

Método de Punto Interior basado en movimientos ortogonales dentro de la región factible

Interior point method based on orthogonal movements within the feasible region

C. Fabian-Ruiz  ; J. Morales-Paredes  ; L. Eduardo-Ruiz 

DOI: <https://doi.org/10.22517/23447214.25167>

Artículo de investigación científica y tecnológica

Abstract— This work proposes a new interior point method to solve linear programming problems. Known interior point methods are difficult to understand since the tools used in them are too technical, the method proposed here is easy to understand and didactic when presented. Within the feasible region, an orthonormal basis of vectors can be found, which will serve as directions to move an arbitrary initial interior point, giving way to the generation of a search path (sequence) of points within the feasible region that converges to the optimal solution to the problem. If the path converges to a vertex of the feasible region, the solution to the problem has been found, while if the path leads to one side of the feasible region, it means that one of the variables is found to be zero and the algorithm is restarted with a decrease in the number of variables. The tools used here for the construction of the new method are basic concepts of linear algebra, this allows a pedagogical facility when it comes to understanding the algorithm.

Index Terms— Interior Point, Karmarkar's algorithm, Kernel, Linear Programming, Orthonormal Vectors, Optimization.

Resumen— Este trabajo propone un nuevo método de punto interior para resolver problemas de programación lineal. Los métodos conocidos de punto interior son de difícil comprensión dado que las herramientas usadas en ellos son demasiado técnicas, el método aquí propuesto es de fácil comprensión y didáctico a la hora de ser presentado. Al interior de la región factible se puede hallar una base ortonormal de vectores los cuales servirán como direcciones para trasladar un punto interior inicial arbitrario dando paso a la generación de una senda de búsqueda (sucesión) de puntos al interior de la región factible que converge a la solución óptima del problema. Si la senda converge a un vértice de la región factible se encontró la solución del problema, mientras que si la senda nos conduce a un lado de la región factible quiere decir que se encuentra que una de las variables es cero y se reinicia el algoritmo con una disminución en el número de variables. Las herramientas aquí utilizadas para la construcción del nuevo método son conceptos básicos de álgebra lineal, esto permite una facilidad pedagógica a la hora de entender el algoritmo.

Este manuscrito fue enviado el 19 de junio de 2021 y aceptado el 14 de julio de 2023

C. Fabian-Ruiz: Candidato a Doctor en economía de la Universidad del Rosario, Bogotá, Colombia. Correo electrónico: carlosf.ruiz@urosario.edu.co.

Palabras claves— Algoritmo de Karmarkar. Optimización, Punto interior, Programación Lineal, Núcleo, Vectores Ortonormales.

I. INTRODUCCIÓN

LA optimización matemática es una herramienta fundamental en una amplia variedad de campos. Por ejemplo, la optimización se utiliza para maximizar el beneficio de una empresa que produce un bien teniendo en cuenta sus ingresos, costos y los impuestos por unidad. La función de beneficio (1), la cual modela $B(q)$ como la diferencia entre los ingresos $I(q)$ y los costos $C(q)$, menos el impuesto t por unidad, es decir

$$B(q) = I(q) - C(q) - tq \quad (1)$$

El objetivo es encontrar la cantidad óptima q^* que maximice esta función de beneficio. Para hacerlo, se aplica la condición de beneficio máximo, que establece que el beneficio se maximiza cuando la tasa de cambio de los ingresos con respecto a la cantidad producida es igual a la tasa de cambio de los costos con respecto a la cantidad producida más el impuesto por unidad, es decir,

$$I'(q^*) = C'(q^*) + t \quad (2)$$

Esta condición (2) es crucial en economía y gestión empresarial, ya que permite determinar la cantidad óptima que la empresa debe producir para maximizar sus ganancias, teniendo en cuenta los costos y los impuestos. Si se satisface esta condición, significa que la empresa está produciendo la cantidad correcta para maximizar su beneficio.

J. Morales-Paredes: Docente de la Escuela Superior de Administración pública ESAP, Colombia. Correo electrónico: jorge.moralesp@esap.edu.co

L. Eduardo-Ruiz: Candidato a Doctor en Gestión de la Universidad EAN. Correo electrónico: lruizpar362@universidadean.edu.co



La optimización matemática se utiliza ampliamente en análisis económicos, planificación financiera y toma de decisiones en la industria para encontrar soluciones óptimas en situaciones donde se deben considerar múltiples variables y restricciones. En este caso, la optimización ayuda a comprender el comportamiento económico de la empresa y a tomar decisiones informadas sobre la producción y los precios de los bienes.

