

Método de Punto Proximal para Desigualdad Variacional Aplicada a un Problema de Equilibrio de Redes.

Elvia Pérez Barturén¹, Rosa Medina Aguilar² y Erik Papa Quiroz³

Resumen: En este artículo presentamos un método proximal con distancias proximales para encontrar la ruta óptima de una red vehicular modelado como un problema de desigualdad variacional. Realizamos la implementación computacional y la experimentación numérica en Python utilizando el método de Newton y una regla de viabilidad para resolver los subproblemas que se transforman en sistemas de ecuaciones no lineales con condiciones de no negatividad sobre las variables. La principal contribución del artículo es el abordaje numérico utilizado para resolver los subproblemas del método proximal y que son verificados a través de ejemplos prácticos. Este abordaje numérico sigue la metodología existente para resolver los subproblemas en métodos proximales pero introduce criterios novedosos para obtener el punto siguiente en cada iteración del algoritmo proximal. Debemos resaltar que este trabajo es la primera implementación computacional del método proximal con distancias proximales aplicada en equilibrio de redes.

Palabras clave: método de punto proximal, equilibrio de redes, distancias proximales, variational inequalities.

Proximal Point Method for Variational Inequality Applied to a Routing Problem

Abstract: In this paper we present a proximal method with proximal distances for finding the optimal route of a vehicular network modeled as a variational inequality problem. We perform the computational implementation and numerical experimentation in Python using Newton's method and a viability ruler to solve the subproblems that become systems of nonlinear equations with nonnegative constraints on the variables. The main contribution of the article is the numerical approach used to solve the subproblems of the proximal method and which are verified through practical examples.

Keywords: proximal point method, network equilibrium problems, proximal distances, variational inequalities.

Recibido: 16/06/2024 *Aceptado:* 30/10/2024 *Publicado online:* 30/12/2024

¹UNMSM, Facultad de Ciencias Matemáticas. eperezb@unmsm.edu.pe

²UNMSM, Facultad de Ciencias Matemáticas. rmedinaa1@unmsm.edu.pe

³UNMSM, Facultad de Ciencias Matemáticas, epapaq@unmsm.edu.pe

1 Introducción

La congestión de las redes de transporte urbano representan sistemas complejos, con la cual los usuarios interactúan a menudo para determinar las rutas de costo mínimo de viaje entre su punto de origen y destino. Diariamente cada usuario debe tomar decisiones para escoger la ruta viable que lo lleve a su destino en menos tiempo y con menos costo. Aunque la mayoría de los problemas de decisión involucran situaciones de optimización multiobjetivo, las soluciones computacionales tradicionales tienden a simplificarlos convirtiéndolos en problemas de un solo objetivo. Sin embargo, la necesidad de abordar la naturaleza multiobjetivo de estos problemas ha llevado al desarrollo de enfoques más avanzados, como la optimización multiobjetivo que se remonta a los trabajos pioneros de Wilfredo Pareto (1896) [11]. La formalización matemática de estos problemas comenzó con el trabajo seminal de Kuhn y Tucker en 1951 [2], seguido de importantes contribuciones de otros investigadores a lo largo de las décadas, ver por ejemplo [1, 3, 4, 5, 6, 9, 13].

En este artículo desarrollamos un problema de equilibrio de redes (tráfico urbano [10]) modelado como un problema de desigualdad Variacional (VIP por sus siglas en inglés Variational Inequality Problems) y usamos el método de punto proximal propuesto en [13] considerando distancias proximales para resolverlo. Métodos de punto proximal y distancias proximales han sido ya estudiadas por varios investigadores, ver por ejemplo el artículo [14] y sus referencias relacionadas.

Al usar el algoritmo proximal propuesto en [13] para resolver el problema (VIP) en cada paso iterativo nos encontramos ante un problema de resolver un sistema de ecuaciones no lineales que no fue desarrollado en el artículo mencionado. Así, la contribución de este trabajo está justamente en presentar un algoritmo adecuado para encontrar numéricamente la solución de este sistema en cada iteración del algoritmo. De esta manera, este trabajo contribuye al algoritmo introducido en [13] presentando una alternativa de como resolver estos subproblemas de forma práctica. Para ello, hemos utilizado el método de Newton con una regla de viabilidad motivado por el método de Armijo [12] para asegurar la no negatividad de las soluciones. La implementación del programa fue realizada en Python. Debemos observar también que este artículo es la primera implementación computacional que se da del algoritmo presentado en [13] para resolver un problema de equilibrio de redes.

La organización del artículo es la siguiente: Sección 1 es la introducción, la Sección 2 presenta los resultados previos que se debe conocer para entender el algoritmo, la Sección 3 modela el problema de equilibrio de redes como un problema (VIP) y presentamos un ejemplo para explicar detalladamente el modelo, en Sección 4 presentamos el algoritmo proximal y sus resultados teóricos, en la Sección 5 presentamos los resultados numéricos, en la Sección 6 damos nuestras conclusiones y en la Sección 7 los agradecimientos respectivos.

2 Resultados Previos

A lo largo de este artículo \mathbb{R}^n es el espacio euclidiano dotado del producto interior canónico $\langle \cdot, \cdot \rangle$ y la norma. x está dada por $\|x\| := \langle x, x \rangle^{1/2}$. Sea $B \in \mathbb{R}^{n \times n}$ una matriz simétrica positiva, denotamos $\|x\|_B := \langle Bx, x \rangle^{1/2}$. Del mismo modo la bola Euclidiana centrada en x con radio $\epsilon > 0$ como $B(x, \epsilon) = \{y \in \mathbb{R}^n : \|y - x\| < \epsilon\}$. El interior, clausura y frontera del subconjunto $X \subset \mathbb{R}^n$ se denota por $\text{int}(X)$, \bar{X} y $\text{bd}(X)$.

Definición 2.1 Sea $T : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ una aplicación. El dominio y el gráfico de T están definidos como

$$D(T) = \{x \in \mathbb{R}^n : T(x) \neq \emptyset\}.$$

$$G(T) = \{(x, v) \in \mathbb{R}^n \times \mathbb{R}^n : x \in D(T), v \in T(x)\}.$$

A seguir damos algunos conceptos relacionados a monotonia de aplicaciones punto-conjunto que son importantes para obtener buena definición y convergencia global del algoritmo introducido.

Definición 2.2 Una aplicación $T : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ es cerrada para \bar{x} si para alguna secuencia $\{x^k\} \subset \mathbb{R}^n$ y alguna secuencia en $\{v^k\} \subset \mathbb{R}^n$ tal que $(x^k, v^k) \in G(T)$ y $(x^k, v^k) \rightarrow (\bar{x}, \bar{v})$, implica que $\bar{v} \in T(\bar{x})$.

