Problemas de cubrimiento

Cubrimiento, partición y empaquetamiento de conjuntos.

Andrés L. Medaglia, Ph.D.
(amedagli@uniandes.edu.co)

Actualizado: mayo 9, 2024

Cubrimientos, particiones y empaquetamientos

Sea $M = \{1, \dots, m\}$ un conjunto con m objetos. Sea $M_j \subseteq M$ un subconjunto de objetos en M, con $j \in N = \{1, \dots, n\}$. Para cada subconjunto M_j ($j \in N$), definimos un peso c_j asociado con el valor de seleccionarlo. Sea $S \subseteq N$ una colección de los subconjuntos.

Cubrimiento. Decimos que S es un *cubrimiento* de M si $\bigcup_{j \in S} M_j = M$. En otras palabras, todos los elementos de M pertenecen a por lo menos un subconjunto M_j $(j \in S)$.

Partición. Decimos que S es una *partición* de M si $\bigcup_{j \in S} M_j = M$ y $M_j \cap M_k = \emptyset$, para todo $j \in S$ y $k \in S$ con $j \neq k$. En otras palabras, todos los elementos de M pertenecen a exactamente un subconjunto $M_j (j \in S)$.

Empaquetamiento. Decimos que S es un *empaquetamiento* de M si $M_j \cap M_k = \emptyset$, para todo $j \in S$ y $k \in S$ con $j \neq k$. En otras palabras, los elementos de M pertenecen a máximo un subconjunto $M_j (j \in S)$. Algunos elementos podrían no estar en algún subconjunto.

Para ilustrar estos conceptos, consideremos los conjuntos $M = \{1, \dots, 16\}$ y $N = \{1, \dots, 7\}$. Los subconjuntos de objetos están dados por: $M_1 = \{1,2,3,4,5,6,7,8\}$, $M_2 = \{3,4,7,8,11,12\}$, $M_3 = \{9,10,11\}$, $M_4 = \{5,6,9,10,13,14\}$, $M_5 = \{11,12,15,16\}$, $M_6 = \{12,16\}$ y $M_7 = \{13,14,15\}$. La ***Figura 1*** muestra un ejemplo de cubrimiento dado por $S = \{1,2,4,5\}$; la ***Figura 2*** muestra un ejemplo de partición dado por $S = \{1,3,6,7\}$; y la ***Figura 3*** muestra un ejemplo de empaquetamiento dado por $S = \{2,4\}$.

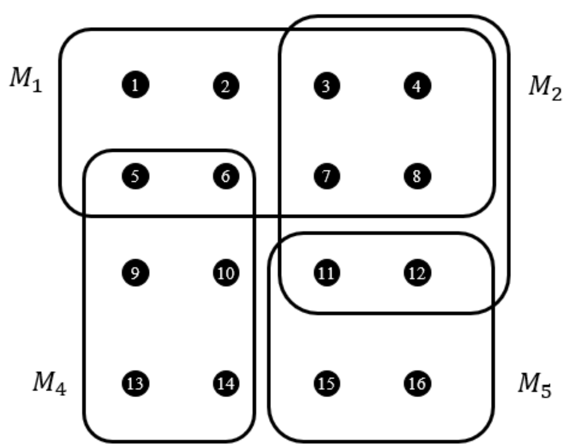

Figura 1. Cubrimiento

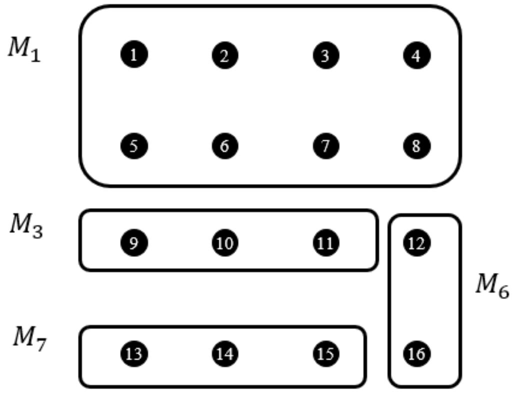

Figura 2. Partición

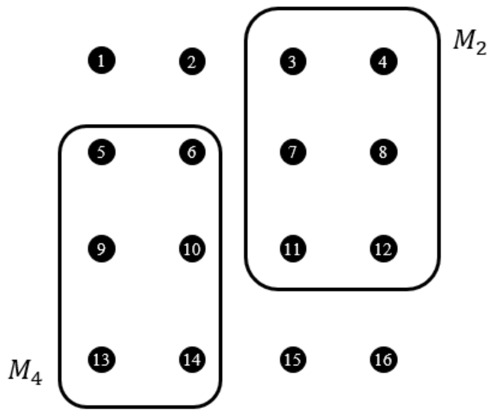

Figura 3. Empaquetamiento

Problemas de cubrimiento, partición y empaquetamiento de conjuntos

El problema de *cubrimiento de conjuntos* trata de encontrar un cubrimiento S de mínimo peso; el problema de *partición de conjuntos* busca una partición S de mínimo peso (o de máximo peso); y el problema de *empaquetamiento* busca un empaquetamiento S de máximo peso.

