IDENTIFICACIÓN DE RUTAS ROBUSTAS UTILIZANDO ACO

Amilkar Puris[1*], Bryan Steven Cortez Chichande[*], Edison Leodan Vicente Illescas[*], Pavel Novoa-Hernández[**]
[1]Universidad Técnica Estatal de Quevedo, Ecuador.
[2]Escuela de Ciencias Empresariales, Universidad Católica del Norte, Coquimbo, Chile.

ABSTRACT
In this work, a study based on the ACO (Ant Colony Optimization) metaheuristic is carried out, on which different models are proposed to study robust problems (several scenarios) taking as reference an instance of the TSP problem (Traveling Salesman Problem). The objective was to analyze how the level of importance of the scenarios in a ROOT problem affects the performance of the algorithms. For this, a case study was built 4 random variants of the Oliver30.tsp instance. The results revealed that the quality of the results largely depends on the importance of the scenarios in the time windows. The best rated approaches were the further, the more important and the less important. While considering the same importance for all scenarios proved to be a poor strategy for robust problems.

KEYWORDS: Resource allocation problems, Optimization model, Ant Colony System, Robust optimization over time.

MSC: 91B32

RESUMEN
En este trabajo se realiza un estudio basado en la metaheurística ACO (Ant Colony Optimization) sobre el cual se propone diferentes modelos para estudiar problemas robustos (varios escenarios) tomando como referencia una instancia del problema TSP (Traveling Salesman Problem). El objetivo fue analizar cómo el nivel de importancia de los escenarios en un problema ROOT, incide en el rendimiento de los algoritmos. Para ello, se construyó un caso de estudio generando 4 variantes aleatorias de la instancia Oliver30.tsp. Los resultados develaron que la calidad de los resultados depende en gran medida de la importancia de los escenarios en las ventanas de tiempo. Los enfoques mejor valorados fueron; mientras más lejos, más importante y menos importante. Mientras que considerar la misma importancia para todos los escenarios demostró ser una mala estrategia para problemas robustos.

PALABRAS CLAVES: Asignación de recursos, Modelo de optimización, Sistema de Colonia de Hormigas

1. INTRODUCCIÓN

En el área de la optimización, los problemas combinatorios resultan cada vez más atrayentes para los investigadores y académicos, ya que cualquier contribución en este ámbito tiene repercusiones directas en la industria. La atención de los investigadores se ha centrado específicamente en los problemas de ruteo debido a la complejidad que presentan en la vida real. El problema del Viajante de Comercio o problema del Agente Viajero (Traveling Salesman Problem, TSP) ha sido un problema de optimización combinatoria extensamente estudiado por matemáticos, científicos informáticos y otros [1, 15]. En este problema se dispone de un agente viajero que debe visitar todos los clientes (nodos) en una sola ruta y a un costo mínimo. Este problema ha sido abordado desde la perspectiva de un problema estático donde las distancia entre los nodos no varía con el tiempo.
La optimización robusta determina un área de la optimización que se aplica a problemas donde existen más de un escenario y el objetivo se reduce a encontrar cual solución es la mejor evaluada para varios escenarios a la vez (ventana de tiempo). Esta área de la optimización se aplica como mecanismo en los problemas dinámicos [9, 10, 13]. Por su parte, los problemas de encontrar una ruta robusta ha sido interés de estudio de varios investigadores desde un enfoque estocástico, donde los modelos metaheurísticos [16] ha sido la estrategia predominante para resolver estos problemas por la complejidad que presenta.
En este trabajo presentamos un estudio teórico-práctico sobre el uso de la metaheurística ACO (Ant Colony Optimization) para resolver el problema de ruteo robusto desde una perspectiva dinámica. Analizaremos diferentes heurísticas con el objetivo de identificar cuál ajusta de mejor manera a la métrica de Robustez Promedio.