Definición 2.3 Una aplicación $T : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ es:

i. Fuertemente monótona si existe $\alpha > 0$ tal que

$$\langle u - v, x - y \rangle \geq \alpha \|x - y\|^2, \quad (1)$$

para todo $(x, u), (y, v) \in G(T)$.

ii. Monótona si

$$\langle u - v, x - y \rangle \geq 0, \quad (2)$$

para todo $(x, u), (y, v) \in G(T)$.

iii. Pseudo-monótona si

$$\langle v, x - y \rangle \geq 0 \Rightarrow \langle u, x - y \rangle \geq 0, \quad (3)$$

para todo $(x, u), (y, v) \in G(T)$.

iv. Cuasi-monótona si

$$\langle v, x - y \rangle > 0 \Rightarrow \langle u, x - y \rangle \geq 0, \quad (4)$$

para todo $(x, u), (y, v) \in G(T)$.

2.1 Distancia Proximal

Esta sección presenta las definiciones de distancia proximal y distancia proximal inducida, introducida por Auslender y Teboulle. [3]. Estos conceptos son importantes para la introducción de nuevos algoritmos donde las iteraciones se mantengan en el interior del conjunto de las restricciones viables. Este enfoque se ha utilizado en los trabajos de Villacorta y Oliveira [16], Papa Quiroz y Oliveira [14], Papa Quiroz et al. [15].

Definición 2.4 Una función $d : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}_+ \cup \{+\infty\}$ es llamada distancia proximal con respecto a un conjunto convexo abierto no vacío C si para cada $y \in C$ satisface las siguientes propiedades:

i. $d(\cdot, y)$ es propia, semicontinua inferior, estrictamente convexa y continuamente diferenciable en C .

ii. $\text{dom}(d(\cdot, y)) \subset \bar{C}$ y $\text{dom}(\partial_1 d(\cdot, y)) = C$, donde $\partial_1 d(\cdot, y)$ denota el subdiferencial de la función $d(\cdot, y)$ con respecto a la primera variable.

iii. $d(\cdot, y)$ es coerciva en \mathbb{R}^n (i.e., $\lim_{\|u\| \rightarrow \infty} d(u, y) = +\infty$);

iv. $d(y, y) = 0$.

Denotamos $D(C)$ la familia de funciones que satisfacen la definición anterior.

La propiedad **i**. es necesaria para preservar la convexidad de $d(\cdot, y)$, la propiedad **ii** forzará la iteración del método proximal a permanecer en C , y la propiedad **iii** es útil para asegurar la existencia de iteraciones proximales. Para cada $y \in C$, $\nabla_1 d(\cdot, y)$ denota el vector gradiente de la función $d(\cdot, y)$ con respecto a la primera variable. Tenga en cuenta que por definición $d(\cdot, \cdot) \geq 0$ y por **iv** el mínimo global de $d(\cdot, y)$ se obtiene y , lo cual implica que $\nabla_1 d(y, y) = 0$.

Definición 2.5 Dado $d \in D(C)$. Una función $H : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}_+ \cup \{+\infty\}$ se llama distancia proximal inducida a d si H es una función de valor finito en $C \times C$ y para cada $a, b \in C$ tenemos:

(Ii) $H(a, a) = 0$.

(Iii) $\langle c - b, \nabla_1 d(b, a) \rangle \leq H(c, a) - H(c, b), \quad \forall c \in C$.

Denotemos por $(d, H) \in \mathcal{F}(C)$ la distancia proximal que satisface las condiciones de la definición 2.5.

También denotamos $(d, H) \in \mathcal{F}(\bar{C})$ si existe $H : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}_+ \cup \{+\infty\}$ satisfaciendo:

(Iiii) H es valor finito en $\bar{C} \times C$ y satisface (Ii) y (Iii), para cada $c \in \bar{C}$.

(Iiv) Para cada $c \in \bar{C}$, $H(c, \cdot)$ tiene conjuntos de nivel acotados en C .

Finalmente, denotamos $(d, H) \in \mathcal{F}_+(\bar{C})$ si

(Iv) $(d, H) \in \mathcal{F}(\bar{C})$.

(Ivi) $\forall y \in \bar{C}$ y $\forall \{y^k\} \subset C$ acotado con $\lim_{k \rightarrow +\infty} H(y, y^k) = 0$, luego $\lim_{k \rightarrow +\infty} y^k = y$.

(Ivii) $\forall y \in \bar{C}$, y $\forall \{y^k\} \subset C$ tal que $\lim_{k \rightarrow +\infty} y^k = y$, entonces $\lim_{k \rightarrow +\infty} H(y, y^k) = 0$.

El principal resultado del método del punto proximal se ha probado cuando $(d, H) \in \mathcal{F}_+(\bar{C})$. Existen varios ejemplos de distancias proximales que satisfacen las definiciones anteriores, por ejemplo, distancias de Bregman, distancias basadas en φ -divergencia y distancias proximales homogéneas de segundo orden, fueron dadas por Auslender and Teboulle, [3], Section 3. Ver también los trabajos de Langenberg y Tichatschke [9], Kaplan y Tichatschke [7], [8]. Algunos ejemplos de distancias proximales son:

Ejemplo 2.1 Considerando que la distancia proximal

$$d(x, y) = \|x - y\|^2,$$

cumple con la definición 2.4, es decir, que $d \in D(\mathbb{R}^n)$. Además, la distancia proximal inducida es dada por $H(x, y) = \|x - y\|^2$, que cumple con las condiciones de la definición 2.5.

Ejemplo 2.2 La distancia proximal Logaritmo-Cuadrático:

$$d(w, v) = \begin{cases} \sum_{i=1}^n \frac{1}{2}(w_i - v_i)^2 + \mu \left(v_i^2 \log \left(\frac{v_i}{w_i} + w_i v_i - v_i^2 \right) \right), & \text{si } w \in \mathbb{R}^n \\ +\infty, & \text{en otro caso} \end{cases}$$

donde $0 < \mu < 1$ es un valor fijo dado, satisface las condiciones de las definiciones 2.4 y 2.5. En este caso la distancia inducida es $H(x, y) = (3/4)\|x - y\|^2$.

La distancia proximal dada en el ejemplo anterior será utilizada en la implementación del algoritmo y su aplicación a problemas de equilibrio de redes.

3 Problema de Equilibrio de Redes Modeladas como Problema de Desigualdad Variacional

En esta sección consideramos el problema de equilibrio de la red de tráfico con demandas elásticas en el que se supone que se conocen las funciones de desutilidad del viaje asociadas con los viajes entre pares origen/destino y presentamos la formulación de desigualdad variacional con las condiciones de equilibrio que gobiernan el modelo descrito por Nagurney y Zhang [10].

3.1 Formulaciones de desigualdad variacional

La red de transporte estará representada por un grafo dirigido denotado por $G(N, L)$, siendo N los nodos y L los enlaces dirigidos, denotamos con $p, q, \dots etc$ las rutas, que consisten en una secuencia de enlaces con cierta orientación, asimismo, denotamos como ω el par de nodos origen-destino O/D de la red y como Ω al conjunto de todos los pares O/D de la red, asumimos que hay J pares O/D .

Además P_ω denota el conjunto de todos los caminos conectando el par origen destino ω y sea $P = \cup_{\omega \in \Omega} P_\omega$, el conjunto de todos los caminos en la red.

