Aplicación de una nueva variante de algoritmo basado en Ant Colony Optimization (ACO) al TSP *Many-objective*)

Francisco Prisciliano Riveros Núñez
Facultad Politécnica
UNA
San Lorenzo, Paraguay
friverosn@gmail.com

Néstor Benítez Dietrich
Facultad Politécnica
UNA
San Lorenzo, Paraguay
nestorbdietrich@gmail.com

Abstract— **Este trabajo propone un análisis comparativo de los algoritmos basados en colonias de hormigas multi-objetivos (MOACO) aplicados al TSP Many-objective. Para las pruebas experimentales fueron generadas instancias del TSP con baja correlatividad para representar problemas de 2, 4 y 8 objetivos. Se aplicaron las métricas del Hipervolumen y Coverage a los Frentes Pareto resultantes con el objetivo de verificar su rendimiento, luego, se ajustaron empíricamente los parámetros de los algoritmos con el objetivo de mejorar el Hipervolumen.**

Keywords— ***MOACO, TSP, Many-objective, Hypervolume, Coverage***

I. INTRODUCCIÓN

Los algoritmos basados en colonias de hormigas multi-objetivos (MOACOs) son utilizados para resolver problemas de optimización Multi-objetivos. En trabajos anteriores como [1], [2] y [3] se aplicaron estos algoritmos en la resolución de problemas reales. Usualmente los MOACOs trabajan en forma correcta con problemas de dos y hasta tres objetivos, sin embargo, el rendimiento de estos, disminuye en la medida en que crece el número de objetivos como sucede con los algoritmos evolutivos [13]. Este trabajo propone evaluar el rendimiento de los MOACOS aplicados a instancias del TSP Many-objective con 2, 4 y hasta 8 objetivos utilizando las métricas del Hipervolumen y *Coverage* a los Frentes Pareto resultantes. También se propone modificaciones a los algoritmos MOACO con el objetivo de mejorar el rendimiento de los mismos cuando son aplicados a problemas Many-objective. En las siguientes secciones se describen el TSP Many-objective, los algoritmos MOACO considerados, las modificaciones realizadas a los mismos y finalmente las conclusiones y trabajos futuros.

II. TSP MANY-OBJECTIVE

El TSP puede ser representado por un grafo ponderado completamente conexo $G = (N, A)$, donde N representa el conjunto de nodos y A el conjunto de aristas que conectan todos los nodos de N. En un caso mono-objetivo, cada arista tiene asociado un costo d_{ij}, que representa la distancia entre los nodos $i, j \in N$. En otras palabras, TSP puede formularse como encontrar la mínima distancia del ciclo Hamiltoniano, desde un nodo inicial, visitando cada nodo exactamente una vez y volver al nodo inicial [11]. En un TSP simétrico, $d_{ij} = d_{ji}$ para cada par de nodos. Para el Many-objective, son considerados k funciones de costos, teniendo $d^1_{ij}, d^2_{ij}, \ldots, d^k_{ij}$ para cada arista (i, j) y el problema consiste en minimizar cada función de costo al mismo tiempo [12].

En la literatura son conocidas varias instancias de TSP bi-objetivo[1]; no siendo así para el caso Many-objective con 4 o más objetivos. Este trabajo propone la evaluación del TSP Many-objective. Para el efecto, se generaron instancias de 2, 4 y 8 objetivos. Las matrices fueron creadas asegurando una baja correlación entre ellos con el objetivo de representar objetivos contradictorios. Como indicador de la correlación se utilizó el coeficiente de correlación de Pearson [4]. Las correlaciones aceptadas oscilaron entre -1 y 0.1, lo que supone una correlación razonablemente baja para representar objetivos contradictorios.

III. ALGORITMOS MOACO CONSIDERADOS

En este trabajo fueron implementados el MOACS [1], M3AS [2] y el MAS [3]. Cada MOACO fue objeto de varias pruebas utilizando las instancias Many-objective generadas de 2, 4 y 8 objetivos. Las pruebas fueron ejecutadas en un contexto de minimización de costos. En base a los resultados preliminares, el MOACS (Ver pseudocódigo 1), que obtuvo un mejor desempeño, fue seleccionado para realizar modificaciones buscando mejorar el rendimiento para problemas Many-objective. Las modificaciones realizadas al MOACS corresponden a la cantidad de hormigas, el parámetro de distribución λ y el impacto visibilidad. Además, fue implementado una modificación del área de dominancia de soluciones del Frente Pareto con el fin de verificar el impacto

[1] http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/tsp/

en las métricas de rendimiento de acuerdo a trabajos anteriores como el de Sato et al. [5].