2. CONTEXTO DE LA INVESTIGACIÓN

Optimización Robusta

[1] apuris@uteq.edu.ec, bryan.cortez2015@uteq.edu.ec, edison.vicente2015@uteq.edu.ec, pavel.novoa@ucn.cl

La optimización robusta permite encontrar soluciones factibles sobre el curso del tiempo. Una solución robusta es aquella que no necesariamente es la mejor en el escenario actual, pero es aceptable en varios escenarios, estas soluciones pueden ser utilizadas hasta que su calidad disminuya a un nivel no aceptable. El proceso de encontrar soluciones robustas es llamado optimización robusta en el tiempo (ROOT) [12, 18] y es un campo de investigación relativamente nuevo, si lo analizamos desde la perspectiva de problemas de optimización dinámica (DOP), donde los escenarios cambian cada cierto tiempo.
Reducidas han sido las investigaciones reportadas en el ámbito de ROOT desde que se publicó el primer trabajo en el 2010 [18]. En este estudio los autores definieron el término ROOT como la búsqueda de la secuencia de soluciones robustas en un grupo de escenarios futuros. También construyeron problemas de prueba (funciones continuas dinámicas) para dicho problema.
No fue hasta el 2013 [8] cuando, los mismos autores reportaron evidencias de un marco de trabajo para resolver este tipo de problemas en su variante más compleja (problemas donde se desconocen los cambios en escenarios futuros) y la evaluación de una solución en el pasado está limitada). Para ello, definieron 4 componentes principales:

1. Optimizador: algoritmo que se encarga del proceso de búsqueda en el escenario actual
2. Memoria: forma de conocer las soluciones exploradas en cada escenario
3. Aproximador: modelo subrogado capaz de evaluar una solución actual en escenarios pasados
4. Predictor: modelo para estimar la calidad de una solución actual en escenarios del futuro

Este enfoque fue probado en 3 problemas continuos de estudio utilizando el algoritmo de Enjambre de Partículas (PSO) como optimizador, una Red de Funciones de base Radial (RBFN) como aproximador y como predictor del futuro un modelo Autorregresivo (AR). Como métricas para el estudio emplearon el error y la sensibilidad promedio. Los autores concluyeron con la necesidad de explorar más profundamente cada uno de los componentes del marco de trabajo y la definición de otras métricas más representativas en el contexto.
Fue en el mismo 2013 [4], donde se definen por primera vez las dos métricas más utilizadas en ROOT, Calidad Promedio (cp) y Tiempo de Supervivencia (ts) de una solución robusta. La primera establece el promedio de una solución actual en una cantidad de ambientes futuros consecutivos y el tiempo de supervivencia determina la cantidad de ambientes donde la solución actual sigue siendo buena (dependiendo de un umbral). La definición formal de estas métricas se describe a continuación en la ecuación 1 y 2 respectivamente:

$$\mathcal{R}^{cp}(x) = \frac{1}{T}\sum_{i=0}^{T-1} f_{t+i}(x) \tag{1}$$

$$\mathcal{R}^{ts}(x) = \begin{cases} 0, & f_t(x) < \varepsilon \\ 1 + \max\{l \mid \forall i \in \{t, t+1, \dots, t+l\}: f_i(x) \geq \varepsilon\}, & f_t(x) \geq \varepsilon \end{cases} \tag{2}$$

Donde *x* representa la solución actual que se desea evaluar, *T* la cantidad de ambientes futuros y $f_t(x)$ la función objetivo que caracteriza cada ambiente.
Lugo de estos trabajos se presentaron otros estudios encaminados a profundizar un poco más en el tema como, por ejemplo:

- En [7] se presentan 3 medidas de rendimiento para los algoritmos en ROOT, pero su aplicación se ve limitada porque es necesario conocer la solución óptima en cada uno de los ambientes. Esta característica muchas veces se aleja de la realidad.
- También en [6] se presenta un algoritmo multi-enjambre basado en PSO para el proceso de optimización local. Los problemas tratados no requirieron del marco de trabajo presentado en [8] debido a que se tuvo información de todos los ambientes. Las métricas utilizadas fueron la calidad promedio y el tiempo de supervivencia.
- Otro enfoque como la optimización multiobjetivo también fue probado en ROOT [17]. En este caso las propuestas definieron un proceso de búsqueda para optimizar varias métricas a la vez. Los resultados obtenidos fueron alentadores en el contexto de estudio.
- En el caso [11] los autores utilizan el marco de trabajo de [8] y estudian diferentes aproximadores del pasado para determinar la influencia de esta etapa en la búsqueda de una solución robusta. Se concluye que la calidad de las soluciones reportadas depende en gran medida del aproximador utilizado.

