

Algoritmo de Generación de Columnas: Una revisión desde su aplicación al problema de asignación de cupos escolares

Column Generation Algorithm: A revision from its application to the Student Assignment Problem

*Pablo Andrés Maya**

Departamento de Ingeniería Industrial, Universidad de Antioquia, Calle 67 N.º 53-108, A.A. 1226, Medellín, Colombia.

(Recibido el 26 de febrero de 2008. Aceptado el 30 de junio de 2008)

Resumen

El algoritmo de generación de columnas (GC) se cita usualmente como alternativa para la solución de problemas de optimización de gran escala; este artículo aborda la presentación de dicho algoritmo en el contexto de la asignación de cupos escolares en el sistema de educación pública del distrito de Bogotá. Se discuten algunas de las falencias evidenciadas en la puesta en práctica del algoritmo, presentando estrategias para enfrentarlas. Se ilustra además, la forma como la generación de columnas se acopla dentro del algoritmo *Branch and Price* para dar solución a problemas de mayor complejidad.

----- *Palabras clave:* Algoritmo de generación de columnas, problema de asignación, algoritmo *Branch and Price*.

Abstract

The Column Generation Algorithm (CGA) is commonly cited in the bibliography as alternative to solve large scale optimization problems. This article deals with the description of the GC algorithm in the context of the student assignment to the public schools. Some of the weaknesses and shortcomings encountered in the implementation and application to a real problem are discussed and some strategies to sort it out are presented. This work pointed out how the CGA could be used within a general Branch and Price procedure to solve problems with additional constrains.

----- *Keywords:* Column generation algorithm, assignment problem, branch and price algorithm.

* Autor de correspondencia: teléfono: + 57 + 4 + 219 53 66, fax: + 57 + 4 + 263 82 82, correo electrónico: pmaya@udea.edu.co (P.A. Maya).

Introducción

El algoritmo de generación de columnas es reconocido en la literatura como una estrategia de solución para cierto tipo de problemas lineales de gran escala. El algoritmo toma mayor relevancia cuando se considera su aplicación dentro de estrategias para la solución de problemas enteros o mixtos, bien sea mediante métodos exactos como el algoritmo *Branch and Price* o cuando se emplea para la construcción de heurísticas para la solución de dicho tipo de problemas. En este artículo se pretende recoger la experiencia adquirida en torno al algoritmo de generación de columnas, exponiendo su aplicación en el marco de un problema real, presentando algunas de las estrategias empleadas para mejorar su desempeño y señalando algunas de las oportunidades de investigación que el trabajo hasta ahora realizado permite identificar, en relación tanto con el algoritmo como con la situación problemática abordada con él.

Este artículo está organizado de la siguiente manera: inicialmente se describe el contexto del problema de asignación en el marco del cual se expondrá el algoritmo de generación de columnas y se plantea la pregunta de investigación. Se presenta después la formulación matemática del modelo. En las dos secciones siguientes, se describe el algoritmo de generación de columnas y se discuten aspectos relacionados con su implementación. En la siguiente sección se discute el problema de convergencia del algoritmo. Posteriormente, se ilustra la forma como el algoritmo de generación de columnas podría acoplarse dentro de un esquema *Branch and Price* para la solución de problemas más complejos y considerando extra restricciones. Por último, se presentan las conclusiones y se presentan los lineamientos para trabajos futuros.

Descripción de la situación problemática

Antecedentes

El estudio del proceso de asignación de cupos a los estudiantes en el sistema de educación pública de Bogotá desarrollado en 2005 por Maya

[1], pretendía identificar los modelos de apoyo al proceso de la toma de decisiones en la asignación que en dicho proceso subyace. El algoritmo de generación de columnas se vislumbró como una estrategia de solución prometedora para el modelo matemático formulado, dada su pertinencia en la solución de problemas de gran escala con una estructura similar a la del problema que se pretendía abordar. Sin embargo, en su implementación se evidenciaron algunos de los aspectos que son ya reconocidos en relación con su convergencia, lo cual obligó la búsqueda de alternativas de solución diferentes. Este trabajo retoma las experiencias adquiridas al intentar emplear el algoritmo de generación de columnas en el marco de la instancia de asignación de cupos escolares, identificando los puntos que condicionan su aplicación y señalando líneas de investigación al respecto.

Descripción del problema de asignación de cupos escolares

La instancia del problema que se abordó podría describirse brevemente de la siguiente forma:

La Secretaria de educación distrital de Bogotá (SED) define el proceso de matrículas para los establecimientos educativos distritales. Con base en la definición de este proceso, el conjunto de estudiantes cubiertos por el sector oficial es caracterizado de la siguiente forma:

- A. *Estudiantes Antiguos*: Matriculados para el año anterior en los Institutos Educativos distritales (IED), Centros Educativos Distritales (CED), los colegios distritales en concesión y los colegios privados en convenio.
- B. *Estudiantes Nuevos*: Aquellos que deseen acceder al servicio, que no se encuentran registrados como matriculados en la base de datos del sistema de la SED. Dentro de este conjunto de estudiantes podrían distinguirse:
 - i.) *Estudiantes Nuevos transición y primero*: Aspiran a ingresar al sistema educativo público del distrito en el grado cero (0º o transición) y primer grado (1º). El

distrito garantizará el cupo escolar para los estudiantes de este grupo.

ii.) *Estudiantes Inscritos Contra Oferta Educativa*: Estudiantes interesados en ingresar al sistema educativo oficial para cualquiera de los grados o niveles ofrecidos. Esta población se denomina inscritos contra oferta educativa disponible.

Dentro de cada uno de estos conjuntos de estudiantes, es importante distinguir aquellos pertenecientes a Grupos Vulnerables (desvinculados del conflicto armado, desplazados, hijos de desmovilizados, hijos de víctimas del secuestro y desaparición forzada entre otros) a los cuales se otorgará prioridad en la asignación de cupos.