IV. Metricas de rendimiento

En este trabajo fueron utilizados las métricas del Hipervolumen y *Coverage* [7]. El Hipervolumen considera el tamaño de la región de dominancia en el espacio objetivo, y el *Coverage* compara dos conjuntos de soluciones no dominadas estimando la fracción de soluciones dominadas de un conjunto al otro. El Hipervolumen combina las métricas de distancia, distribución y extensión en un solo valor [8]. El *Coverage* puede ser utilizado para demostrar que un algoritmo domina otro, sin embargo, no es factible cuantificar que tanto un algoritmo es mejor que otro [8]. El algoritmo implementado para el cálculo del Hipervolumen es el *Hypervolume by Slicing Objectives* (HSO), propuesto en [9].

V. Resultados experimentales

En esta sección se presentan los resultados experimentales obtenidos, y los diferentes ajustes realizados a los parámetros. Todas las pruebas fueron ejecutadas en una computadora personal Mac OS X con procesador Intel i7 y 8 GB de memoria RAM. Todos los algoritmos fueron ejecutados 4 veces con cada instancia hasta 1.500 generaciones. La Tabla I muestra la configuración de los parámetros utilizados por los algoritmos MOACO.

A. Comparacion de algoritmos MOACO

Se ejecutaron varias pruebas utilizando los algoritmos MOACS [1], M3AS [2], el MAS [3] y NSGAII [14] aplicados a instancias de 2, 4 y 8 objetivos. En las Tablas II y III se puede observar un comportamiento similar de todos los algoritmos MOACO (MOACS, M3AS y MAS) en términos de Hipervolumen.

En general, el Hipervolumen disminuye en la medida que crece el número de objetivos. En los algoritmos MOACO, se puede observar que el Hipervolumen tiende a 0, y la cantidad de soluciones tiende al total de soluciones posibles generadas por cada hormiga en todas las generaciones. El tiempo necesario para realizar el cálculo del Hipervolumen aumenta considerablemente cuando se aplican a problemas con 16, 32 o más objetivos por lo tanto se convierte en inviable. Teniendo en cuenta esta limitación para calcular el Hipervolumen en un tiempo razonable, este trabajo limita el número de objetivos a 8. Considerando los resultados obtenidos, el MOACS muestra una leve mejoría por lo que fue seleccionado para realizar varios ajustes en sus parámetros con el fin de aumentar el Hipervolumen obtenido.

A continuación, se muestran los ajustes realizados con sus respectivos resultados.

B. Modificaciones en la cantidad de hormigas

Inicialmente, cada algoritmo fue ejecutado utilizando una cantidad constante de hormigas $m = 10$. Este parámetro se modificó mediante las siguientes estrategias:

1) Múltiplos del número de objetivos: La cantidad de hormigas fue modificada a $m = k.n$, donde k representa la cantidad de objetivos y $n \in \{1, 2, 3, k\}$. Por lo tanto, la cantidad de hormigas será un múltiplo de k.

2) Seleccion aleatoria de la cantidad de hormigas: Se realizó una selección aleatoria de la cantidad de hormigas de acuerdo a $m = 10 + ram$, donde *ram* representa un número entre k y k^2, siendo k el número de objetivos.

3) *Aumento constante de la cantidad de hormigas:* Se realizó un aumento constante de la cantidad de hormigas de acuerdo a $m = m + const$, donde $const = MaxGenerationNumber / (m_0^2 - m_0)$, y m_0 es la cantidad inicial de hormigas. Para el proposito de esta prueba fue considerado $m_o = 10$.

En la Tabla IV se puede observar la evolución del Hipervolumen obtenido por cada estrategia implementada y cada instancia del problema. Como se puede observar en la Tabla IV, ninguna de las estrategias propuestas muestra una mejora significativa del Hipervolumen.

