

Un Algoritmo para el Problema de Asignación Generalizado con Múltiples Recursos.

Alejandro Crema.

Escuela de Computación.Facultad de Ciencias. U.C.V.

Apartado:47002 Caracas/Venezuela.

Fax:6627121 (Facultad de Ciencias/U.C.V.

RESUMEN

En este trabajo se presenta un Algoritmo de Ramificación y Acotación para el problema de Asignación Generalizado con Múltiples Recursos. La selección de los Problemas Candidato es hecha vía LIFO;se considera una cadena de Relaxaciones:la Lagrangiana conducente a un conjunto de problemas independientes tipo Mochila con Múltiples Recursos,la resultante de considerar únicamente los recursos peor evaluados en términos relativos y su correspondiente Relaxación Lineal y,finalmente,la resultante de eliminar las restricciones de Asignación.

Un conjunto de Lemas asociados a los valores optimales de las Relaxaciones y sus relaciones es demostrado.

El Algoritmo sugerido explota la cadena de Relaxaciones a los efectos de la clausura de los Problemas Candidato.La implantación actual se restringe a las dos últimas Relaxaciones consideradas.

Se presentan resultados computacionales satisfactorios,en microcomputadores personales,para problemas asociados a una aplicación con un máximo de 1000 variables binarias y 15 restricciones.

I.- Introducción:

El objetivo fundamental de los problemas clásicos de Asignación y sus extensiones es encontrar apareamientos óptimos de "Agentes" y "Tareas" de tal manera de minimizar un funcional de costos. En [11] y [12] los autores presentan un conjunto de problemas que pueden ser tratados en tal contexto.

Un común denominador de tales problemas es la presencia de cotas para la disponibilidad de un "Recurso". La asignación de una Tarea a un Agente trae como consecuencia el consumo de una predeterminada cantidad del Recurso y la asignación global debe respetar la disponibilidad asociada a cada uno de los Agentes. La extensión del problema al caso de Múltiples Recursos puede verse en [2], [4] y [12].

En este trabajo se presenta un Algoritmo de Ramificación y Acotación (R. y A.) para el Problema de Asignación Generalizado con Múltiples Recursos. En la sección II puede verse la formulación Matemática del problema. El Algoritmo es presentado en la sección III y en IV pueden verse los Resultados y Conclusiones.

II.- El Problema de Asignación Generalizado con Múltiples Recursos y su Formulación Matemática:

M Tareas han de ser asignadas a N Agentes de tal manera que se minimice el Costo Total a la vez que se respeten cotas en la disponibilidad de L Recursos asociados a cada uno de los Agentes.

Sean $I = \{1, \dots, N\}$, $J = \{1, \dots, M\}$ y $K = \{1, \dots, L\}$.

Se supone conocido: el Costo por asignar la j-ésima Tarea al i-ésimo Agente: $c(i, j)$, con i en I y j en J; el Consumo de unidades

del Recurso del tipo-k que debe hacer el i-ésimo Agente para realizar la j-ésima Tarea: $r(i,j,k)$, con i en I , j en J y k en K ; la disponibilidad del recurso del tipo-k asociada al i-ésimo Agente: $R(i,k)$, con i en I y k en K . Todos los parámetros involucrados en el problema se asumen positivos (con $R(i,k)$ estrictamente positivo) y cada Tarea debe ser asignada a un único Agente.

A continuación se presenta la formulación matemática del problema al cual nos referiremos como P en lo que resta del trabajo. Las variables de decisión binarias, $X(i,j)$, denotan si la j-ésima Tarea debe ser asignada al i-ésimo Agente ($X(i,j)=1$) o no ($X(i,j)=0$). En (1) puede verse el funcional de Costo Total. En (2) observamos las $N \times L$ restricciones asociadas a la disponibilidad de los Recursos. Las restricciones clásicas de Asignación aparecen en (3).

$$(1) \min \sum_{j=1}^N c(i,j) \times X(i,j) \quad \text{s.a:}$$

$$(2) \sum_{j=1}^N r(i,j,k) \times X(i,j) \leq R(i,k) \quad \forall i \in I, \forall k \in K$$

$$(3) \sum_{i=1}^N X(i,j) = 1 \quad \forall j \in J$$

$$(4) X(i,j) \in \{0,1\} \quad \forall i \in I, \forall j \in J$$

Si hacemos $L=1$ en P , estaremos en presencia del Problema de Asignación Generalizado (General Assignment Problem, - GAP -, en

la lit. del área) para el cual existen algoritmos eficientes ([1] y [10]).