Asumimos que hay m_ω caminos conectando el par origen destino ω y denotamos como m el número total de caminos conectados en la red, así $m = \sum_{\omega \in \Omega} m_\omega$

Sea x_p el flujo del tráfico sobre el camino P entonces el patrón de flujo de la ruta de tráfico x es un vector columna de dimensión m no negativo que se puede escribir como

$$x = \{x_p, p \in P\} \in \mathbb{R}_+^m.$$

La demanda del tráfico entre el par origen destino ω se denota por d_ω , y satisface

$$d_\omega = \sum_{p \in P_\omega} x_p, \quad x_p \geq 0, \forall p, \quad (5)$$

es decir, la suma de los flujos de rutas entre el par O/D ω debe ser igual a la demanda d_ω . Sea $d = \{d_\omega / \omega \in \Omega\}$ el vector de las demandas de viaje.

Sea f_a el enlace de carga sobre el enlace a , que debe satisfacer la siguiente ecuación de conservación del flujo.

$$f_a = \sum_{p \in P} \delta_{ap} x_p, \quad (6)$$

donde $\delta_{ap} = 1$, si el enlace a está contenido en la ruta p , y $\delta_{ap} = 0$ en caso contrario. La ecuación (6) establece que la carga de tráfico sobre el enlace a es igual a la suma de todos los flujos de rutas sobre las rutas p que atraviesan el enlace a .

Asumamos que $c = \{c_a, a \in L\}$ sea el vector fila de los costos de enlaces, donde c_a denota el costo de atravesar el enlace a . En general asumimos que el costo de enlace depende del flujo de cada enlace, esto es.

$$c = c(f), \quad (7)$$

donde c es una función dada y f denota el vector columna de las cargas enlaces. El usuario que viaja por la ruta p incurre en un costo de viaje C_p satisfaciendo

$$C_p = \sum_{a \in L} \delta_{ap} c_a, \quad (8)$$

es decir, el costo de atravesar a lo largo de un camino p es igual a la suma del costo de los enlaces que componen la ruta.

De (6), (7) y (8) observamos que tanto el costo de enlace y el costo de las rutas son funciones del patrón de flujo de ruta x , y por lo tanto se pueden escribir los siguientes vectores fila

$$c = c(f(x)) = c(x) \text{ y } C = C(f(x)) = C(x), \quad (9)$$

Asociada con cada par origen destino ω , hay una desutilidad de viaje λ_ω , que es función de la demanda de viaje d , es decir,

$$\lambda_\omega = \lambda_\omega(d) \quad (10)$$

Definición 3.1 (Equilibrio de la red de tráfico) [10] *Un patrón de flujo de ruta x^* , que induce un patrón de demanda d^* a través de (5) es un equilibrio de la red de tráfico, si para cada par entrada salida ω y cada ruta $p \in P_\omega$ las siguientes igualdades y desigualdades se mantienen:*

$$C_p(x^*) = \begin{cases} = \lambda_w(d^*), & \text{si } x_p^* > 0 \\ \geq \lambda_w(d^*), & \text{si } x_p^* = 0 \end{cases} \quad (11)$$

y

$$d_w(\lambda^*) = \begin{cases} = \sum_{p \in P_w} x_p^*, & \lambda_w^* > 0 \\ \leq \sum_{p \in P_w} x_p^*, & \lambda_w^* = 0 \end{cases} \quad (12)$$

Es decir, en equilibrio, sólo se utilizan aquellos caminos que conectan un par O/D que tienen costos mínimos para el usuario, y sus costos son iguales a la desutilidad del viaje asociado con viajar entre el par O/D .

Teorema 3.1 (Formulación de la Desigualdad Variacional) (Ver Nagurney y Zhang (1996) Pág 201 [10]). Dado $K \equiv \left\{ (x, d) : x \geq 0 \text{ y } d_w = \sum_{p \in P_w} x_p \right\}$, donde P_w denota el conjunto de todos los caminos conectando el par origen destino w y x_p el flujo del tráfico sobre el camino P . Decimos que $(x^*, d^*) \in K$ es un equilibrio de la red de tráfico si y solo si resuelve el problema de desigualdad variacional:

$$\langle C(x^*), x - x^* \rangle - \langle \lambda(d^*), d - d^* \rangle \geq 0, \forall (x, d) \in K. \quad (13)$$

Usando (5), podemos definir

$$\hat{\lambda} \equiv B^T \lambda(d(x)) = B^T \lambda(d), \quad (14)$$

donde, B es la matriz dimensional $J \times m$ cuya (w, p) -ésima componente es igual a 1 si el camino p está contenido en el par O/D w , y 0 de lo contrario.

3.2 Un Caso Particular

Para nuestro ejemplo damos el siguiente grafo dado en la Figura 1 de 4 nodos $N = \{A, B, G, H\}$ y 7 enlaces $L = \{a, b, g, h, s, e, l\}$. Consideremos tres pares origen-destino O/D en la red, donde

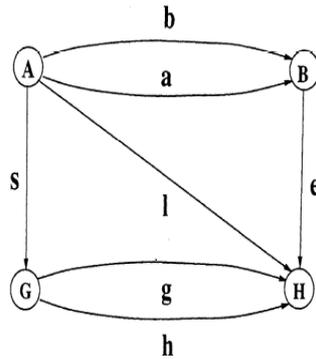


Figure 1: Ejemplo de Red de Transporte, tomado de [10]

$\Omega = \{\omega_1, \omega_2, \omega_3\}$ con $\omega_1 = (A, B)$, $\omega_2 = (G, H)$, $\omega_3 = (A, H)$. Así, $J = 3$.

Los caminos que unen estos O/D se enumeran como:

$P_{\omega_1} = \{p_1, p_2\}$, donde $p_1 = \{a\}$, $p_2 = \{b\}$,

$P_{\omega_2} = \{p_3, p_4\}$, donde $p_3 = \{g\}$, $p_4 = \{h\}$,

$P_{\omega_3} = \{p_5, p_6, p_7, p_8, p_9\}$, donde $p_5 = \{a, e\}$, $p_6 = \{b, e\}$, $p_7 = \{s, g\}$, $p_8 = \{s, h\}$, $p_9 = \{l\}$.

Así tenemos $P = \{p_1, p_2, p_3, \dots, p_9\}$.

