# UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

DIVISION DE ESTUDIOS DE POSTGRADO
DE LA FACULTAD DE INGENIERIA

"UN ALGORITMO PARA RESOLVER EL PROBLEMA DE LOCALIZACION DE SERVICIOS SIMPLE"

# T E S I S

QUE PARA OBTENER EL GRADO DE:

**MAESTRA EN INGENIERIA ESPECIALIDAD INVESTIGACION DE OPERACIONES**

P R E S E N T A :

**HERICA SANCHEZ LARIOS**

DIRECTORES DR RICARDO ACEVES GARCIA
Y M EN I ELI DE LA TORRE VEGA

MEXICO, D. F. DICIEMBRE, 1998

## Agradecimientos


- Quiero agradecer en especial a mis directores de tesis: Dr. Ricardo Aceves García y M. en I. Eli de la Torre Vega por sus asesorías que me dieron y tiempo que me dedicaron para que yo lograra realizar mi tesis.

- También agradezco a todos mis profesores por los conocimientos que adquirí con sus enseñanzas en la maestría.

- Agradezco a mis padres, Angel Sánchez Colmenares y Elízabeth Larios Mendoza por todo el apoyo que me han dado.

- Y agradezco a todas las personas que me ayudaron en algún momento para la realización de esta tesis.

# ÍNDICE

I. INTRODUCCIÓN ... 1

II. PROBLEMAS PROTOTIPO DE LOCALIZACIÓN DISCRETA SIN LÍMITES DE CAPACIDAD ... 4

2.1 Introducción ... 4

2.2 Conceptos Básicos de Redes ... 5

2.3 Problemas *p-Median*: *p-MP* ... 7

2.4 Problemas *p-Center*: *p-CP* ... 10

2.5 Problemas de Localización de Servicios sin Restricciones de Capacidad (Problema de Localización de Servicios Simple): *PLSS* ... 12

2.5.1 Problemas de Cobertura de Conjuntos (*PCC*) ... 13

2.5.2 Formulaciones Matemáticas de los Problemas *p-MP*, *p-CP*, y *PLSS* ... 16

2.5.3 El *p-* Problema de Localización de Servicios Simple: *p-PLSS* ... 18

2.6 El Problema de Asignación Cuadrática: *PAC* ... 20

III. COMPLEJIDAD COMPUTACIONAL ... 24

IV. FORMULACIÓN DEL PROBLEMA DE LOCALIZACIÓN DE SERVICIOS SIMPLE Y SU FACTIBILIDAD ... 28

V. PROPIEDADES DE LA DUALIDAD EN EL PROBLEMA DE LOCALIZACIÓN DE SERVICIOS SIMPLE ... 34

5.1 Cota Inferior y Superior de la Solución del Problema de Localización de Servicios Simple ... 34

5.2 Solución Óptima del Problema de Localización de Servicios Simple ... 36

VI. SOLUCIÓN DEL PROBLEMA DE LOCALIZACIÓN SIMPLE CON ALGUNOS ALGORITMOS EXISTENTES ... 37

6.1 Algoritmo de Partición (Descomposición de Benders) ... 37

6.1.1 Solución del Programa Lineal Dual....................39
6.2 Ejemplo Ilustrativo Resuelto por el Método de Benders....................41
6.3 Método de Ramificación y Acotamiento....................44
6.4 Ejemplo Ilustrativo Resuelto por el Método de Ramificación y Acotamiento....................45

VII. DESCRIPCIÓN DEL ALGORITMO PROPUESTO....................47
7.1 Nivelación Dual....................47
7.1.1 Conservación de la Factibilidad Dual en el Procedimiento de Nivelación Dual....................52
7.2 Pulido del Primal....................54
7.3 Ramificación por Selección....................59

VIII. EXPERIENCIA COMPUTACIONAL....................63
8.1 Ejemplo Comparativo....................63
8.2 Tiempo Computacional....................66

IX. CONCLUSIONES Y TRABAJOS FUTUROS....................68

APÉNDICE A....................70
A.1 Un Ejemplo del Problema de Localización de Servicios Simple Resuelto por el Método de Partición de Benders....................70

BIBLIOGRAFÍA....................81

# CAPÍTULO I. INTRODUCCIÓN

Los Problemas de Localización se presentan cuando los encargados de tomar decisiones deben seleccionar el sitio en que se ubicarán una o varias instalaciones. Por instalación debe entenderse cualquier servicio público o privado. Como ejemplos de tales servicios tenemos: hospitales, estaciones de policía y bomberos, bancos, oficinas, escuelas, bibliotecas, estadios deportivos, estaciones de gasolina, plantas de tratamiento de aguas negras y de basura, aeropuertos, parques, plantas industriales, centros comerciales, mercados, etc.

Una forma de clasificar los problemas de localización es en términos de la disponibilidad de sitios opcionales que se consideran como candidatos para ubicar las nuevas instalaciones. Si hay una lista de sitios específicos en consideración, el problema se denomina problema de *espacio discreto*. Por otra parte, si las nuevas instalaciones se pueden situar en cualquier parte dentro de un área, el problema se clasifica como uno de *espacio continuo*. Por ejemplo, si tratamos de reubicar el centro de distribución de PEMEX ubicado en Azcapotzalco y la administración ha indicado que se debe trasladar el centro ya sea a Pachuca, Salamanca o Toluca, entonces se está tratando con un *problema de localización discreto*, y en este caso se está tratando de una sola instalación. Pero si la administración desea considerar *todos* los puntos posibles del centro del País en lugar de dar tres sitios específicos, entonces se trata de un *problema de localización de espacio continuo* y también de una sola instalación.

Otra forma de clasificar a los problemas de localización es en términos de la cantidad de servicios a localizar. Un problema de localización se clasifica como problema de una sola instalación si implica la determinación del sitio para una sola instalación. Si el problema incluye la selección simultanea de sitios de localización para varias instalaciones, se denomina problema de localización de instalaciones múltiples.

Y por último, podemos hacer una clasificación de los problemas de localización en términos de las *capacidades* de los servicios. Si la capacidad de cada servicio es *ilimitada*, entonces un sólo servicio es sufuciente para satisfacer la demanda total. Cuando se tienen capacidades ilimitadas, se trata de Problemas de Localización de Servicios sin Restricciones de Capacidad. Por otro lado, si la capacidad de cada servicio es *limitada*, entonces puede ocurrir que un solo servicio no pueda satisfacer toda la demanda de los clientes, en estos casos la capacidad de cada servicio es de importante consideración. A este tipo de problemas se le llama Problemas de Localización de Servicios con Restricciones de Capacidad.

El problema que se considera en este trabajo de investigación es el Problema de Localización de Servicios sin Restricciones de Capacidad, el cual está dentro de la clasificación de los problemas de espacio discreto.

En los modelos de localización discretos generalmente se tiene un balance entre simplicidad y verosimilitud[1].

Estos modelos son flexibles porque permiten elegir cualquier medida de distancia y cualquier conjunto finito de sitios como localizaciones factibles. Sin embargo, la solución de estos modelos puede ser un asunto difícil y complejo.

Existen varios algoritmos para resolver el *problema de localización de servicios sin restricciones de capacidad* o más frecuentemente llamado *Problema de Localización de Servicios Simple* (*PLSS*). El adjetivo "simple" en el *PLSS*, ahora ampliamente aceptado como sinónimo de "sin restricciones de capacidad" fue originalmente mencionado por Spielberg (1969). Y el primero en formular formalmente el *PLSS* se dice que fue Balinski en 1966.

Hay algoritmos exactos y algoritmos heurísticos para resolver el Problema de Localización de Servicios Simple. El algoritmo que se presenta en este trabajo requiere de algunas etapas heurísticas como lo son la fase I y II: Nivelación Dual y Pulido del Primal, no siendo así la fase III: la fase de Ramificación por Selección, esta última es un método exacto. Cuando las fases I y II llegan a la solución óptima, entonces ya no se requiere completar la solución del *PLSS* con la parte exacta del algoritmo.

Con el trabajo de investigación de la tesis doctoral de Ricardo Aceves García [1], surgió el motivo de probar si funcionaba en el Problema de Localización de Servicios sin Restricciones de Capacidad su forma de repartir el costo fijo a los clientes. Al ir desarrollando esta prueba se vio que se generaban algunas complicaciones, las cuales se podían evitar si se cambiaba esa forma de repartir los costos fijos; es así como surgió una nueva idea de repartir el costo fijo de un servicio y agregarlo al costo variable de cada cliente.

[1] Tomado de la Tesis Doctoral de Ricardo Aceves García.[1]

*El objetivo de este trabajo de investigación es resolver el Problema de Localización de Servicios Simple (PLSS) de una manera sencilla. Por lo tanto desarrollamos y probamos un método para resolver el PLSS, que a través de una "Nivelación Dual", (fase I), y un "Pulido del Primal", (Fase II), muchas veces se obtiene la solución óptima del problema. Y en algunos casos, en los cuales hay violación de las condiciones de holgura complementaria, no es suficiente la fase II aplicada al final de la nivelación dual, en estos casos se parte de los resultados obtenidos en estas dos fases para realizar la fase III, que llamamos "Ramificación por Selección". Cabe mencionar que en la mayoria de los casos, la nivelación dual nos proporciona una solución muy cercana a la solución óptima del problema, esto implica que cuando se requiere la fase de Ramificación por Selección, la búsqueda de la solución óptima es rápida.*

El contenido de este trabajo se da enseguida: En el capítulo I, se da una idea general de lo que es un Problema de Localización y su clasificación. Como el algoritmo que se desarrolla en este trabajo resuelve el Problema de Localización de Servicios Simple (PLSS), el cual es un problema que pertenece a la clasificación de los Problema de Localización Discreta, en el capítulo II se dan las familias que se consideran prototipos de los Problemas de Localización Discreta sin Límites de Capacidad. En el capítulo III, se da la complejidad computacional del problema abordado (PLSS). Para poder formalizar el problema considerado en este trabajo, en el capítulo IV se da el planteamiento matemático del problema de localización de servicios simple y su factibilidad. El algoritmo propuesto considera el dual del PLSS, es por eso que en el capítulo V se dan las propiedades de la dualidad en este problema, en este capítulo también se demuestran las condiciones de optimalidad en las que se basa el algoritmo propuesto. Para conocer algunos de los algoritmos que existen para resolver el PLSS, en el capítulo VI se presentan dos métodos conocidos que resuelven este tipo de problemas. En el capítulo VII se da la descripción del funcionamiento del algoritmo propuesto, indicando las fases que lo componen. En el capítulo VIII se recopilan los resultados que se obtuvieron de algunos de los problemas que se resolvieron con el algoritmo que se presenta en este trabajo de investigación. Y por último, en el capítulo IX se dan las conclusiones a las que se llegó con el algoritmo desarrollado en esta tesis.

# CAPÍTULO II. PROBLEMAS PROTOTIPO DE LOCALIZACIÓN DISCRETA SIN LÍMITES DE CAPACIDAD.

## 2.1 INTRODUCCIÓN

Se inicia esta parte con una explicación de algunos conceptos relacionados a redes en general, los cuales servirán para la explicación de los problemas prototipo de localización. Las familias que componen a los problemas prototipo de localización son cuatro:

a) Problemas *p-Median, p-MP*.
b) Problemas *p-Center, p-CP*.
c) Problema de Localización de Servicios sin restricciones de capacidad (Problema de Localización de Servicios Simple), *PLSS*.
d) Problema de Asignación Cuadrática, *PAC*.

Se considerarán ejemplos de los problemas *p-Median*, que incluye una versión como red y la versión "bipartita" que es más general. Siguiendo en la misma línea, se discuten los problemas *p-Center* y después se procede con el problema de localización de servicios simple (*PLSS*), el cual se parece al problema *p-Median* pero incluye otras propiedades importantes. Se bosqueja el llamado Problema de Cobertura de Conjuntos como un caso especial del *PLSS*, aunque quizá, la colección de tales problemas podría constituir por sí mismos una familia más de los problemas de localización. Las primeras tres familias: *p-Median*, *p-Center*, y *el PLSS*, comparten varias características en común, es por eso que después de describirlos a estos tres tipos de problemas, se presenta una formulación unificada de éstos. Después se presenta el *p-Problema de Localización de Servicios Simple*, el cual es un modelo híbrido que combina características del problema *p-Median* y del *PLSS*. El cuarto y último problema prototipo, es el Problema de Asignación Cuadrática (*PAC*) el cual tiene una esencia diferente a las otras tres familias ya mencionadas.

## 2.2 CONCEPTOS BÁSICOS DE REDES

La red no dirigida $N = (V, A)$ mostrada en la figura (1) consiste de un conjunto $V = (v_a, v_b, \ldots, v_h)$ de 8 nodos y un conjunto $A = ([v_a, v_b]$ ó $[v_b, v_a], [v_a, v_c]$ ó $[v_c, v_a], \ldots, [v_g, v_h]$ ó $[v_h, v_g])$ de 12 pares no necesariamente *ordenados* de nodos *distintos* en $V$. Cada par, por ejemplo $[v_a, v_b]$ ó $[v_b, v_a]$ de nodos en $A$ es un arco en $N$. Con cada arco $[v_i, v_j]$ está asociado un número real positivo $\alpha(v_i, v_j)$ llamado la *longitud* de ese arco. De este modo, $\alpha(v_d, v_h) = \alpha(v_h, v_d) = 8$ se refiere a la longitud del arco $[v_d, v_h]$, esto puede ser interpretado como "la distancia entre $v_d$ y $v_h$ es de 8 millas" o "el tiempo de recorrido entre $v_d$ y $v_h$ es de 8 minutos" o "el costo de envío por unidad desde $v_d$ hasta $v_h$ a través del arco $[v_d, v_h]$ es de 8 dólares".

Los arcos son algunas veces transitables en una sola dirección (por ejemplo calles de un solo sentido). Este caso se muestra en la figura (2).

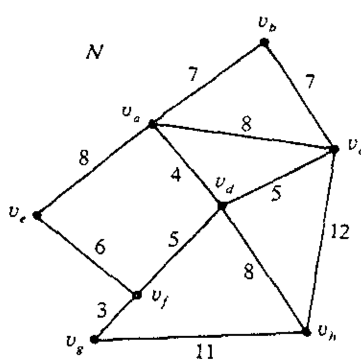



Fig. (1)

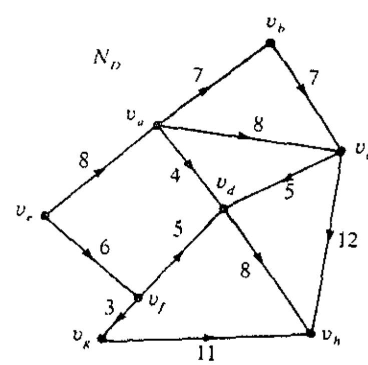



Fig. (2)

Un arco de una red dirigida es un par ordenado de distintos vértices. En esta parte de nuestro trabajo, nos interesa la red no dirigida de la figura (1) para definir algunos conceptos.

Informalmente, una ruta entre $v_c$ y $v_f$ en $N$ es una secuencia de arcos, por ejemplo $[v_c, v_a], [v_a, v_d], [v_d, v_f]$ la cual conecta a los nodos $v_c$ y $v_f$. Una notación convencional para esta ruta puede ser escrita como $(c-a-d-f)$. En general, la longitud de una ruta es la suma de las longitudes de los arcos que la constituyen; para este ejemplo particular la longitud de la ruta es igual a $\alpha(v_c,v_a)+\alpha(v_a,v_d)+\alpha(v_d,v_f)=8+4+5=17$. Tal como parece, sin embargo, hay varias rutas entre $v_c$ y $v_f$. Otra ruta es por ejemplo: $(c-h-d-f)$ de longitud 25, $(c-b-a-c-d-f)$ de longitud 32 y $(c-d-f)$ de longitud 10. La ruta $(c-b-a-c-d-f)$ incluye el *ciclo* $(c-b-a-c)$. Para un par de nodos en una red, estamos interesados principalmente en este caso, en una *ruta corta*, que es la ruta de mínima longitud que conecta a dos nodos. Las rutas *libres de ciclos* pueden proporcionar las *mejores* rutas que hay entre dos nodos. Claramente, si consideramos a todas las rutas libres de ciclos que conectan a $v_c$ y $v_f$, vemos que la ruta $(c-d-f)$ es la ruta más corta.

A la ruta más corta entre los nodos $v_i$ y $v_j$ en una red $N$, la llamaremos $P[v_i,v_j]$, su longitud es $d(v_i,v_j)$ también referida como la *distancia* entre $v_i$ y $v_j$.

Para nuestro trabajo se supondrá que todas las longitudes de los arcos son positivas; de aquí que la distancia entre dos nodos distintos será también positiva. También supondremos que no se permitirán arcos que conecten a un mismo nodo. Sin embargo, a veces es práctico considerar la distancia de un nodo a él mismo, en nuestro caso, definimos tal distancia con valor de cero.

La red de la figura (1) la presentamos ahora en la figura (3) acompañada por una matriz $D$ simétrica de (8x 8) de las distancias entre cualquiera de los $\binom{8}{2}=28$ pares de los distintos nodos en $N$.

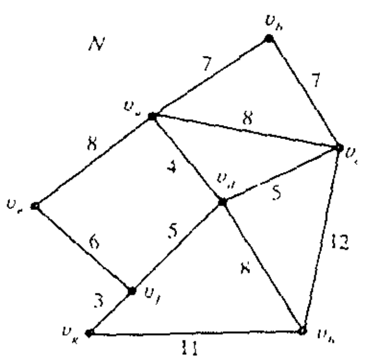



Fig. (3)

$$
\begin{array}{c|cccccccc}
 & v_a & v_b & v_c & v_d & v_e & v_f & v_g & v_h \\
\hline
v_a & 0 & 7 & 8 & 4 & 8 & 9 & 12 & 12 \\
v_b & 7 & 0 & 7 & 11 & 15 & 16 & 19 & 19 \\
v_c & 8 & 7 & 0 & 5 & 16 & 10 & 13 & 12 \\
v_d & 4 & 11 & 5 & 0 & 11 & 5 & 8 & 8 \\
v_e & 8 & 15 & 16 & 11 & 0 & 6 & 9 & 19 \\
v_f & 9 & 16 & 10 & 5 & 6 & 0 & 3 & 13 \\
v_g & 12 & 19 & 13 & 8 & 9 & 3 & 0 & 11 \\
v_h & 12 & 19 & 12 & 8 & 19 & 13 & 11 & 0
\end{array}
\; D
$$

Matriz D

## 2.3 PROBLEMAS $P-Median$: $p-MP$

Supongamos ahora que cada uno de los ocho nodos en $N$ corresponde a un cliente, y que el $j-ésimo$ cliente tiene una demanda fija de $w_j$ unidades (por periodo) para un cierto servicio, la cual es independiente de la localización (ubicación) del servicio que la satisfará. Además, supongamos que cada nodo es un lugar potencial de los servicios, los cuales pueden producir cualquier cantidad de beneficio y enviarlo a través de los caminos representados por los arcos de $N$. Todos los servicios están controlados por una sola compañía que tiene control completo sobre el número de servicios a ser abiertos y su localización.

Para obtener el primer modelo de localización discreto, partimos de una clara estructura de costos. Sea $c_{ij} = w_j d(v_i, v_j)$ el costo total incurrido si todas las unidades $w_j$ demandadas por el cliente $j$ son surtidas por el servicio $i$.

Supongamos que todos los clientes tienen una demanda unitaria, esto es que $w_j = 1$, entonces $c_{ij} = w_j d(v_i, v_j)$ es simplemente la distancia entre los nodos $v_i$ y $v_j$. Entonces la matriz de (8 x 8), $C = \{c_{ij}\}$ será idéntica a la matriz $D$ de la figura (3).

Los servicios pueden estar en cualquiera de los siguientes dos casos: *abierto* o *cerrado*. El servicio $i$ se llama cerrado si no es colocado en el lugar $v_i$ y por tanto los clientes no pueden ser atendidos desde este lugar o por este sitio. Un servicio se llama abierto en el caso contrario, y a menos que se especifique lo contrario, los servicios abiertos son de capacidad ilimitada en el sentido de que éstos pueden satisfacer incluso a todos los clientes.

Para un valor dado de $p$ especificado, el llamado problema $p-median$ establece $p$ servicios en $p$ lugares potenciales y satisfará a cada cliente desde un subconjunto de los servicios establecidos, tal que las demandas de todos los clientes sean satisfechas y tal que los costos totales incurridos sean minimizados.

El problema $p-median$ es una formulación prototipo que refleja a muchos problemas reales de localización. Frecuentemente es empleado como punto de partida en casos donde, por ejemplo, debido a otras consideraciones de estrategia, se debe decidir establecer un número predeterminado de nuevos servicios para satisfacer nuevas demandas o para modificar una estructura de servicios existentes, tal que se tenga $p$ nuevos (y/o restablecidos) servicios.

Aunque el requerimiento "exactamente $p$ servicios" puede ser algo riguroso, en aplicaciones prácticas $p$ es frecuentemente considerado como un parámetro, permitiéndose insignificantes variaciones. Solamente los nodos de $N$ se consideran como lugares potenciales para los servicios a establecer.

El $p-MP$ es también un ejemplo de un *problema tipo*, esto es, un ejemplo de una pregunta general a ser respondida, usualmente en términos de la asignación de valores específicos a un conjunto de variables tales que la solución resultante satisfaga ciertas propiedades. Los algoritmos para resolver un problema bien definido pueden ser propuestos sin tener en cuenta los valores específicos de los parámetros del problema; para el caso de los $p-MP$, éstos dependen de $p$, del número de renglones y columnas de $C$ y de cada $c_{ij}$ individual. Sin embargo, si todos los valores de los parámetros se especifican, entonces hablamos de una *instancia de datos* o mejor dicho de una *instancia* del correspondiente problema.

Supongamos que $p=3$ y que estos tres servicios han sido localizados en los nodos $v_b, v_g$ y $v_h$, respectivamente. Consideremos un cliente localizado en $v_f$. El costo total por satisfacer la demanda del cliente $f$ desde cada servicio será: 16, 3 y 13 respectivamente. En términos de costos, ningún beneficio se obtiene si la demanda se satisface por dos o más servicios, la elección óptima es simplemente satisfacer toda la demanda de $f$ por un solo servicio (ya que éstos tienen capacidad ilimitada) en este caso, el servicio localizado en $g$ sería el de mejor costo. Para un cliente localizado en $v_d$, los costos serían 11, 8 y 8 respectivamente. En este caso el costo por satisfacer al cliente es de 8, el cual daría lo mismo que entre $v_g$ y $v_h$ lo satisfagan o exclusivamente $v_g$ o $v_h$. Pero nuevamente caemos en lo mismo, nada se ganaría por que más de un solo servicio satisfaga al cliente, si este servicio tiene capacidad ilimitada, lo cual implica que él solo puede satisfacer la demanda del cliente ubicado en $v_d$. Sin embargo, esta observación podría parecer determinante. El hecho de que solamente se tienen que considerar soluciones donde cada cliente sea surtido por un solo servicio, conlleva a una significante regla para la apreciación del problema, así como para los algoritmos desarrollados para su solución. En este contexto, es usual decir que los $p-MP$ poseen la *propiedad de una sola asignación.*

Para una matriz dada de costos, $C$ y especificando el número $p$ de servicios a establecer, tenemos una instancia de datos completa de un $p-MP$. Para $p=1$ y con $C$ tomada como la matriz de costos dada en la figura (3), el costo total por atender a todos los clientes desde un solo servicio es igual a la suma de todos los $c_{ij}$ correspondientes al renglón. Estos costos se dan en la tabla1.

| Servicio localizado en | $v_a$ | $v_b$ | $v_c$ | $v_d$ | $v_e$ | $v_f$ | $v_g$ | $v_h$ |
|---|---|---|---|---|---|---|---|---|
| Costo Total | 60 | 94 | 71 | 52 | 84 | 62 | 75 | 94 |

Tabla 1

Así, una solución óptima para el $1-MP$ (o para el $p-MP, p=1$) es ubicar un servicio en el nodo $v_d$. Este nodo es entonces el *"median"* (nodo activo) de la red $N$. Para $p=2$ existen $\binom{8}{2}=28$ distintos pares de nodos donde dos servicios pueden ser localizados. Supóngase ahora que los servicios están establecidos en los nodos $v_c$ y $v_d$. Para minimizar el costo total, ¿cómo deberán ser arreglados los envíos de los servicios hacia los ocho clientes?.

