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

[Introducción. 6](#_Toc436295146)

[Planteamiento el problema. 6](#_Toc436295147)

[Objetivos. 6](#_Toc436295148)

[Antecedentes y Estado del Arte. 7](#_Toc436295149)

[Marco Teórico. 7](#_Toc436295150)

[Definiciones. 7](#_Toc436295151)

[Heurística. 7](#_Toc436295152)

[Meta-Heurística. 8](#_Toc436295153)

[Algoritmo Evolutivo. 9](#_Toc436295154)

[Diseño de las soluciones. 11](#_Toc436295155)

[SPEA2. 11](#_Toc436295156)

[MOACO. 12](#_Toc436295157)

[Conclusiones y Trabajos Futuros. 14](#_Toc436295158)

[Bibliografía 15](#_Toc436295159)

# 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 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. 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.

### Meta-Heurística.

Las meta-heurísticas generalmente se aplican a problemas que no tienen un algoritmo o heurística específica que dé una solución satisfactoria; o bien cuando no es posible implementar ese método óptimo. La mayoría de las meta-heurística tienen como objetivo los problemas de optimización combinatoria, pero por supuesto, se pueden aplicar a cualquier problema que se pueda reformular en términos heurísticos-

Las meta-heurísticas no son la panacea y suelen ser menos eficientes que las heurísticas específicas, en varios órdenes de magnitud, en problemas que aceptan este tipo de heurísticas puras.

### 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 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.

# 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.

En un primer paso se calcula para cada solución de la unión de conjuntos el número de soluciones 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 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.

La Optimización por Colonia de Hormigas Multi-Objetivo (MOACO por sus siglas en inglés) se basa en un modelo de hallazgo de rutas implementado usualmente por las colonias de hormigas en la que la posibilidad de una hormiga para recorrer una ruta será función de una heurística que determina si en algún momento del tiempo otra hormiga pasó por la misma ruta, dejando un rastro de feromona que la hormiga siguiente a pasar usará como guía, opcionalmente.

Usualmente en el problema del Agente Viajero, las probabilidades de pasar de un nodo a otro usando colonia de hormigas se basan en una probabilidad dependiente de la cantidad de feromonas dejadas en la ruta del nodo a seguir, y también por una heurística con un cierto grado de influencia, dicha heurística puede ser en función del tiempo de desplazamiento de nodo a nodo, o también la distancia euclidiana entre dos nodos. Para el problema del MOACO multi-objetivo se debe aplicar una nueva probabilidad que permita obtener resultados basados tanto en la cantidad de feromonas, propio del algoritmo de Colonia de Hormigas, pero también que se obtenga en función de las dos heurísticas, o variables de decisión que se quieren obtener para este laboratorio, es decir, tanto distancia como tiempo.

Para este algoritmo se define iteración como el camino obtenido por una hormiga desde su partida desde un nodo origen hasta llegar a un destino y regresar al nodo origen.

Como parámetros iniciales se definen una matriz de feromonas donde se coloca un rastro inicial para comenzar la búsqueda y se definen el grado de influencia que tienen las feromonas, distancia o tiempo, en la escogencia de una solución.

Inicialmente, se establece un rastro inicial de feromonas no significativo con el objetivo de que no se obtengan probabilidades iguales a 0 y se inicializa un camino de hormigas que hallará una ruta inicial de manera aleatoria, dejando a su paso un rastro de feromonas que será factor decisivo de acuerdo a la influencia de este en el cálculo de la probabilidad de escogencia de camino.

Con este algoritmo en cada iteración se obtiene una solución que es puesta en competencia y se analiza en función a un frente de Pareto ya existente y de esa forma hasta conseguir un máximo de soluciones aceptables, nuestro frente de Pareto se irá optimizando hasta solo dejar las mejores soluciones del problema planteado.

Un factor que debe tenerse en cuenta en cada iteración es que este rastro de hormonas se va evaporando. Para esto se define inicialmente un factor de evaporación, entre 0 y 1 que al final de cada iteración va reduciendo el rastro de feromonas evitando el hallazgo de óptimos locales y el hallazgo de nuevas soluciones.

# Ejecución y Análisis de resultados.

## Consideraciones especiales:

La forma en cómo se aborda el TSPMO (TSP Multi-Objetivo) influye directamente sobre los criterios de avaluación, análisis e interpretación de resultados, por tal motivo a continuación se describen el conjunto de reglas, parámetro y premisas bajos los cuales se ejecutaran las pruebas.

En primer lugar el objetivo final de los dos algoritmos a comparar (SPEA2 y MOACO) es, dado un conjunto de ciudades donde todas están interconectadas (un número exageradamente grande en una conexión representa infinito, o una conexión nula, y es usado para indicarle al algoritmo que no puede tomar por allí) encontrar el camino más corto para alcanzar todos las ciudades y volver a la ciudad inicial, de este modo, no se da importancia a los nodos iniciales o finales así como al recorrido, sino al costo en tiempo y distancia asociados a este último.