La programación lineal es un campo fundamental dentro de la optimización matemática que se utiliza para resolver problemas de optimización en los cuales se busca maximizar o minimizar una función lineal (llamada función objetivo) sujeta a un conjunto de restricciones lineales. Estos problemas se pueden representar de la siguiente manera

$$\begin{array}{ll} \text{Max} & Z = \mathbf{c}^t \mathbf{x} \\ \text{Sujeto a:} & \end{array} \quad (3)$$

$$\begin{array}{l} \mathbf{Ax} \leq \mathbf{b} \\ \mathbf{x} \geq 0 \end{array}$$

donde en (3) $\mathbf{c} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}$ son los coeficientes de la función

objetivo, $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$ son las variables de decisión que queremos

encontrar, $\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$ son los coeficientes de

las restricciones lineales, $\mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$ son los límites superiores

de las restricciones, m es el número de restricciones.

El objetivo en la programación lineal es encontrar los valores de las variables x_1, x_2, \dots, x_n que maximizan o minimizan la función objetivo sujeta a las restricciones dadas. Estos problemas tienen una amplia variedad de aplicaciones en la industria, la economía, la logística y otras áreas.

El Método Simplex, desarrollado por George Dantzig en la década de 1940, es uno de los algoritmos más conocidos y ampliamente utilizados para resolver problemas de programación lineal. Este método se basa en un proceso iterativo que mueve sistemáticamente de una solución a otra, mejorando gradualmente el valor de la función objetivo hasta encontrar la solución óptima.

Es importante destacar que, la programación lineal a menudo se utiliza para modelar situaciones del mundo real. Los problemas de programación lineal pueden representar de manera efectiva una amplia gama de problemas de toma de decisiones, desde la asignación de recursos limitados hasta la optimización de la producción y la distribución. La capacidad de resolver eficientemente estos problemas tiene un impacto significativo en la planificación y la toma de decisiones en

muchas industrias y campos de estudio.

En la práctica, la solución de problemas de programación lineal puede estar sujeta a diversas fuentes de incertidumbre y errores, como la precisión de los datos y el error de redondeo en las computadoras. Esto plantea preguntas importantes sobre la robustez y la confiabilidad de las soluciones obtenidas.

Aquí hay algunas consideraciones clave relacionadas con las soluciones de problemas de programación lineal:

Existencia de solución: Una de las primeras preguntas que surgen al abordar un problema de programación lineal es si el modelo propuesto tiene una solución factible, es decir, una solución que cumple con todas las restricciones. En algunos casos, el problema puede no tener una solución factible, lo que indica que las restricciones son inconsistentes o que las limitaciones impuestas son demasiado restrictivas.

Unicidad de la solución: En general, los problemas de programación lineal pueden tener múltiples soluciones factibles que optimizan la función objetivo al mismo valor. Sin embargo, en ciertas condiciones, como cuando las restricciones son linealmente independientes, la solución puede ser única. La unicidad de la solución depende de las características específicas del problema.

Estabilidad de la solución: La estabilidad de la solución se refiere a cómo pequeños cambios en los parámetros del modelo afectan la solución óptima. Un modelo de programación lineal se considera "estable" o "dependiente continuamente de los parámetros" si cambios pequeños en los coeficientes de las restricciones o en los coeficientes de la función objetivo generan cambios pequeños en la solución óptima. Esta propiedad es esencial en la aplicabilidad del modelo a situaciones de la vida real, donde los datos pueden variar.

Análisis de sensibilidad: El análisis de sensibilidad es una técnica importante que permite evaluar cómo cambian las soluciones óptimas ante cambios en los parámetros del modelo. La teoría de la dualidad es una herramienta valiosa en este contexto, ya que proporciona una forma sistemática de analizar cómo los cambios en los coeficientes de las restricciones afectan los valores óptimos de las variables duales y, por lo tanto, los precios sombra asociados a las restricciones.

En resumen, la programación lineal es una herramienta poderosa para abordar problemas de toma de decisiones en una variedad de campos, pero es importante tener en cuenta la incertidumbre y los errores en la práctica. El análisis de sensibilidad y la teoría de la dualidad son herramientas fundamentales para evaluar y comprender la estabilidad y la confiabilidad de las soluciones obtenidas.

