



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA MATEMÁTICA

EL PROBLEMA DE LOS PROVEEDORES CAPACITADOS CON MENOR SUMA DE
RADIOS EN LA LÍNEA

TESIS PARA OPTAR AL GRADO DE MAGÍSTER EN CIENCIAS DE LA
INGENIERÍA, MENCIÓN MATEMÁTICAS APLICADAS

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL MATEMÁTICO

KEVIN EDGAR CONTRERAS MAYR

PROFESOR GUÍA:
JOSÉ SOTO SAN MARTÍN

MIEMBROS DE LA COMISIÓN:
IVÁN RAPAPORT ZIMERMANN
WALDO GÁLVEZ VERDUGO

Este trabajo ha sido parcialmente financiado por ANID FONDEF ID18I10250,
FONDECYT REGULAR 1181180, CMM ANID/PIA ACE210010 y FB210005

SANTIAGO DE CHILE

2022

Resumen

RESUMEN DE LA TESIS PARA OPTAR AL GRADO DE MAGÍSTER CIENCIAS DE LA INGENIERÍA, MENCIÓN MATEMÁTICAS APLICADAS,
MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL MATEMÁTICO
POR: KEVIN EDGAR CONTRERAS MAYR
FECHA: 2022
PROFESOR GUÍA: JOSÉ SOTO SAN MARTÍN

EL PROBLEMA DE LOS PROVEEDORES CAPACITADOS CON MENOR SUMA DE RADIOS EN LA LÍNEA

El presente trabajo estudia un problema de asignación de clientes a centros de atención: dado un conjunto de clientes en un grafo métrico, y un conjunto de centros en los cuales hay servidores disponibles, se desea asignar a cada cliente su centro de atención, de manera de minimizar el tiempo que toma a los servidores de cada punto atender a los clientes asignados. Distintas restricciones y formas de medir el tiempo de atención de los servidores definen tres variantes: makespan, soft-capacidades y hard-capacidades. Se estudia la relación que tienen estas variantes entre sí: en particular, la variante makespan se reduce (perdiendo un factor de $\frac{3}{2}$) a la variante con soft-capacidades. Posteriormente el estudio se concentra en un tipo particular de instancia, la línea, para la cual se detalla un algoritmo exacto para la variante con soft-capacidades, y una 2-aproximación para la variante con hard-capacidades.

*A mis padres, quienes me enseñaron a vivir,
y al profe Carlos, quien me enseñó a sumar.*

Agradecimientos

Gracias, mi Dios, por darme la vida, la capacidad de estudiar, y ser el Ser con más paciencia que me ha amado. Esta tesis no sería posible sin ti, y no es digna de ti, pero lo mismo aplica hacia mí como persona y me sigues ayudando. ¡Me gustaría escribir mucho más de un párrafo!

Gracias, mamá y papá, por haberme criado con curiosidad y alimentado mis ganas de leer y aprender. También, por tenerme casi tanta paciencia como el Señor, y por amarme también casi tanto como él.

Gracias, abuelos, abuelas, tíos, hermanas, primos, etcétera. Son la mejor familia en que he nacido, y aunque no nos podamos ver todos ahora, el amor sobrepasa toda barrera. ¡Los quiero mucho!

Gracias Débora, Madrid, Mati, Fran, Karen, Valentín, Fabro, Cony, Juan, Jomar, Rayén, Scarlet, Polyn... amigos cercanos, gente de altísima calidad, gracias a los cuales soy ahora como soy. Gracias Matus, Diego, Bisente, Suil, Bidim, Obed, y demás compañeros de la oficina, por el compañerismo en la tele y la matraca, en los rojos y azules. Gracias Panchito, Araceli, Tyara, que además me dieron buenas ideas para el desarrollo de la tesis y me motivaban a avanzar, cada uno a su estilo.

Gracias Carlos, Belfor, Nicolás, Nelson, Luisa, y tantos otros profesores notables en distintos colegios. Sus caras de a poco se me olvidan, pero sus enseñanzas permanecen.

Gracias José, profe guía, y Daniela, psicóloga, quienes también me tuvieron no solo la paciencia que merecía, sino la que necesitaba. Gracias Iván y Waldo, miembros de la comisión, que me ayudaron a ajustar los últimos detalles.

Gracias amigos de la agrupación de adventistas de la Chile. Desde los seis primeros hasta las docenas de hoy, todos me han dado un buen consejo, una palabra divina, una amonestación amorosa, un rato agradable, una colacioncita. ¡Sigámonos viendo!

Gracias al proyecto Anid fondef y fondecyt, y al CMM, por el financiamiento mediante los proyectos CMM ANID PIA AFB170001, CMM ANID BASAL ACE210010 y CMM ANID BASAL FB210005.

Gracias a tí, que lees esto: quizá no cupo tu nombre acá (¡lo siento! pero sí me acordé, pregúntame), o quizá no te conozco, pero nadie lee los agradecimientos sin tener algo de curiosidad por la persona detrás del trabajo, así que me caes bien.

Tabla de Contenido

Introducción	1
1. Preliminares	4
1.1. Definición del problema	4
1.2. Tres variantes del problema	5
1.2.1. Variante makespan	5
1.2.2. Variante hard-capacitada	5
1.2.3. Caso soft-capacitado	7
1.3. Dificultad de las variantes	8
1.4. Interpretación de las figuras y nomenclatura	9
1.4.1. Nuestros resultados	10
2. Algunos resultados generales	11
2.1. Casos polinomiales	11
2.1.1. Un solo servidor	11
2.1.2. Dos servidores, versión soft-capacitada	12
2.2. Reducciones entre las variantes	14
2.2.1. Makespan versus soft-capacitada	14
2.2.2. Soft versus hard-capacitada	18
3. Caso lineal	20

3.1. Definición de WIPC y notación específica	20
3.2. Solución para el caso $p = \infty$	21
3.3. Solución de soft CSORS en la línea	24
3.3.1. Caso uniforme con $p = 2$	26
3.3.2. Caso uniforme con $p > 2$	27
3.3.3. Capacidades arbitrarias	31
3.4. Caso hard-capacitado	32
3.4.1. Construcción de un algoritmo de aproximación	32
3.4.2. Propiedades adicionales de la variante hard-capacitada	40
Conclusión	44
Bibliografía	45

Índice de Tablas

3.1. Resumen de la división en casos de la Proposición 3.13. Los casos triviales o infactibles para $i_1 \leq i_2$ se muestran en gris, los casos infactibles para $a = p$ en violeta. Los casos en azul se cumplen para cualquier elección de a , y los casos en verde, con $a = r - p$	38
--	----

Índice de Ilustraciones

1.1. En esta solución, los centros i_1 e i_2 están atendiendo a los clientes j_1 y j_2 respectivamente.	10
2.1. Ejemplo de la reducción de 2-CSORS a Matching perfecto de peso mínimo. El grafo de la izquierda, G , es una instancia de 2-CSORS. A la derecha está su correspondiente instancia G' de Matching. Ambos tienen destacado un óptimo, de costo 3.	13
2.2. En esta instancia de hard 2-CSORS, debe haber un par de clientes atendidos desde un centro que no es el más cercano.	14
2.3. Ejemplo de la atención definida en el paso 4. La matriz $A^{(i)}$ de la izquierda se traduce en la atención representada en el grafo de la derecha: las aristas de color negro, verde y morado representan al primer, segundo y tercer servidor respectivamente.	16
3.1. Las dos maneras (salvo reetiquetados) en que los intervalos pueden no ser disjuntos.	23
3.2. En esta instancia, con $p = 2$, la atención óptima consiste en atender desde cada centro a los clientes del mismo color: los correspondientes intervalos de atención no son disjuntos	24
3.3. Dos intervalos que violan la laminaridad. La inducción es en el número de clientes en la intersección, representados en blanco.	25
3.4. Caso base de la inducción en el Lema 3.5	25
3.5. Paso inductivo del Lema 3.5	26
3.6. Representación gráfica de la demostración de la Proposición 3.7. En la columna de la izquierda, vemos que es posible ‘llevar’ a j a ser vecino de r_2 ; en la columna derecha, que podemos intercambiar estos clientes, disminuyendo el cardinal de $S(J_1) \cup J_2$	28

3.7. Ilustración del segundo caso de la demostración del Lema 3.9. Si el intervalo $[l, r]$ atiende a menos de p clientes (arriba), puede ‘absorber’ alguno de los de sus intervalos interiores (abajo).	30
3.8. En esta instancia de <i>CSORS</i> , las capacidades están escritas bajo cada centro. El óptimo activa dos intervalos, ninguno de los cuales es atendido desde el centro más cercano a su punto medio.	31
3.9. En la instancia de hard 2-CSORS de la figura, σ_1 y σ_2 son atenciones óptimas para C y para \mathcal{C} respectivamente. Notamos que $C(\sigma_1) = 1 + 4\epsilon$, mientras que $C(\sigma_2) = 2 + 2\epsilon$; mostrando que el factor 2 de esta aproximación es ajustado.	34
3.10. Ilustración de la situación del Lema 3.12.	34
3.11. Una ilustración con los casos del Lema 3.16, en el mismo orden del texto.	42
3.12. Ilustración de la demostración de la segunda propiedad del Lema 3.16. Las líneas punteadas representan los cambios. En el grafo de arriba, se ve que el ‘desplazamiento’ del primer intervalo reduce su costo; en el de abajo, que la separación de éstos produce el mismo efecto.	42
3.13. En esta instancia, los extremos de J_1 y J_2 están marcados con los colores de i_1 e i_2 respectivamente. Notamos que la solución es óptima, e $i_2 \notin J_2$ pese a que J_2 contiene un centro.	43

Introducción

Dentro de la selva de problemas de optimización combinatorial, que se ha vuelto más y más tupida los últimos años, uno de los géneros con más aplicaciones y variantes es el de problemas de *ubicación de centros*.

Estos problemas estudian la siguiente pregunta: dado un conjunto de clientes que están en un grafo métrico, y la posibilidad de abrir centros de atención, ¿dónde deben abrirse para que la atención sea lo más rápida posible? Si los clientes se atienden secuencialmente en cada centro, el problema se conoce como *Facility Location*, con aplicaciones como la ubicación óptima de bodegas comerciales [30, p. 91]. En otras ocasiones, se desea minimizar el máximo tiempo requerido para atender a algún cliente: Por ejemplo, cuando el centro corresponde a un hospital o cuartel de bomberos, se busca que las ambulancias o camiones lleguen lo más rápido posible a la casa o zona donde ocurre la emergencia. Esto da lugar a problemas como *Minimax Facility Location* [6] o el más general *k-center problem* [30, p.37]. Cuando los centros deben estar conectados, también hay aplicaciones en diseño de redes [13].

Una pregunta natural de hacer es: ¿qué pasa cuando cada centro tiene una capacidad, es decir, puede atender cierta cantidad de clientes simultáneamente? Por ejemplo, podríamos considerar una cadena de supermercados con un cierto número de cajeros cada uno; o un camión de repartición de bebidas con un puñado de peonetas, que luego de estacionar en un lugar apropiado pueden entregar la carga cada uno a un local distinto. En el primer caso, el problema más similar en la literatura es *Soft-capacitated Facility Location* [21]; el segundo caso tiene la particularidad de que aunque deseamos minimizar el máximo tiempo de atención, la función objetivo es la *suma* de estos tiempos máximos por local. En ese sentido, la pregunta da pie a problemas como *capacitated sum-of-radii clustering* [16].

Dentro de esta última categoría, surge otra pregunta: asumiendo que un centro tiene asignados a sus clientes, y consta de p empleados que trabajan independientemente, ¿cómo “repartirse” a los clientes para demorarse lo menos posible, considerando los diferentes tiempos de atención, o distancias de los clientes al centro? Es decir, interpretamos cada centro como una instancia de *Asignación de tareas a máquinas paralelas* [25, p.111]. Dado que este problema ya es NP-difícil por sí solo, adicionalmente surge la duda de cómo se relaciona este problema a la versión que no incorpora este elemento.

Es posible también plantear una última diferencia desde *Capacitated sum-of-radii clustering*: en éste el centro de un cluster es necesariamente uno de los clientes; en cambio, podríamos definir el conjunto de centros y el de clientes como objetos separados. Esta característica es también la que define el problema de *k-suppliers* [30, p.57], una variante del más conocido

problema *k-center*.

Al problema resultante de estas consideraciones le llamamos en este trabajo *Capacitated Sum-Of-Radii Suppliers* (en adelante CSORS). Distinguimos tres variantes, de acuerdo a las distintas interpretaciones de la capacidad de cada centro: número de trabajadores que atienden en paralelo (variante makespan); hard-capacitada, donde cada centro puede atender un número fijo de clientes; y soft-capacitada, en la que permitimos que cada centro se abra múltiples veces.

Esta tesis se concentra en la siguiente simplificación: si el grafo de clientes y centros es un camino, es posible resolver de manera polinomial la variante soft-capacitada, y construir una aproximación del óptimo para la variante hard-capacitada. Veremos también que en el caso general, la variante makespan se puede reducir a la soft-capacitada, perdiendo un factor de aproximación no mayor a $\frac{3}{2}$. Este factor mejora conforme aumenta la capacidad de los centros.

Trabajos relacionados

Un recuento de los modelos principales para *Facility Location* se puede encontrar en [28]. Para la versión conectada, dos algoritmos destacables están en [27] y [8].

Para el *k-center problem* existe una 2-aproximación, que es óptima [14]. Algunas variantes, como una versión capacitada, se tratan en [18]. Para *k-median* también existen aproximaciones constantes [2] [19]. Existen también aproximaciones FPT [18]

Sum-of-radii clustering también ha sido estudiado. Por ejemplo, ver [3], [11]. Para la versión capacitada hay una aproximación FPT [17]

El problema de asignación de tareas a máquinas paralelas también es muy estudiado: un recuento de los resultados más importantes hasta el 2010 puede verse en [4]. Trabajos más recientes incluyen el desarrollo de algoritmos exactos [22], restricciones adicionales como ventanas de tiempo [1], cambios en la función objetivo [9], o la incorporación de más variables, como recursos flexibles [7].

Es posible extender *k-CSORS* pidiendo que los centros abiertos deben estar conectados por un árbol o un ciclo: la primera extensión incorpora elementos de *Terminal Steiner Tree*, el problema de conectar clientes con un árbol cuyos nodos interiores son centros; la segunda se puede ver como una versión ‘Steiner’ del clásico *Travelling Salesman Problem*, en el que se busca la ruta más corta que pase por todos los clientes: en esta versión, la ruta en vez de pasar por clientes pasa por los centros, y añadimos la dificultad de que desde cada centro hay múltiples servidores atendiendo. Algunos resultados para *Terminal Steiner Tree* se pueden ver en [20] y [5]. Existen también estudios sobre el *Steiner Travelling Salesman Problem* [26].

