

Optimización de modelos de Stackelberg no estacionarios mediante un algoritmo evolutivo auto-adaptativo

Optimization of non-stationary Stackelberg models using a self-adaptive evolutionary algorithm

Olga P. Cedeño-Fuentes¹, Lorena Arboleda-Castro²
Iván Jacho-Sánchez³ y Pavel Novoa-Hernández⁴

Recibido: 30 de marzo de 2017,
Aceptado: 03 de mayo de 2017

Cómo citar / How to cite

O. P. Cedeño-Fuentes, L. Arboleda-Castro, I. Jacho-Sánchez y P. Novoa-Hernández "Optimización de modelos de Stackelberg no estacionarios mediante un algoritmo evolutivo auto-adaptativo", *TecnoLógicas*, vol. 39, no. 20, 2017.

-
- 1 MSc en Gestión Empresarial, Facultad de Ciencias Empresariales, Universidad Técnica Estatal de Quevedo, Quevedo-Ecuador, ocedeno@uteq.edu.ec
 - 2 MSc en Finanzas y Proyectos Corporativos, Facultad de Ciencias Empresariales, Universidad Técnica Estatal de Quevedo, Quevedo-Ecuador, larboleda@uteq.edu.ec
 - 3 MSc en Tributación y Finanzas, Facultad de Ciencias Empresariales, Universidad Técnica Estatal de Quevedo, Quevedo-Ecuador, ijacho@uteq.edu.ec
 - 4 PhD en Tecnologías de la Información y la Comunicación, Facultad de Ciencias de la Ingeniería, Universidad Técnica Estatal de Quevedo, Quevedo-Ecuador, pnovoa@uteq.edu.ec



Resumen

Los modelos de Juegos de Stackelberg engloban una importante familia de problemas de la Teoría de Juegos, que encuentra aplicaciones directas en economía. El principal objetivo es encontrar un equilibrio óptimo entre las decisiones que pueden tomar dos actores que se relacionan jerárquicamente. En general estos modelos son complejos de resolver dada su estructura jerárquica, y la frecuente aparición en estos de funciones objetivos o restricciones intratables analíticamente. Otra causa de dicha complejidad es la existencia de incertidumbre, particularmente debido a la variabilidad en el tiempo de las condiciones del mercado, estrategias de los competidores, entre otras. Un análisis de la literatura relacionada muestra muy pocos trabajos abordando estos problemas de optimización no estacionarios. En este sentido, la presente investigación propone una técnica meta-heurística auto-adaptativa para resolver modelos de Juegos de Stackelberg no estacionarios. Los resultados experimentales obtenidos muestran una mejoría significativa sobre un método existente.

Palabras clave

Juegos de Stackelberg, optimización evolutiva de dos niveles no estacionaria, evolución diferencial, auto-adaptación, pruebas no paramétricas.

Abstract

Stackelberg's game models involve an important family of Game Theory problems with direct application on economics scenarios. Their main goal is to find an optimal equilibrium between the decisions from two actors that are related one to each other hierarchically. In general, these models are complex to solve due to their hierarchical structure and intractability from an analytical viewpoint. Another reason for such a complexity comes from the presence of uncertainty, which often occurs because of the variability over time of market conditions, adversary strategies, among others aspects. Despite their importance, related literature reflects a few works addressing this kind of non-stationary optimization problems. So, in order to contribute to this research area, the present work proposes a self-adaptive meta-heuristic method for solving online Stackelberg's games. Experiment results show a significant improvement over an existing method.

Keywords

Stackelberg games, non-stationary bi-level optimization, differential evolution, self-adaptation, pruebas no paramétricas.

1. INTRODUCCIÓN

Los denominados problemas de competencia de Stackelberg [1] involucran una clase importante de escenarios de decisión que encuentran aplicaciones directa en el ámbito económico. Estos problemas han sido estudiados dentro de disciplinas como la Teoría de Juegos, siendo abordados, por tanto, desde la perspectiva de los métodos de la Investigación de Operaciones.

Desde un punto de vista general, los problemas de competencia de Stackelberg se pueden ver como modelos de optimización de dos niveles [2], [3], en los que dos agentes o compañías compiten entre sí teniendo en cuenta las reacciones óptimas del adversario. A diferencia de otros modelos de juego clásicos (ej. de suma cero) en los de Stackelberg existe una jerarquía: uno de los dos competidores será el líder, mientras que el otro se podrá ver como un subordinado. No resulta difícil concluir que varios escenarios de decisión reales pueden encajar en esta descripción general.

En la actualidad existen numerosos trabajos que poseen como objeto de estudio a los modelos de competencia de Stackelberg. Véase por ejemplo [4]–[9]. Sin embargo, hasta donde se conoce, en muy pocos se asume incertidumbre en el modelo.