Gracias a los trabajos de Victor Klee y George Minty [9] se mostró que el algoritmo Simplex no posee un tiempo polinomial de ejecución, de hecho, Jeroslow mostró en [6] que

existe problemas de programación lineal que su tiempo de ejecución es polinomial, por lo tanto, en problemas lineales a gran escala este método puede no ser eficiente, sin embargo, los métodos de punto interior probaron tener un tiempo de ejecución polinomial y por ende ser más efectivos en problemas lineales a gran escala, el primero en probar dicha afirmación fue Khachiyan en [8] con su algoritmo del elipsoide, desafortunadamente las experiencias prácticas de dicho algoritmo fueron decepcionantes.

El método de punto interior más conocido, mostrado en el famoso trabajo de Karmarkar [7], llamado algoritmo de Karmarkar, posee un tiempo de ejecución polinomial y se basa en movimientos sobre el interior de la región factible por medio del gradiente proyectado, al igual que el método simplex, el método del punto interior presenta ventajas y desventajas a la hora de encontrar la solución al problema tratado, este argumento será esencial para el desarrollo de la investigación.

Los algoritmos de punto interior juegan un papel importante en la solución de problemas de programación lineal, su implementación y uso radican principalmente en problemas con un gran número de variables. Garcés, Toro y Galvis [12] muestran una aplicación del método de punto interior clásico al problema de transporte y realizan una propuesta empírica basado en sus resultados, por otra parte, Carmona, Gallego y Garcés [13] presentan una versión alterna del método de punto interior primal-dual con el fin de buscar soluciones a problemas que tengan múltiples soluciones. Carreño, Toro y Escobar [14] discuten distintos métodos de punto interior y mencionan sus ventajas sobre el método tradicional Simplex.

El trabajo aquí presentado desarrolla un algoritmo de punto interior para la solución de problemas de programación lineal basado en conceptos básicos de álgebra lineal y programación lineal.

El desarrollo teórico del algoritmo aquí propuesto se basa en la búsqueda de direcciones dadas por el núcleo de una matriz correspondiente y en movimientos inspirados en el método de punto interior de Karmarkar, con lo cual, este algoritmo pretende buscar la mejor dirección de optimización de las dentro de la región factible del problema, partiendo de un punto inicial arbitrario (perteneciente al interior de la región factible), con el fin de encontrar la solución óptima.

Inicialmente se plantea un algoritmo basado en direcciones establecidas por los vértices de la región factible, sin embargo, conocer todos vértices en un problema muy amplio (cuando el número de variables y restricciones es grande) requiere de un manejo matemático ineficiente en términos de tiempo computacional, por ende, fue necesario encontrar un mecanismo que permitiera definir direcciones de movimiento alterna a la propuesta inicial. El comando “null” en Matlab permite encontrar una base del núcleo de una matriz (cuyos vectores son ortonormales) y es desde aquí el sustento para la propuesta final del método de punto interior aquí presentado.

Precisamos de manera muy somera alguna notación estandar que será necesaria para la realización de este trabajo.

Como ya se mencionó la ecuación (3) es la forma matricial del problema de programación lineal.

Definición 1: Una solución factible es aquella que verifica todas las restricciones de un PPL, es decir, un vector $x \geq 0$ tal que $Ax \leq b$

Teorema 1 La región factible para cualquier PPL es un poliedro convexo. Además, si la región factible es cerrada y acotada el PPL tiene por lo menos una solución, el óptimo debe ser un punto vértice de la región factible.

Definición 2 Las variables de holgura son variables que se agregan a la restricción para que la relación de dicha restricción sea de igualdad, representa el valor faltante del lado izquierdo de la restricción para que sea igual al lado derecho.

Definición 3 La forma aumentada dada en la ecuación (4), es decir con las respectivas variables de holgura, del modelo de programación lineal será

$$\begin{aligned} \text{Max} \quad & Z = c^t x & (4) \\ \text{Sujeto a:} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

II. ALGORITMO PROPUESTO

Los algoritmos de punto interior son una clase de métodos utilizados para resolver problemas de programación lineal y programación lineal entera. A diferencia del método simplex, que se mueve de vértice en vértice en la región factible, los algoritmos de punto interior operan en el interior de la región factible y siguen una serie de pasos iterativos. Aquí se describen con más detalle los pasos básicos del algoritmo de punto interior basado en el enfoque original de Karmarkar:

Selección del punto inicial $p^{(0)}$: El algoritmo comienza eligiendo un punto de prueba inicial en la región factible del problema de programación lineal. Este punto de inicio puede seleccionarse de varias maneras, pero debe estar dentro de la región factible.

Transformación lineal: Se realiza una transformación lineal de la región factible de modo que el punto de prueba actual $p^{(i)}$ esté alejado de la frontera de la región factible. Esto es fundamental para garantizar que el algoritmo se mantenga dentro del interior de la región factible.