Nuestros resultados

El trabajo de esta tesis se concentra principalmente en el caso en que los clientes y centros de atención están en *una sola línea*, es decir, que el grafo subyacente es un camino. Esta restricción es lo suficientemente fuerte como para que la variante soft-capacitada tenga solución en tiempo polinomial. (Teorema 3.10). También se construye una 2-aproximación para el caso hard-capacitado (Sección 3.4.1). El primer resultado es válido para capacidades arbitrarias en los centros; el segundo, para una capacidad fija cualquiera. Además, para el caso general, establecemos una reducción de la variante makespan a la variante soft-capacitada, con un factor de aproximación de $\frac{3}{2}$ (Teorema 2.4).

La estructura de esta tesis es como sigue: en el primer capítulo se definen formalmente los problemas y la notación a ocupar el resto del documento; en el capítulo siguiente establecemos algunos resultados para el caso general. Por último, el capítulo 3 estudia el problema restringido a un grafo lineal.

Capítulo 1

Preliminares

1.1. Definición del problema

Sea $G = K_{m,n}$ un grafo bipartito completo. A los m nodos del primer conjunto los llamaremos *centros*, y los indexaremos con $i \in \{1, \dots, m\}$. A los n nodos del segundo conjunto los llamaremos *clientes* y los indexaremos con $j \in \{1, \dots, n\}$. Anotaremos como \mathcal{I} y \mathcal{J} al conjunto de centros y de clientes respectivamente.

Consideraremos también una función de pesos w en las aristas entre \mathcal{I} y \mathcal{J} , llamada *tiempo de atención*; además, cada centro i posee una *capacidad* $p_i \in \mathbb{N} \cup \{\infty\}$. Interpretaremos esta capacidad como la presencia de p_i *servidores* en el centro i . Abreviaremos como $[p]$ al conjunto $\{1, \dots, p\}$.

Definimos el problema *Capacitated Sum-Of-Radii Suppliers* (CSORS), o simplemente problema de los proveedores, como el problema de asignar, bajo ciertas restricciones, cada cliente a algún servidor de algún centro. Cada servidor k atiende a sus clientes asignados uno tras otro, incurriendo en tiempos de atención dado por w . Definimos el tiempo de atención de un centro como la máxima suma de los tiempos requeridos por uno de sus servidores; el objetivo es minimizar la suma de los tiempos de atención de cada centro.

Definimos, para un cliente j , $\sigma(j)$ como el centro desde el cual se lo está atendiendo. Para un centro i , diremos que está *abierto* si $\sigma^{-1}(i)$ es no vacío (es decir, si alguno de sus servidores atiende a un cliente). Por conveniencia, anotaremos este conjunto como S_i . Denotaremos también, para un centro i y un servidor k , S_{ik} al conjunto de clientes atendidos desde i por k .

Reconocemos entonces tres variantes del problema, según cómo entendemos el procedimiento de atención a los clientes. Llamaremos, para cualquiera de estas tres variantes, *uniforme* a la versión en que cada centro tiene la misma capacidad p , y denotaremos el problema como p -CSORS.

1.2. Tres variantes del problema

1.2.1. Variante makespan

La restricción más natural solo pide que todo cliente sea atendido; cada servidor puede atender a múltiples clientes, secuencialmente. Podemos entonces formular el problema como el problema lineal (\mathcal{P}_m) definido abajo; interpretamos las variables binarias x_{ijk} como la decisión de que “el centro i atiende al cliente j con el servidor k ”. Las variables auxiliares y_i representan el máximo tiempo que un servidor está ocupado en i : se desea minimizar la suma de estos máximos.

$$\begin{aligned} (\mathcal{P}_m) \quad & \text{mín} \quad \sum_i y_i \\ \text{s.a} \quad & y_i \geq \sum_j w_{ij} x_{ijk} \quad \forall i \in \mathcal{I}, \forall k \in [p_i] \\ & \sum_{i,k} x_{ijk} \geq 1 \quad \forall j \in \mathcal{J} \\ & x_{ijk} \in \{0, 1\} \quad \forall i, j, k \\ & y_i \geq 0 \quad \forall i \end{aligned}$$

Llamamos a esta variante Makespan CSORS.

1.2.2. Variante hard-capacitada

Un segundo caso de interés es cuando cada centro i solo puede atender a lo más a p_i clientes, en paralelo (esto es, que cada servidor trabaja solo una vez). Podemos entonces escribir la función de costos como

$$\sum_{i \in \mathcal{I}} \max_{j \in S_i} w_{ij}.$$

Esto introduce una nueva restricción (respetar la capacidad estrictamente). Agregándola a la formulación anterior, nos queda:

$$\begin{aligned} (\mathcal{P}_h) \quad & \text{mín} \quad \sum_i y_i \\ \text{s.a} \quad & y_i \geq \sum_j w_{ij} x_{ij} \quad \forall i \in \mathcal{I} \\ & \sum_j x_{ij} \leq p_i \quad \forall i \in \mathcal{I} \\ & \sum_j x_{ij} \geq 1 \quad \forall j \in \mathcal{J} \\ & x_{ij} \in \{0, 1\} \quad \forall i, j. \end{aligned}$$

Alternativamente, podemos interpretar este problema como un problema de *clustering*: desde cada centro i escogemos un radio r_i (posiblemente 0) de forma que las bolas $B(i, r_i)$ cubran a todos los clientes, y además podemos asignar cada cliente a una bola que lo cubra de forma de respetar las capacidades de cada centro.

En otras palabras, si *restringimos* el grafo $K_{n,m}$ a solo las aristas $\{i, j\}$ tales que $w_{ij} \leq r_i$, entonces los radios r_i son factibles si y sólo si podemos encontrar una asignación $\sigma : \mathcal{J} \rightarrow \mathcal{I}$ de forma que $|S_i| \leq p_i$ para todo centro i .

No es difícil ver que la existencia de esta asignación es equivalente a la existencia de un matching perfecto en el siguiente grafo bipartito $G' = (A, B, E)$:

- A es \mathcal{J} , el conjunto de clientes, aumentado con nodos auxiliares para que su cardinal sea igual al de B .
- El conjunto B contiene p_i copias de cada centro i .
- Para j cliente e i centro, si $i' \in B$ es una copia de i , entonces la arista $e = \{j, i'\}$ pertenece a E si $w_{ij} \leq r_i$.

Si $j \in A$ no es cliente, existe una arista entre él y todo nodo de B .

Supongamos que existe una asignación factible σ . Es claro que podemos construir un matching M_σ en G' : para cada centro i , existen p_i copias en B , y $|S_j| \leq p_i$, por lo que podemos escoger una arista entre cada copia y algún cliente distinto de S_j . Las copias que sobran las podemos conectar con clientes auxiliares de $A \setminus \mathcal{J}$.

Por otro lado, dado un matching M en G' , podemos definir una asignación que atienda desde i al conjunto de todos los clientes conectados por M con una copia de i : por la construcción de G' , esta asignación es factible.

Esto da pie a una segunda formulación,

$$\begin{aligned}
 (\mathcal{P}_{h_2}) \quad & \text{mín} \quad \sum_{i,r} r y_{ir} \\
 \text{s.a} \quad & \sum_i p_i \sum_{r: \exists j \in A, w_{ij} \leq r} y_{ir} \geq |A| \quad \forall A \subseteq \mathcal{J} \\
 & \sum_r y_{ir} \leq 1 \quad \forall i \in \mathcal{I} \\
 & y_{i,r} \in \{0, 1\} \quad \forall i, r
 \end{aligned}$$

donde y_{ir} representa la decisión de abrir el centro i con un radio r . Notamos que solo hay $O(|\mathcal{I}||\mathcal{J}|)$ radios posibles, por lo que la cantidad de variables es polinomial. La primera restricción es equivalente a la condición de Hall para G' ; esto es, que todos los nodos de A estén cubiertos. La segunda asegura que cada centro se abra una sola vez.

La formulación (\mathcal{P}_{h_2}) es menos compacta que (\mathcal{P}_h) : sin embargo, las restricciones se pueden verificar en tiempo polinomial, ya que corresponden a la existencia de un matching perfecto en el grafo G' de arriba. Sabemos que

$$|B| = \sum_{i \in \mathcal{I}} p_i \leq m \max_i p_i;$$

por otro lado, $|E|$ es $O(|B|^2)$. Utilizando, por ejemplo, el algoritmo de Hopcroft-Karp [15], podemos entonces verificar las restricciones en tiempo $O(m^2 \sqrt{mp^*})$, donde p^* es la máxima capacidad de algún centro.

Llamamos a esta variante Hard CSORS.

1.2.3. Caso soft-capacitado

Por último, podemos relajar un poco las restricciones del caso hard-capacitado y permitir que cada centro se abra más de una vez. Una interpretación de esto es que cada centro i atiende en aperturas o *rondas* de tamaño p_i a sus clientes, y el costo de cada ronda es su *radio* r , definido como la mayor distancia a alguno de sus clientes atendidos. Cada servidor atiende a lo más a un cliente por ronda, y los servidores operan simultáneamente.

El número de radios posibles es a lo más nm , el número de aristas entre centros y clientes: por lo tanto, podemos formular el problema como

$$\begin{aligned}
 (\mathcal{P}_s) \quad & \text{mín} \quad \sum_{i,r} r y_{ir} \\
 & \text{s.a} \quad \sum_{i,r} x_{ijr} \geq 1 \quad \forall j \\
 & \quad \sum_j x_{ijr} \leq p_i y_{ir} \quad \forall i, r \\
 & \quad x_{ijr} \leq y_{ir} \quad \forall i, j, r \\
 & \quad x_{ijr}, y_{ir} \in \{0, 1\} \quad \forall i, j, r \geq w_{ij}
 \end{aligned}$$

donde y_{ir} representa el abrir el centro i con radio r y x_{ijr} representa el atender desde i al cliente j bajo la bola de radio r .

Llamamos a esta variante Soft CSORS.

1.3. Dificultad de las variantes

Las tres variantes, makespan, hard, y soft CSORS, son NP-difíciles en general: más aún, la versión makespan mantiene su complejidad en casos muy particulares.

Proposición 1.1 *La versión makespan es NP-difícil incluso en instancias con 1 centro y 2 servidores.*

DEMOSTRACIÓN. Consideremos un solo centro con capacidad p . Podemos interpretar cada cliente j como un *trabajo* que toma tiempo w_j , que nos interesa asignar a una de p máquinas minimizando el tiempo total de atención. Esto es justamente la definición del problema *Asignación de tareas a máquinas paralelas* [30, p.304], que se sabe NP-difícil incluso para dos máquinas. [10, p.65] \square

Para las otras dos versiones el resultado es más débil: para la NP-dificultad es suficiente que haya a lo menos 3 servidores, y que m sea arbitrario. (Veremos en el próximo capítulo que estas condiciones también son necesarias.)

Proposición 1.2 *3-CSORS, tanto en su versión hard como soft, es NP-difícil.*

DEMOSTRACIÓN. Mediante reducción desde 3-Dimensional Matching.

Recordemos que una instancia de 3-d-matching consiste en un hipergrafo $H = (X \cup Y \cup Z, E)$, donde $|X| = |Y| = |Z| = n$ y $E \subseteq X \times Y \times Z$: es decir, cada hiperarista toca a cada conjunto exactamente una vez. El problema consiste en determinar la existencia de algún subconjunto $A \subseteq E$ de n hiperaristas tal que todos los nodos de X, Y y Z estén cubiertos exactamente una vez.

Sea entonces una instancia tal. Consideremos una instancia (G, w) de 3-CSORS construida como sigue:

- Hay $3n$ clientes, que identificaremos con los nodos de H .
- Hay $m = |E|$ centros, que identificaremos con las hiperaristas de E . Cada centro $i_{xyz} = \{x, y, z\} \in \mathcal{I}$ está conectado a los nodos x, y y $z \in \mathcal{J}$ mediante aristas de peso $w_{ix} = w_{iy} = w_{iz} = 1$.
- Llamamos (G, w) a la completación métrica del grafo anterior: es decir, entre un cliente x y un centro i no conectado a él, agregamos un arco cuyo peso equivale al mínimo largo de un camino de aristas que los conecten. (Definimos el peso como infinito si no existe tal camino).

Es fácil ver que si existe un 3-d matching A , entonces existe una solución para esta instancia de costo igual a n : basta tomar cada hiperarista $\{x, y, z\}$ de A , abrir el centro correspondiente $i_{xyz} \in \mathcal{I}$, y asignarle los clientes x, y y z . Como A cubre a todos los nodos de H , todos los clientes de G son atendidos; el peso de cada centro escogido es 1, y hay n centros abiertos, por lo que el costo de A como asignación factible para 3-CSORS es n .

Como esta solución es factible para capacidades tipo hard, también lo es para capacidades tipo soft.

Por otro lado, sabemos que no puede haber una solución de costo menor a n : esto es porque por el principio del palomar, se necesitan a lo menos n centros para atender a los $3n$ clientes, y el menor costo que puede tener un centro abierto es 1 (ya que no hay aristas que pesen menos).

Más aún, en cualquier solución de costo n , cada centro i_{xyz} debe pagar exactamente 1, y por lo tanto, sus clientes tienen que ser x, y y z : Así, podemos tomar

$$A = \{\{x, y, z\} : i_{xyz} \text{ está abierto}\} \subseteq E$$

y vemos que debe cubrir a todos los nodos (puesto que todos los clientes de G son atendidos), y tiene exactamente n hiperaristas. Observamos que A está bien definido porque la capacidad $p = 3$ fuerza a los centros a atender a no más de 3 clientes, y el costo n obliga a que esa capacidad se complete. Esto concluye la reducción para la variante hard-capacitada.

Para la variante soft-capacitada, notamos que ningún centro del óptimo está abierto dos veces: ya que solo hay tres clientes a distancia 1 de él. Luego, los óptimos para soft-capacitado lo son también para hard-capacitado, y en consecuencia A sigue bien definido. \square

Podemos hacer una reducción similar para cualquier $p > 3$, utilizando k -dimensional Matching (con $k = p$). Los casos $p \in \{1, 2\}$ son polinomiales, como veremos en el capítulo 2.

Observación La demostración de la Proposición 1.2 muestra que 3-CSORS es NP-difícil incluso en las instancias en las que w corresponde a la distancia geodésica entre nodos de G .

1.4. Interpretación de las figuras y nomenclatura

Al representar las instancias gráficamente, es conveniente que el grafo bipartito completo de la instancia quede implícito. La representación, pues, será un grafo desde el que se puede construir un bipartito completo haciendo la completación métrica, y luego borrando las aristas que no sean entre un cliente y un centro.

Siempre representaremos los centros como *cuadrados* y los clientes como *círculos*. Representaremos por medio de colores a la función de atención σ , es decir, el color de un centro i y de un cliente j con $\sigma(j) = i$ será el mismo, como en el ejemplo de la Figura 1.1.

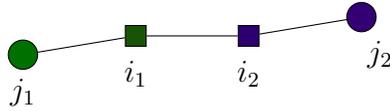


Figura 1.1: En esta solución, los centros i_1 e i_2 están atendiendo a los clientes j_1 y j_2 respectivamente.

1.4.1. Nuestros resultados

El resto de este trabajo se divide en dos capítulos:

El Capítulo 2 establece algunos resultados generales sobre las tres variantes: entre ellos los más destacados son la solución de soft 2-CSORS (Sección 3.3.1) y la reducción de la variante makespan a la soft-capacitada (Sección 2.2.1).

