

Protection with Quality of Service in optical WDM networks using Many-Objective Ant Colony Optimization

Tania Núñez*, Víctor Ayala*, Julio Paciello[†] y Benjamín Barán[†]

Universidad Nacional de Asunción, Paraguay

*{belen.zymanski,victor.a.ayala}@gmail.com

[†]{julio.paciello,bbaran}@pol.una.py

Abstract—This paper presents a new many-objective formulation of the Routing and Wavelength Assignment (RWA) problem in wavelength-routed optical WDM networks with protection, considering Quality of Service (QoS). A modification of the Multi-Objective Ant Colony System (MOACS) algorithm is proposed to solve the problem. Experimental results show that the proposed approach is a promising alternative compared to the original MOACS. The proposed algorithm is also compared to a Mixed Integer Linear Programming (MILP) implementation proving the results obtained are pretty closed, with a significant decrease in runtime.

Keywords—Multi-Objective Optimization, Many-Objective Optimization, Wavelength Division Multiplexing (WDM), Multi-Objective Ant Colony System (MOACS), Quality of Service (QoS), Routing and Wavelength Assignment (RWA).

I. INTRODUCCIÓN

El problema de ruteo y asignación de longitud de onda (*Routing and Wavelength Assignment*, RWA) es un problema de redes ópticas que típicamente consiste en maximizar el número de requerimientos (conexiones) atendidas. Para cada requerimiento de conexión, se debe establecer una ruta y asignar una longitud de onda que, en ausencia de conversores de longitud de onda¹, debe ser la misma a lo largo de todo el camino. El problema RWA puede ser clasificado como estático o dinámico. En el caso estático, las demandas de conexión son conocidas *a priori* y se resuelve en tiempo de diseño; mientras que en el caso dinámico, las demandas son atendidas al momento que van llegando a la red, sin conocer el futuro de las demandas.

En las redes ópticas, la falla en la comunicación como interrupción accidental de un enlace o fallo de un dispositivo de red, afectaría una gran cantidad de información transmitida. En la actualidad, se proponen dos técnicas para el problema de las fallas: la protección y la restauración de conexiones [1]. En el esquema de protección, los recursos de respaldo o *backup* son reservados con anticipación para cada conexión, antes de ocurrir la falla. Por el contrario, en el esquema de restauración, el camino de respaldo o *backup* es calculado dinámicamente después de ocurrir la falla.

¹Un convertor de longitud de onda es un dispositivo capaz de conmutar datos desde un puerto de entrada en una longitud de onda (λ_1) a un puerto de salida con otra longitud de onda (λ_2). Ayuda a reducir la probabilidad de bloqueo de la red.

El esquema de protección puede ser clasificado en protección de conexión (o de *lightpath*) y protección de enlace [2]. En el caso de la protección de conexión, un camino principal y un camino de respaldo son pre-calculados para cada conexión. En la protección de enlace, cada enlace que forma el camino principal es protegido por recursos de respaldo separados.

Los esquemas de protección también pueden ser clasificados en protección dedicada y protección compartida [2]. En la protección dedicada, los recursos de respaldo son utilizados para una sola conexión, y no son compartidos con los caminos de respaldo de otras conexiones. En la protección compartida, los recursos de un camino de respaldo pueden ser compartidos con otros caminos de respaldo.

Este trabajo propone dar solución al problema RWA estático con protección de conexiones en una red WDM donde no se disponen de conversores de longitud de onda en los nodos, mediante la aplicación de un algoritmo de optimización multi-objetivo basado en colonias de hormigas (*Multiobjective Ant Colony Optimization*, MOACO); específicamente, se propone la variante MOACS [3] dada su reconocida eficacia en la resolución de problemas de varios objetivos [4], [5] y [6]. Además, se implementa el concepto de Calidad de Servicio (*Quality of service*, QoS) para cada conexión, consistente en un conjunto de clasificaciones posibles que determinan una calidad mínima de servicio para las conexiones demandadas.

El problema RWA con protección de conexiones ha sido estudiado en trabajos anteriores. En la mayoría de estos trabajos ya publicados, se trata de atender la mayor cantidad de demandas minimizando los recursos de red utilizados. Sridharan y Somani en [7] resuelven el problema de protección de conexiones considerando diferenciación de servicios, buscando maximizar las ganancias (aceptar la mayor cantidad de demandas posibles) en la red y formulando el problema como un MILP (*Mixed Integer Linear Programming*). En [8], Zang y Mukherjee resuelven el problema de protección de conexiones bajo la restricción de ductos, en donde los caminos de trabajo y caminos de respaldo no pueden tener ductos (enlaces) comunes. En el trabajo de Arteta et al. [9], se implementan varios algoritmos heurísticos para resolver el problema RWA, optimizando simultáneamente dos objetivos: el número de saltos y el número de conversiones de longitudes de onda. En la tesis de Arévalos [6] se resuelve el problema

MRWA (*Multicast RWA*) bajo restricciones de capa óptica mediante algoritmos de colonias de hormigas multi-objetivas. En el trabajo de Rodas-Britez y Pinto-Roa [20] se presentó un esquema de QoS basado en la probabilidad de recuperación tras la falla simple de enlace, y se implementó un algoritmo genético que calcula los caminos de trabajo de las demandas de conexión.

