

3. Análisis y modelado para la agrupación de productos en células de fabricación reconfigurable.

3.1. Formulación del problema.

Tras introducir las motivaciones que nos han hecho tratar este tema y los paradigmas de fabricación que emplearemos, definiremos de una forma más concreta el problema que vamos a tratar. Para un sistema de producción dado, se desean crear una serie de células de fabricación, de forma que a cada una de ellas se le asignará la fabricación de una o varias familias de productos.

Para acometer este problema, deberemos distinguir dos fases: el diseño de las familias de productos y la planificación de la producción. La fase de diseño se centrará en estudiar las similitudes entre unos productos y otros, en función de las características que definen un SFR, y en base a ellas, cómo pueden ser agrupados en familias. Basándose en la fase de diseño, la fase de planificación se encargará de estudiar qué familias de productos son las mejores, y en qué orden deben ser producidas.

Consideremos en primer lugar la fase de diseño del sistema productivo, analizando cómo se generan unas familias de productos para un conjunto dado de productos a fabricar. Los métodos de agrupación de productos en familias son muy variados, y surgieron con el desarrollo de los SFC. Entre los más extendidos se encuentran los métodos de agrupamiento jerárquico. Las clasificaciones jerárquicas se pueden representar por estructuras de árbol invertido, también denominados dendogramas. Los dendogramas están constituidos por varios niveles, y en cada uno de ellos se muestran las diferentes formas de las que pueden agruparse los productos, dependiendo del grado de similitud de los productos que componen una familia. Los primeros niveles están compuestos por familias con pocos productos y alta similitud entre ellos, mientras las familias de los últimos niveles poseen muchos productos poco parecidos entre sí.

En la Figura 25 podemos ver un ejemplo que ilustra el concepto de dendograma. Supongamos que tenemos cinco productos a fabricar, denominados A, B, C, D y E. Como podemos apreciar en la figura, hay cinco conjuntos de familias posibles, uno por cada nivel del dendograma:

- El primer nivel está compuesto por cinco familias de productos, {A}, {B}, {C}, {D} y {E}.
- El segundo nivel está formado por cuatro familias, {A}, {B-C}, {D} y {E}, teniendo la familia {B-C} un coeficiente de similitud del 87% entre los productos que la conforman.
- El tercer nivel está formado por tres familias, {A-B-C}, {D} y {E}, teniendo la familia {A-B-C} un coeficiente de similitud del 69% entre los productos que la conforman.
- El cuarto nivel está formado por dos familias, {A-B-C} y {D-E}, teniendo la familia {D-E} un coeficiente de similitud de un 40% entre los productos que la conforman.
- El quinto nivel está formado por una única familia, {A-B-D-C-E}, con un coeficiente de similitud de un 10% entre los productos que la conforman.

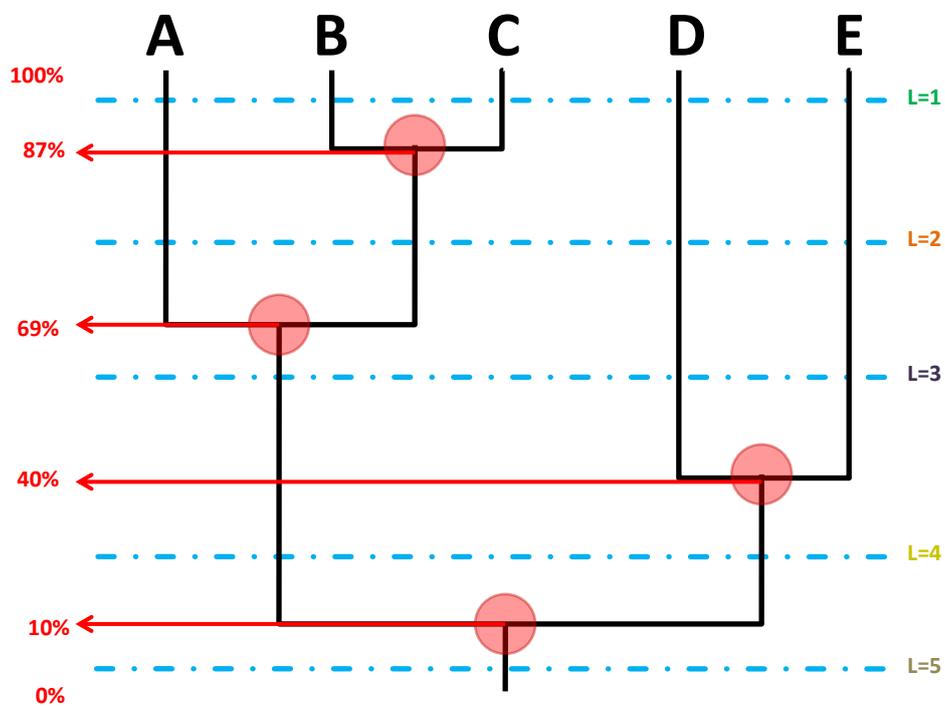


Figura 25: Ejemplo de dendograma.

Empecemos suponiendo que vamos a trabajar con un SFR clásico, en el que no organizamos las máquinas en células. La fabricación de las familias de productos será secuencial: se selecciona la primera familia a producir, y se configura el SFR para fabricar todos los productos de la familia seleccionada en la cantidad demandada. Una vez terminada la fabricación de la familia, el sistema es reconfigurado para producir la siguiente familia (Xiaobo *et al.*, 2000), y así sucesivamente hasta que el sistema fabrica todos los productos, volviendo el sistema a su configuración inicial. Esto queda ejemplificado en la Figura 26. Como mencionamos en el capítulo anterior, dos de las razones que llevan a

repetir cíclicamente la producción de todas las familias son, por un lado, la llegada de nuevas órdenes de productos de familias ya fabricadas y, por otro lado, las limitaciones en la capacidad de almacenamiento por parte de las empresas. Sólo cuando las empresas añaden o quitan productos de sus sistemas de producción, o sus demandas son muy variables, la metodología propuesta puede seleccionar familias de productos distintas o secuencias de producción distintas.