A manera de resumen se pudo identificar que las investigaciones en ROOT han estado vinculadas solamente a la solución de problemas continuos y de pocas dimensiones. Esta conclusión justifica el desarrollo de la presente investigación donde se pretende analizar un problema discreto desde un enfoque de ROOT.

Optimización basada en Colonia de Hormigas

Las técnicas de Optimización de Colonia de Hormigas (ACO) [3] se han posicionado como una metaheurística relativamente novel para los problemas complejos de optimización combinatoria. Su diseño está basado en las habilidades de las colonias de hormigas para determinar los caminos cortos hasta llegar a la comida. Las hormigas reales pueden comunicarse indirectamente por medio de feromonas sin la utilización de pistas visuales. Las hormigas artificiales imitan el comportamiento de las hormigas reales en su proceso de búsqueda de comida, pero pueden resolver problemas muchos más complejos que en la realidad [14]. En el proceso artificial, las hormigas se mueven de manera estocástica por los estados de un grafo G que representa el problema. Para el movimiento, las hormigas analizan dos componentes fundamentales; información específica del problema (distancia entre nodos, pesos de los arcos, ...) y la información histórica de la colonia (rastros de feromona). El marco general de trabajo de los algoritmos ACO supone los siguientes pasos:

1- Representación del problema en forma de grafo (nodos y arcos).
2- Representación de la matriz de feromona (información memorística de la colonia).
3- Definición de los parámetros; cantidad de hormigas (m), condición de parada de la búsqueda (sc), factor de influencia de la heurística (α) y de la feromona (β), porcentaje de feromona evaporada en cada iteración (ρ).
4- Construcción de soluciones: cada hormiga parte de un nodo del grafo y se mueve por el mismo, construyendo una solución de forma iterativa. Para elegir los posibles movimientos, las hormigas utilizan una función que relaciona la información del problema con los rastros de feromona.
5- Actualización de los rastros de feromona: este proceso se encarga de actualizar la memoria general de la colonia, de esta forma los trayectos mejor valorados serán más deseados por la colonia en el futuro.

6- Evaporación de la feromona: en esta etapa los algoritmos reducen los valores de feromona para evitar sobre aprendizaje y de esta forma disminuir el estancamiento de soluciones.

Entre los algoritmos ACO que existen, el Sistema de Colonias de Hormigas (Ant Colony System. ACS) [2] es uno de los más citados y trabajados en la literatura. Este algoritmo presenta una regla de movimiento pseudoaleatoria que permite un adecuado balance entre exploración e intensificación del espacio de búsqueda (ver Ecuaciones 3 y 4). Su proceso de actualización de los rastros de feromona también es diferente a las otras variantes de ACO. Esto se debe a que aplica dos procesos; la actualización en línea y una actualización general. El primer mecanismo aplica una función de actualización a cada paso que realiza la hormiga dentro del grafo, mientras que la actualización general, actualiza todos los pasos que realizó la hormiga mejor evaluada de la colonia hasta el momento.

$$\rho_k(r,u) = \begin{cases} \frac{[\tau(r,u)]\cdot[\eta(r,u)]^\beta}{\sum_{u\epsilon J_k(r)}[\tau(r,u)]\cdot[\eta(r,u)]^\beta}, & if\ u \in J_k(r) \\ 0 & ,Otro\ casos \end{cases} \quad (3)$$

$$s = \begin{cases} max_{u\epsilon J_k(r)}\{[\tau(r,u)]\cdot[\eta(r,u)]^\beta\}, \text{if } q \leq q0 & \text{(intensificación)} \\ Random(u) & \text{(exploración)} \end{cases} \quad (4)$$

donde $\tau(r,u)$ representa la cantidad de feromona presente en el arco (r,u) y $\eta(r,u)$ una función heurística relacionada con el objetivo del problema.