El trabajo está organizado como sigue: En la Sección II se explica el esquema de clasificaciones utilizado para implementar QoS y se muestra la formulación matemática del problema. En la Sección III se explica el algoritmo de optimización multi-objetivo basado en colonia de hormigas y las heurísticas propuestas. En la Sección IV se muestra el pseudocódigo de los algoritmos implementados. En la Sección V se presentan los resultados de las pruebas experimentales y por último, en la Sección VI se concluye el trabajo realizado.

II. FORMULACIÓN DEL RWA CON QoS

II-A. Calidad de servicio

El esquema de clasificación de QoS implementado en este trabajo, que se basa en el trabajo propuesto por Kim y Lumetta [10], se define como sigue:

1. **Priority Class (Clase 'A')**: La protección dedicada utiliza este tipo de servicio y alcanza el mayor nivel de protección. Pueden ser de dos tipos: (i) protección 'A₊', en donde el tráfico es enviado tanto por el camino principal como por el camino de respaldo, y (ii) protección 'A₁', en donde el tráfico se envía por el camino principal, y sólo se cambia al camino de respaldo cuando ocurre una falla.
2. **Protected Class (Clase 'B')**: La Protección compartida (1:N), la cual establece que varios caminos pueden ser protegidos por un único canal de *backup*, utiliza este tipo de servicio. Las rutas de respaldo son calculadas, pero los conmutadores a lo largo del camino no son pre-configurados. Una vez que hay una falla en la red, una sola conexión podrá ser respaldada por un camino de *backup* específico.
3. **Reroutable Class (Clase 'C')**: Este tipo de servicio no ofrece recursos de protección, pero cuando ocurre un fallo, se realiza el mejor esfuerzo para realizar un redireccionamiento y recuperar el tráfico de la conexión.
4. **Unprotected Class (Clase 'D')**: Este tipo de servicio no ofrece protección de tráfico ante fallas.
5. **Pre-emptable Class (Clase 'E')**: Las conexiones de este tipo no poseen camino de trabajo ni de respaldo. Las conexiones con este servicio pueden utilizar como camino de trabajo los recursos de respaldo de las Clases 'A' (1:1) o 'B' (1:N) para reducir costos. Consecuentemente, al existir una falla que obliga a una comunicación de clase 'A' o 'B' a usar su *backup*, el tráfico de clase 'E' se puede perder por ser el de menor prioridad.

II-B. Formulación matemática

Para expresar la formulación matemática del problema, consideramos la siguiente nomenclatura:

$G(N, E, W)$ grafo no dirigido de la red donde serán encaimadas las demandas, donde N es el conjunto de nodos de la red, E el conjunto de enlaces y W el conjunto de longitudes de onda disponibles.

$|\cdot|$ operador de cardinalidad de un conjunto.

q_i calidad de servicio asignada a una demanda i , $q_i \in \{A_+, A_1, B, C, D, E\}$ donde: 'A₊' corresponde a la calidad de servicio de tipo A 1+1, 'A₁' corresponde a la calidad de servicio de tipo A 1:1, 'B' al tipo B, 'C' al tipo C, 'D' al tipo D, y 'E' al tipo E;

D conjunto de demandas de conexión, donde cada demanda de conexión $d_i(s_i, n_i, q_i)$ queda unívocamente caracterizada por un identificador de demanda i (ID), que está asociado a un nodo fuente $s_i \in N$, un nodo destino $n_i \in N$ y la calidad de servicio q_i .

w_e cantidad de longitudes de onda utilizadas en el enlace e considerando tanto los caminos de trabajo como los de respaldo, donde $w_e \leq |W|$.

T_i camino de trabajo de una demanda $d_i(s_i, n_i, q_i)$, que utiliza m_i enlaces, dado por el conjunto de enlaces que forman el camino desde el nodo fuente s_i hasta el nodo destino n_i .

τ_i *lightpath* de la demanda $d_i(s_i, n_i, q_i)$, que incorpora la asignación de una única longitud de onda λ_i a cada enlace del camino de trabajo T_i , esto es: $\tau_i = (T_i, \lambda_i)$.

R_i camino de respaldo de una demanda $d_i(s_i, n_i, q_i)$, que utiliza m_i enlaces, dado por el conjunto de enlaces que forman el camino.

ρ_i *lightpath* de respaldo de la demanda $d_i(s_i, n_i, q_i)$, que incorpora la asignación de una única longitud de onda λ'_i a cada enlace del camino de respaldo R_i , esto es: $\rho_i = (R_i, \lambda'_i)$.