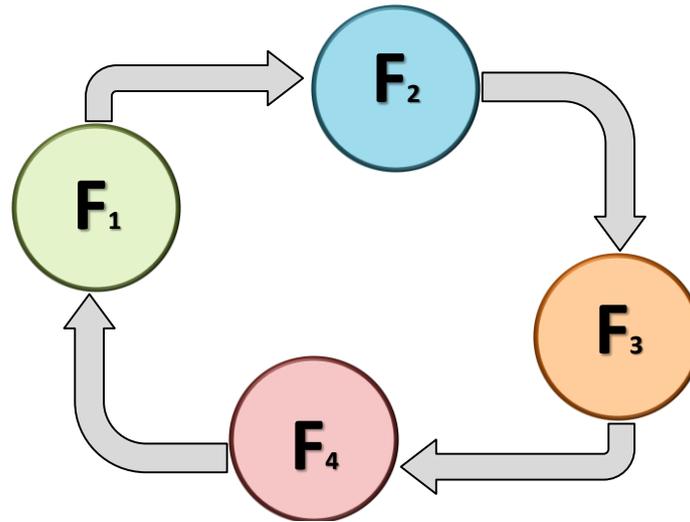


Figura 26: Secuenciación de familias de productos.

La selección de las familias se basará en el dendograma: hay que seleccionar un único nivel, y ordenar las familias de dicho nivel. Si analizamos la formación de familias de productos en cada nivel del dendograma, podemos distinguir distintos casos (Eguía *et al.*, 2006):

- En el caso de que todos los productos estuviesen agrupados en una única familia (el nivel inferior del dendograma), el sistema productivo estaría compuesto por todas las máquinas necesarias para fabricar los productos. En ese caso, no se incurrirían en costes de reconfiguración del sistema, pero existirían máquinas ociosas y/u otras cuyas funcionalidades que no son completamente utilizadas cuando se fabriquen los productos. Además, la capacidad de las máquinas suele ser mayor que la requerida para la fabricación de cada producto de forma independiente.
- En el caso de seleccionar una familia para cada producto (el nivel superior del dendograma), habría que hacer frente a costes de reconfiguración del sistema al pasar de fabricar una familia a otra, pero en cambio el número de máquinas ociosas desaparece y sus funcionalidades y capacidades quedan completamente aprovechadas.

- El último caso ocurre cuando se seleccionan menos familias que productos (niveles intermedios del dendograma). En esta ocasión, aunque existen costes de reconfiguración, éstos serán menores que en el segundo caso al existir menos cambios. También el número de máquinas ociosas es inferior a las del primer caso.

Los parámetros clave a tomar en consideración para la selección de familias de productos son: el coste de reconfiguración del sistema productivo, el coste de tener máquinas ociosas, el coste de no-uso de las funcionalidades de las máquinas y el coste de no-uso de la capacidad de las máquinas. Estos parámetros se simplifican en dos:

- Costes de reconfiguración, $R_{F_i-F_j}$, asociados a los cambios en las máquinas y los módulos que las conforman, necesarios para cambiar de familia a fabricar, de la familia F_i a la F_j .
- Costes de no-uso, H_{F_i} , asociados a no utilizar todos los recursos durante la fabricación de una familia F_i en concreto.

Tras analizar estos casos, podemos concluir que hay que llegar a un equilibrio entre los costes de reconfiguración en los niveles superiores del dendograma y los costes de no-uso de los niveles inferiores. En el caso de que dispusiésemos de varias células, también consideraremos que dentro de cada célula existirán unos costes de reconfiguración y de no-uso de recursos.

En la Figura 27 mostramos un ejemplo de un sistema de fabricación con nueve productos a fabricar, para mostrar de una forma gráfica la fase de diseño y de planificación, y cómo se plantearía ésta en el caso de tener una o dos células. Gracias a esta figura, podemos ver cómo, gracias al uso de células, se reduce significativamente el tiempo empleado en producir los productos demandados, dado que permite la fabricación en paralelo.

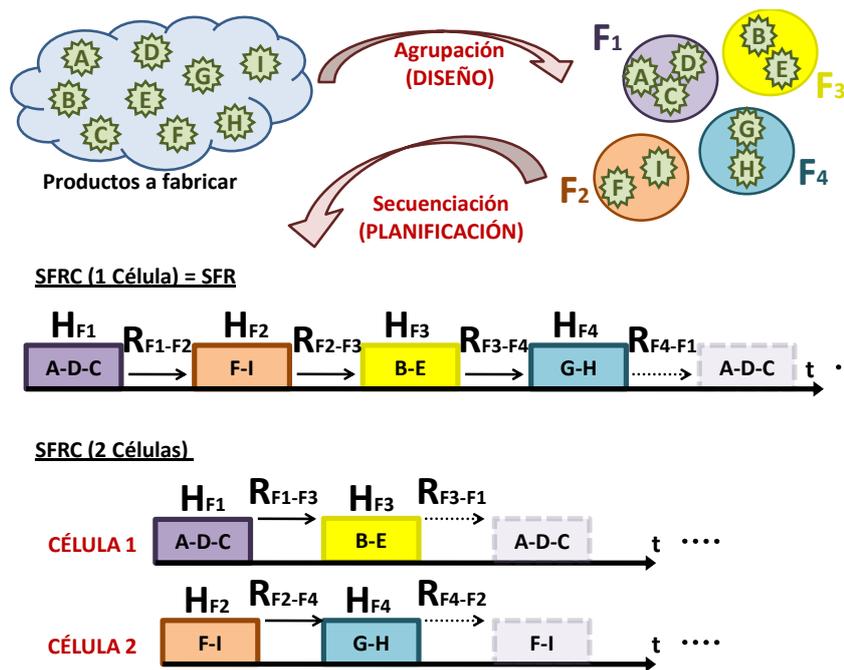


Figura 27: Comparación entre SFR y SFRC.