El Capítulo 3 se concentra en el caso donde podemos interpretar (G, w) como un camino de centros y clientes: esto hace que la variante soft-capacitada sea pseudopolinomial (Teorema 3.10). También construimos una 2-aproximación para la variante hard-capacitada (Teorema 3.15).

Capítulo 2

Algunos resultados generales

En este capítulo exponemos algunos resultados sobre la versión más general de las variantes definidas en la sección anterior. Algunos casos particulares se pueden resolver en tiempo polinomial; también comparamos la complejidad de aproximar las distintas variantes.

2.1. Casos polinomiales

Para algunos casos particulares, podemos resolver los problemas en tiempo polinomial. Esto complementa los resultados del capítulo anterior, en los que comprobamos condiciones suficientes para que las tres versiones sean NP-difíciles.

2.1.1. Un solo servidor

Cuando todos los centros tienen un solo servidor, i.e. para 1-CSORS, el costo de cada uno es simplemente la suma de todos los tiempos de atención de sus clientes: la función objetivo se convierte entonces en

$$C(\sigma) = \sum_{i \in I} \sum_{j \in S_i} w_{ij} = \sum_{j \in \mathcal{J}} w_{\sigma(j)j}.$$

este costo es idéntico al problema Facility Location (FL), salvo que en FL los centros tienen costos de apertura. En consecuencia es fácil verificar que

Proposición 2.1 *Para 1-CSORS, tanto soft-capacitado como makespan, el óptimo consiste en asignar a cada cliente su centro más cercano.*

DEMOSTRACIÓN. Denotemos, para cada cliente j , a su centro más cercano como $\sigma^*(j)$ (de haber empates, escogemos arbitrariamente). Consideremos una asignación factible cualquiera

σ . Entonces, su costo total es

$$\begin{aligned} C(\sigma) &= \sum_{j \in \mathcal{J}} w_{\sigma(j)j} \\ &\geq \sum_{j \in \mathcal{J}} w_{\sigma^*(j)j} \\ &= C(\sigma^*), \end{aligned}$$

que es el costo total de σ^* . Por lo tanto, σ^* es asignación óptima. \square

Observación Si no existen empates en la construcción de σ^* , el óptimo es único; si existen, las diferentes elecciones posibles dan luz a todas las asignaciones óptimas.

Hard 1-CSORS, por otro lado, se puede ver como un caso de Matching máximo de costo mínimo. Consideremos el grafo bipartito completo $K_{n,m}$, donde las partes corresponden con los n clientes y m centros respectivamente. Definimos como w_{ij} el costo de la arista $\{j, i\}$ para cada $j \in \mathcal{J}$ e $i \in \mathcal{I}$. Puesto que todo centro solo puede atender a lo más a un cliente, y todo cliente debe ser atendido por exactamente un centro, buscamos un matching que empareje los n clientes con centros distintos, pagando el menor costo total posible. Concluimos que

Proposición 2.2 *1-CSORS, en cualquiera de las tres variantes, es polinomial.*

2.1.2. Dos servidores, versión soft-capacitada

Sea (G, w) una instancia de soft 2-CSORS. Supongamos que en un óptimo dos clientes j, k son atendidos ‘juntos’, es decir, que $\sigma(j) = \sigma(k) = i$. Notamos que este centro debe cumplir con

$$i \in \operatorname{argmín}_{\ell \in \mathcal{I}} \operatorname{máx}(w_{\ell j}, w_{\ell k}),$$

es decir, debe minimizar el costo de atender a j y k desde el mismo centro (de no ser así, podríamos reemplazar i con un centro i' que cumpla esto, y disminuir el costo). De aquí en adelante asumiremos sin pérdida de generalidad que este centro minimizante es único.

Esto nos permite reducir soft 2-CSORS a Matching perfecto de peso mínimo. Consideremos un grafo G' con pesos en las aristas dados por w' , construido de la siguiente manera:

- Los nodos del grafo son los clientes \mathcal{J} , más otros $n = |\mathcal{J}|$ nodos auxiliares.
- Entre cada par de clientes distintos j, k hay una arista de peso

$$w'_{jk} = \operatorname{mín}_{i \in \mathcal{I}} \operatorname{máx}(w_{ij}, w_{ik});$$

entre cada par de nodos auxiliares distintos hay una arista de peso 0.

- Cada cliente está conectado a exactamente un nodo auxiliar, en una correspondencia 1 a 1. La arista entre un cliente j y su correspondiente nodo auxiliar u_j es

$$w'_{ju_j} = \min_{i \in \mathcal{I}} w_{ij}.$$

Podemos observar un ejemplo de esta construcción en la Figura 2.1

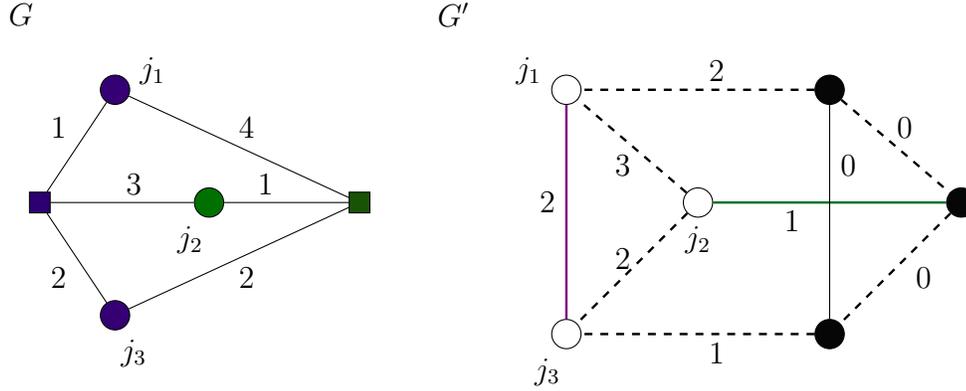


Figura 2.1: Ejemplo de la reducción de 2-CSORS a Matching perfecto de peso mínimo. El grafo de la izquierda, G , es una instancia de 2-CSORS. A la derecha está su correspondiente instancia G' de Matching. Ambos tienen destacado un óptimo, de costo 3.

Notamos de la figura que los óptimos de cada instancia tienen el mismo costo para sus problemas respectivos. Podemos comprobar que esto siempre sucede:

Proposición 2.3 *Las soluciones óptimas para soft 2-CSORS corresponden a matchings perfectos de peso mínimo en G' , y el costo de ambos es el mismo.*

DEMOSTRACIÓN. Consideremos una asignación óptima σ de clientes a centros. Podemos construir un matching perfecto M a partir de σ como sigue:

- Para cada par de clientes atendidos juntos j, k , agregamos las aristas $\{j, k\}$ y $\{u_j, u_k\}$ al matching. Notamos que el costo que aportan los clientes a $C(\sigma)$ y a M es el mismo, por lo tratado al comienzo de esta subsección. Recordamos que las aristas entre nodos auxiliares tienen costo 0.
- Para cada cliente j atendido ‘solo’, agregar la arista $\{j, u_j\}$ a M . Al ser atendido solo, el centro más conveniente es el más cercano, por lo que nuevamente el costo que aporta j al matching y al $C(\sigma)$ es el mismo.

Como todos los clientes son atendidos, todos los nodos de G' son cubiertos por alguna arista de M ; y por construcción el costo de M es $C(\sigma)$.

Por el otro lado, sea M un matching perfecto de peso mínimo en G' . Construimos una solución factible para soft 2-CSORS como sigue:

- Para cada arista de M' entre dos clientes j, k , definimos $\sigma(j) = \sigma(k) = i$, donde i es un centro que cumpla con

$$\text{máx}(w_{ij}, w_{ik}) = w'_{jk}.$$

Sabemos que existe, por la definición de w' .

- Para cada arista entre un cliente j y su nodo auxiliar u_j , atendemos j desde el centro más cercano.

Es directo que el costo de σ es el mismo que el de M (puesto que las aristas entre nodos auxiliares pesan 0), y que todos los clientes son atendidos, ya que todos eran alcanzados por el matching. Esto concluye la demostración. \square

Observación Si se desea extender este resultado al caso hard-capacitado, no es posible asegurar que un par de clientes sea atendido por el centro que minimiza costos: ver, por ejemplo, la instancia de la Figura 2.2. La construcción del análogo a G' requiere entonces múltiples aristas entre cada par de nodos, una por cada centro; el problema al cual nos reducimos es entonces una instancia particular de matching perfecto arcoíris de peso mínimo, problema que requiere un análisis más profundo, pues, en general, dicho problema es NP-difícil.

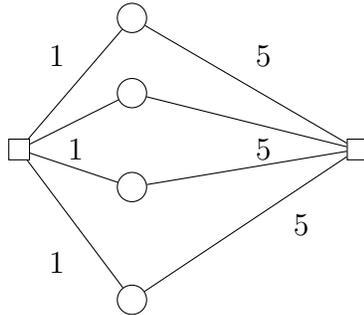


Figura 2.2: En esta instancia de hard 2-CSORS, debe haber un par de clientes atendidos desde un centro que no es el más cercano.

2.2. Reducciones entre las variantes

2.2.1. Makespan versus soft-capacitada

Del capítulo de Preliminares (en particular, las Proposiciones 1.1 y 1.2), vemos que la versión makespan parece ser sustancialmente más difícil que las otras. Sin embargo, cuando la capacidad de los centros aumenta, la diferencia eventualmente desaparece: esto es porque si cada centro puede atender a todos sus clientes simultáneamente, las tres variantes alcanzan el mismo valor objetivo.

No es de extrañar entonces que podamos aproximar al óptimo de makespan con uno soft-capacitado, y el factor de aproximación disminuya cuando la capacidad de los centros aumenta.

Teorema 2.4 *Para todo $p \geq 2$, toda solución factible de Makespan p -CSORS de costo α se puede transformar en una solución factible de soft p -CSORS de costo a lo más $\frac{p+1}{p}\alpha$. Toda solución factible de soft p -CSORS se puede transformar en una solución factible de Makespan p -CSORS con a lo más el mismo costo.*

DEMOSTRACIÓN. La idea de la demostración es la siguiente: dada una solución factible para makespan, ¿cuál es la mejor asignación soft-capacitada en la que cada centro mantiene el conjunto de clientes que atiende? Será útil aquí la interpretación de atención en rondas para la variante soft-capacitada.

Notemos en primer lugar que de toda solución factible para makespan podemos construir una para el caso soft-capacitado: para cada centro i , tomamos el número máximo t de atenciones que realice un servidor y separamos la atención en t rondas, manteniendo los clientes asignados a cada servidor y el orden en que son atendidos. Esto puede introducir ‘tiempos muertos’ en los que los servidores esperen el fin de cada ronda: esto hace que el costo soft-capacitado sea mayor.

En consecuencia, podemos hablar de costo soft-capacitado para una solución makespan. Denotamos por C al costo de la solución para la versión makespan, y por C^s al costo soft-capacitado. Para cada centro i , sean C_i y C_i^s los aportes que hace a los costos respectivos.

Sabemos entonces que C_i está dado por

$$C_i = \max_{k=1..p} \sum_{j \in S_{ik}} w_{ij},$$

recordando que S_{ik} es el conjunto de clientes atendidos por k desde i .

Para expresar C_i^s , supongamos que la atención es en ℓ rondas, y denotemos por $j(r, k)$ al cliente atendido por el servidor k en la ronda r . Entonces,

$$C_i^s = \sum_{r=1}^{\ell} \max_{k=1..p} w_{ij(r,k)}.$$

Podemos visualizar ambos costos como funciones de una matriz $A^{(i)}$ de ℓ filas y p columnas, donde las entradas están definidas por

$$A_{rk}^{(i)} = w_{ij_k(r)}$$

(que consideramos 0 si el servidor no atiende a ningún cliente en esa ronda). C_i^s es entonces la suma de los máximos de cada fila, mientras que C_i con esa misma asignación corresponde

a la máxima suma de alguna columna. Claramente, esta segunda función es menor que la primera, sin importar cómo distribuimos los costos.

Antes de continuar, podemos modificar un poco el proceso de convertir la solución en factible soft-capacitada. Consideremos el procedimiento que en cada centro abierto i , hace lo siguiente:

1. Definir ℓ , el número de rondas, como el menor entero tal que $|S_i| \leq p\ell$.
2. Ordenar de manera decreciente los costos de atención a los clientes de S_i como $w_1 \geq w_2 \geq \dots \geq w_{p\ell}$, donde si $|S_i| < p\ell$ introducimos costos auxiliares iguales a 0.
3. Completar la matriz $A^{(i)}$ con $\{w_i | i \in [p\ell]\}$ por filas, como sigue:

$$\ell \text{ rondas} \left\{ \begin{array}{c} \overbrace{\hspace{10em}}^{p \text{ servidores}} \\ \left[\begin{array}{cccc} w_1 & w_2 & \cdots & w_p \\ w_{p+1} & w_{p+2} & \cdots & w_{2p} \\ \vdots & & & \vdots \\ w_{p(\ell-1)+1} & \cdots & & w_{p\ell} \end{array} \right] \end{array} \right.$$

4. Definir cada ronda según las filas de $A^{(i)}$, atendiendo en la ronda r con el servidor k al cliente cuyo costo esté dado por $A_{rk}^{(i)}$ (es decir el valor $w_{p(r-1)+k}$). A los servidores asociados a costos auxiliares, no asignarles ningún cliente, como en el ejemplo de la Figura 2.3



Figura 2.3: Ejemplo de la atención definida en el paso 4. La matriz $A^{(i)}$ de la izquierda se traduce en la atención representada en el grafo de la derecha: las aristas de color negro, verde y morado representan al primer, segundo y tercer servidor respectivamente.

Veamos que esta matriz $A^{(i)}$ representa una asignación soft-capacitada óptima de servidores a clientes en S_i . Esto se desprende del siguiente hecho: en toda asignación óptima, los p costos mayores w_1, \dots, w_p deben pertenecer a la misma ronda.

Esto se puede comprobar por contradicción: supongamos que w_1 está en la ronda r , y sea w_j , $2 \leq j \leq p$ el primero de los p mayores costos que está en una ronda distinta r' (el caso $p = 1$, así como $\ell = 1$, son directos). El costo que aportan r y r' a la función objetivo es $w_1 + w_j$: ya que w_1 es por definición el máximo valor de la ronda en la que está, y w_j es el

mayor valor en cualquiera de las otras, incluyendo r' . Además, en r debe haber un costo $w_{j'}$ con $j' > p$.

Pero entonces podemos intercambiar de lugar a w_j con $w_{j'}$: el costo de la ronda r sigue siendo w_1 , y el costo de r' es ahora el máximo entre $w_{j'}$ y sus otros costos, todos menores o iguales a w_j . Podemos continuar este proceso hasta que en la ronda r estén todos los costos w_1, \dots, w_p .

