000
	<b>BBP</b>	100.0000	100.0000	70.0000	0.0000
	<b>SEP</b>	20.0000	0.0000	10.0000	0.0000
	<b>SEPA</b>	60.0000	0.0000	0.0000	0.0000
<b>M3</b> $m = 3$	<b>BB</b>	80.0000	0.0000	0.0000	0.0000
	<b>BBP</b>	80.0000	0.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	10.0000	0.0000	0.0000	0.0000
<b>M4</b> $m = 4$	<b>BB</b>	0.0000	0.0000	0.0000	0.0000
	<b>BBP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	0.0000	0.0000	0.0000	0.0000
<b>M5</b> $m = 5$	<b>BB</b>	0.0000	0.0000	0.0000	0.0000
	<b>BBP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	0.0000	0.0000	0.0000	0.0000

Tabla 8.36: Valores de *porcentual de casos de solución óptima hallada*  $\eta_o^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B2-CD1-PR1**.

## Análisis de los procedimientos de acotación

$m$	Cota	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	Máquinas	0.0000	0.0510	14.5445	22.2345
	Robot	14.9979	4.4437	0.0000	0.0000
<b>M3</b> $m = 3$	Máquinas	0.1645	12.7745	26.6770	39.7025
	Robot	10.6992	0.0000	0.0000	0.0000
<b>M4</b> $m = 4$	Máquinas	0.0893	15.0411	30.6787	41.0335
	Robot	8.1197	0.0000	0.0000	0.0000
<b>M5</b> $m = 5$	Máquinas	0.0000	18.2640	32.8693	42.9008
	Robot	10.2619	0.0000	0.0000	0.0000

Tabla 8.37: Valores de *error relativo porcentual de la cota referido a la mejor cota*  $r_3^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B2-CD1-PR1**.

$m$	Cota	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	Máquinas	100.0000	80.0000	0.0000	0.0000
	Robot	0.0000	20.0000	100.0000	100.0000
<b>M3</b> $m = 3$	Máquinas	90.0000	0.0000	0.0000	0.0000
	Robot	10.0000	100.0000	100.0000	100.0000
<b>M4</b> $m = 4$	Máquinas	90.0000	0.0000	0.0000	0.0000
	Robot	10.0000	100.0000	100.0000	100.0000
<b>M5</b> $m = 5$	Máquinas	100.0000	0.0000	0.0000	0.0000
	Robot	0.0000	100.0000	100.0000	100.0000

Tabla 8.38: Valores de *porcentual de casos de mejor cota hallada*  $\eta_c^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B2-CD1-PR1**.

## Capítulo 8: Experiencia computacional

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	6.5690	14.2344	14.1126	18.9220
	<b>BBP</b>	6.5690	14.2344	14.1126	25.7820
	<b>SEP</b>	8.7188	16.6538	18.6288	20.8561
	<b>SEPA</b>	6.5919	23.4709	22.0078	23.5545
<b>M3</b> $m = 3$	<b>BB</b>	4.6731	12.2236	15.7009	13.7326
	<b>BBP</b>	4.6731	16.1458	27.9057	32.5216
	<b>SEP</b>	9.4960	18.1096	13.8759	11.4831
	<b>SEPA</b>	4.8725	24.4512	10.5763	12.7129
<b>M4</b> $m = 4$	<b>BB</b>	4.7594	15.5801	10.3742	10.5112
	<b>BBP</b>	4.8039	19.8512	25.9733	31.3418
	<b>SEP</b>	10.2323	18.0221	9.2496	8.8593
	<b>SEPA</b>	5.4520	29.2212	8.6582	9.9706
<b>M5</b> $m = 5$	<b>BB</b>	7.0519	20.1710	10.2431	8.7582
	<b>BBP</b>	8.4324	27.1433	27.7093	31.6758
	<b>SEP</b>	11.6279	23.7618	10.9812	7.5654
	<b>SEPA</b>	7.3959	34.5333	8.8983	9.2181

Tabla 8.39: Valores de *error relativo porcentual de las soluciones referido a la mejor cota  $r_1^k$* , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B2-CD2-PR1**.

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	0.0000	0.0000	0.0000	2.2139
	<b>BBP</b>	0.0000	0.0000	0.0000	7.9556
	<b>SEP</b>	2.0141	2.0796	3.9273	3.7884
	<b>SEPA</b>	0.0215	8.0206	6.8739	6.0951
<b>M3</b> $m = 3$	<b>BB</b>	0.0000	0.0000	4.8366	3.0071
	<b>BBP</b>	0.0000	3.4465	15.8755	19.9838
	<b>SEP</b>	4.6250	5.2756	3.1615	0.9827
	<b>SEPA</b>	0.1920	10.9241	0.1866	2.0993
<b>M4</b> $m = 4$	<b>BB</b>	0.9308	0.3099	2.3476	2.4813
	<b>BBP</b>	0.9655	4.0058	16.7936	21.7711
	<b>SEP</b>	6.2101	2.4532	1.2924	0.9231
	<b>SEPA</b>	1.6098	12.1526	0.7573	1.9611
<b>M5</b> $m = 5$	<b>BB</b>	0.6909	0.2656	1.4429	1.4899
	<b>BBP</b>	1.9645	6.1596	17.5088	22.8724
	<b>SEP</b>	4.9769	3.2805	2.1297	0.3759
	<b>SEPA</b>	1.0045	12.3312	0.2085	1.9225

Tabla 8.40: Valores de *error relativo porcentual de las soluciones referido a la mejor solución  $r_2^k$* , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B2-CD2-PR1**.

## Análisis de los procedimientos de acotación

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	100.0000	100.0000	100.0000	80.0000
	<b>BBP</b>	100.0000	100.0000	100.0000	40.0000
	<b>SEP</b>	20.0000	0.0000	10.0000	10.0000
	<b>SEPA</b>	80.0000	0.0000	0.0000	10.0000
<b>M3</b> $m = 3$	<b>BB</b>	100.0000	100.0000	0.0000	30.0000
	<b>BBP</b>	100.0000	30.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	10.0000	70.0000
	<b>SEPA</b>	40.0000	0.0000	90.0000	0.0000
<b>M4</b> $m = 4$	<b>BB</b>	40.0000	70.0000	20.0000	30.0000
	<b>BBP</b>	60.0000	20.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	10.0000	30.0000	70.0000
	<b>SEPA</b>	10.0000	0.0000	50.0000	0.0000
<b>M5</b> $m = 5$	<b>BB</b>	50.0000	50.0000	20.0000	20.0000
	<b>BBP</b>	20.0000	20.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	30.0000	10.0000	80.0000
	<b>SEPA</b>	30.0000	0.0000	70.0000	0.0000

Tabla 8.41: Valores de *porcentual de casos de mejor solución hallada*  $\eta_s^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B2-CD2-PR1**.

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	100.0000	100.0000	100.0000	70.0000
	<b>BBP</b>	100.0000	100.0000	100.0000	40.0000
	<b>SEP</b>	20.0000	0.0000	10.0000	0.0000
	<b>SEPA</b>	80.0000	0.0000	0.0000	0.0000
<b>M3</b> $m = 3$	<b>BB</b>	100.0000	40.0000	0.0000	0.0000
	<b>BBP</b>	100.0000	30.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	40.0000	0.0000	0.0000	0.0000
<b>M4</b> $m = 4$	<b>BB</b>	10.0000	0.0000	0.0000	0.0000
	<b>BBP</b>	10.0000	0.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	0.0000	0.0000	0.0000	0.0000
<b>M5</b> $m = 5$	<b>BB</b>	0.0000	0.0000	0.0000	0.0000
	<b>BBP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	0.0000	0.0000	0.0000	0.0000

Tabla 8.42: Valores de *porcentual de casos de solución óptima hallada*  $\eta_o^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B2-CD2-PR1**.

## Capítulo 8: Experiencia computacional

$m$	Cota	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	Máquinas	0.0000	0.2538	0.8540	7.9194
	Robot	25.0833	10.2778	5.5301	1.0104
<b>M3</b> $m = 3$	Máquinas	0.0000	0.6461	12.3065	21.5982
	Robot	22.3994	1.7966	0.0000	0.0000
<b>M4</b> $m = 4$	Máquinas	0.0000	3.4718	18.5216	28.8501
	Robot	24.8879	1.7849	0.0000	0.0000
<b>M5</b> $m = 5$	Máquinas	0.0000	3.4383	19.5319	31.3700
	Robot	21.1195	0.7260	0.0000	0.0000

Tabla 8.43: Valores de *error relativo porcentual de la cota referido a la mejor cota*  $r_3^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B2-CD2-PR1**.

$m$	Cota	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	Máquinas	100.0000	80.0000	70.0000	30.0000
	Robot	0.0000	20.0000	30.0000	70.0000
<b>M3</b> $m = 3$	Máquinas	100.0000	70.0000	0.0000	0.0000
	Robot	0.0000	30.0000	100.0000	100.0000
<b>M4</b> $m = 4$	Máquinas	100.0000	40.0000	0.0000	0.0000
	Robot	0.0000	60.0000	100.0000	100.0000
<b>M5</b> $m = 5$	Máquinas	100.0000	30.0000	0.0000	0.0000
	Robot	0.0000	70.0000	100.0000	100.0000

Tabla 8.44: Valores de *porcentual de casos de mejor cota hallada*  $\eta_c^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B2-CD2-PR1**.

## Análisis de los procedimientos de acotación

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	3.1824	0.5280	10.4985	23.3706
	<b>BBP</b>	3.1824	0.5280	12.5701	40.1976
	<b>SEP</b>	5.0115	5.7880	13.5828	19.1817
	<b>SEPA</b>	3.3150	1.0835	10.1272	11.7983
<b>M3</b> $m = 3$	<b>BB</b>	0.1003	3.2956	14.4554	48.5968
	<b>BBP</b>	0.1003	6.1551	58.8941	91.1155
	<b>SEP</b>	1.9677	7.8774	10.8963	15.0830
	<b>SEPA</b>	0.1186	1.5686	3.9385	6.9010
<b>M4</b> $m = 4$	<b>BB</b>	0.1736	4.5480	22.0239	41.4945
	<b>BBP</b>	0.1736	16.8631	100.6131	159.9379
	<b>SEP</b>	2.6683	4.6140	11.2602	16.7671
	<b>SEPA</b>	0.3501	1.0057	3.5555	7.1023
<b>M5</b> $m = 5$	<b>BB</b>	1.6753	3.8351	22.9292	39.2952
	<b>BBP</b>	0.3034	19.6689	98.8323	176.2184
	<b>SEP</b>	4.1022	8.5525	11.4420	14.2931
	<b>SEPA</b>	0.4004	2.7886	4.0007	9.7029

Tabla 8.45: Valores de *error relativo porcentual de las soluciones referido a la mejor cota  $r_1^k$* , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B2-CD1-PR2**.

