Capítulo 2. Revisión de la literatura

2.1 Introducción

En este capítulo se pretende explicar en qué contexto de investigación y desarrollo de los sistemas de asignación de recursos se encuentra enmarcado el proyecto fin de carrera.

A continuación se realiza una descripción de los métodos óptimos utilizados. Se describirán como es cada modelo, haciendo una clasificación según las características que tiene cada uno.

A lo largo de este capítulo también se analizará la literatura más relevante relacionada con las metaheurísticas, analizando cada modelo aproximado y relacionándolo con los métodos óptimos descritos anteriormente.

2.2 Métodos óptimos de resolución

En este apartado se pretende realizar una clasificación de los modelos óptimos que se irán utilizando a lo largo del proyecto. Se mostraran pequeños ejemplos con los que se facilitará la comprensión de los mismos.

2.2.1 Cubrimiento de conjuntos

El problema consiste en que dado un conjunto de conductores factibles, escoger un subconjunto de conductores que permitan que todas los servicios sean cubiertos y tengan un coste mínimo. Es decir encontrar una partición del conjunto de conductores factibles, que tenga el costo mínimo y cubra todas las tareas, o la partición de costo mínimo.

2.2.1.1 Modelo básico

Se realiza una asignación de conductores a trabajos que hay que realizar, de tal forma que todos los trabajos se realicen exactamente una vez, cada uno de los trabajos sea factible, y el conjunto total de trabajos sea mínimo.

$$\min \quad \sum_{k \in K} f_k x_k$$

$$\text{S.A.} \quad \sum_{k \in K(i)} x_k = 1 \qquad \forall i \in I$$

$$x_k = \{0,1\} \qquad \forall k \in K$$

Ecuación 1: Modelo básico de cubrimiento de conjuntos

Sea K el conjunto de todas los conductores factibles e I el conjunto de todas las tareas a realizar, se define $K(i) \subseteq K$ como el conjunto de todos los conductores que cubren una tarea $i \in I$, si fk denota el costo de elegir una trabajo k.

La función objetivo min $\sum_{k \in K} f_k x_k$ busca que el costo total de las jornadas sea mínimo, x_k corresponde a la decisión binaria de elegir o no el trabajo k. La restricción de igualdad obliga a que cada tarea se realice exactamente una vez.

Ejemplo: Se plantea un modelo de ejemplo que se usará en todos los apartados en los que sea posible.

Se van a tener dos líneas de autobuses, la línea 1 la recorren tres autobuses y la línea 2 dos autobuses. En la plantilla se tiene 8 conductores para la línea 1 (x_1,…, x_8), de los cuales x_1, x_2, x_3 son para el autobús 1, x_4, x_5, x_6 para el autobús 2, y x_7, x_8 para el autobús 3, y tenemos 5 conductores para la línea 2 (x_9,…, x_{13}), en los que x_9, x_{10}, x_{11} son para el autobús 1, y x_{12}, x_{13} son para el autobús 2. El coste de escoger cada trabajador será: {f_1=2, f_2=4, f_3=5, f_4=4, f_5=3, f_6=7, f_7=2, f_8=4, f_9=8, f_{10}=6, f_{11}=5, f_{12}=3, f_{13}=6}.

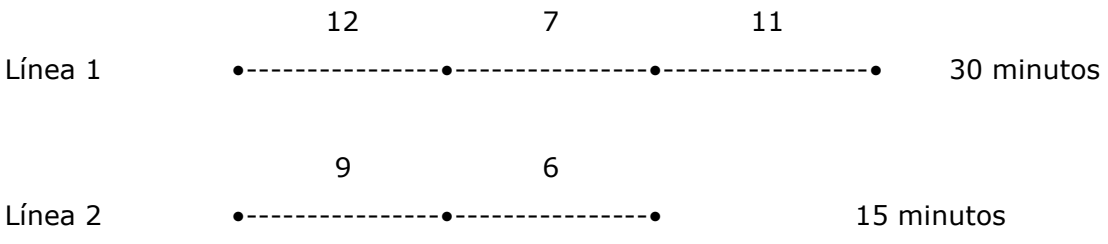

Figura 2.1: Ejemplo para modelo básico

El punto final de cada línea coincide con el punto inicial.

Para este primer modelo básico los trabajadores escogidos serían x_1, x_5, x_7, x_{11}, x_{12} y los demás valen cero.

F.O. $f_1x_1+ f_5x_5+ f_7x_7+ f_{11}x_{11}+ f_{12}x_{12} = 2+3+2+5+3 = 15$

S.A. $x_1+x_2+x_3 = 1$
$X_4+x_5+x_6 = 1$
$x_7+x_8 = 1$
$x_9+x_{10}+x_{11} = 1$
$x_{12}+x_{13} = 1$

2.2.1.2 Cobertura por más de un recurso

El problema también puede plantearse de una manera un poco diferente, relajando las condiciones y suponiendo que la tarea debe ser cubierta mínimo una vez. En algunos casos este tipo de formulación simplifica el problema.

$$\min \quad \sum_{k \in K} f_k x_k$$

$$\text{S.A.} \quad \sum_{k \in K(i)} x_k \geq 1 \qquad \forall i \in I$$

$$x_k = \{0,1\} \qquad \forall k \in K$$

Ecuación 2: Modelo de cobertura por más de un recurso

El significado de las variables es el mismo que en el modelo anterior, solo teniendo en cuenta la diferencia aclarada anteriormente.

Ejemplo: para este modelo usemos el mismo ejemplo que en el modelo anterior, la diferencia es la posible existencia en algunos autobuses de conductores suplentes.
Los trabajadores escogidos son x_1, x_2, x_5, x_7, x_8, x_{10}, $x_{11,}$ x_{12}

F.O. $f_1x_1+ f_2x_2+ f_5x_5+ f_7x_7+ f_8x_8+ f_{10}x_{10}+ f_{11}x_{11}+ f_{12}x_{12} =$
$2+4+3+2+4+6+5+3 = 29$

S.A. $x_1+x_2+ x_3 = 2$
$X_4+x_5+x_6 = 1$
$x_7+ x_8 = 2$
$x_9+x_{10}+ x_{11} = 2$
$x_{12}+x_{13} = 1$

El algoritmo para resolver estos modelos es el siguiente:

- Paso 0: Inicialización

  Generar un conjunto de piezas de manera que cada una de las tareas pueda ser cubierta al menos por una pieza. El conjunto inicial de las columnas consta de estas piezas.