La *propiedad de una sola asignación* nos da la respuesta: cada cliente es surtido solamente desde un servicio abierto, donde el costo $c_{ij}$ correspondiente sea el más pequeño. En otras palabras, consideramos solamente una submatriz de $C$ la cual consiste de los renglones $v_c$ y $v_d$ y escoger el mínimo de cada columna, dando los siguientes costos que se muestran en la tabla 2.

| Servicio | Cliente ubicado en | | | | | | | | Costo Total |
|---|---|---|---|---|---|---|---|---|---|
| | $v_a$ | $v_b$ | $v_c$ | $v_d$ | $v_e$ | $v_f$ | $v_g$ | $v_h$ | |
| $v_c$ | 8 | 7 | 0 | 5 | 16 | 10 | 13 | 12 | |
| $v_d$ | 4 | 11 | 5 | 0 | 11 | 5 | 8 | 8 | |
| Distancia al servicio más cercano | 4 | 7 | 0 | 0 | 11 | 5 | 8 | 8 | 43 |

Tabla 2.

Mientras que el ejemplo del $1-MP$ fue resuelta óptimamente por numeración completa (todas las soluciones factibles se listaron y el correspondiente costo total fue evaluado), un enfoque similar para el $2-MP$ involucrará la evaluación y comparación de 28 soluciones factibles.

## 2.4 PROBLEMAS $P-Center$ ( $p-CP$)

El problema $p-center$, difiere significativamente del $p-MP$ en varios aspectos, principalmente con respecto al criterio utilizado para valorar la calidad de una solución factible. Mientras que el $p-MP$ es un problema de *suma mínima*, el $p-CP$ tiene una función objetivo *minimax*: abrir $p$ servicios y asignar cada cliente a exactamente uno de ellos, tal que la distancia máxima desde cualquier servicio a cualquiera de los clientes asignados sea mínima. Tal criterio minimax en la función objetivo, ocurre frecuentemente en problemas de localización de servicios de emergencia, tales como estaciones de bomberos, policías y ambulancias. Un criterio común para la eficacia en la cobertura de tales servicios, es que cualquier punto de demanda puede ser alcanzado desde el servicio más cercano dentro de una distancia dada, tiempo o costo.

Un factor adicional que distingue al $p-CP$ del $p-MP$ es que el nodo con propiedades de optimalidad no es precisamente obligatorio. Por ejemplo, si un solo servicio es para atender a dos clientes de igual peso y si el conjunto de localización no se restringe a cualquiera de los dos nodos que representan a esos clientes, sino que también se incluye cualquier punto sobre el arco que los conecta, entonces, el *center* veríamos obviamente que es el *punto medio de ese arco*. Por simplicidad sin embargo, restringimos en esta parte del estudio, que el conjunto de localización sea cualquiera de los nodos de la red considerada o uno de los conjuntos de nodos definidos por una red bipartita. Además, para facilitar la comparación de las soluciones del $p-CP$ y $p-MP$, regresamos a la misma serie de ejemplos que se utilizaron en la descripción de los problemas $p-MP$.

Consideremos la instancia del $p-CP$ definida por $p=1$, $w_a = w_b = ... = w_h = 1$ y con la matriz $D$ de distancias como se mostró en la figura (3), para un solo servicio localizado en algún nodo de $N$, la máxima distancia que genera una solución factible se obtiene encontrando el elemento más grande en el correspondiente renglón de $D$. Entonces se obtienen las distancias máximas como se muestra en la tabla 3.

| Servicio localizado en: | $v_a$ | $v_b$ | $v_c$ | $v_d$ | $v_e$ | $v_f$ | $v_g$ | $v_h$ |
|---|---|---|---|---|---|---|---|---|
| Distancia máxima | 12 | 19 | 16 | **11** | 19 | 16 | 19 | 19 |

Tabla 3.

El *center* es el nodo $v_d$, el cual por coincidencia fue también el *median*. Para la colocación de dos servicios hay 28 posibilidades distintas. Si por ejemplo, el conjunto de nodos seleccionados es $\{v_f, v_g\}$, para calcular la máxima distancia consideramos los renglones correspondientes, tomamos el más pequeño en cada columna y registramos el más grande como tal distancia, esto se muestra en la tabla 4.

| | $v_a$ | $v_b$ | $v_c$ | $v_d$ | $v_e$ | $v_f$ | $v_g$ | $v_h$ | |
|---|---|---|---|---|---|---|---|---|---|
| $v_f$ | **9** | **16** | **10** | **5** | **6** | **0** | 3 | 13 | |
| $v_g$ | 12 | 19 | 13 | 8 | 9 | 3 | **0** | **11** | |
| Distancia al servicio más cercano | 9 | <u>16</u> | 10 | 5 | 6 | 0 | 0 | 13 | Máxima distancia=**16** |

Tabla 4.

Por lo tanto el servicio se localizará en $v_b$. Si todas las 28 soluciones factibles se consideraran, encontraremos que el conjunto del $2-center$ es $\{v_a, v_d\}$ con $d(v_a,v_e) = d(v_d,v_g) = d(v_d,v_h) = 8$ como la distancia minimax. El único tipo de algoritmo utilizado hasta aquí para resolver instancias específicas del $p-MP$ o $p-CP$, ha sido enumeración completa; es decir, evaluar todas las soluciones factibles y escoger la mejor.

## 2.5 PROBLEMAS DE LOCALIZACIÓN DE SERVICIOS SIN RESTRICCIONES DE CAPACIDAD (PROBLEMA DE LOCALIZACIÓN DE SERVICIOS SIMPLE): PLSS.

Sea $I = \{1,...,m\}$ un conjunto finito de $m$ posibles lugares para establecer servicios. El PLSS trata de satisfacer la demanda de un conjunto de clientes $J = \{1,...n\}$ con la capacidad u oferta de un subconjunto de esos servicios potenciales, ya sea desde un solo servicio o varios sin que más de un solo servicio satisfaga a un cliente, esto se debe a que el PLSS tiene la propiedad de una sola asignación. Los servicios se suponen que tienen capacidad ilimitada tal que en principio cualquier servicio puede satisfacer todas las demandas. Dados los costos asociados con los servicios y los costos de transporte por enviar desde los servicios hasta los clientes, buscamos un plan de mínimo costo de producción y transporte (en términos del número de servicios establecidos, sus localizaciones, y la cantidad enviada desde cada servicio hasta cada cliente) satisfaciendo todas las demandas.

Dos aspectos distinguen al PLSS del $p-MP$: (1) un *costo fijo* no negativo está asociado con cada servicio potencial y se incurre en este costo fijo únicamente si algún servicio es establecido, y (2) el número de servicios a establecer *no* está preestablecido.

El PLSS puede verse como un problema de localización para el cual la red subyacente es bipartita. La red bipartita se muestra en la figura (4).

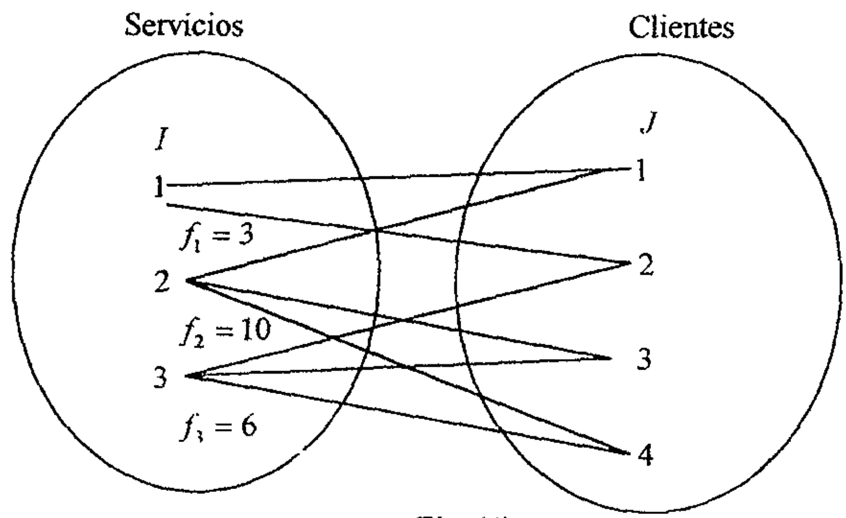



Fig. (4)

Con la matriz de costos, $C$, dada por:

$$C = \begin{bmatrix} 7 & 2 & \infty & \infty \\ 5 & \infty & 4 & 0 \\ \infty & 0 & 0 & 2 \end{bmatrix}$$

Si los servicios abiertos están en $\{1,2\}$ del conjunto $I$, entonces el costo fijo será $f_1 + f_2 = 3 + 10 = 13$. Entonces los clientes 1, 3 y 4 serán asignados al servicio 2 con un costo de $c_{21} + c_{23} + c_{24} = 5 + 4 + 0 = 9$ mientras que el cliente 2 será atendido por el servicio 1 con un costo de $c_{12} = 2$. El costo total, dado por la suma de los costos variables $c_{ij}$ más los costos fijos $f_i$, es: $(9+2)+13 = 24$. Existen $2^m - 1 = 2^3 - 1 = 7$ diferentes combinaciones de abrir y cerrar servicios, donde $m$ es el número de servicios.

Calculando los costos totales para cada una de estas combinaciones, se obtienen los siguientes resultados:

| Servicios abiertos | Costos fijos, $f_i$ | Costos variables, $c_{ij}$ | Costos totales |
|---|---|---|---|
| $\{1\}$ | 3 | $\infty$ | $\infty$ |
| $\{2\}$ | 10 | $\infty$ | $\infty$ |
| $\{3\}$ | 6 | $\infty$ | $\infty$ |
| $\{1,2\}$ | 3+10 | 5+2+4+0 | 24 |
| $\{1,3\}$ | 3+6 | 7+0+0+2 | **18** |
| $\{2,3\}$ | 10+6 | 5+0+0+0 | 21 |
| $\{1,2,3\}$ | 3+10+6 | 5+0+0+0 | 24 |

Tabla 5.

La solución óptima es 18, por lo tanto se requiere establecer los servicios en los lugares $\{1,3\}$

## 2.5.1 PROBLEMAS DE COBERTURA DE CONJUNTOS, *(PCC)*

Como ya se mencionó en la introducción de este capítulo, el problema de Cobertura de Conjuntos es un caso especial del Problema de Localización de Servicios Simple.

En el problema de Cobertura de Conjuntos se debe "cubrir" cada miembro de un conjunto dado (conjunto 1) mediante un miembro aceptable de algún otro conjunto (llamado conjunto 2). El objetivo en un problema de cobertura de conjuntos es minimizar el número de elementos del conjunto 2 que se necesitan para cubrir todos los elementos del conjunto 1.

Consideremos un ejemplo del PLSS, sea $m = 4$ y $n = 5$, donde todos los $c_{ij}$ son cero o infinito:

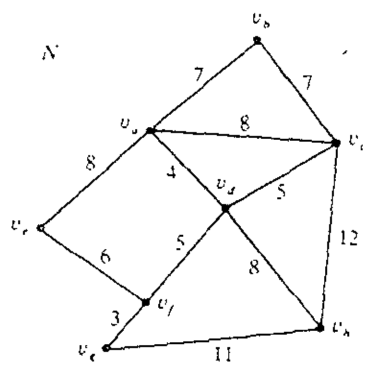
$$C = \begin{bmatrix} 0 & \infty & \infty & 0 & 0 \\ 0 & 0 & 0 & \infty & \infty \\ 0 & \infty & 0 & \infty & 0 \\ \infty & 0 & 0 & 0 & \infty \end{bmatrix}; \; f_i = \begin{bmatrix} 2 \\ 6 \\ 3 \\ 8 \end{bmatrix}$$

y sea la matriz $A$ de (4 x 5) con $\{a_{ij}\}$ definidos como: $a_{ij} = \begin{cases} 1 \text{ para } c_{ij} = 0 \\ 0 \text{ para } c_{ij} = \infty \end{cases}$, entonces la matriz $A$ viene dada por: $A = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$.

Si un servicio se establece en el lugar $i$ a un costo fijo $f_i$, éste puede atender al cliente $j$ con costo nulo $(c_{ij} = 0)$ o no puede atender al cliente $j$ con costo $c_{ij} = \infty$ ($c_{ij} = \infty$, indica que la ruta de transporte $P[i, j]$ está bloqueada). En términos de la matriz $A$, introducimos una ligera diferencia en la terminología diciendo que el servicio $i$, si está establecido, ***cubre*** a todos los clientes $j$ para cualquier $a_{ij} = 1$. Además, un subconjunto $Q$ de índices renglón $Q \subseteq I = \{1,...,m\}$, define una *cobertura* del conjunto $J = \{1,...,n\}$ de índices columna, sí $\sum_{i \in Q} a_{ij} \geq 1$ para toda $j$. El costo de una cobertura $Q$ es $\sum_{i \in Q} f_i$ y el conjunto *ponderado* del problema de cobertura (cobertura de conjuntos) es encontrar una cobertura de costo mínimo. En particular, para el ejemplo de datos mostrado, una cobertura óptima es $\{1,2\}$ de costo 2+6=8. Es decir, que con 1 y 2 se cubrirá a todos los clientes.

Si sucede que todos los costos fijos son iguales a 1, nos referimos al correspondiente problema de cobertura *no ponderado*. En este caso una cobertura óptima es simplemente una cobertura de mínima cardinalidad. Si consideramos en el ejemplo de datos esta modificación en los $f_i$, las coberturas óptimas son $\{1,2\}$, $\{1,4\}$ y $\{3,4\}$.

El problema de cobertura de conjuntos relaciona en forma natural con ciertos problemas de localización que involucran restricciones de distancia. Usaremos la red $N$ mostrada en la figura (3) para ilustrar este aspecto. Supongamos que deseamos determinar el mínimo número de servicios y sus localizaciones tal que la distancia entre un servicio y cualquier cliente asignado no exceda de 8. La matriz $A$ resultante para toda $i, j$ con $a_{ij} = 1$ para $d(v_i, v_j) \leq 8$, o $a_{ij} = 0$ en caso contrario, se muestra en la figura (5).




$D$:

| | $v_a$ | $v_b$ | $v_c$ | $v_d$ | $v_e$ | $v_f$ | $v_g$ | $v_h$ |
|---|---|---|---|---|---|---|---|---|
| $v_a$ | 0 | 7 | 8 | 4 | 8 | 9 | 12 | 12 |
| $v_b$ | 7 | 0 | 7 | 11 | 15 | 16 | 19 | 19 |
| $v_c$ | 8 | 7 | 0 | 5 | 16 | 10 | 13 | 12 |
| $v_d$ | 4 | 11 | 5 | 0 | 11 | 5 | 8 | 8 |
| $v_e$ | 8 | 15 | 16 | 11 | 0 | 6 | 9 | 19 |
| $v_f$ | 9 | 16 | 10 | 5 | 6 | 0 | 3 | 13 |
| $v_g$ | 12 | 19 | 13 | 8 | 9 | 3 | 0 | 11 |
| $v_h$ | 12 | 19 | 12 | 8 | 19 | 13 | 11 | 0 |

$A$:

| | $v_a$ | $v_b$ | $v_c$ | $v_d$ | $v_e$ | $v_f$ | $v_g$ | $v_h$ |
|---|---|---|---|---|---|---|---|---|
| $v_a$ | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| $v_b$ | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| $v_c$ | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| $v_d$ | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| $v_e$ | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| $v_f$ | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 |
| $v_g$ | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| $v_h$ | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |

Fig. (5) Ejemplo del conjunto de cobertura: $a_{ij} = 1$ para $d(v_i, v_j) \leq 8$ ó 0 en caso contrario.

Una solución óptima en este caso (pero ciertamente no en general) es bastante visible: puesto que un solo servicio no puede cubrir a todos los clientes, el par $\{v_a, v_d\}$ cubre a todos los ellos.

De esta forma, es posible considerar al Problema de Cobertura de Conjuntos ($PCC$) como una base para el análisis de problemas, los cuales puedan interpretarse como la localización de un subconjunto de nodos en una red, tal que todos los nodos de la red sean cubiertos. El $PCC$ también es un modelo adecuado para analizar una amplia variedad de otros problemas de optimización discreta.

## 2.5.2 FORMULACIONES MATEMÁTICAS DE LOS PROBLEMAS: P-MEDIAN *(p-MP)*, P-CENTER *(p-CP)* Y DEL PROBLEMA DE LOCALIZACIÓN DE SERVICIOS *(PLSS)*.

Como las primeras tres familias comparten en común varias características, se presenta enseguida una formulación unificada de éstas. Para lo cual consideremos a los tres problemas: el $p-MP$, el $p-CP$ y el $PLSS$ en su forma más general, donde la matriz subyacente es bipartita, los componentes de estos problemas se definen como:

$m$: el número finito de servicios; $i = 1,...,m$

$n$: el número de clientes; $j = 1,...,n$

$p$: el número de servicios a abrirse o establecerse, $1 \le p \le n$

$f_i$: el costo fijo de establecer o abrir el servicio $i$.

$c_{ij}$: el costo total variable por atender toda la demanda del cliente $j$ por el servicio $i$.

El costo $c_{ij}$ puede incluir medidas de la distancia desde el servicio $i$ hasta el cliente $j$. Por ejemplo $c_{ij}$ puede ser interpretado como $c_{ij} = w_j\left(h_i + t_{ij}\right)$ donde:

$w_j$: es el número de unidades demandadas por el cliente $j$,

$h_i$: es el costo por unidad de operar el servicio $i$ (incluyendo los costos variables de producción, costos administrativos, etc.)

$t_{ij}$: es el costo de transporte por enviar una unidad desde el servicio $i$ hasta el cliente $j$.

El costo $t_{ij}$ puede interpretarse también como $t_{ij} = d_{ij}$, donde:

$d_{ij}$: es la distancia física (o su equivalencia en tiempo o dinero) de una ruta desde el servicio $i$ hasta el cliente $j$.

Entonces, cuando $h_i = 0$, $c_{ij}$ se reduce a $c_{ij} = w_j\left(t_{ij}\right)$ ó $c_{ij} = w_j\left(d_{ij}\right)$. Con $c_{ij}$ conocido, podemos, sin pérdida de generalidad, suponer que todos los clientes tienen demanda unitaria. Además, para cada uno de los tres problemas, no se imponen restricciones de capacidad y tampoco se restringe el número de clientes que cada servicio potencial pueda atender. Y por último, existe el supuesto de que todos los datos son no negativos.

En pocas palabras, expresamos por conveniencia que las decisiones de localización son hechas en términos de $Q : Q \subseteq I$, un subconjunto de servicios potenciales a ser abiertos y desde los cuales todos los clientes serán atendidos. La cardinalidad de $Q$ es denotada por $|Q|$. Conceptualmente, un enfoque para resolver cada uno de los tres problemas, se puede decir que involucra dos fases:

(1) Una **fase de localización,** la cual determina $Q$, y

(2) Una **fase de asignación,** en la cual cada cliente es asignado a exactamente un servicio abierto y toda la demanda de ese cliente será cubierta desde el servicio que le fue asignado, tal que cierta función objetivo sea minimizada.

DATOS QUE SE REQUIEREN PARA EL PROBLEMA *P-MEDIAN* $(p-MP)$: $m,n,p,C=\{c_{ij}\}$

Considerar el problema de abrir $p$ servicios y asignar cada cliente a exactamente uno de ellos tal que el costo total variable se minimice. Para un subconjunto de los servicios potenciales, $Q$ dado, una asignación que minimiza el costo total variable puede ser determinada *"por inspección"*: eso es, el cliente $j$ es asignado a un servicio $i$ establecido, tal que el correspondiente costo $c_{ij}$ sea el más pequeño, esto es, al servicio $i$ tal que $\min_{i \in Q} c_{ij}$. Asignando a todos los clientes de esta manera, el costo total variable resultante viene dado por $\sum_{i \in J} \min c_{ij}$. De esta manera el problema se puede formular como:

$$p-MP: \qquad \min_{Q \subseteq I, |Q|=p} \left\{ \sum_{j \in J} \min_{i \in Q} c_{ij} \right\}.$$

DATOS QUE SE REQUIEREN PARA EL PROBLEMA *P-CENTER* $(p-CP)$: $m,n,p,C=\{c_{ij}\}$

Sea el problema de abrir $p$ servicios y asignar cada cliente a exactamente uno de estos servicios, tal que, el máximo costo variable por atender a cualquier cliente sea mínimo. Si se supone $Q$ conocido, es posible asignar el $j-ésimo$ cliente a aquél servicio para el cual el costo $c_{ij}$ sea el mínimo, esto es, asignar al servicio $i$ tal que $\min_{i\in Q} c_{ij}$. Asignando a todos los clientes de esta manera, el costo máximo resultante viene dado por $\max_{j\in J}\left\{\min_{i\in Q} c_{ij}\right\}$. De esta manera obtenemos la siguiente formulación:

$$p-CP: \qquad \min_{Q\subseteq I,|Q|=p}\left\{\max_{j\in J}\left\{\min_{i\in Q} c_{ij}\right\}\right\}.$$

DATOS QUE SE REQUIEREN PARA EL PROBLEMA DE LOCALIZACIÓN DE SERVICIOS SIMPLE (*PLSS*): $m$, $n$, $C=\{c_{ij}\}$ $f=(f_i)$

Abrir un subconjunto $Q\subseteq I$ de servicios y asignar a cada cliente a exactamente uno de esos servicios, tal que la suma de los costos fijos y los costos variables sea minimizada. Esto es,

$$PLSS: \qquad \min_{Q\subseteq I}\left\{\sum_{i\in Q} f_i + \sum_{j\in J}\min_{i\in Q} c_{ij}\right\}.$$

Como vemos, el número de servicios a establecer no está predeterminado, pero lo determinará la solución.

### 2.5.3 EL *P*-PROBLEMA DE LOCALIZACIÓN DE SERVICIOS SIMPLE ($p-PLSS$)

Vistos los problemas *p-Median* (*p-MP*) y el Problema de Localización de Servicios Simple (*PLSS)* como ejemplos "puros" de los problemas prototipo de localización, el problema *p-PLSS* es un modelo híbrido en el sentido de que combina ciertas características del $p$-$MP$ y del $PLSS$.

