3 MODELADO DEL PROBLEMA

En el presente capítulo se ofrecen modelos MILP de los 4 escenarios que posteriormente se resuelven con el algoritmo genético. En el apartado 3.1 se detallan las hipótesis que se han hecho. Con estas consideraciones se facilita información acerca del entorno de producción y se desgranan las limitaciones con las que se modela el problema.

Por último, en el apartado 3.2 se ofrece el modelado de cada uno de los escenarios. Para ello, se proporciona una notación común a todos los modelos y posteriormente se presentan y se explican las restricciones de cada modelo.

3.1 Hipótesis y función objetivo

En el presente trabajo se trata el problema con lotes de tamaño consistente y permitiendo holguras. Es decir, se pueden procesar sublotes de un trabajo distinto entre 2 sublotes del mismo trabajo. Las rutas de los trabajos son parciales. Esto significa que un trabajo puede tener una ruta que no contiene todas las máquinas.

A continuación se ofrecen 4 modelos con características diferentes en cuanto a tiempos de setup y restricciones temporales (ventanas temporales o turnos). En la Tabla 1 se resumen las características.

Modelo	**Tiempos de Setup**	**Restricción de turnos**
LSJSP	Independientes de la secuencia	No
LSJSP con turnos	Independientes de la secuencia	Si
LSJSP Setup dependiente	Dependientes de la secuencia	No
LSJSP Setup dependiente con turnos	Dependientes de la secuencia	Si

Tabla 1. Modelos matemáticos desarrollados para el LSJSP.

Se consideran las siguientes hipótesis en el modelado del problema:

- Cada trabajo j con una ruta parcial asignada R_j se considera un lote.

- Todos los lotes están disponibles en el instante inicial.
- Todas las máquinas están disponibles en el instante inicial.
- Se permiten holguras entre sublotes de diferentes trabajos (intermingling*)*.
- No se permiten interrupciones (preemption*)* en el procesado de sublotes. Es decir, una vez que un lote ha empezado a procesarse debe terminarse sin interrupciones.
- Los tiempos de transferencia de los sublotes entre máquinas se consideran despreciables e iguales a 0.
- Las máquinas no pueden procesar más de un sublote a la vez.
- El *buffer* para el trabajo en proceso es ilimitado. Es decir, se considera que el sistema de producción tiene capacidad infinita.
- No se consideran inventarios iniciales, finales o entre estaciones.
- No se consideran fallos en las máquinas ni paradas por mantenimiento.
- Los tiempos de setup no son anticipatorios. Esto significa que no se puede realizar el setup antes de que se libere el sublote correspondiente.

Respecto a la métrica de rendimiento. La función objetivo considerada es minimizar el makespan, es decir, el tiempo comprendido entre el inicio de la primera operación y la finalización de la última operación.

3.2 Modelado MILP

El modelado del problema se basa en el modelo propuesto por Manne [21]. Es conocido como modelo disyuntivo, ya que se basa en el grafo disyuntivo. Se ha probado que es el modelo más eficiente (mejora más rápido) para instancias pequeñas en entornos tipo Job Shop [22].

3.2.1 Notación

Naturaleza	**Notación**	**Definición**
Constantes y Parámetros	J	Número de trabajos o productos.
	M	Número de máquinas
	U	Número de sublotes máximo.
	d_j	Demanda de cada trabajo j
	p_{mj}	Tiempo de procesado de cada unidad del trabajo j en la máquina m
	R_j	Ruta del trabajo j
	SHT	Duración de un turno
	s_{mjk}	Setup en la máquina m si un lote del trabajo j precede directamente a

		un lote del trabajo k

Variables continuas	C	Makespan
	x_{mju}	Instante en el que el lote u del trabajo j empieza a procesarse en la máquina m
	y_{mju}	Instante en el que el setup asociado al lote u del trabajo j empieza a procesarse en la máquina m
	p_{mju}	Tiempo de procesado del lote u del trabajo j empieza a procesarse en la máquina m
	c_{mju}	Makespan del lote u del trabajo j en la máquina m