- Paso 1: Cálculo del límite inferior

  Resolver un problema dual Lagrangiano con el actual conjunto de las columnas.

- Paso 2: Generación de columnas

  Generar columnas con coste reducido negativo. Si no existen tales columnas (u otro criterio de terminación está satisfecho), vaya al paso 3, de lo contrario, vuelva al paso 1.

- Paso 3: Construcción de soluciones factibles

  Utilice todas las columnas generadas en el paso 0 y el paso 2 para construir una solución factible.

2.2.1.3 Separación en la cobertura de recursos

Se trata de un modelo intermedio entre los dos anteriores, la diferencia es que la asignación de tareas es dividida en dos grupos, y cada uno equivale a un modelo de los anteriores, por lo tanto, un conjunto de tareas debe ser cubierto exactamente una vez, y para el resto se relajan las condiciones y deben ser cubiertas mínimo una vez.

Minimizar $\sum_{j=1}^{N} D_j x_j$

Sujeto a $\sum_{j=1}^{N} A_{ij} x_j = 1$ for i = 1,…, L

$\sum_{j=1}^{N} A_{ij} x_j \geq 1$ for i = L+1,..., N

x_j = 0 o 1, para j = 1,…, N

Ecuación 3: Modelo de separación en la cobertura de recursos

X_j es igual a 1 si el turno j es usado en la solución y vale 0 en cualquier otro caso.
$D_j = W + C_j$ para j=1,…, N
C_j combina los salarios y gastos de desplazamiento durante el turno j.
X es el número de turnos en la solución inicial.

El peso W debe ser suficientemente grande para asegurar que la reducción de los turnos tenga prioridad sobre la reducción de otros gastos a fin de garantizar el número mínimo de cambios en la solución.

L denota el número de piezas de trabajo las cuales pueden ser identificadas por restricciones de igualdad.

La restricción asegura que cada pieza de trabajo es cubierta por al menos un conductor, donde los A_{ij} identifican las piezas de trabajo las cuales son cubiertas por los turnos, A_{ij} vale 1 si el turno j es cubierto por la pieza de trabajo i, y vale 0 en cualquier otro caso.

Para resolver el modelo hay dos técnicas en función del número de turnos disponibles, aunque al igual que en los apartados anteriores se utilizara el algoritmo de generación de columnas explicado anteriormente.

2.2.2 Crew Scheduling con Rostering

2.2.2.1 Organización de vehículos (Vehicle Scheduling)

Sea N={1,2,…,n} el conjunto de viajes, numerados de acuerdo al incremento del tiempo desde un punto inicial, y sea E={(i,j) | i<j, i,j compatible, tanto i como j pertenecen a N} el conjunto de arcos correspondientes a los tiempos de incorporación y descanso.

$$\min \sum_{(i,j)\in A} c_{ij} y_{ij} \qquad (1)$$

$$\sum_{j:(i,j)\in A} y_{ij} = 1 \qquad \forall i \in N \qquad (2)$$

$$\sum_{i:(i,j)\in A} y_{ij} = 1 \qquad \forall j \in N \qquad (3)$$

$$y_{ab} + y_{bc} \leq 1 \qquad \forall (a,b,c) \in R \qquad (4)$$

$$y_{ij} \in \{0,1\} \qquad \forall (i,j) \in A \qquad (5)$$

Ecuación 4: Modelo Vehicle Scheduling

C_{ij} es el coste del vehículo de arco (i,j) ∈ A, el cual suele ser parte de la función de los viajes y del tiempo de inactividad. Por otra parte, un costo fijo K para utilizar un vehículo puede ser añadido al coste de los arcos (s,i) o (j,t) para todo i,j ∈ N.

Se usa la variable de decisión y_{ij}, con y_{ij} =1 si un vehículo cubre el viaje j inmediatamente después que el viaje i, y_{ij} =0 en otro caso.

Las restricciones (2) y (3) aseguran que cada viaje es asignado exactamente a un predecesor y a un sucesor, esto es, estas restricciones garantizan que la red está dividida en una serie de caminos separados desde s a t.

En la restricción (4) se ve que los viajes a, b y c no pueden ser asignados a un mismo vehículo.

Ejemplo: Se tiene una línea de autobús formada por 12 viajes, cada viaje tiene una duración de 30 minutos, nuestro objetivo es conocer cuántos autobuses son necesarios para tener un coste mínimo y satisfacer las necesidades del problema.

Para cada viaje se especifica la hora de salida y la hora de llegada, C_{ij} tiene el valor del tiempo de espera de un viaje a otro, por cada 5 minutos de espera el coste incrementa en una unidad. También vamos a poner un coste 10 para el trayecto que realiza un autobús desde el garaje hacia el primer viaje, y un coste 6 para el trayecto desde el último viaje al garaje.

(1) 9:00_____9:30	(2) 9:15_____9:45	(3) 9:20_____9:50
(4) 9:25_____9:55	(5) 9:35_____10:05	(6) 9:50_____10:20
(7) 9:55_____10:25	(8) 10:10_____10:40	(9) 10:25_____10:55
(10) 10:30_____11:00	(11) 10:45_____11:15	(12) 11:05_____11:35

La solución óptima para este ejemplo sería la utilización de 4 autobuses. El autobús 1 realizaría los viajes 1, 5, 8 y 11, el autobús 2 los viajes 2, 6 y 9, el autobús 3 los viajes 3, 7, 10 y 12, y el autobús 4 solo realiza el viaje 4.
Al viaje inicial le llamaremos viaje 0 y el viaje final será el viaje 13.