III.- El Algoritmo:

III.1.- Marco Conceptual:

P es un problema de Programación Entera-Lineal-0-1. La estrategia de solución que se presenta es la de R. y A.. Los conceptos fundamentales y la terminología utilizada en relación a algoritmos basados en R. y A. pueden verse en el trabajo ,fundamental en el área, de Geoffrion y Marsten ([6]).

III.1.1.- Los Problemas Candidato (PC) :

A los efectos del presente estudio un Problema-Candidato, PC , resulta de añadir a P restricciones del tipo: $X(i,j)=1$ ó $X(i,j)=0$, para pares (i,j) seleccionados.

En otras palabras alguna(s) Tarea(s) es(son) asignada(s) a algún(os) Agente(s) o bien algún(os) Agente(s) es(son) excluidos del conjunto que puede realizar determinada(s) Tarea(s).

III.1.2.- Relajaciones de P:

III.1.2.1.- Relajación Básica:

La Relajación Básica consiste en excluir de P el bloque de restricciones tipo (2), asociadas a la disponibilidad de los Recursos ,obteniéndose un problema clásico de Asignación ([8]) cuya solución es trivial.

En lo que sigue nos referiremos a tal Relajación como a PASIG ó PC-ASIG.

III.1.2.2.- Relajaciones sucesivas

tipo Mochila:

(i) Considérese la Relajación Lagrangiana de P con respecto al bloque de restricciones (3), asociadas a la obligatoriedad de asignar cada Tarea a un único Agente, como sigue:

$$(1)' \text{ MIN } \sum_{i=1}^N \sum_{j=1}^M c(i,j) \times X(i,j) + \sum_{j=1}^M y(j) \times (1 - \sum_{i=1}^N X(i,j)) \text{ s.a:}$$

$$(2) \sum_{j=1}^M r(i,j,k) \times X(i,j) \leq R(i,k) \quad \forall i \in I, \forall k \in K$$

$$(4) X(i,j) \in \{0,1\} \quad \forall i \in I, \forall j \in J$$

En (1)' puede observarse la incorporación a (1) del bloque de restricciones (3) ponderadas según los multiplicadores $y(j)$, para $j=1, \dots, M$. Sean $i_1(j)$, con $j=1, \dots, M$, tales que $X(i_1(j), j) = 1$ en una solución óptima de PASIG y sean $i_2(j)$ tales que:

$$c(i_2(j), j) \leq c(i, j) \quad \forall i \in I \text{ t. q. } i \neq i_1(j), \forall j \in J$$

En [10], Ross y Soland sugieren el uso de $y(j) = c(i_2(j), j)$ para el caso de asignación con un solo recurso. En este trabajo se asumen los mismos multiplicadores. Obsérvese que $i_2(j)$ no es más que el Agente que realiza la j -ésima Tarea al mínimo costo con la excepción, posiblemente, de $i_1(j)$. Así, la relajación Lagrangiana, que denotaremos $P(y)$, puede escribirse como sigue:

$$(1)' \sum_{j=1}^M c(12(j), j) - \max \sum_{i=1}^N \sum_{j=1}^M (c(12(j), j) - c(1, j)) \times X(1, j) \text{ s.t.}$$

$$(2) \sum_{j=1}^M r(1, j, k) \times X(1, j) \leq R(1, k) \quad \forall i \in I, \forall k \in K$$

$$(4) X(1, j) \in \{0, 1\} \quad \forall i \in I, \forall j \in J$$

Tal interpretación permite separar $P(y)$ en N subproblemas independientes tipo Mochila con múltiples Recursos (uno para cada uno de los Agentes). Para una discusión extensa sobre los problemas tipo Mochila con múltiples recursos puede verse [4] y [13].