En un trabajo previo [10], desarrollado por los autores de la presente investigación, esta temática fue abordada considerando incertidumbre de tipo dinámica, en la función objetivo del modelo subordinado. Esta incertidumbre se asumió no conocida por el algoritmo y por tanto no tratable desde el punto de vista probabilístico. En otras palabras, se estudiaron modelos de optimización de dos niveles no estacionarios. Como método de solución se implementó una técnica metaheurística [11] que resulta un híbrido entre un enfoque co-evolutivo y un enfoque multipoblacional para ambientes dinámicos. No obstante, los resultados obtenidos, es aún una interrogante si se puede diseñar un algoritmo más eficiente para resolver este tipo de proble-

ma. De acuerdo con [12], un posible punto de mejora podría estar relacionado con la adaptación de la búsqueda. Concretamente, los autores de [12] proponen una estrategia auto-adaptativa que es capaz de controlar la intensidad de la búsqueda durante la ejecución y que mostró un rendimiento significativo en problemas continuos no estacionarios.

Basados en esta importante evidencia, en el presente trabajo se propone incluir esta estrategia auto-adaptativa en el método propuesto en [10]. De esta forma se espera que el nuevo método exhiba una mejoría apreciable sobre el ya propuesto.

Para una mejor comprensión de nuestra propuesta, el resto del trabajo queda estructurado de la forma siguiente: en lo que sigue se revisa la literatura relacionada con el tema. Más adelante, se describe el modelo que será objeto de estudio y el método de solución para resolverlo. Las dos secciones que siguen están dedicadas a presentar y discutir los resultados obtenidos de los experimentos computacionales realizados. Las conclusiones y planes de trabajos futuros son expuestos al final del artículo.

2. REVISIÓN DE LA LITERATURA

En la actualidad el uso de métodos metaheurísticos para resolver modelos de Stackelberg (problemas de optimización de dos niveles) ha recibido un gran interés por parte de la comunidad científica [13]. Una de las principales razones de este interés es la capacidad de estos métodos en obtener soluciones suficientemente buenas en un tiempo razonable en escenarios complejos [11].

Como sucede en otros escenarios de decisión, la incertidumbre puede estar presente en los modelos de Stackelberg, siendo la principal causa la naturaleza dinámica de los datos involucrados. Un ejemplo de estos modelos dinámicos de dos niveles pueden ser problemas de localización y

ruteo, con un número variable en el tiempo de depósitos y clientes. De igual forma, la distribución óptima de ayuda en desastres recurrentes puede ser modelada como un modelo de Stackelberg no estacionario. Es importante notar que esta incertidumbre incrementa la complejidad del problema, dado que ahora el método de solución no solo debe encontrar un único equilibrio óptimo, si no varios equilibrios a lo largo del tiempo.

En este sentido, la literatura refleja algunos trabajos interesantes. Por ejemplo, Sun y colaboradores [14] emplean un algoritmo genético para resolver un problema de señales de tráfico dinámico. Los autores abordaron redes de tráfico con demandas variables y elecciones de rutas estocásticas. Aquí, el modelo superior correspondió al control de señales, mientras que el viaje del usuario fue representado en el modelo inferior.

Por su parte, Linnala y colaboradores [15] presentaron un modelo general para problemas de dos niveles dinámicos y multi-objetivo. Los autores analizaron la interacción entre los dos submodelos a lo largo del tiempo, y describen los beneficios de su propuesta. Con el objetivo de ilustrarlas, los autores analizaron un caso de estudio en el ámbito industrial relacionado con el proceso de fabricación de papel. En el modelo superior se consideró el proceso de diseño, mientras que en el modelo inferior el proceso de fabricación. En particular, se resolvió este modelo de dos niveles empleando un algoritmo basado en el paradigma Evolución Diferencial [16].

Otro estudio interesante sobre optimización en dos niveles dinámica fue llevado a cabo por Cheng y Song [17]. Aquí, los autores abordaron el modelado y solución de un problema de selección de portafolio multi-período en mercados estocásticos con control de riesgo de quiebra. En esta investigación se asume que el inversor desea encontrar la estrategia de inversión que maximice sus beneficios, considerando el riesgo de quiebra se encuentra controlado

en cada período. Dado que este problema pudo ser abordado desde el punto de vista de la Programación Dinámica, los autores emplearon un método clásico para obtener soluciones analíticas para cada período.

