

Tipo de artículo: Artículo original  
Temática: Inteligencia Artificial.  
Recibido: 17/02/2022 | Aceptado: 04/05/2023

## Integración de técnicas de visualización a la metaheurística Búsqueda Tabú

Integration of visualization techniques to the Tabu Search metaheuristic

Denis Rojas García <sup>1\*</sup> <https://orcid.org/0000-0002-3898-2690>

Andy Morfa Hernández <sup>1</sup> <https://orcid.org/0000-0002-2003-0974>

Carlos Pérez Risquet <sup>1</sup> <https://orcid.org/0000-0002-3547-3049>

<sup>1</sup>Centro de Investigaciones Informáticas. Universidad Central “Marta Abreu” de Las Villas, Carretera Camajuaní km 5 ½, Santa Clara, Villa Clara, Cuba. [drojas@uclv.cu](mailto:drojas@uclv.cu), [andym@uclv.cu](mailto:andym@uclv.cu), [cperez@uclv.edu.cu](mailto:cperez@uclv.edu.cu)

\*Autor para la correspondencia: [drojas@uclv.cu](mailto:drojas@uclv.cu)

---

### RESUMEN

Las técnicas de integración visual para apoyar algoritmos metaheurísticos en la resolución de problemas de optimización han sido muy relevantes durante los últimos tiempos. Estos algoritmos pueden ser aplicados a un amplio conjunto de problemas. El presente artículo propone la resolución del Problema del Viajero Vendedor y el Problema de Ruteo de Vehículos usando la metaheurística Búsqueda Tabú con el objetivo de optimizar los resultados, donde se ven reflejadas las técnicas de integración visual para ofrecer un mejor análisis e interacción del usuario. Todo ello propone un nuevo modelo de técnicas de integración visual el cual refleja notoriamente las interacciones entre el usuario y el algoritmo en tiempo de ejecución.

Basándonos en lo anterior se desarrolló un software para darle solución a estos problemas, al cual se le realizaron las pruebas correspondientes demostrando así la eficiencia de estas nuevas técnicas en el análisis del modelo propuesto con la asistencia del usuario.

**Palabras clave:** Búsqueda tabú; problema del viajero vendedor; problema de ruteo de vehículos; interacción; visualización.

## **ABSTRACT**

Visual integration techniques to support metaheuristic algorithms in solving optimization problems have been very relevant in recent times. These algorithms can be applied to a wide set of problems. This article proposes the resolution of the Travelling Salesman Problem and the Vehicle Routing Problem using the Tabu Search metaheuristic in order to optimize the results, where visual integration techniques are reflected to offer a better analysis and user interaction. All this proposes a new model of visual integration techniques which clearly reflects the interactions between the user and the algorithm at runtime. Based on the above, a software was developed to solve these problems, to which the corresponding tests were carried out, thus demonstrating the efficiency of these new techniques in the analysis of the proposed model with the assistance of the user.

**Keywords:** Tabu Search; travelling salesman problem; vehicle routing problem; interaction; visualization.

---

## **Introducción**

El dominio de la optimización y la inteligencia computacional es una de las partes inherentes y esenciales de la mayoría de las disciplinas asociadas con la utilización de los recursos al máximo. Es la necesidad de hoy y del futuro. En el corazón del dominio de optimización se encuentran el modelado matemático del problema y las metodologías de solución. Hoy en día, los problemas son cada vez más grandes con una

complejidad creciente. Dichos problemas se están volviendo engorrosos para ser manejados por métodos de optimización tradicionales. Esto motivó a los investigadores a recurrir a metodologías o algoritmos de solución inspirados en la naturaleza y basados en inteligencia artificial. Se les conoce comúnmente como metaheurísticas. Se prefieren estos algoritmos ya que se basan en reglas simples y pueden manejar grandes problemas complejos del mundo real en una cantidad de tiempo razonable. Se clasifican en algoritmos bioinspirados, basados en la física y socioinspirados y algoritmos basados en enjambres. (Kulkarni y Siarry, 2021)

Las heurísticas son técnicas diseñadas específicamente para resolver un problema en particular. Por el contrario, las metaheurísticas se definen como metodologías generales de nivel superior, que pueden utilizarse como estrategias orientadoras para el diseño de heurísticas subyacentes para la resolución de un problema. (Crawford et al., 2021)

Una Metaheurística, en tanto método de resolución para problemas complejos, puede ser entendida como una estrategia que combina diferentes Heurísticas, las cuales, individualmente, exhiben menor eficiencia o capacidad de solución para el problema abordado, tomando ventaja de sus habilidades parciales, así como de la estructura del propio problema a resolver, fundamentalmente de los aspectos que caracterizan el Espacio de Soluciones inherente al mismo (Schweickardt, 2018).