Selección de dirección de movimiento: En este paso, se elige una dirección en la que moverse desde el punto de prueba actual $p^{(i)}$ hacia otro punto de prueba dentro de la región factible que mejore el valor de la función objetivo Z . Esta dirección se elige de manera que el algoritmo se acerque a la solución óptima.

Condición de parada: El algoritmo verifica ciertas condiciones de parada para determinar si debe detenerse. Estas condiciones pueden incluir la convergencia a una solución óptima dentro de cierta tolerancia o alcanzar un número máximo de iteraciones permitidas.

Iteración o finalización: Si se cumple la condición de parada, el algoritmo se detiene, y se informa la solución óptima o aproximada. Si no se cumple la condición de parada, se vuelve al paso 2 y se repiten los pasos anteriores para buscar una solución mejorada.

El algoritmo de punto interior se repite iterativamente, moviéndose a través del interior de la región factible en busca de la solución óptima del problema de programación lineal. A medida que avanza, se acerca cada vez más a la solución óptima hasta que se alcanza la condición de parada.

Es importante destacar que existen diversas variantes y mejoras de los algoritmos de punto interior, y los detalles específicos pueden variar según la implementación y el enfoque utilizado. Estos algoritmos son altamente eficientes para resolver problemas de programación lineal de gran escala y son una alternativa importante al método simplex.

Para iniciar nuestro trabajo consideremos el problema de programación lineal en su forma aumentada (4). Denotaremos por \mathcal{R} la región factible descrita en (5) y que es representada en la Fig. 1 .

$$\mathcal{R} = \{ \mathbf{x} = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n \mid \mathbf{Ax} = \mathbf{b}, x_i \geq 0 \} \quad (5)$$

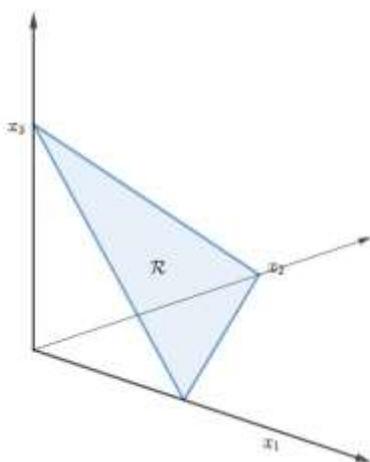


Fig. 1

Para un punto interior arbitrario de \mathcal{R} , es decir, sea $\mathbf{p} \in \text{Int}(\mathcal{R})$ se define el conjunto $\mathcal{R}_{\mathbf{p}}$ en (6) como todos los vectores que tienen como punto inicial \mathbf{p} y punto final un punto de la región factible, es decir,

$$\mathcal{R}_{\mathbf{p}} = \{ \mathbf{r} - \mathbf{p} \in \mathbb{R}^n \mid \mathbf{r} \in \mathcal{R} \} \quad (6)$$

el cual es una traslación de la región factible \mathcal{R} de tal manera que la nueva región trasladada $\mathcal{R}_{\mathbf{p}}$ pase por el origen (en el caso $n = 3$ se muestra en la Fig. 2.)

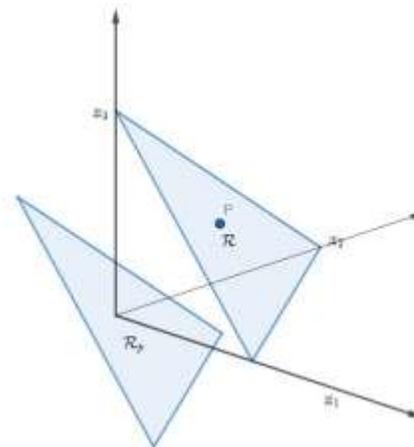


Fig. 2

Como una observación importante para nuestros propósitos es que si $\mathbf{x} \in \mathcal{R}_{\mathbf{p}}$ entonces $\mathbf{Ax} = \mathbf{0}$, esto se debe a que si $\mathbf{x} \in \mathcal{R}_{\mathbf{p}}$ existe un $\mathbf{r} \in \mathcal{R}$ talque $\mathbf{x} = \mathbf{r} - \mathbf{p}$ con lo cual

$$\mathbf{Ax} = \mathbf{A}(\mathbf{r} - \mathbf{p}) = \mathbf{Ar} - \mathbf{Ap} = \mathbf{b} - \mathbf{b} = \mathbf{0} \quad (7)$$

Aquí (7) muestra que $\mathcal{R}_{\mathbf{p}} \subset N_A$ donde N_A es el nucleo de la matriz A que se ve graficamente en la Fig.3.