La fase de diseño ha sido desarrollada en investigaciones anteriores (Galan *et al.*, 2007a), y no es objeto del desarrollo de este proyecto. Partiendo de la fase de diseño, se desea formar un número de células conocido, cada una de ellas con una o más familias asignadas. Para cada célula, se obtendrá una secuencia con el orden de fabricación de las familias, de forma que se incurrirán en costes de reconfiguración por cambio de familias y en costes de no-uso de recursos dentro de cada familia de las células. Por lo tanto, nuestro estudio se centrará en **la planificación de un sistema de fabricación reconfigurable organizado de forma celular**.

Al crear exactamente el número de células que se indique como dato de partida, no son admisibles niveles inferiores del dendograma con menos familias que dicho número, puesto que daría lugar a células vacías. Si el número de células a formar es uno, el problema coincidirá con el ya estudiado de los Sistemas de Fabricación Reconfigurable, es decir, la solución final seleccionará un nivel del dendograma y ordenará las familias del mismo a mínimo coste.

Además, como veremos más adelante, y queda ilustrado en la Figura 28, en esta investigación se modificará el modelo anterior en el siguiente aspecto: se establecerá un estado inicial al que hay que volver tras la fabricación de todas las familias asignadas a cada célula, en lugar de volver a la familia inicial.

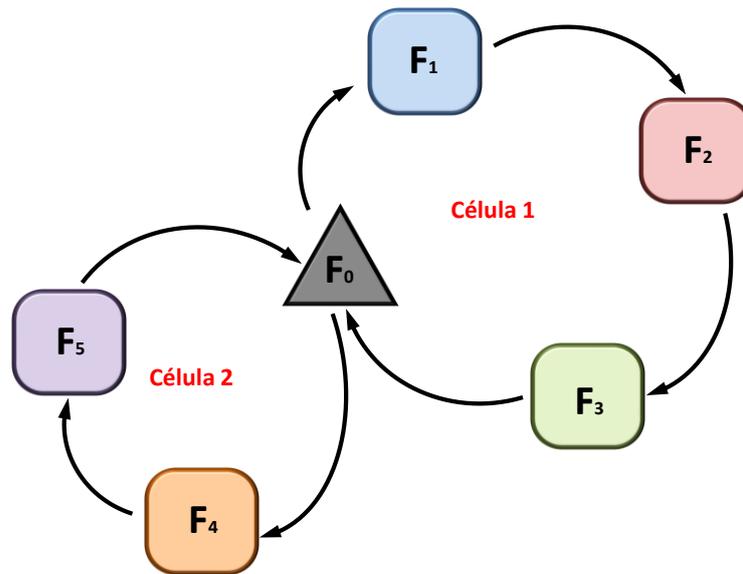


Figura 28: Secuenciación de familias en SFRC.

3.2. Analogías con otros problemas combinatorios.

3.2.1. Analogía de nuestro problema, para una única célula, con otros problemas combinatorios.

En el caso de que utilizásemos una única célula, podríamos modelar nuestro problema como el problema del viajante de comercio (TSP, “Traveling Salesman Problem”). Este problema ha sido ampliamente tratado en la literatura, y su objetivo es encontrar una ruta Hamiltoniana que minimice la distancia total recorrida por un comerciante que tiene que visitar un cierto número de ciudades una sola vez, partiendo de una de ellas, denominada ciudad base o depósito, y acabando su viaje en la misma. El objetivo de minimizar la distancia es equivalente a minimizar el tiempo empleado o, de una forma más genérica, el coste invertido en los viajes c_{ij} (ver Figura 29).

