CAPÍTULO IV

SOLUCIÓN DEL PROBLEMA

Una vez descritos los diferentes conceptos y técnicas utilizados para resolver este tipo de problemas de manera exacta. En este capítulo se presentará el trabajo hecho para obtener la solución óptima al Problema de Localización de Máxima Cobertura Capacitado.

4.1 Relajación Lagrangeana

Como se describió en el Capítulo 2, un valor para la cota superior se puede obtener por medio de una relajación lineal o una relajación Lagrangeana. Como se mencionó anteriormente, se sabe que las cotas obtenidas con el dual Lagrangeano son al menos tan buenas como las de la relajación lineal. Por esta razón, para el presente proyecto de investigación se decidió probar la efectividad de trabajar con la relajación Lagrangeana para obtener cotas superiores de mejor calidad. El que las cotas sean buenas permitirá que se reduzca el esfuerzo computacional en un esquema de ramificación, además de que se espera arroje un conjunto de instalaciones inicial de buena calidad, utilizando una heurística primal.

Para considerar un enfoque de relajación Lagrangeana, se debe detectar el conjunto de restricciones que al ser omitidas del problema original, hagan de éste un sub-problema para el cual se conozca algún método efectivo de solución. El conjunto (8) de restricciones de asignación (que se puede observar en el modelo del CMCLP en el Capítulo 1) evitan que un cliente sea asignado a más de una instalación. Sí se omiten, en la solución del problema

relajado se permitirá que, por ejemplo, algún cliente esté asignado a varias instalaciones. Se considera una relajación Lagrangeana en la cual, las restricciones de asignación son incorporadas a la función objetivo y su violación es penalizada (60). Para $\lambda \in \Re_{+}^{|J|}$, sea:

$$Max \sum_{i \in I} \sum_{j \in J} h_j x_{ij} + \sum_{j \in J} \lambda_j \left(1 - \sum_{i \in I_\delta(j)} x_{ij} \right)$$

Lo cual se puede simplificar obteniendo la siguiente relajación:

(LR_λ)

$$Max \sum_{i \in I} \sum_{j \in J_\delta(i)} \left(h_j - \lambda_j \right) x_{ij} + \sum_{j \in J} \lambda_j \tag{60}$$

Sujeto a

$$\sum_{i \in I} y_i \leq p \tag{61}$$

$$\sum_{j \in J_\delta(i)} h_j x_{ij} \leq b_i y_i \qquad \forall i \in I \tag{62}$$

$$y_i = \{0,1\} \qquad \forall i \in I \tag{63}$$

$$x_{ij} = \{0,1\} \qquad \forall i \in I, j \in J \tag{64}$$

La restricción (61) indica que a lo más se pueden abrir p instalaciones. El conjunto de restricciones (62) se refiere a que sólo puede ser asignado un cliente $j \in J$ a una instalación en el nodo $i \in I$ si ésta ha sido abierta, además de que la demanda total asignada a una instalación en el nodo $i \in I$ no puede sobrepasar la capacidad de la misma. Los conjuntos (63) y (64) son restricciones sobre la naturaleza binaria de las variables de decisión.

Para resolver (LR_λ), se observa que se buscan abrir como mucho p instalaciones, con las cuales se pueda dar la mayor cobertura de servicio posible, dado un radio de cobertura δ. Además, se tiene que si una instalación $i \in I$ no es abierta, ningún cliente se podrá asignar a ella y su contribución a la función objetivo sería cero. Por lo que la formulación descrita

anteriormente, (LR_λ), se puede descomponer en n sub-problemas de la mochila que pueden resolverse de manera independiente, para $i=1, 2, ..., |I|$.

Se define el i-ésimo problema de la mochila de la siguiente manera:

$KP_i(\lambda)$

$$Max \sum_{i \in I} \sum_{j \in J_\delta(i)} (h_j - \lambda_j) x_{ij} \tag{65}$$

Sujeto a

$$\sum_{j \in J_\delta(i)} h_j x_{ij} \leq b_i y_i \qquad \forall i \in I \tag{66}$$

$$y_i = \{0,1\} \qquad \forall i \in I \tag{67}$$

$$x_{ij} = \{0,1\} \qquad \forall i \in I, j \in J \tag{68}$$

Con la función objetivo (65) que maximiza la demanda cubierta por las instalaciones seleccionadas y penaliza la violación a las restricciones de asignación; y el conjunto de restricciones (66) – (68) que funcionan de la misma forma descrita en el modelo anterior.