Para este caso, el problema $p-PLSS$ puede verse como una extensión del problema $p-MP$ en el cual los costos fijos presentados en el PLSS están asociados con los servicios potenciales. Alternativamente, el problema $p-PLSS$ es similar al PLSS pero con la restricción adicional de que el número de servicios a establecer (a abrir) es un parámetro preestablecido como $p$. Por otra parte, un ejemplo de datos para el problema $p-PLSS$, consiste en los enteros positivos $m$, $n$, $p$, el $m$-vector $f=(f_1,\ldots,f_m)$ y la matriz $C$ de $(mxn)$. Quedando las $m+nm$ variables definidas como:

$$y_i = \begin{cases} 1, & \text{si el servicio } i \text{ es abierto.} \\ 0, & \text{si el servicio } i \text{ no se abre.} \end{cases}$$

$$x_{ij} = \begin{cases} 1, & \text{si el cliente } j \text{ satisface toda su demanda desde el servicio } i. \\ 0, & \text{si el cliente } j \text{ no es atendido por servicio } i. \end{cases}$$

Entonces el problema $p-PLSS$ puede establecerse como el siguiente programa entero (0,1):

$$p-PLSS \quad \begin{cases} Min \quad \sum_{i\in I} f_i y_i + \sum_{i\in I} \sum_{j\in J} c_{ij} x_{ij} & & (1) \\ \text{s.a}: & & \\ \sum_{i\in I} x_{ij} = 1, & \forall j & (2) \\ x_{ij} \le y_i, & \forall i, \forall j & (3) \\ \sum_{i\in I} y_i = p & & (4) \\ y_i \in (0,1), \ \forall i; \quad x_{ij} \in (0,1), \ \forall i, \forall j & & \end{cases}$$

La función objetivo indica que se debe minimizar los costos totales. La restricción (2) indica que un solo servicio atenderá a un cliente, es decir, que no se permitirá que entre varios servicios se atienda a un solo cliente. La restricción (3) establece que un servicio podrá atender a un cliente únicamente cuando ese servicio esté abierto. Y la restricción tipo (4) indica que exactamente $p$ servicios se establezcan.

Si $f_i = 0$, para toda $i$, el poblema $p-PLSS$ se reduce al problema *p-Median*, $p-MP$. Si se ignora la restricción (4), se obtiene el Problema de Localización de Servicios Simple (*PLSS*).

## 2.6 EL PROBLEMA DE ASIGNACIÓN CUADRÁTICA, (*PAC*).

El cuarto y último problema prototipo de Localización, es el caso especial formulado como el Problema de Asignación Cuadrática, *PAC*. Para lo cual, supongamos que estamos en el proceso de diseño de los departamentos de una universidad los cuales tendrán cuatro funciones (unidades operacionales) definidas como:

*a*: Unidad para personal administrativo,

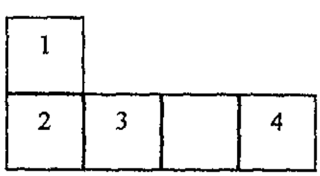
*b*: Unidad para servicios recreativos y refrigerios (por ejemplo, restaurante),

*c*: Unidad de estancia para los investigadores, y

*d*: Unidad para librería.

Se supondrá una situación en la que hay un presupuesto limitado, por lo cual, el que decide construir no puede permitirse erigir un edificio nuevo, sino que tendrá que aceptar el terreno disponible de un edificio existente para estos propósitos con capacidad suficiente para acomodar las cuatro unidades: *a* - *d*. En la figura (6) se muestra el edificio existente con cuatro áreas de igual tamaño (1,2,3 y 4) y con un espacio libre entre las áreas 3 y 4 que se ocupará con los elevadores y escaleras.

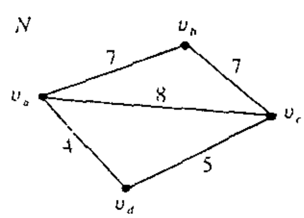
La matriz $A = \{a_{ij}\}$ de la figura (6) es una matriz simétrica de las *distancias rectangulares* entre cada par de áreas.




$$\begin{bmatrix} 0 & 1 & 2 & 4 \\ 1 & 0 & 1 & 3 \\ 2 & 1 & 0 & 2 \\ 4 & 3 & 2 & 0 \end{bmatrix} = A$$

Fig.(6) Primera parte de la instancia del *PAC* con la matriz $A$ de distancias.

Se supone que cada área puede albergar exactamente a una de las cuatro unidades. Entonces, el problema es asignar cada unidad a un área, tal que, la distribución resultante en algún sentido bien definido venga a ser el óptimo. Supongamos ahora que las cuatro unidades (*a* - *d*) están representadas por los nodos $v_a$, $v_b$, $v_c$ y $v_d$ de la red $N$ de la figura (7), con la matriz $B = \{b_{ij}\}$ de conexión, con $b_{ij} = b_{ji}$ igual a la "longitud" $\alpha\left(v_i, v_j\right)$ del arco $\left[v_i, v_j\right]$ si existe tal arco, y $b_{ij} = 0$ si no existe ese arco. En otras palabras podemos decir de lo anterior que la "longitud" corresponde a una medida de la conexión entre las unidades *i* y *j*; y cuando $b_{ij} = 0$ significa que no hay relación entre las unidades *i* y *j*.




$$\begin{array}{c|cccc} & a & b & c & d \\ \hline a & - & 7 & 8 & 4 \\ b & 7 & - & 7 & 0 \\ c & 8 & 7 & - & 5 \\ d & 4 & 0 & 5 & - \end{array} \Bigg\} = B$$

Fig. (7) Segunda parte del ejemplo del Problema de Asignación Cuadrática.

Los elementos de la diagonal principal de $B$ permanecen indefinidos (-). A la matriz $B$ se llama la matriz de conexión porque, por ejemplo, $b_{bc} = b_{cb} = 7$ unidades abarca todos los aspectos de "conexión" entre los "investigadores" y el "restaurante", esta "conexión" se refiere a, por ejemplo, el número de veces que acuden los investigadores al restaurante, y otros casos semejantes que "conecta" a los investigadores con dicha unidad.

Si las unidades $c$ y $d$ son asignadas a las áreas 2 y 4, respectivamente, entonces la correspondiente contribución a la función objetivo será $b_{cd}a_{24} = (5 \times 3) = 15$, que es el producto de la conexión entre las unidades $a$, $d$ y la distancia entre las áreas 2 y 4. Con todas las cuatro unidades colocadas, el valor de la función objetivo se determina como la suma de $\binom{4}{2} = 6$ contribuciones individuales, y una distribución que minimiza a la suma que se considera como óptima. En este caso se tendrán que evaluar 4!=24 distintas posibles distribuciones. Estas 24 combinaciones se listan en la tabla (6).

| Unidades ubicadas en las áreas | | | | Distancias asociadas con parejas de unidades (distancia rectangular) | | | | | $\sum(conexión \times distancia)$ |
|---|---|---|---|---|---|---|---|---|---|
| **1** | **2** | **3** | **4** | **(a, b)** | **(a, c)** | **(a, d)** | **(b, c)** | **(c, d)** | |
| a | b | c | d | 1 | 2 | 4 | 1 | 2 | 56 |
| a | b | d | c | 1 | 4 | 2 | 3 | 2 | 78 |
| a | c | b | d | 2 | 1 | 4 | 1 | 3 | 60 |
| a | c | d | b | 4 | 1 | 2 | 3 | 1 | 70 |
| a | d | b | c | 2 | 4 | 1 | 2 | 3 | 79 |
| a | d | c | b | 4 | 2 | 1 | 2 | 1 | 67 |
| b | a | c | d | 1 | 1 | 3 | 2 | 2 | 51 |
| b | a | d | c | 1 | 3 | 1 | 4 | 2 | 73 |
| b | c | a | d | 2 | 1 | 2 | 1 | 3 | 52 |
| b | c | d | a | 4 | 3 | 2 | 1 | 1 | 72 |
| b | d | a | c | 2 | 2 | 1 | 4 | 3 | 77 |
| b | d | c | a | 4 | 2 | 3 | 2 | 1 | 75 |
| c | d | a | b | 2 | 2 | 1 | 4 | 1 | 67 |
| c | d | b | a | 2 | 4 | 3 | 2 | 1 | 77 |
| c | a | b | d | 1 | 1 | 3 | 2 | 4 | 61 |
| c | a | d | b | 3 | 1 | 1 | 4 | 2 | 71 |
| c | b | a | d | 1 | 2 | 2 | 1 | 4 | 58 |
| c | b | d | a | 3 | 4 | 2 | 1 | 2 | 78 |
| d | a | b | c | 1 | 3 | 1 | 2 | 4 | 69 |
| d | a | c | b | 3 | 1 | 1 | 2 | 2 | 57 |
| d | b | a | c | 1 | 2 | 2 | 3 | 4 | 72 |
| d | b | c | a | 3 | 2 | 4 | 1 | 2 | 70 |
| d | c | a | b | 2 | 1 | 2 | 3 | 1 | 56 |
| d | c | b | a | 2 | 3 | 4 | 1 | 1 | 66 |

Tabla (6) Un ejemplo del Problema, *PAC*, resuelto por enumeración completa.

Como vemos, la distribución óptima, en términos de minimizar la cantidad "$\sum(conexión \times distancia)$" es asignar las unidades $\{b,a,c,d\}$ a las áreas $\{1,2,3,4\}$ respectivamente, como se muestra en la figura (8).

| | | | |
|---|---|---|---|
| 1<br>b | | | |
| 2<br>a | 3<br>c | | 4<br>d |

Fig. (8). Asignación óptima del ejemplo del problema *PAC*.

Notamos que en contraste con las otras tres familias de problemas prototipo consideradas antes, los términos "cliente" y "servicios" no están explícitamente empleados en el Problema de Asignación Cuadrática. En estos problemas cada objeto a ser ubicado para que proporcione (reciba) conexión a (desde) otros objetos pueden verse como un cliente o un servicio.

En términos generales, suponemos que un conjunto $M = \{1,\ldots,m\}$ de $m$ unidades de igual tamaño a ser ubicadas en un conjunto $N = \{1,\ldots,n\}$ de $n \geq m$ áreas, cada una de las cuales pueda albergar a lo más una unidad.

Sean las matrices $A$ y $B$ interpretadas como antes y sean las $m \times n$ variables de decisión $x_{is}$, $i \in M$, $s \in N$ definidas como: $x_{is} = 1$ si la unidad $i$ es localizada o ubicada en el área $s$; y $x_{is} = 0$ en caso contrario. Si un par $\{i,j\}$ de unidades se asignan a las áreas $\{s,t\}$, respectivamente, entonces la contribución a la función objetivo es $b_{ij}a_{st}$, la cual, con las variables de decisión introducidas, pueden expresarse por el término cuadrático $x_{is}x_{jt}b_{ij}a_{st}$. Una formulación al problema de programación 0 - 1 de *PAC* es entonces:

$$Min \quad Z = \sum_{i \in M} \sum_{j \in M} \sum_{s \in N} \sum_{t \in N} x_{ij} x_{jt} b_{ij} a_{st}$$

$s.a:$

$$\sum_{s \in N} x_{is} = 1 \quad , \qquad \forall i \qquad \text{(cada unidad deberá ser ubicada)}$$

$$\sum_{i \in M} x_{is} \leq 1 \quad , \qquad \forall s \qquad \text{(a lo más una unidad en cada área)}$$

$$x_{is} \in (0,1) \quad , \qquad \forall i, \forall s$$

# CAPÍTULO III. COMPLEJIDAD COMPUTACIONAL

Para nuestros propósitos, consideremos a los algoritmos como procedimientos que se realizan paso por paso para resolver problemas en un tiempo finito de cálculo para todos los ejemplos de datos. Es frecuentemente aceptado caracterizar a los algoritmos por una medida relacionada a su *función de complejidad de tiempo*. Para un problema de optimización (abreviándolo de aquí en adelante como OPT), una familia de ejemplos de datos de un tamaño $q$ dado correspondiente a la duración de entrada, y un algoritmo específico $\Omega$, denotamos a la función de complejidad de tiempo del algoritmo por $f_\Omega(q)$ donde $f_\Omega(q)$ expresa la gran cantidad de tiempo requerido para resolver el problema para una instancia de datos arbitraria de tal tamaño. Obviamente, $f_\Omega(q)$ depende de tres aspectos: de la computadora utilizada, del esquema de codificación empleado por el algoritmo y del ejemplo, pero ninguno de estos tres aspectos tiene un efecto significativo sobre los cálculos del algoritmo. Aproximadamente $f_\Omega(q)$ expresa el máximo número de operaciones elementales, tales como sumas, multiplicaciones y comparaciones ejecutadas por el algoritmo para resolver cualquier instancia de tamaño $q$.

Para caracterizar el orden de la función $f_\Omega(q)$, se dice que $f_\Omega(q)$ es $O(g(q))$ siempre que exista alguna constante $\alpha$ tal que $|f_\Omega(q)| \leq \alpha|g(q)|$ para toda $q$, $q = 1,2,3,...$ Entonces nos referimos al algoritmo $\Omega$ como uno de orden $O(g(q))$ para el problema de optimización OPT. Un algoritmo $\Omega$ con una función de complejidad de tiempo $f_\Omega(q)$, se llama *polinomial* (tiempo acotado) o bueno, o eficiente si $f_\Omega(q)$ es de orden $O(g(q))$, $q = 1,2,3,...,$ y si $g(q)$ es algún polinomio del tamaño $q$, (donde $q$ es el tamaño del problema). Si $f_\Omega(q)$ no puede ser acotado, el algoritmo $\Omega$ se llama *exponencial*. Ya que esta definición involucra a "todas los ejemplos de datos", podemos alternativamente decir que un algoritmo es polinomial ó que en el *peor de los casos* es exponencial. Un problema para el cual existe un algoritmo polinomial, a veces es referido como *bien resuelto*.

Para ejemplificar esta notación, consideremos un problema OPT, un algoritmo $\Omega$, y para $q = 1,2,3,...,$ supongamos $f_\Omega(q) = 7q^5 + 3q^2 + \log q$, así, se puede usar

$g(q)=q^5$, y $|f_{\Omega}(q)|\leq 10g(q)$, entonces $\Omega$ es de orden $O\left(q^5\right)$ y por lo tanto es un algoritmo polinomial para el problema de optimización OPT.

En contraste a los problemas de optimización, ahora nos referiremos a los *problemas de decisión*,$\pi$, como los que tienen dos posibles resultados: sí o no. A todo problema de optimización se le puede asociar un problema de decisión. Por ejemplo, dado un problema de optimización *PLSS* para un ejemplo dado de $m$, $n$, $C$, $f$, y un valor de entrada $k$, el problema de decisión asociado, $\pi(PLSS)$ es, ¿el *PLSS* tiene una solución con valor de a lo más $k$?.

La teoría de los problemas **NP-Completos** (**NP**: polinomial no determinista) se basa en el concepto de máquina de Turing determinista y no determinista, diseñada para proporcionar respuestas sí/no a problemas de decisión puestos en términos de "lenguaje de reconocimiento". Para un alfabeto y un "lenguaje" dados, una entrada consiste de una cadena finita de símbolos del alfabeto aceptados por la máquina (es decir, que resuelva el problema de decisión) sí y solo sí esta cadena pertenece al lenguaje.

Consideremos como una cadena un ejemplo de datos y un lenguaje como un tipo de problema o como el conjunto de todos sus ejemplos factibles. Un problema de decisión verifica la factibilidad de un ejemplo de datos para un tipo de problema dado, se dice que un problema pertenece a la clase **P** si la factibilidad o infactibilidad de cualquier ejemplo de datos puede ser determinado por algún algoritmo en tiempo polinomial. De este modo, $\pi(PLSS)\in\mathbf{P}$, si para cualquier $m$, $n$, $C$, $f$, $k$, dados, podemos en tiempo polinomial confirmar o rechazar su asociación con el conjunto de ejemplos factibles. Actualmente no se conoce si $\pi(PLSS)$ pertenece a la clase **NP**, la cual puede ser caracterizada de la siguiente forma: Para un ejemplo $\gamma$ dado de un problema de decisión, su factibilidad implica la existencia de una estructura apropiada $\delta$ asociada con $\gamma$. Si la longitud de $\delta$ está acotada por algún polinomio de longitud $\gamma$, y si para $(\gamma,\delta)$ dados podemos afirmar la factibilidad de $\delta$ en tiempo polinomial, entonces este problema de decisión se dice que pertenece a la clase polinomial.

Para $\pi(PLSS)$ consideremos un ejemplo $\gamma$ definido por $m$, $n$, $C$, $f$, $k$. Entonces $\gamma$ es factible si se le puede asignar valores 0 o 1 a todas las variables $y_i, x_{ij}$ en la formulación del problema entero *PLSS*, tal que todas las restricciones se satisfagan y tal que el valor de la función objetivo no exceda de un cierto valor $k$. Aquí la estructura $\delta$ asociada con $\gamma$ es un vector 0 - 1 con $n+mn$ elementos que representan los valores de las $n+mn$ variables $y_i, x_{ij}$. Entonces, una conversión binaria de $\delta$ es de longitud $n+mn$ la cual tiene el mismo orden que el orden de la longitud $\gamma$. Para afirmar la factibilidad de $\delta$, esto es, para verificar que $\delta$ satisface las restricciones y que la función objetivo no excede a $k$, se requieren cálculos del orden $n+mn$. Ya que ambas condiciones satisfacen a sus miembros, se confirma que $\pi(PLSS) \in \mathbf{NP}$. Sin embargo, para asegurar que no existe una solución factible es necesario recorrer todas las soluciones y que ninguna sea factible.

Si para algún ejemplo de datos de un problema de decisiones $\pi'$, es posible construir en tiempo polinomial un ejemplo de datos de un problema de decisiones $\pi$, tal que el ejemplo de $\pi'$ es factible sí y solo sí el ejemplo de $\pi$ es factible, entonces $\pi'$ se dice que es polinomialmente transformable a $\pi$. Usaremos la notación $\pi' \alpha \pi$ para indicar que $\pi'$ es polinomialmente transformable a $\pi$. Así, $\pi' \alpha \pi$ implica que $\pi'$ puede verse como un caso especial de $\pi$, y en consecuencia, $\pi$ es al menos tan difícil de resolver como $\pi'$.

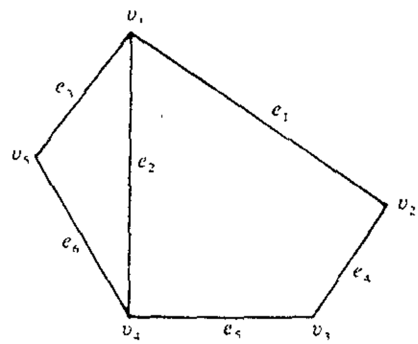
Sí $\pi' \alpha \pi$ para todo $\pi' \in \mathbf{NP}$, entonces algún problema perteneciente a la clase **NP** puede verse como un caso especial de $\pi$, el cual es entonces llamado **NP** - duro. Finalmente, $\pi$ es llamado **NP** - Completo, si $\pi$ es **NP** - duro y $\pi \in$ **NP**. Además, si algún $\pi' \alpha \pi$, con $\pi$ perteneciendo a la clase **NP** y $\pi'$ es **NP** - Completo, entonces $\pi$ es también **NP** - Completo.

**Teorema 1. El problema de Localización de Servicios Simple es NP - duro.**

**Prueba.** Primero necesitamos introducir el *Problema de Cobertura de Nodos* (*PCN*). Dado un grafo $G$ y un entero $k$, ¿existe un subconjunto de $k$ nodos de $G$ que cubran a todos los arcos de $G$?. (el nodo $v$ se dice que cubre al arco $e$ si $v$ es un punto terminal de $e$). El problema de cobertura de nodos es **NP - Completo.**

Ahora reducimos el *PCN* al problema *PLSS*. Consideremos un grafo $G=(V,E)$, donde $V$ representa el conjunto de nodos y $E$ el conjunto de arcos. Construimos una instancia del PLSS con el conjunto de servicios potenciales $I=V$ y el conjunto de clientes $J=E$. Sea

$c_{ij} = 2$ si $v_i \in V$ es un punto terminal de $e_j \in E$ y sea $c_{ij} = 0$ en caso contrario. También sea $f_i = 1$ para toda $v_i \in V$. Esta transformación es polinomial para el tamaño del grafo. Un ejemplo de la transformación se muestra en la figura (9).




$$f = (1 \quad 1 \quad 1 \quad 1 \quad 1)$$

$$C = \begin{pmatrix} 2 & 2 & 0 & 0 & 0 \\ 2 & 0 & 0 & 2 & 0 \\ 2 & 0 & 0 & 0 & 2 \\ 0 & 2 & 2 & 0 & 0 \\ 0 & 0 & 2 & 2 & 0 \\ 0 & 0 & 0 & 2 & 2 \end{pmatrix}$$

Fig. (9) Transformación de un problema de cobertura de nodos a un problema *PLSS*.

Un ejemplo del *PLSS* definido en esta forma consiste en cubrir todos los arcos del grafo $G$ con el mínimo número de nodos. De este modo, una solución óptima del *PLSS* proporciona la respuesta al *PCN*. Esto prueba que el *PLSS* es **NP-duro.**

En el ejemplo, $y_1 = y_3 = y_4 = 1$ y $y_2 = y_5 = 0$ es una solución óptima para la instancia del *PLSS*. Es decir, tres nodos son necesarios para cubrir todos los arcos de $G$.

Una transformación polinomial que reduce un problema conocido **NP-duro** a un problema $(Q)$, muestra que $(Q)$ es también **NP-duro.**

Un corolario inmediato del teorema anterior es que el problema de localización de $p-$servicios es **NP-duro** puesto que al resolverlo para cada $p = 1,\ldots,m$ proporciona una solución para el *PLSS*.

No obstante que el *PLSS* es **NP-duro,** algunos casos especiales se han podido resolver en tiempo polinomial. Kolen (1983) mostró que el *PLSS* es soluble en tiempo $O(r^3)$ cuando el problema es definido como un árbol con $r$ nodos y algunas otras suposiciones como las siguientes:

- Tanto los clientes como los servicios están colocados en los nodos del árbol;
- Una longitud positiva está asociada con cada borde del árbol;
- $d_{ij}$ es la longitud de la ruta entre los nodos $i$ y $j$.

# CAPÍTULO IV. FORMULACIÓN DEL PROBLEMA DE LOCALIZACIÓN DE SERVICIOS SIMPLE (PLSS) Y SU FACTIBILIDAD.

El Problema de Localización de Servicios Simple (*PLSS*) se puede establecer como sigue. Dado un conjunto finito de posibles lugares para ubicar nuevos servicios (o reubicación de servicios existentes), el *PLSS* trata de satisfacer la demanda de un conjunto de clientes con la oferta de un subconjunto de esos servicios potenciales, ya sea desde un solo servicio o varios, *sin que más de un solo servicio satisfaga a un solo cliente* (propiedad de una sola asignación). Además se supone que los servicios son de capacidad ilimitada, tal que en principio, cualquier servicio puede satisfacer todas las demandas. Para los costos asociados con los servicios y los costos de transporte para enviar desde los servicios hasta los clientes, lo que se busca es un plan de mínimo costo que satisfaga a todas las demandas.

Para formular un modelo para el *PLSS*, definimos la siguiente notación.

$m$ : es el número de servicios indexados por $i$, $i = 1,\dots,m$.

$n$ : es el número de clientes indexado por $j$, $j = 1,\dots,n$.

$x_{ij}$ : es la fracción de la demanda $j$ atendida por el servicio $i$.

$y_i$ : es igual a 0 si no se puede enviar nada desde el servicio $i$, es decir, si el servicio $i$ es "cerrado"; y es igual a 1 si se puede enviar cualquier cantidad desde el servicio $i$, es decir, si el servicio $i$ es "abierto".