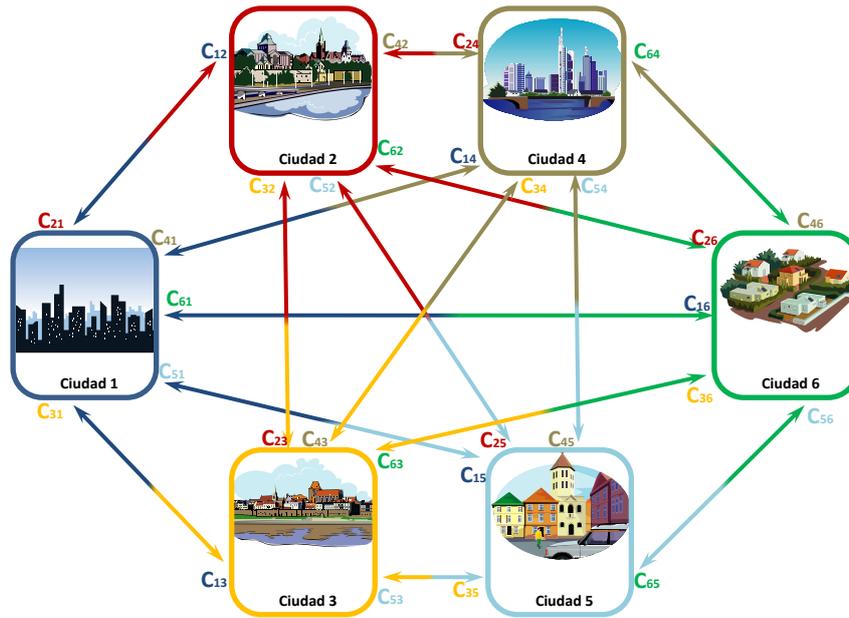


Figura 29: Problema del TSP

La analogía entre ambos problemas es directa. Las familias de productos a secuenciar con el menor coste posible en el SFR, se traducirían en el TSP como las ciudades que debe recorrer el comerciante empleando la menor distancia posible (o equivalentemente, en el menor tiempo posible o con el menor coste). Por último, en el TSP el comerciante tiene que acabar en la misma ciudad en la que inició el trayecto, la ciudad base, mientras que los SFR están configurados para que una vez terminada la fabricación de la última familia se reconfigure el sistema para fabricar de nuevo la primera familia.

Existen múltiples variantes a la hora de plantear un TSP. El problema al que nosotros nos enfrentamos puede ser expresado como un TSP asimétrico, ya que los costes de reconfiguración entre dos familias difieren según el sentido seleccionado, es decir, dependen de la configuración inicial y final.

Las familias de productos a las que aplicar el problema del TSP variarán según el nivel del dendograma en el que nos encontremos. Por ello, el problema de un SFR será el de un TSP-multinivel, en el que iremos resolviendo el problema TSP en cada uno de sus niveles, y elegiremos el nivel que posea el mínimo coste. A la hora de resolver el TSP en cada nivel, sólo es necesario considerar los costes de reconfiguración, ya que los costes de no-uso son iguales independientemente de la secuencia de fabricación obtenida, puesto que dependen únicamente de las familias que se forman en ese nivel. Cuando

encontremos la secuencia con el menor coste de reconfiguración, procederemos a sumarle los costes de no-uso, obteniendo los costes totales, que nos permitirán elegir entre los distintos niveles. Podemos ver un ejemplo que ilustra esto de una forma simple para un SFR de tres productos en la Figura 30.

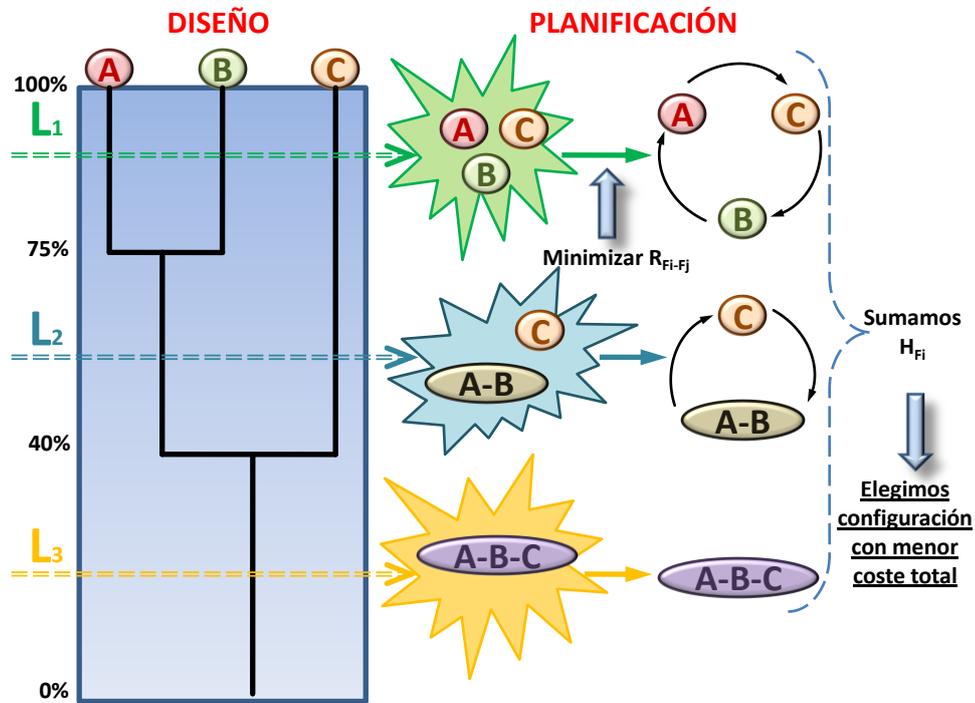


Figura 30: Diseño y planificación de familias de productos.

Es importante destacar que el TSP es un problema de optimización combinatoria de clase NP (Non-Determinist Polynomial Time) (Rinnooy Kan, 1976). Ese tipo de problemas se caracterizan porque los tiempos de resolución no aumentan de forma lineal con el número de productos, sino de forma exponencial. Ello conllevará que la selección y secuencia de las familias de productos en un SFR también sea de tipo NP.

Para que el problema sea más fácil de resolver, haremos tres consideraciones. En primer lugar, supondremos que las máquinas tienen una capacidad infinita, con lo que serán capaces de soportar toda la demanda. En segundo lugar, supondremos que no existirán rutas alternativas para fabricar un producto, es decir, existirá un único plan de proceso para cada componente del producto, y cada operación o conjunto de operaciones será realizado en una sola máquina o módulo de máquina. Por último, cada máquina tendrá sus propios módulos, que no serán compartidos con otra máquina.

3.2.2. Analogía de nuestro problema, para varias células, con otros problemas combinatorios.

Pero, ¿qué ocurriría si agrupásemos las familias de productos en una serie de células? Como ya hemos analizado en profundidad, las células no serían más que distintas zonas dispuestas en el sistema de fabricación en las que fabricaríamos una serie de familias. La inserción de estas células presenta un gran interés, dado que permitiría la fabricación en paralelo, disminuyendo el tiempo que emplea el sistema en fabricar los productos, con lo que también podría aumentar su capacidad de producción.