C. Impacto de la Visibilidad

Este trabajo propone 3 estrategias para disminuir el impacto de la visibilidad cuando se calcula la probabilidad del siguiente estado MOACO. Esta modificación se utiliza para guiar la búsqueda basadas en los rastros feromonas cuando ha transcurrido cierta cantidad de generaciones. Las estrategias utilizadas fueron:

1) Reducir el impacto de la visibilidad 1% cada 15 generaciones.
2) Reducir el impacto de la visibilidad 6% cada 100 generaciones.
3) Reducir el impacto de la visibilidad 30% cada 500 generaciones.

En la Tabla V se puede observar la evolución de Hipervolumen utilizando instancias con 2, 4 y 8 objetivos y las diferentes estrategias implementadas. En este caso, tampoco se observa una mejora significativa.

D. Modificación en el área de dominancia

Fue implementado el algoritmo presentado en [5], donde se contrae y amplia el área de dominancia de las soluciones del Frente Pareto. El área de dominancia se modificó según un valor S_i, en el intervalo [0.25, 0.7]. En la Tabla V se pueden observar los resultados obtenidos. En este caso, ninguna de las modificaciones produce una mejora significativa en el Hipervolumen. Por esta razón, esta variante también fue descartada.

E. Distribucion Lambda (λ)

Se realizaron algunos ajustes en el esquema de asignación del parámetro λ que pondera el impacto visibilidad considerando todos los objetivos a optimizar.

```
procedure MOACS
    inicializarFeromonas()
    while no cumple criterio de parada do
      for i = 1 a m do
        (T = ConstruirSolucion)
      end for
      if (Yknow was changed) then
        inicializarFeromonas()
      else
       for all (T ∈ Yknow)
         actualizarFeromonas()
       end for
      end if
    end while
end procedure
```

Pseudocódigo. 1. MOACS genérico.

En otros términos, para resolver problemas Multi-objetivos utilizando algoritmos MOACO, se debe asignar el parámetro λ a cada una de las *m* hormigas; por cada hormiga, λ es asignado *k* veces donde *k* es la cantidad objetivos, es decir, cada hormiga tendrá los siguientes valores $\lambda_1, \lambda_2, ..., \lambda_k$. En general, cada λ_i toma uno de *m* valores posibles sin repetir los valores asignados a otra λ_j. Por lo tanto, la distribución de λ representa una permutación de *k-m*, lo cual dificulta el uso de todas las posibles permutaciones cuando aumenta la cantidad de objetivos. Para abordar esta cuestión, este trabajo propone 2 estrategias: una selección aleatoria de la permutación de valores para λ utilizando una Búsqueda Tabú [10], y una nueva variante denominada "Asignación aleatoria de λ en base p".

Distribución aleatoria de λ utilizando Búsqueda Tabú:

En este enfoque, este trabajo utilizó una Búsqueda Tabú con una cola de 5 valores. Se implementó un proceso de selección aleatoria de la permutación para el λ_i a utilizar, donde $i \in \{1, 2, ..., k\}$, sin repetir las últimas 5 permutaciones de acuerdo a la Búsqueda Tabú. Se probaron 2 variantes de asignación de λ, valores normalizados de acuerdo a *m*, y sin normalización.

Asignación aleatoria de λ en base p:

En esta caso, se aplicó una restricción en la cantidad de valores posibles que λ_i puede tomar. Teniendo en cuenta los objetivos $f_1, f_2, ...f_k$, cada hormiga necesita los valores $\lambda_1, \lambda_2, ..., \lambda_k$ para cada visibilidad. Para este propósito, se genera un valor numérico en base decimal entre 0 y $p^k -1$, donde p es un parámetro de esta estrategia que representa el número de valores posibles que puede tomar cualquier λ_i. El valor numérico generado se transforma a base p, de esta forma se obtienen k dígitos $d_1, d_2, ..., d_k$, donde $d_i \in \{0,1,...,p-1\}$ para cada $i \in \{1,2,...,k\}$. La asignación de λ se realiza de la siguiente manera $\lambda_1 = d_1, \lambda_2 = d_2, ..., \lambda_k = d_k$, donde 0 representa el valor de menor peso y p-1 el de mayor peso. Para este trabajo se utilizó $p = 3$, por ende, cada λ_i puede tomar los valores 0 (ponderación baja), 1 (ponderación media), 2 (ponderación alta). Para evaluar este método se proponen dos estrategias:

1) Asignación aleatoria de λ en base *p* al inicio de la ejecución del algoritmo.