Dado esto, podemos aplicar el mismo razonamiento para mostrar que w_{p+1}, \dots, w_{2p} deben conformar también una sola ronda. Por inducción, las rondas de una asignación óptima soft-capacitada corresponden justamente con las filas de $A^{(i)}$ (no necesariamente en el mismo orden). Como el orden de las rondas no altera el costo soft-capacitado, $A^{(i)}$ representa también una asignación óptima.

Denotemos por $\zeta_1, \zeta_2, \dots, \zeta_p$ a las sumas por columnas de $A^{(i)}$. Así, $\zeta_1 = C_i^s$, puesto que la primera columna contiene a los máximos de cada fila.

Solo queda entonces demostrar que $\zeta_1 \leq \frac{p+1}{p}C_i$. Con ello, sumando por centros tenemos que $C_i^s \leq \frac{p+1}{p}C$ y concluimos.

Procedemos por contradicción. Utilizaremos los pesos $w_1, \dots, w_{p\ell}$ tal como en la construcción de la matriz $A^{(i)}$. Llamaremos además ζ a la suma de todos los pesos de los clientes atendidos.

Supongamos entonces que $\zeta_1 > \frac{p+1}{p}C_i$. Sabemos además, por definición, que

$$C_i \geq \frac{\zeta}{p}$$

y por lo tanto,

$$\zeta_1 > \frac{p+1}{p^2}\zeta. \quad (2.1)$$

Pero $\zeta = \zeta_1 + (\zeta - \zeta_1)$. Ahora, por la manera en que fue construida $A^{(i)}$, sabemos que para todo $k > 1$, $\zeta_k \geq \zeta_1 - w_1$: esto viene de que la entrada j -ésima de la columna k es mayor que la entrada $j+1$ -ésima de la columna 1.

Sumando estas desigualdades obtenemos que $\zeta - \zeta_1 \geq (p-1)(\zeta_1 - w_1)$, y por lo tanto,

$$\zeta_1 > \frac{p+1}{p^2}(\zeta_1 + (p-1)(\zeta_1 - w_1))$$

Reordenando los términos, obtenemos que

$$w_1 > \frac{p}{p^2-1}\zeta_1,$$

y combinando esta desigualdad con la inecuación (2.1) tenemos que

$$p(p-1)w_1 > \zeta. \quad (2.2)$$

Ahora, como $\zeta \geq C_i$, esto implica que $p(p-1)w_1 > C_i$: en consecuencia,

$$\frac{\zeta_1}{C_i} \leq \frac{\zeta_1}{p(p-1)w_1} \quad (2.3)$$

Pero $\zeta_1 = w_1 + \zeta_1 - w_1$. Por otro lado, ya habíamos visto que $\zeta_k \geq \zeta_1 - w_1$ para todo $k \in \{2, \dots, p\}$, luego

$$\zeta - w_1 = \zeta_1 - w_1 + \zeta_2 + \dots + \zeta_p \geq p(\zeta_1 - w_1),$$

y la inecuación 2.2 implica que $\zeta - w_1 < (p(p-1) - 1)w_1$; juntando ambas inecuaciones, obtenemos

$$\zeta_1 - w_1 < \frac{(p(p-1) - 1)}{p} w_1.$$

Incorporando esto a la desigualdad 2.3, y desarrollando,

$$\begin{aligned} \frac{\zeta_1}{C_i} &\leq \frac{w_1 + \zeta_1 - w_1}{p(p-1)w_1} \\ &< \frac{1}{p(p-1)} \left(1 + \frac{(p(p-1) - 1)}{p}\right) \\ &= \frac{p+1}{p^2}, \end{aligned}$$

que es menor a 1 para todo $p > 1$ entero. Pero eso contradice que $\zeta_1 = C_i^s \geq C_i$. Esto concluye la demostración. \square

2.2.2. Soft versus hard-capacitada

La demostración de la NP-dificultad de las variantes soft y hard-capacitada (propiedad 1.2) sugiere que éstas tienen una relación estrecha entre sí. Incluimos en esta sección una reducción sencilla de la primera a la segunda.

Proposición 2.5 *Soft p -CSORS se reduce a hard p -CSORS, preservando el factor de aproximación.*

DEMOSTRACIÓN. Sea $(\mathcal{I}, \mathcal{J}, w)$ una instancia de soft p -CSORS, y sea σ^* una asignación óptima. Denotemos por a^* al número de aperturas máximo de algún centro. Consideremos entonces la siguiente instancia $(\mathcal{I}', \mathcal{J}', w')$ de hard p -CSORS:

- $\mathcal{J}' = \mathcal{J}$, es decir el conjunto de clientes es el mismo.
- Para cada centro $i \in \mathcal{I}$ hay a^* copias de él en \mathcal{I}' , que denotamos por i_1, \dots, i_{a^*} .
- Se preservan todas las distancias: es decir, para todo $\ell \in [a^*]$, todo centro $i \in \mathcal{I}$ y cliente $j \in \mathcal{J}$,

$$w'_{i_\ell j} = w_{ij},$$

y además para dos copias i_1, i_2 cualquiera, $w'_{i_1 i_2} = 0$

Llamemos G' al grafo así formado. Intuitivamente, solo multiplicamos los centros de forma que cada apertura necesaria para σ^* equivalga a la apertura de centros diferentes de \mathcal{I}' , sin cambiar el costo de la asignación. Es sencillo verificar que esto es lo que sucede:

Para una asignación factible cualquiera $\sigma : \mathcal{J} \rightarrow \mathcal{I}$, podemos construir su asignación correspondiente $\sigma' : \mathcal{J} \rightarrow \mathcal{I}'$. Si un centro i es abierto a veces, basta asignar cada ronda de clientes a una copia distinta de i en \mathcal{I} .

Conversamente, para una asignación σ' factible hard-capacitada en G' , definimos σ asignación en G mediante la siguiente regla: en cada cliente j tal que $\sigma'(j)$ sea una copia de i , hacemos $\sigma(j) = i$. Es trivial que el costo de ambas asignaciones es el mismo.

En consecuencia, de tener un algoritmo de aproximación para el caso hard-capacitado podemos obtener la misma aproximación para la versión soft-capacitada utilizando esta transformación.

Finalmente, sabemos que a^* es a lo más $\left\lceil \frac{n}{p} \right\rceil$. Así, la reducción se puede hacer en tiempo polinomial. \square

Capítulo 3

Caso lineal

Este capítulo se concentra en la restricción del problema a instancias donde el grafo de clientes y centros es un camino: podemos entonces ubicarlos en la recta real, de forma que la distancia entre dos puntos $a, b \in \mathcal{I} \cup \mathcal{J}$ es simplemente $w_{ab} = |b - a|$.

Sabemos que la versión makespan mantiene su dificultad, como vimos en la Proposición 1.1: este capítulo se concentra entonces en las versiones soft-capacitadas y hard-capacitadas. Los resultados principales son el Algoritmo 3, un algoritmo exacto a tiempo polinomial para capacidades uniformes tipo soft; su extensión a capacidades arbitrarias, en la Sección 3.3.3, y una 2-aproximación para el caso uniforme de las capacidades tipo hard (Teorema 3.15).

Para conseguir lo primero, empezaremos estudiando la relación entre nuestro problema y otro ya conocido, *Weighted Interval Point Cover* (WIPC). Este problema reduce a soft p -CSORS cuando p es suficientemente grande para considerarse infinito (es decir, cada centro puede atender a cualquier cantidad de clientes simultáneamente). Llamaremos a esta variante ∞ -CSORS, y extenderemos una solución de ésta hasta obtener una para soft CSORS. Luego, una función objetivo alternativa nos permitirá aproximar el caso hard-capacitado.

3.1. Definición de WIPC y notación específica

El problema de *Weighted Interval Point Cover* consiste en lo siguiente: dado un conjunto de n puntos o clientes en la recta real x_1, \dots, x_n y una familia de m' intervalos $\Gamma = \{J_1, \dots, J_{m'}\}$ con pesos no negativos $\omega_\ell = \omega(J_\ell)$, ¿cuál es la subfamilia de intervalos de menor peso total que cubra a todos los puntos? Este problema es un ejercicio conocido de programación dinámica [23]. El uso que esta tesis le da considera solo el caso particular en el que los bordes de los intervalos J_ℓ son siempre clientes: esto es, que para todo $J \in \Gamma$, $J = [x_l, x_r]$ con $l, r \in [n]$ (dependientes de J).

Podemos considerar la restricción adicional de que cada intervalo J tiene una *capacidad* p_J , de forma de escoger a lo más p_J puntos de entre los que cubre: necesitamos en este caso que todo punto sea escogido por algún intervalo que lo cubra. Anotaremos como $S(J)$ al conjunto

de clientes escogidos por J .

En general usaremos la letra J para referirnos a un intervalo. Llamaremos *intervalos activos* a los intervalos escogidos por una solución. También hacemos el siguiente abuso de notación: en vez de escribir x_j , escribiremos simplemente j : así, el intervalo $[x_l, x_r]$ lo escribiremos simplemente como $[l, r]$. Otra consecuencia es que $j + 1$ denotará al cliente inmediatamente a la derecha de j (sin importar la distancia a la que estén).

Es fácil ver que este problema reduce a soft CSORS, mediante la siguiente interpretación:

- Cada cliente $j \in \mathcal{J}$ es un punto x_j en la recta.
- Cada par de clientes l, r , y centro i , define un intervalo $[l, r]_i$ de costo

$$C_i([l, r]) = \text{máx}(w_{il}, w_{ir}).$$

- Activar el intervalo $J = [l, r]_i$ corresponde a abrir el centro i con un radio de atención de $C_i([l, r])$.
- Escoger un subconjunto de puntos $S(J) \subseteq J$ corresponde a atender a los clientes de $S(J)$ desde i : esto es, para todo $j \in S(J)$, $\sigma(j) = i$.

Sin pérdida de generalidad asumiremos siempre que $S(J)$ contiene a los extremos de J , puesto que de no ser así, podemos reemplazar J por un subintervalo que cumpla esta condición.

Soft CSORS corresponde así a un caso particular de WIPC capacitado, en el que Γ es la unión de familias $\mathcal{Y}_i = \{[l, r]_i : l, r, \in \mathcal{J}\}$, donde i recorre los centros. Hard CSORS corresponde al mismo problema, con la restricción adicional de que a lo más un intervalo de cada familia debe estar activo.

3.2. Solución para el caso $p = \infty$

Es fácil ver que ∞ -CSORS, en que cada centro puede atender a todos sus clientes simultáneamente, es el punto de encuentro de las tres variantes del problema (makespan, soft-capacitado y hard-capacitado): ya que cada servidor solo necesita trabajar una vez. Como vimos en la sección anterior, en la línea este es un caso particular de WIPC.

WIPC admite algoritmos polinomiales, como programación dinámica [29]. Detallamos, por su posterior utilidad, una solución de este tipo.

Nuestra función de costos para los intervalos es muy particular: su valor solo depende de los puntos extremos del intervalo, y de su centro i asociado. Esto permite verificar las siguientes propiedades:

Lema 3.1 *En cualquier óptimo de ∞ -CSORS o de soft p -CSORS, el centro que atiende a un intervalo activo $[l, r]$ es el que está más cerca del punto medio entre l y r .*

DEMOSTRACIÓN. Consideremos la función real definida por $\delta(x) = \max(w_{xl}, w_{xr})$. Sea $a = \frac{l+r}{2}$ el punto medio entre l y r . Notamos que si x está a la izquierda de a , el costo cumple con

$$\delta(x) = w_{xr} = w_{xa} + w_{ar};$$

análogamente, cuando x está a la derecha de a , $\delta(x) = w_{xa} + w_{al}$.

Pero $w_{ar} = w_{al} = \frac{r-l}{2}$: por lo tanto,

$$\delta(x) = \frac{r-l}{2} + w_{ax} = \frac{r-l}{2} + |a-x|.$$

Claramente δ es convexa y simétrica, con su mínimo en $x = a$: así, restringiendo el dominio a las posiciones de los centros, el mínimo será en el centro más cercano a a . Como en ∞ -CSORS y soft p -CSORS este centro siempre está disponible, podemos ocuparlo para atender a $[l, r]$. \square

Con esto, los intervalos $[l, r]_i$ son redundantes, ya que solo hay un centro i^* que atiende a $[l, r]$ de la mejor forma: podemos hablar entonces de *el* intervalo $[l, r]$, cuyo costo será $C[l, r] = C_{i^*}[l, r]$.

Lema 3.2 *Sea i un centro. Entonces $C_i(\cdot)$ es creciente para la inclusión: es decir, dados J_1, J_2 intervalos de clientes con $J_1 \subseteq J_2$, tenemos que $C_i(J_1) \leq C_i(J_2)$.*

DEMOSTRACIÓN. Sean $J_1 = [l_1, r_1], J_2 = [l_2, r_2]$ intervalos tales que $J_1 \subseteq J_2$. Supongamos que i está a la izquierda del punto medio de J_1 : esto es, $C_i(J_1) = w_{ir_1}$. Como $J_1 \subseteq J_2$, tenemos que $r_1 \leq r_2$ y luego

$$C_i(J_1) = w_{ir_1} \leq w_{ir_1} + w_{r_1 r_2} = w_{ir_2} \leq C_i(J_2).$$

El caso en que i está a la derecha del punto medio de J_1 es análogo. \square

Observación Este lema es una propiedad de la función de costos C_i en la línea, no de esta variante en particular: es válida también para la variante soft-capacitada, puesto que el costo de cada apertura es independiente de las otras.

Proposición 3.3 *En un óptimo de ∞ -CSORS podemos asumir que los intervalos activos son disjuntos.*

DEMOSTRACIÓN. Consideremos dos intervalos $J_1 = [l_1, r_1], J_2 = [l_2, r_2]$ no disjuntos, atendidos por i_1 e i_2 respectivamente. Hay entonces, salvo reetiquetados, dos casos: o bien $J_1 \subseteq J_2$, o bien se “atravesan” y $l_1 < l_2 < r_1 < r_2$. Ambos casos se ilustran en la Figura 3.1

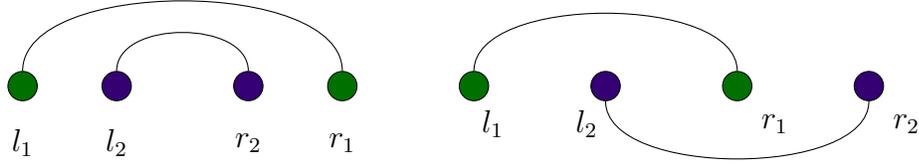


Figura 3.1: Las dos maneras (salvo reetiquetados) en que los intervalos pueden no ser disjuntos.

En el primer caso, la apertura de J_1 es redundante, pues sus clientes pueden ser atendidos por J_2 ; en el segundo, podemos asignar a los clientes de $[l_2, r_1]$ a J_1 . Definiendo entonces $J'_2 = [r_1 + 1, r_2]$, tenemos una solución factible de costo

$$C(J_1) + C(J'_2) \leq C(J_1) + C_{i_2}(J'_2) \leq C(J_1) + C(J_2),$$

donde para la última desigualdad hemos usado que $J'_2 \subseteq J_2$ y el Lema 3.2.