Entre los problemas que pueden ser estudiados en la optimización es la teoría de grafos ya que ofrece un marco de referencia acertado para iniciar la construcción de un algoritmo conveniente a la situación específica que se va a abordar. Entre ellos se encuentra el Problema del Viajero Vendedor (TSP) y Problema de Ruteo de Vehículos (VRP).

En general, las metaheurísticas toman inicialmente una solución factible, la cual es luego mejorada usando heurísticas de mejoramiento embebidas en una estructura más general. La característica común de estos enfoques es el uso de mecanismos para encontrar en lo posible soluciones óptimas evadiendo óptimos locales. (Tetzlaff et al., 2021)

Uno de los algoritmos metaheurísticos que se utilizan para este tipo de problemas es la Búsqueda Tabú (TS, Tabu Search). A pesar de que muchos de los softwares que son empleados por estas metaheurísticas generan buenas soluciones, no presentan una abundante interacción visual con el usuario.

El objetivo general fue la implementación de una herramienta para darle solución al Problema del Viajero Vendedor y al Problema de Ruteo de Vehículos usando la metaheurística Búsqueda Tabú mediante el manejo de interacciones visuales.

## Métodos o Metodología Computacional

El campo de la investigación metaheurística se vuelve cada vez menos transparente en términos de nuevos enfoques y extensiones de los algoritmos o marcos existentes. Esto hace que sea cada vez más difícil evaluar las características y capacidades de los algoritmos respectivos y, por lo tanto, la selección de un enfoque adecuado. Si bien este problema podría aliviarse mediante un análisis teórico extenso de las metaheurísticas, nuevamente se trata de una tarea difícil que requiere tiempo y experiencia. (Stegherr y Hähner, 2021)

La optimización viene dada por la reducción al mínimo de costo, tiempo, distancia y riesgo o la maximización de calidad, satisfacción y beneficios. Encontrar la mejor solución para determinados problemas de optimización con importancia científica o industrial deriva muchas veces en presentaciones computacionales que son de complejidad intratables. Sin embargo, el uso de algoritmos de aproximación es la principal alternativa para resolver esta clase de problemas ya que permite obtener una solución óptima o “casi” óptima en un tiempo razonable. (Alancay, Villagra y Villagra, 2016)

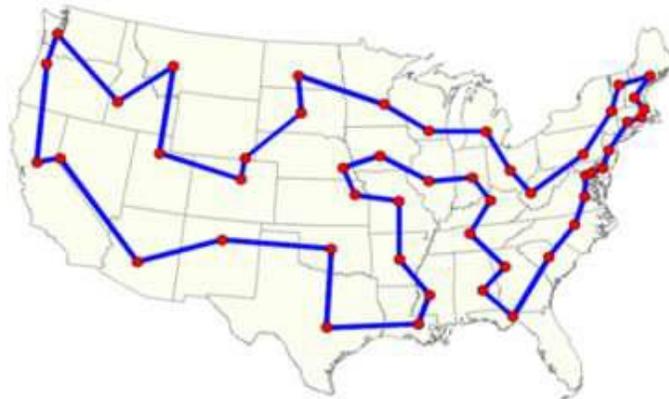
Debido a esta causa, en este tipo de escenario de complejidad, se muestran como una opción factible la usanza de metaheurísticas surtiendo un resultado posible que satisface todas las condiciones del problema. La mayoría de los problemas de optimización combinatoria son, en general, difíciles de resolver en la práctica. Estos problemas están incluidos en la clase de problemas NP-hard (Garey y Johnson, 1979), ya que no se conocen algoritmos exactos con complejidad polinómica que permitan resolverlos. (Alancay, Villagra y Villagra, 2016)

## Problema del Viajero Vendedor

El Problema del Viajero Vendedor es un problema popular y bien estudiado en el campo de la optimización combinatoria que llama la atención de científicos informáticos, matemáticos y otros expertos. Su enunciado parece sencillo, pero es uno de los problemas prácticos suficientemente difíciles de la investigación operativa. También es un problema de optimización en términos de determinar el recorrido cerrado más corto que visita todas las ciudades dadas. (Rufai et al., 2021)

El TSP se puede formular de la siguiente manera: “Dado un conjunto de ciudades, de las cuales se conoce para cada par de ellas, la distancia que las separa, un agente viajero ha de partir de una ciudad de origen y debe visitar exactamente una vez cada ciudad del conjunto, y retornar al punto de partida”. Un recorrido con estas características, es conocido dentro de este contexto, como un tour o ciclo hamiltoniano. El problema consiste en encontrar el tour para el cual la distancia total recorrida sea mínima. (Mendoza Casanova, 2017)

Según (Mendoza Casanova, 2017) el problema fue definido en el siglo XIX por el matemático irlandés W. R. Hamilton y por el matemático británico Thomas Kirkman y formulado por primera vez en 1930.