Recientemente, en [10], los autores del presente trabajo, propusieron un método evolutivo basado en evolución diferencial que resuelve modelos de Stackelberg dinámicos. Lo novedoso de esta investigación es que se propone un método híbrido que aprovecha los beneficios de las metaheurísticas en dos contextos diferentes. Por un lado, el enfoque coevolutivo que resulta particularmente exitoso en el ámbito de la optimización en dos niveles [13], y por otro, estrategias de diversidad durante la ejecución que lo son en el ámbito de la optimización dinámica evolutiva [18], [19]. Un enfoque similar será adoptado en la presente investigación, pero aprovechando una técnica de auto-adaptación de parámetros. La auto-adaptación es una técnica de control de parámetros muy popular en la optimización evolutiva y que ha resultado particularmente efectiva en la optimización dinámica [20], [21]. Se trata específicamente de incluir en el método propuesto en [10], estrategias de diversidad durante la ejecución de tipo auto-adaptativas.

3. DEFINICIÓN DEL MODELO A RESOLVER

El modelo que se abordará para estudiar el método propuesto es el presentado previamente en [10]. Matemáticamente queda definido como (1):

$$\begin{aligned} \max_{x,y,Q} \Pi_l &= \sum_{i=1}^n (P(Q_i)x_i - C(x_i)) \\ \text{sujeto a} & \\ \max_y \Pi_f &= \sum_{i=1}^n (P(Q_i)y_i - C(y_i, \phi_i, t)) \\ x_i + y_i &\geq Q_i \quad \forall i \\ x_i, y_i, Q_i &\geq 0 \quad \forall i \end{aligned} \tag{1}$$

donde $x_i, y_i \in \mathbb{R} (i = 1, \dots, n)$ son variables de decisión que expresan el nivel de producción correspondiente al producto i en las firmas l (líder) y f (subordinada), respectivamente. Por su parte, $Q_i \in \mathbb{R} (i = 1, \dots, n)$ son variables de decisión asociadas a la demanda del producto i , mientras que P y C son funciones relacionadas al precio de venta y costo total en cada firma, respectivamente. En particular, la función de costo de la firma subordinada $C(y_i, \phi_i, t)$ es dinámica. Esta depende de una función de control de cambios ϕ_i y el tiempo t . Finalmente, es importante mencionar que el objetivo global es maximizar las ganancias de la firma líder (Π_l), teniendo en cuenta las reacciones óptimas de la firma subordinada (Π_f).

El lector interesado podrá encontrar una explicación más detallada sobre este modelo en la referencia [10].

4. DESCRIPCIÓN DEL MÉTODO PROPUESTO

El método de solución propuesto en esta sección es una extensión al del trabajo previo [10]. En aquella ocasión se desarrolló un algoritmo sin adaptación en los parámetros que definen su estrategia de búsqueda. Como bien fue advertido en el propio trabajo previo, el grado de aplicabilidad de ese algoritmo en escenarios con diferentes características (ej. intensidad de los cambios) es limitado. Esto se debe a que su efectividad dependerá significativamente de la configuración manual que el investigador elija de los parámetros. Lo ideal sería que el algoritmo seleccionara, en tiempo de ejecución, la configuración más apropiada de acuerdo con el estado actual de la búsqueda. En la literatura existen evidencias importantes sobre el beneficio de este enfoque que ha sido denominado como *auto-adaptación* [22]. Véase por ejemplo [20].

Sin embargo, resulta una tarea compleja diseñar estrategias auto-adaptativas que involucren a todos los parámetros del algoritmo. La principal razón es la explosión combinatoria del número de configuraciones que pueden ser consideradas. Una alternativa efectiva es seleccionar solo un subconjunto de aquellos parámetros que más influyen en la solución del problema en cuestión.

En este sentido, se ha considerado la estrategia auto-adaptativa propuesta en [12], [20], para resolver problemas dinámicos de optimización. Específicamente, esta estrategia modifica en tiempo de ejecución el radio de diversidad que permite al algoritmo explorar localmente soluciones prometedoras. Es de notar que no en todas las etapas del proceso de optimización se requiere el mismo grado de exploración. Por ejemplo, cuando ha ocurrido un cambio reciente, es de esperar que el espacio de búsqueda haya cambiado también y las mejores soluciones obtenidas en el pasado hayan quedado desplazadas o simplemente no existan. Por tanto, se necesitaría un mayor grado de exploración capaz de identificar nuevas zonas prometedoras. En cambio, cuando al transcurrir varias iteraciones y una vez confirmada la calidad de las nuevas zonas identificadas, es preciso explotarlas (refinarlas) con la intención de maximizar la calidad de las soluciones encontradas. Es evidente que se necesitaría un grado de exploración menor que en las iteraciones iniciales después del cambio. Es precisamente este comportamiento el que logra la estrategia auto-adaptativa propuesta en [12].

Dicho lo anterior, se espera por tanto que el rendimiento del nuevo algoritmo supere al propuesto en [10] al incluir dicha estrategia. Como se aprecia en el trabajo original [12], esta estrategia depende de dos parámetros. Estos son la tasa de mutación (τ) y el factor de escala (λ). De manera que estos parámetros resultan de interés para analizar la influencia del método propuesto en diferentes escenarios.