Podemos hacer esto iterativamente en cada par de intervalos no disjuntos: esto llevará a una solución de costo no menor. Podemos asumir entonces que en un óptimo ya se ha hecho este procedimiento. \square

El problema entonces cumple la propiedad de subestructura óptima: en particular, si en un óptimo identificamos a $[l^*, n]$, el intervalo activo de más a la derecha, sabemos que el óptimo atenderá de manera separada a los clientes anteriores a l^* .

Denotemos por $\text{OPT}[j]$ al costo de la forma óptima de atender a los primeros j clientes. Tenemos entonces la siguiente relación recursiva:

$$\text{OPT}[j] = \min_{\ell=1, \dots, j} (C[\ell, j] + \text{OPT}[\ell - 1]), \quad (3.1)$$

adoptando la convención de que $\text{OPT}[0] = 0$.

La tabla de costos $(C([l, r]))_{l \leq r \in \mathcal{J}}$ la podemos calcular en tiempo polinomial: para este análisis, lo haremos en tiempo $O(n^2 \log(m))$, buscando mediante búsqueda binaria el centro más cercano al punto medio de cada par l, r .

Así, podemos encontrar el óptimo con un algoritmo de programación dinámica como el siguiente:

Algoritmo 1

Entrada: Tabla $(C[i, j])_{i, j \in \mathcal{J}}$ con los costos de cada intervalo; n el número de clientes.

- 1: Inicializar tabla OPT que guardará los costos de los subproblemas.
 - 2: $OPT[1] = C([1, 1])$
 - 3: **para** $j = 2, \dots, n$ **hacer**
 - 4: **para** $\ell = 1, \dots, j$ **hacer**
 - 5: Computar $C([\ell, j]) + OPT[\ell - 1]$ con los valores de ambas tablas. Almacenar ℓ^* minimizante.
 - 6: $OPT[j] \leftarrow C([\ell^*, j]) + OPT[\ell^* - 1]$
 - 7: Entregar $OPT[n]$
-

Concluimos que

Proposición 3.4 *Existe un algoritmo para el caso $p = \infty$ que es polinomial. Más aún, el problema se puede resolver en tiempo $O(n^2 + n^2 \log m)$, donde n es el número de clientes y m el número de centros.*

3.3. Solución de soft CSORS en la línea

Vimos en la primera sección que soft CSORS en la línea es análogo a WIPC con capacidades: cada intervalo activo $[l, r]$ escoge atender a lo más a p clientes pertenecientes a él. Una consecuencia inmediata de esto es que los intervalos de atención óptimos ya no son necesariamente disjuntos, como en la instancia de la Figura 3.2.

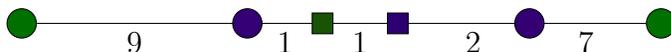


Figura 3.2: En esta instancia, con $p = 2$, la atención óptima consiste en atender desde cada centro a los clientes del mismo color: los correspondientes intervalos de atención no son disjuntos

Sin embargo, el *costo* de un intervalo sigue calculándose de la misma forma que para ∞ -CSORS. Es más, observemos que el Lema 3.1, que identifica al centro que atiende un intervalo activo, es válido para soft p -CSORS (puesto que múltiples intervalos pueden tener asignado el mismo centro). También aplica a este caso el Lema 3.2, que establece que los costos de atención desde un centro son crecientes para la inclusión de intervalos. La única propiedad, entonces, de la Sección 3.2 que no se cumple para soft p -CSORS es la Propiedad 3.3, la separación de los intervalos

Sin embargo, se sigue cumpliendo la propiedad de subestructura óptima, ya que:

Proposición 3.5 *En un óptimo de soft-CSORS en la línea, los intervalos de atención forman una familia laminar: para todo par J_1, J_2 de intervalos activos, se tiene que o bien son disjuntos, o uno está contenido en el otro.*

DEMOSTRACIÓN. Sean $J_1 = [l_1, r_1]$, $J_2 = [l_2, r_2]$ intervalos activos en un óptimo. Sin pérdida de generalidad, podemos asumir que $l_1 < l_2$ (como l_1 es atendido por J_1 , si fueran iguales podríamos cambiar l_2 por $l'_2 = l_2 + 1$). Supongamos que los intervalos intersectan no vacío y que $J_2 \not\subseteq J_1$: luego $l_2 \leq r_1$.

Si $l_2 = r_1$, podemos reemplazar l_2 con el cliente de su derecha, separando los intervalos. En caso contrario, tenemos un caso como el de la Figura 3.3, que podemos resolver inductivamente en el número de clientes entre l_2 y r_1 de la siguiente manera:

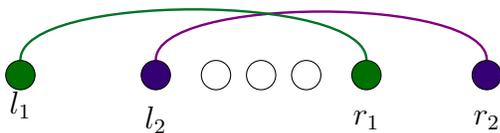


Figura 3.3: Dos intervalos que violan la laminaridad. La inducción es en el número de clientes en la intersección, representados en blanco.

Si no hay clientes entre l_2 y r_1 , podemos definir intervalos de menor costo

$$\begin{aligned} J'_1 &= [l_1, l_2] \\ J'_2 &= [r_1, r_2] \end{aligned}$$

que siguen atendiendo a todos los clientes, como en la Figura 3.4.

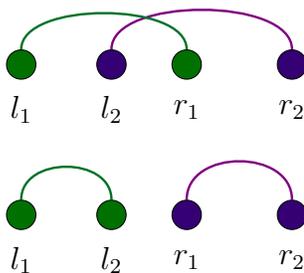


Figura 3.4: Caso base de la inducción en el Lema 3.5

Por otro lado, si hay clientes entre l_2 y r_1 , sea j el cliente atendido por J_2 de más a la derecha. Definimos entonces $J'_1 = [l_1, r_1 - 1]$, $J'_2 = I_2$, con los conjuntos de atención

$$\begin{aligned} S(J'_1) &= S_1 - r_1 + j \\ S(J'_2) &= S_2 - j + r_1. \end{aligned}$$

Estos conjuntos mantienen la capacidad, siguen atendiendo a todos los clientes, y los costos no aumentaron: más aún, el número de clientes en su intersección disminuye al menos en uno, como se ilustra en la Figura 3.5. Notamos que en el caso de que j sea l_2 , los intervalos ya están separados. Esto concluye la demostración.

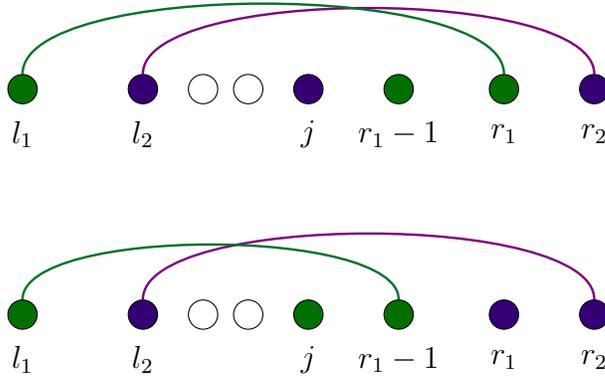


Figura 3.5: Paso inductivo del Lema 3.5

□

Esto nos permite nuevamente establecer una recursión, apuntando a un algoritmo de programación dinámica: extenderemos la notación utilizada anteriormente y escribimos como $\text{OPT}[l, r]$ al óptimo tomando de la instancia solo los clientes en el intervalo $[l, r]$.

3.3.1. Caso uniforme con $p = 2$

El caso con dos servidores es más simple que el caso general, ya que cada intervalo solo atiende a sus extremos: en cada paso de la recursión 3.2, o bien separamos un intervalo $[l, r]$ en dos instancias independientes, o bien activamos dicho intervalo y computamos el óptimo para los clientes de su interior $[l + 1, r - 1]$.

$$\text{OPT}[l, r] = \min \left(\min_{l \leq \ell < r} (\text{OPT}[l, \ell] + \text{OPT}[\ell + 1, r]), \quad (3.2) \right. \\ \left. C([l, r]) + \text{OPT}[l + 1, r - 1] \right)$$

Los casos base son $\text{OPT}[j, j] = C([j, j])$ para todo cliente j . El objetivo es calcular $C([1, n])$: una forma de hacerlo está bosquejada en el Algoritmo 2. Observamos que su ejecución toma tiempo $O(n^3)$ asumiendo que la tabla de costos está construida; se puede concluir entonces lo siguiente:

Proposición 3.6 *El caso lineal con dos servidores se puede resolver en tiempo polinomial. Es más, existe un algoritmo que encuentra un óptimo en tiempo $O(n^3 + n^2 \log m)$.*

Observación Se sabe de la Sección 2.1.2 que cuando hay dos servidores el problema general (soft-capacitado) es polinomial: se incluye este resultado para utilizarlo como escalón a los casos de las siguientes secciones.

Algoritmo 2

Entrada: Tabla $(C([i, j]))_{i, j \in \mathcal{J}}$ con los costos de cada intervalo; n el número de clientes.

- 1: Inicializar tabla OPT que guardará los costos de los subproblemas.
 - 2: Definir, para todo $j \in \mathcal{J}$, $\text{OPT}[j, j] = C([j, j])$, $\text{OPT}[j + 1, j] = 0$
 - 3: **para** $k = 1, 2, \dots, n - 1$ **hacer**
 - 4: **para** $i = 1, \dots, n - k$ **hacer**
 - 5: Calcular $\text{OPT}[i, i + k]$ con la fórmula recursiva. ▷ Esto toma tiempo $O(n)$.
 - 6: Entregar $\text{OPT}[1, n]$
-

3.3.2. Caso uniforme con $p > 2$

Cuando hay más de dos servidores, un intervalo activo debe escoger a qué clientes de su interior atender: esto quiere decir que de haber intervalos anidados $J_1 \supseteq J_2$ no es claro *a priori* cómo éstos se reparten a los clientes al interior de J_2 . Utilizaremos para esto la siguiente propiedad:

Proposición 3.7 *En un óptimo para soft p -CSORS en la línea, con $p > 2$, los intervalos siguen una estructura laminar fuerte: para todo par J_1, J_2 de intervalos activos tales que $J_2 \subset J_1$, se tiene que $S(J_1) \cap J_2 = \emptyset$: esto es, los clientes atendidos por J_1 están fuera de J_2 .*

DEMOSTRACIÓN. Sean $J_1 = [l_1, r_1]$, $J_2 = [l_2, r_2]$ intervalos como en el enunciado, y supongamos que $S(J_1) \cap J_2$ es distinto de vacío. Sea j el cliente de más a la derecha de este conjunto, como en la Figura 3.6.

Sea j' el cliente de $S(J_1) \cup S(J_2)$ inmediatamente posterior a j . Supongamos que es distinto a r_2 . Sabemos entonces que j' debe ser atendido por J_2 (por definición de j). Podemos entonces “intercambiar” j con j' de atención:

$$\begin{aligned} S'(J_1) &= S(J_1) - j + j' \\ S'(J_2) &= S(J_2) - j' + j \end{aligned}$$

En esta operación la solución permanece factible y el costo no aumenta: asumimos entonces que j es el vecino izquierdo de r_2 . Pero en ese caso, el intercambio

$$\begin{aligned} S'(J_1) &= S(J_1) - j + r_2 \\ J'_2 &= [l_2, j] \\ S'(J_2) &= S(J_2) - r_2 + j \end{aligned}$$

nuevamente no aumenta el costo y ahora el número de clientes de $S(J_1) \cap J_2$ disminuyó en uno. Podemos concluir la propiedad por inducción. \square

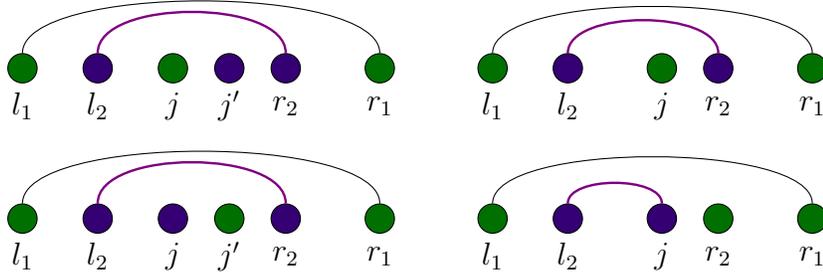


Figura 3.6: Representación gráfica de la demostración de la Proposición 3.7. En la columna de la izquierda, vemos que es posible ‘llevar’ a j a ser vecino de r_2 ; en la columna derecha, que podemos intercambiar estos clientes, disminuyendo el cardinal de $S(J_1) \cup J_2$.

Esto nos posibilita adaptar el Algoritmo 2. Consideramos ahora que un intervalo $[l, r]$ puede tener cierta cantidad de clientes “gratuitos”, atendidos por otro intervalo que lo contenga. Por laminaridad fuerte, estos clientes gratuitos dividen a $[l, r]$ en secciones cuyas atenciones óptimas se pueden calcular de manera independiente unas de otras: nuestra recursión busca la primera de estas secciones.

Denotemos $OPT[l, r, t]$ al costo de la mejor solución en el intervalo $[l, r]$ que atiende a todos los clientes, y en la que hay t clientes *gratuitos*. El objetivo del algoritmo es encontrar $OPT[1, n, 0]$.

Cuando no hay clientes gratuitos, tenemos que

$$OPT[l, r, 0] = \min \left(\begin{aligned} &\min_{\ell=l, \dots, r-1} OPT[l, \ell, 0] + OPT[\ell + 1, r, 0], \\ &C([l, r]) + \min_{\ell=l+1, \dots, r-1} OPT[l + 1, \ell - 1, 0] + OPT[\ell + 1, r - 1, p - 3] \end{aligned} \right), \quad (3.3)$$

donde el primer término corresponde a la elección de separar la atención de $[l, r]$ en dos secciones separadas $[l, \ell]$, $[\ell + 1, r]$, y el segundo corresponde a atender simultáneamente a l y r , y encontrar el cliente gratuito de más a la izquierda (para el caso de que este sea $l + 1$ sea gratuito podemos definir $OPT[l + 1, l, 0] := 0$).

En el caso de haber t clientes gratuitos en el intervalo $[l, r]$, buscamos también el de más a la izquierda:

$$OPT[l, r, t] = \min_{\ell=l, \dots, r} OPT[l, \ell - 1, 0] + OPT[\ell + 1, r, t - 1], \quad (3.4)$$

donde convenimos en que $OPT[j, j - 1, t] := 0$ para todo j cliente y $t \in \{0, \dots, p - 3\}$.

La correctitud de esta recursión depende de un hecho más: que el óptimo efectivamente atienda los clientes gratuitos desde el intervalo exterior siempre que sea posible. Esta propiedad se demuestra en el Lema 3.9.

Los casos base de las recursiones son

$$\begin{aligned} OPT[j, j, 0] &= C([j, j]) & \forall j \in \mathcal{J} \\ OPT[j, j, t] &= 0 & \forall j \in \mathcal{J}, \forall t \in [p-3] \\ OPT[j+1, j, t] &= 0 & \forall j \in \mathcal{J}, \forall t \in [p-3]. \end{aligned}$$

El algoritmo de programación dinámica asociado es de orden $O(n^3p)$, como se puede apreciar en el Algoritmo 3; notar que el paso 6 requiere $O(n)$ lecturas de la tabla dada la recurrencia descrita arriba. Juntando esto con los resultados vistos en 3.2 y 3.3.1, podemos concluir el siguiente resultado.