F.O. $c_{01}y_{01}+ c_{02}y_{02}+ c_{03}y_{03}+ c_{04}y_{04}+ c_{15}y_{15}+ c_{26}y_{26}+ c_{37}y_{37}+ c_{413}y_{413}+ c_{58}y_{58}+ c_{69}y_{69}+ c_{710}y_{710}+ c_{811}y_{811}+ c_{913}y_{913}+ c_{1012}y_{1012}+ c_{1113}y_{1113}+ c_{1213}y_{1213} = 10+10+10+10+1+1+1+6+1+1+1+1+6+1+6+6 = 72$

S.A. $Y_{15}+Y_{16}+Y_{17}+Y_{18}+Y_{19}+Y_{110}+Y_{111}+Y_{112}+Y_{113}=1$

$Y_{26}+Y_{27}+Y_{28}+Y_{29}+Y_{210}+Y_{211}+Y_{212}+Y_{213}=1$

$Y_{37}+Y_{38}+Y_{39}+Y_{310}+Y_{311}+Y_{312}+Y_{313}=1$

$Y_{48}+Y_{49}+Y_{410}+Y_{411}+Y_{412}+Y_{413}=1$

$Y_{58}+Y_{59}+Y_{510}+Y_{511}+Y_{512}+Y_{513}=1$

$Y_{69}+Y_{610}+Y_{611}+Y_{612}+Y_{613}=1$

$Y_{710}+Y_{711}+Y_{712}+Y_{713}=1$

$Y_{811}+Y_{812}+Y_{813}=1$

$Y_{912}+Y_{913}=1$

$Y_{1012}+Y_{1013}=1$

$Y_{1113}=1$

$Y_{1213}=1$

$$Y_{01}=1$$
$$Y_{02}=1$$
$$Y_{03}=1$$
$$Y_{04}=1$$
$$Y_{05}+Y_{15}=1$$
$$Y_{06}+Y_{16}+Y_{26}=1$$
$$Y_{07}+Y_{17}+Y_{27}+Y_{37}=1$$
$$Y_{08}+Y_{18}+Y_{28}+Y_{38}+Y_{48}+Y_{58}=1$$
$$Y_{09}+Y_{19}+Y_{29}+Y_{39}+Y_{49}+Y_{59}+Y_{69}=1$$
$$Y_{010}+Y_{110}+Y_{210}+Y_{310}+Y_{410}+Y_{510}+Y_{610}+Y_{710}=1$$
$$Y_{011}+Y_{111}+Y_{211}+Y_{311}+Y_{411}+Y_{511}+Y_{611}+Y_{711}+Y_{811}=1$$
$$Y_{012}+Y_{112}+Y_{212}+Y_{312}+Y_{412}+Y_{512}+Y_{612}+Y_{712}+Y_{812}+Y_{912}+Y_{1012}=1$$

La forma de resolver este modelo es con optimización subgradiente y con la relajación de Lagrange, donde se relaja la condición 4. Entonces el resto del problema es casi una asignación, que se resuelve en cada iteración de la optimización subgradiente con un algoritmo de subasta. De esta manera obtenemos un límite inferior de la solución óptima. Además, también calculamos una solución factible en cada iteración mediante el cambio de la solución del subproblema tal que la condición ida-vuelta no sea violada nunca más. Esto requiere por lo menos un vehículo más. Hemos terminado la optimización subgradiente si la diferencia entre el límite inferior y el valor de la mejor solución posible indica que hemos encontrado una (casi) solución óptima.

2.2.2.2 Organización de vehículos y personal (Vehicle and Crew Scheduling)

N es el conjunto de viajes, K representa el conjunto de conductores, y $A^s \subset A$ y $A^l \subset A$ son el conjunto de arcos cortos y largos, respectivamente. Por otra parte K(i) es el conjunto de trabajos que abarcan el viaje $i \in N$, K(i,j) denota el conjunto de las tareas correspondientes a los tiempos de incorporación $(i,j) \in A^s$ y K(i,t) y K(s,j) denotan las tareas desde el lugar final del viaje i, hacia el garaje y desde el garaje a el lugar inicial del viaje j, respectivamente. La variable de decisión y_{ij} indica si un vehículo cubre el viaje j directamente después del viaje i o no, mientras que x_k indica si el trabajo k es seleccionado en la solución o no.

$$\min \quad \sum_{(i,j)\in A} c_{ij} y_{ij} + \sum_{k\in K} d_k x_k \qquad (9)$$

$$\sum_{j:(i,j)\in A} y_{ij} = 1 \qquad \forall i \in N \qquad (10)$$

$$\sum_{i:(i,j)\in A} y_{ij} = 1 \qquad \forall j \in N \qquad (11)$$

$$\sum_{k\in K(i)} x_k = 1 \qquad \forall i \in N \qquad (12)$$

$$\sum_{k\in K(i,j)} x_k - y_{ij} = 0 \qquad \forall (i,j) \in A^S \qquad (13)$$

$$\sum_{k\in K(i,t)} x_k - y_{it} - \sum_{\{j:(i,j)\in A^l\}} y_{ij} = 0 \qquad \forall i \in N \qquad (14)$$

$$\sum_{k\in K(s,j)} x_k - y_{sj} - \sum_{\{i:(i,j)\in A^l\}} y_{ij} = 0 \qquad \forall j \in N \qquad (15)$$

$$x_k, y_{ij} \in \{0,1\} \qquad \forall k \in K, \forall (i,j) \in A \qquad (16)$$

Ecuación 5: Modelo Vehicle and Crew Scheduling

Como antes, los coeficientes objetivo c_{ij} y d_k representan el coste del vehículo de arco (i,j) ∈ A, y el coste de los trabajadores del trabajo k ∈ K, respectivamente. El objetivo es minimizar la suma total de vehículos y coste de trabajadores.

Los dos primeros conjuntos de restricciones (10) y (11) son equivalentes a la parte de casi-asignación de la formulación para el VSP discutido anteriormente.

La restricción (12) asegura que cada viaje i será cubierto por un trabajo en el conjunto K(i). Por otra parte, las restricciones (13), (14) y (15) garantizan la unión entre las tareas de los tiempos de incorporación y tiempos de incorporación en la solución, donde tiempos de incorporación correspondientes a cortos y largos arcos en A son considerados por separado.

