UNIVERSIDA D TECNOLOGICA NACIONAL
FACULTAD REGIONAL HAEDO

ASIGNACIÓN

Romina Miccige – Matias Patterlini

ASIGNACIÓN:

El problema de asignación es un tipo especial de problema de programación lineal en el que los asignados son recursos que se destinan a la realización de tareas. Por ejemplo, los asignados pueden ser empleados a quienes se tiene que dar trabajo. La asignación de personas a trabajos es una aplicación común del problema de asignación. Sin embargo, los asignados no tienen que ser personas. También pueden ser máquinas, vehículos o plantas, o incluso periodos a los que se asignan tareas. El primero de los siguientes ejemplos se refiere a máquinas asignadas a lugares, de manera que la tarea en este caso se trata sólo de tener una máquina. Un ejemplo posterior se refiere a plantas a las que se asignan productos que deben fabricar.

Para que se ajuste a la definición de un problema de asignación, es necesario que este tipo de aplicaciones se formule de manera tal que se cumplan los siguientes supuestos.

1. El número de asignados es igual al número de tareas. (Este número se denota por n.)
2. A cada asignado se le asigna sólo *una* tarea.
3. Cada tarea debe realizarla sólo *un* asignado.
4. Existe un costo c_{ij} asociado con el asignado i ($i = 1, 2, \ldots, n$) que realiza la tarea j ($j = 1, 2, \ldots, n$).
5. El objetivo es determinar cómo deben hacerse las n asignaciones para minimizar los costos totales.

Cualquier problema que satisface todos estos supuestos se puede resolver en forma muy eficiente mediante los algoritmos diseñados de manera especial para los problemas de asignación.

Los primeros tres supuestos son bastante restrictivos. Muchas aplicaciones potenciales no las satisfacen por completo. Con frecuencia es posible reformular el problema para hacerlo que se ajuste. Por ejemplo, muchas veces se pueden usar asignados ficticios o tareas ficticias con este fin. En los ejemplos se ilustran estas técnicas de formulación.

La JOB SHOP COMPANY compró tres máquinas nuevas de diferentes tipos. Existen cuatro sitios disponibles dentro del taller en donde se podría instalar una máquina. Algunos de ellos son más adecuados que otros para ciertas máquinas en particular por su cercanía a los centros de trabajo que tendrían un flujo intenso de trabajo hacia y desde estas máquinas. (No habrá flujo de trabajos entre las nuevas máquinas.) Por tanto, el objetivo es asignar las nuevas máquinas a los lugares disponibles de manera que se minimice el costo total del manejo de materiales. En la tabla 1 se proporciona el costo estimado por unidad de tiempo del manejo de los materiales en cuestión, con cada una de las máquinas en los sitios respectivos. El lugar 2 no se considera adecuado para la máquina 2 por lo que no se proporciona un costo para este caso.

Para formular éste como un problema de asignación, debe introducirse una máquina ficticia en el lugar adicional. Además, debe asignarse un costo muy grande M a la asignación de la máquina 2 en el lugar 2 para evitarla en la solución óptima. En la tabla 2 se muestra la tabla de costos que resulta para este problema de asignación. Esta tabla de costos contiene todos los datos necesarios para resolver el problema. La solución óptima es asignar la máquina 1 al lugar 4, la máquina 2 al lugar 3 y la máquina 3 al lugar 1 con un costo total de $29 por hora. La máquina ficticia se asigna al lugar 2, con lo que esta posición queda disponible para alguna asignación de máquina real futura.

Más adelante se discutirá cómo se obtiene esta solución después de formular el modelo matemático para el problema general de asignación.

Modelo del problema de asignación

El modelo matemático para manejar el problema de asignación utiliza las siguientes variables de decisión:

$$\text{Minimizar} \qquad Z = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij},$$

■ **TABLA 1** Costos de manejo de materiales de la Job Shop Co. ($)

		Localidad			
		1	**2**	**3**	**4**
	1	13	16	12	11
Máquina	2	15	—	13	20
	3	5	7	10	6

■ **TABLA 2** Tabla de costos del problema de asignación de la Job Shop Co.