Las etapas para desarrollar la asignación de cupos están definidas para cada uno de estos grupos de estudiantes. La asignación de los estudiantes provenientes del DABS y el ICBF así como los estudiantes Nuevos, son instancias particulares del problema de asignación que se desea abordar y estudiar. El proceso de asignación para este conjunto de estudiantes se desarrolla, principalmente, de acuerdo con el siguiente lineamiento de la SED:

“La SED asignará los cupos escolares preferiblemente en las opciones seleccionadas por el padre, madre o acudiente. De no existir cupo en las opciones solicitadas, se asignará en las instituciones cercanas a su lugar de residencia (de acuerdo con la dirección suministrada en el formulario). Si una vez surtido este proceso, no es posible asignar un cupo, la SED recurrirá a otras estrategias de cobertura” (SED, Resolución 4040 de 2004)

La asignación de cupos se lleva a cabo de acuerdo con el cronograma del proceso de matrículas, de acuerdo con la información suministrada por la SED, el total de estudiantes Nuevos inscritos hasta marzo de 2005 ascendía 223.537 de los cuales cerca de 170.753 se encontraban matriculados. La distribución de la demanda generada por los alumnos nuevos en cada uno de los grados, hace que el problema de asignación involucrado en este proceso deba considerar en algunos casos

más de 30.000 estudiantes simultáneamente. Este hecho genera complejidad dentro del proceso de asignación y otorga mayor importancia a su estudio y análisis.

Respecto a la forma como actualmente se desarrolla el proceso de asignación de cupos en el distrito, es notoria su similitud con los algoritmos *greedy* diseñados para problemas similares, como el problema de asignación generalizada (GAP) o el problema de transporte. Si bien el procedimiento actual permite encontrar una solución factible al problema, podría perjudicar en el objetivo de encontrar una asignación que favorezca a un mayor número de estudiantes, respecto a la asignación de colegios más cercanos y convenientes a sus necesidades y expectativas.

Descripción del problema de investigación

Las dificultades que evidenció el algoritmo de generación de columnas al ser considerado en la solución del problema de asignación de cupos escolares, permitió plantearse diferentes cuestionamientos que dan origen al problema de investigación que se pretende abordar en este documento y que mediante el trabajo realizado señala los lineamientos para futuros trabajos en la misma dirección. La pregunta fundamental de investigación es:

¿Cómo puede mejorarse la eficiencia del algoritmo de generación de columnas para ser implementado en la solución de problemas de asignación como el de la instancia considerada?

De este cuestionamiento principal se desprenden distintas preguntas, algunas de las cuales se discuten en este documento y otras que continúan siendo tema de investigación. Algunas de dichas preguntas son:

¿Cuáles estrategias pueden considerarse para mejorar la convergencia y eficiencia del algoritmo de generación de columnas?

¿Cuáles de las estrategias identificadas son aplicables en la solución del problema de asignación considerado?

¿Cómo puede acoplarse el conocimiento adquirido respecto a al algoritmo de generación de columnas dentro de estrategias de solución fundamentadas en él (Branch and Price y Heurísticas fundamentadas en GC)?

Formulación de problema

Se pretende abarcar el problema de la asignación de cupos escolares para el conjunto de estudiantes nuevos (DABS-ICBF, Transición y primero e inscritos contra la oferta educativa) para el sistema de educación oficial del distrito. De manera general, el problema puede enunciarse de la siguiente forma:

Dado un conjunto de estudiantes que aspiran a un cupo, en determinado grado, en alguno de los colegios distritales, y dado un conjunto de colegios, con una capacidad (oferta) determinada (por grado); deben asignarse los estudiantes a los colegios, de modo que se maximice una función del beneficio (en términos de la distancia y prioridades, entre otras) para dicha asignación.

Este problema puede formularse como se presenta en el problema P1, que en adelante se denominará formulación estándar

Sea:

- I : Conjunto de Estudiantes para asignar. Indexado con $i, |I| = n$
- J : Conjunto de Instituciones. Indexado con $j, |J| = m$
- c_{ij} : Beneficio de asignar el estudiante i a la institución j
- b_j : Número de estudiantes que está en capacidad de atender la institución j

$$x_{ij} = \begin{cases} 1, & \text{si el estudiante } i \text{ es asignado al colegio } j \\ 0, & \text{de lo contrario} \end{cases}$$

$$\begin{aligned} \text{Max} \quad & \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \\ \text{s.a.} \quad & \sum_{j \in J} x_{ij} = 1 \quad \forall i \in I \end{aligned} \quad (1)$$

$$\sum_{i \in I} x_{ij} = b_j \quad \forall j \in J \quad (2)$$

$$x_{ij} \in \{0,1\} \quad \forall i \in I \wedge \forall j \in J \quad (3)$$

El conjunto de restricciones (1) y (2) garantizan, respectivamente, que cada estudiante será asignado a algún colegio y que el número de estudiantes

asignados a un colegio particular no sobrepasara la capacidad de cupos de este. Nótese que se asume que el número de estudiantes que deben asignarse es igual al número de cupos en los colegios. Para la instancia descrita esta condición no se satisface; sin embargo, puede modificarse ligeramente el problema adicionando un nodo ficticio, sin afectar su solución óptima.

Es importante resaltar algunos aspectos respecto al problema básico presente en el proceso de asignación de cupos escolares (PI):

- La estructura particular de este problema puede asemejarse a algunos problemas típicos y bien estudiados [2] y [3]. Específicamente el problema en (PI) se asemeja a un Problema de Asignación Generalizada (GAP), en el que el costo o consumo de la asignación de cada estudiante i (ítem) al colegio j (recurso) es de una unidad. De igual forma, el problema en (PI) puede ser visto como un caso particular de un problema de transporte en el cual la demanda de cada uno de los nodos de demanda es unitaria, o como el caso particular de un problema de asignación, asumiendo que existen para cada colegio un número, igual a su capacidad, de cupos (objetos) iguales.
- El problema en (PI) es bastante general, en él se asume que cualquier estudiante puede ser asignado a cualquier colegio y que las restricciones de asignación para todos los colegios son iguales. Sin embargo, podría ser necesario considerar restricciones adicionales respecto a los colegios a los cuales un estudiante dado puede ser asignado o las asignaciones factibles de cada colegio. Posteriormente se discutirá la forma de abordar estas consideraciones.
- El valor c_{ij} que representa el beneficio de asignar el estudiante i a la institución j , es en realidad el resultado de una función que involucra aspectos como la prioridad de asignación, las preferencias expresadas por los padres y la ubicación (distancia) del estudiante respecto a la institución.