Para formular estos problemas como modelos de optimización lineal entera, definimos un parámetro a_{ij} que toma el valor de 1, si el objeto $i \in M_j$; y toma el valor de 0, en caso contrario. Como se conocen los objetos que hacen parte de cada uno de los subconjuntos M_j, este parámetro binario es fácil de calcular. El otro parámetro es el peso c_j asociado con cada subconjunto M_j, para todo $j \in N$. Se define también, para todo $j \in N$, la variable de decisión x_j que toma el valor de 1 si el subconjunto M_j hace parte de S; y toma el valor de 0, en caso contrario.

Problema de cubrimiento de conjuntos. Este problema se modela de la siguiente manera:

$$\min \sum_{j \in N} c_j x_j$$

s. a,

$$\sum_{j \in N} a_{ij} x_j \geq 1, \qquad \forall i \in M$$
$$x_{ij} \in \{0,1\}, \qquad \forall j \in N$$

Es común ver este modelo expresado matricialmente. Para hacerlo, basta con organizar los pesos en un vector $\mathbf{c}^T = [c_1 \dots c_j \dots c_n]$, las variables en un vector $\mathbf{x}^T = [x_1 \dots x_j \dots x_n]$ y los parámetros a_{ij} en una matriz de incidencia $\mathbf{A} = [a_{ij}] \in \{0,1\}^{m \times n}$. Definimos también un vector de unos $\mathbf{1} = [1]_{m \times 1}$ para el lado derecho de las restricciones. El problema de cubrimiento de conjuntos de forma matricial queda expresado así:

$$\min \mathbf{c}^T \mathbf{x}$$

s. a,

$$\mathbf{A}\mathbf{x} \geq \mathbf{1}$$
$$\mathbf{x} \in \{0,1\}^n$$

Problema de partición de conjuntos. Haciendo uso de la misma notación presentada en el problema de cubrimiento, el problema de partición de conjuntos se modela de la siguiente manera:

$$\min \text{(ó max)} \sum_{j \in N} c_j x_j$$

s. a,

$$\sum_{j \in N} a_{ij} x_j = 1, \qquad \forall i \in M$$
$$x_{ij} \in \{0,1\}, \qquad \forall j \in N$$

En su forma matricial, el modelo de partición es:

$$\min \text{(ó max)}\ \mathbf{c}^T \mathbf{x}$$

s. a,

$$\mathbf{A}\mathbf{x} = \mathbf{1}$$
$$\mathbf{x} \in \{0,1\}^n$$

Problema de empaquetamiento de conjuntos. Haciendo uso de la notación que se presentó en el problema de cubrimiento, el problema de empaquetamiento de conjuntos se modela de la siguiente manera:

$$\max \sum_{j \in N} c_j x_j$$

s.a,

$$\sum_{j \in N} a_{ij} x_j \leq 1, \qquad \forall i \in M$$
$$x_{ij} \in \{0,1\}, \qquad \forall j \in N$$

En su forma matricial, el modelo de empaquetamiento es:

$$\max \mathbf{c}^T \mathbf{x}$$

s.a,

$$\mathbf{Ax} \leq \mathbf{1}$$
$$\mathbf{x} \in \{0,1\}^n$$

De forma general, hablamos de formulaciones de tipo cubrimiento, partición y empaquetamiento, definidas con el vector $\mathbf{b} = [b_i] \in \mathbb{R}_+^m$, la matriz $\mathbf{A} = [a_{ij}] \in \mathbb{R}_+^{m \times n}$, el vector de pesos $\mathbf{c} = [c_j] \in \mathbb{R}^n$ y el de variables decisión $\boldsymbol{x} \in \{0,1\}^n$.