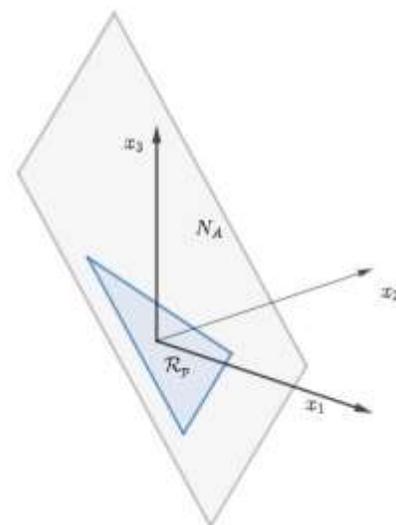


Fig. 3

En particular, para cada vector $\vec{u} \in \mathcal{R}$ con punto inicial $\mathbf{p} \in \mathcal{R}$, es decir, $\vec{u} = \mathbf{r} - \mathbf{p}$ para algún $\mathbf{r} \in \mathcal{R}$, además \vec{u} se puede escribir como $\vec{u} = (\mathbf{r} - \mathbf{p}) - \mathbf{0}$ y así $\vec{u} \in \mathcal{R}_{\mathbf{p}}$ con punto inicial el origen y como ya se menciona $\mathcal{R}_{\mathbf{p}} \subset N_A$, entonces $\vec{u} \in N_A$

Sea $\mathcal{B} = \{\beta_1, \dots, \beta_l\}$ ortonormal de N_A y denotemos por l la nulidad de A , es decir, $l = \nu(A)$.

Ya con esta notación dada lo que haremos es construir a partir de un punto arbitrario \mathbf{p}^0 en el interior de la región factible, es decir, $\mathbf{p}^0 \in \text{Int}(\mathcal{R})$ un punto $\mathbf{p}^1 \in \text{Int}(\mathcal{R})$ el cual sea una mejor aproximación de la solución que el punto \mathbf{p}^0 , en el sentido dado en (8)

$$\mathbf{c}^t \mathbf{p}^0 \leq \mathbf{c}^t \mathbf{p}^1 \tag{8}$$

Para esto formamos un conjunto de $2l$ valores posibles para \mathbf{p}^1 dados por

$$p^{1,i,\pm} = \pm \alpha \cdot S \cdot \beta_i \tag{9}$$

Para $i = 1, \dots, l$

Donde $0 < \alpha < 1$ es un suavizador y S es el valor mínimo de las componentes de \mathbf{p}^0 , lo anterior se ilustra en las Fig. 4 y Fig. 5

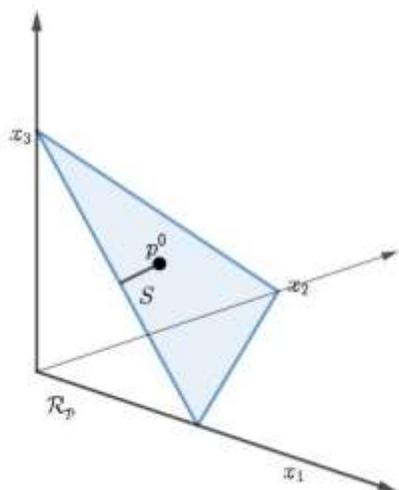


Fig. 4

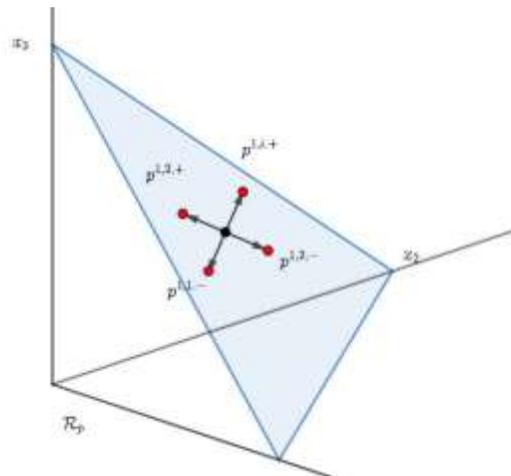


Fig. 5

De los puntos en (9) escogemos $\mathbf{p}^1 = p^{1,k,sg}$ (10)

Con $k = 1, \dots, l$ y $sg = +, -$ de tal manera que

$$\mathbf{c}^t p^{1,i,\pm} \leq \mathbf{c}^t p^{1,k,sg} \tag{11}$$

Con la elección de α y S nos aseguramos que $\mathbf{p}^1 \in \text{int}(\mathcal{R})$.