		Tarea (localidad)			
		1	**2**	**3**	**4**
	1	13	16	12	11
Asignado	2	15	*M*	13	20
(máquina)	3	5	7	10	6
	4(*D*)	0	0	0	0

sujeta a

$$\sum_{j=1}^{n} x_{ij} = 1 \qquad \text{para } i = 1, 2, \ldots, n,$$

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad \text{para } j = 1, 2, \ldots, n,$$

y

$$x_{ij} \geq 0, \qquad \text{para toda } i \text{ y } j$$
$$(x_{ij} \text{ binaria}, \qquad \text{para toda } i \text{ y } j).$$

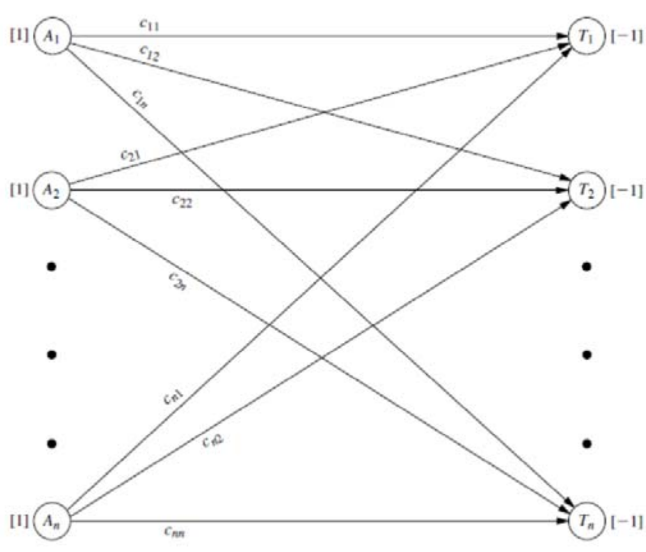
El primer conjunto de restricciones funcionales especifica que cada asignado realice sólo una asignación, mientras que el segundo conjunto requiere que cada asignación sea realizada sólo por un asignado. Si se elimina la restricción entre paréntesis de que xij sean binarias, resulta claro que el modelo es un tipo especial de problema de programación lineal, por lo que se puede resolver de inmediato. Por fortuna, debido a las razones que se expondrán en seguida, se puede eliminar esta restricción.

En realidad, el problema de asignación es sólo un caso especial de los problemas de transporte en donde los orígenes son ahora los asignados y los destinos son las asignaciones o tareas y donde

Número de orígenes m = número de destinos n,
Cada recurso $s_i = 1$,
Cada demanda $d_j = 1$.

Como ahora toda si y dj son enteros (5 1), esta propiedad significa que toda solución BF (incluso la óptima) es una solución entera para un problema de asignación. Las restricciones funcionales del modelo de asignación evitan que las variables sean mayores que 1, y las restricciones de no negatividad impiden que existan valores menores que cero. Por tanto, si se elimina la restricción binaria para poder resolver el problema de asignación como un problema de programación lineal, las soluciones BF que se obtienen (incluso la solución óptima final) satisfarán en forma automática la restricción binaria.

Igual que el problema de transporte se puede representar con una red, el problema de asignación puede describirse de manera similar. En este caso la primera columna enumera los n asignados y la segunda las n tareas. Los números entre corchetes indican el número de asignados que se proporcionan en ese lugar de la red, por lo cual los valores de la izquierda son 1 de manera automática, mientras que los valores de -1 de la derecha indican que cada tarea utiliza un asignado.



Por lo general, los que aplican esta técnica a problemas de asignación específicos no se toman la molestia de escribir todo el modelo matemático. Es mucho más sencillo formularlo en una tabla de costos (como la tabla2), con la identificación de los asignados y las asignaciones, puesto que contiene todos los datos esenciales en una forma mucho más compacta.