$c_{ij} = t_{ij}D_j$, donde $t_{ij}$ es el costo por transportar desde el servicio $i$ hasta el cliente $j$, incluyendo el costo de producción y $D_j$ es la demanda del cliente $j$.

$f_i$ : es el costo fijo por abrir el servicio $i$. Este costo es positivo e independiente de la cantidad enviada desde el servicio $i$.

Por las características anteriores en estos problemas nos enfrentamos a dos tipos de restricciones.

**Restricción tipo 1:**

Cada cliente debe ser atendido por un solo servicio, ya que cada servicio tiene capacidad ilimitada.

**Restricción tipo 2:**

Si un cliente es atendido por un servicio, este servicio tiene que estar abierto.

La restricción tipo 1 establece que para cada cliente $j$; $(j = 1,\ldots,n)$ exactamente una de las variables $x_{1j}, x_{2j}, \ldots, x_{mj}$ debe ser igual a 1, y las otras tienen que ser iguales a 0. Esto se puede lograr con la siguiente restricción para cada cliente: $\sum_{i=1}^{m} x_{ij} = 1$. Esta restricción se debe a que el problema de localización de servicios simple tiene la ***propiedad de una sola asignación*** como ya se explicó anteriormente

La restricción tipo 2 establece que sí $x_{ij} = 1$, es decir, que el cliente $j$ es atendido por el servicio $i$, entonces $y_i$ debe ser igual a 1. Por ejemplo, supóngase que $x_{12} = 1$, entonces, debe haber un servicio abierto en $i = 1$, y por lo tanto, se debe satisfacer $y_1 = 1$. Pero, si $x_{ij} = 0$, $y_i$ puede ser 1 o 0. Esto se puede obtener con el siguiente tipo de restricción para cada servicio y para cada cliente: $x_{ij} \le y_i$.

Sí $x_{ij} = 1$, entonces la restricción anterior asegura que $y_i = 1$, como se quiere. En resumen, las restricciones tipo 2 aseguran que se va a pagar por abrir un servicio $i$ sí y solo sí se usa el servicio $i$. Como vemos, la variable $y_i$ es de carácter binario.

Teniendo en cuenta que el objetivo del problema es minimizar los costos totales, un modelo para formular el *PLSS* es la siguiente:

$$
\left.
\begin{array}{llr}
Min\ Z = \sum\limits_{i=1}^{m} \sum\limits_{j=1}^{n} c_{ij} x_{ij} + \sum\limits_{i=1}^{m} f_i y_i & & (1) \\
s.a: & & \\
\quad \sum\limits_{i=1}^{m} x_{ij} = 1; & \forall j & (2) \\
\quad x_{ij} \le y_i; & \forall i, \forall j & (3a) \\
\quad x_{ij} \ge 0; & \forall i, \forall j & (4) \\
\quad y_i \in \{0,1\}; & \forall i & (5)
\end{array}
\right\} \quad (P1)
$$

En la solución óptima, exactamente $n$ de las $x_{ij}$ tomarán el valor de uno y todas las demás, serán cero.

Existe otra manera para poder modelar las restricciones $(3a)$. En lugar de las $(m \times n)$ restricciones de la forma $x_{ij} \leq y_i$ se puede incluir las $m$ restricciones del tipo:

$$\sum_{j=1}^{n} x_{ij} \leq ny_i; \ \forall i \qquad (3b)$$

Para un servicio dado, cada una de estas restricciones asegura que si se usa ese servicio, se tendrá que pagar por abrirlo. Por ejemplo, consideremos una restricción para el servicio $i = 1$ con $n = 3$ clientes: $x_{11} + x_{12} + x_{13} \leq 3y_1$, se utilizará el servicio 1, sí $x_{11} = 1$, $x_{12} = 1$, ó $x_{13} = 1$. Si cualquiera de las variables es igual a 1, entonces la restricción del servicio $i = 1$, asegurará que $y_1 = 1$, y se tendrá que pagar por establecer ese servicio. Ahora, si todas estas variables $(x_{11}, x_{12}, x_{13})$ son 0, el hecho de minimizar los costos hará que $y_1 = 0$, y no se incurrirá en el costo por un servicio en $i = 1$.

La razón de que en el lado derecho de la restricción $(3b)$ se tenga $ny_i$, es para asegurar que para cada servicio sea posible que atienda a todos los clientes.

A las dos formas de restricciones $(3a)$ y $(3b)$ se les llama *restricciones fuertes* y *restricciones débiles*, respectivamente. Al modelo con restricciones fuertes le llamaremos Problema de Localización de Servicios Simple con Restricciones Fuertes (*PLSS-F*) y al modelo con las restricciones débiles, le llamaremos Problema de Localización de Servicios Simple con Restricciones Débiles (*PLSS-D*). Ambos modelos se presentan enseguida nuevamente:

$$Min\ Z = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} + \sum_{i=1}^{m} f_i y_i \qquad (1)$$

s.a:

$$\sum_{i=1}^{m} x_{ij} = 1\ ; \qquad \forall j \qquad (2)$$

$$x_{ij} \leq y_i\ ; \qquad \forall i, \forall j \qquad (3a) \qquad \text{(restricciones fuertes).}$$

$$n_i y_i - \sum_{j=1}^{n} x_{ij} \geq 0\ ; \qquad \forall i \qquad (3b) \qquad \text{(restricciones débiles).}$$

$$x_{ij} \geq 0\ ; \qquad \forall i, \forall j \qquad (4)$$

$$y_i = 0 \ \text{ ó } \ 1\ ; \qquad \forall i \qquad (5)$$

Cualquiera de las restricciones (*3a*) o (*3b*) que sea empleada, no afectará a la solución óptima. Para su solución, existen varios procedimientos computacionales que están basados en la relajación PL (programa lineal), en la cual las restricciones de tipo entero (5), son reemplazadas por los requerimientos de que $y_i \geq 0,\ \forall i$. En tales casos es importante saber el efecto que se tiene al reemplazar las $m \times n$ restricciones "desagregadas" (*3a*) por el conjunto de las $m$ desigualdades "agregadas" (*3b*). Algunos de los efectos o consecuencias que tiene el tomar las restricciones del tipo (*3b*) es que usualmente el número de clientes atendidos por el servicio $i$ será pequeño comparado con $n_i$. Esto causará en la solución del Programa Lineal que los servicios absorben solamente una pequeña parte del cargo fijo (costo fijo).

Las relajaciones LP de los dos planteamientos presentados, los resumimos en la tabla 7.

| *Relajación fuerte del PLSS* **(RF-PLSS)** | *Relajación débil del PLSS* **(RD-PLSS)** | **Ambas** |
|---|---|---|
| $m \times n$ desigualdades | $m$ desigualdades | $Min \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} + \sum_{i=1}^{m} f_i y_i$ |
| $y_i - x_{ij} \geq 0$ | $n_i y_i - \sum_{j=1}^{n} x_{ij} \geq 0$ | $\sum_{i=1}^{m} x_{ij} = 1$ |
| Desagregado | Agregado | $y_i \geq 0,\ x_{ij} \geq 0$ |

Tabla 7. Relajaciones LP fuerte y débil del *PLSS*.

Todas las variables de holgura para las restricciones fuertes $(y_i - x_{ij} \geq 0)$ están acotadas superiormente por +1, mientras que la variable de holgura de una desigualdad del planteamiento "débil", esto es, para alguna $i$ su restricción: $n_i y_i - \sum_{j=1}^{n} x_{ij} \geq 0$ puede conseguir cualquier valor no negativo menor o igual que $n_i$, que es el número de clientes que pueden ser satisfechos por ese servicio $i$. Para $n_i = n$ cada una de las restricciones "agregadas" es simplemente la suma de $n$ restricciones "desagregadas" o simplificadas.

Una solución factible para el *PLSS* con (3a) será factible para el *PLSS* con (3b), mientras que una solución factible para el *PLSS* con (3b) generalmente no será factible para el *PLSS* con (3a). Así, podemos decir que los adjetivos "fuerte" y "débil" tienen un significado intuitivo.

Algunos algoritmos existentes con enfoque de ramificación y acotamiento, adoptan la formulación "débil". Una desavenencia para los que realizan algoritmos de Branch and Bound para el *PLSS*, es seleccionar entre la relajación fuerte del *PLSS* (RF-PLSS) y la relajación débil del *PLSS* (RD-PLSS), o sus duales.

La relajación débil del *PLSS* (RD-PLSS) con significativamente pocas restricciones requiere en general menos esfuerzo computacional en la evaluación de cada cota, mientras que la relajación fuerte del *PLSS* (RF-PLSS) en general producirá cotas más estrechas. Ya que cualquiera de los dos programas lineales tendrá que ser resuelto repetidamente, la diferencia substancial en el número de restricciones conducirá a explorar la RD-PLSS, para el caso de algoritmos de ramificación y acotamiento. La RF-PLSS cumple con alguna característica que la hacen finalmente elegible para muchos algoritmos que no se basan en principio en ramificación y acotamiento, esta "cualidad" es la de proporcionar generalmente soluciones enteras.

De los dos planteamientos presentados anteriormente, el *PLSS* con (3a) y el *PLSS* con (3b) y sus respectivas relajaciones de programación lineal, esto es, la RD-PLSS y la RF-PLSS, se ha notado que para varios problemas, al resolverlos en su forma relajada, se han obtenido soluciones enteras para las $y_i$ y $x_{ij}$ si se toma el caso de la RF-PLSS. Pero no debemos olvidar que esto no sucede siempre, es decir, es falso afirmar que para el *PLSS* con (3a) automáticamente se obtienen soluciones enteras.

Cornuejols, Fisher y Nemhauser [1], tomaron la formulación relajada fuerte a través del dual Lagrangeano, con el fin de construir cotas para verificar las soluciones primales obtenidas por métodos heurísticos. Por otro lado, Donald Erlenkotter [2] desarrolló un algoritmo para el *PLSS* basado en la solución del dual y un ajuste en éste. Erlenkotter consideró para su algoritmo la formulación "fuerte" del *PLSS*.

En este trabajo de investigación, nosotros desarrollamos un algoritmo para el Problema de Localización de Servicios Simple, *PLSS*, y partimos de la formulación **"fuerte"** del *PLSS*. El algoritmo que presentamos consiste en primer lugar, en obtener el dual de la relajación fuerte del *PLSS* (RF-PLSS). Este dual se resuelve mediante un método que denominamos "nivelación dual". Esta nivelación proporciona una cota inferior para la búsqueda de la solución óptima del problema. La solución del programa lineal dual nos proporciona de alguna manera, la solución entera del primal, esto es debido a que el dual nos conduce a resolver un problema primal mucho más pequeño que el problema original relajado, en otras palabras, podemos decir que el dual nos indica, generalmente con gran certeza, donde se busque la solución óptima del *PLSS*.

Con el fin de mejorar este acercamiento que nos proporciona el dual, aplicamos un método de mejoramiento que le hemos llamado: "Pulido del Primal", el cual proporciona la solución del primal. Con estos dos métodos: Nivelación Dual y Pulido del Primal, llegamos entonces a obtener la solución del primal y del dual, respectivamente. A las funciones objetivo de ambos problemas, las hemos llamado $Z$ y $G$, respectivamente. Si al finalizar estos dos métodos se llega a que $Z = G$, entonces se ha encontrado la solución óptima entera del *PLSS*. Algunas veces, ambas funciones objetivo no son iguales, $Z \neq G$. Esto sucede por que la cota inferior proporcionada por el dual y la cota superior proporcionada por el primal no son iguales. Si ocurre esto, se busca la solución óptima en esta brecha (frecuentemente muy reducida), aplicando una tercera fase que se le ha llamado "Ramificación por Selección", la cual se parece al procedimiento de ramificación y acotamiento.

# CAPÍTULO V. PROPIEDADES DE LA DUALIDAD EN EL PROBLEMA DE LOCALIZACIÓN DE SERVICIOS SIMPLE.

## 5.1 COTA INFERIOR Y SUPERIOR DE LA SOLUCIÓN DEL PROBLEMA DE LOCALIZACIÓN SIMPLE.

Partimos de la formulación "fuerte" del Problema de Localización de Servicios Simple, (*PLSS-F*), mencionada en el capítulo III, con la relajación sobre las variables enteras, $y_i$. El problema relajado es el siguiente.

$$
\left.
\begin{array}{lr}
MinZ = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} + \sum_{i=1}^{m} f_i y_i & (1) \\
s.a: & \\
\quad \sum_{i=1}^{m} x_{ij} = 1 \quad ; \quad \forall j & (2) \\
\quad y_i - x_{ij} \geq 0 \quad ; \quad \forall i, \forall j & (3) \\
\quad x_{ij} \geq 0 \quad ; \quad \forall i, \forall j & (4) \\
\quad y_i \geq 0 \quad ; \quad \forall i & (5)
\end{array}
\right\} (P1R)
$$

El dual para $(P1R)$ es el siguiente:

$$
\left.
\begin{array}{lr}
MaxG = \sum_{j=1}^{n} \alpha_j & (6) \\
s.a: & \\
\quad \alpha_j - \lambda_{ij} \leq c_{ij} \quad ; \forall i, \forall j & (7) \\
\quad \sum_{j=1}^{n} \lambda_{ij} \leq f_i \quad ; \quad \forall i & (8) \\
\quad \lambda_{ij} \geq 0 \quad ; \quad \forall i, \forall j & (9)
\end{array}
\right\} (DP1R)
$$

Lo que se busca es encontrar soluciones factibles del problema primal $(P1R)$, a partir de soluciones factibles del dual $(DP1R)$.

La solución que el dual proporciona es menor o igual que la solución del primal, es decir, $G \leq Z$. Veamos por qué afirmamos esto a través del siguiente teorema.

**Teorema no. 1**

Si $(\bar{y}_i, \bar{x}_{ij})$ y $(\bar{\alpha}_j, \bar{\lambda}_{ij})$ son soluciones factibles de un par de programas primario $(P1R)$ y su correspondiente dual $(DP1R)$ Entonces: $G = \sum_{j=1}^{n} \overline{\alpha_j} \leq \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} \overline{x_{ij}} + \sum_{i=1}^{m} f_i \overline{y_i} = Z.$

Prueba:

Sea $\bar{y}_i, \bar{x}_{ij}$ una solución factible para el problema primal $(P1R)$; y sea $\bar{\lambda}_{ij}, \bar{\alpha}_j$ una solución factible para el problema dual $(DP1R)$

Para la solución factible del primal, se cumpliría entonces lo siguiente:

De (3) se tiene que $\bar{x}_{ij} - \bar{y}_i \leq 0$ y de (9) se tiene que $\bar{\lambda}_{ij} \geq 0$, entonces $\bar{\lambda}_{ij}(\bar{x}_{ij} - \bar{y}_i) \leq 0 \quad ...(10)$

De (2) se tiene que $1 - \sum_{i=1}^{m} \bar{x}_{ij} = 0$; por lo tanto, $\sum_{j=1}^{n} \bar{\alpha}_j \left(1 - \sum_{i=1}^{m} \bar{x}_{ij}\right) = 0 \quad ...(11)$

De estas expresiones llegamos a lo siguiente:

$$Z = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} \bar{x}_{ij} + \sum_{i=1}^{m} f_i \bar{y}_i \geq \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} \bar{x}_{ij} + \sum_{i=1}^{m} f_i \bar{y}_i + \sum_{i=1}^{m} \sum_{j=1}^{n} \bar{\lambda}_{ij}(\bar{x}_{ij} - \bar{y}_i) + \sum_{j=1}^{n} \bar{\alpha}_j \left(1 - \sum_{i=1}^{m} \bar{x}_{ij}\right) =$$

$$= \sum_{i=1}^{m} \sum_{j=1}^{n} \left(c_{ij} + \bar{\lambda}_{ij} - \bar{\alpha}_j\right) \bar{x}_{ij} + \sum_{i=1}^{m} \left(f_i - \sum_{j=1}^{n} \bar{\lambda}_{ij}\right) \bar{y}_i + \sum_{j=1}^{n} \bar{\alpha}_j$$

Por lo tanto:

$$Z \geq \sum_{i=1}^{m} \sum_{j=1}^{n} \left(c_{ij} + \bar{\lambda}_{ij} - \bar{\alpha}_j\right) \bar{x}_{ij} + \sum_{i=1}^{m} \left(f_i - \sum_{j=1}^{n} \bar{\lambda}_{ij}\right) \bar{y}_i + \sum_{j=1}^{n} \bar{\alpha}_j \qquad ...(12)$$

Para el problema dual, tenemos que la solución factible cumple con lo siguiente:

De (7) $\left(c_{ij} + \bar{\lambda}_{ij} - \bar{\alpha}_j\right) \geq 0$ y de (4) $\bar{x}_{ij} \geq 0$; por lo tanto $\left(c_{ij} + \bar{\lambda}_{ij} - \bar{\alpha}_j\right) \bar{x}_{ij} \geq 0 \quad ...(13)$.

De (8) tenemos que $\left(f_i - \sum_{j=i}^{n} \bar{\lambda}_{ij}\right) \geq 0$, y de (5) tenemos que $\bar{y}_i \geq 0$, por lo tanto: $\left(f_i - \sum_{j=i}^{n} \bar{\lambda}_{ij}\right) \bar{y}_i \geq 0 \quad ...(14)$

Por lo que se llega a lo siguiente:

$$G = \sum_{j=1}^{n} \bar{\alpha}_j \leq \sum_{j=1}^{n} \bar{\alpha}_j + \sum_{i=1}^{m} \sum_{j=1}^{n} \left(c_{ij} + \bar{\lambda}_{ij} - \bar{\alpha}_j\right) \bar{x}_{ij} + \sum_{i=1}^{m} \left(f_i - \sum_{j=1}^{n} \bar{\lambda}_{ij}\right) \bar{y}_i$$

Por lo tanto:

$$G \leq \sum_{i=1}^{m}\sum_{j=1}^{n}\left(c_{ij}+\bar{\lambda}_{ij}-\bar{\alpha}_j\right)\bar{x}_{ij}+\sum_{i=1}^{m}\left(f_i-\sum_{j=1}^{n}\bar{\lambda}_{ij}\right)\bar{y}_i+\sum_{j=1}^{n}\bar{\alpha}_j \qquad \ldots(15)$$

Comparando las expresiones (12) y (15) llegamos a que: $G \leq Z$.

El teorema 1, solo dice que para cualquier par de soluciones factibles (en el primal y en el dual), la función objetivo del dual (DP1R), es siempre menor o igual a la función objetivo del primal (P1R), es decir, que la solución del dual del problema de localización relajado proporciona la cota inferior y el primal proporciona la cota superior de la solución.

## 5.2. SOLUCIÓN ÓPTIMA DEL *PLSS*.

Cuando los valores de ambas funciones objetivo sean iguales, es decir cuando $G = Z$, entonces se ha encontrado la solución óptima del problema.

**Teorema no. 2**: Dado el par de programas primario $(P1R)$ y su correspondiente dual $(DP1R)$, una condición suficiente para que $\bar{x}_{ij}, \bar{y}_i$ y $\bar{\alpha}_j$ , $\bar{\lambda}_{ij}$, sean óptimas respectivamente del $(P1R)$ y $(DP1R)$, es que se cumplan las condiciones de holgura complementaria:

$$\bar{\lambda}_{ij}\left(\bar{x}_{ij}-\bar{y}_i\right)=0 \qquad \left(c_{ij}+\bar{\lambda}_{ij}-\bar{\alpha}_j\right)\bar{x}_{ij}=0$$

$$\bar{\alpha}_j\left(1-\sum_{i=1}^{m}\bar{x}_{ij}\right)=0 \qquad \text{y} \qquad \left(f_i-\sum_{j=1}^{n}\bar{\lambda}_{ij}\right)\bar{y}_i=0$$

Si suponemos que se cumplen las condiciones de holgura complementaria, se llega a lo siguiente:

$$\sum_{i=1}^{m}\sum_{j=1}^{n}\left(c_{ij}+\bar{\lambda}_{ij}-\bar{\alpha}_j\right)\bar{x}_{ij}+\sum_{i=1}^{m}\left(f_i-\sum_{j=1}^{n}\bar{\lambda}_{ij}\right)\bar{y}_i+\sum_{j=1}^{n}\bar{\alpha}_j=\sum_{i=1}^{m}\sum_{j=1}^{n}c_{ij}\bar{x}_{ij}+\sum_{i=1}^{m}f_i\bar{y}_i=Z$$

Y por parte del dual:

$$\sum_{i=1}^{m}\sum_{j=1}^{n}\left(c_{ij}+\bar{\lambda}_{ij}-\bar{\alpha}_j\right)\bar{x}_{ij}+\sum_{i=1}^{m}\left(f_i-\sum_{j=1}^{n}\bar{\lambda}_{ij}\right)\bar{y}_i+\sum_{j=1}^{n}\bar{\alpha}_j = \sum_{j=1}^{n}\bar{\alpha}_j=G$$

Por lo tanto, $G$=$Z$ y entonces, se ha alcanzado el óptimo; es decir, cumplir con las condiciones de holgura complementaria es suficiente para garantizar el óptimo. Así, que sí $G \neq Z$, es por que se violó alguna condición de holgura complementaria.

# CAPÍTULO VI. SOLUCIÓN DEL PROBLEMA DE LOCALIZACIÓN DE SERVICIOS SIMPLE CON ALGUNOS ALGORITMOS EXISTENTES.

## 6.1 ALGORITMO DE PARTICIÓN (DESCOMPOSICIÓN DE BENDERS).

El procedimiento de partición de Benders "parte" el programa entero mixto en un programa entero y en un programa lineal, consistiendo respectivamente de las variables enteras y continuas del problema original (de aquí el nombre de "Algoritmo de Partición"). El "Algoritmo de Partición" trabaja resolviendo sucesivamente un programa lineal y un programa entero. El programa lineal produce un punto extremo o un rayo extremo y una nueva restricción para el programa entero. También, el valor de la solución óptima del programa lineal proporciona una cota superior para la solución óptima del programa entero mixto. Cuando se resuelve el programa entero, el cual es el equivalente al programa entero mixto cuando este tiene todas sus restricciones, proporciona una cota inferior que no va decreciendo. Cuando las dos cotas coinciden, la solución óptima del programa entero mixto ha sido encontrada y el proceso termina.

La estructura especial del PLSS permite obtenerle soluciones óptimas al programa lineal (*PL*) y a su dual (*DPL*) que aparecen en el algoritmo de partición.

El problema de localización de servicios simple lo plantea [3] como sigue.

$$\left.\begin{array}{llll}
Min\ Z = \sum\limits_{i=1}^{m}\sum\limits_{j=1}^{n} c_{ij}x_{ij} + \sum\limits_{i=1}^{m} f_i y_i & & & \\
sa: & & & \\
 & -\sum\limits_{j=1}^{n} x_{ij} + n_i y_i \ge 0; & & i = 1,\dots,m \\
 & \sum\limits_{i=1}^{m} x_{ij} \quad = 1; & & j = 1,\dots,n \\
 & x_{ij} \quad \ge 0; & & i = 1,\dots,m \\
 & & & j = 1,\dots,n \\
 & y_i \quad = 0,1; & & i = 1,\dots,m
\end{array}\right\} \quad (P1)$$

El significado de las variables se explicó en el capitulo IV.