Las siguientes secciones describen una estrategia para la solución de este problema en instancias de gran escala, es decir con un número considerable de estudiantes e instituciones. Además, se propone la forma como esta estrategia puede ser considerada en la solución del problema cuando se involucran restricciones adicionales para hacerlo más cercano a la realidad.

Estrategia de solución: Generación de Columnas

La estrategia de solución que se expone se fundamenta en el principio de descomposición y pretende exponer, en un caso real, el uso del esquema de generación de columnas, ideas que fueron desarrolladas por Dantzing y Wolfe y Gilmore y Gomory respectivamente ([4] y [5]).

Descomposición de Dantzing-Wolfe

El Problema en la formulación estándar, (P1), puede ser entendido como la partición del conjunto de estudiantes en subconjuntos que pueden ser asignados a cada institución maximizando en dicha partición el beneficio de la asignación [6]. Bajo esta óptica el problema puede reformularse de una forma alterna considerando:

- i.) $K_j = \{x_1^j, x_2^j, \dots, x_{k_j}^j\}$ Conjunto de las asignaciones factibles para la institución j
- ii.) $x_k^j = \{x_{1k}^j, x_{2k}^j, \dots, x_{nk}^j\}$ Corresponde a la k -ésima asignación factible para la institución j , si:

$$\sum_{i \in I} x_{ik}^j = b_j \quad x_{ik}^j \in \{0,1\} \quad (4)$$

iii.) Se define y_k^j para todo $j \in J$ y $k \in K_j$ como:

$$y_k^j = \begin{cases} 1, & \text{Si la asignación factible } x_k^j \text{ es seleccionada para} \\ & \text{la institución } j \\ 0, & \text{en caso contrario} \end{cases}$$

Así el problema se puede formular como:

$$\text{Max} \quad \sum_{j \in J} \sum_{k \in K_j} \left(\sum_{i \in I} c_{ij} x_{ik}^j \right) y_k^j$$

$$\text{s.a.} \sum_{j \in J} \sum_{k \in K_j} x_{ik}^j y_k^j = 1 \quad \forall i \in I \quad (5)$$

$$\sum_{k \in K_j} y_k^j = 1 \quad \forall j \in J \quad (6)$$

$$y_k^j \in \{0,1\} \quad \forall j \in J \wedge \forall k \in K_j \quad (7)$$

La formulación en (P2) se denominará en adelante *formulación desagregada*. En esta formulación, las restricciones (6) y (7) garantizan, respectivamente, que cada estudiante es asignado a una institución y que para cada institución es seleccionada una asignación factible. Esta formulación corresponde a la descomposición de *Dantzing-Wolfe* para la formulación estándar (P1), donde las restricciones de capacidad han sido definidas para el subproblema en el algoritmo de generación de columnas [7]

Descripción del Algoritmo de Generación de Columnas

El algoritmo de generación de columnas permite resolver la relajación lineal del problema en la formulación desagregada (P2). Si bien el interés recae en la solución del problema entero, posteriormente se evidencia la utilidad que la solución de la relajación lineal tiene para este fin.

La idea del algoritmo es resolver el problema lineal por medio de la aplicación del *Método Simplex*; sin embargo, dada la enorme cantidad de columnas (asignaciones factibles) para cada colegio, en lugar de evaluar todas las columnas respecto a su costo reducido, para determinar cual debe adicionarse a la base, el problema de encontrar la columna con el mayor costo reducido es en sí mismo un conjunto de m problemas de optimización. El procedimiento desarrollado en el algoritmo puede describirse de la siguiente forma [8] y [9]:

1. *Inicialización.* Debe disponerse de un subconjunto de columnas que provean el *Problema Lineal Maestro Restringido (PLMR)* para el cual sea posible determinar una solución básica factible, de modo que cualquier solución básica factible del problema restringido lo es también para el problema no restringido.
2. *Prueba de Optimalidad para el Problema no Restringido.* Debe determinarse si la solución factible encontrada es óptima para el problema restringido, para ello se determina si es posible hallar una columna con costo

reducido positivo para adicionar a la base, usando para ello los precios duales asociados a la solución del problema restringido. De no existir dicha columna la solución actual de problema restringido es óptima, y también lo es para el problema no restringido, con lo cual finaliza el algoritmo.

3. *Generación de una Nueva Columna.* En caso de no satisfacerse las condiciones de optimalidad, una (o varias) columna(s) con costo reducido positivo debe(n) ser adicionada(s) generándose un nuevo Problema Maestro Restringido, el cual puede ser reoptimizado para volver al segundo paso del algoritmo. El problema auxiliar empleado para generar esta columna se describe posteriormente.

Algunos de los elementos más importantes del algoritmo de generación de columnas, para la instancia del problema de asignación que se pretende abordar, se describen a continuación.

Definición del Problema Maestro Restringido Inicial

El algoritmo debe comenzar con un conjunto de columnas que provean una solución básica factible para la relajación lineal del problema original, este conjunto de columnas constituyen el Problema Maestro Restringido Inicial. Una alternativa válida, que de hecho corresponde a la empleada en la implementación del algoritmo, consiste en construir una asignación factible para cada colegio y adicionar a este conjunto de m columnas una columna artificial adicional. La columna para cada institución podría corresponder a la solución del *Problema Knapsack* asociado, es decir a asignar los b_j estudiantes con mayor beneficio de asignación a dicho colegio. La columna artificial es tal que sus primeros n elementos son uno y los m restantes son cero, de esta forma esta columna junto con las columnas asociadas a cada colegio permite siempre construir una solución básica factible. El beneficio de la columna artificial deberá ser $-M$ donde M es una constante suficientemente grande (por ejemplo, la suma de todos los beneficios).