Ahora, para cada instalación $i \in I$ se tiene lo siguiente, y_i puede ser 0 o 1. Si $y_i = 0$, entonces $KP_i(\lambda) = 0$ y $x_{ij} = 0 \ \ \forall j \in J$. Por otro lado, si $y_i = 1$, entonces se considera que la instalación $i \in I$ es seleccionada y que los clientes asignados a dicha instalación están dados por $\{i : x_{ij} = 1 \text{ en la solución ótima de } KP_i(\lambda)\}$. Dado que como mucho se pueden abrir p instalaciones (restricción (61)) entonces el procedimiento para la obtención de la solución óptima de (LR_λ) es el siguiente. Se ordenan los índices del conjunto I de forma descendente

con respecto a los valores $KP_i(\lambda)$, de tal manera que $KP_{i_1}(\lambda) \geq KP_{i_2}(\lambda) \geq \ldots \geq KP_{i_{|I|}}(\lambda)$. Definimos a continuación el conjunto de las instalaciones que se abrirán, tomando en cuenta sólo aquellas para las cuales $KP_i(\lambda) > 0$; es decir, sea $p^* := \min\{p, \max\{r : KP_{i_r} > 0\}\}$. Una vez encontrado el conjunto p^* se calcula $(LR_\lambda) := \sum_{l=1}^{p^*} KP_{i_l}(\lambda) + \sum_{j \in J} \lambda_j$. Como se mencionó antes, puede haber varios clientes asignados a una o más instalaciones y otros tantos que no pudieron ser asignados. La mejor cota superior se obtiene resolviendo el dual Lagrangeano:

$$Z^*_{LD} := \min_{\lambda}((LR_\lambda))$$

El dual Lagrangeano se resuelve utilizando el algoritmo de optimización subgradiente, en cada iteración del mismo se resuelve el siguiente algoritmo:

Algoritmo para obtener la solución óptima de (LR_λ)

Sea

$\lambda \in \Re_+^{|J|}$, el vector de multiplicadores Lagrangeanos,

e_j, los coeficientes de la función objetivo para el sub-problema de la mochila,

$\overline{Z}^*$, la mejor cota superior conocida.

$\overline{Z}^* := \infty$

Paso 0. $e_j := h_j - \lambda_j \ \forall j \in \{J_\delta(i)\}$. Resolver $KP_i(\lambda)$ $i \in I$ con e_j y los coeficientes de las restricciones de capacidad h_j.

Paso 1. Ordenar los índices del conjunto I de forma descendente con respecto a los valores $KP_i(\lambda)$, de tal manera que $KP_{i_1}(\lambda) \geq KP_{i_2}(\lambda) \geq \ldots \geq KP_{i_{|I|}}(\lambda)$.

Paso 2. Hacer $p^* := \min\{p, \max\{r : KP_{i_r} > 0\}\}$.

Paso 3. Obtener $(LR_\lambda) := \sum_{l=1}^{p^*} KP_{i_l}(\lambda) + \sum_{j \in J} \lambda_j$.

Paso 4. Si $(LR_\lambda) < \overline{Z}^*$, hacer $\overline{Z}^* := (LR_\lambda)$.

Paso 5. Terminar.

Con lo anterior se obtiene una cota superior válida para el valor de la solución óptima del problema original y un conjunto de instalaciones seleccionadas con su respectiva matriz de asignación que sirve como solución inicial para poder encontrar una solución factible para el CMCLP con una heurística primal.

4.2 Optimización Subgradiente

La resolución del dual Lagrangeano, que nos dará la mejor cota superior e inferior, se obtiene por medio de la optimización subgradiente que se describe a continuación:

Sea,

$\lambda \in \Re_+^{|J|}$, el vector de multiplicadores Lagrangeanos,

k, el número de iteraciones,

ρ_k, un escalar entre 0 y 2 que se modifica cada vez que se alcanza un cierto número de iteraciones sin mejorar el valor de la cota superior.

Paso 0. Inicializar k, $\lambda \geq 0$ y ρ_k.

Paso 1. Resolver (LR_λ).

Paso 2. Calcular el subgradiente $S_k := 1 - \sum_{i \in I_\delta(j)} x_{ij}, \forall j \in J$.