En particular, la restricción (13) garantiza que cada tiempo de incorporación desde i a j es cubierto por un trabajo en el conjunto K(i,j). Las otras dos restricciones (14) y (15) aseguran que los tiempos de incorporación desde la localización final del viaje i a t y desde s a la localización inicial del viaje j, posiblemente correspondan a largos arcos (i,j) ∈ A son ambos cubiertos por un

trabajo. Tenga en cuenta que la estructura de estos tres últimos conjuntos de restricciones es tal que cada restricción corresponde a la selección de un trabajo de un gran conjunto de trabajos, si la correspondiente variable *y* tiene valor 1.

Ejemplo: Este modelo es la unión de dos modelos, la primera parte se trata de "Organización de vehículos" (apartado 2.2.2.1), y la segunda parte consiste en el "Modelo básico de Cubrimiento de Conjuntos" (apartado 2.2.1.1), a esto se le añaden 3 restricciones que su función es asegurar que algún conductor está asignado a los autobuses cuando se pasa de un viaje a otro, también es necesario asegurar que haya algún conductor en el trayecto que va hasta el primer viaje y en el que va desde el ultimo viaje al garaje.

2.2.2.3 Organización de conductores de autobús (The bus driver scheduling problem)

El problema de organización de conductores de autobús es una parte extremadamente compleja de los procesos de planificación de las compañías de transporte. El gran tamaño de los problemas reales ha conducido al desarrollo de un gran número de modelos y técnicas que se aplican en la práctica, de acuerdo a la complejidad y las características particulares de cada empresa.

Cabe señalar que el verdadero problema de organización de conductores no puede ser simplemente como un Cubrimiento de Conjuntos o un problema de Particiones, es necesario añadir restricciones con dificultad adicional.

$$\text{Minimise} \quad \sum_{j \in P} c_j x_j$$

$$\text{Subject to} \quad \sum_{j \in P} a_{ij} x_j \geq 1 \qquad \forall i \in I \qquad (1)$$

$$\sum_{j \in P} b_{ij} x_j \geq u_j \qquad \forall i \in I \qquad (2)$$

$$x_j \in \{0,1\} \qquad \forall j \in P \qquad (3)$$

Ecuación 6: Modelo The bus driver scheduling problem

Donde:

I= {i: pieza de trabajo};

P= {j: trabajo candidato viable};

c_j = coste del trabajo viable j;

x_j = 1, si el trabajo j está en la solución,

= 0, cualquier otro caso;

a_{ij}= 1, si la pieza de trabajo i pertenece al trabajo j,

= 0, cualquier otro caso;

$b_{ij,}$ u_j = valores reales que representan las propiedades de funcionamiento de la solución.

La restricción (1) impone que cada pieza de trabajo es parte de al menos un servicio de la solución.

Las restricciones (2) obligan a satisfacer algunas propiedades de funcionamiento de la solución (Ej. Porcentaje de tipos de servicios, número de servicios o máximo tiempo trabajando)

2.2.3 Otros métodos de resolución

2.2.3.1 Modelo de Asignación Relajado (Relaxed set partitioning model)

En este modelo se propone una relajación del modelo Cubrimiento de Conjuntos en el cual sobran o son descubiertas las piezas de trabajo, las piezas de trabajo no asociadas o cubiertas por cualquier posible trabajo, son permitidas. Los restos son evidentemente indeseables, puesto que corresponden a viajes que no se les han asignado conductores, por lo que deben ser sancionadas en la función objetivo. En la práctica, los restos son asignados a la plantilla en tiempos extras de trabajo, o se agrupan con otros restos de la ruta, a veces haciendo caso omiso de las restricciones.

Minimise $\sum_{j \in P} c_j x_j + \sum_{i \in I} z_i y_i$

Subject to $\sum_{j \in P} a_{ij} x_j + y_i = 1 \qquad \forall i \in I$

$x_j \in \{0,1\} \qquad \forall j \in P$

$y_i \in \{0,1\} \qquad \forall i \in I$

Ecuación 7: Modelo de asignación relajado

Donde:

I= {i: pieza de trabajo};

P= {j: trabajo candidato viable};

c_j = coste del trabajo viable j;

z_i = penalización asociada con no abarcar la pieza de trabajo i;

x_j = 1, si el trabajo j está en la solución,

= 0, cualquier otro caso;

y_i = 1, si la pieza de trabajo i no está cubierta,

= 0, cualquier otro caso;

a_{ij}= 1, si la pieza de trabajo i pertenece al trabajo j,

= 0, cualquier otro caso;

2.2.3.2 Modelado con coordinación entre turnos

Denotamos por un par (i,j) a una pieza de trabajo que empieza en el instante i y termina en el instante j. Solo pares (i,j) viables son considerados.

Un cuarteto (i,j,k,h) denota un día de trabajo formado por los posibles pares (i,j) y (k,h). Si (i,j) o (k,h) es una pieza nula, el día de trabajo es sin descanso o corresponde a un tiempo de incorporación.

L es el número de bloques, K es el número de piezas distintas, I es el conjunto de cuartetos (i,j,k,h) viables, $I(i,j) \subset I$ es el conjunto de cuartetos donde una de las piezas es (i,j), T_l es el conjunto de todos los tiempos k los cuales son tiempos de intercambio, tiempos de inicio o tiempos de final para el bloque l, x_{ijkh} es un entero no negativo que corresponde a el número de conductores asignados a un día de trabajo (i,j,k,h), x es el vector con componentes x_{ijkh}, $(i,j,k,h) \in I$ c_{ijkh} es el coste de un día de trabajo (i,j,k,h); y'_{ij} es una variable binaria que vale 1 si la pieza (i,j) es usada en el bloque l siendo parte del día de trabajo de un conductor, y´es el vector con componentes y'_{ij} para todos los conjuntos (i,j), y es el vector con componentes y´, l=1,…,L, y Y es el conjunto de todos los y tal que y'_{ij} =0 if $i \notin T_l$ o $j \notin T_l$, l=1,2,…,L.