Para un vector $\mathbf{y}$ fijo cero - uno, el problema $(P1)$ se reduce al problema lineal $(PL)$ siguiente.

$$\left.\begin{array}{lr}
\underset{x_{ij}}{Min} \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} & \\
s.a: & \\
\quad - \sum_{j=1}^{n} x_{ij} \geq -n_i y_i ; & i = 1,\ldots,m \\
\quad \sum_{i=1}^{m} x_{ij} = 1 ; & j = 1,\ldots,n \\
\quad x_{ij} \geq 0 ; & i = 1,\ldots,m \\
 & j = 1,\ldots,n
\end{array}\right\} \quad (PL)$$

Cuyo problema dual $(DPL)$ es el siguiente:

$$\left.\begin{array}{lr}
\underset{u_i, v_j}{Max} \sum_{j=1}^{n} v_j - \sum_{i=1}^{m} n_i y_i u_i & \\
s.a: & \\
\quad v_j - u_i \leq c_{ij} ; & i = 1,\ldots,m \\
 & j = 1,\ldots,n \\
\quad u_i \geq 0 ; & i = 1,\ldots,m \\
\quad v_j \text{ srs} & j = 1,\ldots,n
\end{array}\right\} \quad (DPL)$$

Consideremos el polihedro convexo:

$U = \{(u_1,\ldots,u_m, v_1,\ldots,v_n) : v_j - u_i \leq c_{ij}$ para toda $i, j$; y $u_i \geq 0$ para $i = 1,\ldots,m\}$

el cual es independiente de $\mathbf{y}$, y tiene $P$ puntos extremos y $S$ rayos dirigidos, cada uno teniendo por componentes $\left(u_i^P, v_j^P\right)$ y $\left(u_i^S, v_j^S\right)$.

El programa entero mixto es equivalente al siguiente programa entero $(PE)$.

$$\left.\begin{array}{ll} Min\ Z & \\ s.a: & \\ Z \geq \sum_{i=1}^{m} f_i y_i + \sum_{j=1}^{n} v_j^p - \sum_{i=1}^{m} n_i y_i u_i^p & (p = 1,\dots,P) \\ 0 \geq \sum_{j=1}^{n} v_j^s - \sum_{i=1}^{m} n_i y_i u_i^s & (s = 1,\dots,S) \\ \quad y_i = 0 \text{ ó } 1 & (i = 1,\dots,m) \end{array}\right\} \quad (PE)$$

donde:

$u^p$ : son los puntos extremos $(p = 1,\dots,P)$

$v^s$ : son las rayos extremos dirigidos $(s = 1,\dots,S)$

### 6.1.1 SOLUCIÓN DEL PROGRAMA LINEAL DUAL.

Para resolver el dual del programa lineal,$(DPL)$, primero resolvemos su problema dual $(PL)$. Usando las propiedades de holgura complementaria, los vectores óptimos para las variables duales $u_i$ y $v_j$ serán encontradas.

Para un vector fijo $\mathbf{y}$ $(\mathbf{y}\neq 0)$ cero - uno, el programa lineal $(PL)$ es un problema de transporte el cual puede ser resuelto por inspección. En particular, cuando $y_i = 1$, la planta o el servicio $i$ es abierto y puede atender a cualquiera de los $n$ clientes. Como $c_{ij} \geq 0$ y los envíos no pueden hacerse desde una planta o servicio cerrado (cuando $y_i = 0$), una solución óptima es servir a cada cliente $j$ desde el servicio abierto $i(j)$, con el costo de envío más barato.

Sea $I_1 = \{i : y_i = 1\}$ ...(1)

Una solución óptima al problema lineal $(PL)$ es:

$x_{i(j),j} = 1$ $(j = 1,\dots,n)$

y

$x_{ij} = 0$ en caso contrario.

donde: $c_{i(j),j} = \min_{i \in I_1} c_{ij}$; $(j = 1,\ldots,n)$ ...(2)

Para obtener la solución correspondiente del dual, desde el programa lineal, las propiedades de holgura complementaria implican que $v_j - u_{i(j)} = c_{i(j),j}$ ó $v_j = c_{i(j),j} + u_{i(j)}$; $(j = 1,\ldots,n)$ ...(3)

Sustituyendo $v_j$ en el problema dual $(DPL)$, y notando que $y_i = 1$ sí $i \in I_1$ y $y_i = 0$ en caso contrario, se obtiene el programa lineal equivalente $(DPL')$, como se muestra enseguida.

$$\left.\begin{array}{ll}
Max \ \sum_{j=1}^{n} u_{i(j)} - \sum_{i \in I_1} n_i u_i & \\
s.a: & \\
\quad u_{i(j)} - u_i \le c_{ij} - c_{i(j),j}; & \forall i, \forall j \\
\quad u_i \ge 0 \ ; & i = 1,\ldots,m
\end{array}\right\} \quad (DPL')$$

Donde la constante $\sum_{j=1}^{n} c_{i(j),j}$ ha sido omitida de la función objetivo.

Ahora, para cada $j$; $i(j)$ corresponde a alguna planta $i$, donde $i$ deberá estar en el conjunto $I_1$. Supongamos que definimos $N_i$ como el número de veces que el índice $i$ es un $i(j)$ y como 0 (cero) sí el índice $i$ nunca es un $i(j)$, entonces:

$$\sum_{j=1}^{n} u_{i(j)} = \sum_{i \in I_1} N_i u_i$$

Otra forma de escribir el problema $(DPL')$ es la siguiente:

$$\left.\begin{array}{ll}
Max \ \sum_{i \in I_1} (N_i - n_i) u_i & \\
s.a: & \\
\quad u_i \ge c_{i(j),j} - c_{ij} + u_{i(j)} \ ; & (i = 1,\ldots,m \ \text{y} \ j = 1,\ldots,n) \\
\quad u_i \ge 0 \ ; & (i = 1,\ldots,m)
\end{array}\right\} \quad (DPL'')$$

Ya que $N_i$ puede ser a lo más $n_i$, $N_i - n_i$ es no positivo, y entonces maximizando la función objetivo se logrará que la $u_i$ sea tan pequeña como sea posible. Como $u_{i(j)}$ debe ser no negativa y solamente ésta puede incrementar el valor de $u_i$, una solución óptima del problema $(DPL'')$ y por lo tanto del $(PL)$, tiene a $u_{i(j)} = 0$ para cada $i(j)$. Entonces, para cada índice $i$, $u_i$ deberá ser no negativa y satisfará también $u_i \geq c_{i(j),j} - c_{ij}$ para cada $j = 1,\dots,n$. Por lo tanto, si definimos como:

$$\left.\begin{array}{l} m_i = \max\limits_{j=1,\dots,n} \left\{c_{i(j),j} - c_{ij}\right\} \\ \text{para cada } i = 1,\dots,m \;;\; i \neq i(j) \text{ para toda } j = 1,\dots,n \end{array}\right\} \quad (4)$$

una solución óptima es $u_i = \max\{0, m_i\}$ Resumiendo y usando (3), tenemos que una solución óptima al problema dual $(PDL)$ es:

$$\left.\begin{array}{ll} v_j = c_{i(j),j} \;; & (j = 1,\dots,n) \\ u_{i(j)} = 0 \;; & (j = 1,\dots,n) \\ u_i = \max(0, m_i) & (i = 1,\dots,m;\; i \neq i(j) \text{ para toda } j = 1,\dots,n) \end{array}\right\} \quad (5)$$

Donde $i(j)$ está definida por (2) y $m_i$ por (4).

Una cota superior para la solución óptima para el problema de localización de servicios es obtenida a partir del problema original dada por:

$$Z^u = \sum_{i \in I_1} f_i + \sum_{j=1}^{n} c_{i(j),j} \qquad (6)$$

## 6.2 EJEMPLO ILUSTRATIVO RESUELTO POR EL MÉTODO DE BENDERS

En la tabla (8) se muestran los datos de un problema de tamaño (4X4) tomado de [4], el cual se resolverá por el método de Benders.

| Servicio | Cliente | | | | Costo fijo |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | $f_i$ |
| 1 | 0 | 12 | 20 | 18 | 7 |
| 2 | 12 | 0 | 8 | 6 | 7 |
| 3 | 20 | 8 | 0 | 6 | 7 |
| 4 | 18 | 6 | 6 | 0 | 7 |

Tabla (8) Datos del problema ilustrativo.

El procedimiento que comprende el método de Descomposición de Benders es el siguiente.

Paso 1. Iniciamos con una solución arbitraria, digamos: $\mathbf{y} = (1\ 1\ 0\ 1)$; Con las cotas iniciales: $Z^u = \infty$ , $Z^l = -\infty$

Paso 2. (#1)

Los servicios abiertos son: $I_1 = \{1,2,4\}$; entonces asignamos: $i(1) = 1,\ i(2) = 2,\ i(3) = 4,\ i(4) = 4$

Obtenemos las $v_j$ utilizando (5):

$$v_1 = c_{11} = 0;\ v_2 = c_{22} = 0;\ v_3 = c_{43} = 6;\ v_4 = c_{44} = 0 \rightarrow \mathbf{v} = (0\ 0\ 6\ 0)$$

Obteniendo $u_i$; $u_i = 0$ si $i \in I_1$, es decir $u_1 = u_2 = u_4 = 0$

$$m_i = max(c_{i(j),j} - c_{ij})$$

$u_i = max(0, m_i)$ si $i \notin I_1$, de aquí que $m_3 = 6, \Rightarrow u_3 = 6$

Cálculo de $Z^u$, utilizando (6):

$$Z^u = \sum_{i \in I_1} f_i + \sum_{j=1}^{n} c_{i(j),j} - \sum_{i \in I_1} n_i u_i$$

$$Z^u = f_1 + f_2 + f_4 + c_{11} + c_{22} + c_{43} + c_{44} - n_1 u_1 - n_2 u_2 - n_4 u_4 =$$
$$= 7 + 7 + 7 + 0 + 0 + 6 + 0 - 0 - 0 - 0 = 27$$

Paso 3. (#1) El programa entero es:

$Min\ Z$

$s.a:$

$$Z \geq \sum_{i=1}^{m} f_i y_i + \sum_{j=1}^{n} v_j^p - \sum_{i=1}^{m} n_i y_i u_i^p$$

$y_i = 0,1 \qquad (i = 1,2,3,4)$

Esto es:

$Min\ Z$

$s.a:$

$$Z \geq f_1 y_1 + f_2 y_2 + f_3 y_3 + f_4 y_4 + (v_1 + v_2 + v_3 + v_4) - [4y_1(0) + 4y_2(0) + 4y_3(6) + 4y_4(0)]$$

Sustituyendo:

$$Z \geq 7y_1 + 7y_2 + 7y_3 + 7y_4 + (0 + 0 + 6 + 0) - 24y_3$$

Simplificando:

$$Z \geq 6 + 7y_1 + 7y_2 - 17y_3 + 7y_4$$

Quedando el siguiente problema:

$$
\begin{aligned}
&Min \quad Z \\
&s.a: \\
&\qquad Z \geq 6 + 7y_1 + 7y_2 - 17y_3 + 7y_4 \\
&\qquad y_i = 0,1
\end{aligned}
$$

Por inspección hallamos que la solución es: $\mathbf{y} = (0\ \ 0\ \ 1\ \ 0)$, con $Z^l = -11 < 27 = Z^u$

Continuando de esta manera, se tuvieron que realizar 12 iteraciones para llegar a la solución óptima. Enseguida se presenta la última iteración que se realizó para poder llegar a la solución óptima.

Paso 2. (#12) Con $\mathrm{y} = (1 \quad 0 \quad 1 \quad 1)$ obtenida en la iteración #11.

$I_1 = \{1,3,4\}; \quad i(1) = 1,\ i(2) = 4,\ i(3) = 3,\ i(4) = 4$

$v_1 = c_{11} = 0,\ v_2 = c_{42} = 6,\ v_3 = c_{33} = 0,\ v_4 = c_{44} = 0$

$u_1 = 0,\ u_2 = 6,\ u_3 = 0,\ u_4 = 0$

Con $Z = f_1 + f_3 + f_4 + (0 + 6 + 0 + 0) = 27 > 26 = Z^u \quad \therefore \quad Z^u = 26$

Paso 3. (#12)

$$
\begin{aligned}
&Min \quad Z \\
&s.a: \\
&\qquad Z \geq 6 + 7y_1 + 7y_2 - 17y_3 + 7y_4 \\
&\qquad Z \geq 34 - 73y_1 - 25y_2 + 7y_3 - 17y_4 \\
&\qquad Z \geq 14 + 7y_1 - 25y_2 + 7y_3 - 17y_4 \\
&\qquad Z \geq 12 - 41y_1 + 7y_2 + 7y_3 + 7y_4 \\
&\qquad Z \geq 6 + 7y_1 + 7y_2 + 7y_3 - 17y_4 \\
&\qquad Z \geq 7y_1 + 7y_2 + 7y_3 + 7y_4 \\
&\qquad Z \geq 30 - 65y_1 - 17y_2 - 17y_3 + 7y_4 \\
&\qquad Z \geq 26 - 41y_1 + 7y_2 - 25y_3 - 17y_4 \\
&\qquad Z \geq 12 + 7y_1 - 17y_2 - 17y_3 + 7y_4 \\
&\qquad Z \geq 14 + 7y_1 + 7y_2 - 25y_3 - 17y_4 \\
&\qquad Z \geq 50 + 7y_1 - 41y_2 - 73y_3 - 65y_4 \\
&\qquad Z \geq 6 + 7y_1 - 17y_2 + 7y_3 + 7y_4 \\
&\qquad y_i = 0,1
\end{aligned}
$$

La solución de este programa entero con 12 restricciones es: $y = (1\ 0\ 0\ 1)$ ; con $Z^l = 26 = 26 = Z^u$

Como vemos, se ha llegado a $Z^l = Z^u$, lo cual nos indica que se ha encontrado la solución óptima.

En el apéndice A se presentan todas las iteraciones realizadas para el ejemplo ilustrativo del Problema de Localización de Servicios Simple resuelto por el método de Partición de Benders.

Este método puede resultar muy costoso, por eso se siguen buscando otros métodos que sean más eficientes.

## 6.3 MÉTODO DE RAMIFICACIÓN Y ACOTAMIENTO.

Este método aplicado al Problema de Localización de Servicios Simple, fue propuesto por Efroymson, M., y T. Ray [7]. El problema primero se resuelve como un programa lineal (sin los requerimientos sobre las $y_i$ a ser enteras) proporcionando un valor, $Z_0$. Si todas las $y_i$ son enteras, entonces la solución se ha encontrado. Si alguna $y_k$ es fraccionaria entonces ésta se fija a ser cero y el programa lineal otra vez a resolver proporcionará $Z_1$, y luego se fija esa $y_k$ a ser uno y el programa lineal resuelto producirá $Z_2$.

Una nueva cota inferior sobre el valor de la solución será $\overline{Z} = min(Z_1, Z_2)$, $\overline{Z} \geq Z_0$. Se construye un árbol cuyos nodos están representados por las $Z$ y los correspondientes valores de las $y$ fijas. Ahora se ramifica sobre el nodo determinado por $\overline{Z}$ fijando alguna otra $y_r$ fraccionaria, primero fijándola a cero, entonces se determinan los nodos $Z_3$ y $Z_4$. Se continúa ramificando sobre: $\overline{Z} = min(Z_3, Z_4, Z')$, $Z' = max(Z_1, Z_2)$ La cual es una nueva cota inferior. Obviamente, necesitamos guardar solamente los resultados de los nodos "terminales", y si algún nodo es no factible, no podrán emanar ramas de él. El proceso termina cuando un nodo ha alcanzado que todas las $y$ sean enteras y su valor sea menor o igual que cualquier otro nodo terminal. La principal dificultad con ramificación y acotamiento es de carácter computacionaL.

Si un gran número de programas lineales se tuvieron que resolver, el tiempo computacional para cada programa lineal es alto, y esto ocasionará que este método resulte ser caro. Para tratar de evitar un poco este tiempo perdido en la solución de los programas lineales, se recomienda la formulación del problema con el conjunto de restricciones "agregadas" que se mencionaron en el capítulo IV.

Esta formulación es:

$$MinZ = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} + \sum_{i=1}^{m} f_i y_i$$

$s.a:$

$$\sum_{i=1}^{m} x_{ij} = 1 \;; \qquad (j = 1,\ldots,n)$$

$$\sum_{j=1}^{n} x_{ij} \leq n_i y_i \;; \qquad (i = 1,\ldots,m)$$

$$x_{ij} \geq 0 \;; \qquad (i = 1,\ldots,m \text{ y } j = 1,\ldots,n)$$

$$y_i = 0 \text{ ó } 1 \;; \qquad (i = 1,\ldots,m)$$

## 6.4 EJEMPLO ILUSTRATIVO RESUELTO POR EL MÉTODO DE RAMIFICACIÓN Y ACOTAMIENTO.

Resolveremos enseguida el ejemplo ilustrativo que se presentó en la sección 6.2 de este capítulo, el cual fue resuelto por el método de Descomposición de Benders.

Los datos para este problema se dan nuevamente en tabla (9).

| Servicios | Clientes | | | | Costo fijo |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | $f_i$ |
| 1 | 0 | 12 | 20 | 18 | 7 |
| 2 | 12 | 0 | 8 | 6 | 7 |
| 3 | 20 | 8 | 0 | 6 | 7 |
| 4 | 18 | 6 | 6 | 0 | 7 |

Tabla (9). Datos para el ejemplo ilustrativo

La solución del programa lineal es $Z_o = 7$, con $y_1 = y_2 = y_3 = y_4 = 0.25$

Como no es solución entera, entonces se ramifica obteniéndose el árbol de la figura (10).

La búsqueda resultó exhaustiva, llegándose a resolver 22 problemas. Este método también resulta muy costoso para problemas grandes de Localización de Servicios.

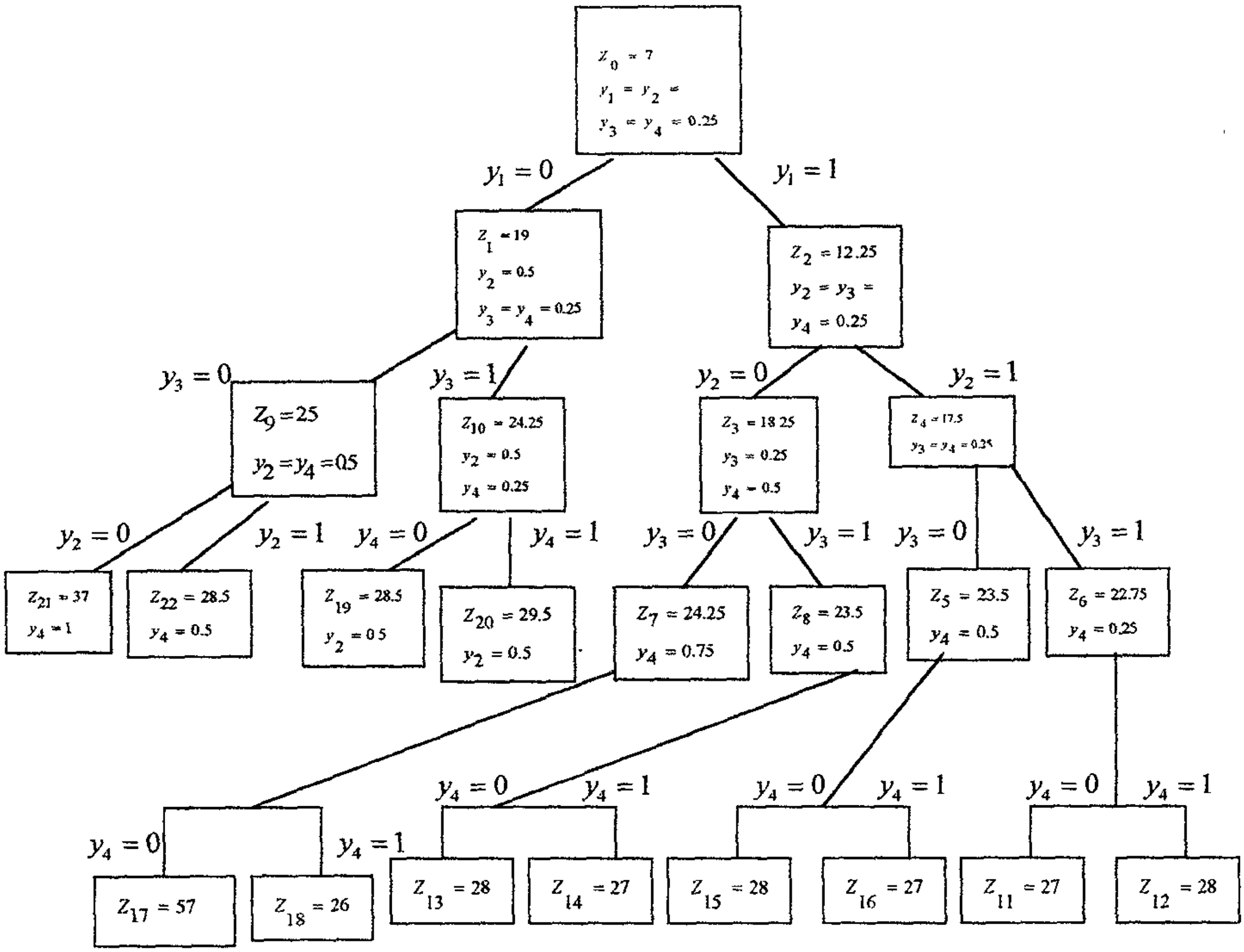



Figura (10). Diagrama en bloques del método de ramificación y acotamiento aplicado al ejemplo ilustrativo. La solución óptima se encontró en el nodo correspondiente a la $Z_{18} = 26$.

El subíndice de la $Z$ indica el orden en que se fueron resolviendo los subproblemas.

# CAPÍTULO VII. DESCRIPCIÓN DEL ALGORITMO PROPUESTO.

En este capítulo se describe el procedimiento de cada una de las fases que involucra nuestro algoritmo para resolver el Problema de Localización de Servicios Simple. Como ya se mencionó en capítulos anteriores, nuestro algoritmo consta de tres fases que llamamos:

a) Fase I: Nivelación Dual

b) Fase II: Pulido del Primal

c) Fase III: Ramificación por Selección

## 7.1 PROCEDIMIENTO DE LA NIVELACIÓN DUAL.

Para poder dar los pasos que componen a esta fase del algoritmo propuesto, enseguida se describe el significado de cada una de las variables involucradas en el dual del problema, las cuales se presentaron en el capítulo V.

$\alpha_j$ : Es la cantidad que va a pagar en total el cliente $j$.

$\alpha_{jsegunda}$ : Es el costo que paga el cliente $j$ si estuviera asignado al servicio que le sigue al más barato hasta ese momento.

$\alpha_{jmax}$ : Es la cota superior de lo que cada cliente $j$ pagaría en total.

$\lambda_{ij}$ : Es la cantidad con la que coopera el cliente $j$ para el pago del costo fijo del servicio $i$.

$\Delta_j$ : Es la parte del costo fijo que está dispuesto a pagar el cliente $j$ sin que le convenga cambiarse de servicio.