Puesto que es útil a los efectos de la presentación del Algoritmo, supongamos que hemos resuelto PASIG. Excepto para problemas triviales alguna(s) de las restricciones en el bloque (2) resultará(n) violada(s) por la solución obtenida para PASIG. Sean entonces:

$$J(1) = \{j \in J \text{ t.q. } 11(j) = 1\}, \quad \forall i \in I$$

$$V(1) = \{k \in K \text{ t.q. } \sum_{j \in J(1)} r(1, j, k) > R(1, k)\} \quad \forall i \in I$$

$$IINF = \{i \in I \text{ t.q. } V(1) \neq \emptyset\}$$

$$EXC(1, k) = \sum_{j \in J(1)} r(1, j, k) - R(1, k) \quad \forall i \in IINF \text{ y } k \in V(1)$$

Así, $J(i)$ es el conjunto de las Tareas asignadas al i -ésimo Agente en la solución de PASIG; $V(i)$ es el conjunto de recursos para los cuales se violó la disponibilidad asociada al i -ésimo Agente; $IINF$ es el conjunto de Agentes para los cuales existe algún Recurso en situación de infactibilidad y $EXC(i,k)$ es el exceso en el k -ésimo Recurso asignado al i -ésimo Agente. alguna(s) de las Tareas asignada(s) a Agentes en $IINF$ debe(n) ser reasignadas en la búsqueda de factibilidad. El mínimo incremento posible en tales casos, para el funcional de Costos, viene dado por:

$$DIF(j) = c(12(j), j) - c(11(j), j)$$

Considérense los siguientes problemas independientes tipo Mochila con Múltiples recursos, uno para cada uno de los Agentes en $IINF$, que denotaremos $M(i)$:

$$\text{MIN} \sum_{j \in J(i)} DIF(j) \times Y(i, j) \quad \text{s.a:}$$

$$\sum_{j \in J(i)} r(i, j, k) \times Y(i, j) \geq EXC(i, k) \quad \forall k \in V(i)$$

$$Y(i, j) \in \{0, 1\} \quad \forall j \in J(i)$$

La interpretación de las variables binarias en $M(i)$ es como sigue: Si $Y(i, j) = 1$ entonces la j -ésima Tarea debe ser reasignada en la búsqueda de factibilidad.

En el resto del trabajo si Q es un problema de optimización

entonces $v(Q)$ denota el valor de una solución óptima de Q (si existe) y $SF(Q)$ el conjunto de soluciones factibles.

Acerca de la relación entre $v(P(y))$, $v(PASIG)$ y $v(M(i))$ para i en $IINF$, podemos demostrar el siguiente Lema-1 (ver prueba en Anexo):

$$v(P(y)) = v(PASIG) + \sum_{i \in IINF} v(M(i))$$

(ii) En la búsqueda de un compromiso entre la calidad de las Cotas y la eficiencia Computacional se relajó $P(y)$ al considerar en (2) solo aquellas restricciones con $k(i)$ escogido como sigue:

sean,

$$EXR(i, k) = EXC(i, k) / R(i, k) \quad \forall i \in IINF, \forall k \in V(i)$$

$$k(i) \text{ t.q. } EXR(i, k(i)) = \max_{k \in V(i)} \{EXR(i, k)\}$$

Sea $PN(y)$ el problema resultante y $MN(i)$ los problemas tipo Mochila asociados (con un solo recurso: $k(i)$). Puede entonces demostrarse el siguiente Lema-2 (ver prueba en Anexo):

$$v(PN(y)) = v(PASIG) + \sum_{i \in IINF} v(MN(i))$$

$PN(y)$ es definido de tal manera que la búsqueda de factibilidad por reasignación de Tareas se concentra sobre

aquellos recursos en los cuales la violación de la disponibilidad es máxima en términos relativos.