$$\min \quad \sum_i C_{ijkh} \, X_{ijkh}$$

s.t. $\sum_{I(i,j)} x_{mnkh} - \sum_{I} y'_{kj} = 0$ Para todos las parejas viables (i,j) (1)

$$Dx \geq d \quad (2)$$

$$\sum_{i \in T_l} y'_{ik}{}' - \sum_{j \in T_l} y'_{kj} = b'_k \quad \text{Para todo } k \in T_l, \text{ para todo } l \quad (3)$$

Ecuación 8: Modelo con coordinación entre turnos

Donde $b'_k = -1$ si k es el tiempo inicial del bloque l

+1 si k es el tiempo final del bloque l

0 cualquier otro caso

La restricción (1) asegura que cualquier pieza (i,j) valida, usada para particiones de un bloque será usada en un día de trabajo de tipo (i,j,k,h) o (k,h,i,j).

La restricción (2) se refiere a cualquier restricción particular que puede ser cualquier problema que surja en la vida real.

En la restricción (3) b'_k vale -1 si k es el momento donde el conductor empieza a trabajar, vale 1 si k es el momento que termina de trabajar, y vale 0 en cualquier otro caso (si estaba trabajando y continua trabajando).

2.2.3.3 Problema con flujos de mínimo coste (Minimum Cost Flow Problem)

Definir G = (N,A) como una red dirigida con N siendo en conjunto de nodos y A el conjunto de arcos directos. Cada arco $(i,j) \in A$ tiene un coste c_{ij} por unidad de flujo. Por otra parte, el costo del flujo varía linealmente con el aumento del flujo. Se define u_{ij} (l_{ij}) como el máximo (mínimo) aumento de flujo en el arco $(i,j) \in A$. Finalmente, se asocia el entero b_i para cada $i \in N$, que puede ser visto como la demanda (oferta) de ese nodo si b_i es negativo (positivo). Si $b_i = 0$, el nodo i es un nodo de trasbordo.

Donde x_{ij} representa el flujo en el arco $(i,j) \in A$.

$$\min \quad \sum_{(i,j)\in A} c_{ij}\, x_{ij} \tag{1}$$

$$\text{s.t.} \quad \sum_{\{j:(i,j)\in A} x_{ij} \; - \sum_{\{j:(j,i)\in A} x_{ji} = b_i \qquad \forall i \in N \tag{2}$$

$$l_{ij} \le x_{ij} \le u_{ij} \qquad \forall (i,j) \in A \tag{3}$$

Ecuación 9: Modelo flujos de mínimo coste

El coste total tiene que ser mínimo, la entrada menos la salida en cada nodo es igual a su demanda/oferta, y el flujo en cada arco es entre su capacidad inferior y superior. Se le llama a la restricción (2) *balance de masas* y a la restricción (3) *flujo obligado.*

2.2.3.4 Problema de flujos con múltiples artículos (Multicommodity Flow Problem)

Es una extensión de el minimum cost flow problem, donde en vez de un solo producto se usan varios productos utilizando la misma red. Diferentes productos básicos tienes diferentes orígenes y destinos, y los productos tienen separados la restricción de balance de masas para cada nodo. Sin embargo, el acoplamiento de los productos es causado por el arco de capacidad (restricción flujo obligado). En el resto de este apartado suponemos que el flujo debe ser integral.

La notación es similar al apartado previo con K como el conjunto de productos y sin restricción en el límite inferior, aparece también un índice extra k para cada producto en b_i^k y c_{ij}^k, y en la variable de decisión x_{ij}^k.

$$\min \quad \sum_{k\in K} \sum_{(i,j)\in A} c_{ij}^{\,k}\; x_{ij}^{\,k}$$

$$\text{s.t.} \quad \sum_{\{j:(i,j)\in A\}} x_{ij}^{\,k} \; - \sum_{\{j:(j,i)\in A\}} x_{ji}^{\,k} \; = b_i^{\,k} \qquad \forall i \in N, \forall k \in K$$

$$\sum_{k \in K} x_{ij}^{k} \leq u_{ij} \qquad \forall (i,j) \in A$$

Ecuación 10: Modelo de flujos con múltiples artículos

2.3 Métodos aproximados de resolución

Problemas de optimización combinatoria aparecen en áreas tan diversas como posicionamiento de satélites, proyectos de ordenador, enrutamiento u organización de vehículos, ubicación de trabajadores, asignación de tareas a máquinas, etc.

Muchos de estos problemas pueden ser modelados como problemas de minimizar (o maximizar) una función cuyas variables deben obedecer a ciertas restricciones. Encontrar soluciones óptimas, o realmente aproximadas, para esos tipos de problemas es un desafío que no siempre es fácil de ser conseguido. Para algunos de esos problemas son conocidos métodos eficientes (rápidos) para resolverlos; para otros, principalmente para problemas reales, métodos de enumeración implícita y relajación son algunas de las aplicaciones con mayor éxito.

En problemas de organización de los trabajadores, como en otros problemas, es un problema de minimización, o sea, para una solución se debe minimizar una función respetándose las restricciones impuestas. Una forma de resolver estos problemas sería simplemente enumerar todas las soluciones posibles y quedarnos con aquella de menor costo. Pero claro, para cualquier problema de un tamaño razonadamente grande (problema real), este método resulta impracticable, ya que el número de soluciones posiblemente será muy grande. Por tanto, resulta extremadamente necesario unos métodos más inteligentes que una simple enumeración de todas las posibles soluciones.

Aquí enumeramos algunos métodos para solucionar diversos problemas de optimización.

2.3.1 Búsqueda Tabú

La búsqueda tabú es una técnica heurística que puede utilizarse en combinación con algún otro método de búsqueda para resolver problemas de optimización combinatoria con un alto grado de dificultad.

Se encarga de evitar que dicha técnica caiga en óptimos locales, prohibiendo (en un sentido más general, penalizando) ciertos movimientos. El propósito de clasificar ciertos movimientos como prohibidos (o "tabú") es para evitar que se caiga en ciclos durante la búsqueda.