Los pasos a seguir son:

1. Inicio de las cotas inferior, segunda y superior para el costo total que paga cada cliente.

La nivelación dual se inicia con $\lambda_{ij} = 0$ (es decir, hasta ese momento el cliente $j$ no ha cooperado para pagar el costo fijo del servicio $i$), de aquí, que el valor inicial de $\alpha_{jmin}$ sea el $\min_i c_{ij}$, para cada $j$. Las $i$ que tuvieron el $\min_i c_{ij}$ se agruparán en el conjunto $I_j^{cm}$. Por otro lado, se inician las $\alpha_{jsegunda}$ con el $\min_{i \notin I_j^{cm}} c_{ij}$, para cada $j$, y se inicia la cota superior de estos costos con $\alpha_{jmax} = \infty$, para cada $j$.

2. Obtener el vector de diferencias, $\Delta_j$, definido como $\Delta_j = \left(\alpha_{jsegunda} - \alpha_{jmin}\right)$, para cada $j$. Estas diferencias pueden ser de alguna de las siguientes dos formas:

a) $\Delta_j = 0$

Este caso se presentaría en alguna iteración después de iniciado el procedimiento, ya que al inicio de la Nivelación Dual, $\Delta_j$ nunca puede ser cero. Si en el transcurso del procedimiento se llega a que $\Delta_j = 0$, entonces se excluirán a esas $j$ de los siguientes pasos, pues cuando $\Delta_j = 0$, se termina la fase de Nivelación Dual para esas $j$. Sí todas las $\Delta_j$ tienen valor de cero, entonces continuar con el paso 5.

b) $\Delta_j > 0$

Para aquellas $j$ que tienen $\Delta_j > 0$, buscar la $\Delta_j$ más grande, llamada $\Delta_{jmax}$. A las $j$ que tienen $\Delta_{jmax}$, las agruparemos en el conjunto $j \in J^M$. Sí las $j \in J^M$ son más de una, encontrar aquella $j$ cuya cardinalidad de $I_j^{cm}$ sea mínima, llamarle a esa $j$, $j^{\Delta}$. Es decir, que $j^{\Delta}$ es la columna que tiene $\Delta_{jmax}$ y menos veces repetido el costo $c_{ijmin}$. Cuando hay empate en la cardinalidad de $I_j^{cm}$, tomar arbitrariamente alguna de esas $j$ con $\Delta_{jmax}$. Si solamente una $j$ tuvo $\Delta_{jmax}$, entonces esa $j$ será $j^{\Delta}$.

3. Para $j^{\Delta}$ encontrar el mínimo costo fijo restante de las $i \in I_j^{cm}$, esto es, el $\min_{i \in I_j^{cm}} f_i$, a este costo fijo se le llama $f_i^r$, y hacer lo siguiente según sea alguno de los siguientes casos:

a) Si $f_i^r \leq \Delta_{jmax}$, se hacen las siguientes actualizaciones:

$f_{inuevo} = f_{iactual} - f_i^r$ , para $i \in I_j^{cm}$

$c_{ijnuevo} = c_{ijactual} + f_i^r$ , para $i \in I_j^{cm}$ y para $j = j^\Delta$

$\alpha_{jmin}nueva = \alpha_{jmin}actual + f_i^r$ , para $i \in I_j^{cm}$ y para $j = j^\Delta$

Si todos los $f_{inuevo} \neq 0,$ entonces actualizar $\alpha_{jsegunda}$ de la siguiente forma:

$\alpha_{jsegunda}nueva = \min_i c_{ij}$ , para $i \notin I_j^{cm}$ y para $j = j^\Delta$

Para aquellas $i$ que tienen $f_{inuevo} = 0,$ entonces llamarle $i^0$ a esa $i$, y continuar con las siguientes actualizaciones para $\alpha_{jsegunda}$ y para $\alpha_{jmax}$.

$\alpha_{jmax}nueva = min\{\alpha_{jmax}actual, max\{\alpha_{jmin}nueva, c_{ij}nuevo\}\}$, para $i = i^0$ y para toda $j$.

$\alpha_{jsegunda}nueva = min\{\alpha_{jsegunda}nueva, \alpha_{jmax}nueva\}$, para toda $j$.

Cuando un $f_i nuevo = 0,$ entonces se ha cubierto todo el costo fijo de ese servicio $i$ y entonces se considera como "servicio abierto". Las $i$ que cumplan con esto, se les agrupa en el conjunto $I^a$, una $i \in I^a$ indica que el costo fijo de esa $i$ ya se pagó.

b) Sí $f_i^r > \Delta_{jmax}$, para $i \in I_j^{cm}$

Se actualiza de la siguiente forma:

$f_i nuevo = f_i - \Delta_{jmax}$, para $i \in I_j^{cm}$

$c_{ij}nuevo = c_{ij}actual + \Delta_{jmax}$ , para $i \in I_j^{cm}$ y $j = j^\Delta$

Cuando algún $c_{ij}$ sea igual a $c_{ij}nuevo,$ incluir en el conjunto $I_j^{cm}$ a esa(s) $i$.

$\alpha_{jmin}nueva = \alpha_{jmin}actual + \Delta_{jmax}$ , para $j = j^\Delta$

* $\boxed{\alpha_{jsegunda}nueva = \min_i c_{ij} \text{ , para } i \notin I_j^{cm} \text{ y } j = j^\Delta}$

*Al actualizar $\alpha_{jsegunda}$, se puede presentar que el $\min_{\substack{i \notin I \\ j}} cm \, c_{ij} > \alpha_{j\Delta max}$, en estos casos

se hace: $\alpha_{j\Delta segunda} = \alpha_{j\Delta max}$

4. Regresar al paso 2.

5. Calcular el valor de la función objetivo del dual, $G = \sum_{j=1}^{n} \alpha_j$, donde

$\alpha_j = \alpha_{jmin} = \alpha_{jsegunda} = \alpha_{jmax}$

**Ejemplo1.**

Enseguida se presentan los datos de un problema tomado de [9] para aplicarle la fase I: Nivelación Dual del algoritmo propuesto.

En la tabla (10) se dan los costos variables $c_{ij}$, y los costos fijos, $f_i$, para un problema de (6X4).

| Servicios | Clientes, $j$ | | | | Costo fijo |
|---|---|---|---|---|---|
| $i$ | 1 | 2 | 3 | 4 | $f_i$ |
| 1 | 6 | 6 | 5 | 2 | 3 |
| 2 | 6 | 8 | ∞ | 3 | 2 |
| 3 | 8 | 6 | 3 | ∞ | 2 |
| 4 | 6 | ∞ | 6 | 2 | 2 |
| 5 | ∞ | 6 | 3 | 4 | 3 |
| 6 | 6 | 6 | ∞ | 4 | 3 |

Tabla (10) Costos para el ejemplo 1

Se inicia con:

$\alpha_{jmin} = (6 \quad 6 \quad 3 \quad 2)$

$\alpha_{jsegunda} = (8 \quad 8 \quad 5 \quad 3)$

$\alpha_{jmax} = (\infty \quad \infty \quad \infty \quad \infty)$

El vector de diferencias entre $\alpha_{jsegunda}$ y $\alpha_{jmin}$ es:

$\Delta_j = (2 \quad 2 \quad 2 \quad 1)$

Donde $\Delta_{jmax} = 2$

En este caso hay varias $j$ que coincidieron en tener la máxima diferencia, $\Delta_{jmax}$. Entonces, el conjunto $J^M$ de las $j$ que tienen $\Delta_{jmax} = 2$, es: $J^M = \{1,2,3\}$.Se deberá desempatar con la cantidad de costos mínimos que tiene cada una de estas $j$. En este caso iniciará la primer iteración en la cuarta columna, es decir que $j^\Delta$ es la correspondiente a $j$=4, ya que ésta tiene un solo costo mínimo. En este caso $f_i^r = f_4 = 2$.

Las actualizaciones obtenidas en cada iteración se resumen en la tabla (11).

1. Actualizaciones de los costos variables, $c_{ij}$ y los costos fijos, $f_i$.

| Servicios | Clientes | | | | Costo Fijo |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | |
| 1 | 6→**8** | 6 | 5 | 2 | 3→**1** |
| 2 | 6→**8** | 8 | ∞ | 3 | 2→**0** |
| 3 | 8 | 6 | 3→**5** | ∞ | 2→**0** |
| 4 | 6→**8** | ∞ | 6 | 2 | 2→**0** |
| 5 | ∞ | 6 | 3→**3** | 4 | 3→**1** |
| 6 | 6→**8** | 6 | ∞ | 4 | 3→**1** |

Tabla (11) Actualizaciones de los costos variables y fijos del ejemplo 1.

El número en negrita es el valor en el cual cambió el dato original al ir aplicando el procedimiento de la nivelación dual, es decir, son los $c_{ijnuevos}$ y los $f_{inuevo}$ en su respectivo lugar, indicando la flecha el cambio que ocurrió.

2. Las actualizaciones de los vectores $\alpha_{jmin}$, $\alpha_{jsegunda}$, $\alpha_{jmax}$ y $\Delta_j$ se dan a continuación:

$$\alpha_{jmin} = (6 \quad 6 \quad 3 \quad 2)$$
$$(6 \quad 6 \quad 5 \quad 2)$$
$$(8 \quad 6 \quad 5 \quad 2)$$
$$\alpha_{jsegunda} = (8 \quad 8 \quad 5 \quad 3)$$
$$(8 \quad 6 \quad 5 \quad 3)$$
$$(8 \quad 6 \quad 5 \quad 2)$$
$$\alpha_{jmax} = (\infty \quad \infty \quad \infty \quad \infty)$$
$$(8 \quad 6 \quad 5 \quad \infty)$$
$$(8 \quad 6 \quad 5 \quad 2)$$
$$\Delta_j = (2 \quad 2 \quad 2 \quad 1)$$
$$(2 \quad 0 \quad 0 \quad 1)$$
$$(0 \quad 0 \quad 0 \quad 0)$$

Las iteraciones terminan hasta que el vector de las diferencias, $\Delta_j$, tenga en todos sus elementos, el valor de cero. Es en este momento en que se llega a que: $\alpha_{jmin} = \alpha_{jsegunda} = \alpha_{jmax}$.

La fase de Nivelación Dual nos proporciona la siguiente solución del dual:

$$G^* = \sum_{j=1}^{4} \alpha_{jmin} = \sum_{j=1}^{4} \alpha_{jsegunda} = \sum_{j=1}^{4} \alpha_{jmax} = 8+6+5+2 = 21$$

Los servicios que la fase I indica abrirlos, son aquellos que terminaron con un costo fijo con valor de cero. En este ejemplo, los servicios que terminaron con costo fijo nulo, se agrupan en el conjunto $I^a$, que se definió, como el conjunto de los servicios abiertos obtenidos en la fase I de nuestro algoritmo, siendo entonces: $I^a = \{2,3,4\}$

## 7.1.1 CONSERVACIÓN DE LA FACTIBILIDAD DUAL EN EL PROCEDIMIENTO DE NIVELACIÓN DUAL

Enseguida se vuelve a presentar la formulación fuerte del Problema de Localización de Servicios Simple y su correspondiente dual, que se presentaron en el capítulo V.

El problema relajado es el siguiente.

$$
\left.\begin{aligned}
&MinZ = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} + \sum_{i=1}^{m} f_i y_i && (1)\\
&s.a:\\
&\qquad \sum_{i=1}^{m} x_{ij} = 1 \quad ; \quad \forall j && (2)\\
&\qquad y_i - x_{ij} \geq 0 \quad ; \quad \forall i, \forall j && (3)\\
&\qquad x_{ij} \geq 0 \quad ; \quad \forall i, \forall j && (4)\\
&\qquad y_i \geq 0 \quad ; \quad \forall i && (5)
\end{aligned}\right\} (P1R)
$$

El dual para $(P1R)$ es el siguiente:

$$
\left.\begin{aligned}
&MaxG = \sum_{j=1}^{n} \alpha_j && (6)\\
&s.a:\\
&\qquad \alpha_j - \lambda_{ij} \leq c_{ij} \quad ; \forall i, \forall j && (7)\\
&\qquad \sum_{j=1}^{n} \lambda_{ij} \leq f_i \quad ; \quad \forall i && (8)\\
&\qquad \lambda_{ij} \geq 0 \quad ; \quad \forall i, \forall j && (9)
\end{aligned}\right\} (DP1R)
$$

La fase de Nivelación Dual inicia con $\lambda_{ij} = 0$, la cual cumple con (9). Los valores iniciales de $\alpha_{jmin}$, se obtienen como $\alpha_{j\text{mín}} = \min_i c_{ij}$, para cada $j$, para cumplir con la factibilidad y tratar de que sean lo más grande posible ya que como lo indica la función objetivo, se están maximizando las $\alpha_j$.

Aquellas restricciones del tipo $\alpha_j - \lambda_{ij} \leq c_{ij}$ que se estén cumpliendo con igualdad les llamaremos *restricciones activas.* Como se desea $Max\sum_j \alpha_j$, tratamos de incrementar cada $\alpha_j$, y para que sigan siendo factibles las restricciones activas, se deberá incrementar su correspondiente $\lambda_{ij}$ en la misma cantidad que $\alpha_j$, pero sin rebasar el costo fijo,

$f_i$, correspondiente; por eso cuando se escoge el aumento para alguna $\alpha_j$, se debe cuidar que ese aumento sea una cantidad menor o igual que $\left(f_i - \sum_j \lambda_{ij}\right)$, por otro lado, si se aumenta mucho $\alpha_j$ puede requerirse aumentar $\lambda_{ij}$ para alguna $i$ que no era activa, a fin de conservarse la factibilidad, mejor lo que se hace es cuidar que la $\alpha_j$ no crezca más que los $c_{ij}$ (los $c_{ij}$ que no han sido nivelados) que son mayores que el valor actual de $\alpha_j$, para eso se busca cuál es el $c_{ij}$ inmediato superior a esa $\alpha_j$, a la cual se le llama $\alpha_{jsegunda}$. La diferencia que hay entre $\alpha_{jsegunda}$ y $\alpha_{jmin}$ $\left(\Delta_j = \alpha_{jseunda} - \alpha_{jmin}\right)$, nos indica qué tanto podría aumentar esa $\alpha_j$ sin tener que aumentar alguna $\lambda_{ij}$ que no pertenece a las restricciones activas.

La razón de escoger la $\Delta_j$ más grande (a esta $\Delta_j$ la llamamos $\Delta_{jmax}$) es por que se desea repartir el (los) costo(s) fijo(s) correspondiente(s) lo más rápido posible.

En resumen, las $\alpha_j$ aumentan en cada iteración a lo más una cantidad igual a $\Delta_{jmax}$ para que se conserve la restricción (7) y a lo más el $\min_i f_i$ de los servicios $i$ que corresponden a las restricciones activas, (a estos costos fijos los llamamos $f_i^r$) esto con el fin de que la restricción (8) siga siendo factible. De este modo, en cada iteración se mantiene la factibilidad y cuando ninguna $\alpha_j$ pueda aumentar, se sabrá que las $\alpha_j$ y las $\lambda_{ij}$ son factibles del problema dual.

## 7.2 PULIDO DEL PRIMAL

A partir de la solución obtenida en la *Nivelación Dual*, se busca una solución factible para el programa primal. Para obtener esta solución del primal, aplicamos la fase, llamada *Pulido del Primal*. Con esta fase se conocerá la *cota superior* de la solución del PLSS. Si esta cota superior es **igual** a la solución proporcionada por la nivelación dual (el dual proporciona la *cota inferior*), entonces se habrá encontrado la **solución óptima** del PLSS. En caso contrario se aplicará la fase de Ramificación por Selección.

Los pasos son los siguientes:

1. Encontrar el $\min_{i \in I^a} c_{ij}$, para cada $j$, las $i$ donde están los $\min_{i \in I^a} c_{ij}$, agruparlas en el conjunto $I_j^{acm}$.

2. Buscar las $j$ con menor cardinalidad en $I_j^{acm}$.

3. El inicio de las asignaciones dependerá del caso que presente la cardinalidad de $I_j^{acm}$. Estos casos se enlistan a continuación.

a) Si hay únicamente una $j$ con cardinalidad de $I_j^{acm}$ igual a uno, comenzar la asignación con esa $j$. Hacer la asignación como lo indica el paso 4.

b) Si hay más de una $j$ con cardinalidad de $I_j^{acm}$ igual a uno, entonces preguntar para cada una de esas $j$ si el $\min_{i \notin I_j^{acm}} c_{ij}$ es igual a infinito, si es así asignar a esas $j$ con su correspondiente $i \in I_j^{acm}$, agrupar a esas $j$ en el conjunto llamado "asignadas"; y si no (si el $\min_{i \notin I_j^{acm}} c_{ij} \neq \infty$) comenzar la asignación en cualquier $j$ que tuvo la cardinalidad de uno en $I_j^{acm}$. Hacer la asignación como lo indica el paso 4.

c) Si ninguna $j$ tuvo un solo $\min_{i \in I^a} c_{ij}$, entonces buscar alguna $j$ con la cardinalidad menor de $I_j^{acm}$ y comenzar la asignación en esa $j$ como se indica en el paso 4.

4. La asignación de cada $j$ se hará en la posición donde esté el $\min_{i \in I^a} c_{ij}$, a la $i$ correspondiente a esa asignación, agruparla en el conjunto "abiertos", etiquetando como "*primer abierto*" a la $i$ que primero entró a ese conjunto.

5. Cada que se asigne alguna $j$, preguntar si esa $j$ había cooperado en otros servicios $i$, es decir en los $i \notin$"*abiertos*".

5.1 Si no cooperó esa $j$ en otro $i$, entonces preguntar que otra $j$ ayudó a pagar el costo fijo del servicio $i$ que se acaba asignar a la $j$, si hubo alguna $j$ que ayudó, entonces asignar (pasar a esa $j$ en el conjunto "asignadas") a esa $j$ que ayudó con el pago del costo fijo del servicio $i$ que acaba de ser "abierto". Regresar al inicio del paso 5.

5.2 Si esa $j$ que se acaba de asignar cooperó en otros servicios, entonces "cerrar" a cada uno de esos servicios $i \notin$"*abiertos*" y agruparlos en el conjunto "cerrados". Cada que se va a cerrar un servicio $i$, preguntar para cada $j \notin$"*asignadas*", si al cerrar esos servicios $i$, el único $c_{ij}$ que existe es infinito, entonces no se podrán cerrar todos los servicios $i$ donde había cooperado la $j$ que se acaba de asignar, en este caso, se cerrarán los $i$ que al cerrarlos sí se permita asignar al cliente $j$ que se desea asignar en ese momento, en una posición donde no están los $c_{ij} = \infty$, tratando a su vez que la asignación de esa nueva $j$ sea donde esté su $c_{ij}$ más barato. Si al cerrar un servicio $i$, los clientes $j \notin$"*asignados*" no tienen como único $c_{ij}$, un $c_{ij} = \infty$, entonces sí se podrán cerrar esos servicios.

6. Ya que se cerraron los servicios $i$ donde había cooperado el cliente $j$ que acaba de ser asignado, preguntar si alguna otra $j$, de las $j \notin$"*asignadas*", le ayudó a pagar una parte del costo fijo del servicio $i$ que se le acaba de asignar a esa $j$, si se encontró que alguna $j$ le ayudó a pagar, entonces asignar a esa $j$ que ayudó y agruparla en el conjunto "*asignadas*", y regresar al inicio del paso 5.

Si ninguna otra $j$, de las $j \notin$"*asignadas*", cooperó, entonces continuar el proceso de asignación con aquella $j$ cuya cardinalidad de $I_j^{acm}$ sea la menor y continuar con el paso 3.

Sí todas las $j$ ($j$=1,2,3,...,$n$) se han asignado a algún servicio de los $i \in I^a$, entonces el proceso del Pulido del Primal ha terminado.

**Ejemplo 2.**

Para mostrar cómo funciona la segunda fase del algoritmo, la aplicaremos al ejemplo 1 que se presentó en la sección 7.1.

Al aplicar la fase I del algoritmo, obtuvimos que los servicios abiertos son las $i$ que agrupamos en el conjunto $I^a = \{2,3,4\}$. Con el fin de facilitar el procedimiento que sigue, formamos la submatriz para los servicios que interesa tener abiertos hasta ahora. Esto se muestra en la tabla 12.

| Servicio $i$ | Clientes, $j$ | | | | Costo fijo $f_i$ |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | |
| 2 | **6**→8 | 8 | $\infty$ | 3 | 2 |
| 3 | 8 | **6** | **3**→5 | $\infty$ | 2 |
| 4 | **6**→8 | $\infty$ | 6 | **2** | 2 |
| $I_j^{acm}$ | $I_1^{acm}=\{2,4\}$ | $I_2^{acm}=\{3\}$ | $I_3^{acm}=\{3\}$ | $I_4^{acm}=\{4\}$ | |
| Cardinalidad de $I_j^{acm}$ | 2 | 1 | 1 | 1 | |

Tabla (12) Submatriz de los servicios abiertos obtenidos en la fase I.

Donde $I_j^{acm}$ es el conjunto de los servicios que atienden al cliente $j$ donde están los $c_{ij}$ más pequeños tomando en cuenta únicamente los servicios que la fase I indicó abrir. Por ejemplo, el cliente $j$=1 tiene su costo mínimo en la posición correspondiente a los servicios 2 y 4, teniendo entonces este conjunto la cardinalidad de 2.

Los costos mínimos de cada columna se han puesto en negrita.

Para aquellas $I_j^{acm}$ que tienen la cardinalidad más pequeña, en este caso, las de cardinalidad de 1, aplicar la **primer pregunta**: ¿El $\min_{i \notin I_j^{acm}} c_{ij} = \infty$?. Si esto ocurre se inicia la asignación en esa $j$. En este caso ninguna de las $j$ presentó esta situación. Como no tenemos esos valores, se comienza la asignación en cualquier $j$ que tuvo la cardinalidad de 1 en su $I_j^{acm}$. En este ejemplo, se inicia la asignación en la posición correspondiente al $c_{32}$; y al iniciar en esta posición, entonces etiquetamos como "**primer abierto**" al servicio $i$=3, quedando asignado el cliente 2 al servicio 3.

La **segunda pregunta** que se hace es: Esa $j$ que acaba de asignarse, ¿En qué otro servicio había cooperado en el desarrollo de la Nivelación Dual?, En este ejemplo se ve que esa $j$ no cooperó en ningún otro servicio (si alguien cooperó, se ve en la tabla anterior cuando el costo $c_{ij}$ tiene flecha). En este caso, en la columna correspondiente al cliente $j$=2, no tienen flecha ninguno de los $c_{ij}$.

La **tercer pregunta** es: ¿Qué otro cliente ayudó a pagar el costo fijo del servicio que se acaba de abrir?. Vemos que el cliente 3 cooperó con $2 para pagar el costo fijo del servicio 3, y por lo tanto, la siguiente asignación ocurre en la posición correspondiente al $c_{33}$, quedando asignado el cliente 3 al servicio 3.

Cuando se asigna al tercer cliente, se regresa a las dos primeras preguntas, y en este caso se completa el ciclo de preguntas para el servicio 3.

Como no han sido asignados todos los clientes, se regresa en busca de la $I_j^{acm}$ con menor cardinalidad, que en este caso, la tiene el cliente 4.

Entonces, la tercer asignación ocurre en la posición correspondiente al $c_{44}$, quedando asignado el cliente 4 con el servicio 4. Llegamos otra vez al punto donde se deben aplicar las tres preguntas. Primer pregunta: ¿Qué otro cliente cooperó para pagar el costo fijo del servicio que acaba de abrirse (en este caso el servicio 4 ha sido el segundo en abrirse)?. Vemos que el cliente 1 cooperó con $2, entonces la cuarta asignación ocurre en la posición $c_{41}$, quedando asignado el cliente 1 al servicio 4. Continuando con la segunda pregunta: ¿En dónde más había cooperado el cliente que se acaba de asignar?, Vemos que el cliente 1 había cooperado en el servicio 2, por lo tanto, se deberá "cerrar" ese servicio, la razón es por que ningún cliente puede ser atendido por más de un servicio. En este ejemplo, se cierra el servicio 2 que la fase I había definido como abierto.