Un último esfuerzo de conciliación Cotas-Eficiencia consiste en considerar la Relajación Lineal de $PN(y)$, que denotaremos $PNL(y)$, resultante de permitir:

$$0 \leq X(i, j) \leq 1 \quad \forall i \in I, \quad \forall j \in J$$

En tal caso, al definir análogamente $MNL(i)$, asumiendo la posibilidad de "reasignaciones parciales", con:

$$0 \leq Y(i, j) \leq 1 \quad \forall i \in IINF, \quad \forall j \in J(i)$$

, se pueden demostrar los siguientes Lemas (ver prueba en anexo)

$$\text{Lema-3 : } v(PNL(y)) = v(PASIG) + \sum_{i \in IINF} v(MNL(y))$$

$$\text{Lema 4 : } v(PASIG) \leq v(PNL(y)) \leq v(PN(y)) \leq v(P(y)) \leq v(P)$$

III.1.3.- Clausura de PC:

III.1.3.1.- Por acotamiento y/o

Factibilidad:

El Algoritmo que se presenta explota la cadena de Relajaciones en el sentido de que se intenta clausurar con PC-ASIG (alternativa asociada a la peor cota pero con la máxima eficiencia computacional) y si se fracasa se comienza a resolver, uno a uno, los problemas tipo Mochila ($MNL(i)$ ó $MN(i)$ ó $M(i)$) con la esperanza de clausurar sin completar toda la cadena. Adicional-

mente y como es usual PC es clausurado si la solución de PC-ASIG es factible en P.

III.1.3.2.- Por Infactibilidad:

Se desarrollaron dos Tests de Infactibilidad, como sigue:

(i) Al evaluar PC se detecta que existe una restricción particular en situación de infactibilidad ,digamos asociada al par (i,k) ,que no puede llevarse a factibilidad porque un subconjunto de las Tareas asignadas al Agente- i ,no reasignables como consecuencia de las asignaciones fijas realizadas durante el proceso,basta para implicar la infactibilidad.

(ii) Al evaluar PC se observa que existe una restricción particular, digamos la correspondiente a (i,k) ,que no puede llevarse a situación de factibilidad porque todas las Tareas asignadas al i -ésimo Agente en PC-ASIG y que son candidatas a reasignación,cumplen con: $r(i,j,k)=0$.

III.1.4.- Definición de las Variables de Ramificación y del Proceso de Ramificación:

En contraposición a los criterios usuales de Ramificación en los cuales, para el presente contexto, se seleccionaría una Tarea a ser definitivamente asignada a un Agente particular (ver el criterio de Ross y Soland,en [10],para un solo Recurso),se desarrolló un criterio de selección en el cual un Agente es excluido del conjunto al cual puede ser asignada una Tarea particular.

El criterio desarrollado es el siguiente:

(i) Seleccionar el par (i,k) , digamos $(IPEOR,KPEOR)$ cuya

restricción es la peor evaluada en términos relativos (en la solución de PASIG). Es decir:

$$EXR(IPEOR, KPEOR) = \max_{i \in IN} EXR(i, k(i))$$

(ii) Seleccionar la Tarea j , digamos JSALE, con el menor cociente diferencial de costos-recurso, es decir:

$$DIF(JSALE) / r(IPEOR, JSALE, KPEOR) = \min_{j \in JJ} DIF(j) / r(IPEOR, j, KPEOR)$$

$$\text{con } JJ = J(IPEOR) \cap \{j \text{ t.q.: } r(IPEOR, j, KPEOR) > 0\}$$

Aquí cabe la siguiente observación: Si JJ es vacío entonces el problema bajo evaluación es infactible según el Test de infactibilidad (ii) de III.1.3.2.

(iii) La restricción $x(IPEOR, JSALE) = 0$ es añadida a PC para definir el próximo problema a considerarse. El PC correspondiente a $X(IPEOR, JSALE) = 1$ es añadido a la Lista de Problemas Candidato.

La idea que sustenta el criterio desarrollado es la siguiente: Se intenta "arreglar" la restricción peor evaluada en términos relativos. La Tarea JSALE es la "primera candidata" a ser reasignada pues posee el menor cociente diferencial-recurso. Tal criterio debe resultar eficiente en aquellos problemas en los cuales el porcentaje de reasignaciones de Tareas sea bajo y adicionalmente el porcentaje de reasignaciones al segundo agente mejor evaluado sea alto. En V podemos observar evidencia empírica de la eficiencia.