**Fig. 1** – Representación gráfica del TSP. (Rufai et al., 2021)

Se han utilizado muchas técnicas heurísticas para encontrar la solución eficiente al problema, como el método *greedy*, los algoritmos de hormigas, el recocido simulado, la búsqueda tabú y los algoritmos genéticos. Pero a medida que aumenta el número de ciudades, el cálculo para encontrar la solución se vuelve más difícil. A pesar de la dificultad computacional, podemos usar métodos como algoritmos genéticos y búsqueda tabú que pueden brindar una solución cercana a la óptima para miles de ciudades. (Dahiya y Sangwan, 2018)

### **Problema de Ruteo de Vehículos**

El Problema de Ruteo de Vehículos según (González La Rotta, González Yazo y Becerra Fernández, 2018) es problema de diseñar rutas para distribuir productos o servicios desde uno o varios depósitos a diferentes usuarios finales es uno de los más tratados dentro del campo de la investigación de operaciones. De manera unánime, los autores revisados plantean que la primera referencia del problema de ruteo de vehículos fue hecha por Dantzing y Ramser en 1959.

### **Capacitated Vehicle Routing Problem**

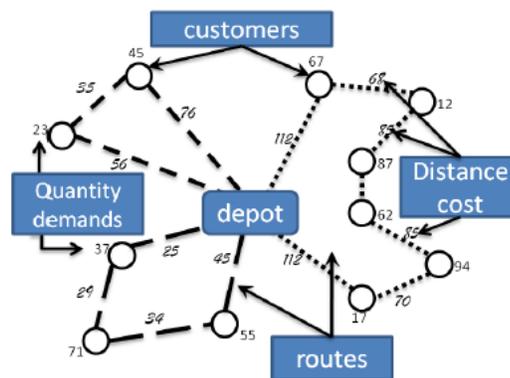
El CVRP es un problema NP-hard ampliamente discutido y, por lo tanto, métodos de optimización que incluyen métodos exactos (programación dinámica, ramificación y enlace); heurística (el algoritmo de Fisher-Jaikumar y el algoritmo de ahorro de Clarke-Wright); algoritmos de enjambre y evolutivos (algoritmos GA, ACO y luciérnaga); Los operadores de búsqueda locales (intercambio, inversión, codificación) y los enfoques híbridos se han empleado ampliamente para resolver el problema del CVRP. (Sajid et al., 2021)

Los clientes son atendidos por una flota de vehículos, cada uno de los cuales tiene una capacidad  $C$ . El principal objetivo en CVRP es encontrar el conjunto de rutas que minimizan el tiempo total de viaje de forma que: (Caballero Arnaldos, 2017):

1. Un cliente es servido una sola vez por un único vehículo
2. La ruta seguida por cada uno de los vehículos empieza y termina en el *depot*.

3. El total de demandas cubiertas por cada vehículo no puede exceder su capacidad C.

En la práctica, resolver el problema VRP es más difícil, ya que implica resolver dos problemas anidados. El primero es un problema de *bin-packing*, donde se busca asignar cada uno de los clientes a un número de rutas a priori indeterminado. A continuación, para cada ruta se busca el recorrido más corto, recorriendo cada uno de los clientes asignados a la ruta, lo que implica resolver el problema TSP. (Caballero Arnaldos, 2017)



**Fig. 2** – Representación gráfica del CVRP. (Konstantinidis, Pericleous y Charalambous, 2014)

### Algoritmo Greedy

Este nombre engloba a un conjunto de algoritmos con unas características específicas, el cual, sí que es reutilizable para algún otro problema distinto al TSP, aunque es en éste donde encuentra una solución más cercana a la óptima. La aplicación del algoritmo parte de la ordenación de los distintos nodos o ciudades a visitar, por pares de nodos a menor distancia. Siguiendo esta dinámica, el *greedy algorithm* siempre elige la mejor solución en cada iteración. Continuaríamos uniendo las ciudades, siempre buscando la menor distancia entre ellas, hasta que llegemos a un ciclo cerrado en el que se incluyan todas las ciudades del problema. (Gómez Lozano, 2021)

```
def Algoritmo_Voraz(Conjunto_Entrada):  
    Conjunto = Conjunto_Entrada  
    Solucion = []  
    encontrada = False  
    while not EsVacio(Conjunto) and not encontrada:  
        x = Seleccionar_Mejor_Candidato(Conjunto)  
        Conjunto = Conjunto - [x]  
        if EsFactible(Solucion + [x]):  
            Solucion = Solucion + [x]  
            if EsSolucion(Solucion):  
                encontrada = True  
    return Solucion
```