Viajante de Comercio

El Viajante de Comercio (Traveling Salesman Problem, (TSP)) [1] es uno de los problemas más utilizados en el campo de la optimización discreta. Su representación se realiza a partir de un grafo completo $G = (N, A)$, donde N representa la cantidad de nodos también llamados ciudades y A el conjunto de arcos bidireccionales que conectan los nodos. Cada arco a_{ij} se le asigna un valor d_i que representa la distancia entre las ciudades i y j. El objetivo de este problema radica en encontrar el camino (ciclo hamiltoniano) más corto desde una ciudad de partida, pasando por todas las demás ciudades solo una vez y regresando a la ciudad de origen. En el caso del TSP simétrico las distancias entre las ciudades son independientes de la dirección de la ruta ($d_{ij} = d_{ji}$). Para estudiar este problema, existe la biblioteca pública TSPLIB[2] con un gran número de instancias de diferentes dimensiones (cantidad de ciudades).

3. DISEÑO DE LA PROPUESTA

En esta sección detallaremos la construcción del caso de estudio a partir de TSP y definiremos las variantes heurísticas que probamos para el algoritmo ACS.

Caso de estudio

Para estudiar el problema del TSP como ROOT, seleccionamos la instancia Oliver30 de TSPLIB y generamos 4 instancias topológicamente iguales (mismo tamaño) pero con valores diferentes de distancias (generados aleatoriamente entre los valores máximo y mínimo de la instancia original). La Figura 1 muestra un extracto de los escenarios obtenidos.