Por otro lado ambos métodos escogidos están programados para calcular determinado número de soluciones en el frente de Pareto, de este modo hacer equiparable una comparación en cuanto a la calidad de las soluciones así del gasto computacional para obtenerlas con cada algoritmo.

Las poblaciones así como la cantidad de frentes de Pareto escogidos son de 30, ya que de acuerdo a la ley de los grandes números en estadística, a partir de este número se prescriben condiciones suficientes para que los promedios de las variables a analizar converjan al promedio de las esperanzas de las variables aleatorias involucradas y este tamaño de muestra nos acerca más a una distribución normal basada en un intervalo de confianza acotando a los valores esperados para las pruebas de corrida de ambos algoritmos.

## Datos obtenidos.

Sean :

**SPEA2:**

|  |  |  |  |
| --- | --- | --- | --- |
| Nodos | Media de las soluciones-Distancias | Desvacion de las soluciones- Distancias | Media de los tiempo de corrida para calcular Soluciones |
| 20 | 791.94 | 69.319 | 13.789 |
| 40 | 1655.2 | 90.511 | 9.8491 |
| 60 | 2529.4 | 141.66 | 6.7926 |

|  |  |  |  |
| --- | --- | --- | --- |
| Nodos | Media de las soluciones- Tiempo | Desvacion de las soluciones- Tiempo | Media de los tiempo de corrida para calcular Soluciones |
| 20 | 676.78 | 77.523 | 13.789 |
| 40 | 1617.7 | 101.87 | 9.8491 |
| 60 | 2569.1 | 111.62 | 6.7926 |

**MOACO:**

|  |  |  |  |
| --- | --- | --- | --- |
| Nodos | Media de las soluciones- Distancia | Desvacion de las soluciones- Distancia | Media de los tiempo de corrida para calcular Soluciones |
| 20 | 817.19 | 49.263 | 3.4182 |
| 40 | 1740.6 | 109.43 | 10.201 |
| 60 | 2678.5 | 107.05 | 23.441 |

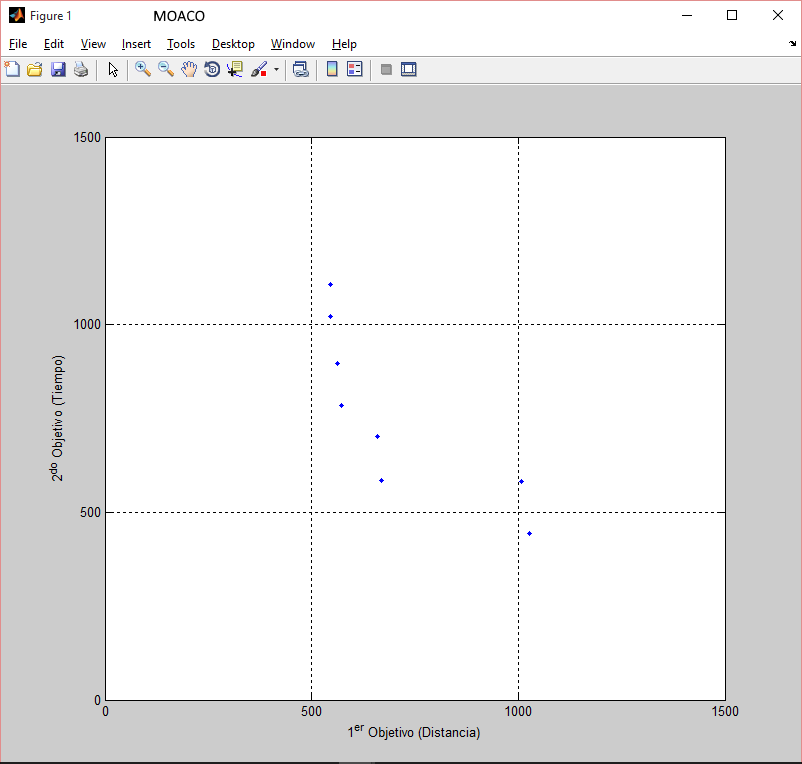
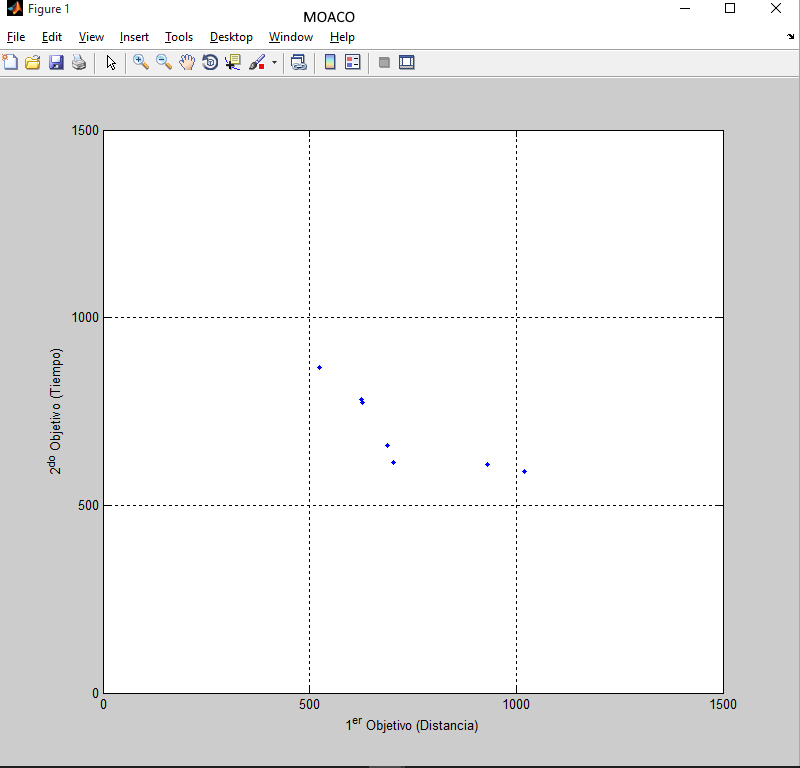
|  |  |  |  |
| --- | --- | --- | --- |
| Nodos | Media de las soluciones- Distancia | Desvacion de las soluciones- Distancia | Media de los tiempo de corrida para calcular Soluciones |
| 20 | 975.57 | 99.879 | 3.4182 |
| 40 | 2000.9 | 128.54 | 10.201 |
| 60 | 2908.4 | 184.5 | 23.441 |

