Búsqueda de soluciones en problemas con múltiples objetivos.

Análisis comparativo entre SPEA2 y MOACO para el TSP en dos variables

Betancur Cervantes Fabian

Blanco Baines Daniel

Triana Gogué Carlos

****

Universidad del Norte

División de Ingenierías

Curso de Optimización

Barranquilla

2015

Proyecto Final

Presentado a:

Ing. Carlos Julio Ardila Hernández.

Tabla de contenido

[Agradecimientos 6](#_Toc436262722)

[Resumen 7](#_Toc436262723)

[Introducción. 8](#_Toc436262724)

[Planteamiento el problema. 8](#_Toc436262725)

[Objetivos. 8](#_Toc436262726)

[Antecedentes. 8](#_Toc436262727)

[Contribución. 9](#_Toc436262728)

[Marco Teórico. 10](#_Toc436262729)

[Definiciones. 10](#_Toc436262730)

[Diseño de las soluciones. 11](#_Toc436262731)

[SPEA2. 11](#_Toc436262732)

[MOACO. 11](#_Toc436262733)

[Experimentación y Análisis de resultados. 12](#_Toc436262734)

[Comparación de Resultados. 13](#_Toc436262735)

[Conclusiones y Trabajos Futuros. 14](#_Toc436262736)

[Bibliografía y Referencias. 15](#_Toc436262737)

# Agradecimientos

# Resumen

# Introducción.

## Planteamiento el problema.

EL Problema del Agente Viajero (Traveler Salesman Problem), responde a la siguiente pregunta: Dada una lista de ciudades y las distancias entre cada par de ellas, ¿cuál es la ruta más corta posible que visita cada ciudad exactamente una vez y regresa a la ciudad origen? Este es un problema NP-Hard dentro en la optimización combinatoria, muy importante en la investigación de operaciones y en la ciencia de la computación.

En la forma tradicional en cómo se trabaja el TSP solo se tiene en cuenta un factor de decisión para el problema, ya sea tiempo, distancia o costo del viaje, sin embargo en un enfoque más realista del realista del problema es posible que se esté interesado en encontrar caminos (soluciones) que permitan un me mejoramiento en dos o más factores con tal de obtener mayores benefician o un balanceo de las cargas con respecto a las variables de decisión, por tanto el TSP pasa a ser un problema multi-objetivo en donde es posible que no se encuentre una solución única que satisfaga con la minimización de todas los objetivos.

## Objetivos.

Para este caso específico se tomas como objetivos minimizar los tiempos y las distancias, y como se mencionó anteriormente la búsqueda tiende a generar conjuntos de soluciones, ya que una solución óptima en tiempo puede no serlo en distancia y viceversa, por tanto se pueden desechar o parcializar las soluciones, lo cual termina por general un conjunto en el cual aunque no sean totalmente óptimos para alguno de los objetivos tampoco se pueden catalogar como peores, ya que el sacrifico o perdida en alguno de los objetivos genera una ganancia en el otro, creando como resultado un estrecha relación entre ambos objetivos en lo que se puede catalogar como una relación costo beneficio

## Antecedentes y Estado del Arte.

Los problemas NP-Hard desde su planteamiento y definición han sido fuertemente estudiados generan una variedad de soluciones bastante interesantes en busca de las mejoras en tiempo de ejecución y conservación de la calidad de las respuestas.

Uno de los avances más significativos en este campo ha sido el desarrollo de heurísticas para obtener soluciones muy aproximadas con bajo costo de recursos computacionales. Posteriormente, en la búsqueda soluciones más cercanas al optimo real, se desarrollaron las híper-heurísticas y las meta-heurísticas, las cuales lo lograron desbancar por completo a las heurísticas tradicionales, ya que aunque las respuestas generadas son mejores su coste de recursos de ejecución es mayor, por tal motivo un buen punto de inicio para inicializar una meta-heurística es pasarle los valores obtenidos previamente con una heurística.

# Marco Teórico.

## Definiciones.

### Heurística.

En computación, dos objetivos fundamentales son encontrar algoritmos con buenos tiempos de ejecución y buenas soluciones, usualmente las óptimas. Una heurística es un algoritmo que abandona uno o ambos objetivos; por ejemplo, normalmente encuentran buenas soluciones, aunque no hay pruebas de que la solución no pueda ser arbitrariamente errónea en algunos casos; o se ejecuta razonablemente rápido, aunque no existe tampoco prueba de que siempre será así. Las heurísticas generalmente son usadas cuando no existe una solución óptima bajo las restricciones dadas (tiempo, espacio, etc.), o cuando no existe del todo.

A menudo, pueden encontrarse instancias concretas del problema donde la heurística producirá resultados muy malos o se ejecutará muy lentamente. Aun así, estas instancias concretas pueden ser ignoradas porque no deberían ocurrir nunca en la práctica por ser de origen teórico. Por tanto, el uso de heurísticas es muy común en el mundo real.

### Meta-Heurística.

Para problemas de búsqueda del camino más corto el término tiene un significado más específico. En este caso una heurística es una función matemática, h(n) definida en los nodos de un árbol de búsqueda , que sirve como una estimación del coste del camino más económico de un nodo dado hasta el nodo objetivo. Las heurísticas se usan en los algoritmos de búsqueda informada como la búsqueda egoísta. La búsqueda egoísta escogerá el nodo que tiene el valor más bajo en la función heurística. A\* expandirá los nodos que tienen el valor más bajo para g(n)+h(n) , donde g(n) es el coste (exacto) del camino desde el estado inicial al nodo actual. Cuando h(n) es admisible, esto es si h(n) nunca sobrestima los costes de encontrar el objetivo; A\* es probablemente óptimo.