De esta forma, las ecuaciones (10) y (11) se crea una sucesión de puntos $\mathbf{p}^0, \mathbf{p}^1, \dots, \mathbf{p}^k$ la cual se va aproximando cada vez mas a la solución del problema.

La sucesión se va a detener en el momento en el cual una de las componentes de \mathbf{p}^k sea cero, en este caso el valor de dicha componente es la correspondiente a la variable x_j , hallo el valor de esa variable y reinicia el procedimiento en donde el numero de variables disminuye y así hasta que encuenter todas las variables. Así se considera el problema

$$\begin{aligned} \text{Max} \quad & Z = \mathbf{c}_{-j}^t \mathbf{x}_{-j} \tag{12} \\ \text{Sujeto a:} \quad & \mathbf{A}_{-j} \mathbf{x}_{-j} = \mathbf{b}_{-j} \\ & \mathbf{x}_{-j} \geq 0 \end{aligned}$$

Donde $\mathbf{c}_{-j}, \mathbf{x}_{-j}, \mathbf{A}_{-j}, \mathbf{b}_{-j}$ son los mismos $\mathbf{c}, \mathbf{x}, \mathbf{A}, \mathbf{b}$ pero omitiendo la componente j -ésima, es decir, reducimos la dimensión de nuestro problema a tamaño $n - 1$. Con lo anterior ya se tiene que en el problema original, el valor $x_j = 0$.

Esta reducción de dimensión se realiza hasta que el nucleo de la matriz asociada al problema de baja dimensión sea cero, es decir, si $N(\mathbf{A}_{-j}) = 0$ tenemos la solución de las variables faltantes.

El algoritmo anterior desarrollado en Matlab tiene el siguiente código

```
while l>0
```

```

B=null(A);
[n,l]=size(B);
if l==0
    break;
end
Dir=B;
for i=1:l
    Dir(:,l+i)=-B(:,i);
end
while min(p)>Tol
S=min(p);
for i=1:2*l
Mp(:,i)=p;
end
Pos=Mp+alpha*S*Dir;
Vz=(Pos'*c);
if condz==1
[zm,k]=max(Vz);
else
    [zm,k]=min(Vz);
end
p=Pos(:,k);
end
[minp,varelim]=min(p);
p(varelim,:)=[]
c(varelim,:)=[];
A(:,varelim)=[];
B=null(A,'r');
[n,l]=size(B);
end

```

mostraremos los resultados para el ejemplo dado en (13), considere el siguiente problema de PL

$$\text{Max} \quad \mathbf{Z} = 3x_1 + 5x_2 + 6x_3 \quad (13)$$

Sujeto a:

$$\begin{aligned} 2x_1 + x_2 + x_3 + x_4 &= 4 \\ x_1 + 2x_2 + x_3 + x_5 &= 4 \\ x_1 + x_2 + 2x_3 + x_6 &= 4 \\ x_1 + x_2 + x_3 + x_7 &= 3 \\ x_i &\geq 0, i = 1, 2, \dots, 7. \end{aligned}$$

Tomando como punto inicial $\mathbf{p}^0 = (0.5, 0.5, 0.5, 2, 2, 2, 1.5)$ tenemos que de 75 iteraciones se llega al punto $\mathbf{p}^{75} = (0, 0.2773, 1.5440, 2.1786, 1.9013, 0.6346, 1.1786)$

en el cual la primera variable es cero, por lo cual se encuentra el valor de la variable $x_1 = 0$ y se repite el algoritmo para el PPL con una dimensión menor como se muestra en (14)

$$\text{Max} \quad \mathbf{Z} = 5x_2 + 6x_3 \quad (14)$$

Sujeto a:

$$\begin{aligned} x_2 + x_3 + x_4 &= 4 \\ 2x_2 + x_3 + x_5 &= 4 \\ x_2 + 2x_3 + x_6 &= 4 \\ x_2 + x_3 + x_7 &= 3 \\ x_i &\geq 0, i = 2, 3, \dots, 7 \end{aligned}$$

con punto inicial $\mathbf{p}^0 = (0, 0.2773, 1.5440, 2.1786, 1.9013, 0.6346, 1.1786)$

repitiendo el algoritmo llegamos a $\mathbf{p}^{63} = (0.9775, 1.5112, 1.5112, 0.5337, 0, 0.5112)$

en el cual la quinta variable es cero, por lo cual se encuentra el valor de la variable $x_5 = 0$ y se repite el algoritmo para el PPL con una dimensión menor como se muestra en (15)

$$\text{Max} \quad \mathbf{Z} = 5x_2 + 6x_3 \quad (15)$$

Sujeto a:

$$\begin{aligned} x_2 + x_3 + x_4 &= 4 \\ 2x_2 + x_3 + x_5 &= 4 \\ x_2 + 2x_3 &= 4 \\ x_2 + x_3 + x_7 &= 3 \\ x_i &\geq 0, i = 2, 3, 4, 5, 7. \end{aligned}$$