**Fig. 3** – Pseudocódigo del Algoritmo *Greedy* (Rodríguez Fuentes y Varona Fernández, 2015)

### **Algoritmo Búsqueda Tabú**

La Búsqueda Tabú (TS, *Tabu Search*) es un procedimiento heurístico de memorias adaptativa para la exploración de óptimos globales en problemas de optimización mono-objetivos. El método se basa en el principio que es mejor realizar un mal movimiento, cuando este es realizado de forma informada, que un buen movimiento ejecutado de forma aleatoria. Está siendo aplicado exitosamente en las últimas décadas en multitud de problemas complejos (continuos y discretos, lineales y no lineales convexos y no convexos, etc.)(Sicilia et al., 2016)

Búsqueda Tabú explora el espacio de soluciones a través de repetidos movimientos desde una solución a la mejor de sus vecinas tratando de evitar los óptimos locales. Para un problema mono-objetivo, TS realiza una búsqueda por entornos en la cual se desplaza en cada iteración a la mejor solución no tabú del vecindario de la solución actual. Los principales atributos de cada solución visitada son almacenados en una lista tabú por un determinado número de iteraciones para evitar que estas soluciones sean revisitadas, es decir, para evitar ciclos en la búsqueda por entornos. Así, un elemento del vecindario de la solución actual es declarado tabú (es decir, es prohibido) si alguno de sus atributos está en la lista tabú. En general, un método basado en Búsqueda Tabú requiere de los siguientes elementos: (Alancay, Villagra y Villagra, 2016):

1. Solución inicial.
2. Movimiento.
3. Vecindad o vecindario.
4. Lista tabú.
5. Criterio de parada.

```
1 solución inicial ← s0
2 tabuList ← ([], s0.length)
3 Repetir
4   solución vecina s ← s0
5   if(not tabuList.contains(s))
6   {
7     if(Target.maximize){
8       if ( f(s) >= f(s0) )
9         s0 = s
10    }else{
11      if( f(s) <= f(s0))
12        s0 = s
13    }
14    tabuList.grabarMovimientoMemoria(s);
15    if (tabuList.size() >= tabuList.getMaxListTabuSize())
16      tabuList.actualizarMemoria()
17  }
18 Hata condición de parada
19 return la mejor de todas s0 encontradas
```

**Fig. 4** – Pseudocódigo de la Búsqueda Tabú (Alancay, Villagra y Villagra, 2016)

## Visualización de Grafos

Uno de los campos de la Visualización de Información es la Visualización de Grafos que direcciona el problema de visualizar información estructural o relacional construyendo representaciones visuales geométricas de grafos o redes. Los grafos son el modelo subyacente de distintas aplicaciones tales como la Ingeniería de Software, el diseño de Interfaces Visuales, la Representación del conocimiento, las Telecomunicaciones y los Sistemas Distribuidos entre otros dominios.(Curino, Martig y Castro, 2014)

Un grafo  $G$  es un par de conjuntos  $G = (V, E)$ , tal que  $V$  es el conjunto de vértices, y  $E \subseteq V \times V$ , el de aristas. Cada arista, denotada  $(u, v) \in E$  se representa por medio de su par de vértices.(González Domínguez, 2021)

Entre las bondades de la visualización podemos mencionar que permite describir el comportamiento de los algoritmos en cada momento de su ejecución, el estado de cada variable, el espacio de búsqueda, con el fin de darle un tratamiento más directo al algoritmo y ser capaz de buscar comportamientos deseados o de localizar regiones de interés, e interactuar con la información proporcionada para encontrar soluciones más cercanas a las mejores y más rápidas. Todo esto basado en la capacidad del cerebro humano para analizar fácilmente imágenes visuales con mucha información.(Morfa Hernández et al., 2018)