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	0.0000	0.0000	1.3123	10.1445
	<b>BBP</b>	0.0000	0.0000	3.0144	24.9609
	<b>SEP</b>	1.7074	5.2398	4.1109	6.5250
	<b>SEPA</b>	0.1288	0.5514	0.9510	0.0000
<b>M3</b> $m = 3$	<b>BB</b>	0.0000	2.1555	10.2835	38.4566
	<b>BBP</b>	0.0000	4.9380	52.8443	78.1084
	<b>SEP</b>	1.8657	6.6440	6.8534	7.7001
	<b>SEPA</b>	0.0182	0.4260	0.1010	0.0000
<b>M4</b> $m = 4$	<b>BB</b>	0.0017	3.6243	17.6850	31.9373
	<b>BBP</b>	0.0017	15.8519	93.2175	142.6353
	<b>SEP</b>	2.4904	3.6925	7.3661	9.0764
	<b>SEPA</b>	0.1771	0.1194	0.0000	0.0000
<b>M5</b> $m = 5$	<b>BB</b>	1.3882	1.2731	18.0649	26.7349
	<b>BBP</b>	0.0227	16.7885	91.1241	151.5683
	<b>SEP</b>	3.8062	5.8391	7.1327	4.2045
	<b>SEPA</b>	0.1191	0.2520	0.0000	0.0000

Tabla 8.46: Valores de *error relativo porcentual de las soluciones referido a la mejor solución  $r_2^k$* , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B2-CD1-PR2**.

**Capítulo 8: Experiencia computacional**

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	100.0000	100.0000	50.0000	0.0000
	<b>BBP</b>	100.0000	100.0000	60.0000	0.0000
	<b>SEP</b>	50.0000	20.0000	30.0000	0.0000
	<b>SEPA</b>	90.0000	30.0000	10.0000	100.0000
<b>M3</b> $m = 3$	<b>BB</b>	100.0000	50.0000	0.0000	0.0000
	<b>BBP</b>	100.0000	50.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	10.0000	0.0000
	<b>SEPA</b>	90.0000	40.0000	90.0000	100.0000
<b>M4</b> $m = 4$	<b>BB</b>	90.0000	0.0000	0.0000	0.0000
	<b>BBP</b>	90.0000	10.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	60.0000	90.0000	100.0000	100.0000
<b>M5</b> $m = 5$	<b>BB</b>	30.0000	20.0000	0.0000	0.0000
	<b>BBP</b>	70.0000	0.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	50.0000	80.0000	100.0000	100.0000

Tabla 8.47: Valores de *porcentual de casos de mejor solución hallada*  $\eta_s^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B2-CD1-PR2**.

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	100.0000	100.0000	40.0000	0.0000
	<b>BBP</b>	100.0000	100.0000	40.0000	0.0000
	<b>SEP</b>	50.0000	20.0000	20.0000	0.0000
	<b>SEPA</b>	90.0000	30.0000	0.0000	0.0000
<b>M3</b> $m = 3$	<b>BB</b>	100.0000	40.0000	0.0000	0.0000
	<b>BBP</b>	100.0000	40.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	90.0000	0.0000	0.0000	0.0000
<b>M4</b> $m = 4$	<b>BB</b>	80.0000	0.0000	0.0000	0.0000
	<b>BBP</b>	80.0000	0.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	40.0000	0.0000	0.0000	0.0000
<b>M5</b> $m = 5$	<b>BB</b>	30.0000	0.0000	0.0000	0.0000
	<b>BBP</b>	30.0000	0.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	20.0000	0.0000	0.0000	0.0000

Tabla 8.48: Valores de *porcentual de casos de solución óptima hallada*  $\eta_o^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B2-CD1-PR2**.

## Análisis de los procedimientos de acotación

$m$	Cota	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	Máquinas	0.0000	0.0000	0.0000	0.0000
	Robot	76.9234	74.4603	71.0038	68.4804
<b>M3</b> $m = 3$	Máquinas	0.0000	0.0000	0.0000	0.0000
	Robot	76.4034	71.5910	66.3067	64.6287
<b>M4</b> $m = 4$	Máquinas	0.0000	0.0000	0.0000	0.0000
	Robot	75.0350	71.3831	64.0976	58.8817
<b>M5</b> $m = 5$	Máquinas	0.0000	0.0000	0.0000	0.0000
	Robot	76.6654	69.1159	63.2479	56.2566

Tabla 8.49: Valores de *error relativo porcentual de la cota referido a la mejor cota*  $r_3^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B2-CD1-PR2**.

$m$	Cota	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	Máquinas	100.0000	100.0000	100.0000	100.0000
	Robot	0.0000	0.0000	0.0000	0.0000
<b>M3</b> $m = 3$	Máquinas	100.0000	100.0000	100.0000	100.0000
	Robot	0.0000	0.0000	0.0000	0.0000
<b>M4</b> $m = 4$	Máquinas	100.0000	100.0000	100.0000	100.0000
	Robot	0.0000	0.0000	0.0000	0.0000
<b>M5</b> $m = 5$	Máquinas	100.0000	100.0000	100.0000	100.0000
	Robot	0.0000	0.0000	0.0000	0.0000

Tabla 8.50: Valores de *porcentual de casos de mejor cota hallada*  $\eta_c^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B2-CD1-PR2**.

## Capítulo 8: Experiencia computacional

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	2.5078	14.2669	10.6233	18.2931
	<b>BBP</b>	2.5078	14.2669	9.6943	36.6754
	<b>SEP</b>	3.7142	22.9189	12.4085	15.3629
	<b>SEPA</b>	2.5078	14.4520	7.8232	9.8802
<b>M3</b> $m = 3$	<b>BB</b>	0.0887	2.2560	16.2332	32.8274
	<b>BBP</b>	0.0887	6.1088	69.0095	106.4959
	<b>SEP</b>	1.3139	8.4097	10.1061	16.1032
	<b>SEPA</b>	0.0887	1.5022	3.5660	5.8969
<b>M4</b> $m = 4$	<b>BB</b>	0.5381	3.7683	15.8388	32.4155
	<b>BBP</b>	0.2039	23.3621	102.5906	136.6745
	<b>SEP</b>	4.7856	6.8485	8.7524	12.6286
	<b>SEPA</b>	0.2579	2.3617	1.6562	3.6639
<b>M5</b> $m = 5$	<b>BB</b>	0.6921	5.8471	18.3297	37.3094
	<b>BBP</b>	0.1616	36.1172	106.4595	166.9911
	<b>SEP</b>	4.3123	7.6510	10.7249	14.4797
	<b>SEPA</b>	0.3168	2.4109	3.0081	5.1375

Tabla 8.51: Valores de *error relativo porcentual de las soluciones referido a la mejor cota  $r_1^k$* , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B2-CD2-PR2**.

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	0.0000	0.0000	2.8654	7.3757
	<b>BBP</b>	0.0000	0.0000	1.9595	24.1400
	<b>SEP</b>	1.1498	7.4283	4.4504	4.8561
	<b>SEPA</b>	0.0000	0.1814	0.2581	0.0742
<b>M3</b> $m = 3$	<b>BB</b>	0.0000	0.8939	12.1099	25.4287
	<b>BBP</b>	0.0000	4.7299	62.9889	94.3026
	<b>SEP</b>	1.2233	6.9706	6.2105	9.6057
	<b>SEPA</b>	0.0000	0.1771	0.0426	0.0000
<b>M4</b> $m = 4$	<b>BB</b>	0.3289	1.4712	13.9097	27.6355
	<b>BBP</b>	0.0000	20.5853	99.1312	128.0293
	<b>SEP</b>	4.5755	4.4985	6.9706	8.6276
	<b>SEPA</b>	0.0539	0.1205	0.0000	0.0000
<b>M5</b> $m = 5$	<b>BB</b>	0.5409	3.3639	14.8581	30.6338
	<b>BBP</b>	0.0114	32.7734	100.4598	153.4806
	<b>SEP</b>	4.1543	5.1009	7.4767	8.9222
	<b>SEPA</b>	0.1662	0.0457	0.0000	0.0000

Tabla 8.52: Valores de *error relativo porcentual de las soluciones referido a la mejor solución  $r_2^k$* , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B2-CD2-PR2**.

## Análisis de los procedimientos de acotación

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	100.0000	100.0000	70.0000	0.0000
	<b>BBP</b>	100.0000	100.0000	70.0000	0.0000
	<b>SEP</b>	10.0000	0.0000	30.0000	10.0000
	<b>SEPA</b>	100.0000	70.0000	30.0000	90.0000
<b>M3</b> $m = 3$	<b>BB</b>	100.0000	60.0000	0.0000	0.0000
	<b>BBP</b>	100.0000	60.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	20.0000	0.0000
	<b>SEPA</b>	100.0000	70.0000	80.0000	100.0000
<b>M4</b> $m = 4$	<b>BB</b>	80.0000	40.0000	0.0000	0.0000
	<b>BBP</b>	100.0000	0.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	40.0000	80.0000	100.0000	100.0000
<b>M5</b> $m = 5$	<b>BB</b>	20.0000	10.0000	0.0000	0.0000
	<b>BBP</b>	70.0000	0.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	20.0000	90.0000	100.0000	100.0000

Tabla 8.53: Valores de *porcentual de casos de mejor solución hallada*  $\eta_s^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B2-CD2-PR2**.