Teorema 3.8 *En la línea, el problema soft p -CSORS es polinomial, y el óptimo se puede computar en tiempo $O(n^3p + n^2 \log m)$, donde n es el número de clientes, m el número de centros, y p el número de servidores.*

Algoritmo 3

Entrada: Tabla $(C([i, j]))_{i, j \in \mathcal{J}}$ con los costos de cada intervalo; n el número de clientes.

- 1: Inicializar tabla OPT que guardará los costos de los subproblemas.
 - 2: $OPT[i, i, 0] \leftarrow C([i, i]); OPT[i, i, t] \leftarrow 0, \quad \forall i \in \mathcal{J}, t \in [p-3]$.
 - 3: **para** $\ell = 1, 2, \dots, n-1$ **hacer**
 - 4: **para** $i = 1, \dots, n-\ell$ **hacer**
 - 5: **para** $t = 0, \dots, p-3$ **hacer**
 - 6: Calcular $OPT[i, i+\ell, t]$ con las fórmulas 3.3 y 3.4 según corresponda.
 - 7: Entregar $OPT[1, n, 0]$
-

Lema 3.9 *Podemos asumir que un óptimo utilizará todos los clientes gratuitos que pueda: es decir, que todo intervalo J activo cumple que*

$$|S(J)| = \min(|J|, p).$$

DEMOSTRACIÓN. Sea $J = [l, r]$ un intervalo activo en un óptimo. Existen dos casos:

- Si $|J| \leq p$, el hacer $S(J) = J$ es factible y su costo es $C[l, r]$. Abrir cualquier otro intervalo al interior de J tiene un costo adicional positivo, y no sucede en un óptimo.
- Si $|J| > p$, supongamos que $S(J)$ es menor a p . Entonces existe un intervalo $J' = [l', r'] \subseteq J$ activo, como en la Figura 3.7.

Si $l' = r'$, podemos desactivar J' y agregar l' a $S(J)$. Si no, podemos hacer la reasignación

$$\begin{aligned} J' &\leftarrow [l'+1, r'] \\ S(J) &\leftarrow S(J) \cup \{l'\}; \end{aligned}$$

el costo disminuye en $C([l', r']) - C([l'+1, r'])$, y $S(J)$ tiene ahora un elemento más. Podemos continuar este proceso hasta que $|S(J)|$ alcance p .

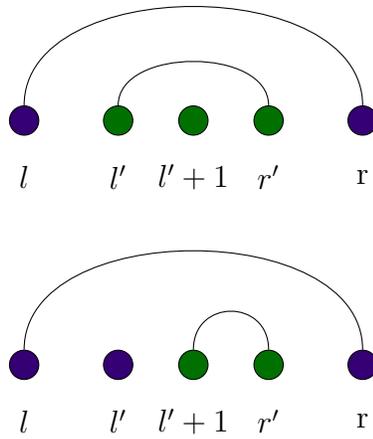


Figura 3.7: Ilustración del segundo caso de la demostración del Lema 3.9. Si el intervalo $[l, r]$ atiende a menos de p clientes (arriba), puede ‘absorber’ alguno de los de sus intervalos interiores (abajo).

□

3.3.3. Capacidades arbitrarias

El análisis de la Sección 3.3.2 es lo suficientemente flexible para permitir capacidades distintas en cada intervalo: bastaría reemplazar la ecuación 3.3 por

$$OPT[l, r, 0] = \min \left(\min_{\ell=l, \dots, r-1} OPT[l, \ell, 0] + OPT[\ell + 1, r, 0], \right. \\ \left. C([l, r]) + \min_{\ell=l+1, \dots, r-1} OPT[l + 1, \ell - 1, 0] + OPT[\ell + 1, r - 1, p_{lr} - 3] \right), \quad (3.5)$$

donde p_{lr} es la capacidad del centro que atiende al intervalo $[l, r]$. Esto permitiría extender el algoritmo 3 para resolver de manera exacta *soft CSORS* en la línea.

No obstante, hay que hacer una salvedad: en este caso ya no se cumple la propiedad 3.1: puede convenir atender a un intervalo desde un centro lejano a su punto medio, como en la instancia de la Figura 3.8. En consecuencia, ya no hay un solo intervalo por par de clientes. Volvemos entonces a usar todos los intervalos $[l, r]_i$, para i centro y l, r clientes: recordemos que el costo de cada uno de ellos es $C_i([l, r])$, y su capacidad p_i .

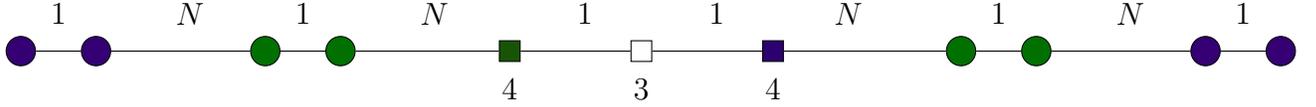


Figura 3.8: En esta instancia de *CSORS*, las capacidades están escritas bajo cada centro. El óptimo activa dos intervalos, ninguno de los cuales es atendido desde el centro más cercano a su punto medio.

La demostración de la propiedad 3.7 se mantiene, puesto que en ella solo utilizamos que las capacidades eran mayores a 2. En consecuencia, las recursiones

$$OPT[l, r, 0] = \min \left(\min_{\ell=l, \dots, r-1} OPT[l, \ell, 0] + OPT[\ell + 1, r, 0], \right. \\ \left. \min_{i \in \mathcal{I}} (C_i([l, r]) + \min_{\ell=l+1, \dots, r-1} OPT[l + 1, \ell - 1, 0] + OPT[\ell + 1, r - 1, p_i - 3]) \right), \quad (3.6)$$

y

$$OPT[l, r, t] = \min_{\ell=l, \dots, r} OPT[l, \ell - 1, 0] + OPT[\ell + 1, r - 1, t - 1] \quad (3.7)$$

son correctas. El algoritmo de programación dinámica toma tiempo $O(n^3 m p^*)$, donde

$$p^* = \max_i p_i.$$

Concluimos así el siguiente resultado:

Teorema 3.10 *Soft CSORS en la línea es polinomial. Más aún, existe un algoritmo que lo resuelve en tiempo $O(n^3mp^*)$, donde p^* es la máxima capacidad de algún centro.*

Observamos que ahora la tabla de costos requiere iterar sobre todos los intervalos $[l, r]_i$: por lo tanto a diferencia de los otros casos construirla toma tiempo $O(n^2m)$.

3.4. Caso hard-capacitado

El caso en que cada centro solo se abre una vez (o equivalentemente, que cada servidor solo atiende una vez por centro), se resiste al análisis hecho para el caso soft-capacitado. Esto se debe a que en los pasos recursivos es necesario tomar nota de los centros ya activos: y esto rompe con la propiedad de subestructura óptima que teníamos hasta este momento. Algunas propiedades sobre la posición de los centros se pueden ver en la Sección 3.4.2.

Sin embargo, es sencillo construir una 2-aproximación para el caso uniformemente capacitado, como veremos a continuación.

3.4.1. Construcción de un algoritmo de aproximación

El problema se vuelve más trabajable con otra función objetivo. Definimos, para un intervalo de clientes $J = [l, r]$ y un centro i ,

$$C_i(J) = C_i([l, r]) = w_{lr} + \text{dist}(i, J),$$

donde dist es la distancia usual entre un punto y un conjunto,

$$\text{dist}(i, J) = \text{mín}\{w_{ix} : x \in J\} :$$

es decir, el nuevo costo es el largo del intervalo, más la distancia al centro que lo atiende.

Definimos también $C(J) = \text{mín}_{i \in \mathcal{I}} C_i(J)$. Al problema con esta función objetivo lo llamaremos hard p-CSORS *modificado*, o hard CSORS *modificado*, dependiendo de si las capacidades son uniformes o no.

Inmediatamente notamos las propiedades siguientes.

Proposición 3.11

1. Para todo centro i , la función C_i es creciente en los intervalos. En consecuencia, C también lo es.
2. Para cualquier intervalo $J = [l, r]$ y centro i ,

$$C_i(J) \leq C(J) \leq 2C_i(J).$$

DEMOSTRACIÓN.

1. Sea i un centro, y $J_1 = [l_1, r_1]$, $J_2 = [l_2, r_2]$ dos intervalos de clientes tales que $J_1 \supseteq J_2$. Si $i \in J_2$, entonces

$$\mathcal{C}_i(J_2) = w_{l_2 r_2} \leq w_{l_1 r_1} = \mathcal{C}_i(J_1).$$

Si i está en $J_1 \setminus J_2$, supongamos que es menor a l_1 . Entonces,

$$\mathcal{C}(J_2) = w_{l_2 r_2} + w_{i l_2} \leq w_{l_1 r_1} = \mathcal{C}(J_1),$$

puesto que $w_{l_1 r_1} \geq w_{l_1 i} + w_{i l_2} + w_{l_2 r_2}$. El caso donde i es mayor a r_2 es análogo, por simetría.

Por último, si i está fuera de J_1 , supongamos que es menor a l_1 . Entonces,

$$\begin{aligned} \mathcal{C}(J_2) &= w_{l_2 r_2} + w_{i l_2} \\ &\leq w_{l_2 r_1} + w_{i l_2} \\ &\leq w_{l_2 r_1} + w_{i l_1} + w_{l_1 l_2} \\ &= w_{l_1 r_1} + w_{i l_1} \\ &= \mathcal{C}(J_1). \end{aligned}$$

El caso donde $i > r_1$ es análogo. Esto concluye la demostración.

2. Si $i \in J$, entonces

$$\mathcal{C}_i(J) \leq w_{lr} \leq 2\mathcal{C}_i(J).$$

Como $\mathcal{C}_i(J) = w_{lr}$, la proposición se cumple.

Por otro lado, si $i \notin J$, basta el caso donde $i < l$ (por simetría). Tenemos que

$$\mathcal{C}_i(J) = w_{ir} = w_{il} + w_{lr} = \mathcal{C}_i(J),$$

por lo que en particular se verifica la propiedad. □

La segunda de estas propiedades implica (sumando sobre los centros ocupados) que para cualquier solución del problema, el costo según \mathcal{C} es a lo más el doble del costo según \mathcal{C} : por lo tanto, un algoritmo exacto para hard CSORS modificado es una 2-aproximación del problema original.

Este factor es ajustado, como se puede ver en el ejemplo de la Figura 3.9.

Veremos que el óptimo para \mathcal{C} lo podemos encontrar en tiempo polinomial, adaptando la recursión de la Sección 3.2. Para esto es necesario probar que los intervalos de una solución óptima para \mathcal{C} son disjuntos: separamos la demostración de esto en dos partes.

Recordemos que para un intervalo activo J_i , denotamos por $S_i \subseteq J_i$ al conjunto de clientes atendidos desde i . Podemos asumir sin pérdida de generalidad que ambos extremos de J_i siempre están en S_i .

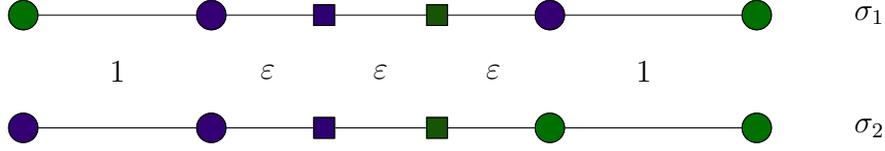


Figura 3.9: En la instancia de hard 2-CSORS de la figura, σ_1 y σ_2 son atenciones óptimas para C y para \mathcal{C} respectivamente. Notamos que $C(\sigma_1) = 1 + 4\varepsilon$, mientras que $C(\sigma_2) = 2 + 2\varepsilon$; mostrando que el factor 2 de esta aproximación es ajustado.

Lema 3.12 *Los intervalos de un óptimo para \mathcal{C} forman una familia laminar.*

DEMOSTRACIÓN. Sean $J_1 = [l_1, r_1]$, $J_2 = [l_2, r_2]$ intervalos de clientes activos en un óptimo. Denotemos por S_1 y S_2 a $S(J_1)$ y $S(J_2)$ respectivamente.

Supongamos que $l_1 < l_2 < r_1 < r_2$ (podemos asumir que son distintos ya que cada cliente es atendido desde un solo centro). Sea j el cliente de $S_2 \cap [l_2, r_1]$ de más a la derecha, como en el esquema de la Figura 3.10. Notamos que como $l_2 < r_1$, j siempre existe.

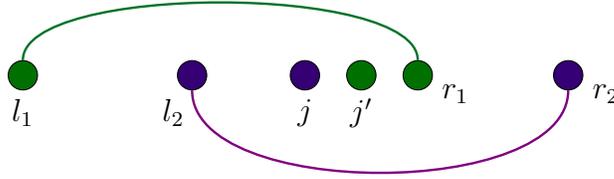


Figura 3.10: Ilustración de la situación del Lema 3.12.

Supongamos que $j + 1 = r_1$: en tal caso, podemos hacer

$$\begin{aligned} J'_1 &= [l_1, j] \\ S'_2 &= S_2 - j + r_1 \end{aligned}$$

(modificando también J_2 si $j = l_2$). Como $J'_1 \subseteq J_1$, su costo no puede ser mayor: lo mismo sucede con J_2 . Por lo tanto, podemos disminuir en 1 el número de clientes de $S_2 \cap [l_1, r_1]$; si $j = l_2$, esto separa los dos intervalos.

Por otro lado, supongamos que $j + 1 = j' < r_1$. Sabemos por maximalidad de j que $j' \notin S_2$. Por otro lado, si $j' \notin S_1 \cup S_2$, nos es irrelevante. Podemos asumir entonces que $j' \in S_1$ y hacer el intercambio de atenciones

$$\begin{aligned} S'_1 &= S_1 - j' + j \\ S'_2 &= S_2 - j + j' \end{aligned}$$

Ningún intervalo aumenta de costo, y el cliente de más a la derecha de $S_2 \cap [l_2, r_1]$ es ahora j' . Podemos hacer esto hasta que j' sea r_1 , que es el caso anterior: por inducción concluimos la laminaridad. \square

Proposición 3.13 *Los intervalos de un óptimo para soft p -CSORS modificado son disjuntos.*

DEMOSTRACIÓN. Sabemos que ya son laminares, por el lema anterior. Sean entonces $J_1 = [l_1, r_1]$, $J_2 = [l_2, r_2]$ atendidos por i_1, i_2 respectivamente, tales que $l_1 < l_2 < r_2 < r_1$. Asumamos también, sin pérdida de generalidad, que $i_1 \leq i_2$ (el caso contrario es simétrico).