En este caso hay $m_{\omega_1} = 2$, $m_{\omega_2} = 2$, $m_{\omega_3} = 5$, obteniendo $m = 9$, el número de caminos en la red. La matriz de incidencia de orden 9×7 , donde las filas están dadas por los caminos

$p_1, p_2, p_3, \dots, p_9$ y las columnas por los arcos a, b, g, h, s, e, l de esta red que muestra los enlaces utilizados en cada par origen destino se denota por A

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

y la matriz de incidencia B de orden 9×3 , formada por las filas que representan a los caminos $p_1, p_2, p_3, \dots, p_9$ y las columnas que representan origen-destino $\omega_1, \omega_2, \omega_3$, indica las rutas que pertenecen al origen-destino del par ω :

$$B = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

Sabemos que x_p es el flujo de tráfico de usar la ruta p . Entonces, el patrón del flujo de tráfico x es el vector $x = (x_{p_1}, x_{p_2}, x_{p_3}, x_{p_4}, x_{p_5}, x_{p_6}, x_{p_7}, x_{p_8}, x_{p_9}) \in \mathbb{R}_+^9$. La capacidad de flujo f se halla multiplicando la transpuesta de la matriz A con el patrón de flujo x

$$f = A^T x = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{p_1} \\ x_{p_2} \\ x_{p_3} \\ x_{p_4} \\ x_{p_5} \\ x_{p_6} \\ x_{p_7} \\ x_{p_8} \\ x_{p_9} \end{pmatrix} = \begin{pmatrix} x_{p_1} + x_{p_5} = f_a \\ x_{p_2} + x_{p_6} = f_b \\ x_{p_3} + x_{p_7} = f_g \\ x_{p_4} + x_{p_8} = f_h \\ x_{p_7} + x_{p_8} = f_s \\ x_{p_5} + x_{p_6} = f_e \\ x_{p_9} = f_l \end{pmatrix}$$

El vector columna d que agrupa las demandas de viaje d_ω se obtiene multiplicando la transpuesta de B con el vector patrón de flujo x

$$d = B^T x = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_{p_1} \\ x_{p_2} \\ x_{p_3} \\ x_{p_4} \\ x_{p_5} \\ x_{p_6} \\ x_{p_7} \\ x_{p_8} \\ x_{p_9} \end{pmatrix} = \begin{pmatrix} x_{p_1} + x_{p_2} = d_{\omega_1} \\ x_{p_3} + x_{p_4} = d_{\omega_2} \\ x_{p_5} + x_{p_6} + x_{p_7} + x_{p_8} + x_{p_9} = d_{\omega_3} \end{pmatrix}$$

Para este ejemplo el vector fila de lo costos es: $(c_a, c_b, c_g, c_h, c_e, c_s, c_l)$. Un usuario que viaja por la ruta p incurre en un costo de viaje C_p que satisface:

$$C_p = \sum_{a \in L} \delta_{ap} c_a \quad (15)$$

Para este ejemplo, los costos son dados por

$$\begin{aligned} c_a(f) &= 15f_a + 100 & c_b(f) &= \begin{cases} 20f_b + 4f_a, & \text{si } f_b \in [0, 5] \\ 2f_b^2 + 4f_a + 50, & \text{si } f_b \in [5, +\infty) \end{cases} \\ c_g(f) &= \begin{cases} 10f_g + 40, & \text{si } f_g \in [0, 10] \\ 0.5f_g^2 + 90, & \text{si } f_g \in [10, +\infty) \end{cases} & c_h(f) &= 2f_h + f_g + 90 \\ c_s(f) &= 10f_s + 40 & c_e(f) &= f_e + 5f_s + 6f_g + 6f_h + 40 \\ c_l(f) &= 10f_l + 2f_a + 20 \end{aligned}$$

Así de (15), tenemos

$$\begin{aligned} C_{p_1} &= c_a(f), \quad C_{p_2} = c_b(f), \quad C_{p_3} = c_g(f), \quad C_{p_4} = c_h(f), \quad C_{p_5} = c_a(f) + c_e(f), \\ C_{p_6} &= c_b(f) + c_e(f), \quad C_{p_7} = c_s(f) + c_g(f), \quad C_{p_8} = c_h(f) + c_s(f), \quad C_{p_9} = c_l(f), \end{aligned}$$

Para cada par origen destino ω , existen desutilidades asociadas al viaje λ^* , que están en función de la demanda de viaje d . En el ejemplo la demanda está dada por:

$$\begin{aligned} d_{w_1} &= x_{p_1} + x_{p_2} \\ d_{w_2} &= x_{p_3} + x_{p_4} \\ d_{w_3} &= x_{p_5} + x_{p_6} + x_{p_7} + x_{p_8} + x_{p_9} \end{aligned}$$

Asimismo las funciones de desutilidad de viaje están dadas por:

$$\begin{aligned} \lambda_{w_1}(d_{w_1}) &= -5d_{w_1} + 100 \\ \lambda_{w_2}(d_{w_2}) &= -4d_{w_2} - 2d_{w_3} + 126 \\ \lambda_{w_3}(d_{w_3}) &= -5d_{w_3} - 5d_{w_1} + 160 \end{aligned}$$

Debemos determinar $x^* \in \mathbb{R}_+^9$ que satisfaga la desigualdad variacional:

$$\langle \mathbf{C}(\mathbf{x}^*) - \lambda(\mathbf{x}^*), \mathbf{x} - \mathbf{x}^* \rangle \geq 0 \quad (16)$$

donde $C(x) = (C_{p_1}, C_{p_2}, C_{p_3}, C_{p_4}, C_{p_5}, C_{p_6}, C_{p_7}, C_{p_8}, C_{p_9})$ y $\lambda(x) = (\lambda_{w_1}, \lambda_{w_2}, \lambda_{w_3})$.

Para encontrar el patrón de flujo de trayectoria de equilibrio hemos utilizado el algoritmo proximal y el método de Newton junto con la regla de Armijo en el lenguaje de programación Python. Los resultados los veremos después de complementar la teoría del algoritmo proximal.

4 Algoritmo Proximal y Resultados teóricos

Estamos interesados en resolver el problema de desigualdad variacional (VIP), el cual consiste en encontrar $x^* \in \bar{C}$, y un $y^* \in T(x^*)$, tal que

$$\langle y^*, x - x^* \rangle \geq 0, \forall x \in \bar{C}, \quad (17)$$

donde $T : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ es una aplicación punto a conjunto (no necesariamente monótono), C es un conjunto convexo abierto no vacío. \bar{C} es la clausura de C en \mathbb{R}^n y $D(T) \cap C \neq \emptyset$. Proponemos una extensión del (PPM) con una distancia proximal, llamada algoritmo de Punto Proximal Inexacto (IPP), para resolver el problema.(17).

Algoritmo (IPP)

Inicialización: Sea $\{\lambda_k\}$ una secuencia de parámetros positivos y sea un punto de partida:

$$x^0 \in C. \tag{18}$$

Pasos Principales: Para $k = 1, 2, \dots$, y dado $x^{k-1} \in C$, encontramos $x^k \in C$, $u^k \in T(x^k)$ y $e^k \in \mathbb{R}^n$, tal que:

$$u^k + \lambda_k \nabla_1 d(x^k, x^{k-1}) = e^k, \tag{19}$$

donde d es la distancia proximal tal que $(d, H) \in \mathcal{F}_+(\overline{C})$ y e^k es un error aproximado que satisface la siguiente hipótesis:

$$\frac{\|e^k\|}{\lambda_k} \leq \eta_k \sqrt{H(x^k, x^{k-1})}, \tag{20}$$

$$\sum_{k=1}^{+\infty} \eta_k < +\infty, \eta_k > 0. \tag{21}$$

Criterio de Parada: Si $x^k = x^{k-1}$ o $0 \in T(x^k)$, entonces finaliza. De lo contrario, hacer $k - 1 \leftarrow k$ y retornar al inicio.