2) Asignación aleatoria de λ en base *p* al inicio de cada generación (iteración).

En la Tabla VII se puede observar la evolución del Hipervolumen. Para obtener un análisis detallado, los valores computados del Hipervolumen (considerando el MOACS con las variantes propuestas) se pueden ver en la Tabla VIII. En ambas tablas se observa una ventaja al utilizar la variante propuesta del MOACS con asignación aleatoria de λ en base p al inicio de cada generación (MOACS λ Iteración) hasta 8 objetivos, lo que se refleja con una mejora en el Hipervolumen. Para reforzar estos resultados, el Frente de Pareto obtenido por el MOACS con asignación aleatoria de λ en base p al inicio de cada generación se comparó con los demás Frentes Pareto, producidos por los otros algoritmos, utilizando la métrica del *Coverage* en ambas direcciones. En la Tabla IX se muestran los resultados del *Coverage* para 8 objetivos.

VI. Conclusiones y trabajos futuros

Este trabajo presenta un análisis comparativo de algoritmos MOACO aplicado al TSP Many-objective, considerando 2, 4 y 8 objetivos. Se implementaron varias modificaciones a diferentes parámetros, como la cantidad de hormigas, el impacto de la visibilidad, el área de dominancia y finalmente, la estrategia de asignación de valores de λ. Analizando el Hipervolumen obtenido, se puede observar una mejora significativa cuando se utiliza la asignación aleatoria de λ en base p por generación. Por esta razón, esta estrategia se recomienda para resolver problemas Many-objective utilizando algoritmos MOACO.

Como trabajos futuros, se pueden resolver instancias con más de 8 objetivos utilizando la estrategia de asignación de λ propuesta. También se puede implementar un análisis comparativo con los algoritmos evolutivos aplicados a problemas Many-objective.

TABLA I. PARAMETROS DE ALGORITMOS BASADOS EN LOS TRABAJOS [1],[2],[3] Y [15].

MOACO	
Parámetro	***Valor***
Máxima generación	1500
Cantidad de Hormigas	10
Impacto de feromonas	1
Impacto de visibilidad	2
Coeficiente de evaporación	0.3
τ_0	0.1
τ_{max}	0.9
q_0	0.5
f_{max}	7782
NSGA-II	
Máxima generación	1500
Tamaño de población	10
Probabilidad Mutacion de padres	0.0
Probabilidad Mutacion de hijos	0.8
f_{max}	7782
Inversión_p	0.02

TABLA II. HIPERVOLUMEN DE ALGORITMOS EVALUADOS.

Algoritmos	*2*	*4*	*8*
M3AS	0,80	0,37	0,04
MAS	0,82	0,39	0,04
MOACS	0,83	0,39	0,04
NSGA-II	0,59	0,15	0,007

TABLA III. CANTIDAD DE SOLUCIONES.

Algoritmos	*2*	*4*	*8*
M3AS	59	1204	9881
MAS	29	1094	9536
MOACS	45	1223	9617
NSGA-II	25	183	118

TABLA IV. HIPERVOLUMEN DEL MOACS PARA CADA ESTRATEGIA DE VARIACIÓN EN LA CANTIDAD DE HORMIGAS.

Algoritmos	*2*	*4*	*8*
MOACS k1	0,80	0,38	0,04
MOACS k2	0,81	0,39	0,04
MOACS k3	0,82	0,38	0,04
MOACS k^2	0,82	0,37	0,04
MOACS $k_{aleatorio}$	0,81	0,38	0,04
MOACS $k_{constante}$	0,81	0,42	0,04

TABLA V. HIPERVOLUMEN DEL MOACS DISMINUYENDO EL IMPACTO DE LA VISIBILIDAD.

Algoritmos	*2*	*4*	*8*
MOACS Dim. 1% / 15 gen.	0,81	0,39	0,04
MOACS Dim. 6% / 100 gen.	0,82	0,38	0,04
MOACS Dim. 30% / 500 gen.	0,82	0,38	0,04

TABLA VI. HIPERVOLUMEN DEL MOACS APLICANDO LA CONTRACCIÓN Y EXPANSIÓN DEL AREA DE DOMINANCIA.