### **Interacciones Visuales Implementadas**

1. Representar los nodos (ciudades/clientes) según las coordenadas extraídas del fichero del problema seleccionado.
2. Modificar los parámetros de ejecución del algoritmo Búsqueda Tabú antes de comenzar o luego de ser detenido por el usuario y permitiéndole al mismo probar las combinaciones de parámetros con los que se alcanzan mejores soluciones.
3. Encontrar una solución inicial eficiente por el *Greedy Algorithm* para el inicio del algoritmo Búsqueda Tabú.
4. Visualizar las rutas que se obtienen durante la ejecución del Búsqueda Tabú sobre el problema importado.
5. Detener el algoritmo Búsqueda Tabú en un instante determinado para hacer variaciones en los parámetros o en el área de representación.
6. Visualizar los resultados numéricos de las soluciones encontradas durante y finalizado la Búsqueda Tabú.
7. Añadir arista en determinadas situaciones, ya sea cuando se detiene el algoritmo o cuando genera una solución final.
8. Eliminar arista en determinadas situaciones, ya sea cuando se detiene el algoritmo o cuando genera una solución final.
9. Contabilizar el tiempo de la corrida del algoritmo Búsqueda Tabú para tener un balance del tiempo que se utilizó en encontrar una determinada solución.
10. Exportar imagen de la solución final encontrada.

11. Exportar fichero con los resultados de todas las ejecuciones que se hagan del algoritmo.
12. Hacer zoom sobre el área de representación para tener conocimiento de las cercanías de los nodos, a través de los botones que propone la interfaz o por la rueda del mouse para acercar y de la combinación de teclas Shift + Rueda del mouse para alejar.
13. Limpiar el área de representación para poder cargar otros ficheros.

## Resultados y discusión

El objetivo fundamental que presenta la realización del estudio es poder establecer una comparación entre los resultados oficiales registrados, disponibles en TSPLIB<sup>i</sup> y CVRPLIB<sup>ii</sup> y los alcanzados por TSVi, ejecutando el software desarrollado sin asistencia del usuario y con asistencia del mismo, a través de las interacciones implementadas, para valorar la efectividad y eficacia del sistema.

El sistema se ejecutó en un ordenador con las siguientes propiedades:

1. Procesador: Intel(R) Core (TM) i5-2520M CPU @ 2.50GHz.
2. Memoria: 4GB RAM.
3. Sistema Operativo: Windows 10 Pro 64 bits (10.0, compilación 18362.239)

Los parámetros necesarios para la ejecución de la Búsqueda Tabú son los siguientes:

1. Criterio de Diversificación (ficheros TSP): es el componente con el que dispone el algoritmo para salir de los mínimos locales porque alcanzado éste se demorará un tiempo en salir del mismo, realiza una solución aleatoria alejada de la solución actual. El criterio se establece en segundos.
2. Vecindario (ficheros TSP): contiene los tres movimientos principales para la ejecución del algoritmo.

1. *SwapMove*: consiste en el intercambio de dos ciudades en la trayectoria.

2. *InsertMove*: consiste en la inserción de una ciudad que se encuentra en una posición predeterminada a otra posición en el camino.
  3. *InvertMove*: consiste en invertir el orden de secuencia de las ciudades contenida en el camino.
3. Tamaño de la Lista Tabú: esta lista contiene los movimientos que se han hecho durante la ejecución del algoritmo. Estos movimientos están marcados como tabú y el tamaño de esta lista va a determinar significativamente el rendimiento del algoritmo para que pueda salir de los mínimos locales.

Para la ejecución sin asistencia del usuario del Problema del Viajero Vendedor, se tomaron 200 000 iteraciones sin pausa, con un criterio de diversificación de 1 segundo, el tamaño de la Lista Tabú de 5 y se seleccionó el Vecindario *InvertMove*, siempre partiendo de una solución inicial generada por el algoritmo *Greedy*. Además, se realizaron 10 ejecuciones con cada fichero para escoger el mejor resultado.

Para la corrida con asistencia del usuario se utilizaron 150 000 iteraciones en conjunto con la variación de los parámetros del criterio de diversificación, el tamaño de la Lista Tabú y la selección del Vecindario y se pudo pausar y continuar según las decisiones del usuario. Se realizaron 10 ejecuciones con cada fichero para escoger el mejor resultado.

**Tabla 1** – Ejecuciones sin asistir y asistidas por el usuario para los ficheros TSP.

Ficheros (.tsp)	TSPLIB	TSVis sin asistencia del usuario	Tiempo de Ejecución sin asistencia del usuario (s)	TSVis con asistencia del usuario	Tiempo de Ejecución con asistencia del usuario (s)
eil51	426.0	429.0	50.71	426.0	85.63
berlin52	7542.0	7734.0	51.451	7542.0	92.24
st70	675.0	680.0	93.09	675.0	134.45
eil76	538.0	549.0	116.68	541.0	231.14
kroA100	21282.0	21320.0	250.63	21282.0	349.81
ch150	6528.0	6727.0	779.76	6579.0	958.36
a280	2579.0	3877.0	4012.68	2654.0	5147.54

Para la ejecución sin asistencia del usuario del Problema de Ruteo de Vehículos, se tomaron 200 000 iteraciones sin pausa, con tamaño de la Lista Tabú de 3, siempre partiendo de una solución inicial generada por el algoritmo *Greedy*. Además, se realizaron 10 ejecuciones con cada fichero para escoger el mejor resultado.