Un problema clásico que usa heurísticas es el puzzle-n. Contar el número de casillas mal colocadas y encontrar la suma de la distancia Manhattan entre cada bloque y su posición al objetivo son heurísticas usadas a menudo para este problema. Se realiza a partir de la categoría gramatical.

En cualquier problema de búsqueda donde hay b opciones en cada nodo y una profundidad d al nodo objetivo, un algoritmo de búsqueda ingenuo deberá buscar potencialmente entre b^d nodos antes de encontrar la solución. Las heurísticas mejoran la eficiencia de los algoritmos de búsqueda reduciendo el factor de ramificación de b a (idealmente) una constante b\*.

Aunque cualquier heurística admisible devolverá una respuesta óptima, una heurística que devuelve un factor de ramificación más bajo es computacionalmente más eficiente para el problema en particular. Puede demostrarse que una heurística h\_2(n) es mejor que otra h\_1(n), si h\_2(n) domina h\_1(n), esto quiere decir que h\_2(n) < h\_1(n) para todo n.

### Algoritmo Evolutivo.

Los algoritmos evolutivos son métodos de optimización y búsqueda de soluciones basados en los postulados de la evolución biológica. En ellos se mantiene un conjunto de entidades que representan posibles soluciones, las cuales se mezclan, y compiten entre sí, de tal manera que las más aptas son capaces de prevalecer a lo largo del tiempo, evolucionando hacia mejores soluciones cada vez.

Los algoritmos evolutivos, y la computación evolutiva, son una rama de la inteligencia artificial. Son utilizados principalmente en problemas con espacios de búsqueda extensos y no lineales, en donde otros métodos no son capaces de encontrar soluciones en un tiempo razonable.

Siguiendo la terminología de la teoría de la evolución, las entidades que representan las soluciones al problema se denominan individuos o cromosomas, y el conjunto de éstos, población. Los individuos son modificados por operadores genéticos, principalmente el cruce, que consiste en la mezcla de la información de dos o más individuos; la mutación, que es un cambio aleatorio en los individuos; y la selección, consistente en la elección de los individuos que sobrevivirán y conformarán la siguiente generación. Dado que los individuos que representan las soluciones más adecuadas al problema tienen más posibilidades de sobrevivir, la población va mejorando gradualmente.

# Diseño de las soluciones.

## SPEA2.

Se basa en la identificación de las soluciones no dominadas y en función del número de soluciones a las que dominan y por las que son dominadas se les asigna una calidad que es utilizada en torneo binario para obtener el conjunto de padres. Las soluciones no dominadas encontradas a lo largo de la búsqueda son almacenadas en un conjunto exterior (Archivo) de tamaño fijo igual al de la población.

Al final de cada iteración se unen la población recién creada de hijos P con el conjunto de archivo para actualizar el nuevo conjunto de archivo que es el conjunto sobre el que se hace la selección para obtener a los padres de la siguiente generación. El sistema evalúa a cada solución de la unión (P U ). En un primer paso se calcula para cada solución de la unión de conjuntos el número de soluciones de (P U ) a las que la solución i domina (nci). La calidad inicial de la solución j es la suma de los valores nci de las soluciones i por las que j es dominada. Por lo tanto, las soluciones no dominadas que pertenecen a la Frontera tienen una calidad inicial de 0. A esta calidad inicial de cada solución se le añade una cantidad extra relacionada con la distancia al resto de soluciones (denominada distancia k-th), de tal forma que si la solución pertenece a una zona densamente poblada en el espacio de objetivos, esta parte de la calidad estará penalizada asignándola un valor más alto que si la solución estuviera aislada. Este mecanismo es el proceso de diversificación en el espacio de objetivos que el SPEA2 implementa para formar FP uniformemente distribuidas.

De la unión (P U ) se seleccionan a los individuos no dominados. Si el número de no dominados es inferior al tamaño del conjunto de archivo (N del mismo tamaño que la población de hijos) todas las soluciones no dominadas sobreviven y son almacenadas en el nuevo conjunto de archivo. El resto de puestos del conjunto de archivo no ocupados se rellenan con las mejores soluciones dominadas (según la calidad descrita anteriormente). Si por el contrario el número es mayor, se realiza un truncado, que consiste en ir eliminando las soluciones que tengan la menor distancia a su individuo más cercano, hasta quedarnos con las N soluciones no dominadas con las que se rellena el

conjunto de archivo. Una vez actualizado el nuevo conjunto de archivo se realiza el torneo binario por el que se obtiene a los padres de la siguiente generación. El torneo lo ganan los individuos de menor calidad. Se realiza el proceso de reproducción sobre el conjunto de padres para obtener el nuevo conjunto de hijos que sirve para volver a actualizar el conjunto de archivo y así cerrar el ciclo del proceso evolutivo.

## MOACO.

# Experimentación y Análisis de resultados.

# Comparación de Resultados.

# Conclusiones y Trabajos Futuros.

# Bibliografía y Referencias.