III.2.- Algoritmo de Ramificación y Acotación:

A continuación se presenta el Algoritmo de R. y A. desarrollado. A los efectos de simplificar la exposición se utilizan solamente las Relajaciones PASIG y PNL(y). La extensión al caso en el cual se explota toda la cadena de Relajaciones es inmediata. El estilo de presentación del Algoritmo, ampliamente respetado en la literatura del área, es el sugerido por Geoffrion y Marsten ([6]).

ALGORITMO PROPUESTO

PASO-0: Iniciar ZMEJOR en un valor suficientemente grande.

PASO-1:

$\forall j \in J$ y $\forall k \in K$ sea $i_{\min}(j,k)$ t.q: $r(i_{\min}(j,k), j, k) \leq r(i, j, k) \forall i \in I$

$$\text{Si } (k \text{ t.q: } \sum_{j=1}^N r(i_{\min}(j,k), j, k) > \sum_{i=1}^N R(i, k)) \neq \emptyset$$

finaliza el proceso con mensaje de Infactibilidad.

PASO-2: Resolver PASIG.

PASO-3: Si la solución de PASIG es factible en P:

- (i) Se actualiza ZMEJOR como sigue: $ZMEJOR = v(PASIG)$.
- (ii) Se asignan los valores de las variables de decisión como sigue: $X(i1(j), j) = 1$ para cada j en J .
- (iii) Finaliza el proceso.

PASO-4: Hacer $PC = P$.

PASO-5: Si PC resulta infactible según Test (i) ir al PASO-15.

Si JJ es vacío (infactibilidad según (ii)) ir al PASO-15.

PASO-6: Se encuentra la Variable de Ramificación: (IPEOR, JSALE)

PASO-7: La restricción $X(\text{IPEOR}, \text{JSALE})=0$ es añadida a PC definiendo el próximo problema a ser evaluado. El problema con $X(\text{IPEOR}, \text{JSALE})=1$ ingresa a la LISTA. Se almacena el valor de la relajación asociada: $v(\text{PC-ASIG})$.

PASO-8: Se reasigna JSALE (equivale a resolver PC-ASIG).

PASO-9: Se intenta clausurar por acotación:

Si $v(\text{PC-ASIG}) \geq \text{ZMEJOR}$ PC es clausurado e ir PASO-15

PASO-10: Si la solución de PC-ASIG es factible en P:

(i) Se actualiza ZMEJOR como sigue: $\text{ZMEJOR} = v(\text{PC-ASIG})$.

(ii) Se actualiza el valor de las variables de decisión con los valores correspondientes a PC-ASIG.

(iii) Ir PASO-15 (PC es Clausurado).

Si no se conoce aún alguna solución factible ir al PASO-5.

PASO-11: Sea i en IINF. Se resuelve MNL(i) (asociado a PC).

PASO-12: $v(\text{PC-ASIG}) := v(\text{PC-ASIG}) + v(\text{MNL}(i))$.

PASO-13: Se intenta clausurar por acotación:

Si $v(\text{PC-ASIG}) > \text{ZMEJOR}$ ir al PASO-15 (PC es clausurado)

PASO-14: $\text{IINF} := \text{IINF} - i$.

Si $\text{Card}(\text{IINF})=0$ ir al PASO-5 (Se fracasó en el intento de clausurar PC por acotación).

Si no ir PASO-11.

PASO-15: Si la Lista de Problemas Candidato esta vacía finaliza el proceso. En caso contrario se encuentra el nuevo PC por aplicación de la regla LIFO e ir PASO-11.

IV.-Resultados y Conclusiones:

El Algoritmo está implantado en FORTRAN sobre una PS-50.

Las relajaciones consideradas en la implantación son PASIG y PNL(y).

En las Tablas 1,2 y 3 se presenta un resumen de los resultados computacionales asociados a una aplicación con 5 Agentes y 3 recursos (para un total de 15 restricciones). La cantidad de Tareas se hizo variar en 100,150 y 200 (para un total de 500,750 y 1000 variables binarias).