[2] http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/

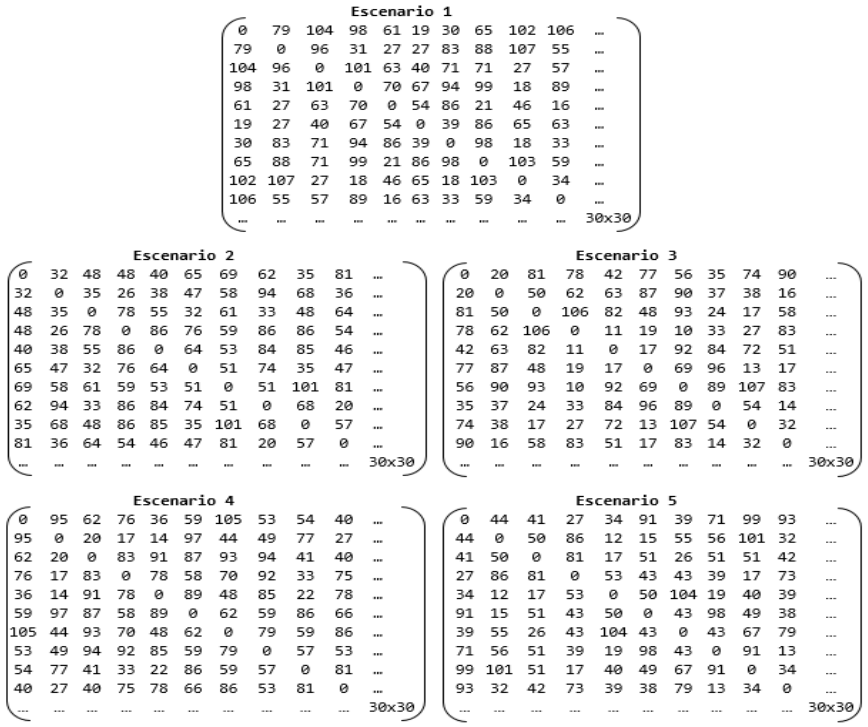

Figura 1. Escenarios para el estudio

La Tabla 1 presenta el diseño de los casos de estudio, donde $\mathbf{V_i}$ determina el tamaño de la ventana (cantidad de escenarios para analizar) y $\mathbf{E_j}$ el escenario de partida para el análisis. Se puede observar que partiendo de **E1** tenemos 4 casos de estudio (**V1**, **V2**, **V3** y **V4**), partiendo de **E2** tenemos 3 casos de estudio y así sucesivamente para **E3** y **E4.** En total contamos con 10 casos de estudio. El objetivo en cada caso es encontrar la solución que mejor robustez promedio presente.

Tabla 1. Estructura del problema de estudio.

	V1	**V2**	**V3**	**V4**
E1	E1|E2	E1|E2|E3	E1|E2|E3|E4	E1|E2|E3|E4|E5
E2	E2|E3	E2|E3|E4	E2|E3|E4|E5	-
E3	E3|E4	E3|E4|E5	-	-
E4	E4|E5	-	-	-

Funciones heurísticas

Nuestra propuesta de solución, utiliza el algoritmo ACS clásico presentado en [14] para resolver el TSP, pero adaptado a las condiciones de la variante ROOT. Esto supone la definición de nuevas funciones heurísticas que tiene en cuenta el trabajo con las ventanas de tiempo. A continuación, definimos con más detalles las funciones estudiadas bajo el siguiente enfoque general:

$$\eta^{V_k}(i,j) = 1 - \frac{1}{\sum_{t=1}^{T} w_t * \left(d_{ij}{}^{E_t}\right)}, E_t \in V_k,\ w_t \in v(0,1], T = dim(V_k) \quad (5)$$

Donde la función heurística $\eta^{V_k}(i,j)$ calcula la distancia $(d_{ij}{}^{E_t})$ ponderada (w) entre el nodo i al j en cada uno de los escenarios (E_t) de la ventana V_k y $dim(V_k)$ representa la cantidad de escenarios de dicha ventana.

Las ponderaciones utilizadas se presentan en las ecuaciones (6), (7), (8) y (9) estableciendo diferentes niveles de importancia de los escenarios; Balanceada (B), Descendente (DES), Ascendente (ASC) y Aleatorio (A) respectivamente.

$$w = \left[1, \frac{1}{2}, \dots, \frac{1}{T}\right]_T, T = dim(V_k) \quad (6)$$

$$w = \left[\frac{1}{T}, \frac{1}{T-1}, \dots, 1\right]_T, T = dim(V_k) \quad (7)$$

$$w = [1,1,\dots,1]_T \ , T = dim(V_k) \quad (8)$$

$$w = [rand(0,1],\dots,rand(0,1]]_T \ , T = dim(V_k) \quad (9)$$

4. RESULTADOS

Los experimentos se realizaron siguiendo un enfoque comparativo para determinar cuáles funciones de ponderación alcanza mejores valores de robustez promedio. Para ello, utilizamos la configuración de parámetros del algoritmo ACS definidos en [2]:

- $\beta = 2$: Importancia de la heurística en la búsqueda.
- $\alpha = 3$: Importancia de feromona.
- $\tau_0 = 0.1$: Feromona inicial.
- $\rho = 0.1$: Constante de evaporación
- $q_0 = 0.9$: Nivel de exploración del algoritmo

Se llevaron a cabo 25 ejecuciones independientes para caso de estudio y los valores reportados en la Figura 2 representan el promedio de los mejores resultados.

En la Figura 2 se puede apreciar que las heurísticas con ponderación ascendente y descendente obtienen los mejores resultados en todos los estudios realizados. Además, el desempeño de estos dos enfoques, no se ven afectados ni por el tamaño de la ventana ni por el escenario de partida. Para el caso de las heurísticas balanceada y aleatoria, se puede observar que el rendimiento disminuye en cada escenario a medida que aumenta el tamaño de ventana, siendo más enmarcada en la balanceada.