Variables enteras	q_{ju}	Tamaño del sublote u del trabajo j
binaria	z_{mklju}	Variable binaria que vale 1 si el lote l del trabajo k precede al lote u del trabajo j en la máquina m
	z_{mklju}	Variable binaria que vale 1 si el lote l del trabajo k precede INMEDIATAMENTE al lote u del trabajo j en la máquina m
binaria	Q_{ju}	Variable binaria que vale 1 si el tamaño del lote q_{ju} es mayor que 0
binaria	W_{moju}	Variable binaria que vale 1 si el lote ju para por la máquina m en el turno o

Usadas para la modelización de turnos

Usadas para la inclusión de tiempos de setup dependientes de la secuencia

3.2.2 LSJSP

A continuación se presenta el modelo de programación lineal entera mixta para el caso más sencillo. Consta de 14 expresiones: 1 función objetivo y 13 restricciones, incluyendo las restricciones que determinan la naturaleza de las variables.

Objetivo

$\min C$ (1)

Restricciones

$y_{R^h ju} \geq x_{R^{h-1} j,j,u} + p_{R^{h-1} j,j,u}$ $\quad \forall\, h \in (2, \dots, M)\ in\ R_j;\ \forall\, j \in J;\ \forall\, u \in U$ (2)

$y_{mju} \geq x_{mj,u-1} + p_{mj,u-1}$ $\quad \forall\, m \in (M \wedge R_j);\ \forall\, j \in J;\ \forall\, u \in U\backslash\{0\}$ (3)

$z_{mklju} + z_{mjukl} = 1$ $\quad \forall\, m \in (M \wedge R_j \wedge R_k); \forall j, k \in J; \forall u, l \in U$
$j \neq k\ \vee u \neq l$ (4)

$y_{mju} + V * (1 - z_{mklju}) \geq (x_{mkl} + p_{mkl})$ $\quad \forall\, m \in (M \wedge R_j \wedge R_k); \forall j, k \in J; \forall u, l \in U$
$j \neq k\ \vee u \neq l$ (5)

$x_{mju} \geq y_{mju} + s_{mj} \cdot Q_{ju}$ $\quad \forall\, m \in (M \wedge R_j);\ \forall\, j \in J;\ \forall\, u \in U$ (6)

$C \geq x_{mju} + p_{mju}$ $\quad \forall\, m \in M;\ \forall\, j \in J;\ \forall\, u \in U$ (7)

$\sum_{u}^{U} q_{ju} = d_j$ $\quad \forall\, j \in J$ (8)

$q_{ju} \leq V \cdot Q_{ju}$ $\quad \forall\, j \in J;\ \forall\, u \in U$ (9)

$Q_{ju} \leq q_{ju}$ $\quad \forall\, j \in J;\ \forall\, u \in U$ (10)

$p_{mju} = p_{mj} \cdot q_{ju}$ $\quad \forall\, m \in M;\ \forall\, j \in J;\ \forall\, u \in U$ (11)

$x_{mju} \geq 0$ $\quad \forall\, m \in M;\ \forall\, j \in J;\ \forall\, u \in U$ (12)

$y_{mju} \geq 0$ $\quad \forall\, m \in M;\ \forall\, j \in J;\ \forall\, u \in U$ (13)

$z_{mklju} \in \{0,1\}$ $\quad \forall\, m \in M;\ \forall\, k, j \in J, k \neq j$ (14)

$Q_{ju} \in \{0,1\}$ $\quad \forall\, j \in J;\ \forall\, u \in U$ (15)

La restricción (2) describe la ruta de trabajo de cada producto j. Se impone que el *setup* asociado a un sublote no puede comenzar en la siguiente máquina hasta que no haya terminado de procesarse en la máquina anterior.

La restricción (3) indica que en cada máquina, los sublotes de cada producto j deben procesarse en orden. El sublote número $x - 1$ debe preceder al lote número x.

Las restricciones (4) y (5) describen la precedencia entre sublotes en cada máquina. En (4) se indica que un sublote siempre precede a otro sublote en la misma máquina, siempre que sean sublotes con igual número pero asociados a distintos trabajos o sublotes con distinto número pero asociados al mismo trabajo. En ningún caso ambos sublotes pueden estar asociados al mismo trabajo y tener el mismo índice, pues se estaría imponiendo que el propio sublote se precede a sí mismo, lo cual lógicamente no es posible. En (5) se especifica que, en cada máquina, el *setup* del sublote sucesor tiene que empezar después de que acabe de procesarse el sublote predecesor.