Las formulaciones tipo cubrimiento se expresan así:

$$\min \mathbf{c}^T \mathbf{x}$$

s. a,

$$\mathbf{Ax} \geq \mathbf{b}$$
$$\mathbf{x} \in \{0,1\}^n$$

Las formulaciones tipo partición se expresan así:

$$\min \text{ (ó max) } \mathbf{c}^T \mathbf{x}$$

s. a,

$$\mathbf{Ax} = \mathbf{b}$$
$$\mathbf{x} \in \{0,1\}^n$$

Finalmente, las formulaciones tipo empaquetamiento se expresan así:

$$\max \mathbf{c}^T \mathbf{x}$$

s. a,

$$\mathbf{Ax} \leq \mathbf{b}$$
$$\mathbf{x} \in \{0,1\}^n$$

Ejemplo: subastas combinatorias

Una empresa que tuesta café tiene que comprar insumos para empacar sus productos para distribuirlos en supermercados. La empresa usa una subasta combinatoria como mecanismo para hacer su proceso de compra más eficiente. A través de un portal, permite a sus proveedores registrar sus ofertas sobre los insumos que están dispuestos a ofrecer. En particular, la empresa requiere cajas en formato grande y pequeño; con impresiones gráficas de sus cafés especial y de origen. Así, la empresa requiere tres tipos de insumos que son 10,000 cajas pequeñas para empacar café especial; 800 cajas grandes para empacar café de origen; y 10,000 cajas grandes para empacar café especial. Hay proveedores que tienen la facilidad para hacer las cajas de formato grande y otros las pequeñas; igualmente algunos proveedores ya tienen el arte listo para imprimir las cajas de café especial, mientras otros pueden imprimir más fácilmente las de café de origen. Esto hace que las ofertas de los proveedores varíen, ya que podrán hacer algunos insumos más eficientemente que otros, traduciéndose así en una mejor oferta para la empresa compradora. El portal deja abierto el período de ofertas hasta una fecha y hora de cierre de la subasta. En ese momento, procesa todas las ofertas y debe asignar cuáles de las ofertas son las ganadoras. Al final, paga por las ofertas ganadoras y accede a los insumos deseados. La ***Tabla 1*** presenta las ofertas al cierre de la subasta.

Insumo		Ofertas							
		1	2	3	4	5	6	7	8
1	10,000 cajas pequeñas de café especial	✓		✓	✓			✓	✓
2	800 cajas grandes de café de origen		✓	✓			✓	✓	
3	10,000 cajas grandes de café especial	✓	✓	✓		✓			✓
Valor de la oferta		150	125	300	125	115	90	270	160

Tabla 1. Ofertas al cierre de la subasta

Para determinar los ganadores, la empresa compradora formula un modelo de cubrimiento de conjuntos de la siguiente manera. Sea $M = \{1,2,3\}$ el conjunto formado por los insumos que la empresa debe comprar. Sea $N = \{1,2,3,4,5,6,7,8\}$ el conjunto de ofertas. Sea a_{ij} el parámetro binario que indica con 1 si el insumo $i \in M$ está presente en la oferta $j \in N$; y toma el valor de 0, en caso contrario. Cada oferta $j \in N$ tiene un valor de c_j dado por el proveedor. La variable de decisión x_j toma el valor de 1 si la oferta $j \in N$ es aceptada; y toma el valor de 0, en caso contrario. La empresa compradora formula el siguiente modelo de cubrimiento de conjuntos después de obtener de las ofertas la matriz $\mathbf{A} = [a_{ij}]$ y el vector de costos $\mathbf{c} = [c_j]$:

$$\min 150x_1 + 125x_2 + 300x_3 + 125x_4 + 115x_5 + 90x_6 + 270x_7 + 160x_8$$

s.a.,

$$\begin{array}{ccccccccccc} x_1 + & & x_3 + & x_4 + & & & x_7 + & x_8 & \geq & 1 \\ & x_2 + & x_3 + & & & x_6 + & x_7 & & \geq & 1 \\ x_1 + & x_2 + & x_3 + & & x_5 + & & & x_8 & \geq & 1 \end{array}$$

$$x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8 \in \{0,1\}$$

En su forma matricial sería así:

$$\min \ [150 \quad 125 \quad 300 \quad 125 \quad 115 \quad 90 \quad 270 \quad 160]\mathbf{x}$$

s. a,

$$\begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \mathbf{x} \geq \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$\mathbf{x} \in \{0,1\}^n$$

Después de resolver el modelo de cubrimiento de conjuntos con la instancia dada, las ofertas aceptadas son la 1 y la 6. La empresa debe pagar 240 a sus proveedores.