Consideremos una enumeración de los clientes de $S_1 \cup S_2$ como $l_1 = 1 < 2 < \dots < r = r_1$; notamos que $r \leq 2p$. Para algún $a \in \{r - p, p\}$, consideremos los intervalos

$$\begin{aligned} J_1^a &= [1, a] \\ J_2^a &= [a + 1, r]. \end{aligned}$$

Intuitivamente, deseamos atender, o bien a los primeros p clientes desde i_1 y los restantes desde i_2 , o viceversa; notamos que ambas opciones son factibles.

Para simplificar la notación, denotaremos por \mathcal{C}_1 a $\mathcal{C}_{i_1}(J_1)$ y por \mathcal{C}_1^a a $\mathcal{C}_{i_1}(J_1^a)$; adoptaremos la misma convención para los costos de i_2 . Por último, denotaremos $\mathcal{C}_{12} = \mathcal{C}_1 + \mathcal{C}_2$, y $\mathcal{C}^a = \mathcal{C}_1^a + \mathcal{C}_2^a$.

Sabemos que

$$\mathcal{C}_1^a = \begin{cases} w_{1a} + w_{i_1 1} & \text{si } i_1 \leq 1 \\ w_{1a} & \text{si } 1 < i_1 \leq a \\ w_{1a} + w_{ai_1} & \text{si } i_1 > a, \end{cases}$$

que podemos reescribir como

$$\mathcal{C}_1^a = \begin{cases} \mathcal{C}_1 - w_{ar}, & i_1 \leq a \\ \mathcal{C}_1 - w_{i_1 r}, & a < i_1 \leq r \\ \mathcal{C}_1, & i_1 > r. \end{cases}$$

De la misma manera, podemos comparar \mathcal{C}_2^a con \mathcal{C}_2 : para esto hay doce casos, dependiendo de las posiciones de $a + 1$ e i_2 con respecto a l_2, r_2 y r :

$$\mathcal{C}_2^a = \begin{cases} \mathcal{C}_2 + w_{r_2r}, & i_2 \leq l_2 \\ \mathcal{C}_2 + w_{r_2r} - w_{l_2i_2}, & l_2 < i_2 \leq r_2 \leq a + 1 \\ w_{i_2r}, & l_2 \leq r_2 < i_2 < a + 1 \\ \mathcal{C}_2 - w_{l_2a+1} + w_{i_2r}, & r_2 \leq a + 1 < i_2 \leq r \\ \mathcal{C}_2 - w_{l_2a+1}, & l_2 < a + 1 < r < i_2 \\ \mathcal{C}_2 + w_{r_2r} - w_{l_2i_2}, & l_2 \leq i_2 \leq a + 1 \leq r_2 \\ \mathcal{C}_2 + w_{r_2r} - w_{l_2a+1}, & l_2 \leq a + 1 \leq i_2 \leq r_2 \\ \mathcal{C}_2 + w_{i_2r} - w_{l_2a+1}, & l_2 \leq a + 1 \leq r_2 \leq i_2 \leq r \\ \mathcal{C}_2 + w_{r_2r} + w_{a+1i_2}, & a + 1 < i_2 \leq l_2 \\ \mathcal{C}_2 + w_{r_2r} + w_{a+1l_2}, & a + 1 < l_2 \leq i_2 \leq r_2 \\ \mathcal{C}_2 + w_{i_2r} + w_{a+1l_2}, & a + 1 < l_2 \leq r_2 < i_2 \leq r \\ \mathcal{C}_2 + w_{a+1l_2}, & a + 1 < l_2 \leq r_2 \leq r < i_2 \end{cases}$$

Basta verificar caso a caso que con algún $a \in \{r-p, p\}$, se cumple que $\mathcal{C}^a \leq \mathcal{C}_{12}$. Indexaremos estos casos con pares (x, y) , donde x es una de las tres opciones para \mathcal{C}_1 e y una de las doce para \mathcal{C}_2 .

Una observación útil es que $p \geq l_2$: de no ser así, i_1 atendería por lo menos p clientes a la izquierda de l_2 , y uno a la derecha, r , lo que no es factible. El mismo argumento sirve para probar que $r - p \leq r_2$. Podemos, al escoger uno de estas posibilidades para a , aplicar estas desigualdades.

Podemos, entonces, agrupar estos pares de casos en tipos. En primer lugar, hay algunos pares triviales:

- Los de la forma $(x, 5)$ son directamente menores que \mathcal{C}_{12} .
- Escogiendo $a = p$, los casos (x, y) con $y \geq 9$ no pueden suceder, pues requieren que a sea menor a l_2 . Lo mismo sucede en el caso $(2, 1)$, usando que $a < i_1 \leq i_2 \leq l_2$.
- El hecho que $i_1 \leq i_2$ hace infactibles todos los casos de la forma $(3, y)$ (salvo el $(3, 5)$, cubierto por el primer ítem).

Por otro lado, hay algunos casos en los que no importa el valor de a que escojamos:

(2,2): Como $i_1 \leq i_2 \leq r_2$, tenemos que

$$w_{i_1r} = w_{i_1r_2} + w_{r_2r}.$$

Así,

$$\begin{aligned} \mathcal{C}_{12}^a &= \mathcal{C}_{12} - (w_{i_1r_2} + w_{r_2r}) + w_{r_2r} - w_{l_2i_2} \\ &= \mathcal{C}_{12} - w_{i_1r_2} - w_{l_2i_2}, \end{aligned}$$

lo cual claramente es menor o igual a \mathcal{C}_{12} .

(2,3): Tenemos aquí que

$$C_{12}^a = C_{12} - w_{i_1 r} + w_{i_2 r} - C_2.$$

Como $i_1 \leq i_2 < r$, podemos descomponer $w_{i_1 r}$ en $w_{i_1 i_2} + w_{i_2 r}$: entonces,

$$C_{12}^a = C_{12} - w_{i_1 i_2} - C_2,$$

que es menor a C_{12}^a .

(2,4): El costo de este caso es el mismo que el del caso (2, 3), reemplazando $-C_2$ por $-w_{l_2 a+1}$: se puede hacer el mismo acotamiento.

(2,6): Aquí tenemos que

$$C_{12}^a = C_{12} - w_{i_1 r} + w_{r_2 r} - w_{l_2 i_2},$$

pero sabemos que $i_1 \leq i_2 \leq r_2$: por lo tanto, $w_{r_2 r} - w_{i_1 r} \leq 0$ y concluimos.

(2,7): Este caso es idéntico al anterior reemplazando $w_{l_2 i_2}$ por $w_{l_2 a+1}$.

(2,8): De manera similar a los dos casos anteriores, tenemos que

$$C_{12}^a = C_{12} - w_{i_1 r} + w_{i_2 r} - w_{l_2 a+1};$$

como $i_1 \leq i_2 \leq r$, tenemos que

$$w_{i_2 r} - w_{i_1 r} \leq 0$$

y concluimos.

En los casos restantes podemos verificar la propiedad escogiendo $a = r - p$:

(1,1): El costo aquí es

$$C_{12}^a = C_{12} - w_{ar} + w_{r_2 r},$$

que es menor o igual a C_{12} cuando $a \leq r_2$: pero sabemos que $r - p \leq r_2$. Notamos que los casos (1, 2), (1, 6), (1, 7) y (1, 8) se pueden acotar superiormente por este, por lo que su demostración es la misma.

(1,3): Tenemos que

$$\begin{aligned} C_{12}^a &= C_1 - w_{ar} + w_{i_2 r} \\ &= C_1 - (w_{aa+1} + w_{a+1 r}) + w_{i_2 a+1} + w_{a+1 r} \\ &= C_1 - w_{aa+1} + w_{i_2 a+1}, \end{aligned}$$

que es menor a C_1 si $a \leq i_2$, lo que se cumple para $r - p$ (pues $r - p \leq r_2 \leq i_2$).

(1,4): Aquí notamos que $r - p \leq r_2 \leq i_2$, por lo tanto $w_{i_2 r} \leq w_{r_2 r}$, y

$$C_{12}^{r-p} \leq C_{12} - w_{ar} + w_{r_2 r},$$

reduciéndonos al caso 1.1.

$\mathcal{C}_1 \backslash \mathcal{C}_2$	1	2	3	4	5	6	7	8	9	10	11	12
1												
2												
3												

Tabla 3.1: Resumen de la división en casos de la Proposición 3.13. Los casos triviales o infactibles para $i_1 \leq i_2$ se muestran en gris, los casos infactibles para $a = p$ en violeta. Los casos en azul se cumplen para cualquier elección de a , y los casos en verde, con $a = r - p$.

La Tabla 3.1 resume con colores la forma de agrupar los casos.

Podemos hacer esto para cada par de intervalos anidados. Concluimos, entonces, que todo par de intervalos debe ser disjunto. \square

A continuación, es necesario asegurar que los centros desde los que se atienden los intervalos sean distintos. Para esto probamos la propiedad siguiente:

Lema 3.14 Sean J_1, J_2 intervalos activos en un óptimo, y sean i_1, i_2 los centros que los atienden (respectivamente). Si J_1 está a la izquierda de J_2 , entonces $i_1 \leq i_2$.

DEMOSTRACIÓN. Escribamos $J_1 = [l_1, r_1]$ y $J_2 = [l_2, r_2]$. Sabemos que $l_1 \leq r_1 < l_2 \leq r_2$. Supongamos que $i_1 > i_2$. Veremos que entonces, el costo al intercambiar los intervalos de centro es menor: esto es, que

$$\mathcal{C}_1(J_2) + \mathcal{C}_2(J_1) \leq \mathcal{C}_1(J_1) + \mathcal{C}_2(J_2);$$

donde hemos abreviado $\mathcal{C}_{i_1}(\cdot)$ y $\mathcal{C}_{i_2}(\cdot)$ como \mathcal{C}_1 y \mathcal{C}_2 respectivamente.

Aplicando la definición de \mathcal{C} , vemos que esto es equivalente a probar que

$$\text{dist}(i_1, J_2) + \text{dist}(i_2, J_1) \leq \text{dist}(i_1, J_1) + \text{dist}(i_2, J_2). \quad (3.8)$$

Haremos esto por casos:

- Si $i_1 \in J_1$, entonces $\text{dist}(i_1, J_1) = 0$ podemos reescribir el lado derecho de la inecuación como

$$\text{dist}(i_2, J_2) = \text{dist}(i_2, J_1) + w_{l_1 i_1} + \text{dist}(i_1, J_2),$$

lo que es mayor que el lado izquierdo.

- Si $i_1 < l_1$, tenemos que

$$\text{dist}(i_2, J_2) = w_{i_2 i_1} + \text{dist}(i_1, J_2).$$

Por otra parte,

$$\text{dist}(i_1, J_1) = \text{dist}(i_2, J_1) - w_{i_2 i_1}.$$

Sumando ambas identidades, vemos que la inecuación 3.8 se cumple con igualdad.

- Si $r_2 \leq i_2 < i_1$, tenemos simplemente que

$$\begin{aligned} \text{dist}(i_1, J_1) &= w_{i_2 i_1} + \text{dist}(i_2, J_1) \\ \text{dist}(i_2, J_2) &= w_{i_2 i_1} + \text{dist}(i_1, J_2), \end{aligned}$$

y la desigualdad 3.8 se cumple estrictamente.

- Finalmente $i_1 > r_1$, pero $i_2 \leq r_2$, tenemos que

$$\text{dist}(i_2, J_2) \geq \text{dist}(i_2, J_1) + \text{dist}(i_1, J_1) + \text{dist}(i_1, J_2),$$

que ya es mayor al lado izquierdo de la desigualdad buscada.

Los casos restantes son simetrías de estos, y se pueden resolver de manera análoga. \square

Esto tiene como consecuencia que los intervalos y los centros que los atienden vienen en el mismo orden: si sabemos que un intervalo $[\ell, j]$ pertenece al óptimo, y es atendido por un centro i , todos los centros que atiendan a los clientes anteriores a ℓ son anteriores a i . Podemos, entonces, establecer la recursión

$$\text{OPT}[j, i] = \min_{\substack{\ell=2, \dots, j \\ i'=2, \dots, i}} (\mathcal{C}_{i'}([\ell, j]) + \text{OPT}[\ell - 1, i' - 1]), \quad (3.9)$$

donde $\text{OPT}[j, i]$ es el costo de atender óptimamente a los clientes de $[1, j]$ con centros menores o iguales a i . Los casos base son

$$\text{OPT}[j, i] = \infty \quad \forall i, j, j > pi$$

(pues la instancia se vuelve infactible) y

$$\text{OPT}[j, 1] = \mathcal{C}_1([1, j]) \quad \forall j \leq p,$$

pues solo hay una apertura factible. El Algoritmo 4 encuentra $\text{OPT}[n, m]$ mediante programación dinámica. Notamos que el paso 5 requiere $ij = O(nm)$ lecturas de las tablas \mathcal{C} y OPT , por lo que este algoritmo toma tiempo $O(m^2 n^2)$. En conclusión,

Teorema 3.15 *Existe una 2-aproximación para hard p -CSORS en la línea, para cualquier $p \geq 3$.*

Algoritmo 4

Entrada: Tabla $(C([i, j]))_{i, j \in \mathcal{J}}$ con los costos de cada intervalo; n el número de clientes, m el número de centros.

- 1: Inicializar tabla OPT que guardará los costos de los subproblemas.
 - 2: Definir los casos base como descrito más arriba.
 - 3: **para** $j = 2, \dots, n$ **hacer**
 - 4: **para** $i = 2, \dots, m$ **hacer** Calcular $\text{OPT}[j, i]$ usando la fórmula 3.9. ▷ Esto toma tiempo $O(mn)$.
 - 5: Entregar $\text{OPT}[n, m]$
-

3.4.2. Propiedades adicionales de la variante hard-capacitada

Una manera alternativa de estudiar la variante hard-capacitada es intentar aprovechar propiedades de la estructura lineal, sin alterar la función objetivo. Esta sección estudia algunas de estas propiedades, no para utilizarlas en esta tesis, sino por su posible utilidad en el futuro.

Sabemos que en el caso hard-capacitado, varias propiedades del caso soft-capacitado se mantienen: para empezar, la función objetivo sigue siendo creciente para los intervalos (Lema 3.2). Además, los intervalos de clientes siguen teniendo una estructura laminar fuerte, ya que la demostración de la propiedad 3.7 no cambia los centros que atienden a cada intervalo.

El principal desafío es encontrar una relación entre los intervalos activos y los centros que los atienden. Con respecto a ello, podemos probar las siguientes propiedades:

Lema 3.16 Sean i_1, i_2 centros utilizados en un óptimo, abriendo intervalos J_1, J_2 respectivamente. Entonces,

1. Si J_1 está a la izquierda de J_2 , entonces $i_1 \leq i_2$.
2. Si $J_1 \subseteq J_2$, entonces $i_1 \in J_2$.
3. Si J_2 contiene algún centro no utilizado, o bien $J_1 \subseteq J_2$, entonces $i_2 \in J_2$.

DEMOSTRACIÓN. Sean $J_1 = [l_1, r_1]$ y $J_2 = [l_2, r_2]$. Llamaremos, por comodidad, C_1 y C_2 a las funciones de costo C_{i_1} y C_{i_2} respectivamente.