Los pasos generales del algoritmo, que hemos denominado *CDEA+SQ* (*Coevolutionary Differential Evolution Algorithm with Self-adaptive Quantum diversity*), son los siguientes:

Algoritmo 1. CSQDEA.

Entrada:

Sub-modelos M_{sup} y M_{inf} .

Sub-algoritmos A_{sup} y A_{inf} que incluyen Evolución Diferencial y Estrategia de diversidad auto-adaptativa quantum [12].

Tamaño de la población μ .

Frecuencia de intercambio de información $\Delta h = 30$.

Tasa de mutación (τ) y factor de escala (λ) de la estrategia de diversidad auto-adaptativa.

Salida:

Mejores soluciones de ambos sub-modelos en cada ambiente (antes del cambio).

Paso 1. Crear una población P_0 con μ individuos (soluciones candidatas) aleatoriamente siguiendo una distribución uniforme. Inicializar los valores de radio de diversidad para cada individuo de la población según la estrategia de diversidad auto-adaptativa [12]. Inicializar contador de iteraciones $h = 1$.

Paso 2. Inicializar A_{sup} y A_{inf} con copias de la población P_0 .

Paso 3. REPETIR:

Paso 4. SI se detectan cambios en el ambiente mediante la reevaluación de las mejores soluciones de A_{sup} y A_{inf} en sus respectivas funciones objetivo, **ENTONCES:**

Paso 5. Reevaluar todas las soluciones en A_{sup} y A_{inf} .

Paso 6. SI NO:

Paso 7. SI es tiempo de intercambiar información ($h \bmod. \Delta h == 0$), **ENTONCES:**

Paso 8 Intercambiar la mejor solución entre A_{sup} y A_{inf} . Primero A_{sup} le envía su mejor solución a A_{inf} quien distribuye la parte x en toda su población y la reevalúa. La mejor solución de A_{inf} en términos de y es enviada a A_{sup} quien la distribuye en su población y la reevalúa.

Paso 9. SI NO:

Paso 10. Actualizar a las población de A_{sup} y A_{inf} según los pasos del paradigma Evolución Diferencial [16], y la estrategia de diversidad auto-adaptativa [12]. Actualizar la mejor solución encontrada por A_{sup} y A_{inf} .

Paso 11. Incrementar contador de iteraciones ($h = h + 1$).

Paso 12. HASTA se cumple la condición de parada (número máximo de cambios es alcanzado).

Es importante mencionar que al realizar las operaciones de evaluar (o reevaluar) una solución, no sólo se involucran al valor de la función objetivo, sino también al nivel de cumplimiento de dicha solución en la restricción funcional relacionada con la demanda. De esta manera el algoritmo considera como mejor solución aquella con la mayor calidad y menor grado de incumplimiento posible en las restricciones. Más detalles sobre cómo implementar este enfoque pueden ser encontrados en [10] y [23].

5. RESULTADOS

Los experimentos computacionales realizados estuvieron orientados al cumplimiento de dos objetivos. Primeramente, determinar una configuración apropiada de parámetros que resuelva eficientemente distintos escenarios del modelo considerado. En segundo lugar, comparar las mejores configuraciones del algoritmo propuesto con respecto al método presentado en [10].

En la Tabla 1 se muestran los escenarios de prueba que fueron derivados del modelo definido en la sección anterior, y que fueron el objeto de estudio de los experimentos desarrollados.

En relación con el algoritmo se consideraron un total de 25 variantes, las cuales se derivan de la combinación de los siguientes valores para los parámetros τ y λ : {0.1, 0.3, 0.5, 0.7, 0.9}. El resto de los parámetros del algoritmo se establecieron tal y como sugieren los trabajos [10], [12].

Como medidas de rendimiento se consideraron: la media de la calidad (valor de la función objetivo) de la mejor solución del algoritmo antes del cambio [24], [25], y el grado de factibilidad de la mejor solución antes del cambio. La fórmula matemática correspondiente a la primera medida es la siguiente:

$$F_{msac} = \frac{1}{T} \sum_{k=1}^T \Pi_{\{l,f\}}^k(\hat{x}) \quad (2)$$

donde T es la cantidad de cambios que experimentará el ambiente, $\Pi_{\{l,f\}}^k$ es la función objetivo en el ambiente k , y \hat{x} es la mejor solución obtenida por el algoritmo antes de ocurrir el cambio k .