Dado que este trabajo se enmarca dentro de un primer acercamiento al problema, consideraremos una situación ideal, en la que no existe ninguna limitación de recursos, con lo que dispondremos de máquinas repetidas en las distintas células. Gracias a ello, los productos fabricados en una célula no requerirán operaciones de máquinas dispuestas en otra célula, con lo que no se producirán movimientos intercelulares de productos. Esta independencia entre células, que poseerán todos los recursos necesarios para elaborar un producto dado, si bien facilitará el análisis, será poco realista, ya que incurriremos en mayores costes al haber redundancia de máquinas. Es decir, sólo consideraremos los costes que describimos anteriormente: los costes de reconfiguración del sistema y los costes de no-uso. Además, también asumiremos las simplificaciones que detallamos en el apartado anterior para el caso de única célula.

Para modelar la inclusión de células en el SFR, partiremos del problema del TSP y daremos un paso más, incluyendo a varios comerciantes que se reparten el trayecto a recorrer. Esta generalización del TSP se denomina m -TSP, el problema de los múltiples viajeros de comercio. Al tener un número m de comerciantes que partirán y terminarán en la misma ciudad base (o depósito), repartiremos entre ellos las ciudades a visitar (denominadas ciudades intermedias), de forma que todas sean visitadas al menos una vez, minimizando la función objetivo total. Esta función objetivo, como mencionamos al explicar el TSP, puede traducirse de forma equivalente en distancia recorrida, tiempo empleado, o coste invertido. En la Figura 31 podemos ver un ejemplo de solución a este problema en el caso de tener dos comerciantes ($m=2$), que se reparten el trayecto a realizar. El trayecto realizado por cada comerciante será identificado por el subíndice k .

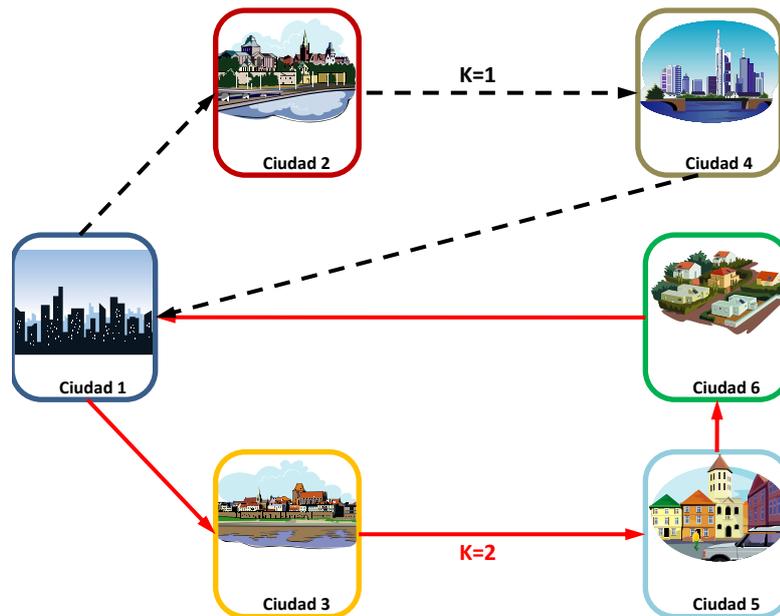


Figura 31: Problema del m-TSP ($m=2$).

A esta definición del problema se le pueden proponer múltiples variantes (Bektas, 2006), variando los siguientes parámetros:

- Número de ciudades base o depósitos. Puede plantearse el problema de forma de que, en vez de existir un único depósito del que parten y en el que terminan todos los comerciantes, existan varios. En ese caso pueden plantearse dos variantes, de destino fijo y de destino no fijo.
 - o El primer caso consiste en que cada comerciante vuelva, tras finalizar su ruta, al mismo depósito del que partió.
 - o En el segundo caso, el comerciante puede volver indistintamente al depósito del que partió o a otro distinto, con la única restricción de que deben partir el mismo número de comerciantes de cada depósito que los que acaban en él tras finalizar su ruta.
- Número de comerciantes. Puede ser un parámetro fijado a priori, o ser una variable más del problema.
- Costes fijos. Cuando el número de comerciantes no está fijado a priori, podemos penalizar en la función objetivo cada vez que un comerciante es usado en la solución, asociándole un coste fijo. Ello se debe a que, en caso contrario, se tendería a la solución obvia de emplear un número de comerciantes igual al número de ciudades a visitar.
- Ventanas de tiempo. Podemos plantear el problema con las restricciones de que ciertas ciudades deban ser visitadas en determinados períodos de

tiempo, denominados ventanas de tiempo (“Time Windows”). Esta es una de las variantes más importantes del m-TSP (m-TSPTW), debido a la gran cantidad de aplicaciones prácticas que posee.

- Otras restricciones. Existen gran variedad de restricciones a este problema, entre las que destacan limitar el número de ciudades que visita cada comerciante, la máxima o la mínima distancia que recorren.

Si bien el TSP ha sido profundamente estudiado, el problema del m-TSP no ha recibido tanta atención. Sin embargo, al involucrar a más de un comerciante, es más versátil a la hora de aplicarlo a problemas de la vida real, ya que permite tener múltiples procesos en paralelo. Estas aplicaciones abarcan múltiples campos, desde la planificación del funcionamiento de una imprenta (Carter y Ragsdale, 2002), la planificación de equipos, el diseño de sistemas de sondeo de navegación global por satélite (Saleh y Chelouah, 2004), hasta la planificación de la producción, que es la aplicación que nos interesa resolver. Además, es importante destacar que el m-TSP puede relacionarse con los ampliamente estudiados problemas de guiado de vehículos de vehículos (VRP, Vehicule Routing Problem), con la inclusión de restricciones adicionales.