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	100.0000	100.0000	30.0000	0.0000
	<b>BBP</b>	100.0000	100.0000	50.0000	0.0000
	<b>SEP</b>	10.0000	0.0000	10.0000	0.0000
	<b>SEPA</b>	100.0000	70.0000	20.0000	0.0000
<b>M3</b> $m = 3$	<b>BB</b>	100.0000	60.0000	0.0000	0.0000
	<b>BBP</b>	100.0000	60.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	100.0000	40.0000	0.0000	0.0000
<b>M4</b> $m = 4$	<b>BB</b>	50.0000	20.0000	0.0000	0.0000
	<b>BBP</b>	50.0000	0.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	20.0000	10.0000	0.0000	0.0000
<b>M5</b> $m = 5$	<b>BB</b>	10.0000	0.0000	0.0000	0.0000
	<b>BBP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	20.0000	0.0000	0.0000	0.0000

Tabla 8.54: Valores de *porcentual de casos de solución óptima hallada*  $\eta_o^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B2-CD2-PR2**.

## Capítulo 8: Experiencia computacional

$m$	Cota	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	Máquinas	0.0000	0.0000	0.0000	0.0000
	Robot	82.9158	79.1269	78.4993	77.2679
<b>M3</b> $m = 3$	Máquinas	0.0000	0.0000	0.0000	0.0000
	Robot	82.2557	77.1542	74.6996	69.4447
<b>M4</b> $m = 4$	Máquinas	0.0000	0.0000	0.0000	0.0000
	Robot	80.4945	76.6706	73.8281	69.7415
<b>M5</b> $m = 5$	Máquinas	0.0000	0.0000	0.0000	0.0000
	Robot	81.9509	76.8402	73.0214	68.3244

Tabla 8.55: Valores de *error relativo porcentual de la cota referido a la mejor cota*  $r_3^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B2-CD2-PR2**.

$m$	Cota	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	Máquinas	100.0000	100.0000	100.0000	100.0000
	Robot	0.0000	0.0000	0.0000	0.0000
<b>M3</b> $m = 3$	Máquinas	100.0000	100.0000	100.0000	100.0000
	Robot	0.0000	0.0000	0.0000	0.0000
<b>M4</b> $m = 4$	Máquinas	100.0000	100.0000	100.0000	100.0000
	Robot	0.0000	0.0000	0.0000	0.0000
<b>M5</b> $m = 5$	Máquinas	100.0000	100.0000	100.0000	100.0000
	Robot	0.0000	0.0000	0.0000	0.0000

Tabla 8.56: Valores de *porcentual de casos de mejor cota hallada*  $\eta_c^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B2-CD2-PR2**.

## Análisis de los procedimientos de acotación

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	8.0266	18.3678	22.5747	26.9833
	<b>BBP</b>	8.0266	18.3678	23.8116	30.5779
	<b>SEP</b>	10.3287	24.7559	25.8744	22.9596
	<b>SEPA</b>	9.1599	31.5843	33.8940	32.4953
<b>M3</b> $m = 3$	<b>BB</b>	11.2171	20.5076	25.2384	33.6794
	<b>BBP</b>	11.2235	22.8895	29.5516	34.2462
	<b>SEP</b>	15.2014	19.4413	23.0000	30.9049
	<b>SEPA</b>	11.3841	30.7057	26.4772	30.0886
<b>M4</b> $m = 4$	<b>BB</b>	12.6790	20.8068	25.5605	31.9276
	<b>BBP</b>	12.7419	22.2098	28.1322	33.6678
	<b>SEP</b>	16.0325	19.5473	23.8885	33.0494
	<b>SEPA</b>	11.7670	33.1847	26.1721	30.4165
<b>M5</b> $m = 5$	<b>BB</b>	18.6653	19.1906	24.0076	34.8644
	<b>BBP</b>	19.3771	20.6115	28.0475	33.6738
	<b>SEP</b>	19.7436	19.2060	24.3650	32.4787
	<b>SEPA</b>	16.0553	34.2439	25.7899	29.7209

Tabla 8.57: Valores de *error relativo porcentual de las soluciones referido a la mejor cota  $r_1^k$* , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B3-CD1-PR1**.

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	0.0000	0.0000	0.7017	3.4599
	<b>BBP</b>	0.0000	0.0000	1.6945	6.3281
	<b>SEP</b>	2.1564	5.3949	3.4310	0.1454
	<b>SEPA</b>	1.0638	11.1693	10.0089	7.9633
<b>M3</b> $m = 3$	<b>BB</b>	0.0000	1.2953	2.0810	3.6103
	<b>BBP</b>	0.0054	3.2986	5.6205	4.0707
	<b>SEP</b>	3.6112	0.4114	0.2528	1.4595
	<b>SEPA</b>	0.1519	9.8775	3.1140	0.8507
<b>M4</b> $m = 4$	<b>BB</b>	1.0773	1.2878	1.8102	1.9229
	<b>BBP</b>	1.1159	2.4676	3.9444	3.2827
	<b>SEP</b>	4.1243	0.2338	0.4693	2.7815
	<b>SEPA</b>	0.2713	11.6580	2.3645	0.7672
<b>M5</b> $m = 5$	<b>BB</b>	2.2645	0.4533	0.7671	4.0961
	<b>BBP</b>	2.9021	1.6547	4.0883	3.1972
	<b>SEP</b>	3.2014	0.4649	1.0487	2.2634
	<b>SEPA</b>	0.0344	13.1480	2.2393	0.1323

Tabla 8.58: Valores de *error relativo porcentual de las soluciones referido a la mejor solución  $r_2^k$* , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B3-CD1-PR1**.

**Capítulo 8: Experiencia computacional**

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	100.0000	100.0000	90.0000	10.0000
	<b>BBP</b>	100.0000	100.0000	70.0000	10.0000
	<b>SEP</b>	20.0000	0.0000	0.0000	80.0000
	<b>SEPA</b>	60.0000	0.0000	0.0000	0.0000
<b>M3</b> $m = 3$	<b>BB</b>	100.0000	40.0000	40.0000	30.0000
	<b>BBP</b>	90.0000	0.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	60.0000	60.0000	30.0000
	<b>SEPA</b>	60.0000	0.0000	0.0000	40.0000
<b>M4</b> $m = 4$	<b>BB</b>	10.0000	20.0000	0.0000	20.0000
	<b>BBP</b>	20.0000	10.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	70.0000	70.0000	20.0000
	<b>SEPA</b>	70.0000	0.0000	30.0000	60.0000
<b>M5</b> $m = 5$	<b>BB</b>	10.0000	30.0000	60.0000	0.0000
	<b>BBP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	70.0000	30.0000	20.0000
	<b>SEPA</b>	90.0000	0.0000	10.0000	80.0000

Tabla 8.59: Valores de *porcentual de casos de mejor solución hallada*  $\eta_s^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B3-CD1-PR1**.

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	100.0000	100.0000	80.0000	0.0000
	<b>BBP</b>	100.0000	100.0000	60.0000	0.0000
	<b>SEP</b>	20.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	60.0000	0.0000	0.0000	0.0000
<b>M3</b> $m = 3$	<b>BB</b>	90.0000	0.0000	0.0000	0.0000
	<b>BBP</b>	90.0000	0.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	50.0000	0.0000	0.0000	0.0000
<b>M4</b> $m = 4$	<b>BB</b>	0.0000	0.0000	0.0000	0.0000
	<b>BBP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	0.0000	0.0000	0.0000	0.0000
<b>M5</b> $m = 5$	<b>BB</b>	0.0000	0.0000	0.0000	0.0000
	<b>BBP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	0.0000	0.0000	0.0000	0.0000

Tabla 8.60: Valores de *porcentual de casos de solución óptima hallada*  $\eta_o^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B3-CD1-PR1**.

## Análisis de los procedimientos de acotación

$m$	Cota	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	Máquinas	0.0000	7.0214	12.3368	23.5705
	Robot	16.5792	0.1942	0.0000	0.0000
<b>M3</b> $m = 3$	Máquinas	0.0000	13.6102	28.1655	37.4750
	Robot	8.5517	0.0000	0.0000	0.0000
<b>M4</b> $m = 4$	Máquinas	0.0000	15.3157	32.5962	41.1183
	Robot	11.5322	0.0000	0.0000	0.0000
<b>M5</b> $m = 5$	Máquinas	0.2551	17.1709	32.9189	43.6016
	Robot	6.0420	0.0000	0.0000	0.0000

Tabla 8.61: Valores de *error relativo porcentual de la cota referido a la mejor cota*  $r_3^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B3-CD1-PR1**.

$m$	Cota	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	Máquinas	100.0000	10.0000	0.0000	0.0000
	Robot	0.0000	90.0000	100.0000	100.0000
<b>M3</b> $m = 3$	Máquinas	100.0000	0.0000	0.0000	0.0000
	Robot	0.0000	100.0000	100.0000	100.0000
<b>M4</b> $m = 4$	Máquinas	100.0000	0.0000	0.0000	0.0000
	Robot	0.0000	100.0000	100.0000	100.0000
<b>M5</b> $m = 5$	Máquinas	80.0000	0.0000	0.0000	0.0000
	Robot	20.0000	100.0000	100.0000	100.0000

Tabla 8.62: Valores de *porcentual de casos de mejor cota hallada*  $\eta_c^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B3-CD1-PR1**.

## Capítulo 8: Experiencia computacional

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	4.0474	4.5264	14.3414	13.4106
	<b>BBP</b>	4.0474	4.5264	14.3414	20.3536
	<b>SEP</b>	7.4139	8.2384	19.1987	15.7804
	<b>SEPA</b>	4.1132	13.5704	21.6407	21.7325
<b>M3</b> $m = 3$	<b>BB</b>	4.0898	11.4305	14.3765	13.8809
	<b>BBP</b>	4.0898	14.0042	28.7605	34.6192
	<b>SEP</b>	9.1396	17.3283	14.6086	12.6002
	<b>SEPA</b>	4.2037	22.7637	10.1806	12.4152
<b>M4</b> $m = 4$	<b>BB</b>	6.2806	16.1399	13.3397	11.3673
	<b>BBP</b>	4.5958	22.1505	28.7307	31.9981
	<b>SEP</b>	11.9165	18.0785	9.9723	8.5838
	<b>SEPA</b>	6.3368	29.2606	9.6372	9.9533
<b>M5</b> $m = 5$	<b>BB</b>	7.8748	15.3967	9.5809	8.9988
	<b>BBP</b>	9.2852	25.0003	26.0527	33.3730
	<b>SEP</b>	11.3493	21.9019	9.4602	7.9676
	<b>SEPA</b>	8.3907	32.5635	8.4533	9.6562

Tabla 8.63: Valores de *error relativo porcentual de las soluciones referido a la mejor cota  $r_1^k$* , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B3-CD2-PR1**.