Otras aplicaciones

Muchos problemas prácticos se pueden representar como problemas de cubrimiento, partición o empaquetamiento. Por ejemplo, los problemas de ruteo de vehículos los podemos representar haciendo uso de un modelo de partición (o cubrimiento). Cada columna binaria $\mathbf{a}_j = [a_{ij}] \in \{0,1\}^m$ de la matriz $\mathbf{A} = [\mathbf{a}_1 \mid \ldots \mid \mathbf{a}_j \mid \ldots \mid \mathbf{a}_n]$ representará la ruta de un vehículo, donde $a_{ij} = 1$ indicará que el cliente $i \in M$ es visitado por la ruta $j \in N$ (y tomará el valor de 0, si no lo visita). El problema se transforma en escoger un cubrimiento de mínimo costo de todos los clientes haciendo uso de las rutas en N.

Otras versiones de estos modelos se han aplicado con éxito a problemas logísticos relacionados a la respuesta efectiva a emergencias, donde debemos instalar unidades de emergencia que cubran zonas en una región; en problemas de programación de turnos, donde queremos que los turnos escogidos cubran todas las tareas a realizar en las franjas horarias (Restrepo et al., 2012); y en problemas de cortes en la industria, donde se debe satisfacer la demanda por un conjunto de cortes de material (e. g., láminas o rollos de acero) que se obtiene de una materia prima de gran tamaño que se debe cortar eficientemente a través de patrones de corte que eviten el desperdicio de material (Medaglia, 2002).

Para saber más: generación de columnas

Por su naturaleza combinatoria, estos problemas de cubrimiento, partición y empaquetamiento pueden tener un número muy grande de subconjuntos a considerar ($n \gg 0$). Esto implica que considerar de forma explícita todas las columnas de la matriz $\mathbf{A}$ sea prácticamente imposible.

Adicionalmente, muchos de los subconjuntos posiblemente no representen opciones atractivas para nuestro problema. Entonces, ¿por qué sería necesario enumerarlos todos antes de resolver el problema?

Afortunadamente, hay una técnica de optimización a gran escala, llamada generación de columnas, que nos permite ir descubriendo las columnas de $\mathbf{A}$ asociadas con los subconjuntos más atractivos de forma iterativa. La intuición es que, sin enumerar todos los subconjuntos explícitamente, podemos explorarlos y generar sólo las mejores alternativas, acelerando el proceso de solución (Medaglia, 2002; Desaulniers et al., 2005).

El esquema de solución por generación de columnas resuelve un modelo de optimización lineal (relajado) con una formulación de tipo cubrimiento (partición o empaquetamiento) con un número muy grande de variables de decisión (asociadas con los subconjuntos), haciendo uso de un esquema iterativo que conecta dos problemas: el maestro y el auxiliar. La idea es usar el problema auxiliar como un oráculo generador de los subconjuntos atractivos (columnas de la matriz $\mathbf{A}$), o en su defecto, para determinar que no existen. El problema maestro, que tiene una formulación de tipo cubrimiento (partición o empaquetamiento), va recibiendo las columnas generadas por el auxiliar y envía mensajes a este si es necesario generar más alternativas de subconjuntos. Al final, este proceso iterativo obtiene la solución óptima de la relajación del problema de tipo cubrimiento (partición o empaquetamiento). Este proceso se puede combinar con *branch & bound* para resolver el problema entero (de forma exacta o heurística).

Agradecimientos

Gracias a Alfaima Solano y Johan Camacho quienes revisaron versiones preliminares de este documento. Su profundidad en la lectura y sus aportes, propusieron cambios que lo mejoraron sustancialmente.

Bibliografía

Bertsimas, D., Weismantel, R. Optimization over Integers. Dynamic Ideas. 2005.

Desaulniers, G., Desrosiers, J., Solomon, M. M. (editores). Column Generation. Springer. 2005.

Hoffman, K., Padberg, M. Set Covering, Packing, and Partitioning Problems. In Encyclopedia of Optimization. Editors: C. A. Floudas, P. M. Pardalos. 2009.

Medaglia, A. L. Programación a gran escala: generación de columnas. Notas de clase. Universidad de los Andes. 2002.

Parker, R. G., Rardin, R. L. Discrete Optimization. Academic Press. 1988.

Rardin, R. L. Optimization in Operations Research. Pearson. 2016.

Restrepo, M. I., Lozano, L., Medaglia, A. L. (2012). Constrained network-based column generation for the multi-activity shift scheduling problem. International Journal of Production Economics. 140(1):466-472.