En las Tablas, Radio representa el peor cociente entre el total de recursos requeridos por las Tareas y el total disponible por los Agentes. Para cada Radio en el intervalo $[.4, .925]$, con un paso de 0.05, se resolvió un Lote de 10 problemas generados al azar respondiendo a la configuración particular de los datos de la aplicación:

- la disponibilidad relativa de los Agentes, para todo recurso, viene dada por: 4/11, 2/11, 2/11, 1/11, 2/11.

- los requerimientos de las Tareas son generados al azar en el conjunto $\{0,1,2\}$.

- el costo asociado a la asignación de la j-ésima Tarea al i-ésimo Agente, es proporcional a los requerimientos globales de la Tarea y al tiempo de desplazamiento (que se supone proporcional a la distancia) desde la posición del Agente (en

puntos del Plano preestablecidos) hasta la de la Tarea (generada al azar).

El resto de la nomenclatura de las Tablas es la siguiente: Nodos es el promedio de PC evaluados; Reasig. es el promedio de Tareas reasignadas; TCPU es el promedio de tiempo de procesamiento (en seg.); SF-1 es el PC, en promedio, en el cual se encontró la primera solución factible; MaxTCPU es el peor tiempo de procesamiento para el lote de problemas y Reasig. a su derecha es la cantidad asociada de Tareas reasignadas.

Los tiempos de procesamiento se mantuvieron dentro de límites aceptables, según la experiencia con cerca de 7200 problemas, salvo para problemas con Radio superior, aprox., a .90 (resultados similares se encontraron para tres configuraciones distintas, variando la disponibilidad relativa de los Agentes con radio crítico en .7 para el peor de los casos). En tales casos los problemas fueron resueltos, con optimalidad, en una PS-80 con coprocesador matemático, en tiempos inferiores al límite de procesamiento, establecido en 600 seg. (se estima, conservadoramente, una relación 1/10 sobre los tiempos). Los resultados cualitativos (v(P) vs. Radio; Reasig. vs. Radio; TCPU vs. Radio; TCPU vs. M; TCPU vs. Reasig.) son todos coherentes con lo esperado.

Algunas posibilidades para mejorar el Algoritmo implantado, no evaluadas hasta la fecha, son:

- Implantar toda la cadena de Relajaciones: (PASIG, PNL(y), PN(y) y P(y)) y, adicionalmente, considerar distintas estrategias de solución (puede resolverse toda la

cadena para cada i en IINF ó resolver las Relaxaciones en el orden sugerido por la cadena propuesta). Aquí cabe la siguiente observación: el esfuerzo computacional involucrado en la resolución de PNL(y) no está necesariamente perdido a los efectos de solucionar PN(y) y así sucesivamente.

- Considerar relajaciones adicionales:

Un problema tipo GAP al escoger un Recurso como el único activo y/o a partir de la definida por Jörnsten y Näsberg en [9] para el GAP y/o a partir de la relajación Mixta (Lagrangiana y Surrogate combinadas) que puede verse en [7], utilizada para P, con "pocas" restricciones (unas 5 según la referencia), por Gavish y Pirkul ([2]).

- Considerar reglas no-LIFO.

- Seleccionar los Agentes en IINF en un orden predeterminado según su potencial contribución al valor del funcional de la relajación de turno.

- Considerar otros criterios de ramificación, por ejemplo: el menor cociente diferencial-recurso tomado sobre todas la ternas (i, j, k) con i en IINF, j en $J(i)$ y k en $V(i)$ en lugar de restringir la escogencia a $i = \text{IPEOR}$ y $k = k(i)$.

- Considerar estrategias para la generación de soluciones factibles basadas en la información proveniente de las relajaciones.