Ahora presentamos el resultado de convergencia global que fue probado en el artículo [13], asumiendo las siguientes hipótesis:

(H1) Para cada $k \in \mathbb{N}$, existe $x^k \in C$.

(H2) El conjunto solución de (VIP), denotado por $SOL(T, \overline{C})$, es no vacío.

(H3) T es una aplicación localmente acotada y $G(T)$ es cerrado.

Teorema 4.1 *Sea T una aplicación pseudo-monótona, $(d, H) \in \mathcal{F}_+(\overline{C})$, y supongamos que las hipótesis **(H1)**, **(H2)** y **(H3)** juntamente con **(Iviii)** son satisfechas. Si $0 < \lambda_k < \bar{\lambda}$, entonces $\{x^k\}$ converge para un punto de $SOL(T, \overline{C})$.*

Para el caso cuasi-monótono denote $Acc(x^k)$ como el conjunto de todos los puntos de acumulación de $\{x^k\}$, esto es,

$$Acc(x^k) = \left\{ z \in \overline{C} : \text{existe una subsecuencia } \{x^{k_j}\} \text{ of } \{x^k\} : x^{k_j} \rightarrow z \right\}.$$

Supongamos ahora que T es una aplicación cuasi-monótona y consideremos el siguiente subconjunto $SOL(T, \overline{C})$:

$$SOL^*(T, \overline{C}) = \{x^* \in SOL(T, \overline{C}) : \exists u^* \neq 0, u^* \in T(x^*)\}.$$

Asumimos que $SOL(T, \overline{C}) \cap bd(C) \neq \emptyset$ y sustituiremos la hipótesis **(H2)** por:

(H2)' $SOL^*(T, \overline{C}) \neq \emptyset$.

Así, para el caso cuasi-monótono obtenemos el siguiente resultado, ver [13].

Teorema 4.2 *Sea T una aplicación cuasi-monótona, $(d, H) \in \mathcal{F}_+(\overline{C})$, y suponga que las hipótesis **(H1)**, **(H2)'**, **(H3)** y **(Iviii)** son satisfechas con $0 < \lambda_k < \bar{\lambda}$, entonces*

- (a) $\{x^k\}$ converge debilmente para un elemento de $SOL(T, \overline{C})$, esto es, $Acc(x^k) \neq \emptyset$ y todo elemento de $Acc(x^k)$ es un punto de $SOL(T, \overline{C})$.
- (b) Si $Acc(x^k) \cap SOL^*(T, \overline{C}) \neq \emptyset$, entonces $\{x^k\}$ converge para un elemento de $SOL^*(T, \overline{C})$.

5 Solución de los Subproblemas

El algoritmo (IPP) es bien general pues no nos da la metodología de como calcular x^k , u^k y e^k satisfaciendo (19). En este artículo proponemos el siguiente método para el caso cuando la aplicación T es punto a punto, es decir, una función. Consideremos el siguiente sistema no lineal

$$F_k(x) = 0, \quad x \in C := \mathbb{R}_{++}^n$$

donde $F_k(x) = T(\cdot) + \lambda_k \nabla_1 d(\cdot, x^{k-1})$.

Algorithm 1 Método de Newton con Viabilidad

Paso 0: Dado $x^{k-1} \in C$ y F_k

Paso 1: Tomar un punto inicial $v1$ tal que $F_k(v1) \neq 0$.

Paso 2: Resolver el sistema de ecuaciones lineales

$$J(v1)d = F(v1),$$

donde $J(v1)$ es el Jacobiano de F en el punto $v1$ para obtener la dirección de Newton d .

$n1 \leftarrow \text{length}(v1)$

$t \leftarrow 1$

Mientras $\sum(v1 - (1/t) \cdot d) \geq \text{zeros}(n1) < n1$, hacer*

$t \leftarrow t + 1$

Retornar $v1 \leftarrow v1 - (1/t)d$.

$e \leftarrow F(v1)$

Paso 3: Si $\frac{\|e\|}{\lambda_k} \leq \eta_k \sqrt{H(v1, x^{k-1})}$, entonces finalizar el método de Newton e imprimir $x^k = v1$, $e^k = e$, $u^k = T(v1)**$.

Caso contrario, volver al Paso 2.

* $\text{zeros}(n1)$ es el vector de ceros de dimensión $n1$ y la condición

$$\sum(v1 - (1/t)d) \geq \text{zeros}(n1) < n1, \tag{22}$$

realiza el siguiente procedimiento:

1. Primero compara componente a componente el vector $(v1 - (1/t)d)$ con el vector $\text{zeros}(n1)$. Si en la comparación de cierta componente la desigualdad \geq es verdadera, entonces la componente toma un valor de 1 y si es falsa toma un valor de 0. Luego realiza una suma de los valores entre ceros y unos.
2. Luego el valor de la suma es comparado con el valor de $n1$. Si el valor de la suma es menor que $n1$ entonces la longitud de paso $1/t$ es disminuido para $1/(t+1)$ y de nuevo se realiza el test del Mientras.

**La parte final del paso tres, encuentra el nuevo punto, el vector error y el vector resultado de la aplicación de T en el nuevo punto.

Así en cada iteración del algoritmo (IPP) para obtener los valores x^k , u^k y e^k se encuentra una dirección de Newton y luego se hace el test (22) para encontrar la longitud de paso que sea viable en C . El método de Newton finaliza cuando las condiciones (20) y (21) son satisfechas.

Como el objetivo de este artículo es la parte numérica, no daremos resultados teóricos sobre el algoritmo de Newton con viabilidad para que finalice en un número finito de iteraciones cumpliendo las condiciones (20) y (21).

6 Resultados Numéricos

En esta sección presentamos los resultados numéricos del algoritmo introducido. Resolvemos tres problemas de desigualdad variacional usando la distancia proximal del Ejemplo (2.2).

Problema 1

Dada la siguiente función: $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ definido por:

$$f(x, y) = (x - 3)^2 + (y - 5)^2 \tag{23}$$

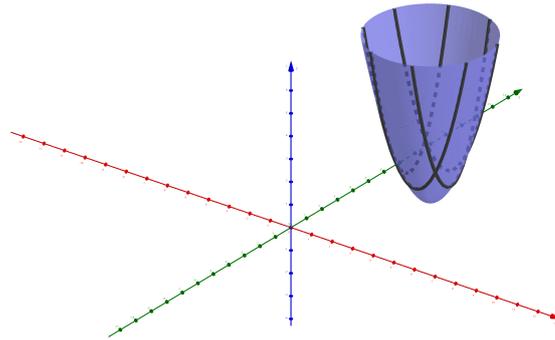


Figure 2: Gráfico de la función f del Problema 1.