Atendiendo a lo anteriormente explicado, la analogía entre nuestro problema y el del m-TSP se presenta bastante clara. Al igual que vimos con el TSP y el SFR, las familias a fabricar con el menor coste posible serían representadas por las ciudades a visitar minimizando una función objetivo, relacionada con distancias, tiempos o, de una forma más genérica, costes incurridos. Las células de nuestro sistema productivo, que se dedicarían a fabricar una o más familias, se trasladarían al m-TSP como los comerciantes que irían recorriendo las distintas ciudades. Por último, el orden en que serían fabricadas las familias contenidas en cada célula, no sería más que la ruta que emplearía cada comerciante para recorrer las ciudades que tiene asignadas.

Podemos ver de una forma un poco más gráfica esta analogía en la Tabla 2:

Problema m-TSP	Problema SFRC
<i>n</i> ciudades	<i>n</i> familias de productos
<i>m</i> comerciantes	<i>m</i> células de trabajo
ruta del <i>i</i> -ésimo comerciante	orden de fabricación de la <i>i</i> -ésima célula.

Tabla 2: Analogía entre el m-TSP y el problema de los SFRC.

3.3. Técnicas para la resolución del m-TSP.

Como ya hemos explicado en el apartado anterior, el problema del m-TSP involucra a m comerciantes que deben visitar un número n de ciudades. Para representar este problema se recurre a un grafo $G(V, A)$, donde V es el conjunto de n nodos (las ciudades a visitar) y A es el conjunto de arcos que los unen. Cada uno de estos arcos tendrá asociado un coste c_{ij} , con lo que queda definida una matriz C de costes. Además, consideraremos que todos los comerciantes comienzan y terminan su recorrido en la misma ciudad base. Podemos ver un ejemplo de este problema en la Figura 32, en la que la ciudad base se encuentra desglosada en dos nodos distintos, para que resulte un poco más visual, si bien hay que incidir en que la ciudad origen/destino se trata de la misma ciudad.

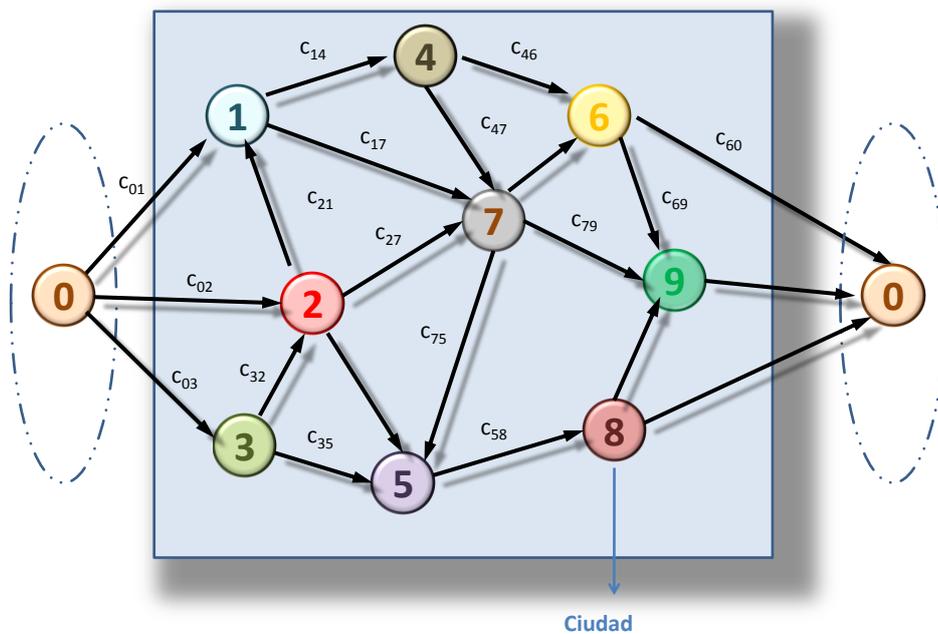


Figura 32: Grafo con costes para de un m-TSP.

Para dar solución a este problema, podemos plantearlo desde varios puntos de vista. A continuación, mostraremos dos filosofías distintas para resolver el problema del m-TSP, incidiendo en las ventajas e inconvenientes que presenta cada una. Esto será de vital importancia a la hora de justificar la elección de una de ellas para resolver el problema. Una posible solución al problema planteado en la Figura 32 es mostrada en la Figura 33, para tres comerciantes ($m=3$).