Definición del Problema Auxiliar (Pricing Problem)

A partir del Problema Maestro Restringido se generan las columnas que vayan siendo necesarias hasta resolver óptimamente el problema. La generación de las columnas se hace por medio de un problema auxiliar denominado *Pricing Problem*.

La columna asociada a y_k^j , es decir $\{x_{1k}^i, x_{2k}^i, \dots, x_{nk}^i, e_j\}$ que se adiciona a la base debe satisfacer las siguientes condiciones:

- i.) Debe tener costo reducido positivo, es decir

$$\bar{c}_k^j > 0$$

- ii.) Debe corresponder a una asignación factible para alguna institución, es decir $\sum_{i \in I} x_k^i = b_j$ para algún j en el conjunto de Instituciones

Bajo estas consideraciones la columna que debe adicionarse se determina como:

$$Max = \{Z(KP_j) - v_j\} \quad (8)$$

En donde, v_j representa el precio dual asociado con la restricción de convexidad para la institución j , restricción (7), en la solución óptima del Problema Maestro Restringido. $Z(KP_j)$ corresponde al valor de la solución óptima del *Problema Knapsack* asociado a la institución j , (P3).

$$\begin{aligned}
 Max \quad & \sum_{i \in I} (c_{ij} - \mu_i) x_i^j \\
 s.a. \quad & \sum_{i \in I} x_i^j = b_j \\
 & x_i^j \in \{0,1\} \quad \forall i \in I
 \end{aligned} \quad (9)$$

En este problema μ_i corresponde al valor óptimo de la variable dual asociada con la restricción de partición para el estudiante i , restricción (6), en la solución óptima del Problema Maestro Restringido

Consideraciones Respecto a la Implementación del Algoritmo

El algoritmo de generación de columnas fue implementado para dar solución a instancias del problema objeto de estudio, en *Xpress-MP*. Algunos aspectos relevantes abordados durante la implementación así como consideraciones y conclusiones obtenidas por medio de la ejecución del modelo, que ameritan ser detallados se presentan a continuación.

Adición y borrado de columnas

Varias estrategias a seguir en el algoritmo de generación de columnas pueden ser implementadas, con el fin de determinar la o las columnas para adicionar. Asumiendo que en cada iteración una única columna es adicionada, pueden plantearse al menos dos estrategias distintas: Adicionar la columna que genere el mayor costo reducido o adicionar la primera columna con costo reducido positivo encontrada.

La primera estrategia requiere resolver $m = |J|$ Problemas *Knapsack* en cada iteración, lo cual puede hacer que el tiempo para realizar cada una de estas se incremente, sin embargo, esta estrategia genera la disminución más grande posible en el valor de la función objetivo, para cada iteración. La segunda estrategia resuelve cuando más $m = |J|$ Problemas *Knapsack* por iteración, en la práctica el número de problemas resueltos en cada iteración es considerablemente menor que m , este hecho puede contribuir a reducir el tiempo de solución, aunque posiblemente el número de iteraciones requeridas sea mayor. El algoritmo implementado para este problema particular, hace uso de la primera estrategia, esto debido principalmente a que como se mencionó anteriormente la solución de cada uno de los *Problemas Knapsack* es de menor complejidad, puesto que para resolverlos deben seleccionarse, en orden decreciente respecto al beneficio generado, los estudiantes que son asignados a un colegio determinado.

Otro aspecto importante considerado en la implementación del algoritmo es el gran número

de columnas que posiblemente debe adicionarse hasta encontrar la solución óptima de la relajación lineal del problema en (*P2*), pues esto puede afectar la eficiencia del algoritmo e incluso aumentar los requerimientos computacionales para su solución. Con el fin de atenuar el efecto de este aspecto, se implementó un procedimiento para eliminar, cada cierto número de iteraciones, aquellas columnas con costo reducido negativo, lo cual puede implicar en algunos casos que columnas que ya habrían sido generadas vuelvan a ser consideradas en iteraciones posteriores, pero alivia en parte el rápido incremento del número de columnas consideradas y por ende del tiempo de solución de cada iteración.

Solución del problema entero (no relajado) P2

Un aspecto de importancia respecto a la solución del problema entero, es decir el problema no relajado, está relacionado con el hecho de que la solución de ambos problemas, la relajación lineal y el problema entero, coinciden [1]. Dada esta consideración, el algoritmo de generación de columnas puede usarse para determinar la solución del problema básico de optimización inmerso en el proceso de asignación de cupos escolares; sin embargo, es importante señalar que la solución de instancias con restricciones adicionales, es decir que no correspondan a la forma básica del problema presentado, no necesariamente satisfacen que la solución del problema relajado sea entera, en cuyos casos sería necesario recurrir a algoritmos como el de *Branch-and-Price*, el cual se describe posteriormente.

Convergencia del Algoritmo

El problema de convergencia lenta del algoritmo de generación de columnas es ampliamente reconocido en la literatura. Generalmente, pueden hallarse soluciones cercanas a la solución óptima rápidamente, sin embargo el progreso obtenido en la función objetivo cuando se está cerca del valor óptimo no es significativo [10]; este hecho puede generar que el algoritmo requiera de un tiempo considerable hasta obtener la optimali-

dad de la solución e incluso tarde en decretar la optimalidad de soluciones óptimas degeneradas. Este hecho se conoce como el efecto *Tailing-off*, denominado así desde el trabajo realizado por Gilmore y Gomory [5].

Dos aspectos se destacan en la literatura respecto al problema de convergencia del algoritmo de generación de columnas y su aplicación particular al problema de asignación o transporte. En primer lugar, el problema presenta una notoria degeneración, lo cual puede afectar la eficiencia del algoritmo, dado que un gran número de los pivotes realizados dan como resultado la adición de columnas con costo reducido positivo, pero que no mejoran el valor de la función objetivo, es decir, pivotes degenerados [3]. El segundo aspecto que reviste gran importancia en la literatura relacionada, es el comportamiento del valor de las variables duales durante el proceso de solución; estas variables presentan grandes oscilaciones durante el proceso y su convergencia no es suave, este comportamiento es considerado como uno de los aspectos principales en relación con la convergencia y la ausencia de dicho comportamiento es vista como una propiedad deseable [10].