Algunas veces surgen problemas que no se ajustan del todo al modelo de un problema de asignación, porque ciertos asignados realizarán más de una tarea. En este caso, el problema se puede reformular para ajustarse al modelo al dividir cada uno de esos asignados en asignados nuevos separados (pero idénticos) donde cada nuevo asignado realizará sólo una tarea. (En la tabla 3 se ilustrará este asunto para un ejemplo subsecuente.) De manera similar, si una tarea será realizada por asignados múltiples, ésta se puede dividir en tareas nuevas separadas (pero idénticas) donde cada nueva tarea será desarrollada sólo por un asignado de acuerdo con el modelo reformulado. En la sección Worked Examples en el sitio de internet de este libro se proporciona un ejemplo que ilustra ambos casos y la reformulación para ajustarse al modelo de un problema de asignación. También se muestra una formulación alternativa como un problema de transporte.

TABLA 8.5 Tabla de costos para formular el problema de asignación de la opción 2 del problema de la Better Products Co.

		Tarea (producto)				
		1	**2**	**3**	**4**	**5(*D*)**
Asignado (planta)	1*a*	820	810	840	960	0
	1*b*	820	810	840	960	0
	2*a*	800	870	*M*	920	0
	2*b*	800	870	*M*	920	0
	3	740	900	810	840	*M*

Procedimientos de solución de problemas de asignación

El algoritmo opera en forma directa sobre la tabla de costos del problema. Con mayor precisión, convierte la tabla de costos original en una serie de tablas de costos equivalentes hasta llegar a una en la cual la solución óptima sea obvia. Esta tabla de costos equivalente consiste sólo en elementos positivos o iguales a cero donde todas las asignaciones se pueden hacer a las posiciones de los elementos con valor cero. Como el costo total no puede ser negativo, resulta claro que este conjunto de asignaciones con un costo total de cero es óptimo. La pregunta que resta es cómo convertir la tabla de costos original a esta forma.

La clave de esta conversión es la posibilidad de sumar o restar cualquier constante de cualquier elemento de un renglón o columna de la tabla de costos sin cambiar en realidad el problema, esto es, una solución óptima para la nueva tabla de costos también debe ser óptima para la antigua, y a la inversa.

Por tanto, el algoritmo comienza al restar el número más pequeño de cada renglón de todos los números del renglón. Este proceso de reducción de renglón creará una tabla de costos equivalente que tiene elementos iguales a cero en todos los renglones. Si esta tabla de costos tiene algunas columnas sin un elemento cero, el siguiente paso es realizar un proceso de reducción de columna al restar el número más pequeño de cada una de tales columnas del resto de los números de la columna. La nueva tabla de costos equivalente tendrá un elemento cero en todos los renglones y columnas. Si los elementos cero proporcionan un conjunto completo de asignaciones, estas últimas constituyen una solución óptima y el algoritmo termina.

En realidad, los renglones y las columnas individuales se pueden reducir en cualquier orden, pero si se inicia con todos los renglones y después se continúa con todas las columnas se encuentra una forma sistemática de ejecutar el algoritmo.

A manera de ilustración, considere la tabla de costos del problema de la Job Shop Co., que se proporcionó en la tabla 2. Para convertir esta tabla de costos en una tabla de costos equivalente, suponga que se comienza el proceso de reducción de renglón al restar 11 de cada elemento del renglón 1, de donde se obtiene

	1	2	3	4
1	2	5	1	0
2	15	*M*	13	20
3	5	7	10	6
4(*D*)	0	0	0	0

Como cualquier solución factible debe tener exactamente una asignación en el renglón 1, el costo total de la nueva tabla siempre será exactamente 11 menos que el de la tabla antigua. Por tanto, la solución que minimiza el costo total de una tabla debe también minimizar el costo total de la otra.