La restricción (6) describe la ejecución del *setup* para el procesado de cada sublote. Si el tamaño del sublote no es 0, entonces se ejecuta el *setup* y el procesado comenzará cuando este termine. En otras palabras, todos los sublotes tienen que empezar a procesarse cuando haya acabado su *setup* siempre y cuando el tamaño del lote no sea 0.

La restricción (7) describe el *makespan.* Simplemente sirve para calcular el valor de la función objetivo y por ende establecer la dirección de mejora.

La restricción (8) indica que todas las unidades de un lote o producto han de ser repartidas en sublotes. Por lo tanto, la suma del tamaño de los sublotes ha de ser el número total de unidades que componen la orden de trabajo del producto.

Las restricciones (9) y (10) son fundamentales para la ejecución de los *setup.* Son complementarias a la restricción (6). Mediante la variable aleatoria Q_{ju} se indica si el sublote u asociado al producto j va a contener alguna unidad.

Al modelo se le indica el número máximo de sublotes que se quieren considerar, sin embargo puede que lo óptimo no sea considerar ese número de sublotes sino un número menor. Por ejemplo, se podría considerar una división máxima en 10 sublotes pero que para un producto lo óptimo fuese dividir el lote en 3 sublotes y para otro producto dividirlo en 5 sublotes.

La restricción (11) describe el tiempo de proceso de cada sublote en cada máquina, teniendo en cuenta el tamaño de este y el tiempo de proceso unitario del producto en la correspondiente máquina. En el momento en el que el tamaño de los sublotes es consistente, esta restricción es necesaria, pues cada sublote tendrá un tiempo de procesamiento diferente, dependiendo del tamaño del sublote, la máquina y el producto.

Las restricciones (12) a (15) describen la naturaleza de las variables. Las restricciones (12) y (13) indican que los instantes temporales iniciales de la ejecución de los *setups* y del procesado de los sublotes tienen que ser positivos. Por ende, el *makespan* será siempre positivo, por cómo está definido en (7). Las restricciones (14) y (15) indican la naturaleza binaria de las variables z y Q.

3.2.3 LSJSP con turnos

Este modelo incluye la restricción de turnos. Todos los lotes deben comenzar y acabar su procesado dentro de un turno (ventana temporal). Este modelo incluye 18 restricciones.

Objetivo

$$\min C \qquad (16)$$

Restricciones

$$y_{R^h ju} \geq x_{R^{h-1}j,j,u} + p_{R^{h-1}j,j,u} \qquad \forall\, h \in (2,\dots,M)\; in\; R_j;\; \forall\, j \in J;\; \forall\, u \in U \qquad (17)$$

$$y_{mju} \geq x_{mj,u-1} + p_{mj,u-1} \qquad \forall\, m \in (M \wedge R_j);\; \forall\, j \in J;\; \forall\, u \in U\backslash\{0\} \qquad (18)$$

$$z_{mklju} + z_{mjukl} = 1 \qquad \forall\, m \in (M \wedge R_j \wedge R_k); \forall j,k \in J; \forall u,l \in U \quad j \neq k \vee u \neq l \qquad (19)$$

$$y_{mju} + V * (1 - z_{mklju}) \geq (x_{mkl} + p_{mkl}) \qquad \forall\, m \in (M \wedge R_j \wedge R_k); \forall j,k \in J; \forall u,l \in U \quad j \neq k \vee u \neq l \qquad (20)$$

$$x_{mju} \geq y_{mju} + s_{mj} \cdot Q_{ju} \qquad \forall\, m \in (M \wedge R_j);\; \forall\, j \in J;\; \forall\, u \in U \qquad (21)$$

$$C \geq x_{mju} + p_{mju} \qquad \forall\, m \in M;\; \forall\, j \in J;\; \forall\, u \in U \qquad (22)$$

$$\sum_{u}^{U} q_{ju} = d_j \qquad \forall\, j \in J \qquad (23)$$

$$q_{ju} \leq V \cdot Q_{ju} \qquad \forall\, j \in J;\; \forall\, u \in U \qquad (24)$$

$$Q_{ju} \leq q_{ju} \qquad \forall\, j \in J;\; \forall\, u \in U \qquad (25)$$

$$p_{mju} = p_{mj} * q_{ju} \qquad \forall\, m \in M;\; \forall\, j \in J;\; \forall\, u \in U \qquad (26)$$