Con $\mathbf{p}^0 = (0.9775, 1.5112, 1.5112, 0.5337, 0.5112)$

Y llegamos a $\mathbf{p}^{23} = 1.3333 \ 1.3333 \ 1.3333 \ 0.0000 \ 0.3333$

en el cual la cuarta variable es cero, por lo cual se encuentra el valor de la variable $x_4 = 0$ y se repite el algoritmo para el PPL con una dimensión menor como se muestra en(16)

$$\text{Max} \quad \mathbf{Z} = 5x_2 + 6x_3 \quad (16)$$

Sujeto a:

$$\begin{aligned} x_2 + x_3 + x_4 &= 4 \\ 2x_2 + x_3 &= 4 \\ x_2 + 2x_3 &= 4 \\ x_2 + x_3 + x_7 &= 3 \\ x_2 &\geq 0, i = 2, 3, 4, 7 \end{aligned}$$

Sin embargo la matriz asociada a las restricciones tiene nulidad ero por lo cual ya termina el algoritmo, con lo cual $x_1 = 0, x_2 = 1.333, x_3 = 1.333, x_4 = 1.333, x_5 = 0, x_6 = 0, x_7 = 0.333$

La gran ventaja que encontramos aquí es que el problema va solucionando las variables nulas y va reduciendo su dimensión lo cual hace que el calculo de la maquina sea mas sencillo.

III. CONCLUSIONES

Los conceptos matemáticos utilizados para generar este nuevo método de punto interior requieren elementos básicos en álgebra y programación lineales, esto ofrece una ventaja en

términos de facilidad pedagógica y brinda una herramienta con la cual se puede introducir a los métodos de punto interior.

El método basado en direcciones encuentra la solución óptima al problema, no obstante, se reconoce los problemas asociados a la elección del punto de partida del algoritmo o punto inicial, ya que no se garantiza la convergencia del método a la solución óptima, pues si bien es cierto que dicho punto tiene que estar dentro de la región factible, de la elección de este dependerá que el algoritmo busque la convergencia a la solución del problema.

Con lo anterior, la elección del punto inicial es arbitrario, en este método justifica una investigación posterior con el fin de atacar dicho problema, sin embargo, la literatura ofrece mecanismos en los cuales esta elección se define buscando un punto equidistante a las fronteras de la región factible o simplemente un valor arbitrario cercano a la solución del problema.

Mediante los mismos principios utilizados en esta propuesta para la creación de este nuevo método, la posibilidad de generar algoritmos alternos de punto interior radica en establecer más direcciones de las aquí planteadas, esto con el fin de mejorar (si es posible) la convergencia a las soluciones de los problemas estándar de programación lineal.

Para finalizar, el trabajo aquí presentado es teórico y no experimental, esta razón no se incluyen aplicaciones reales, sin embargo, se muestra un ejemplo ilustrativo de siete variables presentado en la sección anterior, lo podemos comparar con el método SIMPLEX y vemos que tiene un tiempo de ejecución menor usando el método aquí presentado en comparación al método SIMPLEX, los tiempos determinados mediante los métodos mencionados se muestran en la Tabla I y en la Tabla II.

TABLA I
TIEMPO ALGORITMO PROPUESTO

Function Name	Calls	Total Time
Tiempo algoritmo propuesto	1	0,051 s

TABLA II
TIEMPO SIMPLEX

Function Name	Calls	Total Time
Simplex	1	0.639 s
linprog	1	0.519 s

Como ya fue mencionado, se observa una mayor eficiencia en tiempos computacionales a la hora de buscar la solución a un PPL, los tiempos ejecutados se definieron mediante el

comando "linprog" del Software MATLAB, a continuación, se presenta el comando utilizado

```
A=[2,1,1,1,0,0,0;1,2,1,0,1,0,0;1,1,2,0,0,1,0;1,1,1,0,0,0,1];
B=[4;4;4;3]; C=[3;5;6;0;0;0;0]; FORMAT SHORT; OPTIONS =
OPTIMSET('LARGE_SCALE','OFF','SIMPLEX','ON');
[X,FVAL,EXITFLAG,OUTPUT]=LINPROG(C,[],[],A,B,ZEROS(
SIZE(C)),[],[],OPTIONS) X.
```