Para la corrida con asistencia del usuario se utilizaron 150 000 iteraciones en conjunto con la variación de los parámetros del tamaño de la Lista Tabú y se pudo pausar y continuar según las decisiones del usuario. Se realizaron 10 ejecuciones con cada fichero para escoger el mejor resultado.

**Tabla 2** – Ejecuciones sin asistir y asistidas por el usuario para los ficheros VRP.

Ficheros (.vrp)	CVRPLIB	TSVis sin asistencia del usuario	Tiempo de ejecución sin asistencia del usuario (s)	TSVis con asistencia del usuario	Tiempo de ejecución con asistencia del usuario (s)
P-n45-k5	510	586.48	23.516	523.51	45.14
A-n53-k7	1010	1232.93	35.581	1080.15	54.68
A-n69-k9	1159	1288.19	45.865	1194.88	68.24
B-n78-k10	1221	1425.73	64.077	1399.21	86.19
E-n101-k8	815	932.94	96.614	887.00	124.80
M-n121-k7	1034	1214.24	132.684	1164.81	164.29

Para la realización del análisis comparativo se utilizó la prueba de los rangos con signo de Wilcoxon. Según (López Jiménez, Barrera Garrido y Bécquer Rodríguez, 2018) es una prueba no paramétrica para comparar la mediana de dos muestras relacionadas y determinar si existen diferencias entre ellas.

Se usa para verificar la H0 de igualdad entre 2 medianas poblacionales, la variable debe ser continua y observaciones emparejadas; es decir, datos de la misma muestra con medición de pre y post prueba (Ramírez Ríos y Polack Peña, 2019).

Según (Demsar, 2006) los problemas con las pruebas de conjuntos de datos múltiples son bastante diferentes, incluso en un sentido complementario: las mediciones de diferentes conjuntos de datos suelen ser inconmensurables, y la normalidad de sus distribuciones y la homogeneidad de la varianza es cuestionable en el mejor de los casos. A través del software *IBM SPSS Statistics Visor* se llevó a cabo este análisis.

Hipótesis definidas para la realización de la prueba:

1. H0 (hipótesis nula): La asistencia del usuario no influye en el resultado del algoritmo.
2. H1 (hipótesis alterna): La asistencia del usuario influye en el resultado del algoritmo.

Los resultados obtenidos se ven reflejados en las siguientes tablas:

		N	Rango promedio	Suma de rangos
TSVisConAsistencia - TSVisSinAsistencia	Rangos negativos	7 <sup>a</sup>	4,00	28,00
	Rangos positivos	0 <sup>b</sup>	,00	,00
	Empates	0 <sup>c</sup>		
	Total	7		

- a. TSVisConAsistencia < TSVisSinAsistencia  
 b. TSVisConAsistencia > TSVisSinAsistencia  
 c. TSVisConAsistencia = TSVisSinAsistencia

**Estadísticos de contraste<sup>a</sup>**

	TSVisConAsistencia - TSVisSinAsistencia
Z	-2,366 <sup>b</sup>
Sig. asintót. (bilateral)	,018

- a. Prueba de los rangos con signo de Wilcoxon  
 b. Basado en los rangos positivos.

**Fig. 5** – Prueba de Wilcoxon para datasets de tipo TSP

**Rangos**

		N	Rango promedio	Suma de rangos
TSVisConAsistencia - TSVisSinAsistencia	Rangos negativos	6 <sup>a</sup>	3,50	21,00
	Rangos positivos	0 <sup>b</sup>	,00	,00
	Empates	0 <sup>c</sup>		
	Total	6		

- a. TSVisConAsistencia < TSVisSinAsistencia  
 b. TSVisConAsistencia > TSVisSinAsistencia  
 c. TSVisConAsistencia = TSVisSinAsistencia

**Estadísticos de contraste<sup>a</sup>**

	TSVisConAsistencia - TSVisSinAsistencia
Z	-2,201 <sup>b</sup>
Sig. asintót. (bilateral)	,028

- a. Prueba de los rangos con signo de Wilcoxon  
 b. Basado en los rangos positivos.