Hallaremos un (x, y) que sea el punto mínimo de la función sujeta a la restricción que el punto se encuentra en el ortante no negativo $\bar{C} = \mathbb{R}_+^2 = \{(x, y) : x \geq 0, y \geq 0\}$. Para ello escribimos el problema en su forma variacional:

$$\langle T(\bar{x}, \bar{y}), (x, y) - (\bar{x}, \bar{y}) \rangle \geq 0, \quad \forall (x, y) \in \bar{C},$$

donde T es el gradiente de la función f , esto es,

$$\begin{aligned} T : \mathbb{R}^2 &\rightarrow \mathbb{R}^2 \\ (x, y) &\rightarrow 2(x - 3, y - 5). \end{aligned}$$

Considerando la distancia

$$d(x, y) = \sum_{i=1}^2 \frac{1}{2} (x_i - y_i)^2 + \mu \left[y_i^2 \ln \left(\frac{y_i}{x_i} \right) + x_i y_i - y_i^2 \right],$$

con $\mu > 0$.

Aplicamos el algoritmo (IPP):

Para $k = 0$, tomamos $\mathbf{x}^0 = (2, 1)$, $\mu = 0.5$. y una tolerancia dada= TOL

Para $k = 1$, sea $\lambda_k = \eta_k = \frac{1}{(k + 1)^2}$, para encontrar $\mathbf{x}^1 = (x^1, y^1)$ debemos resolver el sistema no lineal (24)

$$T(x) + \lambda \nabla d(x, x^0) = 0.$$

$$\begin{pmatrix} x - 3 \\ y - 5 \end{pmatrix} + \lambda \begin{pmatrix} x - x^0 + \mu \left(\frac{(x^0)^2}{-x} + x^0 \right) \\ y - y^0 + \mu \left(\frac{(x^0)^2}{-y} + y^0 \right) \end{pmatrix} = 0. \tag{24}$$

Utilizamos el algoritmo de la Sección 5, método de Newton con viabilidad, para asegurar que la solución no salga de la región convexa $C = \mathbb{R}_{++}^2$. Para encontrar $\mathbf{x}^{(1)}$ consideramos el criterio de parada para las iteraciones internas

$$\|e^1\| \leq \lambda_1 \eta_1 \sqrt{H(x^1, x^0)}.$$

Si se cumple la desigualdad anterior, paramos, sino continuamos. Así sucesivamente vamos generando la sucesión $\{x^k\}$.

La tabla 6 nos muestra los resultados obtenidos al dar una tolerancia de 0.0001. La primera columna nos muestra el número de iteraciones necesarias del método proximal y la segunda columna el número de iteraciones internas que necesitamos del método de Newton para obtener el siguiente punto. La tercera columna son los valores que va tomando λ_k en cada iteración externa las otras dos son las normas y la solución buscada, respectivamente. Finalmente la última columna son los vectores errores e^k en obtenida en cada iteración del algoritmo proximal. Como puede apreciarse se obtiene la solución deseada que es el punto (3, 5) que minimiza la función objetivo dada en (23).

Iter k	Inner-Iter	λ_k	$\ x^{k+1} - x^k\ $	(x_1^k, x_2^k)	e^k
1	3	1	2.03907708	$\begin{pmatrix} 1.88277312 \\ 2.83808242 \end{pmatrix}$	$\begin{pmatrix} -0.03426304 \\ -0.04475215 \end{pmatrix}$
2	2	0.25	1.828385137	$\begin{pmatrix} 2.71865672 \\ 4.46420998 \end{pmatrix}$	$\begin{pmatrix} -0.03061822 \\ -0.07006583 \end{pmatrix}$
3	2	0.111	0.520972790	$\begin{pmatrix} 2.96075073 \\ 4.92551588 \end{pmatrix}$	$\begin{pmatrix} -0.00109155 \\ -0.00237987 \end{pmatrix}$
4	2	0.0625	0.0770052867	$\begin{pmatrix} 2.99664806 \\ 4.9936422 \end{pmatrix}$	$\begin{pmatrix} -0.00001 \\ -0.000029 \end{pmatrix}$
5	2	0.04	0.0067806	$\begin{pmatrix} 2.99981033 \\ 4.99964026 \end{pmatrix}$	$\begin{pmatrix} -0.00000006 \\ -0.00000014 \end{pmatrix}$
6	2	0.0278	0.0003904112	$\begin{pmatrix} 2.99999241 \\ 4.99998561 \end{pmatrix}$	$\begin{pmatrix} 0.0000 \\ 0.0000 \end{pmatrix}$
7	2	0.020409	0.0000157	$\begin{pmatrix} 2.99999977 \\ 4.99999957 \end{pmatrix}$	$\begin{pmatrix} 0.00000 \\ 0.00000 \end{pmatrix}$
8	2	0.015625	0.0000004	$\begin{pmatrix} 2.99999999 \\ 4.99999999 \end{pmatrix}$	$\begin{pmatrix} 0.000000 \\ 0.000000 \end{pmatrix}$

Table 1: Resultados obtenidos del Problema 1

Problema 2

De la misma forma, siguiendo con el ejemplo de la sección 3.2 donde obtuvimos una desigualdad variacional (16) aplicamos el método proximal siguiendo los mismos pasos que en el ejemplo anterior. Los valores x_p^* , f^* , $C(x^*)$, d^* y λ_ω , fueron obtenidos aplicando la desigualdad variacional y el método proximal, hemos utilizado el Método de Newton, a través del método de Newton podemos obtener soluciones negativas, para esto hemos utilizado un pseudocódigo con una condición de no negatividad, empezando con el parámetro $d = 1$ y vamos decreciendo hasta obtener la viabilidad deseada, a continuación presentamos el pseudocódigo

También, mostramos el vector e^k obtenido al reemplazar la función en la última iteración interna.

Problema 3

Como último ejemplo hemos considerado una red diseñada por los autores, tomando 24 nodos y 27 enlaces como se muestra en la Figura 3.

En este caso tenemos 4 rutas: $\omega_1 = (A, X)$, $\omega_2 = (A, Q)$, $\omega_3 = (H, X)$, $\omega_4 = (A, P)$ Los caminos que unen estos O/D son:

Iter k	Inner-Iter	λ_k	$\ x^{k+1} - x^k\ $	x^k
1	5	1	26.43409	(1.147, 3.722, 4.242, 2.402, 0.374, 0.493, 0.798, 0.711, 8.408)
2	2	0.25	1.729	(0.436, 3.816, 4.526, 1.129, 0.134, 0.185, 0.322, 0.261, 7.990)
3	3	0.111	0.521	(0.227, 3.894, 4.719, 0.663, 0.0674, 0.0944, 0.1634, 0.133, 7.969)
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
88	5	0.00013	0.00003	(0.0003, 3.999, 4.999, 0.00123, 0.00009, 0.00013, 0.00022, 0.00018, 7.999)
89	2	0.0278	0.0004	(0.00032, 3.999, 4.999, 0.0013, 0.00009, 0.00013, 0.00021, 0.00018, 7.999)
90	5	0.00012	0.00000	(0, 4, 5, 0, 0, 0, 0, 0, 8)