Radio	Nodos	Reasig.	TCPU	SF-1	MaxTCPU	Reasig.
.50	4	2	1.28	2	2.36	5
.75	34	12	6.28	13	18.95	16
.80	254	18	52.50	22	120.89	22
>.90	2878	23	>600.00	36	-	-

M = 100 Restricciones: 15

TABLA-1

Radio	Nodos	Reasig.	TCPU	SF-1	MaxTCPU	Reasig.
.50	3	1	1.64	2	4.61	7
.75	47	14	13.53	16	31.47	17
.80	253	23	88.05	24	248.87	28
>.90	1931	27	>600.00	33	-	-

M = 150 Restricciones: 15

TABLA-2

Radio	Nodos	Reasig.	TCPU	SF-1	MaxTCPU	Reasig.
.50	8	4	4.09	5	10.44	12
.75	121	25	50.72	26	206.35	23
.80	603	27	283.82	35	571.77	32
>.90	1489	39	>600.00	43	-	-

M = 200 Restricciones: 15

TABLA-3

ANEXO

A continuación se presentan las pruebas de los Lemas introducidos en este trabajo:

$$\text{Lema-1: } v(P(y)) = v(PASIG) + \sum_{i \in IINF} v(M(i))$$

Prueba del Lema-1:

(1) De la definición de $P(y)$ se sigue que:

$$\text{Si } i \notin IINF, \text{ podemos hacer } X(i, j) = 1 \text{ si } i1(j) = 1$$

$$\text{Si } i \in IINF, \text{ podemos hacer } X(i, j) = 0 \text{ si } j \notin J(i)$$

(2) Consideremos los sub-problemas auxiliares $Ma(i)$, para cada i en $IINF$, definidos como sigue:

$$\text{MAX } \sum_{j \in J(i)} (c(i2(j), j) - c(i1(j), j)) \times X(i, j) \text{ s.a:}$$

$$\sum_{j \in J(i)} r(i, j, k) \times X(i, j) \leq R(i, k) \quad \forall k \in K$$

$$X(i, j) \in \{0, 1\} \quad \forall j \in J(i)$$

(3) De (1) y (2) se sigue de inmediato que :

$$\begin{aligned} v(P(y)) &= \sum_{j=1}^M c(i2(j), j) - \sum_{i \in IINF} \sum_{j \in J(i)} (c(i2(j), j) - c(i1(j), j)) \\ &\quad - \sum_{i \in IINF} v(Ma(i)) \end{aligned}$$

(4) Consideremos el cambio de variable $X(i, j) = 1 - Y(i, j)$ en cada $Ma(i)$ y definamos el sub-problema auxiliar $Ma2(i)$ para cada

i en IINF como sigue:

$$\text{MAX} \sum_{j \in J(i)} (c(i2(j), j) - c(i1(j), j)) \times (-Y(i, j)) \text{ s.a.}$$

$$\sum_{j \in J(i)} r(i, j, k) \times Y(i, j) \geq \sum_{j \in J(i)} r(i, j, k) - R(i, k) = \text{EXC}(i, k) \quad \forall k \in K$$

$$Y(i, j) \in \{0, 1\} \quad \forall j \in J(i)$$

(5) De la definición de Ma2(i) se sigue que:

$$v(\text{Ma}(i)) = \sum_{j \in J(i)} (c(i2(j), j) - c(i1(j), j)) + v(\text{Ma2}(i)) \quad \forall i \in \text{IINF}$$

(6) De (3), (4) y (5) se sigue que:

$$v(P(y)) = \sum_{i \in \text{IINF}} \sum_{j \in J(i)} c(i1(j), j) + \sum_{i \in \text{IINF}} \sum_{j \in J(i)} c(i1(j), j) - \sum_{i \in \text{IINF}} v(\text{Ma2}(i))$$

(7) Claramente se tiene que:

$$v(\text{PASIG}) = \sum_{i \in \text{IINF}} \sum_{j \in J(i)} c(i1(j), j) + \sum_{i \in \text{IINF}} \sum_{j \in J(i)} c(i1(j), j)$$

(8) De las definiciones de M(i) y Ma2(i) se tiene que:

$$v(M(i)) = -v(\text{Ma2}(i)) \quad \forall i \in \text{IINF}$$

(9) De (6), (7) y (8) se concluye:

$$v(P(y)) = v(\text{PASIG}) + \sum_{i \in \text{IINF}} v(M(i))$$

$$\text{Lema-2: } v(\text{PN}(y)) = v(\text{PASIG}) + \sum_{i \in \text{IINF}} v(\text{MN}(i))$$