$$\sum_{j=0}^{J}\sum_{u=0}^{U}(s_{mj} + p_{mju}) \cdot W_{moju} \leq SHT \qquad \forall\, m \in M;\; \forall\, o \in O \qquad (27)$$

$$\sum_{o=0}^{O} W_{moju} = Q_{ju} \qquad \forall\, m \in (M \wedge R_j);\; \forall\, j \in J;\; \forall\, u \in U \qquad (28)$$

$$y_{mju} \geq W_{moju} \cdot o \cdot SHT \qquad \forall\, m \in M;\; \forall\, o \in O;\; \forall\, j \in J;\; \forall\, u \in U \qquad (29)$$

$$(x_{mju} + p_{mju}) \cdot W_{moju} \leq (o+1) \cdot SHT \qquad \forall\, m \in M;\; \forall\, o \in O;\; \forall\, j \in J;\; \forall\, u \in U \qquad (30)$$

$$x_{mju} \geq 0 \qquad \forall\, m \in M;\; \forall\, j \in J;\; \forall\, u \in U \qquad (31)$$

$$y_{mju} \geq 0 \qquad \forall\, m \in M;\; \forall\, j \in J;\; \forall\, u \in U \qquad (32)$$

$$z_{mklju} \in \{0,1\} \qquad \forall\, m \in M;\; \forall\, k,j \in J, k \neq j \qquad (33)$$

$$Q_{ju} \in \{0,1\} \qquad \forall\, j \in J;\; \forall\, u \in U \qquad (34)$$

El modelo es idéntico al anterior salvo por las restricciones 27 – 30.

La restricción (27) indica que el tiempo de procesado de los lotes que pasan por la máquina m en el turno o no puede ser mayor al tiempo determinado previamente de duración de un turno (SHT).

La restricción (28) obliga a asociar a cada lote que no está "vacío" al menos un turno en cada máquina m por la que tiene que pasar ese lote. Es decir, si a ese lote se le asocian unidades a procesar, se debe asegurar se asocia con al menos un turno en cada máquina que está en su ruta de proceso.

Las restricciones (29) y (30) indican la ventana temporal en la que se puede procesar el lote. Esto se hace limitando el instante de inicio del setup y el instante de finalización del procesado del lote en máquina perteneciente a su ruta de proceso.

3.2.4 LSJSP con tiempos de setup dependientes de la secuencia

En este MILP se incorporan restricciones adicionales (en total se tienen 19 restricciones) para tener en cuenta los tiempos de setup que dependen de la secuencia de las operaciones. Esta característica es crucial en entornos productivos donde el tiempo de preparación de las máquinas o recursos varía según el orden en que se procesen los trabajos. Este enfoque mejora la capacidad de adaptación del modelo a diferentes escenarios industriales, facilitando su aplicación en una amplia gama de sectores donde los tiempos de setup secuenciales son una variable determinante.

En el modelo LSJSP base, mediante la restricción (4) se conoce si un trabajo precede a otro pero no de forma inmediata. Entre un lote predecesor y otro sucesor, en la misma máquina, pueden existir uno o más lotes que se procesan entre estos dos. A modo de ejemplo: en la Figura 11 se visualiza claramente esta característica. Tras resolver el modelo para una determinada instancia, se tiene $z_{11001} = 1$ y $z_{10110} = 0$, por lo tanto se conoce que el lote 0 (L0) del trabajo 1 (P1) precede al lote 1 (L1) del trabajo 0 (P0), pero no se conoce si esta precedencia es inmediata. Uno o varios trabajos pueden procesarse entre estos dos.

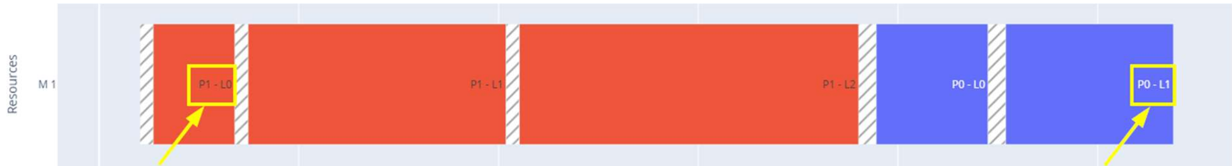

Figura 11. Precedencias en el modelo LSJSP. Se conoce que P1-L0 precede a P0-L1 pero no si lal precedencia es inmediata. En este caso, otros lotes se procesan entre estos 2. [Fuente: elaboración propia].