La figura 1 presenta la forma como evoluciona la solución primal y la solución dual (en realidad una cota para el valor de la solución dual, la cual se discutirá posteriormente) para una instancia de prueba pequeña del problema.

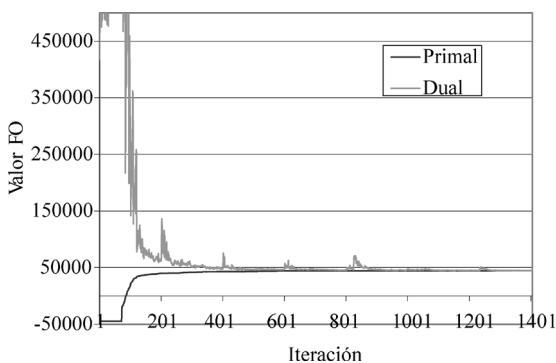


Figura 1 Iteración típica del algoritmo

Es importante resaltar algunos aspectos que pueden observarse en la figura 1.

- La solución primal del problema se aproxima rápidamente al valor óptimo, sin embargo es necesario un número considerablemente grande de iteraciones hasta lograr la optimalidad de dicha solución.
- Puede observarse que existe una gran variabilidad en el valor de la solución dual, esta variación se hace particularmente notoria en las primeras iteraciones del algoritmo.
- Puede observarse como se genera un salto notorio en el valor de la solución dual cada cierto número de iteraciones (200 en este caso). Este salto se presenta cada vez que el algoritmo aplica el procedimiento para eliminar columnas (para el ejemplo particular que se corrió, este procedimiento se ejecuta cada 200 iteraciones)

Con el fin de atenuar el problema de convergencia lenta y degeneración en la aplicación del algoritmo de generación de columnas, distintos autores han propuesto diferentes procedimientos, principalmente enfocados a la estabilización de los valores duales [11]. Una estrategia de simple aplicación, sugiere formular el problema de modo que las restricciones de igualdad puedan ser sustituidas por desigualdades, sin alterar el valor de la solución óptima. El hecho de sustituir las restricciones de igualdad por desigualdades restringe el valor de las variables duales a ser bien sea no negativas o no positivas en lugar de ser variables libres en signo, este hecho reduce la oscilación de dichas variables. En el problema particular, la formulación desagregada (*P2*) puede modificarse, sin alterar el valor de la solución óptima, cambiando el conjunto de restricciones de asignación de los estudiantes, (6), empleando desigualdades en lugar de igualdades. La figura 2 presenta el resultado de correr con cada una de las dos formulaciones, con restricciones de igualdad y con restricciones de desigualdad, la misma instancia de prueba empleada anteriormente; la parte de arriba de la figura corresponde a una vista en detalle de las primeras 400 iteraciones.

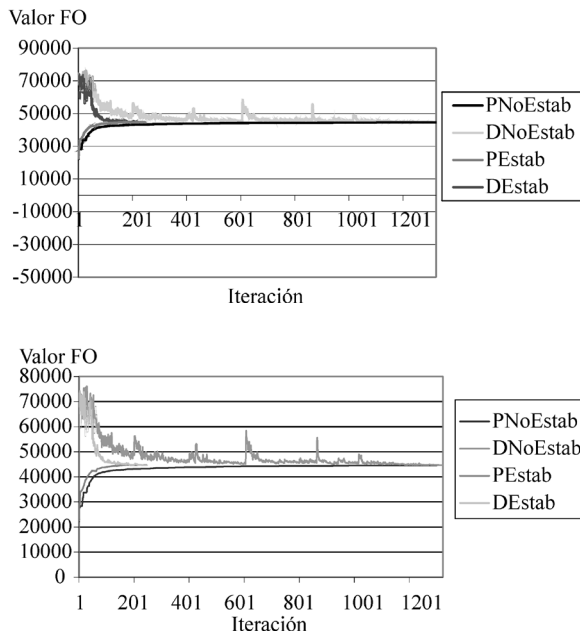


Figura 2 Aplicación de la estrategia de estabilización 1

En esta figura las líneas *PNoEstab* y *DNoEstab* corresponden a la solución primal y dual, respectivamente, en el problema con restricciones de igualdad; mientras que las líneas *PEstab* y *DEstab* corresponden a la solución primal y dual en el problema con restricciones de desigualdad. Se puede observar como el cambio en las restricciones de asignación de los estudiantes, contribuye a disminuir ostensiblemente la variabilidad de los valores duales, principalmente en las primeras iteraciones. Sin embargo, la solución de ambas formulaciones aun presenta una fuerte evidencia del efecto *tailing-off*, el cual ocasiona un gran número de iteraciones hasta alcanzar la solución óptima; esto sugiere la necesidad del uso de otras estrategias de estabilización para mejorar la convergencia del algoritmo.

Para este problema en particular se utilizó un método de estabilización y aceleración del algoritmo, cuya principal característica es permitir su implementación en el marco de la Programación Lineal. La estrategia empleada se describe detalladamente en [12] y [13]; a continuación se presentan los aspectos más relevantes al respecto:

Considere el siguiente programa lineal (P), factible y acotado, y su correspondiente dual (D), en donde (P) tiene un número considerable de variables y se hace uso del algoritmo de generación de columnas para su solución:

<p>PROBLEMA PRIMAL (P)</p> $\text{Max } \mathbf{c}^T \mathbf{x}$ <p>s.a. $\mathbf{Ax} = \mathbf{b}$</p> $\mathbf{x} \geq \mathbf{0}$	<p>PROBLEMA DUAL (D)</p> $\text{Min } \mathbf{b}^T \boldsymbol{\pi}$ <p>s.a. $\mathbf{A}^T \boldsymbol{\pi} \geq \mathbf{c}$</p>
---	---

Una forma de atenuar la degeneración en la solución es mediante la perturbación de (P), adicionando variables de holgura y de exceso acotadas. Alternativamente, podría usarse penalizaciones explícitas para las variables duales. La estrategia propuesta en [12] combina el método de perturbación y de penalizaciones explícitas, de la siguiente forma:

Se define el programa lineal (\tilde{P}) y su correspondiente dual (\tilde{D}):

<p>PROBLEMA PRIMAL (\tilde{P})</p> $\text{Max } \mathbf{c}^T \tilde{\mathbf{x}} - \boldsymbol{\delta}^T \mathbf{y}^- + \boldsymbol{\delta}^T \mathbf{y}^+$ <p>s.a. $\mathbf{A}\tilde{\mathbf{x}} - \mathbf{y}^- + \mathbf{y}^+ = \mathbf{b}$</p> $\mathbf{y}^- \leq \boldsymbol{\varepsilon}^-$ $\mathbf{y}^+ \leq \boldsymbol{\varepsilon}^+$ $\tilde{\mathbf{x}}, \mathbf{y}^-, \mathbf{y}^+ \geq \mathbf{0}$	<p>PROBLEMA DUAL (\tilde{D})</p> $\text{Min } \mathbf{b}^T \tilde{\boldsymbol{\pi}} + \boldsymbol{\varepsilon}^- \mathbf{v} + \boldsymbol{\varepsilon}^+ \mathbf{w}$ <p>s.a. $\mathbf{A}^T \tilde{\boldsymbol{\pi}} \geq \mathbf{c}$</p> $-\tilde{\boldsymbol{\pi}} + \mathbf{v} \geq -\boldsymbol{\delta}$ $\tilde{\boldsymbol{\pi}} + \mathbf{w} \geq \boldsymbol{\delta}$
---	--

En el problema primal (\tilde{P}), \mathbf{y}^- e \mathbf{y}^+ son vectores de variables de exceso y holgura, con cotas superiores $\boldsymbol{\varepsilon}^-$ y $\boldsymbol{\varepsilon}^+$, respectivamente. En el problema Dual (\tilde{D}), las dos últimas restricciones pueden reescribirse como $\boldsymbol{\delta} - \mathbf{w} \leq \boldsymbol{\pi} \leq \boldsymbol{\delta} + \mathbf{v}$ la cual cuantifica la penalización de las variables duales $\boldsymbol{\pi}$ [12]. Obsérvese que la solución de los dos problemas, P y \tilde{P} , son iguales cuando se satisface que $\boldsymbol{\varepsilon}^- = \boldsymbol{\varepsilon}^+ = \mathbf{0}$, lo cual motiva el algoritmo de solución.

El algoritmo empleado es esencialmente el mismo planteado anteriormente, sin embargo es necesario adicionar a dicho algoritmo dos aspectos relacionados con la actualización de los nuevos parámetros considerados ($\boldsymbol{\delta}$, $\boldsymbol{\varepsilon}^-$ y $\boldsymbol{\varepsilon}^+$):

- i.) *Actualización de $\boldsymbol{\delta}$.* Se ajusta el valor del parámetro a lo largo del algoritmo haciéndose

igual al vector de valores duales π que genere la mejor cota dual hasta ese momento encontrada.

ii.) *Actualización de ϵ^- y ϵ^+* . Cada cierto número de iteraciones el valor de estos parámetros debe disminuirse de modo que llegue a cero en un número finito de iteraciones, lo cual situaría el problema en su equivalente P , garantizando que el algoritmo tiene terminación en un número finito de iteraciones.

La figura 3 presenta el resultado de aplicar la segunda estrategia de estabilización sobre el problema en el cual ya se han modificado las restricciones de igualdad por desigualdades. En la figura, la imagen de la derecha representa un detalle de las primeras iteraciones, hasta alcanzar optimalidad en el algoritmo estabilizado. Es posible observar como el número de iteraciones requerido hasta encontrar la solución óptima se reduce considerablemente, pasando en este caso de más de 1.200 iteraciones a cerca de 250 iteraciones. Adicionalmente, puede verse como la variabilidad de la solución dual se reduce, principalmente a partir de cierto número de iteraciones iniciales. La generación de columnas irrelevantes, que aportan poca información para ajustar los valores duales, en las primeras iteraciones del algoritmo es conocida como el efecto “*heading-in*” [11].

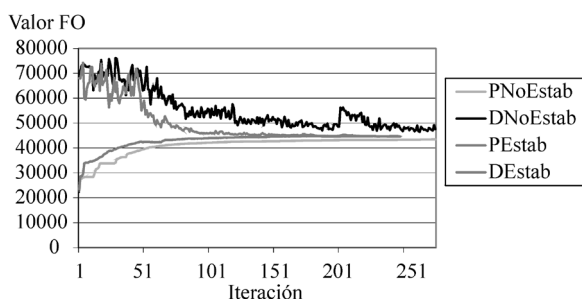


Figura 3 Aplicación de la estrategia de estabilización 1

Consideración de extra restricciones

El problema real podría considerar algunas otras restricciones para la asignación, estas restricciones podrían clasificarse en dos grupos:

- i.) Restricciones propias de los estudiantes, en particular respecto a los colegios a los cuales puede ser asignado un estudiante particular.
- ii.) Restricciones propias de los colegios, en relación con condiciones particulares respecto a los estudiantes que pueden serle asignados o la selección de las asignaciones factibles a dichos colegios.

Como se señaló anteriormente, el algoritmo de generación de columnas por si solo no basta para la obtención de soluciones enteras en casos en los cuales la relajación lineal de la formulación desagregada no corresponda a la solución entera. El Algoritmo de *Branch-and-Price*, el cual hace uso del algoritmo de generación de columnas y del algoritmo de *Branch-and-Bound*, podría constituir una alternativa interesante para la solución de este tipo de problemas; sin embargo, es importante señalar que este algoritmo debería hacer frente a los mismos problemas de convergencia identificados para el algoritmo de generación de columnas.

Algoritmo de Branch and Price

Debido a que la formulación desagregada, ($P2$), tiene un gran número de columnas y para la mayoría de éstas la variable asociada en la solución óptima toma el valor de cero, el algoritmo B-P considera una relajación del problema con sólo un subconjunto de éstas. Mediante la implementación del problema auxiliar (*Pricing Problem*) se generan las columnas que deben adicionarse a la base hasta dar solución a la relajación del problema. La ramificación (*Branching*) ocurre cuando no pueden hallarse columnas para adicionar a la base y la solución del problema relajado no satisface las condiciones de integralidad. En el caso en el cual la solución óptima del Problema Maestro Inicial es entera, ésta corresponde también a la solución óptima del problema no relajado, es decir, para el problema de la formulación desagregada. Esta solución también corresponde a la solución óptima de la formulación estándar ($P1$) del problema de asignación.