De manera similar a la expresión anterior, se agrupará este nivel de factibilidad a través de la media:

$$G_{msac} = \frac{1}{T} \sum_{k=1}^T (\min\{\hat{x}_{i,k} + \hat{y}_{i,k} - \hat{Q}_{i,k} : i = 1, \dots, n\}) \quad (3)$$

donde $\hat{x}_{i,k}$, $\hat{y}_{i,k}$ y $\hat{Q}_{i,k}$ son las i -ésimas componentes de la mejor solución del algoritmo antes del cambio k . En este caso, un valor negativo indica que se incumple con la restricción, mientras que uno positivo o igual a cero que se cumple.

Los experimentos fueron implementados en el software MATLAB 2015b y ejecutados en una computadora con sistema operativo Debian 9, 16GB de RAM, y un procesador AMD. Se desarrollaron 20 ejecuciones independientes por cada algoritmo sobre cada escenario.

Tabla 1. Escenarios del modelo de Juego de Stackelberg no estacionario empleados en los experimentos.

Fuente: Arboleda-Castro y colaboradores [10].

Parámetros	Escenario 1	Escenario 2	Escenario 3
n (cantidad de productos)	2	5	10
Espacio de búsqueda para x_i, Q_i, y_i	$\in [0, 100]$	$\in [0, 100]$	$\in [0, 100]$
α_i	$\in [500, 1000]$	$\in [500, 1000]$	$\in [500, 1000]$
β_i	$\in [0.1, 1.0]$	$\in [0.1, 1.0]$	$\in [0.1, 1.0]$
δ_i	$\in [0.1, 1.0]$	$\in [0.1, 1.0]$	$\in [0.1, 1.0]$
γ_i	$\in [0.1, 1.0]$	$\in [0.1, 1.0]$	$\in [0.1, 1.0]$
c_i	$\in [3, 6]$	$\in [3, 6]$	$\in [3, 6]$
sev (severidad de los cambios)	1% del rango admisible del parámetro	5% del rango admisible del parámetro	10% del rango admisible del parámetro
T (número de cambios)	50	50	50
Δe (frecuencia de los cambios)	Cada 10 000 evaluaciones de las funciones objetivo	Cada 25 000 evaluaciones de las funciones objetivo	Cada 50 000 evaluaciones de las funciones objetivo

En la Fig. 1 se muestran los resultados de los experimentos realizados en términos de las medidas de rendimiento definidas en las expresiones (2) y (3). De manera concentrada se han expresado los resultados para cada uno de los tres escenarios descritos, y el promedio para todos los escenarios. Por ejemplo, los gráficos a), b), c) y d) de la Fig. 1 corresponden al Escenario 1, en relación con la calidad de la mejor solución antes del cambio y el cumplimiento de la restricción en las compañías F y L (modelo superior e inferior, respectivamente). En el caso de las gráficas de calidad (F_{msac}) los colores claros corresponden a las mejores variantes del algoritmo propuesto. Por ejemplo, considere la gráfica a), en esta las mejores variantes del algoritmo son las representadas por las tercera y cuarta columna. Esto es, para $\tau = [0.5, 0.7]$ y $\lambda = [0.1, 0.9]$. En el caso de la medida G_{msac} los colores oscuros corresponden a valores cercanos a 0.

Con el objetivo de realizar un comparación estadística entre las mejores variantes de nuestro algoritmo, y el método propuesto en [10], se aplicaron pruebas no paramétricas como se sugiere en [26]. Las 10 mejores variantes de nuestro algoritmo que

fueron consideradas para este análisis estadístico son las que se muestran en el eje x de la gráfica incluida en la Fig. 2. Nótese que las variantes están denotadas de la forma $(\tau + \lambda)$ y fueron seleccionadas a partir de los resultados que exhibieron en general (véase las gráficas m y n de la Fig. 1). Es importante destacar además que los datos empleados para este análisis corresponden a la medida F_{msac} (compañía L y F) teniendo en cuenta todas las ejecuciones.

De acuerdo con lo que se sugiere en [26], primero se aplicó una prueba de Friedman para obtener los rangos (posiciones) medios de los algoritmos y detectar diferencias a nivel de grupo. Esta prueba arrojó los rangos medios mostrados en la Fig. 2, y un p-valor igual a 4.6×10^{-3} . Por consiguiente, se concluye que sí existen diferencias a nivel de grupo. A partir de este resultado, se prosiguió con la aplicación de una prueba post-hoc de Holm para determinar si la mejor variante de nuestro algoritmo es realmente diferente al de la literatura. En este caso se comprueba que tal diferencia existe dado que se obtuvo un p-valor igual a 8.4×10^{-3} (véase la Fig. 2).