Por otra parte, la Figura 3 muestra un análisis de convergencia (identificando la mejor solución en cada iteración) sobre el mismo estudio. Los resultados revelan que solo la heurística aleatoria presenta un

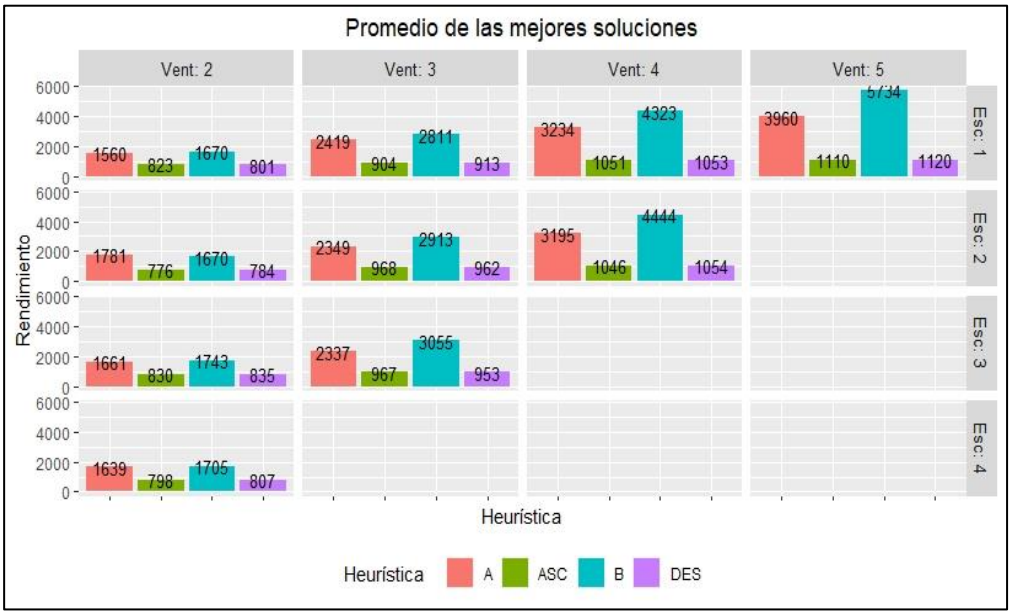

Figura 2: Desempeño de los enfoques utilizados

comportamiento irregular mientras que los otros modelos van mejorando a medida que aumentan las iteraciones. Algo que llama la atención es que, a pesar de la irregularidad en la convergencia de la ponderación aleatoria, sus resultados son mejores a la ponderación balanceada en la mayoría de los estudios.

A continuación, presentamos un análisis estadístico no paramétrico fundamentado en [5] con el objetivo de sustentar los criterios emitidos anteriormente sobre los resultados. Por una parte, aplicamos el test de Friedman para conocer la existencia de diferencias significativas en el grupo de resultados y luego un test de comparaciones múltiples de Holm para identificar entre que pares de resultados de encuentran las diferencias. Los resultados mostraron evidencia estadística favorable para el test de Friedman con un valor p =0.0124. Por su parte, el test de Holm determinó que trabajar con un enfoque ASC garantiza resultados significativamente

mejores (valor-p < Holm) que los enfoques A y B. En cuanto al enfoque DES, los resultados fueron similares (valor-p > Holm), como se resume en la Tabla 2.