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	0.0000	0.0000	0.0000	0.1558
	<b>BBP</b>	0.0000	0.0000	0.0000	6.3109
	<b>SEP</b>	3.2454	3.5473	4.2635	2.2906
	<b>SEPA</b>	0.0601	8.5904	6.3802	7.6283
<b>M3</b> $m = 3$	<b>BB</b>	0.0000	0.4762	3.8282	2.1176
	<b>BBP</b>	0.0000	2.7579	16.9122	20.6898
	<b>SEP</b>	4.8353	5.7934	4.0400	0.9594
	<b>SEPA</b>	0.1113	10.6814	0.0418	0.7972
<b>M4</b> $m = 4$	<b>BB</b>	1.6737	0.3295	3.8050	2.8771
	<b>BBP</b>	0.0952	5.5166	17.8929	21.9215
	<b>SEP</b>	7.0862	2.0044	0.7374	0.2777
	<b>SEPA</b>	1.7527	11.6683	0.4324	1.5584
<b>M5</b> $m = 5$	<b>BB</b>	0.5477	0.0557	1.4582	1.6919
	<b>BBP</b>	1.8621	8.5252	16.7023	24.4245
	<b>SEP</b>	3.8009	5.6652	1.3478	0.7135
	<b>SEPA</b>	1.0489	14.9574	0.4140	2.3015

Tabla 8.64: Valores de *error relativo porcentual de las soluciones referido a la mejor solución  $r_2^k$* , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B3-CD2-PR1**.

## Análisis de los procedimientos de acotación

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	100.0000	100.0000	100.0000	90.0000
	<b>BBP</b>	100.0000	100.0000	100.0000	40.0000
	<b>SEP</b>	10.0000	0.0000	20.0000	20.0000
	<b>SEPA</b>	90.0000	0.0000	0.0000	0.0000
<b>M3</b> $m = 3$	<b>BB</b>	100.0000	90.0000	10.0000	30.0000
	<b>BBP</b>	100.0000	40.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	30.0000
	<b>SEPA</b>	40.0000	0.0000	90.0000	40.0000
<b>M4</b> $m = 4$	<b>BB</b>	40.0000	70.0000	0.0000	10.0000
	<b>BBP</b>	70.0000	0.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	30.0000	40.0000	70.0000
	<b>SEPA</b>	0.0000	0.0000	60.0000	20.0000
<b>M5</b> $m = 5$	<b>BB</b>	40.0000	90.0000	40.0000	40.0000
	<b>BBP</b>	30.0000	10.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	10.0000	60.0000
	<b>SEPA</b>	30.0000	0.0000	50.0000	0.0000

Tabla 8.65: Valores de *porcentual de casos de mejor solución hallada*  $\eta_s^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B3-CD2-PR1**.

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	100.0000	100.0000	100.0000	70.0000
	<b>BBP</b>	100.0000	100.0000	100.0000	40.0000
	<b>SEP</b>	10.0000	0.0000	20.0000	10.0000
	<b>SEPA</b>	90.0000	0.0000	0.0000	0.0000
<b>M3</b> $m = 3$	<b>BB</b>	100.0000	30.0000	0.0000	0.0000
	<b>BBP</b>	100.0000	20.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	40.0000	0.0000	0.0000	0.0000
<b>M4</b> $m = 4$	<b>BB</b>	20.0000	0.0000	0.0000	0.0000
	<b>BBP</b>	10.0000	0.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	0.0000	0.0000	0.0000	0.0000
<b>M5</b> $m = 5$	<b>BB</b>	0.0000	0.0000	0.0000	0.0000
	<b>BBP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	0.0000	0.0000	0.0000	0.0000

Tabla 8.66: Valores de *porcentual de casos de solución óptima hallada*  $\eta_o^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B3-CD2-PR1**.

## Capítulo 8: Experiencia computacional

$m$	Cota	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	Máquinas	0.0000	0.0000	0.6029	7.2102
	Robot	24.7405	22.2960	4.7357	1.1730
<b>M3</b> $m = 3$	Máquinas	0.0000	1.5320	12.9660	20.8420
	Robot	23.4245	2.8737	0.0000	0.0000
<b>M4</b> $m = 4$	Máquinas	0.0000	4.7506	17.6330	29.6251
	Robot	19.0506	0.4719	0.0000	0.0000
<b>M5</b> $m = 5$	Máquinas	0.0000	2.4873	19.8158	31.4116
	Robot	19.6032	1.6359	0.0000	0.0000

Tabla 8.67: Valores de *error relativo porcentual de la cota referido a la mejor cota*  $r_3^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B3-CD2-PR1**.

$m$	Cota	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	Máquinas	100.0000	100.0000	80.0000	20.0000
	Robot	0.0000	0.0000	20.0000	80.0000
<b>M3</b> $m = 3$	Máquinas	100.0000	70.0000	0.0000	0.0000
	Robot	0.0000	30.0000	100.0000	100.0000
<b>M4</b> $m = 4$	Máquinas	100.0000	20.0000	0.0000	0.0000
	Robot	0.0000	80.0000	100.0000	100.0000
<b>M5</b> $m = 5$	Máquinas	100.0000	40.0000	0.0000	0.0000
	Robot	0.0000	60.0000	100.0000	100.0000

Tabla 8.68: Valores de *porcentual de casos de mejor cota hallada*  $\eta_c^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B3-CD2-PR1**.

## Análisis de los procedimientos de acotación

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	3.1021	0.8436	11.4306	22.1119
	<b>BBP</b>	3.1021	0.8436	13.3030	39.0622
	<b>SEP</b>	4.8857	0.9448	12.5133	17.6796
	<b>SEPA</b>	3.1021	1.0389	9.1613	12.9690
<b>M3</b> $m = 3$	<b>BB</b>	0.1277	2.5393	22.5679	40.1216
	<b>BBP</b>	0.1277	9.4671	67.4587	98.2415
	<b>SEP</b>	1.2535	4.0924	14.3666	12.9197
	<b>SEPA</b>	0.1335	1.9381	5.6144	8.2415
<b>M4</b> $m = 4$	<b>BB</b>	0.2434	2.2912	16.8074	35.4416
	<b>BBP</b>	0.2350	22.8775	89.4205	157.3800
	<b>SEP</b>	2.0461	3.3889	6.9545	14.5528
	<b>SEPA</b>	0.2402	1.1064	2.8893	9.0644
<b>M5</b> $m = 5$	<b>BB</b>	0.2607	6.5481	21.4588	36.8499
	<b>BBP</b>	0.1768	54.3630	112.1927	184.8106
	<b>SEP</b>	1.8518	6.1124	9.9030	17.0464
	<b>SEPA</b>	0.1646	1.9269	4.4753	10.3551

Tabla 8.69: Valores de *error relativo porcentual de las soluciones referido a la mejor cota  $r_1^k$* , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B3-CD1-PR2**.

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	0.0000	0.0000	2.4528	8.0197
	<b>BBP</b>	0.0000	0.0000	4.0333	23.0122
	<b>SEP</b>	1.7226	0.0934	3.5370	4.2932
	<b>SEPA</b>	0.0000	0.1803	0.4665	0.2897
<b>M3</b> $m = 3$	<b>BB</b>	0.0000	0.9785	16.0110	29.5010
	<b>BBP</b>	0.0000	7.7725	57.9543	83.1026
	<b>SEP</b>	1.1247	2.5057	8.3777	4.4381
	<b>SEPA</b>	0.0058	0.3749	0.0000	0.1003
<b>M4</b> $m = 4$	<b>BB</b>	0.0246	1.1660	13.5380	23.9818
	<b>BBP</b>	0.0162	21.5645	84.2187	135.4123
	<b>SEP</b>	1.8192	2.2499	3.9621	4.9928
	<b>SEPA</b>	0.0214	0.0000	0.0000	0.0000
<b>M5</b> $m = 5$	<b>BB</b>	0.1062	4.6605	16.2698	23.9352
	<b>BBP</b>	0.0223	51.5799	103.1902	157.6703
	<b>SEP</b>	1.6936	4.2110	5.1821	6.0734
	<b>SEPA</b>	0.0102	0.1170	0.0000	0.0000

Tabla 8.70: Valores de *error relativo porcentual de las soluciones referido a la mejor solución  $r_2^k$* , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B3-CD1-PR2**.

**Capítulo 8: Experiencia computacional**

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	100.0000	100.0000	50.0000	10.0000
	<b>BBP</b>	100.0000	100.0000	60.0000	0.0000
	<b>SEP</b>	40.0000	90.0000	10.0000	20.0000
	<b>SEPA</b>	100.0000	90.0000	40.0000	80.0000
<b>M3</b> $m = 3$	<b>BB</b>	100.0000	50.0000	0.0000	0.0000
	<b>BBP</b>	100.0000	20.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	10.0000
	<b>SEPA</b>	80.0000	70.0000	100.0000	90.0000
<b>M4</b> $m = 4$	<b>BB</b>	50.0000	0.0000	0.0000	0.0000
	<b>BBP</b>	70.0000	0.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	80.0000	100.0000	100.0000	100.0000
<b>M5</b> $m = 5$	<b>BB</b>	50.0000	10.0000	0.0000	0.0000
	<b>BBP</b>	60.0000	0.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	20.0000	0.0000	0.0000
	<b>SEPA</b>	50.0000	70.0000	100.0000	100.0000

Tabla 8.71: Valores de *porcentual de casos de mejor solución hallada*  $\eta_s^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B3-CD1-PR2**.

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	100.0000	100.0000	50.0000	0.0000
	<b>BBP</b>	100.0000	100.0000	60.0000	0.0000
	<b>SEP</b>	40.0000	90.0000	10.0000	0.0000
	<b>SEPA</b>	100.0000	90.0000	0.0000	0.0000
<b>M3</b> $m = 3$	<b>BB</b>	100.0000	30.0000	0.0000	0.0000
	<b>BBP</b>	100.0000	10.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	80.0000	0.0000	0.0000	0.0000
<b>M4</b> $m = 4$	<b>BB</b>	40.0000	0.0000	0.0000	0.0000
	<b>BBP</b>	40.0000	0.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	30.0000	0.0000	0.0000	0.0000
<b>M5</b> $m = 5$	<b>BB</b>	20.0000	0.0000	0.0000	0.0000
	<b>BBP</b>	40.0000	0.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	20.0000	0.0000	0.0000	0.0000

Tabla 8.72: Valores de *porcentual de casos de solución óptima hallada*  $\eta_o^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B3-CD1-PR2**.