Prueba del Lema-2:

La prueba del Lema-2 es análoga a la del Lema-1 colocando

$k=k(i)$ y no $\forall k \in K$ (con lo cual se redefinen $Ma(i)$ y $Ma2(i)$)

$PN(y)$ en lugar de $P(y)$ y $MN(i)$ en lugar de $M(i)$.

$$\text{Lema-3: } v(PNL(y)) = v(PASIG) + \sum_{i \in I \cap J} v(MNL(i))$$

Prueba del Lema-3:

La observación (1) en la prueba del Lema-1 sigue siendo válida en $PNL(y)$. Se redefinen $Ma(i)$ y $Ma2(i)$ en forma análoga a como se hizo en la prueba del Lema-1 pero con las restricciones

$$0 \leq X(i, j) \leq 1$$

La prueba es análoga a la del Lema-1, con los subproblemas auxiliares adecuados y con:

$PNL(y)$ en lugar de $P(y)$; $MNL(y)$ en lugar de $M(y)$

$k=k(i)$ en lugar de $\forall k \in K$.

$$\text{Lema-4: } v(PASIG) \leq v(PNL(y)) \leq v(PN(y)) \leq v(P(y)) \leq v(P)$$

Prueba del Lema-4:

(1) $P(y)$ es una Relajación Lagrangiana de P con lo cual la última desigualdad a la derecha es válida.

(2) El Lema-3 y la siguiente cadena de inclusiones determinan la validez del resto de las desigualdades:

$$SF(PNL(y)) \supseteq SF(PN(y)) \supseteq SF(P(y))$$

BIBLIOGRAFIA

- [1] Fisher, M., Jaikumar, R. y Van Wassenhove, L.: A multiplier adjustment method for the assignment problem. *Management Science* 32 (1986) 1095-1103.
- [2] Gavish, B. y Pirkul, H.: Zero-one integer programs with few constraints-efficient branch and bound algorithms. *European Journal of Operational Research* 22 (1984) 35-43.
- [3] Gavish, B. y Pirkul, H.: Zero-one integer programs with few constraints-efficient-lower bounding theory. *European Journal of Operational Research* 21 (1985) 213-224.
- [4] Gavish, B. y Pirkul, H.: Efficient algorithms for solving multiconstraint zero-one knapsacks to optimality. *Mathematical Programming* 31, 78-105 (1985).
- [5] Geoffrion, A.M.: Lagrangean relaxation for integer programming. *Mathematical Programming Study* 2:82-114, 1974.
- [6] Geoffrion, A.M. y Marsten, R.E.: Integer programming algorithms: a framework and state of the art survey. *Management Science*, Vol 18(9):465-491. 1972.
- [7] Guignard, M. y Rosenwein, M.: An application oriented guide for designing lagrangean dual ascent algorithms. *European Journal of Operational Research* 43(1989) 917-205.
- [8] Hadley, G.: *Linear programming*. Addison Wesley Reading Massachussetts. 1962. Cap 1, pp 1-182.
- [9] Jornsten, K. y Nasberg, M.: A new lagrangean relaxation approach to the generalized assignment problem. *European Journal of Operational Research* 27 (1986) 313-323.
- [10] Ross, G.T. y Soland, R.M.: A branch and bound algorithm for the generalized assignment problem. *Mathematical Programming*, Vol 8(1975) pp 91-105.
- [11] Ross, G.T. y Soland, R.M.: Modeling facility location problems as generalized assignment problems. *Management Science*, Vol 24(3) 1977, pp 345-347.
- [12] Ross, G.T. y Zoltners, A.A.: Weighted assignment models and their application. *Management Science* Vol 25(7) pp 683-694 July 1979.
- [13] Shih, W. (1979): A branch and bound method for the multiconstraint zero-one knapsack problem. *J. Opl. Res.* Vol 30 pp 369 to 378 (1979).