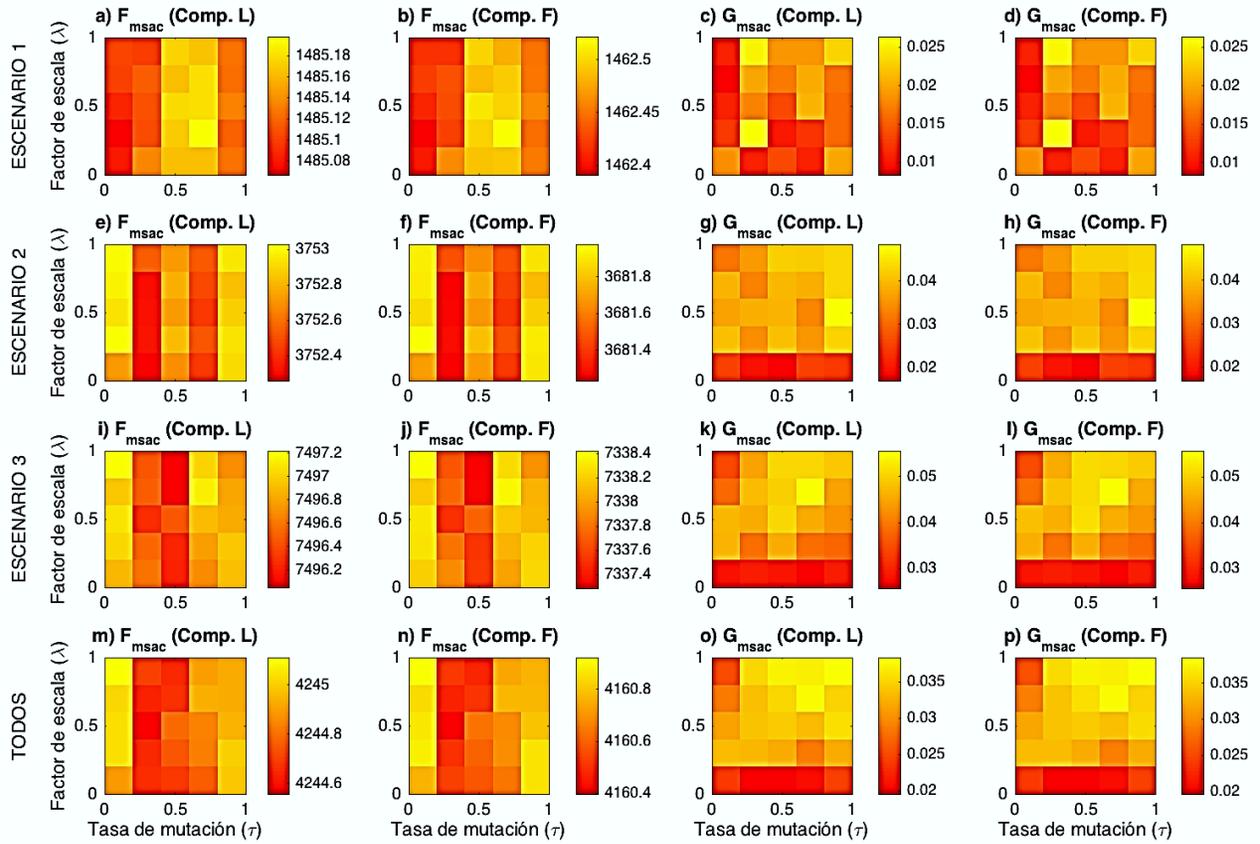


Fig. 1. Resultados de los experimentos para las 25 variantes del algoritmo propuesto en cada escenario del problema y en todos. Fuente: Autores.