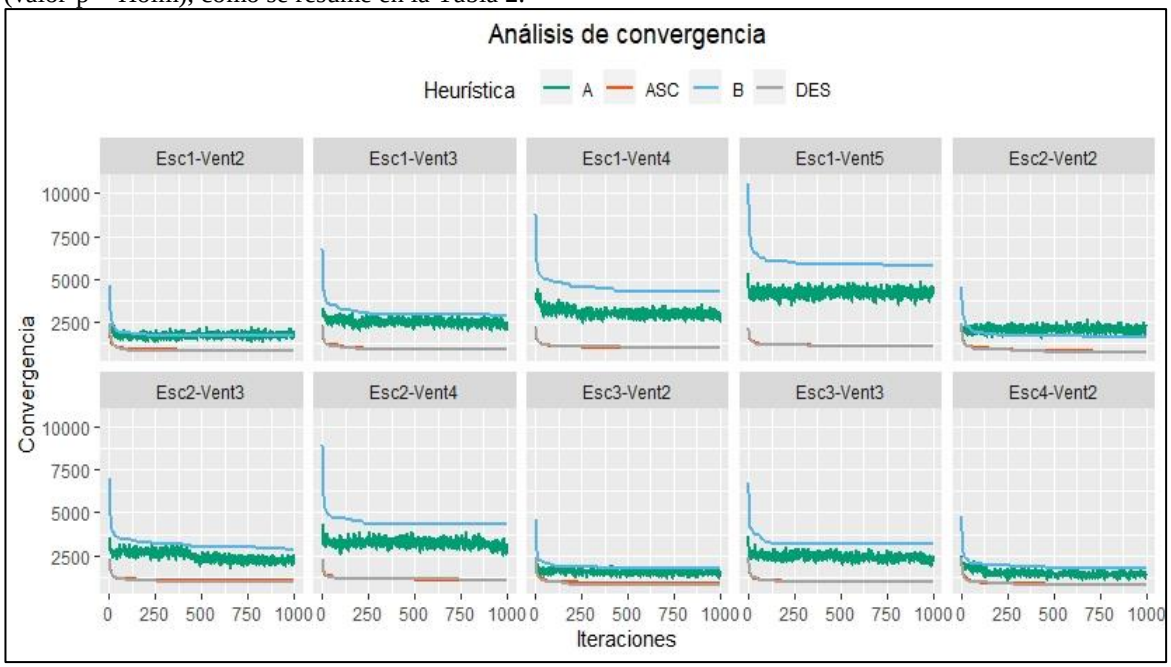

Figura 3: Convergencia de los enfoques utilizados

Tabla 2. Resultados del test de Holm.

ASC vs.	**valor-p**	**Holm**	**¿Se acepta la hipótesis nula de igualdad?**
B	0.0040	0.0166	**NO**
A	0.0220	0.0250	**NO**
6	0.7841	0.0500	**SÍ**

5. CONCLUSIONES

En este trabajo se presentó un estudio para aplicar el algoritmo Sistema de Colonia de Hormigas al problema de Viajante de Comercio desde un enfoque ROOT. Entre los principales aportes del trabajo se destacan los siguientes:

- Se utilizó la instancia Oliver30.tsp para construir un caso de estudio con 5 escenarios haciendo modificaciones aleatorias de las distancias entre las ciudades.
- Se diseñaron 4 funciones heurísticas para el algoritmo ACS, teniendo en cuenta los niveles de importancia de cada escenario en el proceso de búsqueda. Los enfoques fueron:
    - Ascendente (ASC): los escenarios más lejanos, son los más importantes.
    - Descendente (DES): los escenarios más lejos, son menos importantes.
    - Balanceado (B): todos los escenarios tienen el mismo grado de importancia.
    - Aleatorio (A): la importancia de un escenario es seleccionado al azar.

Los resultados experimentales mostraron que el algoritmo ACS alcanza su mejor rendimiento (robustez promedio) utilizando los enfoques controlados; ascendente y descendente. El enfoque aleatorio resultó ser muy inestable en el estudio de convergencia, aunque sus resultados fueron mejores que el enfoque balanceado en cuanto a rendimiento. Los peores resultados se alcanzaron con el enfoque balanceado, resultado que se debe de tener en cuenta a la hora de realizar otros estudios en el área de ROOT.
Como propósito inmediato, se pretende generalizar estos resultados en otros problemas de estudio tales como los problemas continuos en ROOT.