Paso 3. Hacer $\lambda^{k+1} := \lambda^k - \theta_k S_k, \theta_k \geq 0$.

Paso 4. $k := k+1$.

Paso 5. Regresar al paso 1 si no se satisfacen los criterios de parada.

Nemhauser y Wolsey (1999) proponen dos esquemas para seleccionar θ_k:

1) Una serie divergente: $\sum_{k=1}^{\infty} \theta_k \to \infty, \theta_k \to 0$ cuando $k \to \infty$.

2) Una serie geométrica: $\theta_k := \theta_0 \rho_k$ o $\theta_k := \dfrac{\rho_k ((LR_{\lambda^k}) - \underline{Z})}{\|S_k\|^2}$ donde $\underline{Z}$ es una cota inferior o el valor óptimo del problema.

Se sabe que la serie 1 es satisfactoria teóricamente, dado que converge a un punto óptimo. Pero en la práctica la convergencia es demasiado lenta. La serie 2, que es recomendada en la práctica, es menos satisfactoria desde el punto de vista teórico. La convergencia es "geométrica", pero el punto límite solamente es óptimo si los valores iniciales de (θ_0, ρ) o ($\underline{Z}$, ρ) son suficientemente grandes. En la práctica, valores apropiados pueden ser encontrados después de pequeñas pruebas (Nemhauser y Wolsey, 1999).

George Nemhauser y Laurence Wolsey (1999) aseguran que en el mejor escenario, el algoritmo subgradiente se puede detener cuando, en alguna iteración k, se obtiene $S_k = 0 \in \partial LR_{\lambda^k}$. Pero, esto es muy raro en la práctica, dado que el algoritmo sólo elige un

subgradiente S_k y no tiene manera de mostrar $0 \in \partial LR_{\lambda^k}$ como una combinación convexa de subgradientes. Así que, el criterio de parada más típico es: después de un número de iteraciones fijo o cuando el valor de la solución no mejora, al menos en alguna proporción, después de cierto número de iteraciones dado.

4.3 Heurística Primal

Como se explica en la sección 4.1, la solución que se obtiene del problema relajado (LR_λ) puede ser infactible con respecto al problema original, dado que podría contener clientes asignados más de una vez, lo cual violaría las restricciones de asignación (8) del problema original CMCLP. El objetivo de utilizar una heurística primal en cada iteración de la optimización subgradiente es encontrar soluciones factibles para el problema original, con el fin de obtener una cota inferior para el valor óptimo del mismo. La idea general de la heurística es tratar de asignar la mayor cantidad de demanda, asociada a los clientes, a las instalaciones abiertas p^*. En caso de que la solución de (LR_λ) sí sea infactible, respetando el radio de cobertura δ, se intentará lograr una buena asignación para el conjunto de instalaciones p^* en cada iteración de la optimización subgradiente, utilizando como solución inicial la solución de (LR_λ) correspondiente. La heurística funciona de la siguiente manera:

Es un proceso iterativo. Para cada iteración, k = 1, 2, …, $|J|$ se busca asignar el cliente $j \in J$ a la una instalación $i \in p^*$ como se explica a continuación. Sea $\hat{a}_i$ la capacidad residual de cada instalación $i \in p^*$. Para aprovechar al máximo la capacidad de cada instalación $i \in p^*$, una buena opción es asignar primero a los clientes con mayor demanda

dado que en las iteraciones finales del proceso, cuando la capacidad restante $\hat{a}_i$ de cada instalación $i \in p^*$ sea menor, será más fácil asignar cantidades pequeñas de demanda, asociadas a clientes con menor demanda. Entonces, se ordenan las demandas $h_j \ \forall j \in J$ de forma descendente, es decir, $h_{j_1} \geq h_{j_2} \geq ... \geq h_{j_{|J|}}$ y los clientes se procesan con respecto a la ordenación anterior. No todos los clientes se pueden asignar a cualquier instalación, debido al radio de cobertura δ especificado; así que, para cada cliente es necesario generar el conjunto de instalaciones a los cuales puede ser asignado. Sea $I(j_k) := \left\{ i \in I^{p^*} : d_{ij_k} \leq \delta, \hat{a}_i \geq h_{j_k} \right\}$ el conjunto de instalaciones admisibles para el cliente j_k.