De la ejecución iterativa de ambos algoritmos se obtuvieron en promedio los siguientes frentes de Pareto que dan solución para las gráficas en muestra a un modelo con cantidad de nodos igual a 20. Si bien no se establece un punto de partida y de destino para realizar un análisis de la capacidad de encontrar las soluciones óptimas a este problema para un caso específico, se manejó de manera aleatoria la escogencia de rutas óptimas con el objetivo de obtener la mejor ruta en general del grafo en cuestión. Con base a esto, se mostrará en nuestro análisis factores como tiempos de ejecución, conjuntos base de solución y una comparación de superposición de ambas soluciones para determinar de manera visual inicialmente que algoritmo para este caso y dado los parámetros iniciales expresados inicialmente brinda un mejor frente de Pareto.

## Análisis adicionales.

Adicionalmente, dada la complejidad de la forma de abordaje del problema, se presenta una forma alternativa que pude dar una luz a un análisis más estadístico más profundo. Esta alternativa consiste la superposición de los conjuntos de solución (frentes de Pareto) de varias ejecuciones de los dos algoritmos para un mismo archivo de rutas.

El archivo contiene 20 ciudades todas interconectadas todas entre sí en donde se tiene en cuenta la distancia el tiempo como objetivos a minimizar.

**MOACO**:

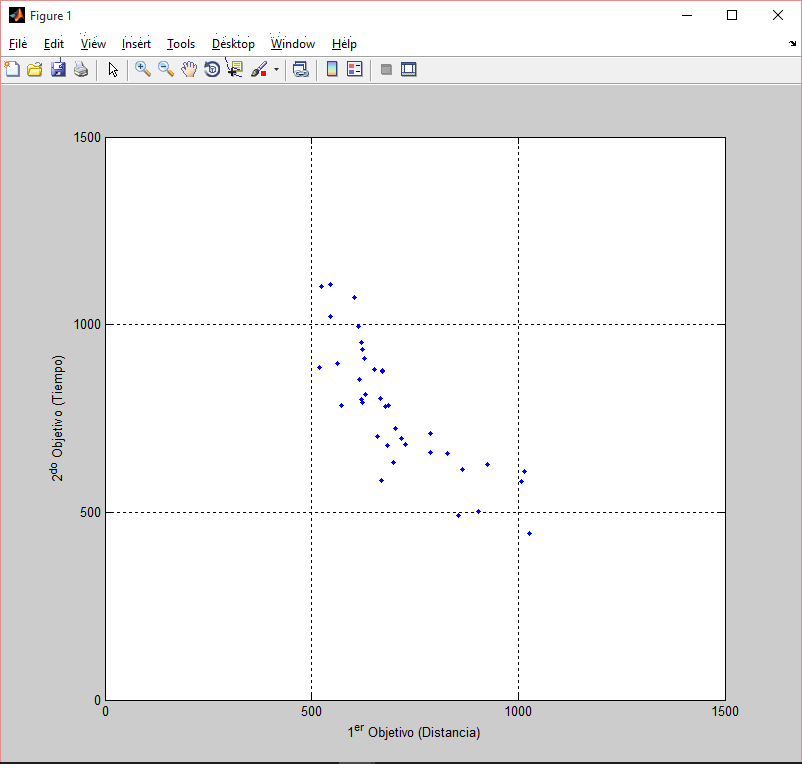
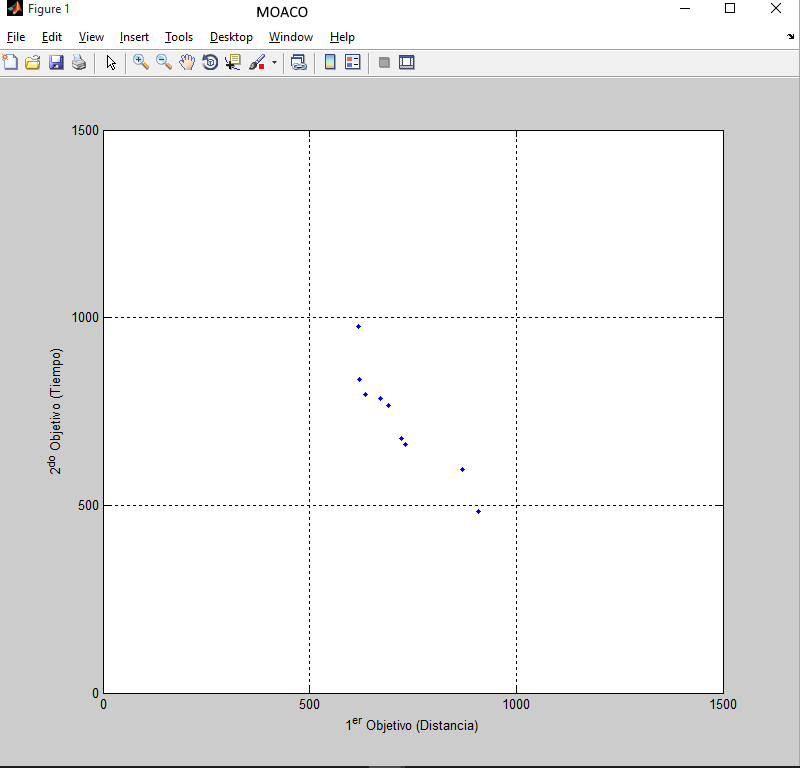
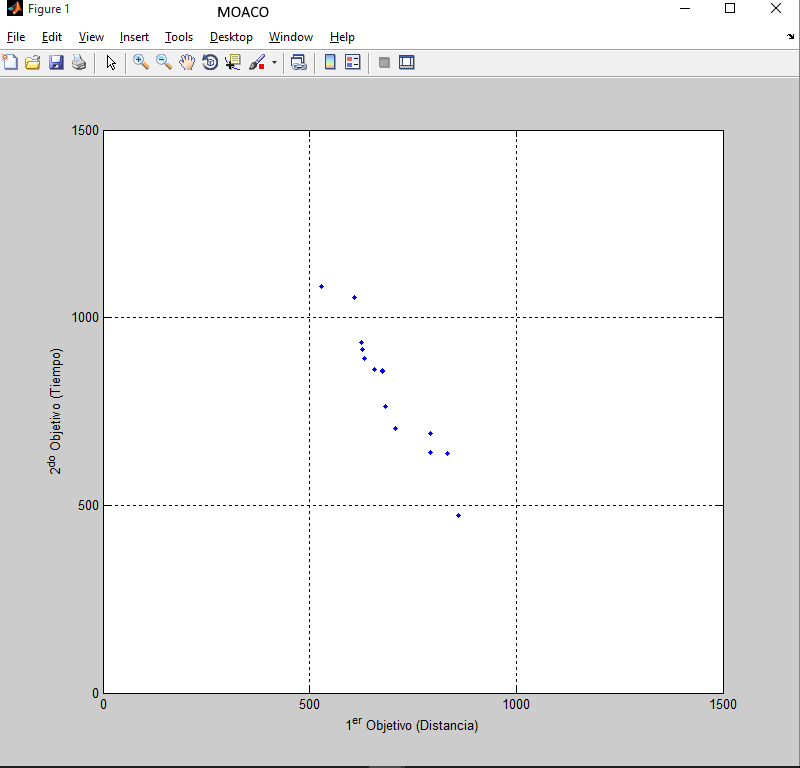
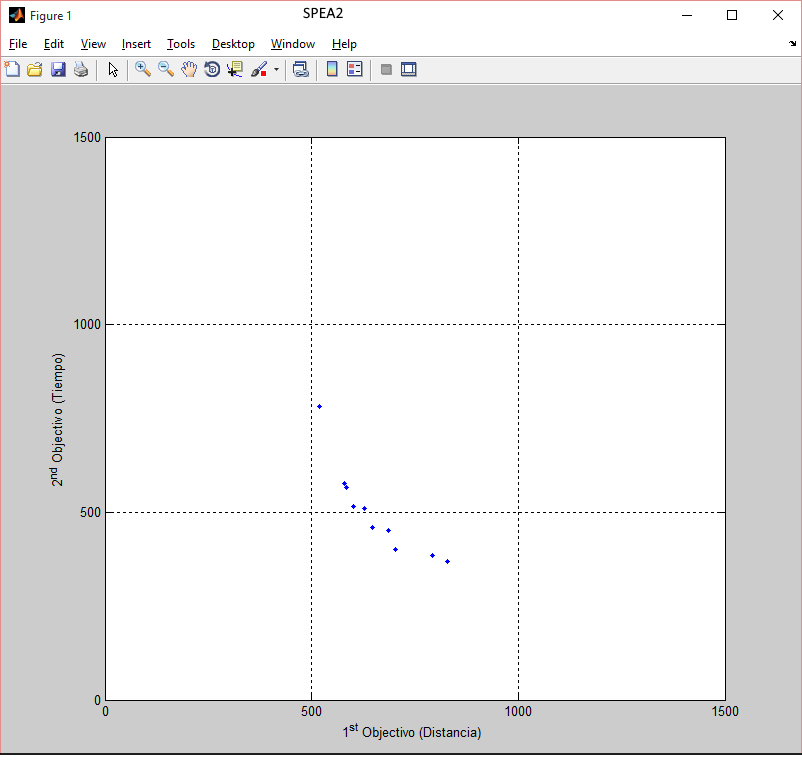
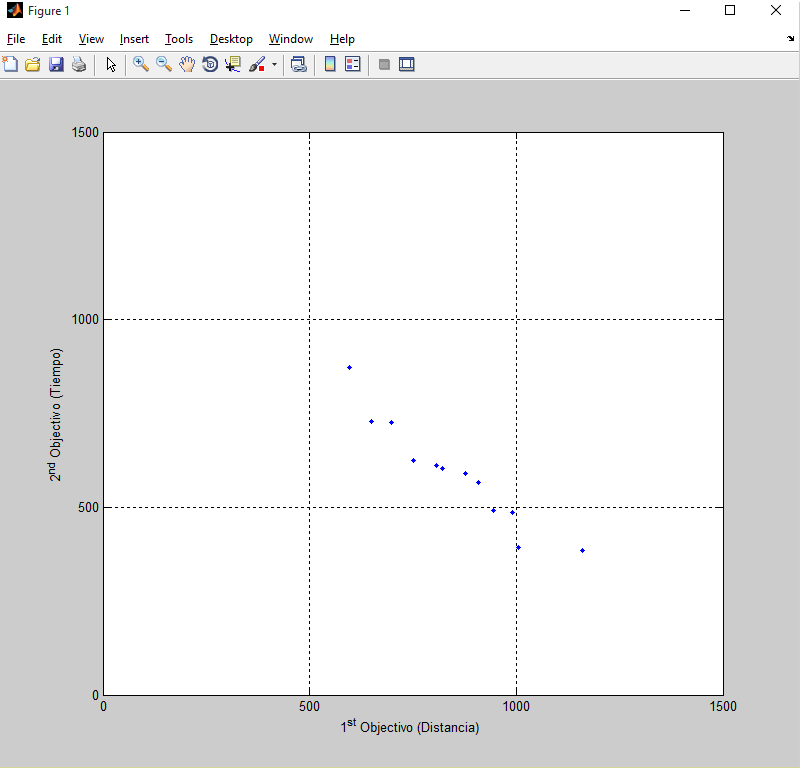
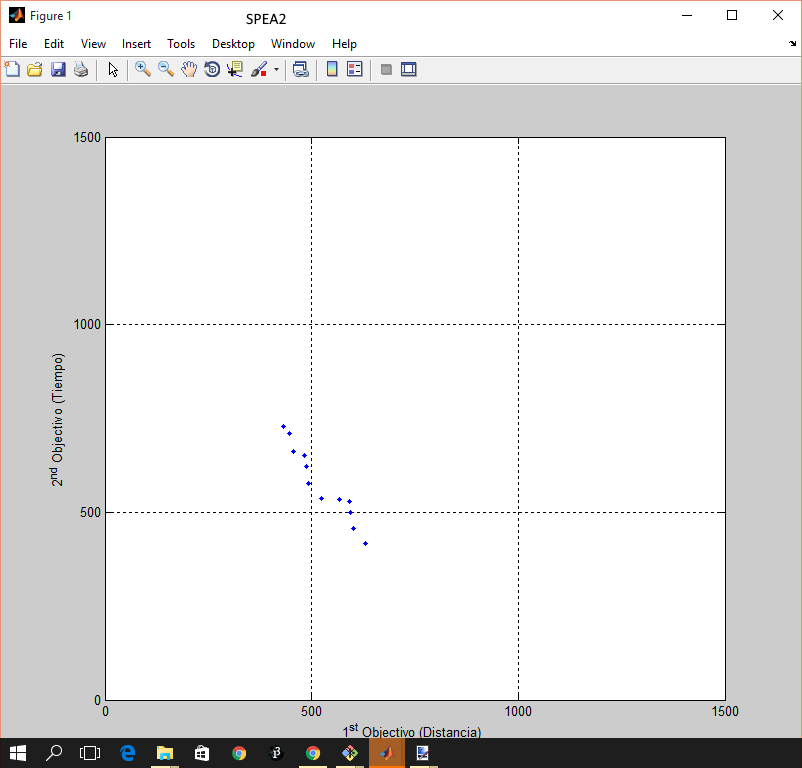
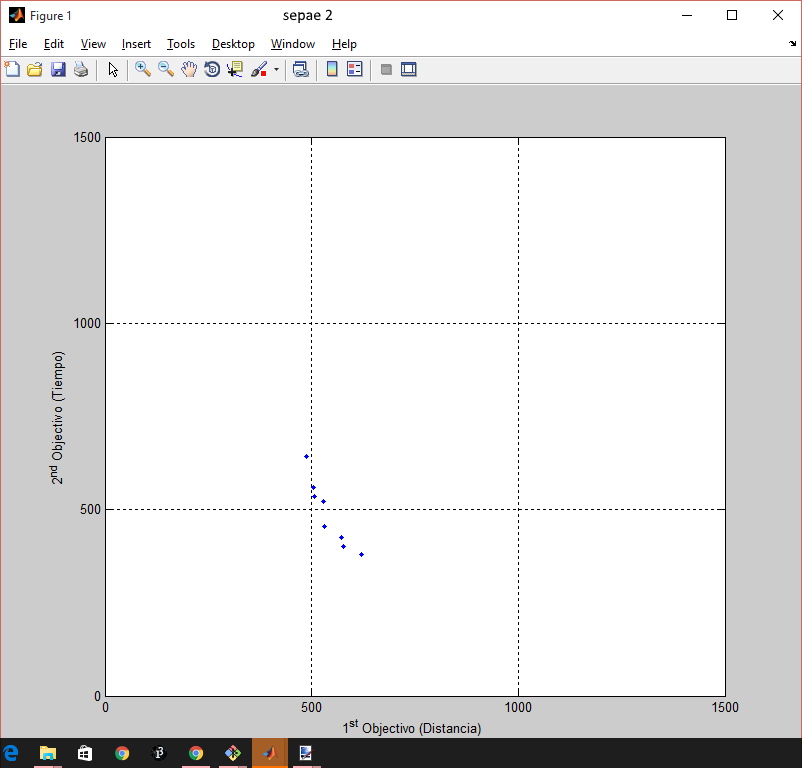
**SPEA 2:**