En el caso en el cual la solución de la relajación del problema maestro inicial no satisface las restricciones de integralidad, el problema debe

ramificarse en dos o más subproblemas, que entran a formar parte del conjunto de subproblemas activos, los cuales deben evaluarse con el objeto de construir cotas para el valor de la función objetivo del problema no relajado.

El algoritmo avanza iterativamente, seleccionando, con algún criterio definido, alguno de los subproblemas activos, ramificando este subproblema y hallando el valor óptimo (o una cota) de la relajación lineal de los problemas obtenidos al ramificar. Para la evaluación de los distintos subproblemas se hace uso del esquema de generación de columnas. Cada vez que se explora un nodo del árbol de búsqueda (ramificar y evaluar), alguno de los subproblemas activos, es posible encontrarse en uno de los siguientes dos casos;

- i.) Se encuentra una solución entera, lo cual implica que en adelante este nodo no debe considerarse. En caso de que la solución encontrada sea mejor que la mejor solución disponible hasta ese momento (incumbete) se actualiza el valor y se depura el conjunto de subproblemas activos eliminando aquellos que tienen peores cotas con respecto al incumbete.
- ii.) No se encuentra una solución entera. En este caso los subproblemas generados se adicionan al conjunto de subproblemas activos si las cotas para la función objetivo encontradas no son peores que la mejor solución conocida. En caso contrario el subproblema se descarta, podando así el árbol de búsqueda.

La figura 4 presenta de modo esquemático la forma como se desarrolla el algoritmo de *Branch-and-Price*.

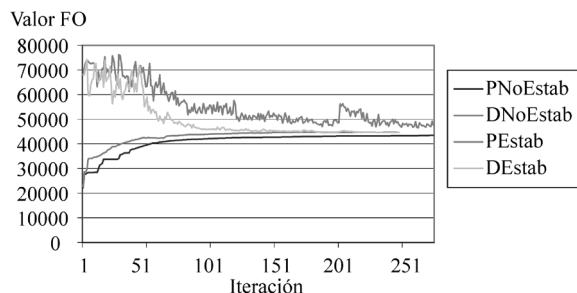


Figura 4 Algoritmo *Branch and Price*

Estrategias de Ramificación y Acotamiento

Aún cuando la solución del Problema Maestro Restringido sea tal que todas las y_k^i sean enteras, esto no garantiza que corresponda a la solución óptima del Problema Original (*P1*); el nodo correspondiente a dicha solución estará resuelto y solo se podrá verificar su optimalidad una vez se haya explorado y acotado apropiadamente el árbol de búsqueda. Esto hace necesario ramificar el problema y explorar los distintos subproblemas activos con el fin de acotar la solución del Problema Entero original. Debe seleccionarse una regla de ramificación que sea compatible con el problema auxiliar de la generación de columnas, es decir una regla de ramificación que permita modificar el *Pricing Problem* de modo que las columnas que no son factibles, debido a las restricciones impuestas por las distintas ramificaciones, no sean generadas y que la solución de dicho problema siga siendo eficiente. La idea básica es desarrollar la ramificación usando la formulación estándar (*P1*) mientras se procede en la solución del problema con la formulación desagregada (*P2*) [14].

Ante los problemas de convergencia señalados para el algoritmo de generación de columnas, una alternativa existente es terminar anticipadamente la ejecución del algoritmo, es decir, detener el proceso de generación de columnas cuando la brecha existente entre la cota inferior, dada por la solución actual del problema maestro, y la cota superior, construida por ejemplo, por *Relajación Lagrangiana*, alcanza un nivel determinado [15].

Una cota superior para el valor óptimo de la función objetivo del problema en (*P2*), puede construirse resolviendo la *Relajación Lagrangiana* resultante de dualizar las restricciones de asignación, la cual esta dada por: [16]

$$\text{Maximizar } \sum_{j \in J} \sum_{k \in K_j} \left(\sum_{i \in I} c_{ij} x_{ik}^j \right) v_k^j + \sum_{i \in I} \mu_i \left(1 - \sum_{j \in J} \sum_{k \in K_j} x_{ik}^j v_k^j \right) \quad (10)$$

Mediante operaciones matemáticas simples, puede mostrarse que el problema en (10) es equivalente a:

Puede notarse como una vez solucionados el problema auxiliar, *Pricing Problem*, en una iteración típica del algoritmo de generación de columnas, se dispone de toda la información necesaria para construir una cota superior respecto al valor de la solución óptima.

Consideración de extra restricciones

Una de las características más importantes de la formulación desagregada, (P2), es que en el problema auxiliar se incorporan las restricciones relacionadas con las asignaciones factibles de cada colegio; por este motivo, las restricciones adicionales, que afectan directamente el conjunto de posibles asignaciones de cada institución, solo afectan el problema auxiliar [6]. De esta forma, pueden considerarse para la solución del problema mediante el algoritmo de *Branch-and-Price*, entre otros, los siguientes tipos de restricciones.

- Considerar restricciones de equilibrio de género, de estrato socioeconómico, o de otros factores, en la asignación de estudiantes para cada institución.
- Considerar restricciones que determinen las asignaciones factibles para distintos conjuntos de colegios, es decir, definir las asignaciones factibles de formas diferentes para distintos grupos de colegios según sea necesario.

Adicionalmente, es importante notar que también es posible considerar restricciones adicionales respecto a las instituciones a las cuales puede ser asignado cada estudiante; para ello, debe determinarse una definición apropiada de la función de beneficio de la asignación de cada estudiante a cada colegio, que recoja las particularidades de la instancia que desea resolverse.

