TSP, QAP, VRPTW:  
Resolución mediante Algoritmos MOEA: SPEA, NSGA y Algoritmos ACO: M3AS, MOACS

Marcelo Ferreira, Christian Gómez, Guido Casco,

Alida Invernizzi.

Electiva III, Inteligencia Artificial, Octavo Semestre,

{jmferreira1978, cgomezpy, guiancs82, alidainvernizzi}@gmail.com

**Resumen.** El documento presentado trata a cerca de las implementaciones realizadas para resolver los problemas del TSP, QAP, VRPTW, por medio de Algoritmos Multiobjetivos Evolutivos tales como SPEA y NSGA, así como también con los Algoritmos de Colonia de Hormigas tales como M3AS y MOACS.

**Palabras Claves:** Optimización multi-objetivo, colonias de hormigas, frente pareto

1. Introducción

Este documento muestra una comparación entre diversos algoritmos que utilizan la optimización multi-objetivo, dos de ellos con enfoque evolutivo y otros dos basados en el modelo del comportamiento de las colonias de hormigas reales, denominada metaheurística ACO (*AntColony Optimización*, u optimización basada en colonia de hormigas).

El trabajo considera algoritmos propuestos recientemente como el M-MMAS[Pinto05] y el MOACS [Paciello06]…

Se realizaron pruebas con dos instancias de cada problema. Se utilizaron reconocidos problemas de prueba de optimización multi-objetivo, el *Quadratic Assignment Problem* (QAP), definido en [Knowles03], el *Traveling Salesman Problem* (TSP) [Garcia04] y el *Vehicle Routing Problem with Time Windows* (VRPTW) [Baran03]. Estos problemas son considerados clásicos en la literatura de optimización combinatoria y del tipo NP-completos.

El trabajo está organizado como sigue: en la sección 2 se explica conceptos fundamentales sobre la optimización multiobjetivo, en la sección 3 se trata la formulación matemática de la optimización multiobjetivo y una descripción de los problemas, en la sección 4 se describen los algoritmos multi-objetivos utilizados. Los resultados experimentales de la comparación se muestran en la sección 5, y finalmente en la sección 6 se presentan algunas conclusiones y trabajos futuros

1. Conceptos de la Optimización Multiobjetivo

La optimización multi-objetivo puede ser definida como el problema de encontrar un vector de variables de decisión que satisfacen restricciones y optimiza un vector de funciones cuyos elementos representan las funciones objetivo. Estas definiciones aparecen en los trabajos de [Coello99] y [Deb99].

Optimizar

|  |  |
| --- | --- |
|  |  |

Sujeto a

|  |  |
| --- | --- |
|  |  |
|  |  |

Donde es el *vector de decisiones* con los valores para las *N* variables de decisión del problema, es el *vector solución* con las evaluaciones de las *M* funciones objetivos , y las funciones y representan respectivamente las *J* restricciones de desigualdad y las *K* restricciones de igualdad sobre el espacio de las variables de decisión. Por *optimizar* se entiende la *minimización* o *maximización* de cada una de las *M* funciones objetivas.

**Definición 1**: *Dominancia de Pareto*: Sean dos soluciones . Se dice que *u* domina a *v* (denotado como ) si es mejor o igual que *v* en cada uno de los objetivos y estrictamente mejor en al menos un objetivo. Como ejemplo, en un contexto de minimización si y solo si:

**Definición 2**: *Soluciones no comparables*: Dados , si ni , se dice que son soluciones no comparables, lo que se denota como *u ~ v*.

**Definición 3**: *Conjunto Pareto*: El conjunto de todas las soluciones no dominadas en se denomina Conjunto Pareto, lo que se denota como CP. Las soluciones que pertenecen a CP se denotarán como *x\**.

**Definición 4**: La imagen del Conjunto Pareto a través de la función se denomina Frente Pareto, denotado por *Y*.

1. Descripción de los Algoritmos

En la figura 1 se muestra el pseudo-código de un algoritmo ACO multi-objetivo genérico, denominado en adelante MOACO (MultiObjective Ant Colony Optimization). El MOACS y M3AS, presentado a continuación, siguen éste pseudo-código.

|  |
| --- |
| procedure MOACO  inicializar\_parametros()  while not condicion\_parada()  generacion=generacion + 1  for ant=1 to m // m es la cantidad de hormigas  construir\_solucion()  evaluar\_solucion()  actualizar\_feromonas()  actualizar\_conjunto\_pareto()  end for  end while  end  procedure construir\_solucion  sol={Ø}  while existen\_estados\_no\_visitados()  siguiente=seleccionar\_siguiente\_estado()  sol=sol U {siguiente}  marcar\_como\_visitado(siguiente)  if(actualizacion\_paso\_a\_paso)  actualizar\_feromonas\_paso\_a\_paso()  end while  end |

Fig. Pseudo-código de un algoritmo ACO multi-objetivo genérico.