Diagrama conjunto agrupado de soluciones arrojado tras varias ejecuciones del MOACO.





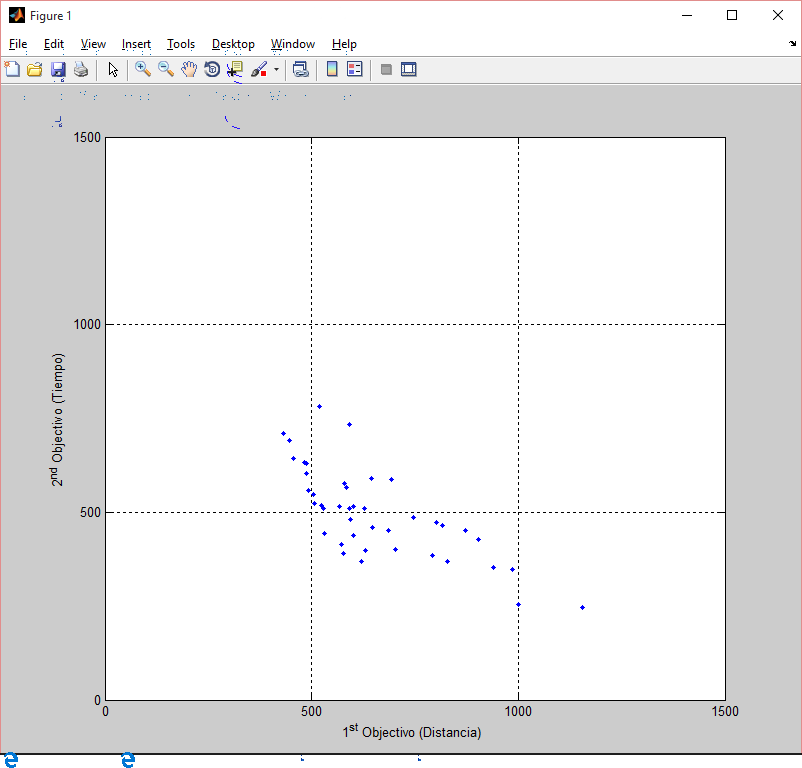
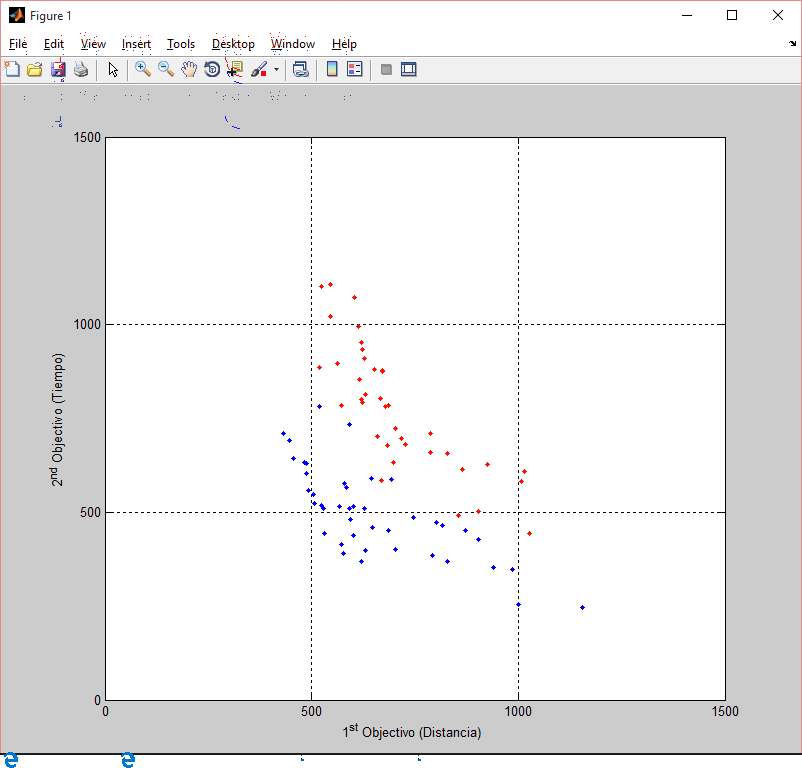


Diagrama conjunto agrupado de soluciones arrojado tras varias ejecuciones del SPEA2.

**MOACO VS SPEA2:**

Dado que no se cuentan con herramientas fuertes para analizar la calidad de las soluciones de un algoritmo frente al otro dada la naturaleza NO determinística en cómo estos algoritmos abordan el problema puesto que requiere de conocimientos superiores a los que contamos actualmente, se propone una comparación grafica de los resultados, en donde :

* Se representan todos los puntos solución alineados en los frentes de Pareto obtenidos para ambos algoritmos, para este diagrama no se tiene en cuenta la dominancia entre las diferentes soluciones ya que lo que se pretende es analizar de forma completa si los frentes de Pareto obtenidos por un algoritmo son mejores que los del otro.
* Se puede apreciar que aunque los dos algoritmos muestran tendencias a generar frentes de Pareto distanciados, al agruparse se puede observar que los frentes generados por MOACO tienen a quedar en la parte superior, lo que hace que queden dominados por los conjuntos de Pareto obtenidos por SPEA2.
* Debido a la naturaleza de las soluciones halladas por los dos algoritmos estudiados, en las cuales no se establecen una serie de parámetros o características iniciales que permitan establecer una relación entre ambos frentes de Pareto, no será posible aplicar técnicas de estadística tradicional debido al carácter no determinístico de las soluciones planteadas.

Sin embargo se han establecido otra serie de parámetros que permitan brindar una aproximación y normalización de resultados de muestra para arrojar información basada en cálculos obtenidos que brinden un informe de la eficiencia y prueba de optimalidad de un algoritmo con base al conjunto de soluciones propuestas por una corrida de iteraciones establecida inicialmente.

Gráficamente se puede observar una dominancia de las soluciones del SPEA2 sobre las soluciones del MOACO pero esta información no determina directamente que un algoritmo sea mejor que otro para la prueba de un solo caso. Para otras pruebas se obtuvieron diferentes resultados, como fue la dominancia de las soluciones del MOACO sobre el SPEA2 y esto va determinado por el carácter aleatorio de ambos algoritmos permitiendo que en diferentes iteraciones de un mismo problema inicial, obtener con un margen de error pequeño un conjunto de soluciones altamente cercana a una de las soluciones optimas globales que pueda tener el problema del agente viajero multi-variable.