Para superar esta limitación y disponer de información exacta sobre las precedencias de los trabajos, en este modelo se introducen 2 trabajos "dummy" o ficticios para representar los nodos iniciales y finales. Estos trabajos ficticios no se dividen en varios lotes, por lo tanto solo se les asigna uno.

Se modifican los siguientes conjuntos:

- $J = \{0, 1, \dots, j+1\}$ siendo $\{0, j+1\}$ trabajos ficticios cuya ruta de proceso incluye todas las máquinas.
- $U_j = \{U_0, U_1, \dots, U_{\{j+1\}}\}$ siendo $|U_0| = \left|U_{\{j+1\}}\right| = 1$ (un lote para cada trabajo ficticio en cada máquina).

Además se añade la variable c_{mju} para facilitar la inclusión de restricciones.

Objetivo

$$\min C \tag{35}$$

Restricciones

$$y_{R^h ju} \geq x_{R^{h-1}j,j,u} + p_{R^{h-1}j,j,u} \quad \forall\ h\ \in (2, \dots, M)\ in\ R_j;\ \forall\ j\ \in\ J;\ \forall\ u\ \in U \tag{36}$$

$$y_{mju} \geq x_{mj,u-1} + p_{mj,u-1} \quad \forall\ m\ \in\ \left(M\ \wedge R_j\right);\ \forall\ j\ \in\ J;\ \forall\ u\ \in\ U \backslash \{0\} \tag{37}$$

$$y_{mju} \geq (x_{mkl} + p_{mkl}) - V * \left(1 - z_{mklju}\right) \quad \forall\ m \in \left(M \wedge R_j \wedge R_k\right); \forall j, k \in J; \forall u, l \in U\ \ j \neq k\ \vee u \neq l \tag{38}$$

$$x_{mju} \geq x_{mkl} + p_{mkl} + \ s_{mjk} \cdot Q_{ju} - V * (1 - z_{mklju}) \quad \forall\ m \in \left(M \wedge R_j \wedge R_k\right); \forall j, k \in J; \forall u, l \in U\ \ j \neq k\ \vee u \neq l \tag{39}$$

$$y_{mju} \geq x_{mju} - \ s_{mjk} \cdot Q_{ju} - V * (1 - z_{mklju}) \quad \forall\ m \in \left(M \wedge R_j \wedge R_k\right); \forall j, k \in J; \forall u, l \in U\ \ j \neq k\ \vee u \neq l \tag{40}$$

$$x_{mju} \geq y_{mj,u} \quad \forall\ m\ \in\ (M\ \wedge R_j);\ \forall\ j\ \in\ J;\ \forall\ u\ \in\ U \tag{41}$$

$$x_{m,j+1,u} \geq c_{mj,u} \quad \forall\ m\ \in\ \left(M\ \wedge R_j\right);\ \forall\ j\ \in\ J, j \neq 1;\ \forall\ u\ \in\ U \tag{42}$$

$$x_{mju} \geq c_{m0u} \quad \forall\ m\ \in\ (M\ \wedge R_j);\ \forall\ j\ \in\ J, j \neq 0;\ \forall\ u\ \in\ U \tag{43}$$

$$C \geq c_{mju} \quad \forall\ m\ \in\ M;\ \forall\ j\ \in\ J;\ \forall\ u\ \in\ U \tag{44}$$

$$c_{mju} \geq x_{mju} + p_{mju} \quad \forall\ m\ \in\ M;\ \forall\ j\ \in\ J;\ \forall\ u\ \in\ U \tag{45}$$

$$\sum_{j \in J \backslash \{0\}} \sum_{u \in U} z_{mklju} = 1 \quad \forall\ m \in \left(M \wedge R_j \wedge R_k\right); \forall\ k\ \in J\{n+1\};\ \forall\ l \in U\ \ j \neq k\ \vee u \neq l \tag{46}$$

$$\sum_{k \in J \backslash \{n+1\}} \sum_{l \in U} z_{mklju} = 1 \quad \forall\ m \in \left(M \wedge R_j \wedge R_k\right); \forall\ j\ \in J\{0\};\ \forall\ u \in U\ \ j \neq k\ \vee u \neq l \tag{47}$$