* 1. MultiObjective Ant Colony System (MOACS)

MOACS, propuesto por Barán y Schaerer en [Baran03], es una extensión del MACS-VRPTW, este último propuesto por Gambardella et al. [Gambardella99]. Fue implementado considerando dos objetivos, utiliza una matriz de feromonas y dos visibilidades, una para cada objetivo del problema. La regla de transición de estados se calcula como:

|  |  |
| --- | --- |
|  |  |

El cálculo de se realiza según:

|  |  |
| --- | --- |
|  |  |

Cada vez que una hormiga se mueve del estado *i* al estado *j*, realiza la actualización local de feromonas según la ecuación:

|  |  |
| --- | --- |
|  |  |

con , el valor inicial para las feromonas, y representa el coeficiente de evaporación y,

|  |  |
| --- | --- |
|  |  |

con donde por motivos de normalización los valores son divididos por un valor máximo definido a priori.

En el caso de encontrar una solución no dominada, se actualiza CP y se reinicializa la tabla de feromonas, considerando que la información fue aprendida por medio de soluciones dominadas. Si la solución encontrada es dominada se realiza la actualización de feromonas según la ecuación (6).

* 1. Multiobjective Max-Min Ant System (M-MMAS o M3AS)

Este algoritmo, propuesto por Pinto et al. en [Pinto05], extiende el Max-Min Ant System para resolver problemas multi-objetivos. Se utilizó inicialmente para resolver el problema de enrutamiento multicast multi-objetivo. Pinto et al. en su trabajo [Pinto05] optimizaron cuatro objetivos. Se mantiene una tabla de feromonas global que mantiene información de feromonas considerando todos los objetivos a optimizar. Una hormiga estando en el estado *i* escoge el siguiente estado a visitar, de acuerdo con la probabilidad *p* dada según la ecuación (5).

Las soluciones no dominadas actualizan la tabla de feromonas según la ecuación (6). Si , entonces , con y si , entonces , con , donde *m* es el número de hormigas, *k* representa la *k-ésima* solución y se calcula con la ecuación (7). De esta manera se impone una cota inferior y otra superior al nivel de feromonas en los arcos.

1. Resultados Experimentales
   1. Descripción del Hardware Utilizada

Todos los algoritmos fueron implementados en Java (v. 1.6) y fueron ejecutados en un entorno Windows, Version Vista, en una máquina AMD Turion 2.2GHz con 3GB de memoria. Se realizaron diez corridas de 10 iteraciones para cada algoritmo y para cada problema de prueba. Como problemas de prueba se utilizaron dos instancias de cada tipo de problema (TSP, QAP y VRPTW). En el caso del TSP se utilizaron las instancias bi-objetivas de 100 ciudades KROAB100 y KROAC100. Para el QAP bi-objetivo, se utilizaron las instancias de 75 localidades qapUni.75.0.1 y qapUni.75.p75.1. Para el VRPTW bi-objetivo se utilizaron las instancias de 100 clientes c101 y rc101.

Se utilizó *m*=10 hormigas, =1, =2, =0.1, =0.8, =0.9, =0.5 , =1.

* 1. Métricas de Comparación

Para poder evaluar los resultados obtenidos en cada corrida de los métodos MOEAs

y MOACOs fueron utilizadas las métricas propuestas por Zitzler et al.

[Ziztzler00], que evalúan respectivamente la calidad de las soluciones, la distribución de las soluciones y la extensión del frente Pareto aproximado *Y’* devuelto en cada corrida. También los métodos fueron comparados con respecto al número de soluciones no dominadas encontradas en cada corrida, denotado por *|Y’|*.

La métrica proporciona una idea de la aproximación al frente Pareto real de un frente Pareto aproximado *Y’*, calculando el promedio de las distancias euclidianas de cada solución en el frente *Y’* a la solución más cercana en el frente .

La métrica estima la distribución promedio de las soluciones a lo largo de un frente Pareto aproximado *Y’*, calculando el número promedio de soluciones que se encuentran separadas de cada solución a una distancia mayor que cierto valor definido a priori.