Para cada iteración realizada, también se pudo apreciar que el tamaño del grafo determina lo disperso o compacto que van a estar ubicadas las soluciones en los frentes de Pareto, esto es, que a mayor cantidad de nodos de análisis para los algoritmos, la dispersión o espaciamiento entre puntos, fácilmente calculable con distancia euclidiana entre los puntos en función de la distancia y el tiempo va a ser mayor, sin importar que algoritmo se esté ejecutando.

Además, el análisis de tiempos de ejecución de ambos algoritmos, se observaron tiempos de ejecución similares dependiendo de la cantidad de nodos analizados y en estas metaherurísticas son determinadas directamente por el usuario, quien es el que decide la cantidad de soluciones que se deben incluir en el frente de Pareto para garantizar una solución óptima, para el caso del SPEA2.

En cuanto al MOACO, al este usar la verificación de no dominados del frente de Pareto, el algoritmo se vuelve determinístico probabilístico, ya que debido a factores como la cantidad feromonas, la influencia de estas en la solución en comparación a la distancia o tiempo, brindaran resultados diferentes basados en estas configuraciones iniciales, por muy pequeñas que sean.

Por último, observamos que las iteraciones que deben realizarse para obtener una misma cantidad de soluciones a colocar en el frente de Pareto varían en sobremanera dependiendo al algoritmo. Para SPEA2 en una iteración es normal que se obtenga un conjunto que pueda ir directamente al frente de Pareto y permanecer como solución o soluciones dominantes. Sin embargo, para MOACO, en cada iteración se obtiene únicamente una posible solución que deberá competir con el frente de Pareto actual, y esto en muchos casos extiende los tiempos e iteraciones necesarias para obtener un conjunto de soluciones optimas, que al final de su ejecución brinda un frente de Pareto aceptable, pero con tiempos mayores al SPEA2, en ciertos casos. Nuevamente se reitera que las observaciones brindadas pueden variar dependiendo a las consideraciones iniciales.

# Conclusiones y Trabajos Futuros.

En el contexto de optimización evolutiva multiobjetivo, no se puede garantizar la convergencia de las soluciones, ya que teóricamente todas las soluciones no dominadas son igualmente buenas entre sí, esto es conocido como elitismo de soluciones.

Para casos específicos como colonia de hormigas, garantizar una mayor evaporación de las feromonas ayuda a la obtención de nuevas soluciones y evita el estancamiento en óptimos locales. Este último concepto se aprecia en los algoritmos estudiados y para el caso de SPEA2 se garantiza esto manteniendo diversificada la población generada genéticamente y por torneo binario.

En un contexto comparativo ambos algoritmos estudiados proveen un diferente enfoque a una misma solución en el sentido que uno provee un conjunto más variado y amplio de soluciones (SPEA2) permitiendo un conjunto de soluciones más óptimos en comparación al otro algoritmo en estudio que por iteración genera una sola solución (COLONIA DE HORMIGAS).

Se observó que Colonia de Hormigas en apariencia tiende a sesgar las soluciones, pero con unos parámetros iniciales bien establecidos dependiendo a los datos de trabajo inicial, permite hallar soluciones óptimas basadas en resultados brindados por el frente de Pareto generado en cada iteración.

# Bibliografía

|  |  |
| --- | --- |
| [1] | M. A. Abido, «Multiobjective Optimal VAR Dispatch Using Strength Pareto Evolutionary Algorithm,» *2006 IEEE Congress on Evolutionary Computation Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada,* 2006. |
| [2] | A. H. B. B. Christian von L¨ucken1, «Algoritmos Evolutivos para Optimizaci´on Multiobjetivo: un Estudio Comparativo en un Ambiente Paralelo As´ıncrono,» 2008. |
| [3] | E. M.-M. y. C. A. C. Coello, «Conceptos de Optimizaci´on Multiobjetivo para el Manejo de Restricciones en Algoritmos Evolutivos: Un Estudio Comparativo,» *Grupo de Computaci´on Evolutiva,* 2010. |
| [4] | E. P. C. M. H. T. F. Pérez Vazquez, «MMSPEA2 mejora del SPEA2 para problemas multi-modales: aplicación al RCMPSP,» 2010. |
| [5] | E. Z. a. L. Thiele, «An Evolutionary Algorithm for Multiobjective Optimization The Strength Pareto Approach,» *Computer Engineering and Communication Networks Lab TIK Swiss Federal Institute of Technology ETH,* 2011. |
| [6] | A. Atehortúa, «Optimización Basada en Colonia de Hormigas: Generalidades y Estudio del Algoritmo Sistema Hormiga y Aplicación a un Job Shop,» vol. Tópicos Acanzados Universidad Nacional de Colombia, 2013. |