Table 2: Resultados obtenidos del Problema 2

Iter k	Inner-Iter	e^k
1	5	(-0.2948, -0.00025, -0.0056, -0.00159, -0.0735, -0.0321, -3.1418, -0.0216, -0.0014)
2	2	(-0.000004, 0, -0.000001, -0.0003, -0.008, -0.000028, -0.0017, -0.0004, 0)
3	3	(0, 0, 0, 0, 0.000001, -0.00012, -0.00014, -0.000003, 0)
⋮	⋮	⋮
⋮	⋮	⋮
88	5	(0, 0, 0, 0, 0, 0, 0, 0, 0)
89	5	(0, 0, 0, 0, 0, 0, 0, 0, 0)
90	5	(0, 0, 0, 0, 0, 0, 0, 0, 0)

Table 3: Resultados de los vectores errores e^k del Problema 2

$$P_{\omega_1} = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}\}, P_{\omega_2} = \{p_7, p_8, p_9, p_{10}\}, P_{\omega_3} = \{p_{11}, p_{12}, p_{13}, p_{14}, p_{15}\}, P_{\omega_4} = \{p_{16}, p_{17}, p_{18}\}.$$

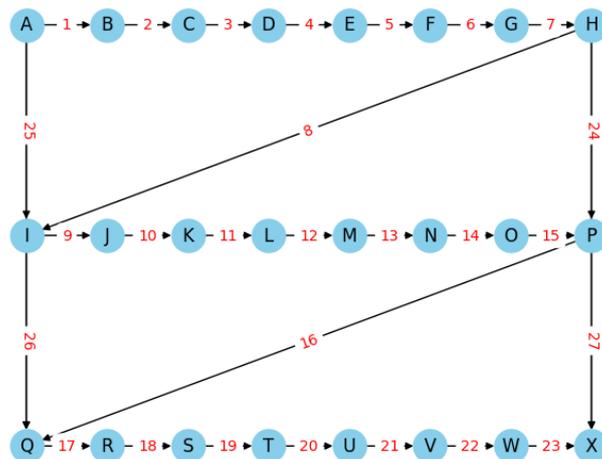


Figure 3: Red de 24 nodos y 27 enlaces

Asimismo se asumen los siguientes costos:

$$\begin{aligned}
 c_1 &= 5 \times 10^{-5} f_1^4 + 5f_1 + 2f_2 + 400 & c_{15} &= 3 \times 10^{-5} f_{15}^4 + 9f_{15} + 2f_{14} + 200 \\
 c_2 &= 3 \times 10^{-5} f_2^4 + 4f_2 + 4f_1 + 250 & c_{16} &= 8f_{16} + 5f_{12} + 300 \\
 c_3 &= 5 \times 10^{-5} f_3^4 + 3f_3 + f_4 + 300 & c_{17} &= 3 \times 10^{-5} f_{17}^4 + 7f_{17} + 2f_{15} + 340 \\
 c_4 &= 3 \times 10^{-5} f_4^4 + 6f_4 + 3f_5 + 400 & c_{18} &= 5f_{18} + f_{16} + 300 \\
 c_5 &= 6 \times 10^{-5} f_5^4 + 6f_5 + 4f_6 + 600 & c_{19} &= 8f_{19} + 3f_{17} + 600 \\
 c_6 &= 7f_6 + 3f_7 + 600 & c_{20} &= 3 \times 10^{-5} f_{20}^4 + 6f_{20} + f_{21} + 300 \\
 c_7 &= 8 \times 10^{-5} f_7^4 + 8f_7 + 2f_8 + 400 & c_{21} &= 4 \times 10^{-5} f_{21}^4 + 4f_{21} + f_{22} + 400 \\
 c_8 &= 4 \times 10^{-5} f_8^4 + 5f_8 + 2f_9 + 650 & c_{22} &= 2 \times 10^{-5} f_{22}^4 + 6f_{22} + f_{23} + 500 \\
 c_9 &= 10 \times 10^{-5} f_9^4 + 6f_9 + 2f_{10} + 700 & c_{23} &= 3 \times 10^{-5} f_{23}^4 + 9f_{23} + 2f_{24} + 300 \\
 c_{10} &= 4f_{10} + f_{12} + 800 & c_{24} &= 2 \times 10^{-5} f_{24}^4 + 8f_{24} + f_{25} + 900 \\
 c_{11} &= 7 \times 10^{-5} f_{11}^4 + 7f_{11} + 4f_{12} + 650 & c_{25} &= 3 \times 10^{-5} f_{25}^4 + 9f_{25} + 3f_{26} + 400 \\
 c_{12} &= 8f_{12} + 2f_{13} + 700 & c_{26} &= 6 \times 10^{-5} f_{26}^4 + 7f_{26} + 8f_{27} + 300 \\
 c_{13} &= 10 \times 10^{-5} f_{13}^4 + 7f_{13} + 3f_{18} + 600 & c_{27} &= 4 \times 10^{-5} f_{27}^4 + 8f_{27} + 3f_{26} + 200 \\
 c_{14} &= 8f_{14} + 3f_{15} + 500 & &
 \end{aligned}$$

y las funciones de desutilidad del viaje, están dadas por:

$$\begin{aligned}
 \lambda_{w_1}(d_{w_1}) &= -5d_{w_1} + 400, & \lambda_{w_2}(d_{w_2}) &= -6d_{w_2} + 600, \\
 \lambda_{w_3}(d_{w_3}) &= -d_{w_3} + 500, & \lambda_{w_4}(d_{w_4}) &= -6d_{w_4} + 800.
 \end{aligned}$$

El método de Newton se utilizó, así como la regla de Armijo para evitar que las variables salgan negativas, El método de Newton converge en 41 iteraciones internas y tres externas, con la secuencia $\lambda_k = \frac{1}{(k+1)^2}$ El método requiere 3 segundos,

Ahora presentamos el patrón de equilibrio en forma resumida, en particular, el patrón de carga de enlace de equilibrio calculado, el patrón de demanda calculado y las desutilidades de viaje inducidas.

Iter k	Inner-Iter	λ_k	$\ x^{k+1} - x^k\ $	x^k
1	3	1	0.00424	x^1
2	4	0.25	0.0008	x^2
3	4	0.111	0.0003	x^3
4	5	0.111	0.0001	x^4
\vdots	\vdots	\vdots	\vdots	\vdots
\vdots	\vdots	\vdots	\vdots	\vdots
16	5	0.003	0.000003	x^{16}
17	6	0.003	0.000002	x^{17}
18	6	0.0028	0.00001	x^{18}
19	6	0.0025	0.00000	x^{19}

Table 4: Resultados obtenidos del Problema 3

Los vectores $x^1, x^2, x^3, \dots, x^{19}$ están dados en la siguiente tabla en forma vertical.