La métrica evalúa la extensión o abarcamiento de un frente Pareto aproximado *Y’* a través de la sumatoria de las máximas separaciones de las evaluaciones en cada objetivo.

La métrica de cantidad de soluciones *|Y’|* da una idea acerca de la diversidad de combinación de las evaluaciones de los objetivos presentadas al Tomador de Decisiones, esta métrica puede ser considerada como un complemento de las demás métricas.

En la fig. 1 se puede apreciar las métricas. La definición formal de dichas métricas es:

donde *Y’* es el frente Pareto aproximado devuelto en una corrida, *d(p, q)* calcula la distancia euclidiana entre las soluciones *p* y *q*, |·| representa la cardinalidad, *M* es la dimensión del espacio objetivo y se estableció al 10% de la distancia entre los puntos extremos del frente Pareto aproximado *Y’*.

Para cada corrida, los valores de sus evaluaciones en cada métrica fueron normalizados a un número en el intervalo [0, 1], de manera a poder utilizar estos resultados en rankings de métodos presentados en la siguiente sección.

Las evaluaciones en la métrica fueron normalizadas restando de 1 el resultado de la división de la evaluación de cada corrida por el mayor valor obtenido en esta métrica en cada problema.

Para la métrica , la evaluación máxima de una corrida es igual al número de soluciones no dominadas encontradas en dicha corrida [Zitzler00], así las evaluaciones de las corridas fueron normalizadas dividiéndolas por el número de soluciones encontradas en dichas corridas.

Con relación a la métrica , las evaluaciones de cada corrida fueron normalizadas dividiéndolas por el valor de evaluación de esta métrica para el frente Pareto real de cada problema.

La cantidad de soluciones no dominadas encontradas *|Y’|* en cada corrida fue normalizada dividiéndola por el mayor valor de evaluación de esta métrica en cada problema.

De esta forma, los valores de evaluación normalizados son siempre menores que 1 y se consideran mejores cuanto más próximos encuentren a dicho valor.



Fig. Ejemplo de criterios (a) calidad (b) distribución (c) extensión para la comparación de los frentes Paretos aproximados [Lima07]

* 1. Resultados de la Comparación

1. Conclusiones
2. Trabajos Futuros
3. Referencias

|  |  |
| --- | --- |
| [Paciello06] | J. Paciello, H. Martínez, C. Lezcano and B. Barán. Algoritmos de Optimización multi-objetivos basados en colonias de hormigas. Proceedings of CLEI’2006. Latin-American Conference on Informatics (CLEI). Santiago, Chile. |
| [Pinto05] | D. Pinto y B. Barán. “Solving Multiobjective Multicast Routing Problem with a new Ant Colony Optimización approach”. LANC’05, Cali, Colombia. 2005. |
| [Knowles013 | J. Knowles y D. Corne. “Instance generators and test suites for the multiobjective quadratic assignment problem”. In: Fonseca, C.M., et al. Editors. Proc of EMO '03, LNCS 2632 page 295-310, Springer-Verlag, 2003 |
| [Garcia04] | C. García-Martínez, O. Cordón y F. Herrera. “An Empirical Análisis of Multiple Objective Ant Colony Optimización Algorithms for the Bi-criteria TSP”. ANTS Workshop 61-72. 2004 |
| [Baran03] | B. Baran y M. Schaerer. “A multiobjective Ant Colony System for Vehicle Routing Problems with Time Window*s”.* Proc. Twenty first IASTED International Conference on Applied Informatics, pg. 97-102. Insbruck, Austria. 2003 |
| [Deb99] | K. Deb. “Evolutionary Algorithms for Multi-Criterion Optimización in Engineering Design”. In Proceedings of Evolutionary Algorithms in Engineering and Computer Science EUROGEN’99. 1999 |
| [Coello99] | C. Coello. An updated Survey of Evolutionary Multiobjective Optimización Techniques: state of the art and future trends. In Congress on Evolutionary Computation. Piscataway, N. J., IEEE Service Center. 3–13. 1999 |
| [Gambardella99] | L. Gambardella, E. Taillard y G. Agazzi. “MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing  Problems with Time Windows”. In D. Corne, M. Dorigo, F. Glover (Eds.), New Ideas in Optimización, McGraw-  Hill, 73-76. 1999 |
| [Ziztzler00] | E. Zitzler, K. Deb and L. Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. Evolutionary Computation, vol. 8, no.2, pp 173–195. 2000 |
| [Lima07] | J. Lima. Optimización de enjambre de partículas aplicada al problema del cajero viajante bi–objetivo, p. 87. 2007 |