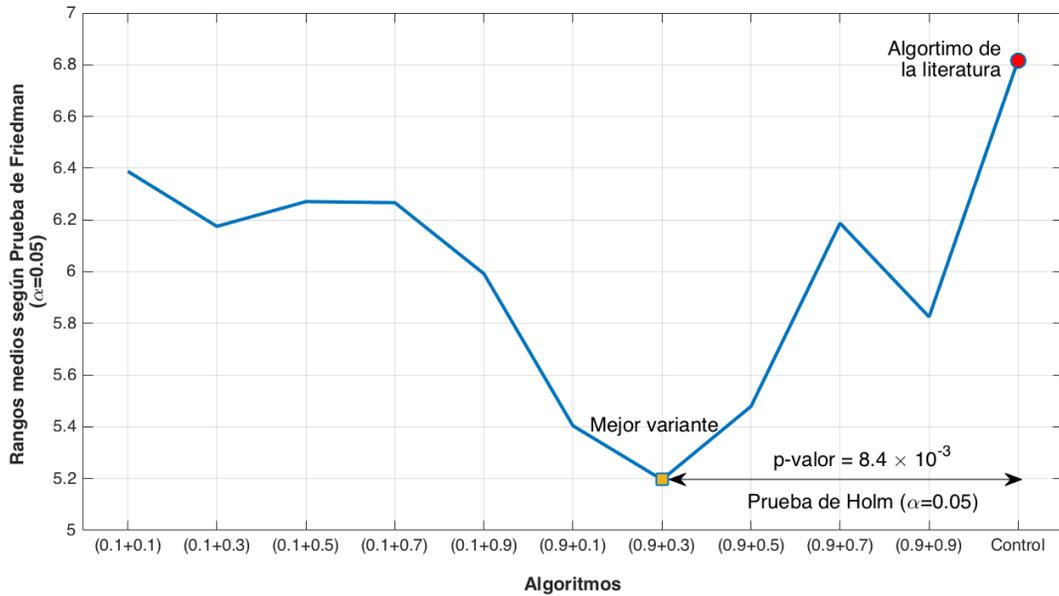


Fig. 2. Resultados de las pruebas no paramétricas realizadas entre las mejores variantes del algoritmo propuesto vs. el algoritmo propuesto en la literatura [10]. Fuente: Autores.

4. DISCUSIÓN

Como se aprecia en los gráficos de la Figura 1, los parámetros τ y λ juegan un papel importante en el rendimiento del algoritmo, especialmente el parámetro τ . Por ejemplo, nótese que, para el Escenario 1 donde la severidad de los cambios y el número de productos es menor, las mejores variantes del algoritmo son aquellas con valores de $\tau = \{0.5, 0.7\}$ y cualquier valor para λ . A diferencia de este escenario simple, en los otros dos escenarios se aprecia otro tipo de configuración. En estos casos las mejores variantes se concentran en torno a los valores $\tau = \{0.1, 0.9\}$.

En sentido general, se observa además que todas las variantes del algoritmo propuesto mantienen niveles de factibilidad aceptables en todos los escenarios. De hecho, como se aprecia en las gráficas dedicadas a la medida G_{msac} , los valores alcanzados por las variantes son siempre positivos.

En relación con el análisis estadístico, se puede ver que las variantes consideradas superan ligeramente al algoritmo de la literatura (véase los rangos medios en la Fig. 2). No obstante estas pequeñas diferencias, bajo la perspectiva de la prueba post-hoc de Holm se puede concluir que al menos la mejor variante, esto es, aquella con $\tau = 0.9$ y $\lambda = 0.3$, es significativamente diferente al algoritmo de la literatura. Lo anterior demuestra que la propuesta resulta efectiva para resolver modelos de Juegos de Stackelberg no estacionarios.

5. CONCLUSIONES

En el presente trabajo se propuso un método metaheurístico auto-adaptativo para la solución de modelos de Stackelberg no estacionarios. A partir de los resultados de las simulaciones numéricas realizadas se concluye que:

1. el método propuesto es sensible a los parámetros que definen a la estrategia auto-adaptativa considerada, y
2. el método propuesto mejora significativamente al presentado previamente.

En el futuro se planea profundizar en este tema con el desarrollo de cuatro líneas de investigación:

1. el estudio de modelos de Juegos de Stackelberg no estacionarios asociados a escenarios reales,
2. la propuesta de nuevos métodos inteligentes que dependan menos de la configuración de parámetros,
3. la aplicación de métodos metaheurísticos híbridos que exploten adecuadamente la estructura del problema, y
4. la comparación de los métodos propuestos con otros similares que representen el estado del arte en la optimización evolutiva de problemas no estacionarios de dos niveles.

6. REFERENCIAS

- [1] H. von Stackelberg, *The Theory of the Market Economy*. Oxford University Press, 1952.
- [2] B. Colson, P. Marcotte, and G. Savard, 'Bilevel programming: A survey', *4OR*, vol. 3, no. 2, pp. 87–107, 2005.
- [3] B. Colson, P. Marcotte, and G. Savard, 'An overview of bilevel optimization', *Ann. Oper. Res.*, vol. 153, no. 1, pp. 235–256, 2007.
- [4] X. León and L. Navarro, 'A Stackelberg game to derive the limits of energy savings for the allocation of data center resources', *Futur. Gener. Comput. Syst.*, vol. 29, no. 1, pp. 74–83, 2013.
- [5] P. Nie, 'A note on dynamic Stackelberg games with leaders in turn', *Nonlinear Anal. Real World Appl.*, vol. 13, no. 1, pp. 85–90, 2012.
- [6] T. Nakamura, 'One-leader and multiple-follower Stackelberg games with private information', *Econ. Lett.*, vol. 127, pp. 27–30, 2015.