La técnica se dirige inicialmente de forma directa hacia un óptimo local, pero eso no importa porque la búsqueda no se detendrá ahí, sino que se reinicializará manteniendo la capacidad inicial de identificación del mejor movimiento posible. Además, se mantiene información referente a los movimientos más recientes en una o más *listas tabú*, a fin de evitar que una cierta trayectoria previamente recorrida se repita, aunque esta prohibición es generalmente condicional y no absoluta, como veremos más adelante.

La búsqueda tabú se encuentra fundamentada en 3 cosas principales:

- El uso de estructuras flexibles de memoria basadas en atributos, diseñadas para permitir una mejor explotación de los criterios de evaluación y la información histórica de la búsqueda que lo que se conseguiría con estructuras rígidas de memoria o con sistemas carentes de memoria (como la técnica de "enfriamiento simulado" y otras técnicas aleatorias similares).

- Un mecanismo asociado de control (para emplear las estructuras de memoria) basado en la interacción entre las condiciones que limitan y hacen más flexible el proceso de búsqueda. Este mecanismo se encuentra inmerso en la técnica en la forma de restricciones y *criterios de aspiración* (un criterio de aspiración es aquél que permite que un movimiento pierda su status de "tabú" debido a que proporciona una mejor solución que la actual).

- La incorporación de memorias de diferente duración (de corto a largo plazo), para implementar estrategias que intensifiquen y diversifiquen la búsqueda.

2.3.1.1 Diagrama de flujo

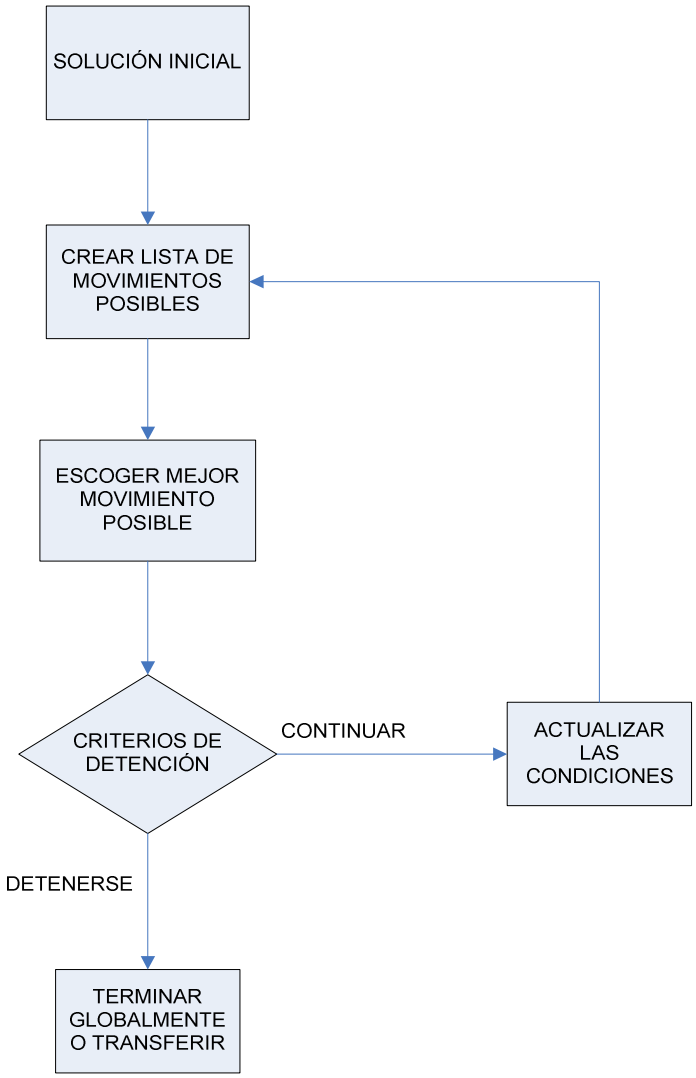

Figura 2.2: Diagrama de flujo de la búsqueda tabú

En la búsqueda tabú se empieza con una solución inicial, que se obtiene a partir de los valores iniciales o del componente de memoria de mediano o largo plazo. A partir de aquí se crea una lista de movimientos posibles, con la que se generará una nueva solución a partir de la actual. La lista de movimientos posibles se basa en las restricciones tabú y en los criterios de aspiración. La solución obtenida pasa a ser la solución actual, y se debe almacenar como la mejor solución si su valor supera el mejor valor registrado hasta el momento. A continuación hay que pasar por unos criterios de detención, donde habrá que detenerse si se ha alcanzado un cierto número máximo de iteraciones, o si la última mejor solución ha sido encontrada. Si en vez de detenerse se continua, hay que actualizar las restricciones tabú y los criterios de aspiración.

Un paso crítico es la selección del mejor movimiento posible. La siguiente figura ilustra este mecanismo.