REFERENCIAS

- [1] T. H. Cormen, C. E. Leiserson, R. L. Rivest y C. Stein, Introduction to Algorithms, Second Edition, Cambridge, Massachusetts London, England: McGraw-Hill Book Company, 2001.
- [2] M. El Ghami, Z. A. Guennoun, S. Bouali y T. Steihaug, «Interior-point methods for linear optimization based on a kernel function with a trigonometric barrier term,» *Journal of Computational and Applied Mathematics*, pp. 3613-3623, 2012. DOI 10.1016/j.cam.2011.05.036
- [3] F. A. Potra y S. J. Wright, «Interior-point methods,» *Journal of Computational and Applied Mathematics*, pp. 281-302, 2000.
- [4] J. Gondzio, «Interior point methods 25 years later,» *European Journal of Operational Research*, pp. 587-601, 2012. DOI 10.1016/j.ejor.2011.09.017
- [5] F. S. Hillier y G. J. Lieberman, Introducción a la investigación de operaciones, McGraw-Hill Interamericana de España, 2010.
- [6] R. G. Jeroslow, «The simplex algorithm with the pivot rule of maximizing criterion improvement,» *Discrete Mathematics*, pp. 367-377, 1973. DOI 10.1016/0012-365X(73)90171-4
- [7] N. Karmarkar, «A new polynomial-time algorithm for linear programming,» *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pp. 302-311, 1984.
- [8] L. G. Khachiyan, «Polynomial algorithms in linear programming,» *USSR Computational Mathematics and Mathematical Physics*, pp. 53-72, 1980. DOI 10.1016/0041-5553(80)90061-0
- [9] V. Klee y G. J. Minty, «How good is the simplex algorithm,» Technical rept., 1970.
- [10] E. Kreyszig, Introductory functional analysis with applications, New York, 1978.
- [11] A. L. Ramirez Leal, O. Y. Buitrago Suescún, R. A. Britto Agudelo y A. Fedosova, «A new algorithm for solving linear programming problems,» *Ingeniería e Investigación*, pp. 68-73, 2012. DOI 10.15446/ing.investig.v32n2.31949
- [12] A. Garces Ruiz, E. M. Toro y J. C. Galvis, «MÉTODO DE PUNTOS INTERIORES APLICADO AL PROBLEMA DE TRANSPORTES,» *Scientia Et Technica*, 2005. DOI 10.22517/23447214.6877
- [13] O. Gomez Carmona, L. A. Gallego y L. P. Garces, «METODO DE PUNTOS INTERIORES PARA PROGRAMACIÓN LINEAL APLICADO A PROBLEMAS CON ÓPTIMOS ALTERNATIVOS,» *Scientia Et Technica*, vol. 1, n° 27, 2005. DOI 10.22517/23447214.6873
- [14] E. Carreño, E. Toro Ocampo y A. Escobar, «Optimización de sistemas lineales usando métodos de punto interior,» *Scientia Et Technica*, vol. 1, n° 24, 2004. DOI 10.22517/23447214.7283
- [15] K. Sydaseter y J. Hammond, Matemáticas para el análisis económico, Pearson Educación, 1996.

Carlos Fabian Ruiz: Economista, Universidad Militar Nueva Granada, Bogotá, Colombia. Magíster en Economía, Universidad Externado de Colombia, Bogotá, Colombia. Magister en investigación de operaciones y estadística, Bogotá, Universidad Tecnológica de Pereira, Colombia. Candidato a Doctor en economía de la Universidad del Rosario, Bogotá, Colombia.
ORCID: <https://orcid.org/0000-0003-1507-1779>

Jorge Morales Paredes: Matemático, Universidad Nacional de Colombia, Bogotá, Colombia. Maestría en Matemáticas, Universidad Nacional de Colombia, Bogotá, Colombia. Doctor en Matemáticas, Universidad Nacional de Colombia, Bogotá, Colombia. Docente tiempo completo Escuela Superior de Administración Pública ESAP, Colombia.
ORCID: <https://orcid.org/0000-0002-0394-7756>

Luis Eduardo Ruiz: Profesional en Finanzas y Comercio Exterior, Fundación Universitaria Empresarial de la Cámara de Comercio de Bogotá, Bogotá, Colombia. Magister en Gerencia de Proyectos, EAN, Bogotá, Colombia. Candidato a Doctor en Gestión de la Universidad EAN.
ORCID: <https://orcid.org/0009-0005-8595-2653>