Las asignaciones terminan cuando todas las $j$ ya han sido asignadas a algún servicio. En este ejemplo, ya se ha asignado a todas las $j$.

La fase de Pulido del Primal nos indicó abrir a los servicios $\{3,4\}$

Hasta aquí, ya podemos obtener la solución del problema primal:

$$Z^* = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} + \sum_{i=1}^{m} f_i \quad \forall \quad i \in \text{"}abiertos\text{"} \;\; \text{y} \;\; j \in \text{"}asignadas\text{"}$$

$$Z^* = c_{41} + c_{32} + c_{33} + c_{44} + f_3 + f_4 = 6+6+3+2+2+2 = 21$$

Al final de esta fase, se compara la solución obtenida del problema dual con la solución obtenida del problema primal. En este ejemplo, la solución del dual que se obtuvo en la fase de Nivelación Dual, fue $G^* = 21$. Al comparar las dos soluciones, llegamos a que $Z^* = G^* = 21$. Y por lo tanto, se ha encontrado la **solución óptima** del **problema original.**

## 7.3 RAMIFICACIÓN POR SELECCIÓN

Cuando la solución del dual, $G$, obtenida en la fase I: "Nivelación Dual", es diferente a la solución del primal, $Z$, obtenida en la fase II: "Pulido del Primal", entonces se deberá aplicar la fase III: "Ramificación por Selección".

El proceso de ramificación por selección se hace **abriendo o cerrando** los servicios $i$ etiquetados en la fase del Pulido del Primal como "*primer abierto*". La ramificación inicia primero abriendo el servicio llamado "*primer abierto*".

Cuando se ramifica abriendo un servicio, el costo fijo de ese servicio se deberá cambiar de su valor original a un costo fijo nulo, $f_i = 0$. Ya que como se indicó anteriormente, un costo fijo de cero, indica que ese servicio está abierto. Cuando se ramifica cerrando un servicio, el costo fijo de ese servicio se deberá cambiar de su valor original a un costo fijo muy caro, $f_i = \infty$. Cada que se hace un cambio en los costos fijos de algún servicio por estar abriendo o cerrando, se regresará el "nuevo" problema a las fases I y II para poder conocer las nuevas soluciones del dual y primal. Si se mejora la $Z$, se guarda como la mejor solución hasta ese momento.

Cuando la solución del dual es menor que la mejor solución del primal conocida hasta ese momento, entonces se continúa la búsqueda a profundidad. Le llamamos a profundidad cuando la rama continúa hacia abajo "abriendo servicios".

Cuando la solución del dual es mayor o igual que la mejor solución del primal conocida hasta ese momento, entonces se suspende la búsqueda a profundidad y lo que se hace es "cerrar" el último servicio etiquetado como "primer abierto". Si se acaba de cerrar un servicio y se alcanzó un criterio de paro, entonces se regresa la búsqueda hacia arriba hasta que se encuentre algún servicio que no haya sido forzado a cerrarse, si se encuentra tal servicio, se continúa con la ramificación como ya se ha mencionado, hasta que se agote la posibilidad de encontrar una mejor solución a la que se tenga hasta ese momento.

La solución óptima se encontrará en aquél nodo donde la $G=Z$ y éste sea el mejor de los casos donde haya sucedido esta igualdad.

**Ejemplo 3.**

En esta sección se presenta un ejemplo en el cual fue necesario aplicar hasta la fase III: Ramificación por Selección del algoritmo propuesto. Este problema fue tomado de Donald Erlenkotter, [6]. Erlenkotter presentó dos versiones de este problema. Nosotros presentamos aquí la segunda versión de este problema.

Es un problema de (5X8), con $m$=5 servicios y $n$=8 clientes. En la tabla (13) se dan los costos totales variables, $c_{ij}$, y los costos fijos, $f_i$.

| servicio | Cliente | | | | | | | | Costo fijo |
|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | $f_i$ |
| 1 | 120 | 180 | 100 | ∞ | 60 | ∞ | 180 | ∞ | 200 |
| 2 | 210 | ∞ | 150 | 240 | 55 | 210 | 110 | 165 | 200 |
| 3 | 180 | 190 | 110 | 195 | 50 | ∞ | ∞ | 195 | 200 |
| 4 | 210 | 190 | 150 | 180 | 65 | 120 | 160 | 120 | 400 |
| 5 | 170 | 150 | 110 | 150 | 70 | 195 | 200 | ∞ | 300 |

Tabla (13) Datos de los costos variables y fijos para el ejemplo 3.

Los resultados obtenidos en la fase de Nivelación Dual, son los siguientes:

$\alpha_{jmin} = \alpha_{jsegunda} = \alpha_{jmax} = (180 \quad 190 \quad 145 \quad 240 \quad 50 \quad 210 \quad 160 \quad 315)$;

Con $G = \sum_{j=1}^{8} \alpha_j = 1490$, como solución del dual.

El vector de los costos fijos resultantes después de fase de Nivelación Dual es:

$f_i = (85 \quad 0 \quad 0 \quad 55 \quad 110)$, que indica abrir los servicios $\{2,3\}$

La solución del primal a partir de estos resultados y aplicando la fase II: Pulido del Primal, es: $Z$=1610 con $i = 2.3$, como servicios abiertos. El **primer servicio** abierto que indicó el primal fue $i$=3.

Al aplicar las dos primeras fases del algoritmo, vemos que $Z \neq G$; y por lo tanto, es necesario aplicar la fase III: Ramificación por Selección.

Se inicia la fase III en el servicio que el pulido del primal indicó como **primer abierto**, en este caso, el servicio 3. Al obligar a abrir el tercer servicio, esto es, hacer $y_3 = 1 \Rightarrow f_3 = 0$, se obtiene la siguiente solución del dual y del primal respectivamente: $G$=1610 y $Z$=1610, con $I^a = \{2,3\}$ como servicios abiertos. En este caso se llegó a una de las condiciones para que la búsqueda a profundidad de la rama se pare.

Ahora se prueba la solución, obligando a cerrar el servicio 3, esto es, $y_3 = 0 \Rightarrow f_3 = \infty$, obteniéndose las siguientes soluciones: $G$=1580 y $Z$=1580, con $I^a = \{1,2\}$, siendo el **primer abierto**, el servicio $i$=1. En este caso se mejoró la solución del dual y también la solución del primal. Nuevamente se ha llegado a una de las condiciones de paro de la búsqueda a profundidad. En este caso, aquí termina la fase de Ramificación por Selección, pues ya se agotaron las posibilidades de encontrar una mejor solución.

En los dos lados de la rama realizada, se llegó a la convergencia entre $Z$ y $G$, en estos casos se selecciona la mejor solución de ambas, la cual es la solución óptima del problema original, la solución óptima es entonces:

$Z^* = G^* = 1580$

$y_1 = y_2 = 1$

$x_{11} = x_{12} = x_{13} = x_{24} = x_{25} = x_{26} = x_{27} = x_{28} = 1$

En la figura (11) mostramos la fase III en forma de árbol.

En la figura (11) mostramos la fase III en forma de árbol.

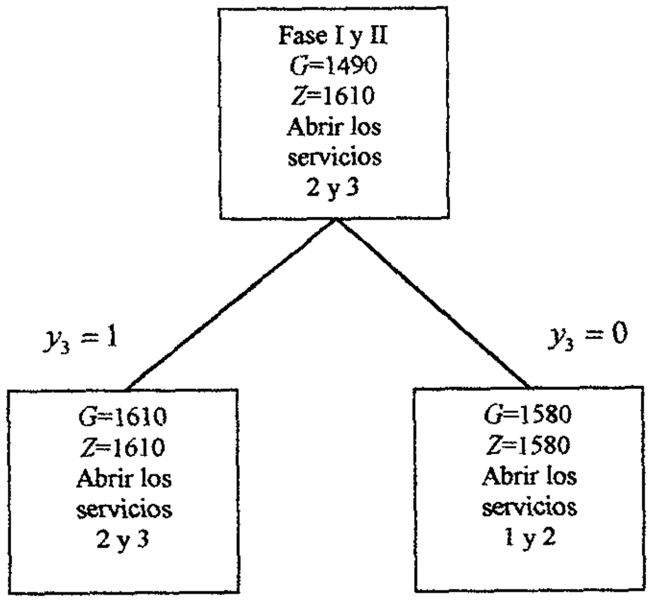



Fig. (11) Árbol que resume la fase de Ramificación por Selección

Las cooperaciones de los clientes para pagar el costo fijo del servicio que los va a atender son:

$$\lambda_{1j} = [90 \quad 60 \quad 50 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$$

$$\lambda_{2j} = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 50 \quad 150]$$

# CAPÍTULO VIII. EXPERIENCIA COMPUTACIONAL

## 8.1 EJEMPLO COMPARATIVO.

En esta sección se resolverá el ejemplo que se presentó en el capítulo VI, en el cual se vio el funcionamiento de algunos algoritmos existentes, con el fin de comparar de alguna manera el algoritmo propuesto con el número de iteraciones realizadas en los dos métodos que se presentaron en el capítulo VI. En la tabla (14) se dan los costos variables, $c_{ij}$, y los costos fijos, $f_i$, del problema de (4x4).

| Servicio | Cliente | | | | Costo fijo |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | $f_i$ |
| 1 | 0 | 12 | 20 | 18 | 7 |
| 2 | 12 | 0 | 8 | 6 | 7 |
| 3 | 20 | 8 | 0 | 6 | 7 |
| 4 | 18 | 6 | 6 | 0 | 7 |

Tabla (14) Datos para el ejemplo comparativo tomado de [9].

**Solución del Dual:**

Para encontrar primero la solución del dual, aplicamos la fase I del algoritmo propuesto.

Fase no. I. Los valores iniciales de las variables duales son:

$$\alpha_{jmin} = (0 \quad 0 \quad 0 \quad 0)$$

$$\alpha_{jsegunda} = (12 \quad 6 \quad 6 \quad 6)$$

$$\alpha_{jmax} = (\infty \quad \infty \quad \infty \quad \infty)$$

El vector de diferencias: $\Delta_j = (12 \quad 6 \quad 6 \quad 6)$

**#1.** $\Delta_{jmax} = \Delta_1 = 12 \geq f_1 = 7$

Las actualizaciones son:

$$\alpha_{jmax} = (7 \quad 12 \quad 20 \quad 18)$$

$$\alpha_{jsegunda} = (7 \quad 6 \quad 6 \quad 6)$$

$$\alpha_{jmin} = (7 \quad 0 \quad 0 \quad 0)$$

$f_1 = 7 - 7 = 0 \qquad \Rightarrow \qquad$ abrir el sevicio $i = 1$.

**#2.** $\Delta_{jmax} = \Delta_2 = \Delta_3 = \Delta_4 = 6 < f_2, f_3, f_4 = 7$

La menor cardinalidad de las $I_j^{cm}$, $\forall$ $j = 2,3,4$ es igual a 1 y en este ejemplo todas estas $j$ están empatadas en este sentido. En estos casos se escoge arbitrariamente a cualquiera de ellas.

Tomando arbitrariamente a $j$=2 a la cual le corresponde el servicio $i$=2 por estar en esa posición el costo mínimo.

Las actualizaciones son:

$\alpha_{2min} = 0 + 6 = 6$

$\alpha_{2segunda} = 8$

$f_2 = 7 - 6 = 1$

$\Delta_2 = 2$

**#3.** $\Delta_{jmax} = \Delta_3 = \Delta_4 = 6 < f_3, f_4 = 7$

$\alpha_{3min} = 0 + 6 = 6$

$\alpha_{3segunda} = 8$

$f_3 = 7 - 6 = 1$

$\Delta_3 = 2$

**#4.** $\Delta_{jmax} = \Delta_4 = 6 < f_4 = 7$

$\alpha_{4min} = 0 + 6 = 6$

$\alpha_{4segunda} = 18$

$f_4 = 7 - 6 = 1$

$\Delta_4 = 18 - 6 = 12$

#5. $\Delta_{jmax} = \Delta_4 = 12 \geq f_2, f_3, f_4 = 1$

$\alpha_{jmax} = (7 \quad 6 \quad 6 \quad 7)$

$\alpha_{jsegunda} = (7 \quad 6 \quad 6 \quad 7)$

$\alpha_{jmin} = (7 \quad 6 \quad 6 \quad 7)$

$f_2 = 1 - 1 = 0 \quad \Rightarrow$ abrir el sevicio $i = 2$.

$f_3 = 1 - 1 = 0 \quad \Rightarrow$ abrir el servicio $i = 3$.

$f_4 = 1 - 1 = 0 \quad \Rightarrow$ abrir el servicio $i = 4$.

Hasta aquí el vector de diferencias es: $\Delta_j = (0 \quad 0 \quad 0 \quad 0)$

Lo cual indica que la fase I ha terminado.

La solución del dual a la que se ha llegado es:

$\alpha_{jmi} = \alpha_{jsegunda} = \alpha_{jmax} = (7 \quad 6 \quad 6 \quad 7)$

$G = \sum_{j=1}^{4} \alpha_j = 26$

Abriendo los servicios: $I^a = \{1,2,3,4\}$

**Solución del Primal:**

Aplicando la fase II: Pulido del Primal, obtenemos la siguiente solución:

$Z$=28 con $i$=1,2 como servicios abiertos, donde $i$=1 es el "primer abierto"

Ya que $Z \neq G$, se requiere aplicar la fase III: Ramificación por Selección.

En el árbol de la fig. (12) se resumen las iteraciones que se tuvieron que hacer en la fase III para encontrar la solución óptima del problema original.

Como se ve en la figura, la solución óptima es:

$Z^* = G^* = 26$, con $i$=1,4 como servicios abiertos.

$y_1 = y_4 = 1$

$x_{11} = x_{42} = x_{43} = x_{44} = 1$

Los costos fijos de estos dos servicios abiertos, serán cubiertos de la siguiente forma:

$$\lambda_{1j} = (7 \quad 0 \quad 0 \quad 0)$$

$$\lambda_{4j} = (0 \quad 0 \quad 0 \quad 7)$$

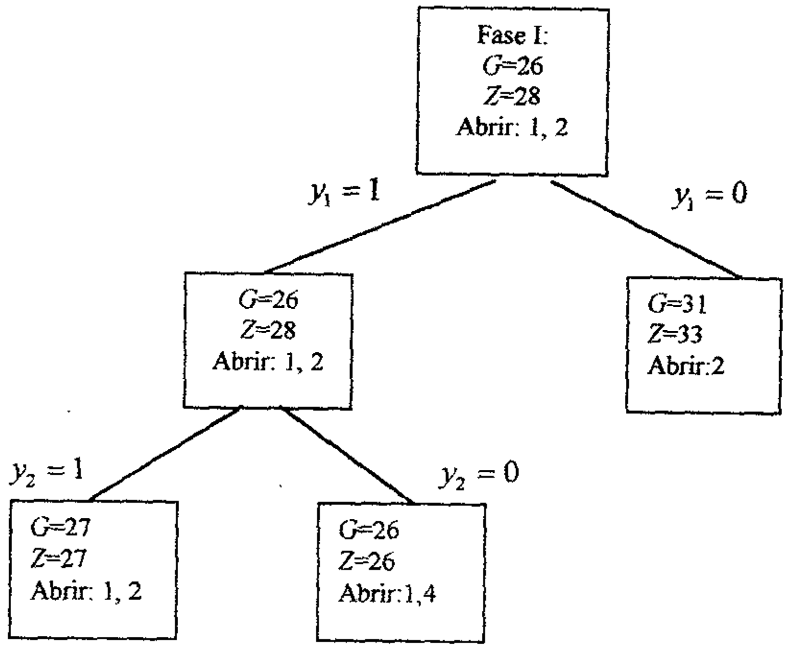



Fig. (12) Árbol que resume las iteraciones hechas en la fase III.

La solución óptima del problema original se encontró en el nodo donde se obligó a cerrar el servicio 2, es decir cuando $y_2 = 0$.

Nota: Obsérvese que al hacer $y_1 = 0$, se obtuvo una $G > Z^* = 26$, y por eso se suspendió la búsqueda a profundidad, aunque no se encontró el óptimo de esa rama.

## 8.2 TIEMPO COMPUTACIONAL.

Se realizó un programa del algoritmo propuesto en VisualBasic, se probó en una PC Pentium con velocidad de 166 MHZ. El programa es interactivo, y corre para cualquier tamaño de problema (siempre y cuando lo permita la PC), este programa inicia preguntando el tamaño del problema, después se deben introducir los costos variables y fijos, en cualquier orden, ya que así lo permite la presentación de datos que tiene el programa.

El tamaño de algunos de los problemas que se corrieron, sus características de solución y el tiempo de CPU, se muestran en la tabla (15).

| Tamaño del problema | Solución del dual | Solución del primal | Solución óptima | Se requirió fase III | No. de ramas | Tiempo de CPU (seg.) |
|---|---|---|---|---|---|---|
| *(5x8) | 1235 | 1235 | 1235 | No | 0 | 0.3233seg |
| **(4X4) | 26 | 28 | 26 | Sí | 2 | 0.3215seg |
| ***(16x16) | 176 | 248 | 190 | Sí | 4 | 10.3812seg |
| ***(20x20) | 416 | 456 | 446 | Sí | 7 | 18.1112seg |
| ***(30x30) | 2186 | 2294 | 2248 | Sí | 6 | 27.3561seg |
| ***(50x50) | 1572 | 1572 | 1572 | No | 0 | 15.2320seg |

* Se tomaron los datos de este problema de la referencia [6].

** Se tomaron los datos de este problema de la referencia [8].

*** La matriz de datos de estos problema se generaron aleatoriamente.

Tabla (15) Resumen de los resultados obtenidos en algunos problemas que se corrieron en el programa del algoritmo propuesto.

# CAPÍTULO IX. CONCLUSIONES Y TARABAJOS FUTUROS

## 9.1 CONCLUSIONES

Para resolver el Problema de Localización de Servicios Simple a partir de su planteamiento como un programa entero mixto, resulta costoso utilizar los métodos tradicionales que existen, tales como el método de Descomposición de Benders y el método de Branch and Bound, por mencionar algunos.

Se vio que la relajación del problema sobre las variables enteras y después obtener el dual de este problema relajado resultó un dual en el que ya no se tenía la presencia del cargo fijo en la función objetivo que hace que el problema primal sea "complicado" para resolver. A partir de la formulación del dual, la repartición del costo fijo fue mucho más fácil de encontrar. Otro aspecto interesante fue la forma en que a partir de la solución del dual se obtuvieron soluciones enteras del primal. Con la ayuda del Pulido del Primal, generalmente se evita la violación de las condiciones de holgura complementaria y lo que se logra con esto es que el problema ya no requiera la fase III: Ramificación por Selección. Pero, si no es suficiente esta fase de pulido, entonces la búsqueda de la solución óptima con la fase III es mucho menos exhaustiva que las ramificaciones obtenidas por el método tradicional de Branch and Bound, el cual inicia ramificando partiendo únicamente de la solución fraccionaria del problema primal relajado.

En este algoritmo que se presentó, se vio que generalmente la solución obtenida después de aplicar las dos primeras fases resultaba muy cercana a la solución óptima del problema, si no es que la óptima, por que la cota inferior y superior proporcionadas por este algoritmo son también muy cercanas.

El algoritmo presentado resultó muy eficiente para problemas grandes en los que se requería la fase III ya que la diferencia en el número de búsquedas resultó ser mucho menor que las ramificaciones si se realizaran desde el inicio de la solución del problema como se hace en Branch and Bound.

Cabe mencionar que con la idea de encontrar aquél servicio que convenía abrir en la fase II, resultaba en la mayoría de los problemas donde se probó el algoritmo, que ese servicio (llamado "primer abierto" en nuestro algoritmo), tenía que estar abierto en la solución óptima del problema original.

Con este algoritmo y otros que se han desarrollado para resolver el Problema de Localización de Servicios, se elimina la dificultad que se presenta cuando se resuelven problemas de programación entera-mixta.

## 9.2 TRABAJOS FUTUROS

Con el desarrollo del algoritmo que se presentó, cuya aplicación es la solución del Problema de Localización de Servicios Simple, surgió la idea de aplicar las estrategias utilizadas, en el Problema de Localización de Servicios con Restricciones de Capacidad.

Resulta de interés retomar algunas ideas del algoritmo presentado para aplicarlas a otros problemas con alguna semejanza en su estructura matemática por ser parte de la familia de los problemas de Cargo Fijo.

La formulación que ahora se debe considerar es la del Problema de Localización de Servicios con Restricciones de Capacidad:

$$MinZ = \sum_{i=1}^{m}\sum_{j=1}^{n} C_{ij}x_{ij} + \sum_{i=1}^{m} f_i y_1$$

$s.a:$

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

$$x_{ij} \leq y_i \; ; \qquad i = 1,2,\ldots,m \quad \text{y} \quad j = 1,2,\ldots,n$$

$$\sum_{j=1}^{n} d_j x_{ij} \leq a_i y_i \; ; \qquad i = 1,2,\ldots,m$$

$$x_{ij} \geq 0 \quad \text{y} \quad y_i = 0,1 \qquad i = 1,2,\ldots,m \quad \text{y} \quad j = 1,2,\ldots,n$$

Como vemos el planteamiento del problema ahora considera las capacidades de cada servicio $(a_i)$ y las demandas de cada cliente $(d_j)$.

Este planteamiento es el que ahora se tomaría en cuenta para tratar de resolverlo con la misma idea de funcionamiento del algoritmo propuesto.

## APÉNDICE A

### A.1 UN EJEMPLO DEL PROBLEMA DE LOCALIZACIÓN DE SERVICIOS RESUELTO CON EL PROCEDIMIETO DE PARTICIÓN DE BENDERS.

Para ver en cuantas iteraciones se llega a la solución óptima, resolvemos el siguiente ejemplo propuesto por M. Balinski [4]. Es un problema de (4x4), los costos variables y los costos fijos se dan enseguida:

| Servicio | Cliente | | | | Costo fijo |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | $f_i$ |
| 1 | 0 | 12 | 20 | 18 | 7 |
| 2 | 12 | 0 | 8 | 6 | 7 |
| 3 | 20 | 8 | 0 | 6 | 7 |
| 4 | 18 | 6 | 6 | 0 | 7 |

Tabla (A.1) Datos del ejemplo propuesto por [4].