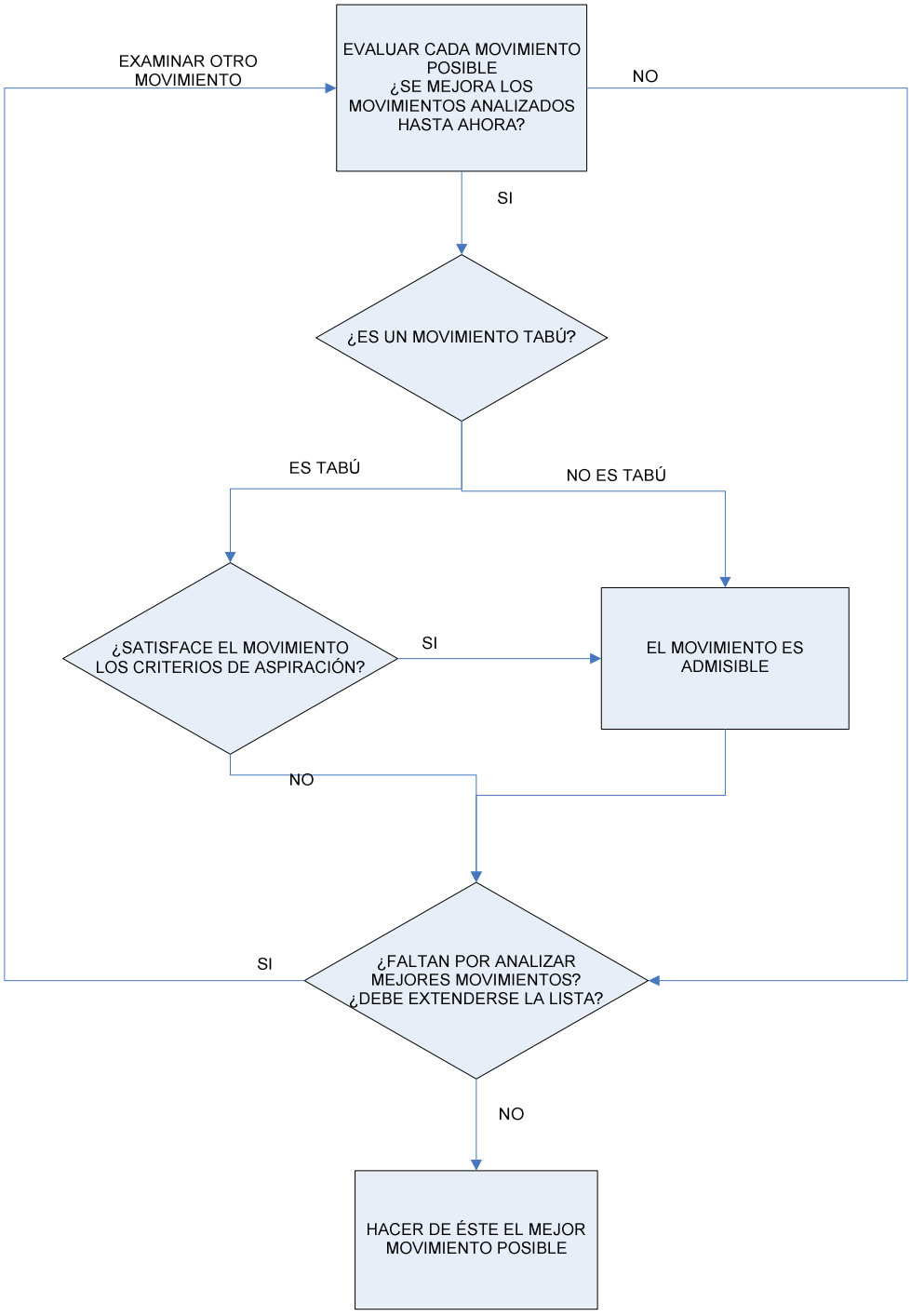

Figura 2.3: Diagrama de flujo para mejor movimiento posible

Lo primero que hay que preguntarse al evaluar cada movimiento posible es ¿Produce un valor más alto que cualquier otro movimiento analizado hasta ahora? Si la respuesta es negativa habría que volver a analizar la lista de movimientos posibles, en cambio si es positiva la siguiente pregunta que habría que resolver sería ¿Es un movimiento tabú? Si no lo es el movimiento se considera admisible, si fuera un movimiento tabú se tendría que verificar si satisface los criterios de aspiración, y ya se podría considerar un movimiento admisible o se tendría que analizar de nuevo la lista de movimientos posibles. Para determinar si hay que examinar otro movimiento o en cambio éste es el mejor movimiento posible, bastaría con hacerse la pregunta ¿Hay una "buena probabilidad" de que falten por analizar mejores movimientos, o debiera extenderse la lista de movimientos posibles?

2.3.1.2 Niveles de Aspiración

La lista tabú puede prohibir movimientos deseables que no produzcan ciclarse o también nos puede llevar a un punto en donde no es posible moverse. Todos los algoritmos de búsqueda tabú permiten revocar o cancelar tabús. A estos se les llama criterios de aspiración, que permiten movimientos, aunque sean tabú.

Lo más común es permitir movimientos que producen una mejor solución que la mejor solución actual, o sea si se hace un movimiento que va de una x a s(x) si c(s(x)) < MEJOR(c(x)).

Si se piensa en términos de atributos para describir estados, se pueden tener listas tabú para cada atributo y una función de aspiración que depende de cada atributo. Si uno o más atributos pasan la prueba individual de aspiración, entonces se puede asumir que los demás atributos automáticamente también la pasan.

2.3.1.3 Intensificación y Diversificación

La idea de intensificación es buscar mas es porciones del espacio que aparentan ser mejores o más prometedoras. Cada determinado tiempo se puede realizar un proceso de intensificación. Normalmente se reinicia la búsqueda a partir de la mejor solución actual. Algunas opciones:

- Mantener fijos los componentes o atributos que parecen mejores.
- Cambiar el esquema de vecindad.

Una extensión es considerar los comportamientos de los patrones producidos por una lista tabú. Uno de los efectos de esto es analizar por ejemplo movimientos oscilatorios evaluando lo atrayente de los movimientos dependiendo de la localización y dirección de la búsqueda. Con esto se puede por ejemplo especificar un número de movimientos necesarios en una cierta dirección antes de permitir algún retorno.

Las funciones de memoria a mediano plazo sirven para registrar y comparar atributos de las mejores soluciones obtenidas durante un periodo de búsqueda. Los atributos comunes puedes servir para guiar nuevas soluciones a espacios en donde existan tales atributos.

Diversificación obliga buscar en áreas no exploradas. Las funciones de memoria a largo plazo tratan de diversificar la búsqueda, empleando principios más o menos contrarios a los de memoria a mediano plazo. La idea es explorar regiones que contrastan fuertemente con las regiones exploradas hasta el momento. No se trata de inyectar diversidad aleatoria, sino tomando en cuenta el proceso de búsqueda llevado hasta ese momento.