## Análisis de los procedimientos de acotación

$m$	Cota	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	Máquinas	0.0000	0.0000	0.0000	0.0000
	Robot	77.8440	77.8560	71.2558	68.0966
<b>M3</b> $m = 3$	Máquinas	0.0000	0.0000	0.0000	0.0000
	Robot	76.9509	70.4694	65.2062	63.8167
<b>M4</b> $m = 4$	Máquinas	0.0000	0.0000	0.0000	0.0000
	Robot	75.8156	70.8385	64.9973	57.5068
<b>M5</b> $m = 5$	Máquinas	0.0000	0.0000	0.0000	0.0000
	Robot	76.4651	69.7608	63.2121	56.5746

Tabla 8.73: Valores de *error relativo porcentual de la cota referido a la mejor cota*  $r_3^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B3-CD1-PR2**.

$m$	Cota	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	Máquinas	100.0000	100.0000	100.0000	100.0000
	Robot	0.0000	0.0000	0.0000	0.0000
<b>M3</b> $m = 3$	Máquinas	100.0000	100.0000	100.0000	100.0000
	Robot	0.0000	0.0000	0.0000	0.0000
<b>M4</b> $m = 4$	Máquinas	100.0000	100.0000	100.0000	100.0000
	Robot	0.0000	0.0000	0.0000	0.0000
<b>M5</b> $m = 5$	Máquinas	100.0000	100.0000	100.0000	100.0000
	Robot	0.0000	0.0000	0.0000	0.0000

Tabla 8.74: Valores de *porcentual de casos de mejor cota hallada*  $\eta_c^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B3-CD1-PR2**.

## Capítulo 8: Experiencia computacional

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	0.9351	1.0863	11.1952	26.6191
	<b>BBP</b>	0.9351	1.0863	15.8041	32.4448
	<b>SEP</b>	1.6094	6.4701	14.1376	14.8551
	<b>SEPA</b>	0.9351	1.5256	10.2109	11.7684
<b>M3</b> $m = 3$	<b>BB</b>	0.2737	2.0172	18.3487	46.4437
	<b>BBP</b>	0.2737	14.9223	68.5507	117.3123
	<b>SEP</b>	1.2905	4.2788	13.2835	21.5792
	<b>SEPA</b>	0.2737	1.5486	3.5397	8.6530
<b>M4</b> $m = 4$	<b>BB</b>	0.1615	2.7001	16.1093	26.2671
	<b>BBP</b>	0.0626	28.3504	87.6747	137.2457
	<b>SEP</b>	1.2635	4.3261	10.8575	12.3440
	<b>SEPA</b>	0.0989	1.0389	3.2907	4.5546
<b>M5</b> $m = 5$	<b>BB</b>	0.8013	3.9582	22.5553	40.1086
	<b>BBP</b>	0.2737	42.8504	111.3955	180.5817
	<b>SEP</b>	1.7567	5.6394	11.4606	13.1509
	<b>SEPA</b>	0.3172	2.0453	3.3232	6.7310

Tabla 8.75: Valores de *error relativo porcentual de las soluciones referido a la mejor cota  $r_1^k$* , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B3-CD2-PR2**.

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	0.0000	0.0000	1.3645	12.9915
	<b>BBP</b>	0.0000	0.0000	5.5563	18.6365
	<b>SEP</b>	0.6692	5.3081	3.9651	2.7953
	<b>SEPA</b>	0.0000	0.4366	0.5613	0.1592
<b>M3</b> $m = 3$	<b>BB</b>	0.0000	0.5755	14.3602	34.5272
	<b>BBP</b>	0.0000	13.1420	62.6929	99.7614
	<b>SEP</b>	1.0139	2.7913	9.4146	11.9650
	<b>SEPA</b>	0.0000	0.1267	0.0000	0.0000
<b>M4</b> $m = 4$	<b>BB</b>	0.0989	1.8285	12.3108	20.7905
	<b>BBP</b>	0.0000	27.2611	81.4271	126.3842
	<b>SEP</b>	1.2003	3.4428	7.3423	7.5098
	<b>SEPA</b>	0.0362	0.1810	0.0000	0.0000
<b>M5</b> $m = 5$	<b>BB</b>	0.5654	1.8832	18.6151	31.1079
	<b>BBP</b>	0.0388	40.2373	104.7855	162.9408
	<b>SEP</b>	1.5158	3.5060	7.8857	5.9855
	<b>SEPA</b>	0.0823	0.0030	0.0000	0.0000

Tabla 8.76: Valores de *error relativo porcentual de las soluciones referido a la mejor solución  $r_2^k$* , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B3-CD2-PR2**.

## Análisis de los procedimientos de acotación

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	100.0000	100.0000	60.0000	0.0000
	<b>BBP</b>	100.0000	100.0000	30.0000	0.0000
	<b>SEP</b>	40.0000	10.0000	20.0000	30.0000
	<b>SEPA</b>	100.0000	30.0000	40.0000	70.0000
<b>M3</b> $m = 3$	<b>BB</b>	100.0000	60.0000	0.0000	0.0000
	<b>BBP</b>	100.0000	30.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	100.0000	70.0000	100.0000	100.0000
<b>M4</b> $m = 4$	<b>BB</b>	60.0000	20.0000	0.0000	0.0000
	<b>BBP</b>	100.0000	0.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	30.0000	80.0000	100.0000	100.0000
<b>M5</b> $m = 5$	<b>BB</b>	20.0000	0.0000	0.0000	0.0000
	<b>BBP</b>	50.0000	10.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	20.0000	0.0000	0.0000
	<b>SEPA</b>	50.0000	70.0000	100.0000	100.0000

Tabla 8.77: Valores de *porcentual de casos de mejor solución hallada*  $\eta_s^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B3-CD2-PR2**.

$m$	Proc.	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	<b>BB</b>	100.0000	100.0000	40.0000	0.0000
	<b>BBP</b>	100.0000	100.0000	30.0000	0.0000
	<b>SEP</b>	40.0000	10.0000	20.0000	0.0000
	<b>SEPA</b>	100.0000	30.0000	0.0000	0.0000
<b>M3</b> $m = 3$	<b>BB</b>	100.0000	30.0000	0.0000	0.0000
	<b>BBP</b>	100.0000	20.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	100.0000	10.0000	0.0000	0.0000
<b>M4</b> $m = 4$	<b>BB</b>	30.0000	0.0000	0.0000	0.0000
	<b>BBP</b>	30.0000	0.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	10.0000	0.0000	0.0000	0.0000
<b>M5</b> $m = 5$	<b>BB</b>	20.0000	0.0000	0.0000	0.0000
	<b>BBP</b>	20.0000	0.0000	0.0000	0.0000
	<b>SEP</b>	0.0000	0.0000	0.0000	0.0000
	<b>SEPA</b>	20.0000	0.0000	0.0000	0.0000

Tabla 8.78: Valores de *porcentual de casos de solución óptima hallada*  $\eta_o^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B3-CD2-PR2**.

## Capítulo 8: Experiencia computacional

$m$	Cota	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	Máquinas	0.0000	0.0000	0.0000	0.0000
	Robot	83.5765	80.7093	77.4899	78.0947
<b>M3</b> $m = 3$	Máquinas	0.0000	0.0000	0.0000	0.0000
	Robot	82.5226	78.0187	75.1654	67.8644
<b>M4</b> $m = 4$	Máquinas	0.0000	0.0000	0.0000	0.0000
	Robot	83.4978	77.3726	72.5925	70.6538
<b>M5</b> $m = 5$	Máquinas	0.0000	0.0000	0.0000	0.0000
	Robot	81.4612	77.0980	72.4635	67.8861

Tabla 8.79: Valores de *error relativo porcentual de la cota referido a la mejor cota*  $r_3^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B3-CD2-PR2**.

$m$	Cota	P05	P10	P15	P20
		$n = 5$	$n = 10$	$n = 15$	$n = 20$
<b>M2</b> $m = 2$	Máquinas	100.0000	100.0000	100.0000	100.0000
	Robot	0.0000	0.0000	0.0000	0.0000
<b>M3</b> $m = 3$	Máquinas	100.0000	100.0000	100.0000	100.0000
	Robot	0.0000	0.0000	0.0000	0.0000
<b>M4</b> $m = 4$	Máquinas	100.0000	100.0000	100.0000	100.0000
	Robot	0.0000	0.0000	0.0000	0.0000
<b>M5</b> $m = 5$	Máquinas	100.0000	100.0000	100.0000	100.0000
	Robot	0.0000	0.0000	0.0000	0.0000

Tabla 8.80: Valores de *porcentual de casos de mejor cota hallada*  $\eta_c^k$ , como función del número de máquinas  $m$  y el número de piezas  $n$ . Ejemplares de la variante **B3-CD2-PR2**.

# Capítulo 9

## Conclusiones

### 9.1 Introducción

En esta tesis se resuelve un problema de manufactura de un conjunto de piezas en un sistema productivo conocido con el nombre de celda robotizada, hasta ahora inédito en la literatura. En este problema se tienen en cuenta características que no habían sido consideradas antes simultáneamente.

- ▶ Se ha adoptado el número de máquinas como una variable, a diferencia de los numerosos trabajos publicados donde este valor se asume como una constante.
- ▶ Se ha empleado un modelo de tiempos sumamente completo (inédito en la literatura) en el que es posible definir individualmente el valor de cada uno de los tiempos de carga de cada pieza en cada máquina; el valor de cada uno de los tiempos de descarga de cada pieza en cada máquina; y el valor de cada uno de los tiempos de viaje del robot entre cada par de máquinas transportando cada pieza y cuando el robot se desplaza sin pieza. Este modelo de tiempos permite fácilmente incorporar los distintos efectos que puede tener influencia en estos tiempos, como, por ejemplo, el peso y la geometría de las piezas.
- ▶ Así mismo, el modelo adoptado para los tiempos del robot permite incorporar la posibilidad de giro completo del robot. Esta característica, que se encuentra disponible en muchos de los robots actuales, permite reducir los tiempos de viaje y con esto los programas pueden ser más eficientes.
- ▶ Se considera (como en pocos trabajos publicados) la presencia de buffers para el almacenamiento de las piezas en proceso, situados entre cada par de máquinas. En este trabajo se asume que la capacidad los buffers es una variable del problema (una característica inédita en la literatura) siendo sus valores individualmente definidos por los datos del problema a resolver. El valor nulo es admitido para que pueda ser considerado el caso de no existencia de buffer.
- ▶ Con respecto a los tipos de piezas, se ha adoptado el caso más general donde las piezas son distintas. Luego, el tratamiento del caso de piezas iguales también se

admite, mediante la asignación de iguales tiempos de proceso de todas las piezas en cada máquina. Esta característica se asume en pocos trabajos.