Conclusiones

El algoritmo de generación de columnas se presenta como una alternativa al abordar la solución de problemas de asignación de gran escala, como el descrito en este artículo. Su relevancia se extiende al considerar la su incorporación en el al-

goritmo de *Branch and Price*, lo cual permitiría además la solución de problemas enteros o mixtos de mayor complejidad.

En el desarrollo de este trabajo pudo evidenciarse la convergencia lenta del algoritmo de generación de columnas, comúnmente referida en la literatura. Así mismo, fue posible evidenciar el comportamiento oscilatorio y poco estable de las variables en el problema dual, que junto con la degeneración son señaladas de ser una de las causas del problema de convergencia de dicho algoritmo. Este hecho podría presentarse como una limitante en su uso en estrategias de solución como el algoritmo *Branch-and-Price*.

Fue posible observar alternativas para atenuar las problemáticas evidenciadas en el algoritmo de generación de columnas. Sin embargo, son necesarias consideraciones adicionales o el uso de estrategias diferentes, con el fin de acelerar y mejorar su desempeño, de modo que pueda pensarse en su uso en problemas de mayor escala. Así por ejemplo, es necesario considerar la forma de inicializar los métodos de estabilización con mejores estimaciones de los valores duales, de modo que las primeras iteraciones sean más productivas. Adicionalmente se planteó el uso de cotas respecto al valor de la función objetivo, que permitan terminar anticipadamente el algoritmo de generación de columnas. Esto con el fin de intentar construir buenas soluciones a partir de la solución encontrada o emplear dichas cotas para la exploración del árbol de búsqueda en el algoritmo *Branch-and-Price*.

El algoritmo de generación de columnas puede ser usado como estrategia de solución exacta o como parte de estrategias de solución aproximada, dependiendo de la complejidad del problema abordado. Dentro de los problemas dentro de este mismo marco susceptibles a ser abordados mediante este tipo de estrategias se destacan el problema de programación de cursos (*Timetabling problema*) y el problema de asignación de turnos (*Rostering Problem*)

Dentro del proceso de asignación de cupos escolares en la SED, es posible identificar situaciones

en las cuales el uso de modelos matemáticos puede contribuir en el mejoramiento del resultado del proceso. La forma en la que se desarrolla actualmente el proceso dista de pretender optimizar el beneficio obtenido en la asignación y se asemeja a los algoritmos *greedy* empleados para diversos problemas enteros (*Knapsack*, *Binpacking*, asignación).

Oportunidades de investigación

El trabajo desarrollado en torno al algoritmo de generación de columnas reveló algunos temas en los que se hace necesario profundizar o hacia los cuales sería importante dirigir futuros proyectos de investigación.

En relación con el algoritmo de generación de columnas, se presenta como un tema abierto el estudio de mejores estrategias de estabilización, así como la consideración de otras estrategias de reoptimización que hagan más eficiente su implementación. Además, la incorporación de este algoritmo dentro del algoritmo *Branch and Price*, dan importancia al estudio de cotas para la terminación temprana del algoritmo de generación de columnas, así como al estudio de la definición del problema auxiliar (*Pricing Problem*) en presencia de extra restricciones.

Por último, el trabajo desarrollado pretende despertar el interés en la aplicación modelos matemáticos como herramientas de apoyo a los distintos procesos que se desarrollan en el sector de la educación pública.

Referencias

1. P. Maya. *Estudio de Modelos de Apoyo al Proceso de Asignación de Cupos Escolares en el Sistema de Educación Pública del Distrito de Bogotá*. Tesis de Maestría. Universidad de los Andes. Facultad de Ingeniería. 2005. pp. 104
2. T. Magnanti, R. Orlin, J. B. Orlin. *Network Flows. Theory, Algorithms and Applications*. Ed. Prentice Hall. New York. 1993. pp 461-463.
3. J. Jarvis, H. Sherali, M. Bazaara. *Programación Lineal y Flujo en Redes*. Ed. Limusa. México. 1999. 2ª ed. pp. 553-594
4. F. Vanderbeck. *Descoposition and Column Generation for Integer Programs*. Tesis de Doctorado. Universidad Católica de Louvain. 1994, pp. 9-31.
5. M. Lubbecke. *Engine Scheduling by Column Generation*. Tesis de Doctorado, Universidad de Braunschweig. 2001, pp. 37-87.
6. W. Johnson, G. Nemhauser, M. Savelsbergh. "Progress in linear programming-based algorithms for integer programming: An exposition. *Journal on Computing*. Vol. 12. 2000. pp. 2-22.
7. M. Savelsbergh. "A branch-and-price algorithm for the generalized assignment problem". *Operations Research*. Vol. 45. 1997. pp. 831-841.
8. J. Desrosiers, M. Solomon, G. Desaulniers. (Eds). *Column Generation*. Ed. Springer. Boston. 2005. pp. 1-32.
9. L. Wolsey. *Integer Programming*. Ed. Jhon Wiley. 1998. pp 185-200.
10. M. Lubbecke, J. Desrosiers. "Selected topics in column generation". *Operations Research*. 2005. Vol. 53. 2005. pp. 1007-1023.
11. F. Vanderbeck. "Implementing Mixed Integer Column Generation". *Column Generation*. Ed. Springer. Boston. 2005. pp. 331-358.
12. J. Desrosiers, P. Hansen, O. Merle, D. Villeneuve. "Stabilized column generation". *Discrete Mathematics*. Vol. 194. 1999. pp. 229-237.
13. A. Pigatti, M. Aragão, E. Uchoa. "Stabilized Branch-and-cut-and-price for the Generalized Assignment Problem." *Electronic Notes in Discrete Mathematics*. Vol. 19. 2005. pp. 389-395.
14. M. Savelsbergh. "A branch-and-price algorithm for the generalized assignment problem". *Operations Research*. Vol. 45. 1997. pp. 831-842.
15. C. Barnhat, E. L. Johnson, G. L. Nemhauser, M. Savelsbergh, P. Vance. *Branch-and-price column generation for solving huge integer programs*. *Operations Research*. Vol. 46. 1998. pp. 316-329.
16. M. Savelsbergh. *Branch-and-price: Integer programming with column generation*. No publicado. 2002.