$$\sum_{u}^{U} q_{ju} = d_j \quad \forall\ j\ \in\ J \tag{48}$$

$$q_{ju} \leq V \cdot Q_{ju} \quad \forall\ j\ \in\ J;\ \forall\ u\ \in\ U \tag{49}$$

$$Q_{ju} \leq q_{ju} \quad \forall\ j\ \in\ J;\ \forall\ u\ \in\ U \tag{50}$$

$$p_{mju} = p_{mj} * q_{ju} \quad \forall\ m\ \in\ M;\ \forall\ j\ \in\ J;\ \forall\ u\ \in\ U \tag{51}$$

$x_{mju} \geq 0 \qquad \forall\, m\, \in\, M;\ \forall\, j\, \in\, J;\ \forall\, u\, \in\, U \qquad (52)$

$y_{mju} \geq 0 \qquad \forall\, m\, \in\, M;\ \forall\, j\, \in\, J;\ \forall\, u\, \in\, U \qquad (53)$

$z_{mklju}\, \in\, \{0,1\} \qquad \forall\, m\, \in\, M;\ \forall\, k, j\, \in\, J, k \neq j \qquad (54)$

$Q_{ju}\, \in\, \{0,1\} \qquad \forall\, j\, \in\, J;\ \forall\, u\, \in\, U \qquad (55)$

Respecto al modelo LSJSP base, se elimina la restricción (4).

Se modifican las restricciones (6) a la que se le añade la nueva naturaleza de los tiempos de setup secuenciales y la restricción (7) que se descompone en dos restricciones (44), (45) para tener en cuenta la nueva variable incluida c_{mju}. Esto se hace principalmente para facilitar la compresión de los mecanismos del modelo, aunque no es necesario.

Se añaden las restricciones (39) a (47), marcadas a color en el modelo. Por razones de brevedad, solo se explican estas restricciones en detalle. El resto están explicadas en el apartado 3.2.2.

En la restricción (39) se consideran los tiempos de setup dependientes de la secuencia, alojados en la programación entre el makespan y el instante de incio de 2 operaciones procesadas en la misma máquina, una después de otra. Facilita la determinación del instante de inicio del procesado después de completar el setup.

Las restricciones (40) y (41) sirven para determinar el instante de inicio del tiempo de setup de cada lote teniendo en cuenta cuál es el lote predecesor en cada máquina.

Con la restricción (42) se impone que el trabajo ficticio que representa el nodo final del grafo tiene que “empezar” cuando han acabado todos los anteriores. Es decir, su instante de inicio siempre será mayor o igual que el instante de finalización de todos los demás.

Con la restricción (43) se impone que todos los lotes tienen que procesarse después de que “acabe” el trabajo ficticio inicial. Se trata de un artificio similar al anterior para imponer que en todas las máquinas este trabajo sea el primero en la secuencia.

Como se ha comentado antes, las restricciones (44) y (45) son la descomposición de la restricción (7). Esta descomposición es fundamental para incluir las 2 restricciones anteriores.

Las restricciones (46) y (47) permiten tener un control absoluto de las precedencias entre lotes, sabiendo exactamente quien es el sucesor y quien es el predecesor de cada lote.

La restricción (46) especifica que todos los lotes, en cada máquina en la que se procesan, tienen un sucesor menos el trabajo ficticio final. Conjuntamente se impone que el trabajo ficticio inicial no puede ser un sucesor.

La restricción (47) especifica que todos los lotes, en cada máquina en la que se procesan, tiene un predecesor. El trabajo ficticio final no puede ser predecesor y el trabajo ficticio inicial no tiene predecesor.

3.2.5 LSJSP con tiempos de setup dependientes de la secuencia y turnos

En este modelo se simplifican las restricciones del modelo anterior, eliminando aquellas que no son necesarias. En concreto, se omiten las restricciones (41), (42) y (43). Las restricciones (44) y (45) se vuelven a componer en una sola restricción. Adicionalmente, se añaden restricciones de turnos, de forma que los lotes solo puedan ser procesados (inicializados y terminados) en determinadas ventanas temporales.

Este modelo está compuesto por 20 restricciones. Todas ellas han sido explicadas en los modelos ya presentados y por razones de brevedad no serán descritas de nuevo.