- ▶ Se ha asumido un patrón de flujo del sistema de tipo flowshop permutativo, por ser un tipo de flujo bastante frecuente. Sin embargo, el análisis de estos sistemas productivos operando bajo otros patrones de flujo son de interés para ser tratados en investigaciones futuras.
- ▶ Se ha asumido una función objetivo de optimización de tipo minimización del makespan  $C_{max}$ . Sin embargo, la optimización de estos sistemas productivos bajo otras funciones objetivo son de interés para ser estudiados en otras investigaciones.
- ▶ No se han asumido restricciones temporales impuestas para los tiempos de espera entre procesos sucesivos y para los tiempos de espera dentro del mismo proceso. Esta investigación se ha limitado al caso de ventanas temporales ilimitadas, (como se clasificó en el apartado 1.8) por ser éstas las más corrientes en muchos de los procesos industriales, conformados, por lo general, por procesos de naturaleza mecánica. La presencia de restricciones temporales adicionales se presentan como una alternativa interesante para ser tratadas en investigaciones futuras.

## 9.2 Propuestas de procedimientos de resolución y acotación

Para el problema definido en el apartado 2.2 se ha propuesto un modelo de espacio de estados (apartado 2.3) sobre el cual se ha basado el diseño de los distintos procedimientos. En cuanto a los procedimientos de resolución diseñados, se pueden dividir en dos categorías:

- ▶ dos procedimientos exactos basados en la técnica branch and bound codificados **BB** y **BBP**,
- ▶ y dos procedimientos heurísticos basados en la determinación, de forma separada, de la secuencia de piezas y la secuencia de movimientos del robot, codificados **SEP** y **SEPA**.

Por otro lado se han probado los dos métodos de acotación siguientes:

- ▶ una cota inferior basada en los tiempos de las máquinas (fundamentada en la idea central de desprestigiar algunos de los tiempos del robot) en dos versiones:
  - ▷ para el caso en que la secuencia de piezas es dada (capítulo 3),
  - ▷ y para el caso en que la secuencia de piezas no es dada (capítulo 4);
- ▶ y una cota inferior basada en los tiempos del robot (fundamentada en la idea de desprestigiar los tiempos de las máquinas) presentada en el capítulo 5.

### 9.3 Resultados alcanzados

En este apartado se resumen los resultados obtenidos por los procedimientos de acotación y de resolución mediante los cuales se demuestra que los objetivos planteados en esta investigación han sido alcanzados satisfactoriamente.

#### 9.3.1 Resultados de los procedimientos de acotación

Los procedimientos de acotación tienen incidencia en los procedimientos exploratorios que emplean mecanismos de poda por cota. Sobre el desempeño de las dos cotas propuestas en este trabajo, el resultado ha sido el siguiente.

En los ejemplares de tamaños pequeños la cota de máquinas ha funcionado mejor siempre, sin embargo, en los ejemplares de mayor tamaño (donde hay mayor interés) ambas cotas se han complementado mutuamente. En efecto:

- ▶ en los ejemplares donde los tiempos de proceso son pequeños o del mismo orden de magnitud respecto de los tiempos de carga y descarga, se ha impuesto la cota del robot sobre la cota de máquinas;
- ▶ mientras que, en los ejemplares donde los tiempos de proceso son grandes respecto de los tiempos de carga y descarga se ha dado el resultado inverso, es decir, se ha impuesto la cota basada en los tiempos de las máquinas sobre la cota basada en los tiempos del robot.

#### 9.3.2 Resultados de los procedimientos de resolución

De los procedimientos de resolución diseñados, los procedimientos **BB** y **SEPA** se han destacado sobre el resto. En el caso de buffers nulos el procedimiento **BB** ha mostrado (en promedio) el mejor desempeño en ejemplares de todos los tamaños.

Por otro lado, en el caso de buffers no nulos los mejores desempeños se observan separados en dos subconjuntos de ejemplares:

- ▶ por un lado, el procedimiento **BB** ha mostrado el mejor desempeño en los ejemplares donde los tiempos de proceso son pequeños o del mismo orden de magnitud respecto de los tiempos de carga y descarga;
- ▶ mientras que por otro lado, el procedimiento **SEPA** ha mostrado el mejor desempeño (con diferencia notable respecto de los otros procedimientos) en los ejemplares donde los tiempos de proceso son grandes respecto de los tiempos de carga y descarga; en especial en los ejemplares de mayor tamaño (que son los más interesantes desde el punto de vista práctico).

### 9.4 Líneas futuras de investigación

Una línea de trabajo futuro puede plantearse en torno a incorporar las características que se mencionó en el apartado 9.1. Por ejemplo, se puede considerar el análisis de

sistemas productivos de tipo celda robotizada operando bajo otros patrones de flujo, la optimización de otras funciones objetivo y la introducción de restricciones temporales para los tiempos de espera entre procesos sucesivos y para los tiempos de espera dentro del mismo proceso.

Con respecto al enfoque basado en la separación de los procedimientos de determinación de la secuencia de piezas por un lado y la secuencia de movimientos del robot por el otro, en esta tesis se ha dado un paso inicial y se advierte que este enfoque no ha sido explotado suficientemente y puede dar lugar a prometedoras investigaciones. Por ejemplo, ensayando distintas heurísticas o metaheurísticas, tanto para la determinación de la secuencia de piezas, como para la determinación de los movimientos del robot.

# Referencias

- Abdulkader, M. M. S., El Beheiry, M. M., Afia, N. H. y El-Kharbotly, A. K. (2013), 'Scheduling and sequencing in four machines robotic cell: Application of genetic algorithm and enumeration techniques', *Ain Shams Engineering Journal* **4**(3), 465–474.
- Agnetis, A. (2000), 'Scheduling no-wait robotic cells with two and three machines', *European Journal of Operational Research* **123**(2), 303–314.
- Agnetis, A. y Pacciarelli, D. (2000), 'Part sequencing in three-machine no-wait robotic cells', *Operations Research Letters* **27**(4), 185–192.
- Akturk, M. S., Gultekin, H. y Karasan, O. E. (2005), 'Robotic cell scheduling with operational flexibility', *Discrete Applied Mathematics* **145**(3), 334–348.
- Amraoui, A. E., Manier, M., Moudni, A. E. y Benrejeb, M. (2013a), 'A genetic algorithm approach for a single hoist scheduling problem with time windows constraints', *Engineering Applications of Artificial Intelligence* **26**(7), 1761–1771.
- Amraoui, A. E., Manier, M., Moudni, A. E. y Benrejeb, M. (2013b), 'A linear optimization approach to the heterogeneous r-cyclic hoist scheduling problem', *Computers & Industrial Engineering* **65**(3), 360–369.
- Arbib, C. y Marinelli, F. (2003), 'The lazy cook problem, or scheduling two parallel machines to optimize vehicle utilization', *International Journal of Flexible Manufacturing Systems* **15**(4), 333–354.
- Batur, G. D., Karasan, O. E. y Akturk, M. S. (2012), 'Multiple part-type scheduling in flexible robotic cells', *International Journal of Production Economics* **135**(2), 726–740.
- Błażewicz, J., Pawlak, G. y Walter, B. (2002), 'Scheduling production tasks in a two-stage fms', *International Journal of Production Research* **40**(17), 4341–4352.
- Bocewicz, G., Nielsen, I. y Banaszak, Z. (2014), 'Automated guided vehicles fleet match-up scheduling with production flow constraints', *Engineering Applications of Artificial Intelligence* **30**, 49–62.
- Brauner, N. (2008), 'Identical part production in cyclic robotic cells: Concepts, overview and open questions', *Discrete Applied Mathematics* **156**(13), 2480–2492.
- Brauner, N. y Finke, G. (2001), 'Cycles and permutations in robotic cells', *Mathematical and Computer Modelling* **34**(5), 565–591.

## REFERENCIAS

---

- Brauner, N., Finke, G. y Gueguen, C. (1997), 'Optimal cyclic production in a robotic flow-shop with buffers', *Rapport de Recherche RR 982-I, LEIBNIZ/IMAG, Grenoble*.
- Brucker, P. y Kampmeyer, T. (2008), 'A general model for cyclic machine scheduling problems', *Discrete Applied Mathematics* **156**(13), 2561–2572.
- Che, A., Chabrol, M., Gourgand, M. y Wang, Y. (2012), 'Scheduling multiple robots in a no-wait re-entrant robotic flowshop', *International Journal of Production Economics* **135**(1), 199–208.
- Che, A. y Chu, C. (2005), 'Multi-degree cyclic scheduling of two robots in a no-wait flowshop', *IEEE Transactions on Automation Science and Engineering* **2**(2), 173–183.
- Che, A. y Chu, C. (2009), 'Multi-degree cyclic scheduling of a no-wait robotic cell with multiple robots', *European Journal of Operational Research* **199**(1), 77–88.
- Che, A., Yan, P., Yang, N. y Chu, C. (2010), 'Optimal cyclic scheduling of a hoist and multi-type parts with fixed processing times', *International Journal of Production Research* **48**(5), 1225–1243.
- Che, A., Hu, H., Chabrol, M. y Gourgand, M. (2011a), 'A polynomial algorithm for multi-robot 2-cyclic scheduling in a no-wait robotic cell', *Computers & Operations Research* **38**(9), 1275–1285.
- Che, A., Kats, V. y Levner, E. (2011b), 'Cyclic scheduling in robotic flowshops with bounded work-in-process levels', *Naval Research Logistics* **58**(1), 1–16.
- Che, A. y Yuan, W. (2008), 'Scheduling model and its algorithm for no-wait robotic cell with multiple robots', *Computer Integrated Manufacturing Systems* **3**, 525–534.
- Chtourou, S., Manier, M. y Taïcir, L. (2013), 'A hybrid algorithm for the cyclic hoist scheduling problem with two transportation resources', *Computers & Industrial Engineering* **65**(3), 426–437.
- Chu, C. (2006), 'A faster polynomial algorithm for 2-cyclic robotic scheduling', *Journal of Scheduling* **9**(5), 453–468.
- Chu, C. y Proth, J. M. (1996), 'Single machine scheduling with chain: structured precedence constraints and separation time windows', *IEEE Transactions on Robotics and Automation* **12**(6), 835–844.
- Crama, Y., Kats, V., van de Klundert, J. J. y Levner, E. (2000), 'Cyclic scheduling in robotic flowshops', *Annals of Operations Research* **96**(1-4), 97–124.
- Crama, Y. y van de Klundert, J. J. (1997), 'Cyclic scheduling of identical parts in a robotic cell', *Operations Research* **45**(6), 952–965.
- Crama, Y. y van de Klundert, J. J. (1999), 'Cyclic scheduling in 3-machine robotic flow shops', *Journal of Scheduling* **2**, 35–54.
- Dawande, M., Geismar, H. N., Pinedo, M. y Sriskandarajah, C. (2009), 'Throughput optimization in dual-gripper interval robotic cells', *IIE Transactions* **42**(1), 1–15.