**Fig. 6** – Prueba de Wilcoxon para datasets de tipo VRP

Luego de la realización de la prueba en ambos resultados se obtuvo como Significación asintótica bilateral los valores 0,018 y 0,028 respectivamente. Por tanto, como el valor de la Significación asintótica bilateral es menor que 0,05, podemos afirmar que se rechaza la hipótesis nula (H0) y se acepta la hipótesis alterna (H1): la asistencia del usuario influye en el resultado del algoritmo.

## Conclusiones

En el transcurso de la presente investigación se realizó un análisis través del uso de la metaheurística Búsqueda Tabú a los problemas de optimización combinatoria de TSP y VRP. Se logró la implementación de un programa de cómputo en la cual se integraron diferentes interacciones y visualizaciones que ayudan al usuario a lograr una mejor experiencia en la obtención de los resultados.

La realización del estudio experimental demostró que cuando el usuario interactúa con el algoritmo en tiempo de ejecución se logran mejores soluciones y en varias ocasiones se acercan a las registradas como óptimas en periodos de tiempo aceptables. Estos resultados pueden variar en dependencia de la habilidad y el conocimiento que posea el usuario. Se recomienda la continuación de su estudio debido a la importancia del uso de la integración de las técnicas de visualización en la solución de problemas combinatorios.

## Referencias

- Alancay, N., Villagra, S. Y Villagra, A., 2016. *Algoritmos Metaheurísticos Trayectoriales Para Optimizar Problemas Combinatorios*. 2016. Universidad Nacional De La Patagonia Austral - Unidad Académica Caleta Olivia: S.N.
- Caballero Arnaldos, L., 2017. *Diseño Y Evaluación De Algoritmos Para Vrp*. S.L.: Escuela Técnica Superior De Ingeniería De Telecomunicación.
- Crawford, B., Soto, R., Lemus-Romani, J., Becerra-Rozas, M., Lanza-Gutiérrez, J.M., Caballé, N., Castillo, M., Tapia, D., Cisternas-Caneo, F., García, J., Astorga, G., Castro, C. Y Rubio, J.M., 2021. Q-Learnheuristics: Towards Data-Driven Balanced Metaheuristics. *Mathematics*, Vol. 9, No. 16, Pp. 1-26. Issn 22277390. Doi 10.3390/Math9161839.
- Curino, R., Martig, S. Y Castro, S., 2014. *Visualización De Grafos*. . Buenos Aires:
- Dahiya, C. Y Sangwan, S., 2018. Literature Review On Travelling Salesman Problem. *International Journal Of Research* [En Línea], Vol. 05, No. 16, Pp. 5. Disponible En: [Https://edupediapublications.org/journals](https://edupediapublications.org/journals).
- Demsar, J., 2006. Statistical Comparisons Of Classifiers Over Multiple Data Sets. *Journal Of Machine Learning Research*, Vol. 7, Pp. 1-30.
- Garey, M. Y Johnson, D., 1979. *Computers And Intractability: A Guide To The Theory Of Np-Completeness*. 1979. New York: W.H. Freeman.
- Gómez Lozano, F.J., 2021. *Coordinación Y Dimensionamiento De Flota De Drones En Problemas Tsp A Través De Python*. S.L.: Universidad De Sevilla.
- González Domínguez, F., 2021. *Algoritmo De Enumeración De Cliques Maximales En Grafos K - Outerplanares , Usando Descomposición De Árbol*. S.L.: Centro De Investigación Científica Y De Educación Superior De Ensenada.
- González La Rotta, E.C., González Yazo, O. Y Becerra Fernández, M., 2018. Estado Del Arte Del Problema De Ruteo De Vehículos Con Componentes Estocásticos. , Vol. 13, Pp. 2-14. Doi

10.26620/Uniminuto.Inventum.13.24.2018.2-14.

Konstantinidis, A., Pericleous, S. Y Charalambous, C., 2014. Adaptive Evolutionary Algorithm For A Multi-Objective Vrp. , No. September.

Kulkarni, A.J. Y Siarry, P., 2021. *Handbook Of Ai-Based Metaheuristics*. First Edit. S.L.: Taylor & Francis Group. Isbn 9781003162841.

López Jiménez, O., Barrera Garrido, G. De La C. Y Bécquer Rodríguez, C.R., 2018. Comparación De Clasificadores Sobre Múltiples Datasets Con Pruebas Estadísticas No Paramétricas Comparison. *Spectrochimica Acta - Part A: Molecular And Biomolecular Spectroscopy*, Vol. 192, No. 4, Pp. 64-82. Issn 00162361.

Mendoza Casanova, J.J., 2017. *Diseño De Algoritmos Heurísticos Y Metaheurísticos Eficientes Para Resolver El Problema Del Agente Viajero*. 2017. Managua: Universidad Nacional Autónoma De Nicaragua.