Objetivo

$\min C$ (56)

Restricciones

$y_{R^h ju} \geq x_{R^{h-1}j,j,u} + p_{R^{h-1}j,j,u}$ $\quad \forall\, h \in (2,\ldots,M)\ in\ R_j;\ \forall\, j \in J;\ \forall\, u \in U$ (57)

$y_{mju} \geq x_{mj,u-1} + p_{mj,u-1}$ $\quad \forall\, m \in (M \wedge R_j);\ \forall\, j\backslash\{0,n+1\} \in J;\ \forall\, u \in U\backslash\{0\}$ (58)

$y_{mju} \geq (x_{mkl} + p_{mkl}) - V \cdot (1 - z_{mklju})$ $\quad \forall\, m \in (M \wedge R_j \wedge R_k); \forall j,k \in J; \forall u,l \in U;\ j \neq k \vee u \neq l$ (59)

$x_{mju} \geq x_{mkl} + p_{mkl} + s_{mjk} \cdot Q_{ju} - V * (1 - z_{mklju})$ $\quad \forall\, m \in (M \wedge R_j \wedge R_k); \forall j,k \in J; \forall u,l \in U;\ j \neq k \vee u \neq l$ (60)

$y_{mju} \geq x_{mju} - s_{mjk} \cdot Q_{ju} - V * (1 - z_{mklju})$ $\quad \forall\, m \in (M \wedge R_j \wedge R_k); \forall j,k \in J; \forall u,l \in U;\ j \neq k \vee u \neq l$ (61)

$C \geq x_{mju} + p_{mju}$ $\quad \forall\, m \in M;\ \forall\, j \in J;\ \forall\, u \in U$ (62)

$\sum_{j \in J\backslash\{0\}} \sum_{u \in U} z_{mklju} = 1$ $\quad \forall\, m \in (M \wedge R_j \wedge R_k); \forall\, k \in J\{n+1\};\ \forall\, l \in U;\ j \neq k \vee u \neq l$ (63)

$\sum_{k \in J\backslash\{n+1\}} \sum_{l \in U} z_{mklju} = 1$ $\quad \forall\, m \in (M \wedge R_j \wedge R_k); \forall\, j \in J\{0\};\ \forall\, u \in U;\ j \neq k \vee u \neq l$ (64)

$\sum_{j=0}^{J} \sum_{u=0}^{U} (x_{mj} + p_{mju} - y_{mju}) \cdot W_{moju} \leq SHT$ $\quad \forall\, m \in M;\ \forall\, o \in O$ (65)

$\sum_{o=0}^{O} W_{moju} = Q_{ju}$ $\quad \forall\, m \in (M \wedge R_j);\ \forall\, j \in J;\ \forall\, u \in U$ (66)

$y_{mju} \geq W_{moju} \cdot o \cdot SHT$ $\quad \forall\, m \in M;\ \forall\, o \in O;\ \forall\, j \in J;\ \forall\, u \in U$ (67)

$(x_{mju} + p_{mju}) \cdot W_{moju} \leq (o+1) \cdot SHT$ $\quad \forall\, m \in M;\ \forall\, o \in O;\ \forall\, j \in J;\ \forall\, u \in U$ (68)

$\sum_{u}^{U} q_{ju} = d_j$ $\quad \forall\, j \in J$ (69)

$q_{ju} \leq V \cdot Q_{ju}$ $\quad \forall\, j \in J;\ \forall\, u \in U$ (70)

$Q_{ju} \leq q_{ju}$ $\quad \forall\, j \in J;\ \forall\, u \in U$ (71)

$p_{mju} = p_{mj} \cdot q_{ju}$ $\quad \forall\, m \in M;\ \forall\, j \in J;\ \forall\, u \in U$ (72)

$$x_{mju} \geq 0 \qquad \forall\, m \in M;\ \forall\, j \in J;\ \forall\, u \in U \tag{73}$$

$$y_{mju} \geq 0 \qquad \forall\, m \in M;\ \forall\, j \in J;\ \forall\, u \in U \tag{74}$$

$$z_{mklju} \in \{0,1\} \qquad \forall\, m \in M;\ \forall\, k,j \in J, k \neq j \tag{75}$$

$$Q_{ju} \in \{0,1\} \qquad \forall\, j \in J;\ \forall\, u \in U \tag{76}$$