Para escapar de atractores fuertes se requiere de un esfuerzo adicional. Posibilidades:

1. Imponer restricciones más fuertes en las condiciones tabú para excluir un número más grande de movimientos.

2. Usar información de cuando el proceso apunta hacia arriba y tratar de favorecer movimientos que apunten en esa dirección.

3. Penalizar movimientos que usan atributos muy usados en el pasado.

Una vez que se sale del atractor, se pueden "relajar" las condiciones. Conceptualmente, lo que tratan de estimar es una distancia de escape del óptimo local.

2.3.2 Enfriamiento Simulado

Antes de explicar cómo funciona el algoritmo de enfriamiento simulado hay que entender unas nociones básicas. Estado inicial: es el estado en el que comienza el problema. Espacio de estados: cada estado alcanzable a partir del estado inicial.

Este es un tipo de algoritmo de búsqueda por escalada. Estos algoritmos usan la información del estado actual del problema para hacer la búsqueda en el espacio de estados. A partir de la información del estado actual se intenta llegar a otro estado del proceso que sea mejor, y de esta forma nos vamos acercando a un estado solución óptima. El estado solución puede que no sea el mejor de todos pero si uno lo suficientemente bueno. El estado suele reducirse a un número que nos indica como de lejos estamos de un estado solución del problema, cuanto más grande es el número más lejos estamos de la solución con lo que un estado mejor al actual sería aquel que tenga un número menor.

Los algoritmos de búsqueda por escalada tienen algunos problemas:

- Óptimo local: cuando se llega a un estado que es mejor que todos sus vecinos y descendientes, pero existe otro estado mejor en alguna parte.

- Meseta: todos los estados vecinos tienen el mismo valor, con lo cual no sabemos hacia dónde ir.

- Puente: nos encontramos en un óptimo local y cerca tenemos un nodo solución mejor, sin embargo para llegar al óptimo global tendríamos que pasar por un nodo inaccesible lo que es imposible.

Este algoritmo se plantea como un problema de minimización de la función a optimizar (esta función es la que nos da el valor de cada estado). Existe una temperatura T que nos permite movernos a estados con valor peor al del estado actual, cuanto mayor es T más facilidad tendremos de hacer esto. Al principio T es grande y según se avanza en el proceso va disminuyendo, al final es tan baja que el algoritmo se comporta como uno de escalada simple. Un algoritmo de escalada simple busca siempre un nodo mejor, y nunca tiene la opción de moverse a un estado peor que el actual.

Este enfriamiento permite que, si es lo suficientemente lento, lleguemos a un óptimo absoluto y no nos quedemos en un óptimo local del problema. En metalurgia, se sigue este proceso para templar metales y cristales calentándolos a

una temperatura alta y luego enfriándolos gradualmente, para que el material se funda en un estado cristalino de energía baja.

2.3.3 Generación de columnas

La generación de columnas es una técnica basada en el principio de descomposición de Dantzig-Wolfe, que es una técnica de programación lineal generalizada. En esta técnica, un problema maestro (PM) permite la selección de un mejor subconjunto de columnas (ej. variables de decisión), y es resuelto por una formulación de programación lineal (PL) y normalmente su versión relajada. Ese problema, a su vez, tiene su propio modelo, basado frecuentemente en un modelo de cobertura o particionamiento de conjuntos con restricciones adicionales.

Conocida hace varias décadas por su capacidad de resolver problemas grandes de PL, la idea principal de esta técnica consiste en: considerar un pequeño número de variables (columnas) cada vez; resolver el problema de PL y obtener la solución primaria; generar nuevas columnas interesantes, con coste reducido negativo (sub-problema), a través de la solución primaria del problema principal para mejorar la solución de PL anterior; y repetir ese proceso hasta que ninguna mejora pueda ser obtenida en la solución de PL.

2.3.4 Algoritmos genéticos

Un algoritmo genético es un método de búsqueda dirigida basada en probabilidad. Bajo una condición muy débil (que el algoritmo mantenga elitismo, es decir, guarde siempre al mejor elemento de la población sin hacerle ningún cambio) se puede demostrar que el algoritmo converge en probabilidad al óptimo. En otras palabras, al aumentar el número de iteraciones, la probabilidad de tener el óptimo en la población tiende a 1 (uno).

Los algoritmos genéticos establecen una analogía entre el conjunto de soluciones de un problema, llamado fenotipo, y el conjunto de individuos de una población natural, codificando la información de cada solución en una cadena, generalmente binaria, llamada cromosoma. Los símbolos que forman la cadena son llamados los genes. Cuando la representación de los cromosomas se hace con cadenas de dígitos binarios se le conoce como genotipo. Los cromosomas evolucionan a través de iteraciones, llamadas generaciones. En cada generación, los cromosomas son evaluados usando alguna medida de aptitud. Las siguientes

generaciones (nuevos cromosomas), llamada descendencia, se forman utilizando dos operadores, de cruzamiento y de mutación.

Los algoritmos genéticos son de probada eficacia en caso de querer calcular funciones no derivables (o de derivación muy compleja) aunque su uso es posible con cualquier función. Deben tenerse en cuenta también las siguientes consideraciones:

- Si la función a optimizar tiene muchos máximos/mínimos locales se requerirán más iteraciones del algoritmo para "asegurar" el máximo/mínimo global.

- Si la función a optimizar contiene varios puntos muy cercanos en valor al óptimo, solamente podemos "asegurar" que encontraremos uno de ellos (no necesariamente el óptimo).

Algoritmo genético constructivo

El algoritmo genético constructivo (AGC) fue, primeramente, abordado por Lorena y Lopes (1996). El aspecto constructivo de AGC está relacionado a los hechos de, a lo largo del proceso selectivo, una población de soluciones incompletas (esquemas) y utilizarlas como base para la construcción de una población de soluciones completas (estructuras). Tanto esquemas como estructuras compiten entre si durante el proceso evolutivo. Las estructuras están formadas a partir de recombinación de esquemas y, entonces, se tiene una mejora tanto de estructuras como de esquemas a lo largo de la generación. Otras características importantes e innovadoras de AGC están en la utilización de poblaciones de tamaño variable a lo largo de la generación.