- [7] B. Øksendal, L. Sandal, and J. Ubøe, 'Stochastic Stackelberg equilibria with applications to time-dependent newsvendor models', *J. Econ. Dyn. Control*, vol. 37, no. 7, pp. 1284–1299, 2013.
- [8] D. Wang, G. Du, R. J. Jiao, R. Wu, J. Yu, and D. Yang, 'A Stackelberg game theoretic model for optimizing product family architecting with supply chain consideration', *Int. J. Prod. Econ.*, vol. 172, pp. 1–18, 2016.
- [9] O. A. Wahab, J. Bentahar, H. Otrok, and A. Mourad, 'A Stackelberg game for distributed formation of business-driven services communities', *Expert Syst. Appl.*, vol. 45, pp. 359–372, 2016.
- [10] L. Arboleda-Castro, O. Cedeño-Fuentes, I. Jacho-Sánchez, and P. Novoa-Hernández, 'Un enfoque computacional evolutivo para problemas de competencia de Stackelberg dinámicos (An evolutionary computational approach for the dynamic Stackelberg competition problems)', *Enfoque UTE*, vol. 7, no. 2, pp. 10–24, 2016.
- [11] I. Boussaïd, J. Lepagnot, and P. Siarry, 'A survey on optimization metaheuristics', *Inf. Sci. (N.Y.)*, vol. 237, pp. 82–117, 2013.
- [12] P. Novoa-Hernández, C. C. Corona, and D. A. Pelta, 'Self-adaptive, multipopulation differential evolution in dynamic environments', *Soft Comput.*, vol. 17, no. 10, pp. 1861–1881, Mar. 2013.
- [13] E.-G. Talbi, 'A Taxonomy of Metaheuristics for Bi-level Optimization', in *Metaheuristics for Bi-level Optimization*, vol. 482, E.-G. Talbi, Ed. Springer Berlin Heidelberg, 2013, pp. 1–39.
- [14] D. Sun, R. F. Benekohal, and S. T. Waller, 'Bi-level programming formulation and heuristic solution approach for dynamic traffic signal optimization', *Comput. Civ. Infrastruct. Eng.*, vol. 21, no. 5, pp. 321–333, 2006.
- [15] M. Linnala, E. Madetoja, H. Ruotsalainen, and J. Hämäläinen, 'Bi-level optimization for a dynamic multiobjective problem', *Eng. Optim.*, vol. 44, no. 2, pp. 195–207, 2012.
- [16] R. Storn and K. Price, 'Differential Evolution - A Simple and Efficient Heuristic for global Optimization over Continuous Spaces', *J. Glob. Optim.*, vol. 11, no. 4, pp. 341–359, 1997.
- [17] Z. Chen and Z. Song, 'Dynamic portfolio optimization under multi-factor model in stochastic markets', *OR Spectr.*, vol. 34, no. 4, pp. 885–919, 2012.
- [18] T. T. Nguyen, S. Yang, and J. Branke, 'Evolutionary dynamic optimization: A survey of the state of the art', *Swarm Evol. Comput.*, vol. 6, no. 0, pp. 1–24, 2012.
- [19] E. Alba, A. Nakib, and P. Siarry, *Metaheuristics for Dynamic Optimization*, vol. 433. Springer Berlin Heidelberg, 2013.
- [20] P. Novoa-Hernández, C. C. Corona, and D. A. Pelta, 'Self-adaptation in dynamic environments - a survey and open issues', *Int. J. Bio-Inspired Comput.*, vol. 8, no. 1, pp. 1–13, 2016.
- [21] P. Novoa-Hernández, B. O. Bayas, J. M. Oviedo, A. Puris, M. Menace, and C. Cruz Corona, 'Impacto de la auto-adaptación en ambientes dinámicos con frecuencia de cambio variable', *Investig. Operacional*, vol. 37, no. 3, pp. 292–302, 2016.
- [22] S. Meyer-Nieberg and H.-G. Beyer, 'Self-Adaptation in Evolutionary Algorithms', in *Parameter Setting in Evolutionary Algorithms*, vol. 54, F. Lobo, C. Lima, and Z. Michalewicz, Eds. Springer Berlin / Heidelberg, 2007, pp. 19–46.
- [23] A. Sinha, P. Malo, and K. Deb, 'Test Problem Construction for Single-Objective Bilevel Optimization', *Evol. Comput.*, vol. 22, no. 3, pp. 439–477, Dec. 2014.
- [24] P. Novoa-Hernández, D. A. Pelta, and C. C. Corona, 'Improvement strategies for multi-swarm PSO in dynamic environments', in *Studies in Computational Intelligence*, 2010, vol. 284, pp. 371–383.
- [25] P. Novoa-Hernández, C. C. Corona, and D. A. Pelta, 'Efficient multi-swarm PSO algorithms for dynamic environments', *Memetic Comput.*, vol. 3, no. 3, pp. 163–174, 2011.
- [26] S. García, D. Molina, M. Lozano, and F. Herrera, 'A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 Special Session on Real Parameter Optimization', *J. Heuristics*, vol. 15, pp. 617–644, 2009.