Observe que mientras la tabla de costos original tenía sólo elementos estrictamente positivos en los primeros tres renglones, la nueva tabla tiene un elemento cero en el renglón 1. Como el objetivo es obtener elementos cero colocados en forma tan estratégica como para producir un conjunto completo de asignaciones, este proceso debería continuar en los otros renglones y columnas. Se deben evitar los elementos negativos, de manera que la constante que será restada debe ser el elemento mínimo del renglón o columna. Si se aplica este procedimiento a los renglones 2 y 3 se obtiene la siguiente tabla de costos equivalente:

	1	2	3	4
1	2	5	1	$\boxed{0}$
2	2	*M*	$\boxed{0}$	7
3	$\boxed{0}$	2	5	1
4(*D*)	0	$\boxed{0}$	0	0

Esta tabla de costos tiene todos los elementos cero que se requieren de un conjunto completo de asignaciones, como lo indican los cuatro recuadros, de manera que estas cuatro asignaciones constituyen una solución óptima El costo total de esta solución óptima, Z = 29, el cual es exactamente la suma de los números que se han restado de los renglones 1, 2 y 3.

Desafortunadamente, una solución óptima no siempre se obtiene con tanta facilidad, como se ejemplifica a continuación con la formulación del problema de asignación para la opción 2 de la Better Products Co., que se muestra en la tabla.

Como esta tabla de costos del problema tiene elementos cero en todos los renglones, suponga que se inicia el proceso de convertirla en tablas de costos equivalentes mediante la resta del elemento mínimo de cada columna de entrada. El resultado se muestra a continuación.

	1	2	3	4	5(*D*)
1*a*	80	0	30	120	0
1*b*	80	0	30	120	0
2*a*	60	60	*M*	80	0
2*b*	60	60	*M*	80	0
3	0	90	0	0	*M*

Ahora todos los renglones y columnas tienen al menos un elemento cero, pero esta vez no es posible un conjunto completo de asignaciones con elementos de este tipo. En realidad, el número máximo de asignaciones que puede hacerse en posiciones de elementos cero es de sólo 3. (Intente hacerlo.) Por tanto, se debe implantar una idea más compleja para terminar de resolver este problema, la cual no se requirió en el primer ejemplo.

Creación de elementos cero adicionales

Esta idea implica una nueva forma de crear posiciones adicionales con elementos cero sin crear ningún elemento negativo. En lugar de restar una constante a un solo renglón o columna, ahora se suma o se resta una constante a una combinación de renglones y columnas.

Este procedimiento se inicia dibujando un conjunto de líneas a través de algunos de los renglones y columnas de tal forma que se cubran todos los ceros. Es preferible que este procedimiento se haga con el número mínimo de líneas, como se muestra en la siguiente tabla de costos.

	1	2	3	4	5(*D*)
1*a*	80	0	30	120	0
1*b*	80	0	30	120	0
2*a*	60	60	*M*	80	0
2*b*	60	60	*M*	80	0
3	0	90	0	0	*M*

Observe que el elemento mínimo que no ha sido cruzado es 30, en las dos posiciones superiores de la columna 3. Por tanto, si se resta 30 de todos los elementos de la tabla completa, es decir, de cada renglón y columna, se creará un nuevo elemento cero en estas dos posiciones. Después, para restaurar los elementos cero previos y eliminar elementos negativos, se suma 30 a cada renglón y columna que esté cruzada por una línea, esto es, renglón 3 y columnas 2 y 5(D). De aquí se obtiene la siguiente tabla de costos equivalente.

	1	2	3	4	5(*D*)
1*a*	50	0	0	90	0
1*b*	50	0	0	90	0
2*a*	30	60	*M*	50	0
2*b*	30	60	*M*	50	0
3	0	120	0	0	*M*

Un atajo para obtener esta tabla de costos a partir de la anterior es restar 30 sólo de los elementos que no están cruzados por una línea y después sumar 30 a cada elemento que se encuentre en la intersección de dos líneas.