## REFERENCIAS

---

- Dawande, M., Geismar, H. N., Sethi, S. P. y Sriskandarajah, C. (2005), 'Sequencing and scheduling in robotic cells: Recent developments', *Journal of Scheduling* **8**(5), 387–426.
- Dawande, M., Sriskandarajah, C. y Sethi, S. (2002), 'On throughput maximization in constant travel-time robotic cells', *Manufacturing & Service Operations Management* **4**(4), 296–312.
- Dawande, M. W., Geismar, H. N., Sethi, S. P. y Sriskandarajah, C. (2007), No-wait and interval robotic cells, in 'Throughput Optimization in Robotic Cells', Springer, 233 Spring Street, New York, NY 10013, USA, pp. 363–370.
- Deroussi, L., Gourgand, M. y Tchernev, N. (2008), 'A simple metaheuristic approach to the simultaneous scheduling of machines and automated guided vehicles', *International Journal of Production Research* **46**(8), 2143–2164.
- Drobouchevitch, I. G., Geismar, H. N. y Sriskandarajah, C. (2010), 'Throughput optimization in robotic cells with input and output machine buffers: A comparative study of two key models', *European Journal of Operational Research* **206**(3), 623–633.
- Drobouchevitch, I. G., Sethi, S. P. y Sriskandarajah, C. (2006), 'Scheduling dual gripper robotic cell: One-unit cycles', *European Journal of Operational Research* **171**(2), 598–631.
- Drobouchevitch, I. G., Sethi, S., Sidney, J. y Sriskandarajah, C. (2004), 'A note on scheduling multiple parts in two-machine dual gripper robotic cell: Heuristic algorithm and performance guarantee', *International Journal of Operations and Quantitative Management* **10**(4), 297–314.
- Elmi, A. y Topaloglu, S. (2013), 'A scheduling problem in blocking hybrid flow shop robotic cells with multiple robots', *Computers & Operations Research* **40**(10), 2543–2555.
- Erol, R., Sahin, C., Baykasoglu, A. y Kaplanoglu, V. (2012), 'A multi-agent based approach to dynamic scheduling of machines and automated guided vehicles in manufacturing systems', *Applied Soft Computing* **12**(6), 1720–1732.
- Fathian, M., Kamalabadi, I. N., Heydari, M., Farughi, H. y Naseri, F. (2013), 'Applying metaheuristic algorithms for output rate analysis in two-machine robotic manufacturing cells.', *International Journal of Advanced Robotic Systems* **10**, 1–11.
- Fazel Zarandi, M. H., Mosadegh, H. y Fattahi, M. (2013), 'Two-machine robotic cell scheduling problem with sequence-dependent setup times', *Computers & Operations Research* **40**(5), 1420–1434.
- Feng, J., Che, A. y Wang, N. (2014), 'Bi-objective cyclic scheduling in a robotic cell with processing time windows and non-euclidean travel times', *International Journal of Production Research* **52**(9), 2505–2518.
- Foumani, M., Ibrahim, M. Y. y Gunawan, I. (2013), Scheduling dual gripper robotic cells with a hub machine, in 'IEEE International Symposium on Industrial Electronics', IEEE, pp. 1–6.

## REFERENCIAS

---

- Foumani, M. y Jenab, K. (2013), 'Analysis of flexible robotic cells with improved pure cycle', *International Journal of Computer Integrated Manufacturing* **26**(3), 201–215.
- Galante, G. y Passannanti, G. (2006), 'Minimizing the cycle time in serial manufacturing systems with multiple dual-gripper robots', *International Journal of Production Research* **44**(4), 639–652.
- Geismar, H. N. (2003), Advanced problems in robotic cell scheduling: approximations, parallel machines, and multiple robots, PhD thesis, University of Texas, Dallas.
- Geismar, H. N., Dawande, M. y Sriskandarajah, C. (2004a), 'Robotic cells with parallel machines: Throughput maximization in constant travel-time cells', *Journal of Scheduling* **7**(5), 375–395.
- Geismar, H. N., Dawande, M. y Sriskandarajah, C. (2005a), 'Approximation algorithms for k-unit cyclic solutions in robotic cells', *European Journal of Operational Research* **162**(2), 291–309.
- Geismar, H. N., Dawande, M. y Sriskandarajah, C. (2006), 'Throughput optimization in constant travel-time dual gripper robotic cells with parallel machines', *Production and Operations Management* **15**(2), 311–328.
- Geismar, H. N., Pinedo, M. y Sriskandarajah, C. (2008), 'Robotic cells with parallel machines and multiple dual gripper robots: a comparative overview', *IIE Transactions* **40**(12), 1211–1227.
- Geismar, H. N., Sethi, S. P., Sidney, J. B. y Sriskandarajah, C. (2005b), 'A note on productivity gains in flexible robotic cells', *International Journal of Flexible Manufacturing Systems* **17**(1), 5–21.
- Geismar, H. N., Sriskandarajah, C. y Ramanan, N. (2004b), 'Increasing throughput for robotic cells with parallel machines and multiple robots', *IEEE Transactions on Automation Science and Engineering* **1**(1), 84–89.
- Geismar, N., Dawande, M. y Sriskandarajah, C. (2011), 'Productivity improvement from using machine buffers in dual-gripper cluster tools', *IEEE Transactions on Automation Science and Engineering* **8**(1), 29–41.
- Geismar, N., Manoj, U. V., Sethi, A. y Sriskandarajah, C. (2012), 'Scheduling robotic cells served by a dual-arm robot', *IIE Transactions* **44**(3), 230–248.
- Gnanavel Babu, A., Jerald, J., Noorul Haq, A., Muthu Luxmi, V. y Vigneswaralu, T. P. (2010), 'Scheduling of machines and automated guided vehicles in fms using differential evolution', *International Journal of Production Research* **48**(16), 4683–4699.
- Gultekin, H., Akturk, M. S. y Ekin Karasan, O. (2007), 'Scheduling in a three-machine robotic flexible manufacturing cell', *Computers & Operations Research* **34**(8), 2463–2477.
- Gultekin, H., Akturk, M. S. y Karasan, O. E. (2006a), 'Cyclic scheduling of a 2-machine robotic cell with tooling constraints', *European Journal of Operational Research* **174**(2), 777–796.

## REFERENCIAS

---

- Gultekin, H., Akturk, M. S. y Karasan, O. E. (2006b), Scheduling in a three-machine flexible robotic cell, in 'Information Control Problems in Manufacturing', Vol. 12, pp. 101–106.
- Gultekin, H., Akturk, M. S. y Karasan, O. E. (2008), 'Bicriteria robotic cell scheduling', *Journal of Scheduling* **11**(6), 457–473.
- Gultekin, H., Akturk, M. S. y Karasan, O. E. (2010), 'Bicriteria robotic operation allocation in a flexible manufacturing cell', *Computers & Operations Research* **37**(4), 779–789.
- Gultekin, H., Ekin Karasan, O. y Akturk, M. S. (2009), 'Pure cycles in flexible robotic cells', *Computers & Operations Research* **36**(2), 329–343.
- Hall, N. G., Kamoun, H. y Sriskandarajah, C. (1997), 'Scheduling in robotic cells: Classification, two and three machine cells', *Operations Research* **45**(3), 421–439.
- Hall, N. G., Kamoun, H. y Sriskandarajah, C. (1998), 'Scheduling in robotic cells: Complexity and steady state analysis', *European Journal of Operational Research* **109**(1), 43–65.
- Hall, N. G. y Sriskandarajah, C. (1996), 'A survey of machine scheduling problems with blocking and no-wait in process', *Operations Research* **44**(3), 510–525.
- Hindi, K. S. y Fleszar, K. (2004), 'A constraint propagation heuristic for the single-hoist, multiple-products scheduling problem', *Computers & Industrial Engineering* **47**(1), 91–101.
- Ignall, E. y Schrage, L. (1965), 'Application of the branch and bound technique to some flow-shop scheduling problems', *Operations Research* **13**(3), 400–412.
- Jégou, D., Kim, D., Baptiste, P. y Lee, K. H. (2006), 'A contract net based intelligent agent system for solving the reactive hoist scheduling problem', *Expert Systems with Applications* **30**(2), 156–167.
- Jung, C. y Lee, T. (2012), 'An efficient mixed integer programming model based on timed petri nets for diverse complex cluster tool scheduling problems', *IEEE Transactions on Semiconductor Manufacturing* **25**(2), 186–199.
- Kamalabadi, I. N., Gholami, S. y Mirzaei, A. H. (2007), Considering a cyclic multiple-part type three-machine robotic cell problem, in '2007 IEEE International Conference on Industrial Engineering and Engineering Management', IEEE, pp. 704–708.
- Kamalabadi, I. N., Gholami, S. y Mirzaei, A. H. (2008), 'A new solution for the cyclic multiple-part type three-machine robotic cell problem based on the particle swarm meta-heuristic', *Journal of Industrial and Systems Engineering* **1**(4), 304–317.
- Kamoun, H., Hall, N. G. y Sriskandarajah, C. (1999), 'Scheduling in robotic cells: Heuristics and cell design', *Operations Research* **47**(6), 821–835.
- Kamoun, H. y Sriskandarajah, C. (1993), 'The complexity of scheduling jobs in repetitive manufacturing systems', *European Journal of Operational Research* **70**(3), 350–364.