x^1	x^2	x^3	x^4	...	x^{16}	x^{17}	x^{18}	x^{19}
0.00009	0.00003	0.00002	0.00001	...	0.000001	0.000001	0.0000009	0.00003
0.00027	0.0001	0.00006	0.00004	...	0.000003	0.00003	0.000003	0.0001
0.0007	0.0003	0.0002	0.0001	...	0.000009	0.000009	0.000008	0.0003
0.00025	0.0001	0.00006	0.00004	...	0.000004	0.000003	0.000002	0.0001
0.00031	0.0001	0.00008	0.00005	...	0.000004	0.000004	0.000003	0.0001
0.00042	0.0001	0.0001	0.00006	...	0.000006	0.000005	0.000005	0.0001
0.00013	0.00006	0.00003	0.00002	...	0.000002	0.000002	0.000001	0.00006
0.00005	0.00002	0.00001	0.000007	...	0.0000007	0.0000006	0.0000005	0.00002
0.00002	0.00008	0.00004	0.00002	...	0.000002	0.000002	0.000002	0.00008
0.0009	0.0004	0.0002	0.0001	...	0.00001	0.00001	0.000009	0.0004
0.00007	0.00003	0.00001	0.00001	...	0.000001	0.0000009	0.0000008	0.00003
0.00006	0.00003	0.00001	0.000009	...	0.0000008	0.0000008	0.0000006	0.00003
0.0001	0.00007	0.00004	0.00002	...	0.000002	0.000002	0.000002	0.00007
0.0002	0.00009	0.00006	0.00003	...	0.000003	0.000003	0.000002	0.00009
0.0003	0.0001	0.00008	0.00005	...	0.000004	3.858031225978092e-6	0.000003	0.0001
0.00003	0.00001	0.000008	0.000005	...	0.0000004	4.0135498964300296e-7	0.0000003	0.00001
0.00004	0.00002	0.00001	0.000008	...	0.0000007	6.123849816800025e-7	0.0000005	0.00002
0.0003	0.0001	0.00008	0.00005	...	0.000004	3.7812213311587262e-6	0.000003	0.0001

Table 5: Los valores de los puntos x^k

Los vectores de errores e^k están dados por:

Iter k	Inner-Iter	e^k
1	3	((0.0, 0.0, 0.00, 0.000, 0.0, 0.00, 0.0, 0.0, 0.0, 0.00, 0.00, 0.00, 0.0, 0.0, 0.000, 0.0, 0.0, 0.000))
2	4	((-0.000000000008, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00))
3	4	((-0.000000000001, 0.00000, 0.000, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00))
4	5	((0.000, 0.00000, 0.000, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00))
⋮	⋮	⋮
⋮	⋮	⋮
16	5	((0.0, 0.0, 0.0, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00))
17	6	((0.0, 0.0, 0.0, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00))
18	6	((0.0, 0.0, 0.0, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00))
19	6	((0.0, 0.0, 0.0, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00))

Table 6: Resultados de los errores e^k del Problema 3

7 Conclusión

1. En este trabajo, presentamos un abordaje de como resolver los subproblemas del algoritmo (IPP) aplicado a problemas de equilibrio de redes modelada como un problema de desigualdad variacional. Esta técnica es basada en la búsqueda de direcciones de Newton considerando una técnica de viabilidad. Realizamos tres tests computacionales para verificar la funcionalidad del método en los subproblemas y presentamos nuestros resultados en sus respectivas tablas. Los resultados muestran que el algoritmo (IPP) con el método de solución para los subproblemas puede ser aplicado a problemas de equilibrio de redes.
2. Los resultados de este trabajo motivan a un futuro estudio teórico de la terminación finita del algoritmo de Newton con viabilidad para obtener los puntos del algoritmo (IPP).
3. Un estudio interesante puede ser extender el algoritmo (IPP) para resolver el problema (17) cuando el conjunto C contiene desigualdades como también igualdades.

8 Agradecimientos

Los autores agradecen a la Facultad de Ciencias Matemáticas de la Universidad Nacional Mayor de San Marcos por la infraestructura y los laboratorios que nos permitió desarrollar el trabajo de investigación de manera apropiada y segura.

Este artículo ha sido financiado con fondos de la Universidad a través del proyecto EQUIPAMIENTO CIENTIFICO DE INVESTIGACIÓN - ECI UNMSM 2024, titulado: Métodos proximales para resolver problemas de desigualdad variacional: aplicación a problemas de equilibrio de redes, con número de Proyecto: B241410915e.

Referencias bibliográficas

- [1] Auslender, A., Teboulle, M., Ben-Tiba, S.: Interior proximal and multiplier methods based on second order homogeneous functionals. *Math Oper. Research.* 24, 3, 645-668 (1999)
- [2] Kuhn, H.W. and Tucker, A.W. *Nonlinear Programming. Proceedings of the 2nd Berkeley Symposium on Mathematics, Statistics and Probability*, University of California Press, Berkeley, 481-492 (1951)
- [3] Auslender, A., Teboulle, M.: Interior gradient and proximal methods for convex and conic optimization. *SIAM Journal of Optimization.* 16, 3, 697-725 (2006)
- [4] Facchinei, F., Pang, J.S.: *Finite-Dimensional Variational Inequalities and Complementarity Problems, Vol I and II*, Springer Series in Operations Research. Springer-Verlag, New York (2003)
- [5] Harker, P.T., Pang, J.S.: Finite-dimensional variational inequality and nonlinear complementarity problems: a survey of theory, algorithms and applications. *Mathematical Programming*, 48, 161-220 (1990)
- [6] Kaplan, A., Tichatschke, R.: On inexact generalized proximal methods with a weakened error tolerance criterion. *Optimization.* 53, 3-17 (2004)
- [7] Kaplan, A., Tichatschke, R.: Interior proximal method for variational inequalities on non-polyhedral sets. *Discuss. Math. Diff. Inclusions Control Optim.* 30, 51-59 (2007)
- [8] Kaplan, A., Tichatschke, R.: Note on the paper: interior proximal method for variational inequalities on non-polyhedral sets. *Discuss. Math. Diff. Inclusions Control Optim.* 30, 51-59 (2010)
- [9] Langenberg, N., Tichatschke, R.: Interior proximal methods for quasiconvex optimization. *J. Glob Optim.* 52, 641-661 (2012)
- [10] Nagurney, A., Zhang, D.: *Projected Dynamical Systems and Variational Inequalities with Applications*. En *International series in management science/operations research/International series in operations research & management science*. (1996) <https://doi.org/10.1007/978-1-4615-2301-7>
- [11] Edgeworth., F. Y., 1881.: *Mathematical physics*. P. Keagan, London, England.
- [12] *Mathematics Subject Classification (2000): 90B30, 90C11, 90-01, 90-02* Library of Congress Control Number: 2006923897 ISBN-10: 0-387-30303-0
- [13] Papa Quiroz, S, Cruzado, R: Linear and superlinear convergence of an inexact algorithm with proximal distances for variational inequality problems. *Fixed Point Theory, Volume 23, No. 1, 2022, 311-330, February 1st, 2022.*
- [14] Papa Quiroz, E.A., Oliveira, P.R.: An extension of proximal methods for quasiconvex minimization on the nonnegative orthant. *European Journal of Operational Research.* 216, 26-32 (2012)
- [15] Papa Quiroz, E.A., Mallma Ramirez, L., Oliveira, P.R.: An inexact proximal method for quasiconvex minimization. Accepted for Publication in *European Journal of Operational Research* (2015)
- [16] Villacorta, K.D.V., Oliveira, P.R.: An interior proximal method in vector optimization. *European Journal of Operational Research.* 214, 485-492 (2011)