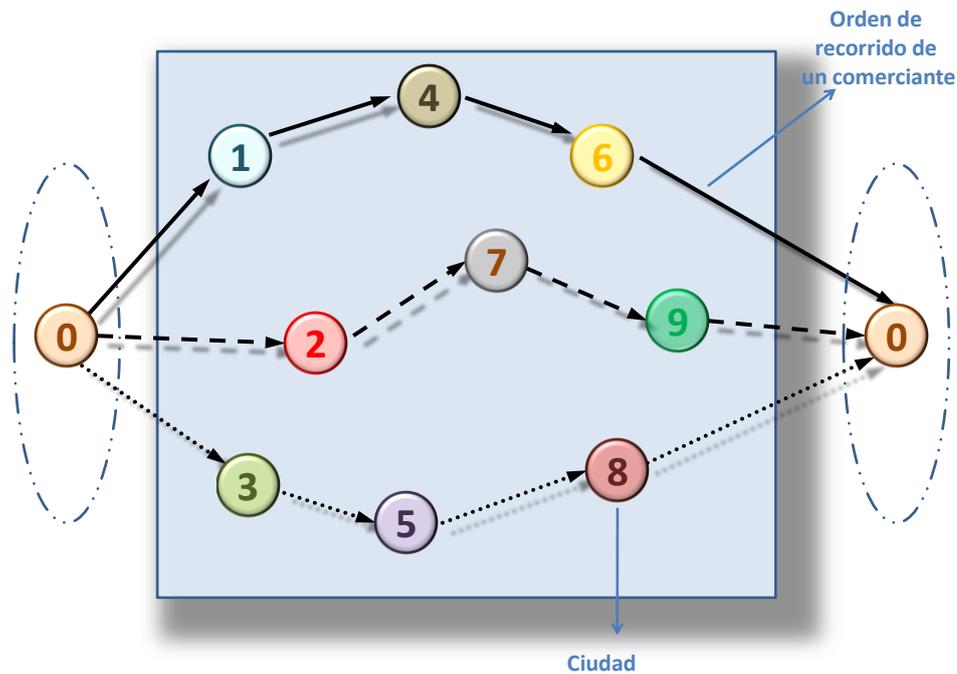


Figura 33: Ejemplo de resolución de un problema m-TSP para $m=3$.

3.3.1. Modelos de programación lineal para la resolución del m-TSP.

Hay diversos modelos de programación lineal que pueden usarse para resolver el problema del m-TSP de una forma exacta. Dentro de los varios planteamientos existentes podemos destacar dos de ellos, la formulación basada en modelos de asignación y la formulación basada en modelos de flujo en redes.

3.3.1.1. Formulación basada en modelos de asignación.

Para este modelo utilizaremos las variables binarias x_{ij} , que valen 1 en el caso de que el arco que une la ciudad i y la ciudad j sea transitado por algún comerciante, y 0 en caso contrario.

El modelo resultante sería:

$$\text{Min} \quad \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}$$

s.a

$$(1): \quad \sum_{j=2}^n x_{1j} = m$$

$$(2): \quad \sum_{j=2}^n x_{j1} = m$$

$$(3): \quad \sum_{i=1}^n x_{ij} = 1 \quad : j = 2, \dots, n$$

$$(4): \quad \sum_{j=1}^n x_{ij} = 1 \quad : i = 2, \dots, n$$

(5): + restricciones de eliminación de subrotas

$$(6): \quad x_{ij} = [0,1] \quad : (i, j) \in A$$

La función objetivo de este modelo representa la necesidad de minimizar el coste empleado por los m comerciantes al recorrer el conjunto de las n ciudades. Pero, como todo modelo de programación lineal, necesitará una serie de restricciones para limitar y definir el problema el problema:

- Las restricciones (1) y (2) asegurarán que comiencen y finalicen su ruta en la ciudad base los m comerciantes.
- Las restricciones (3), (4) y (6) son las restricciones básicas que caracterizan a todo modelo de asignación, asegurando que cada ciudad que no sea la ciudad base, sea visitada por un único comerciante.
- Mediante la restricción (5) debemos imponer que no se produzcan subrotas, es decir, que se formen rutas entre ciudades intermedias sin conexión con la ciudad base. En la literatura podemos encontrar distintas restricciones que solucionen este problema. Nosotros optaremos por aplicar las restricciones de (Miller *et al.*, 1960), que si bien aportan nuevas variables al problema, permiten obtener la solución en tiempo real. Éstas son las siguientes, siendo p el máximo número de ciudades que pueden ser visitadas por cada comerciante:

$$(5.1): \quad px_{ij} + u_i - u_j \leq p-1 \quad : (i, j) \in A; \quad j \neq i; \quad i = 2, \dots, n$$

$$(5.2): \quad u_i \geq 0 \quad : i = 2, \dots, n$$

3.3.1.2. Formulación basada en modelos de flujo en redes.

Se definen las variables binarias x_{ijk} , que valen 1 si el viajante k -ésimo usa el arco (i,j) que une las ciudades i y j en su ruta, y 0 si no.

El modelo que resultante sería:

$$\text{Min} \quad \sum_{i=1}^n \sum_{j=1}^n c_{ij} \sum_{k=1}^m x_{ijk}$$

s.a

$$(1): \quad \sum_{i=1}^n \sum_{k=1}^m x_{ijk} = 1 \quad : j = 1, \dots, n$$

$$(2): \quad \sum_{i=1}^n x_{ipk} - \sum_{j=1}^n x_{pjk} = 0 \quad : k = 1, \dots, m; \quad p = 1, \dots, n$$

$$(3): \quad \sum_{j=1}^n x_{1jk} = 1 \quad : k = 1, \dots, m$$

$$(4): \quad n \cdot \sum_{k=1}^m x_{ijk} + u_i - u_j \leq n - 1 \quad : (i, j) \in A; \quad j \neq i; \quad i = 2, \dots, n$$

$$(5): \quad u_i \geq 0 \quad : i = 2, \dots, n$$

$$(6): \quad x_{ijk} = [0, 1] \quad : (i, j) \in A; \quad k = 1, \dots, m$$

Al igual que en la formulación basada en modelos de asignación, la función objetivo de este modelo representa la necesidad de minimizar el coste empleado por los m comerciantes al recorrer el conjunto de las n ciudades. Las restricciones necesarias para limitar el problema son las siguientes:

- La restricción (1) asegura que cada ciudad sea visitada por un solo comerciante.
- La restricción (2) asegurará que se cumpla un balance de flujo en cada ciudad, es decir, que entrarán y saldrán de cada ciudad el mismo número de comerciantes.
- La restricción (3) asegura que todos los viajeros inicien su recorrido en la ciudad base. Viéndolo en el grafo, como cada viajante k inicia su recorrido en dicho nodo, saldrán m arcos de él.
- Las restricciones (4) y (5) no son más que las condiciones de Miller, necesarias para la eliminación de subrutas.