Paso 1. Iniciamos con $\mathbf{y} = (1\ 1\ 0\ 1)$; Con las cotas iniciales: $Z^u = \infty$ , $Z^l = -\infty$

Paso 2. (#1)

$I_1 = \{1,2,4\}$; entonces asignamos: $i(1) = 1,\ i(2) = 2,\ i(3) = 4,\ i(4) = 4$

Obtenemos las $v_j$ utilizando (5):

$v_1 = c_{11} = 0;\ v_2 = c_{22} = 0;\ v_3 = c_{43} = 6;\ v_4 = c_{44} = 0 \rightarrow \mathbf{v} = (0\quad 0\quad 6\quad 0)$

Obteniendo $u_i$; $u_i = 0$ si $i \in I_1$, es decir $u_1 = u_2 = u_4 = 0$

$u_i = max(0, m_i)$ si $i \notin I_1$, de aquí que $m_3 = 6, \Rightarrow u_3 = 6$

Cálculo de $Z^u$, utilizando (6):

$$Z^u = \sum_{i \in I_1} f_i + \sum_{j=1}^{n} c_{i(j),j} - \sum_{i \in I_1} n_i u_i$$

$$Z^u = f_1 + f_2 + f_4 + c_{11} + c_{22} + c_{43} + c_{44} - n_1 u_1 - n_2 u_2 - n_4 u_4 =$$
$$= 7 + 7 + 7 + 0 + 0 + 6 + 0 - 0 - 0 - 0 = 27$$

Paso 3. (#1) El programa entero es:

$Min\ Z$

$s.a:$

$Z \geq \sum_{i=1}^{m} f_i y_i + \sum_{j=1}^{n} v_j^p - \sum_{i=1}^{m} n_i y_i u_i^p$

$y_i = 0{,}1 \qquad (i = 1{,}2{,}3{,}4)$

Esto es:

$Min\ Z$

$s.a:$

$Z \geq f_1 y_1 + f_2 y_2 + f_3 y_3 + f_4 y_4 + (v_1 + v_2 + v_3 + v_4) - [4y_1(0) + 4y_2(0) + 4y_3(6) + 4y_4(0)]$

Sustituyendo:

$Z \geq 7y_1 + 7y_2 + 7y_3 + 7y_4 + (0 + 0 + 6 + 0) - 24y_3$

Simplificando:

$Z \geq 6 + 7y_1 + 7y_2 - 17y_3 + 7y_4$

Quedando el siguiente problema:

$Min\ Z$

$s.a:$

$$Z \geq 6 + 7y_1 + 7y_2 - 17y_3 + 7y_4$$
$$y_i = 0{,}1$$

Por inspección hallamos que la solución es: $\mathbf{y} = (0\ \ 0\ \ 1\ \ 0)$, con $Z^l = -11 < 27 = Z^u$

Paso 2 (#2). De la pasada iteración tenemos ahora que $I_1 = \{3\}$; $i(1) = 3,\ i(2) = 3,\ i(3) = 3,\ i(4) = 3$

Con las respectivas $v_j$; $v_1 = c_{31} = 20,\ v_2 = c_{32} = 8,\ v_3 = c_{33} = 0,\ v_4 = c_{34} = 6$

$u_1 = 20,\ u_2 = 8,\ u_3 = 0,\ u_4 = 6$

Obteniendo la cota superior: $Z = f_3 + c_{31} + c_{32} + c_{33} + c_{34} = 41\ \therefore\ Z^u$ permanece en 27.

Paso 3. (#2) El programa entero es ahora:

Al corte obtenido en la iteración anterior, agregar el siguiente corte:

$Z \geq 7y_1 + 7y_2 + 7y_3 + 7y_4 + (20 + 8 + 0 + 6) - 4y_1(20) - 4y_2(8) - 4y_3(0) - 4y_4(6)$

Obteniendo el siguiente problema a resolver:

$$Min\ \ Z$$

$$s.a:$$

$$Z \geq 6 + 7y_1 + 7y_2 - 17y_3 + 7y_4$$
$$Z \geq 34 - 73y_1 - 25y_2 + 7y_3 - 17y_4$$
$$y_i = 0,1$$

La solución óptima encontrada por enumeración completa (ver tabla A.1.1) es la siguiente:

$\mathbf{y} = (1\ 0\ 1\ 0)$ con $Z^l = -4 < 27 = Z^u$

Paso 2. (#3)

$I_1 = \{1,3\}; \quad i(1) = 1,\ i(2) = 3,\ i(3) = 3,\ i(4) = 3$

$v_1 = c_{11} = 0,\ v_2 = c_{32} = 8,\ v_3 = c_{33} = 0,\ v_4 = c_{34} = 6$

$u_1 = 0,\ u_2 = 8,\ u_3 = 0,\ u_4 = 6$

Con $Z = f_1 + f_3 + (0 + 8 + 0 + 6) = 28 > 27 = Z^u \ \therefore\ Z^u = 27$

Paso 3. (#3)

$$Min\ \ Z$$

$$s.a:$$

$$Z \geq 6 + 7y_1 + 7y_2 - 17y_3 + 7y_4$$
$$Z \geq 34 - 73y_1 - 25y_2 + 7y_3 - 17y_4$$
$$Z \geq 14 + 7y_1 - 25y_2 + 7y_3 - 17y_4$$
$$y_i = 0,1$$

Cuya solución es: $\mathbf{y} = (0\ 1\ 1\ 1)$; con $Z^l = 3 < 27 = Z^u$

Paso 2. (#4)

$I_1 = \{2,3,4\}; \quad i(1) = 2,\ i(2) = 2,\ i(3) = 3,\ i(4) = 4$

$v_1 = c_{21} = 12,\ v_2 = c_{22} = 0,\ v_3 = c_{33} = 0,\ v_4 = c_{44} = 0$

$u_1 = 12,\ u_2 = 0,\ u_3 = 0,\ u_4 = 0$

Con $Z = f_2 + f_3 + f_4 + (12 + 0 + 0 + 0) = 33 > 27 = Z^u \ \therefore\ Z^u = 27$

Paso 3. (#4)

$Min\ \ Z$

$s.a:$

$$
\begin{aligned}
&Z \geq 6 + 7y_1 + 7y_2 - 17y_3 + 7y_4 \\
&Z \geq 34 - 73y_1 - 25y_2 + 7y_3 - 17y_4 \\
&Z \geq 14 + 7y_1 - 25y_2 + 7y_3 - 17y_4 \\
&Z \geq 12 - 41y_1 + 7y_2 + 7y_3 + 7y_4 \\
&y_i = 0,1
\end{aligned}
$$

La solución por inspección es: $\mathbf{y} = (1\ 1\ 1\ 0)$ ; con $Z^l = 3 < 27 = Z^u$

Paso2. (#5)

$I_1 = \{1,2,3\};\quad i(1) = 1,\ i(2) = 2,\ i(3) = 3,\ i(4) = 2$

$v_1 = c_{11} = 0,\ v_2 = c_{22} = 0,\ v_3 = c_{33} = 0,\ v_4 = c_{24} = 6$

$u_1 = 0,\ u_2 = 0,\ u_3 = 0,\ u_4 = 6$

Con $Z = f_1 + f_2 + f_3 + (0+0+0+6) = 27 = 27 = Z^u \ \therefore\ Z^u = 27$

Paso 3. (#5)

$Min\ \ Z$

$s.a:$

$$
\begin{aligned}
&Z \geq 6 + 7y_1 + 7y_2 - 17y_3 + 7y_4 \\
&Z \geq 34 - 73y_1 - 25y_2 + 7y_3 - 17y_4 \\
&Z \geq 14 + 7y_1 - 25y_2 + 7y_3 - 17y_4 \\
&Z \geq 12 - 41y_1 + 7y_2 + 7y_3 + 7y_4 \\
&Z \geq 6 + 7y_1 + 7y_2 + 7y_3 - 17y_4 \\
&y_i = 0,1
\end{aligned}
$$

Cuya solución es: $\mathbf{y} = (1\ 1\ 1\ 1)$ ; con $Z^l = 10 < 27 = Z^u$

| y | | | | Programa entero en la iteración #: | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | #1 | #2 | #3 | #4 | #5 | #6 | #7 | #8 | #9 | #10 | #11 | #12 |
| 0 | 0 | 0 | 1 | 13 | 17 | -3 | 19 | -11 | 7 | 37 | 9 | 19 | -3 | -15 | 13 |
| 0 | 0 | 1 | 0 | -11 | 41 | 21 | 19 | 13 | 7 | 13 | 1 | -5 | -11 | -23 | 13 |
| 0 | 0 | 1 | 1 | -4 | 24 | 4 | 26 | -4 | 14 | 20 | -16 | 2 | -28 | -88 | 20 |
| 0 | 1 | 0 | 0 | 13 | 9 | -11 | 19 | 13 | 7 | 13 | 33 | -5 | 21 | 9 | -11 |
| 0 | 1 | 0 | 1 | 20 | -8 | -28 | 26 | -4 | 14 | 20 | 16 | 2 | 4 | -56 | -4 |
| 0 | 1 | 1 | 0 | -4 | 16 | -4 | 26 | 20 | 14 | -4 | 8 | -22 | -4 | -64 | -4 |
| 0 | 1 | 1 | 1 | 3 | -1 | -21 | 33 | 3 | 21 | 3 | -9 | -15 | -21 | -129 | 3 |
| 1 | 0 | 0 | 0 | 13 | -39 | 21 | -29 | 13 | 7 | -35 | -15 | 19 | 21 | 57 | 13 |
| 1 | 0 | 0 | 1 | 20 | -56 | 4 | -22 | -4 | 14 | -28 | -32 | 26 | 4 | -8 | 20 |
| 1 | 0 | 1 | 0 | -4 | -32 | 28 | -22 | 20 | 14 | -52 | -40 | 2 | -4 | -16 | 20 |
| 1 | 0 | 1 | 1 | 3 | -49 | 11 | -15 | 3 | -21 | -45 | -57 | 9 | -21 | -81 | 27 |
| 1 | 1 | 0 | 0 | 20 | -64 | -4 | -22 | 20 | 14 | -52 | -8 | 2 | 28 | 16 | -4 |
| 1 | 1 | 0 | 1 | 27 | -81 | -21 | -15 | 3 | 21 | -45 | -25 | 9 | 11 | -49 | 3 |
| 1 | 1 | 1 | 0 | 3 | -57 | 3 | -15 | 27 | 21 | -69 | -33 | -15 | 3 | -57 | 3 |
| 1 | 1 | 1 | 1 | 10 | -74 | -14 | -8 | 10 | 28 | -62 | -50 | -8 | -14 | -122 | 10 |

Tabla A.1.1 Valores de $Z$ obtenidos en las diferentes iteraciones.

Paso 2. (#6)

$I_1 = \{1,2,3,4\};\quad i(1) = 1,\ i(2) = 2,\ i(3) = 3,\ i(4) = 4$

$v_1 = c_{11} = 0,\ v_2 = c_{22} = 0,\ v_3 = c_{33} = 0,\ v_4 = c_{44} = 0$

$u_1 = 0,\ u_2 = 0,\ u_3 = 0,\ u_4 = 0$

Con $Z = f_1 + f_2 + f_3 + f_4 + (0+0+0+0) = 28 > 27 = Z^u \quad \therefore \quad Z^u = 27$

Paso 3. (#6)

$$
\begin{aligned}
&Min \quad Z\\
&s.a:\\
&\qquad Z \geq 6 + 7y_1 + 7y_2 - 17y_3 + 7y_4\\
&\qquad Z \geq 34 - 73y_1 - 25y_2 + 7y_3 - 17y_4\\
&\qquad Z \geq 14 + 7y_1 - 25y_2 + 7y_3 - 17y_4\\
&\qquad Z \geq 12 - 41y_1 + 7y_2 + 7y_3 + 7y_4\\
&\qquad Z \geq 6 + 7y_1 + 7y_2 + 7y_3 - 17y_4\\
&\qquad Z \geq 7y_1 + 7y_2 + 7y_3 + 7y_4\\
&\qquad y_i = 0{,}1
\end{aligned}
$$

Cuya solución es: $\mathbf{y} = (0 \ \ 0 \ \ 0 \ \ 1)$ ; con $Z^l = 19 < 27 = Z^u$

Paso 2. (#7)

$I_1 = \{4\}$; $i(1) = 4,\ i(2) = 4,\ i(3) = 4,\ i(4) = 4$

$v_1 = c_{41} = 18,\ v_2 = c_{42} = 6,\ v_3 = c_{43} = 6,\ v_4 = c_{44} = 0$

$u_1 = 18,\ u_2 = 6,\ u_3 = 6,\ u_4 = 0$

Con $Z = f_4 + (18 + 6 + 6 + 0) = 37 > 27 = Z^u \ \therefore \ Z^u = 27$

Paso 3. (#7)

$$
\begin{aligned}
&Min \quad Z\\
&s.a:\\
&\qquad Z \geq 6 + 7y_1 + 7y_2 - 17y_3 + 7y_4\\
&\qquad Z \geq 34 - 73y_1 - 25y_2 + 7y_3 - 17y_4\\
&\qquad Z \geq 14 + 7y_1 - 25y_2 + 7y_3 - 17y_4\\
&\qquad Z \geq 12 - 41y_1 + 7y_2 + 7y_3 + 7y_4\\
&\qquad Z \geq 6 + 7y_1 + 7y_2 + 7y_3 - 17y_4\\
&\qquad Z \geq 7y_1 + 7y_2 + 7y_3 + 7y_4\\
&\qquad Z \geq 30 - 65y_1 - 17y_2 - 17y_3 + 7y_4\\
&\qquad y_i = 0{,}1
\end{aligned}
$$

Cuya solución es: $\mathbf{y} = (0 \ \ 1 \ \ 0 \ \ 0)$ ; con $Z^l = 19 < 27 = Z^u$

Paso 2. (#8)

$I_1 = \{2\};\quad i(1) = 2,\ i(2) = 2,\ i(3) = 2,\ i(4) = 2$

$v_1 = c_{21} = 12,\ v_2 = c_{22} = 0,\ v_3 = c_{23} = 8,\ v_4 = c_{24} = 6$

$u_1 = 12,\ u_2 = 0,\ u_3 = 8,\ u_4 = 6$

Con $Z = f_2 + (12+0+8+6) = 33 > 27 = Z^u \quad \therefore \quad Z^u = 27$

Paso 3. (#8)

$$
\begin{aligned}
&Min \quad Z \\
&s.a: \\
&\qquad Z \geq 6 + 7y_1 + 7y_2 - 17y_3 + 7y_4 \\
&\qquad Z \geq 34 - 73y_1 - 25y_2 + 7y_3 - 17y_4 \\
&\qquad Z \geq 14 + 7y_1 - 25y_2 + 7y_3 - 17y_4 \\
&\qquad Z \geq 12 - 41y_1 + 7y_2 + 7y_3 + 7y_4 \\
&\qquad Z \geq 6 + 7y_1 + 7y_2 + 7y_3 - 17y_4 \\
&\qquad Z \geq 7y_1 + 7y_2 + 7y_3 + 7y_4 \\
&\qquad Z \geq 30 - 65y_1 - 17y_2 - 17y_3 + 7y_4 \\
&\qquad Z \geq 26 - 41y_1 + 7y_2 - 25y_3 - 17y_4 \\
&\qquad y_i = 0,1
\end{aligned}
$$

Cuya solución es: $\mathbf{y} = (1\ 0\ 0\ 1)$ ; con $Z^l = 20 < 27 = Z^u$

Paso 2. (# 9)

$I_1 = \{1,4\};\quad i(1) = 1,\ i(2) = 4,\ i(3) = 4,\ i(4) = 4$

$v_1 = c_{11} = 0,\ v_2 = c_{42} = 6,\ v_3 = c_{43} = 6,\ v_4 = c_{44} = 0$

$u_1 = 0,\ u_2 = 6,\ u_3 = 6,\ u_4 = 0$

Con $Z = f_1 + f_4 + (0+6+6+0) = 26 < 27 = Z^u \quad \therefore \quad Z^u = 26$

Paso 3. (#9)

$Min \;\; Z$

$s.a:$

$$
\begin{aligned}
&Z \geq 6+7y_1+7y_2-17y_3+7y_4\\
&Z \geq 34-73y_1-25y_2+7y_3-17y_4\\
&Z \geq 14+7y_1-25y_2+7y_3-17y_4\\
&Z \geq 12-41y_1+7y_2+7y_3+7y_4\\
&Z \geq 6+7y_1+7y_2+7y_3-17y_4\\
&Z \geq 7y_1+7y_2+7y_3+7y_4\\
&Z \geq 30-65y_1-17y_2-17y_3+7y_4\\
&Z \geq 26-41y_1+7y_2-25y_3-17y_4\\
&Z \geq 12+7y_1-17y_2-17y_3+7y_4\\
&y_i = 0{,}1
\end{aligned}
$$

Cuya solución es: $\mathbf{y} = (1\ 1\ 0\ 0)$ ; con $Z^l = 20 < 26 = Z^u$

Paso 2. (#10)

$I_1 = \{1,2\}$; $\;i(1) = 1,\ i(2) = 2,\ i(3) = 2,\ i(4) = 2$

$v_1 = c_{11} = 0,\ v_2 = c_{22} = 0,\ v_3 = c_{23} = 8,\ v_4 = c_{24} = 6$

$u_1 = 0,\ u_2 = 0,\ u_3 = 8,\ u_4 = 6$

Con $Z = f_1 + f_2 + (0+0+8+6) = 28 > 26 = Z^u \;\therefore\; Z^u = 26$

Paso 3. (#10)

$$
\begin{aligned}
&Min\ \ Z\\
&s.a:\\
&\quad Z \geq 6 + 7y_1 + 7y_2 - 17y_3 + 7y_4\\
&\quad Z \geq 34 - 73y_1 - 25y_2 + 7y_3 - 17y_4\\
&\quad Z \geq 14 + 7y_1 - 25y_2 + 7y_3 - 17y_4\\
&\quad Z \geq 12 - 41y_1 + 7y_2 + 7y_3 + 7y_4\\
&\quad Z \geq 6 + 7y_1 + 7y_2 + 7y_3 - 17y_4\\
&\quad Z \geq 7y_1 + 7y_2 + 7y_3 + 7y_4\\
&\quad Z \geq 30 - 65y_1 - 17y_2 - 17y_3 + 7y_4\\
&\quad Z \geq 26 - 41y_1 + 7y_2 - 25y_3 - 17y_4\\
&\quad Z \geq 12 + 7y_1 - 17y_2 - 17y_3 + 7y_4\\
&\quad Z \geq 14 + 7y_1 + 7y_2 - 25y_3 - 17y_4\\
&\quad y_i = 0{,}1
\end{aligned}
$$

Cuya solución es: $\mathbf{y} = (1\ \ 0\ \ 0\ \ 0)$ ; con $Z^l = 21 < 26 = Z^u$

Paso 2. (#11)

$I_1 = \{1\};\ \ i(1) = 1,\ i(2) = 1,\ i(3) = 1,\ i(4) = 1$

$v_1 = c_{11} = 0,\ v_2 = c_{12} = 12,\ v_3 = c_{13} = 20,\ v_4 = c_{14} = 18$

$u_1 = 0,\ u_2 = 12,\ u_3 = 20,\ u_4 = 18$

Con $Z = f_1 + (0 + 12 + 20 + 18) = 57 > 26 = Z^u\ \ \therefore\ \ Z^u = 26$

Paso 3. (#11)

$Min \;\; Z$

$s.a:$

$$
\begin{aligned}
&Z \geq 6 + 7y_1 + 7y_2 - 17y_3 + 7y_4 \\
&Z \geq 34 - 73y_1 - 25y_2 + 7y_3 - 17y_4 \\
&Z \geq 14 + 7y_1 - 25y_2 + 7y_3 - 17y_4 \\
&Z \geq 12 - 41y_1 + 7y_2 + 7y_3 + 7y_4 \\
&Z \geq 6 + 7y_1 + 7y_2 + 7y_3 - 17y_4 \\
&Z \geq 7y_1 + 7y_2 + 7y_3 + 7y_4 \\
&Z \geq 30 - 65y_1 - 17y_2 - 17y_3 + 7y_4 \\
&Z \geq 26 - 41y_1 + 7y_2 - 25y_3 - 17y_4 \\
&Z \geq 12 + 7y_1 - 17y_2 - 17y_3 + 7y_4 \\
&Z \geq 14 + 7y_1 + 7y_2 - 25y_3 - 17y_4 \\
&Z \geq 50 + 7y_1 - 41y_2 - 73y_3 - 65y_4 \\
&y_i = 0,1
\end{aligned}
$$

Cuya solución es: $\mathbf{y} = \begin{pmatrix} 1 & 0 & 1 & 1 \end{pmatrix}$; con $Z^l = 21 < 26 = Z^u$

Paso 2. (#12)

$I_1 = \{1,3,4\}; \;\; i(1) = 1, \; i(2) = 4, \; i(3) = 3, \; i(4) = 4$

$v_1 = c_{11} = 0, \; v_2 = c_{42} = 6, \; v_3 = c_{33} = 0, \; v_4 = c_{44} = 0$

$u_1 = 0, \; u_2 = 6, \; u_3 = 0, \; u_4 = 0$

Con $Z = f_1 + f_3 + f_4 + (0 + 6 + 0 + 0) = 27 > 26 = Z^u \;\; \therefore \;\; Z^u = 26$

Paso 3. (#12)

$Min \quad Z$

$s.a:$

$$
\begin{aligned}
&Z \geq 6+7y_1+7y_2-17y_3+7y_4\\
&Z \geq 34-73y_1-25y_2+7y_3-17y_4\\
&Z \geq 14+7y_1-25y_2+7y_3-17y_4\\
&Z \geq 12-41y_1+7y_2+7y_3+7y_4\\
&Z \geq 6+7y_1+7y_2+7y_3-17y_4\\
&Z \geq 7y_1+7y_2+7y_3+7y_4\\
&Z \geq 30-65y_1-17y_2-17y_3+7y_4\\
&Z \geq 26-41y_1+7y_2-25y_3-17y_4\\
&Z \geq 12+7y_1-17y_2-17y_3+7y_4\\
&Z \geq 14+7y_1+7y_2-25y_3-17y_4\\
&Z \geq 50+7y_1-41y_2-73y_3-65y_4\\
&Z \geq 6+7y_1-17y_2+7y_3+7y_4\\
&y_i = 0,1
\end{aligned}
$$

La solución de este programa entero con 12 restricciones es: $\mathbf{y} = (1 \ 0 \ 0 \ 1)$ ; con $Z^l = 26 = 26 = Z^u$

Como vemos, se ha llegado a $Z^l = Z^u$, lo cual nos indica que se ha encontrado la solución óptima.

# BIBLIOGRAFÍA


1. Aceves García Ricardo, tesis doctoral "Un Algoritmo para Resolver el Problema de Localización de Servicios con Restricciones de Demanda y Adicionales". DEPFI, UNAM. 1996.
2. B. M. Khumawala; "An Efficient Branch and Bound Algorithm for the Warehouse Location Problem". Management Science 18. 1072
3. Balinski, M., and P. Wolfe; "On Benders Decomposition and Plant Location Problem". Mathematica Working Paper ARO-27. 1963.
4. Balinski, M.; "Integer Programming: Methods, Uses, Computation". Management Science, 12. 1965.
5. Cornuejols, Fisher, and Nemhauser; "Location of Bank Accounts to Optimize Float: An Analytic Study of Exact and Aproximate Algorithms". Management Science, 23. 1997.
6. Donald Erlenkotter; "A Dual-Based Procedure for Uncapacitated Facility Location". Operation Research vol.26, no. 6. 1978.
7. Efroymson, M., and T. Ray; "A Branch and Bound Algorithm for Plant Location". Operation Research no. 14(3). 1966.
8. Harley M. Salkin and Kamlesh Mathur; "Foundations of Integer Programming". North-Holland. 1989.
9. Pitu B. Mirchandani and Richard L. Francis; "Discrete Location Theory".A Wiley-Interscience Publication; Jhon Wiley and Sons, Inc. 1990.
10. Taha; "Integer Programming", pag. 302.
11. Wayne L. Winston; "Investigación de Operaciones". Grupo Editorial Iberoamérica. 1994.