Algoritmos	*2*	*4*	*8*
MOACS $S_{0,25}$	0,66	0,29	0,02
MOACS $S_{0,3}$	0,66	0,32	0,02
MOACS $S_{0,35}$	0,70	0,37	0,03
MOACS $S_{0,4}$	0,81	0,37	0,04
MOACS $S_{0,45}$	0,81	0,38	0,04
MOACS $S_{0,5}$	0,81	0,38	0,04
MOACS $S_{0,55}$	0,80	0,37	0,03
MOACS $S_{0,6}$	0,77	0,36	0,03
MOACS $S_{0,65}$	0,75	0,34	0,03
MOACS $S_{0,7}$	0,73	0,32	0,02

TABLA VII. HIPERVOLUMEN DEL MOACS SEGÚN LA ESTRATEGIA DE ASIGNACIÓN.

Algoritmos	*2*	*4*	*8*
MOACS λ. Tabú normalizado	0,73	0,28	0,02
MOACS λ. Tabú no-normalizado	0,72	0,28	0,02
MOACS λ. Inicio	0,84	0,41	0,04
MOACS λ. Iteración	0,84	0,50	0,09

TABLA VIII. RESULTADOS OBTENIDOS DE ACUERDO A LA ESTRATEGIA DE ASIGNACIÓN DE LAMBDA.

Algoritmos	*2*	*4*	*8*
M3AS	0,80	0,37	0,04
MAS	0,83	0,39	0,04
MOACS	0,82	0,39	0,04
MOACS λ. Inicio	0,84	0,41	0,04
MOACS λ. Iteración	0,84	0,50	0,09

TABLA IX. COVERAGE.

	MAS	*M3AS*	*MOACS*
COV.(MOACS λ. Iteración,x)	0,0030	0,0018	0,0031
COV.(x, MOACS λ. Iteración)	0,0014	0,0017	0,0014

REFERENCIAS

[1] B. Benjamín, S. Matilde. "A multiobjective Ant Colony System for Vehicle Routing Problems with Time Windows", Proc. Twenty first IASTED International Conference on Applied Informatics, Insbruck, Austria, pp. 97-102, 2003.

[2] P. Diego, B. Benjamín. "Solving Multiobjective Multicast Routing Problem with a new Ant Colony Optimization approach". LANC 05, Cali-Colombia, 2005.

[3] P. Julio, M. Héctor. "Optimización Multiobjetivo basada en Colonias de Hormigas. Teoría y estrategias de paralelización", pp. 31-44, 2006.

[4] M. Etcheverria."Análisis descriptivo de datos en Educación", pp. 172-180, 2005.

[5] S. Hiroyuki, A. Hernán, T. Kiyoshi. "Controlling Dominance Area of Solution in Multiobjetive Evolutionary Algorithms and Performance Analysis on Multiobjective 0/1 knapsack Problems", 2007.

[6] Z. Eckart, T. Lothar. "Multiobjective optimization using evolutionary algorithms – a comparative case study", 1998.

[7] Z. Eckart, T. Lothar. "Multiobjective evolutionary algo- rithms: A comparative case study and the strength pareto approach", 1999.

[8] Z. Eckart. "Evolutionary algorithms for multiobjective optimization: Methods and applications", 1999.

[9] W. Lyndon, H. Philip, B. Luigi, H. Simon. "A Faster Algorithm for Calculating Hypervolume", IEEE Trans. on Evol. Comput., vol. 10, no. 1, 2006.

[10] R. Alicia, "Búsqueda Tabú", pp. 5-9, 2005.

[11] L. Eugene, L. Jan, R. Kan, D. Symons, "The Travelling Salesman Problem", New York: Wiley, 1985.

[12] I. Hisao, N. Tsukamoto, Y Nojima. "Evolutionary Many-Objective Optimization: A Short Review", Proc. of 2008 IEEE Congress on Evolutionary Computation, 2424-2431, Hong Kong, 2008.

[13] V. Christian., B. Benjamín, B. Carlos. "A survey on multy-objective evolucionay algorithms for many-objective problems", Springer Science+Business Media, New York, 2014

[14] D. Kalyanmoy, S. Agrawal, P. Amrit, T. MEYARIVAN. "A fast and elitist Multi-Objective Genetic Algorithm: NSGA II", KanGAL, India, 2000.

[15] K. Kot, A. Pietrazco,"Heuristic methods for multi- criteria combinatorial optimization", Technical Report, Akademia Gorniczo-Hutnicza (AGH), Krakow, 2009.