Observe que las columnas 1 y 4 de esta nueva tabla de costos tienen sólo un elemento cero y ambos están en el mismo renglón (renglón 3). En consecuencia, ahora es posible hacer cuatro asignaciones a posiciones de elementos cero, pero todavía no a cinco. (Intente hacerlo.) Por tanto, se repite el procedimiento anterior, donde ahora el número mínimo de líneas para cubrir todos los ceros es cuatro: la misma cantidad que el número máximo de asignaciones. Una forma de hacerlo es como se muestra a continuación.

	1	2	3	4	5(*D*)
1*a*	50	0	0	90	0
1*b*	50	0	0	90	0
2*a*	30	60	*M*	50	0
2*b*	30	60	*M*	50	0
3	0	120	0	0	*M*

De nuevo, el elemento mínimo no cubierto por una línea es 30, número que ahora aparece en la primera posición de los renglones 2a y 2b. Por tanto, se resta 30 de todos los elementos no cubiertos y se suma 30 a todos los elementos cubiertos dos veces —sólo se pasan por alto los elementos de M—, lo cual proporciona la siguiente tabla de costos equivalente.

	1	2	3	4	5(*D*)
1*a*	50	$\boxed{0}$	0	90	30
1*b*	50	0	$\boxed{0}$	90	30
2*a*	$\boxed{0}$	30	*M*	20	0
2*b*	0	30	*M*	20	$\boxed{0}$
3	0	120	0	$\boxed{0}$	*M*

En realidad, con esta tabla existen algunas formas de hacer un conjunto completo de asignaciones a las posiciones de elementos cero (varias soluciones óptimas), incluso las que indican los cinco cuadros. El costo total resultante Z + 810 + 840 + 800 + 0+ 840 = 3 290

A continuación se resume el algoritmo completo que se acaba de ejemplificar.

1. Reste el número más pequeño de cada renglón a cada número del renglón. (Esto se llama reducción de renglón.) Introduzca los resultados en una nueva tabla.
2. Reste el número más pequeño de cada columna de la nueva tabla a cada número de la columna. (Esto se llama reducción de columna.) Introduzca los resultados en otra tabla.
3. Pruebe si se puede hacer una asignación óptima. Hágalo mediante la determinación del número mínimo de líneas necesario para cubrir (es decir, cruzar) todos los ceros. Puesto que este número mínimo de líneas es igual al número máximo de tareas que pueden realizarse en cero posiciones de elementos, si el número mínimo de líneas es igual al número de renglones, es posible tener un conjunto óptimo de tareas. (Si usted encuentra que no es posible un conjunto completo de tareas, esto significa que usted no redujo el número de líneas que cubren a todos los ceros hasta el número mínimo.) En ese caso vaya al paso 6. En caso contrario continúe con el paso 4.
4. Si el número de líneas es menor que el número de renglones, modifique la tabla de la siguiente forma
   a) Reste el número no cubierto más pequeño de todos los números no cubiertos de la tabla.
   b) Sume el número no cubierto más pequeño a los números que se encuentran en las intersecciones de las líneas.
   c) Los números cruzados pero que no se encuentran en las intersecciones de las líneas permanecen sin cambio en la siguiente tabla.
5. Repita los pasos 3 y 4 hasta que sea posible tener un conjunto de asignaciones óptimo.
6. Haga las asignaciones una a una en las posiciones que tienen elementos cero. Comience con los renglones y columnas que tienen sólo un cero. Como cada renglón y cada columna necesita recibir exactamente una asignación, cruce tanto el renglón como la columna involucrados luego de hacer cada asignación. Después continúe con los renglones y columnas que aún no han sido cruzados para seleccionar la siguiente asignación, y de nuevo dé preferencia a algún renglón o columna que tenga un solo cero que no haya sido cruzado. Continúe hasta que todos los renglones y columnas tengan exactamente una asignación y por ende ya hayan sido cruzados.