## REFERENCIAS

---

- Kats, V. y Levner, E. (2006), A polynomial algorithm for 2-cyclic robotic scheduling, in 'MICAI 2006: Advances in Artificial Intelligence', Springer, New York, pp. 439–449.
- Kats, V. y Levner, E. (2009), 'A polynomial algorithm for 2-cyclic robotic scheduling: A non-euclidean case', *Discrete Applied Mathematics* **157**(2), 339–355.
- Kats, V. y Levner, E. (2011a), 'A faster algorithm for 2-cyclic robotic scheduling with a fixed robot route and interval processing times', *European Journal of Operational Research* **209**(1), 51–56.
- Kats, V. y Levner, E. (2011b), 'Parametric algorithms for 2-cyclic robot scheduling with interval processing times', *Journal of Scheduling* **14**(3), 267–279.
- Kim, M. y Lee, Y. H. (2012), 'Mip models and hybrid algorithm for minimizing the makespan of parallel machines scheduling problem with a single server', *Computers & Operations Research* **39**(11), 2457–2468.
- Kimiagari, A. M. y Mosadegh, H. (2011), 'Two-machine robotic cell considering different loading and unloading times', *Journal of Industrial Engineering International* **7**(14), 41–52.
- Kirkpatrick, S., Gelatt, C. D. y Vecchi, M. P. (1983), 'Optimization by simulated annealing', *Science* **220**(4598), 671–680.
- Kise, H., Shioyama, T. y Ibaraki, T. (1991), 'Automated two-machine flowshop scheduling: A solvable case', *IIE transactions* **23**(1), 10–16.
- Kubiak, W. (1989), 'A pseudo-polynomial algorithm for a two-machine no-wait jobshop scheduling problem', *European Journal of Operational Research* **43**(3), 267–270.
- Kumar, S., Ramanan, N. y Sriskandarajah, C. (2005), 'Minimizing cycle time in large robotic cells', *IIE Transactions* **37**(2), 123–136.
- Lacomme, P., Larabi, M. y Tchernev, N. (2013), 'Job-shop based framework for simultaneous scheduling of machines and automated guided vehicles', *International Journal of Production Economics* **143**(1), 24–34.
- Lawler, E. L., Lenstra, J. K., Kan, A. H. G. R. y Shmoys, D. B. (1985), *The traveling salesman problem: a guided tour of combinatorial optimization*, Vol. 3, Wiley New York.
- Lebacque, V. y Brauner, N. (2005), Two-machine flowshops with material handling, Technical Report 139, Cahier du laboratoire Leibniz.
- Lee, C. y Strusevich, V. A. (2005), 'Two-machine shop scheduling with an uncapacitated interstage transporter', *IIE Transactions* **37**(8), 725–736.
- Leung, J. M. Y. y Levner, E. (2006), An efficient algorithm for multi-hoist cyclic scheduling with fixed processing times, *Operations Research Letters* **34**(4), 465–472.
- Levner, E. y Kats, V. (2009), A parametric algorithm for 2-cyclic robotic scheduling with interval processing times, in 'Information Control Problems in Manufacturing', Vol. 13, pp. 780–785.

## REFERENCIAS

---

- Levner, E., Kats, V., Alcaide López de Pablo, D. y Cheng, T. C. E. (2010), 'Complexity of cyclic scheduling problems: A state-of-the-art survey', *Computers & Industrial Engineering* **59**(2), 352–361.
- Levner, E., Kats, V. y Levit, V. E. (1997), 'An improved algorithm for cyclic flowshop scheduling in a robotic cell', *European Journal of Operational Research* **97**(3), 500–508.
- Lim, S.-Y., Park, Y.-J., Lee, H. y Hur, S. (2014), 'A real-time scheduling method for the cluster tool with wafer transfer delay', *International Journal of Production Research* **52**(4), 934–946.
- Logendran, R. y Sriskandarajah, C. (1996), 'Sequencing of robot activities and parts in two-machine robotic cells', *International Journal of Production Research* **34**(12), 3447–3463.
- Lushchakova, I. N., Soper, A. J. y Strusevich, V. A. (2009), 'Transporting jobs through a two-machine open shop', *Naval Research Logistics* **56**(1), 1–18.
- Mangione, F., Brauner, N. y Penz, B. (2003), Three-tank hoist scheduling problem with unbounded or zero-width processing windows, in '8th International Workshop on Project Management and Scheduling-Abstracts', pp. 253–256.
- Mateo, M. (2001), Procedimientos de secuenciación y programación en un sistema productivo de estaciones en serie con transportadores asíncronos de material, PhD thesis, Universitat Politècnica de Catalunya.
- Mateo, M., Companys, R. y Bautista, J. (2002), Resolution of graphs with bounded cycle time for the cyclic hoist scheduling problem, in '8th International Workshop on Project Management and Scheduling', pp. 257–260.
- McCormick, S. T., Pinedo, M. L., Shenker, S. y Wolf, B. (1989), 'Sequencing in an assembly line with blocking to minimize cycle time', *Operations Research* **37**(6), 925–935.
- Pan, X. Y., Wu, J., Zhang, Q. W., Lai, D., Xie, H. L. y Zhang, C. (2013), 'A case study of agv scheduling for production material handling', *Applied Mechanics and Materials* **411**, 2351–2354.
- Pinedo, M. L. (2012), *Scheduling: theory, algorithms, and systems*, 4 edn, Springer Science, New York.
- Saez, P. (2009), Studies on 2-cyclic robotic scheduling, in 'Computers & Industrial Engineering, 2009. CIE 2009. International Conference on', IEEE, pp. 149–152.
- Seo, J. y Lee, T. (2002), 'Steady-state analysis and scheduling of cyclic job shops with overtaking', *International Journal of Flexible Manufacturing Systems* **14**(4), 291–318.
- Sethi, S. P., Sidney, J. B. y Sriskandarajah, C. (2001), 'Scheduling in dual gripper robotic cells for productivity gains', *IEEE Transactions on Robotics and Automation* **17**(3), 324–341.
- Sethi, S. P., Sriskandarajah, C., Sorger, G., Błażewicz, J. y Kubiak, W. (1992), 'Sequencing of parts and robot moves in a robotic cell', *International Journal of Flexible Manufacturing Systems* **4**(3-4), 331–358.

- Soukhal, A. y Martineau, P. (2005), 'Resolution of a scheduling problem in a flowshop robotic cell', *European Journal of Operational Research* **161**(1), 62–72.
- Sriskandarajah, C., Drobouchevitch, I., Sethi, S. P. y Chandrasekaran, R. (2004), 'Scheduling multiple parts in a robotic cell served by a dual-gripper robot', *Operations Research* **52**(1), 65–82.
- Sriskandarajah, C., Hall, N. G. y Kamoun, H. (1998), 'Scheduling large robotic cells without buffers', *Annals of Operations Research* **76**, 287–321.
- Sriskandarajah, C. y Sethi, S. P. (1989), 'Scheduling algorithms for flexible flowshops: worst and average case performance', *European Journal of Operational Research* **43**(2), 143–160.
- Steiner, G. y Xue, Z. (2005), 'Scheduling in reentrant robotic cells: Algorithms and complexity', *Journal of Scheduling* **8**(1), 25–48.
- Tonke, D. y Lee, T. (2012), Scheduling of a dual-armed cluster tool with two independent robot arms, in 'Automation Science and Engineering (CASE), 2012 IEEE International Conference on', IEEE, pp. 200–205.
- Tzeng, J., Liu, T. y Chou, J. (2006), Applications of multi-objective evolutionary algorithms to cluster tool scheduling, in 'Innovative Computing, Information and Control, 2006. ICICIC'06. First International Conference on', Vol. 2, IEEE, pp. 531–5534.
- Werner, F. y Kravchenko, S. A. (2010), 'Scheduling with multiple servers', *Automation and Remote Control* **71**(10), 2109–2121.
- Yan, P., Che, A., Li, P. y Yang, N. (2010a), 'Improved genetic algorithm for robotic cell scheduling problem with flexible processing times', *Chinese Journal of Computer Integrated Manufacturing Systems* **16**(2), 404–410.
- Yan, P., Chu, C., Che, A. y Yang, N. (2008), An algorithm for optimal cyclic scheduling in a robotic cell with flexible processing times, in 'Industrial Engineering and Engineering Management, 2008. IEEM 2008. IEEE International Conference on', IEEE, pp. 153–157.
- Yan, P., Chu, C., Yang, N. y Che, A. (2010b), 'A branch and bound algorithm for optimal cyclic scheduling in a robotic cell with processing time windows', *International Journal of Production Research* **48**(21), 6461–6480.
- Yan, P., Yang, N. y Che, A. (2010c), 'Hybrid heuristic algorithm for the scheduling problem in robotic cell with makespan criterion', *Chinese Journal of Computer Integrated Manufacturing Systems* **4**(16), 847–854.
- Yildiz, S., Karasan, O. E. y Akturk, M. S. (2012), 'An analysis of cyclic scheduling problems in robot centered cells', *Computers & Operations Research* **39**(6), 1290–1299.
- Zahrouni, W. y Kamoun, H. (2012), 'Sequencing and scheduling in a three-machine robotic cell', *International Journal of Production Research* **50**(10), 2823–2835.

## REFERENCIAS

---

- Zhang, Q., Manier, H. y Manier, M. (2012), A hybrid metaheuristic algorithm for flexible job-shop scheduling problems with transportation constraints, *in* 'Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference', ACM, pp. 441–448.
- Zhou, Z., Che, A. y Yan, P. (2012), 'A mixed integer programming approach for multi-cyclic robotic flowshop scheduling with time window constraints', *Applied Mathematical Modelling* **36**(8), 3621–3629.
- Zhou, Z. y Liu, J. (2008), 'A heuristic algorithm for the two-hoist cyclic scheduling problem with overlapping hoist coverage ranges', *IIE Transactions* **40**(8), 782–794.