- La restricción (6) limita que las variables x_{ijk} sean binarias.

3.3.2. Técnicas heurísticas para la resolución del m-TSP.

El carácter combinatorio del problema al que nos enfrentamos conlleva que sea complejo de resolver, dado el gran número de soluciones posibles que existen. Ello hace que, a medida que aumente la complejidad del problema (es decir, el número de variables implicadas), aumente de forma exponencial su tiempo de resolución.

Por ello, llegará un punto en el que deje de resultar rentable resolver estos problemas mediante modelos matemáticos, ya que, si bien están diseñados para proporcionar soluciones exactas, en el caso de problemas complejos los tiempos de resolución serán tan grandes, que puede plantearse la situación de que no sean capaces de resolver el problema.

Por ello, propondremos una estrategia alternativa a la programación lineal. Analizaremos métodos de resolución aproximados, también denominados técnicas heurísticas. Estas técnicas están diseñadas para encontrar soluciones satisfactorias en un tiempo de cálculo razonable (Adenso *et al.*, 1996). Si bien pueden resultar no ser las mejores soluciones, aceptaremos una cierta desviación respecto a la solución ideal. La calidad de la solución encontrada será tal que con ellas alcancemos un equilibrio entre el valor de su función objetivo y el tiempo de computación empleado para llegar a ella.

Podemos considerar que existen dos tipos de heurísticas:

- Las heurísticas específicas se plantean como algoritmos simples que requieren tiempos de computación relativamente cortos, y que están diseñados *ad-hoc* (a medida) para cada tipo de problema.
- En cambio, las metaheurísticas son técnicas más genéricas, con lo que, con pequeñas modificaciones, son aplicables a un mayor número de problemas combinatorios.

A continuación enumeraremos distintas heurísticas empleadas en la literatura para resolver el problema que nos interesa, el m-TSP (Bektas, 2006):

- Heurísticas simples. Podemos englobar bajo este término a heurísticas que se basan en transformar el problema del m-TSP al de un TSP simple en

un grafo expandido. Entre ellos, puede destacarse un algoritmo desarrollado por (Russell, 1977), versión extendida de la heurística clásica de (Lin y Kernighan, 1973) de resolución del TSP; y un algoritmo desarrollado por (Potvin *et al.*, 1989).

- Algoritmos evolutivos. La aplicación de estos algoritmos a nuestro problema es planteada por (Fogel, 1990), suponiendo que existen dos comerciantes, minimizando la diferencia de las longitudes de las rutas de cada comerciante. Para problemas de 25 y 50 ciudades, este algoritmo ha proporcionado muy buenos resultados.
- Redes neuronales. Varias aproximaciones al problema han sido planteadas desde este punto de vista, pero generalmente son versiones extendidas de las propuestas para el TSP. (Wacholder *et al.*, 1989) desarrollan un algoritmo demasiado complejo, además de no garantizar soluciones realizables. (Hsu *et al.*, 1991) sostienen que mejoran los resultados de este último, mediante una transformación del m-TSP en la resolución de m problemas TSP estándar. Además, existen varias aproximaciones mediante redes neuronales auto-organizadas, como la de (Vakhutinsky y Golden, 1994) (basado en una aproximación de red elástica para el TSP), (Modares *et al.*, 1999) y (Somhon *et al.*, 1999), que minimizan el coste de la ruta más cara de todos los comerciantes, y parece superar los resultados de la aproximación anterior.
- Algoritmos genéticos. El uso de estos algoritmos para la resolución del m-TSP fue propuesto por primera vez por (Zhang *et al.*, 1999). Otra aplicación fue la de (Tang *et al.*, 2000), convirtiendo el m-TSP en un TSP simple y aplicando un algoritmo genético modificado, para problemas de planificación de laminación en caliente. (Yu *et al.*, 2002) también usaron algoritmos genéticos para planificación de caminos. Es importante destacar el trabajo desarrollado por (Carter y Ragsdale, 2006), en el que, como mencionaremos en el capítulo siguiente, nos hemos basado a la hora de implementar nuestro algoritmo metaheurístico.
- Recocido simulado. (Song *et al.*, 2003) proponen una aproximación extendida del algoritmo del recocido simulado, imponiendo unos costes fijos asociados a cada comerciante. Ésta ha sido probada con problemas del orden de 400 ciudades y 3 comerciantes.
- Búsqueda Tabú. Esta heurística ha sido empleada para resolver problemas m-TSP con ventanas de tiempo por (Ryan *et al.*, 1998). Estos autores ofrecen una formulación de programación lineal entera, pero resuelven el problema mediante un algoritmo de Búsqueda Tabú en un entorno de simulación de eventos discretos.

Tras considerar estas técnicas de resolución aproximadas, nos hemos decidido por la metaheurística de la Búsqueda Tabú. Ello se debe a su buen comportamiento en una amplia gama de problemas combinatorios, y en particular en el problema del TSP. Además, es un algoritmo robusto respecto a la configuración de los parámetros (Carlton y Barnes, 1996), con buenos resultados en un amplio espectro de restricciones, funciones objetivo y ventanas de tiempo.

Además, validaremos los resultados de nuestro enfoque metaheurístico de Búsqueda Tabú con un modelo del m-TSP aplicado al problema de los SFRC, mediante una formulación basada en flujo en flujo en redes.