Morfa Hernández, A., Oves García, R., Vázquez Rodríguez, R. Y Pérez Risquet, C., 2018. Integration Of Visualization Techniques To Algorithms Of Optimization Of The Metaheuristics Ant Colony. *Computación Y Sistemas*, Vol. 22, No. 1, Pp. 215–222. Doi 10.13053/Cys-22-1-2769.

Ramírez Ríos, A. Y Polack Peña, A.M., 2019. Estadística Inferencial. Elección De Una Prueba Estadística No Paramétrica En Investigación Científica. *Horizonte De La Ciencia*, Pp. 18. Doi <https://doi.org/10.26490/Uncp.Horizonteciencia.2020.19.597>.

Rodríguez Fuentes, L.J. Y Varona Fernández, A., 2015. *Técnicas De Diseño De Algoritmos. Algoritmos Voraces*. 2015. Universidad Del País Vasco: S.N.

Rufai, K.I., Usman, O.L., Olusanya, O.O. Y Adedeji, O.B., 2021. Solving Travelling Salesman Problem Using An Improved Ant Colony Optimization Algorithm. *Journal Of Science And Logics In Ict Research*,

Sajid, M., Singh, J., Haidri, R.A., Prasad, M., Varadarajan, V., Kotecha, K. Y Garg, D., 2021. A Novel Algorithm For Capacitated Vehicle Routing Problem For Smart Cities. *Symmetry*, Vol. 13, No. 10, Pp. 1923. Issn 20738994. Doi 10.3390/Sym13101923.

Schweickardt, G.A., 2018. Optimización De La Inyección De Potencia Desde Micro Generación Distribuida

En Sistemas Eléctricos De Distribución De Baja Tensión. Desarrollos Teóricos De Un Modelo Hiperheurístico. *Lámpsakos*, No. 20, Pp. 55-67. Issn 2145-4086. Doi 10.21501/21454086.3015.

Sicilia, J.A., Quemada, C., Royo, B. Y Escuín, D., 2016. An Optimization Algorithm For Solving The Rich Vehicle Routing Problem Based On Variable Neighborhood Search And Tabu Search Metaheuristics. *Journal Of Computational And Applied Mathematics* [En Línea], Vol. 291, Pp. 468-477. [Consulta: 17 Diciembre 2019]. Issn 0377-0427. Doi 10.1016/J.Cam.2015.03.050. Disponible En: <https://www.sciencedirect.com/science/article/abs/pii/S0377042715002034>.

Stegherr, H. Y Hähner, J., 2021. Analysing Metaheuristic Components. *Ceur Workshop Proceedings*, Vol. 3007, Pp. 5.

Tetzlaff, T., Gaudiani, A., Rojas Paredes, A., Encinas, D., Fassio, E., Trigila, M., González, R. Y Bertaccini, D., 2021. Metaheurísticas , Búsqueda Estocástica Y Cómputo Eficiente En Optimización Aplicada. *Redunci - Undec*, Pp. 124-128.

### **Conflicto de interés**

Los autores de este trabajo remitimos el manuscrito titulado «Integración de técnicas de visualización en el Problema del Viajero Vendedor y el Problema de Ruteo de Vehículos usando Búsqueda Tabú» y se autoriza la distribución y uso del mismo.

### **Contribuciones de los autores**

1. Conceptualización: Denis Rojas García, Andy Morfa Hernández, Carlos Pérez Risquet
2. Curación de datos: Denis Rojas García
3. Análisis formal: Denis Rojas García, Andy Morfa Hernández
4. Adquisición de fondos: No aplica
5. Investigación: Denis Rojas García
6. Metodología: Denis Rojas García
7. Administración del proyecto: Denis Rojas García, Andy Morfa Hernández, Carlos Pérez Risquet

8. Recursos: Denis Rojas García, Andy Morfa Hernández
9. Software: Denis Rojas García
10. Supervisión: Andy Morfa Hernández, Carlos Pérez Risquet
11. Validación: Denis Rojas García, Andy Morfa Hernández, Carlos Pérez Risquet
12. Visualización: Denis Rojas García, Andy Morfa Hernández
13. Redacción – borrador original: Denis Rojas García, Andy Morfa Hernández, Carlos Pérez Risquet
14. Redacción – revisión y edición: Denis Rojas García, Andy Morfa Hernández

### **Financiación**

El trabajo no requirió financiación.

---

<sup>i</sup> <http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsplib.html>

<sup>ii</sup> <http://vrp.atd-lab.inf.puc-rio.br/index.php/en/>