Si $I(j_k) \notin \varnothing$, el cliente $j_k \in J$ será asignado a la instalación $i \in I(j_k)$ que tenga mayor capacidad restante; dicho de otro modo, se hará $x_{i^* j_k} := 1$ con $i^* \in \arg\max_{i \in I(j_k)} \left\{ \hat{a}_i : i \in I(j_k) \right\}$ y $\sum_{i \neq i^*} x_{ij_k} := 0$. Los empates se romperán utilizando la información de (LR_λ). Finalmente, se define J_a como el conjunto de clientes asignados, y se hace $J_a \leftarrow J_a \cup \{j_k\}$. El proceso se repite para cada cliente, cubriendo la mayor cantidad de demanda posible. Así, el valor de la solución (una cota inferior) para $\underline{Z} := \sum_{j \in J_a} h_j$.

Para mayor claridad, se resume la heurística primal en el siguiente algoritmo.

Heurística Primal

Sea

k, la iteración actual,

$\underline{Z}^*$, la mejor cota inferior conocida,

a_i, la capacidad utilizada $\forall i \in p^*$.

$\underline{Z}^* := 0$

Paso 0. $k := 1,\ a_i := 0\ , \forall i \in p^*$.

Paso 1. Ordenar los índices del conjunto J de forma descendente con respecto a los valores de demanda h_j, de tal manera que $h_{j_1} \geq h_{j_2} \geq ... \geq h_{j_{|J|}}\ \ \forall j \in J$.

Paso 2. Calcular $\hat{a}_i := b_i - a_i\ \ \forall i \in p^*$ y $I(j_k) := \left\{ i \in I^{p^*} : d_{ij_k} \leq \delta, \hat{a}_i \geq h_{j_k} \right\}$.

Paso 3. Si $I(j_k) \notin \varnothing$, $i^* \in \arg\max_{i \in I(j_k)} \left\{ \hat{a}_i : i \in I(j_k) \right\}$, hacer $x_{i^* j_k} := 1$, $\sum_{i \neq i^*} x_{ij_k} := 0$,

$a_{i^*} := a_{i^*} + h_{j_k}$ y $J_a \leftarrow J_a \cup \{j_k\}$. Si no, ir a paso 4.

Paso 4. Hacer $k := k + 1$. Si $k \leq |J|$ ir a paso 2. Si no, ir a paso 5.

Paso 5. $\underline{Z} := \sum_{j \in J_a} h_j$.

Paso 6. Si $\underline{Z} > \underline{Z}^*$, hacer $\underline{Z}^* := \underline{Z}$.

Paso 6. Terminar,

En cada iteración de la optimización subgradiente, se obtiene una cota inferior con la heurística primal que a su vez es una solución factible para el problema original CMCLP. Al final de la optimización subgradiente, se tendrá la mejor cota inferior que se haya logrado obtener.

4.4 Esquema de Enumeración

A continuación se describe el algoritmo de ramificación y acotamiento propuesto para el Problema de Localización de Máxima Cobertura Capacitado. Como se mencionó en el Capítulo 2, se usa la estrategia de búsqueda a profundidad que comienza el proceso de búsqueda en el nodo raíz. Para la ramificación se intentó aprovechar la estructura del problema utilizando dos niveles de ramificación. Primero se ramificó en las variables de selección de las instalaciones y después en la asignación de los clientes. Para mayor claridad, la Figura 4.1 muestra de modo gráfico en qué consiste la ramificación.

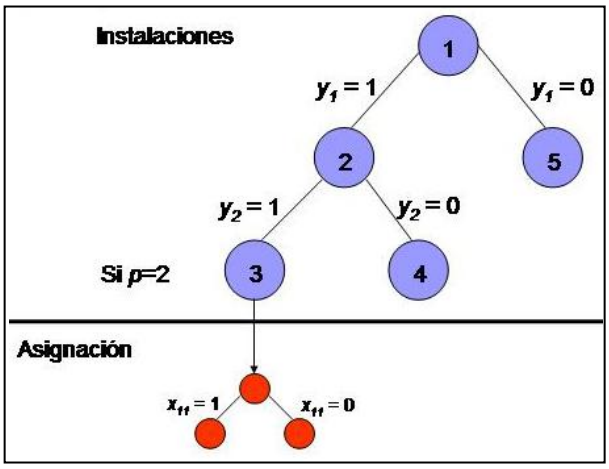

Figura 4.1. Esquema de ramificación.