Probemos la primera propiedad por contradicción: supongamos que J_1 está a la izquierda de J_2 (es decir que $r_1 < l_2$) y que $s > t$. El aporte de estos intervalos a la función objetivo es $C_1(J_1) + C_2(J_2)$. Si se “intercambiaran” los intervalos de centro, el costo de sería $C_1(J_2) + C_2(J_1)$. Llamemos

$$x_1 = \frac{l_1 + r_1}{2}$$
$$x_2 = \frac{l_2 + r_2}{2}$$

a los puntos medios de cada intervalo. Notamos que el que el costo de atender a $J = [l, r]$ desde un centro i sea w_{il} o w_{ir} solo depende de si i está a la izquierda o derecha del punto medio de I . Esto define los siguientes casos, ilustrados en la Figura 3.11:

- Si $i_1 < x_1$, tenemos que $C_1(J_1) = w_{i_1 r_1}$. Además, como $i_2 < i_1$, tenemos que $C_2(J_2) = w_{i_2 r_2}$, $C_1(J_2) = w_{i_1 r_2}$ y $C_2(J_1) = w_{i_2 r_1}$.

Pero entonces podemos intercambiar los intervalos de centro, ya que,

$$\begin{aligned} C_1(J_1) + C_2(J_2) &= w_{i_1 r_1} + w_{i_2 r_2} \\ &= w_{i_2 r_1} - w_{i_2 i_1} + w_{i_2 i_1} + w_{i_1 r_2} \\ &= C_1(J_2) + C_2(J_1). \end{aligned}$$

El caso donde $i_2 > x_2$ es análogo, por simetría.

- Si $x_1 < i_2 < i_1 < x_2$, se contradice optimalidad, ya que el costo de intercambiar sería

$$\begin{aligned} C_1(J_1) + C_2(J_2) &= w_{i_1 r_1} + w_{i_2 l_2} \\ &= (w_{i_1 i_2} + w_{i_2 r_1}) + (w_{i_2 i_1} + w_{i_1 l_2}) \\ &> w_{i_2 r_1} + w_{i_1 l_2} \\ &= C_1(J_2) + C_2(J_1) : \end{aligned}$$

que es el costo de la asignación original.

- Por último, puede ser que $i_2 < x_1 < i_1$. Aquí,

$$\begin{aligned} C_1(J_1) + C_2(J_2) &= w_{i_1 l_1} + w_{i_2 r_2} \\ &\geq w_{i_1 r_1} + w_{i_2 i_1} + w_{i_1 r_2} \\ &\geq w_{i_2 r_1} + w_{i_1 r_2} \\ &= C_1(J_2) + C_2(J_1), \end{aligned}$$

por lo que también conviene hacer el intercambio. Notamos que la desigualdad puede ser estricta o no dependiendo de si i_1 pertenece o no a J_1 .

Ahora la segunda propiedad. Supongamos que $J_1 \subseteq J_2$ pero $i_1 \notin J_2$. Sin pérdida de generalidad, supongamos que i_1 está a la izquierda de J_2 . Sea j el mayor cliente atendido por i_2 que esté a la izquierda de J_1 , como se ilustra en la Figura 3.12. Denotemos por j' al mayor cliente de J_1 que esté a la izquierda de r_1 , si es que existe (es decir, si $|J_1| > 1$).

Podemos disminuir el costo de la solución atendiendo a r_1 desde i_2 y a j desde i_1 : $C_2(J_2)$ se mantiene, pero ahora

$$C_1(J_1) = w_{i_1 j'} \leq w_{i_1 r_1}.$$

Podemos continuar este proceso hasta que J_1 conste de un solo nodo r_1 . Allí, podemos simplemente asignarle r_1 a i_2 , y l_2 a i_1 , como se muestra en la parte inferior de la Figura

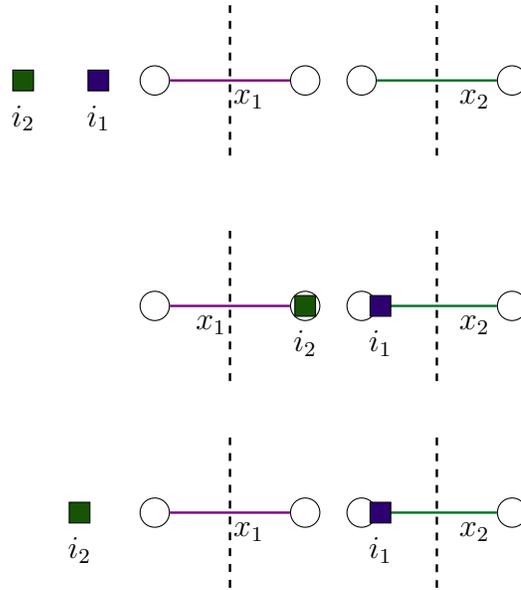


Figura 3.11: Una ilustración con los casos del Lema 3.16, en el mismo orden del texto.

3.12. Aquí ambos costos pueden disminuir, ya que J_2 ahora comienza desde el siguiente nodo $l_2 + 1$. Los intervalos J_1, J_2 nuevos quedan así disjuntos sin aumentar el costo.

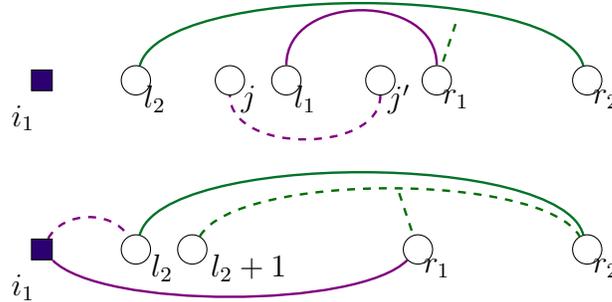


Figura 3.12: Ilustración de la demostración de la segunda propiedad del Lema 3.16. Las líneas punteadas representan los cambios. En el grafo de arriba, se ve que el ‘desplazamiento’ del primer intervalo reduce su costo; en el de abajo, que la separación de éstos produce el mismo efecto.

Para la tercera propiedad, supongamos que i_2 está fuera de J_2 . Sabemos entonces que para todo i centro dentro de J_2 ,

$$C_i(J_2) \leq w_{l_2 r_2} \leq C_2(J_2) : \tag{3.10}$$

por lo que si existe un centro $i \in J_2$ no utilizado, reemplazar i_2 por i es una mejor solución.

Por otro lado, si existe $i_1 \in J_2$ abierto, hay dos opciones: o bien $J_1 \subseteq J_2$, o son disjuntos.

Supongamos, sin pérdida de generalidad, que $i_2 < l_2$. Entonces, sabemos que

$$C_2(J_1) = w_{i_2 r_1} \leq w_{i_2 r_2} = C_2(J_2),$$

y por lo tanto podemos mejorar la solución intercambiando a i_1 e i_2 de roles (puesto que i_1 también cumple la ecuación 3.10). \square

Observación Una afirmación más fuerte que la tercera propiedad puede ser “si existe un centro dentro de J_2 , entonces $i_2 \in J_2$ ”. Sin embargo, esto no necesariamente es así, como se ilustra en el siguiente contraejemplo:

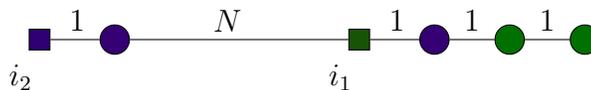


Figura 3.13: En esta instancia, los extremos de J_1 y J_2 están marcados con los colores de i_1 e i_2 respectivamente. Notamos que la solución es óptima, e $i_2 \notin J_2$ pese a que J_2 contiene un centro.

Conclusión

El problema de CSORS presenta sus desafíos. Para el caso general, tanto estrategias del tipo redondeo como el esquema primal-dual resultaron más complicadas de lo esperado. Estrategias como el estudio de la dimensión de Apnik–Chervonenkis también fueron infructuosas (ésta es infinita para ∞ -CSORS, incluso acotándose a árboles). Concentrándose en un caso más acotado como la línea, sin embargo, es posible dar los primeros pasos hacia un estudio más profundo. Hay dos líneas en las que se podría continuar, no excluyentes entre sí:

La primera es extender la estructura sobre la que se trabaja. Por ejemplo es posible extender la Propiedad 3.3 a ∞ -CSORS en árboles, convirtiendo el problema en uno de tipo “minimizar suma de radios en una partición en subárboles”. Para una arborescencia, es simple establecer una relación recursiva como en la Sección 3.2. En árboles, sin embargo, o incorporando capacidades, por ahora hay más preguntas que respuestas: idealmente, tras responderlas, se escalaría hacia grafos generales.

La segunda es extender la pregunta. Por ejemplo, ¿qué pasa si los centros deben estar conectados por un árbol, o por un ciclo? En particular, puede resultar muy interesante (y complicado) el caso en que las distancias entre centros se calculen con una métrica distinta a los tiempos de atención. Un problema así puede interpretarse como una variante del *Vehicle Routing Problem* [24] en la que los vehículos no llegan directamente a los clientes sino a *estacionamientos* o puntos de detención, desde el cual se atiende a los clientes a pie. Otra manera de interpretarlo sería una extensión de Connected Facility Location.

Es el propósito de los resultados obtenidos, más que responder concluyentemente las preguntas formuladas en la introducción, contribuir a las bases necesarias para el desarrollo posterior de más algoritmos para esta familia de problemas. El tiempo y futura investigación al respecto se encargarán de decidir si este propósito fue alcanzado.

Bibliografía

- [1] Jakob Berndorfer and Sophie N. Parragh. Modeling and solving a real world machine scheduling problem with due windows and processing set restrictions. *Procedia Computer Science*, 200:1646–1653, 2022. 3rd International Conference on Industry 4.0 and Smart Manufacturing.
- [2] M Charikar, S Guha, Éva Tardos, and David Shmoys. A constant-factor approximation algorithm for the k -median problem. *Proceedings of 31st Annual ACM Symposium on Theory of Computing*, pages 1–10, 01 1999.
- [3] Moses Charikar and Rina Panigrahy. Clustering to minimize the sum of cluster diameters. *Journal of Computer and System Sciences*, 68(2):417–441, 2004. Special Issue on STOC 2001.
- [4] Senthilkumar Dr and Sockalingam Narayanan. Literature review of single machine scheduling problem with uniform parallel machines. *Intelligent Information Management*, 2:457–474, 01 2010.
- [5] Doratha Drake and Stefan Hougardy. On approximation algorithms for the terminal steiner tree problem. *Information Processing Letters*, 89:15–18, 10 2003.
- [6] Zve Drezner, Avram Mehrez, and George O. Wesolowsky. The facility location problem with limited distances. *Transportation Science*, 25(3):3:183–3:187, August 1991.
- [7] Emrah B. Edis and Ceyda Oguz. Parallel machine scheduling with flexible resources. *Computers & Industrial Engineering*, 63(2):433–447, 2012.
- [8] Friedrich Eisenbrand, Fabrizio Grandoni, Thomas Rothvoß, and Guido Schaefer. Approximating connected facility location problems via random facility sampling and core detouring. pages 1174–1183, 01 2008.
- [9] Kuei-Tang Fang and Bertrand M.T. Lin. Parallel-machine scheduling to minimize tardiness penalty and power cost. *Computers & Industrial Engineering*, 64(1):224–234, 2013.
- [10] Michael R. Garey and David S. Jhonson. *Computers and Intractability*. W. H. Freeman and Company, 1979.
- [11] Matt Gibson, Gaurav Kanade, Erik Krohn, Imran Pirwani, and Kasturi Varadarajan. On metric clustering to minimize the sum of radii. pages 282–293, 07 2008.

- [12] Dishant Goyal and Ragesh Jaiswal. Tight FPT approximation for constrained k-center and k-supplier. *CoRR*, abs/2110.14242, 2021.
- [13] Anupam Gupta, Amit Kumar, and Tim Roughgarden. Simpler and better approximation algorithms for network design. ACM Press, 2003.
- [14] Dorit Hochbaum and David Shmoys. A best possible heuristic for the k-center problem. *Mathematics of Operations Research - MOR*, 10:180–184, 05 1985.
- [15] John E. Hopcroft and Richard M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs*. *Siam Journal on Computing*, 2(4):4:225–4:231, December 1973.
- [16] Tanmay Inamdar and Kasturi Varadarajan. Capacitated Sum-Of-Radii Clustering: An FPT Approximation. In Fabrizio Grandoni, Grzegorz Herman, and Peter Sanders, editors, *28th Annual European Symposium on Algorithms (ESA 2020)*, volume 173 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 62:1–62:17, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.
- [17] Tanmay Inamdar and Kasturi Varadarajan. Capacitated Sum-Of-Radii Clustering: An FPT Approximation. In Fabrizio Grandoni, Grzegorz Herman, and Peter Sanders, editors, *28th Annual European Symposium on Algorithms (ESA 2020)*, volume 173 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 62:1–62:17, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.
- [18] Samir Khuller and Yoram Sussmann. The capacitated k-center problem. *SIAM Journal on Discrete Mathematics*, 13, 03 2001.
- [19] Shi Li and Ola Svensson. Approximating k -median via pseudo-approximation. *Proceedings of the Annual ACM Symposium on Theory of Computing*, 11 2012.
- [20] Guohui Lin and Guoliang Xue. On the terminal steiner tree problem. *Information Processing Letters*, 84(2):103–107, 2002.
- [21] Mohammad Mahdian, Yinyu Ye, and Jiawei Zhang. A 2-approximation algorithm for the soft-capacitated facility location problem. In Sanjeev Arora, Klaus Jansen, José D. P. Rolim, and Amit Sahai, editors, *Approximation, Randomization, and Combinatorial Optimization.. Algorithms and Techniques*, pages 129–140, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.
- [22] Ethel Mokotoff. An exact algorithm for the identical parallel machine scheduling problem. *European Journal of Operational Research*, 152(3):758–769, 2004. Applications of Soft O.R. Methods.
- [23] Nakilon. dynamic programming to find the minimum weight cover of the points, 2016.
- [24] Suresh Nanda Kumar and Ramasamy Panneerselvam. A survey on the vehicle routing problem and its variants. *Intelligent Information Management*, 04, 01 2012.
- [25] Michael Pinedo. *Scheduling: Theory, Algorithms, and Systems*. Grundlehren der mathematischen Wissenschaften. Springer, New York [u.a.], 3. ed edition, 2008.

- [26] Jessica Rodríguez-Pereira, Elena Fernández, Gilbert Laporte, Enrique Benavent, and Antonio Martínez-Sykora. The steiner traveling salesman problem and its extensions. *European Journal of Operational Research*, 278(2):615–628, 2019.
- [27] Chaitanya Swamy and Amit Kumar. Primal–dual algorithms for connected facility location problems. *Algorithmica*, 40:245–269, 01 2004.
- [28] Z. Ulukan and E. Demircioğlu. A survey of discrete facility location problems. *International Scholarly and Scientific Research & Innovation*, 9(15):15:2487–15:2492, 2015.
- [29] uSeemSurprised and sv_jan5. Answer to the above question about minimum weight point cover, 2016.
- [30] David P. Williamson and David B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, 2011.