RECEIVED: JANUARY, 2021.
REVISED: JULY, 2021

REFERENCIAS

[1] APPLEGATE D.L, BIXBY R.E, CHVÁTAL V, and COOK W.J. (2011): **The Traveling Salesman Problem: A Computational Study**. Princeton Series in Applied Mathematics. Princeton University Press, Princeton

[2] DORIGO M, and GAMBARDELLA L. M. (1997): Ant colony system: a cooperative learning approach to the traveling salesman problem. **IEEE Transactions on Evolutionary Computation**, 1, 53–66.

[3] DORIGO M, and SOCHA K. (2007): **Ant colony optimization**. In Gonzalez, T. F., editor, Handbook of Approximation Algorithms and Metaheuristics, chapter 26-I, pages 101–104. Chapman& Hall/CRC Computer and Information Science Series, London.

[4] FU H, SENDHOFF B, TANG K, and YAO X. (2013): **Finding Robust Solutions to Dynamic Optimization Problems**. In: Esparcia-Alcázar AI, editor. Applications of Evolutionary Computation. Berlin, Heidelberg: Springer Berlin Heidelberg, 616–625.

[5] GARCÍA S, MOLINA D, LOZANO M, and HERRERA F. (2009): 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**. 15, 617–644.

[6] HUANG Y, DING Y, HAO K, and JIN Y. (2017): A multi-objective approach to robust optimization over time considering switching cost. **Information Sciences**, 394-395, 183–197.

[7] HUANG Y, JIN Y, and DING Y. (2015): **New performance indicators for robust optimization over time. In: 2015** IEEE Congress on Evolutionary Computation (CEC). 1380–1387.

[8] JIN Y, TANG K, YU X, SENDHOFF B, and YAO X. (2013): A framework for finding robust optimal solutions over time. **Memetic Comput**. 5, 3–18.

[9] NGUYEN TT, YANG S, and BRANKE J. (2012): Evolutionary dynamic optimization: A survey of the state of the art. **Swarm Evol. Comput**. 6, 1–24.

[10] NOVOA-HERNÁNDEZ P, CORONA C.C, and PELTA D.A (2016): Self-adaptation in dynamic environments - a survey and open issues. **International Journal of Bio-Inspired Computation**, 8, 1-13.

[11] NOVOA-HERNÁNDEZ P, PELTA D.A, and CORONA C.C. (2018): **Approximation Models in Robust Optimization over Time - An Experimental Study**. In: IEEE Congress on Evolutionary Computation, CEC 2018. 1339–1344.

[12] NOVOA-HERNÁNDEZ, P. and PURIS, A. (2019): Optimización robusta en el tiempo: una revisión delas contribuciones más relevantes. **Revista Ibérica de Sistemas e Tecnologias de Informação**, E18, 156–164.

[13] NOVOA-HERNÁNDEZ P, CORONA C.C, and PELTA, D.A (2015): A software tool for assisting experimentation in dynamic environments. **Applied Computational Intelligence and Soft Computing**, 2015, 1–12.

[14] PURIS A, BELLO R, and HERRERA F. (2010): Analysis of the efficacy of a Two-Stage methodology for ant colony optimization: Case of study with TSP and QAP. **Expert Systems with Applications,** 37, 5443–5453.

[15] PURIS, A. Y., NOVOA, P., and OVIEDO, B. (2020): **Desarrollo de meta-heurísticas poblacionales para la solución de problemas complejos**. Editorial Compás, Guayaquil.

[16] SOLANO-CHARRIS E, PRINS C, and SANTOS A.C. (2015): Local search based metaheuristics for the robust vehicle routing problem with discrete scenarios. **Applied Soft Computing**. 32, 518--531.

[17] YAZDANI D, NGUYEN T.T, and BRANKE J. (2019): Robust Optimization over Time by Learning Problem Space Characteristics. **IEEE Transactions on Evolutionary Computation,** 23, 143–155.

[18] YU X, JIN Y, TANG K, and YAO X. (2010): Robust optimization over time - **A new perspective on dynamic optimization problems.** In IEEE Congress on Evolutionary Computation, CEC 2010. 1–6.