En la Figura 4.1 se puede apreciar un esquema del árbol de enumeración. Inicialmente, en el nodo raíz (nodo 1) se utiliza, el algoritmo de optimización subgradiente para obtener una cota superior e inferior iniciales. En el primer nivel de ramificación (Instalaciones) los círculos azules ejemplifican los nodos correspondientes a la selección de las instalaciones abiertas. Las variables y_i se fijan a valores de 1 o 0, lo que implica que una instalación $i \in I$ debe ser considerada respectivamente como abierta o cerrada de manera

obligatoria. Del conjunto de instalaciones que dio la mejor cota inferior en el algoritmo de optimización subgradiente, se toma la instalación sin y_i fijo que tenga el mayor valor $KP_i(\lambda)$ y en ella se continua la ramificación. En cada nodo del árbol de exploración correspondiente a la selección de plantas abiertas, si $\sum_{i \in I} y_i < p$, entonces para ese nodo se aplica el algoritmo de optimización subgradiente al problema correspondiente para actualizar el valor de las cotas y se continua ramificando con el criterio mencionado anteriormente. Si $\sum_{i \in I} y_i = p$, se ejecuta un segundo nivel de ramificación (Asignación). En este nivel, donde los círculos rojos ejemplifican los nodos correspondientes a la decisión de asignación de clientes, se resuelve de manera exacta un problema de asignación con $I^* = \{i \in I^p : y_i = 1\}$, como el conjunto de plantas abiertas.

(*PA*)

$$\max \sum_{i \in I^*} \sum_{j \in J} h_j x_{ij} \tag{69}$$

Sujeto a

$$\sum_{j \in J} h_j x_{ij} \leq b_i \qquad \forall i \in I^* \tag{70}$$

$$x_{ij} \in \{0,1\} \qquad \forall i \in I^*, \forall j \in J \tag{71}$$

La función objetivo (69) maximiza la demanda total asignada. El conjunto de restricciones (70) son las restricciones de capacidad que evitan que la demanda total asignada a una planta $i \in I^*$ sea mayor que la capacidad de la misma. El conjunto de restricciones (71) se refieren a la naturaleza binaria de las variables de decisión.

El objetivo es enumerar explícitamente la menor cantidad de nodos a modo de encontrar la solución óptima del CMCLP rápidamente. En cualquier nodo, se describen los siguientes criterios de corte.

Sea

$\overline{Z}^*$, la mejor cota superior conocida,

$\underline{Z}^*$, el valor de la mejor solución factible (cota inferior) conocida,

$\overline{Z}$, la cota superior del nodo actual,

$\underline{Z}$, la cota inferior del nodo actual

1) Optimalidad.
    a. Si $(\overline{Z} - \underline{Z}) < 1$. Se ha encontrado el valor óptimo para esa rama si las demandas de los clientes son enteras. En el caso del nodo raíz, éste sería el valor óptimo del problema original CMCLP y el proceso de enumeración termina.
    b. Al resolver (*PA*). La exploración llega a su fin dado que se ha agotado esta rama y se tiene una solución óptima para el conjunto de instalaciones seleccionadas.
2) Acotamiento.
    a. Si $(\overline{Z} - \underline{Z}^*) < 1$. Ya no es posible encontrar un mejor valor para la $\underline{Z}^*$.

Cada vez que se ha cortado una rama, se debe hacer "*backtracking*". Esto es, ir a la lista de nodos sin explorar y procesar el siguiente siguiendo una disciplina LIFO (último en

entrar, primero en salir por sus siglas en inglés) y cuya exploración no haya terminado para continuar con el proceso de ramificación y exploración.

Si $\underline{Z} > \underline{Z}^*$ en cualquier momento de la ramificación, se debe actualizar el valor $\underline{Z}^* := \underline{Z}$. De igual forma, si $(\overline{Z}^* - \underline{Z}^*) < 1$ se ha encontrado la solución óptima del problema original y el algoritmo termina. Lo mismo ocurrirá cuando la lista de nodos sin explorar se vacíe.

La importancia de tener herramientas que obtengan buenas cotas superior e inferior radica en la aceleración de cortes en el árbol de ramificación lo que lleva a obtener la solución óptima del problema original más rápidamente.