X posible solución que determina un conjunto de *lightpaths* de trabajo y de respaldo, a ser utilizado para atender al conjunto de demandas D . Notar que una solución X debe proponer un τ_i y posiblemente un ρ_i para cada demanda $d_i(s_i, n_i, q_i)$, esto es: $X = \{\tau_1, \rho_1, \tau_2, \rho_2, \dots, \tau_{|D|}, \rho_{|D|}\}$, donde eventualmente algunos conjuntos ρ_i (y/o τ_i) pueden ser vacíos.

Y conjunto Pareto generado por el algoritmo. $Y = \{X_1, X_2, \dots, X_{|Y|}\}$, donde X_i representa una solución no dominada.

Y' frente Pareto formado a partir del conjunto Pareto Y aplicando las funciones objetivo.

$Bloq_x$ conjunto de demandas no atendidas dentro de una solución X .

b valor que indica la cantidad máxima de demandas no atendidas que puede tener cualquier solución X .

$w_{avg} = \frac{\sum_{e \in E} w_e}{|E|}$ carga promedio de enlace para la red.

El problema RWA multi-objetivo considerando protección con calidad de servicio (QoS), se plantea a continuación como un problema de minimización de los siguientes 5 objetivos: **Objetivo 1:** Minimizar la cantidad de demandas no atendidas.

$$F_1 = |Bloq_x| \quad (1)$$

Objetivo 2: Minimizar la cantidad de saltos de los caminos de trabajo y de respaldo para cada demanda de conexión.

$$F_2 = \sum_{d_i \in D} (|T_i| + |R_i|) \quad (2)$$

Objetivo 3: Minimizar la carga total de la red, teniendo en cuenta los caminos de trabajo y de respaldo.

$$F_3 = \sum_{e \in E} w_e \quad (3)$$

Objetivo 4: Balanceo de carga estático para toda la red; minimizar la desviación media considerando todos los enlaces disponibles y sus cargas en función a los caminos de trabajo y de respaldo utilizados.

$$F_4 = \frac{\sum_{e \in E} abs(w_e - w_{avg})}{|E|} \quad (4)$$

La función $abs()$ devuelve el valor absoluto de la expresión.

Objetivo 5: Minimizar la carga máxima de enlace de manera a minimizar el uso máximo de los enlaces de la red.

$$F_5 = \max_{e \in E} \{w_e\} \quad (5)$$

Las siguientes restricciones definen el espacio factible del problema propuesto:

1. $w_e \leq |W|, \forall e \in E$
Las longitudes de onda utilizadas en cada enlace, no pueden superar la cantidad máxima $|W|$ de longitudes de onda disponibles.
2. Sea $d_i(s_i, n_i, q_i)$ una demanda de conexión de un par de nodos origen-destino (s_i, n_i) , entonces la solución X es válida si:
 $T_i \cap R_i = \emptyset, \forall i \in \{1, 2, \dots, |D|\}$
El camino de trabajo y el de respaldo no deben compartir enlaces.
3. Sean $d_i(s_i, n_i, q_i)$ y $d_j(s_j, n_j, q_j)$ demandas de conexión, entonces una solución X es válida si:
 $\tau_i \cap \tau_j = \emptyset, \forall i, j \in \{1, 2, \dots, |D|\} \wedge i \neq j$
i.e., dos caminos de trabajo no deben compartir enlaces con la misma longitud de onda, lo que ocurre si $T_i \cap T_j = \emptyset$ (no comparten enlaces) y/o $\lambda_i \neq \lambda_j$ (no utilizan la misma longitud de onda).
4. $|Bloq_x| \leq b$
Para que X sea solución del problema, solo puede dejar de atender hasta un máximo de b demandas.

A continuación, se detallan las restricciones propias de cada calidad de servicio q_i , para una demanda atendida $d_i(s_i, n_i, q_i)$:

5. Si $q_i = 'A_+'$, debe existir un τ_i y un ρ_i tal que:
 $\rho_i \cap \rho_j = \emptyset,$
 $\rho_i \cap \tau_j = \emptyset,$ y
 $\tau_i \cap \rho_j = \emptyset,$
 $\forall i \neq j.$

El camino de respaldo de la demanda i no comparte enlaces utilizando la misma longitud de onda con ningún otro camino de trabajo o respaldo.

6. Si $q_i = 'A_1'$, debe existir un τ_i y un ρ_i tal que:

$$\begin{aligned} \rho_i \cap \rho_j &= \emptyset, \\ \tau_i \cap \rho_j &= \emptyset, \text{ y} \\ \rho_i \cap \tau_j &= \emptyset \text{ si } q_j \neq 'E', \\ \forall i \neq j. \end{aligned}$$

El camino de respaldo de la demanda i puede compartir enlaces utilizando la misma longitud de onda con el camino de trabajo de otra demanda j si y solo si la calidad de servicio $q_j = 'E'$.

7. Si $q_i = 'B'$, debe existir un τ_i y un ρ_i tal que:

$$\begin{aligned} \tau_i \cap \rho_j &= \emptyset, \\ \rho_i \cap \rho_j &= \emptyset \text{ si } q_j \in \{ 'A_+', 'A_1' \}, \text{ y} \\ \rho_i \cap \tau_j &= \emptyset \text{ si } q_j \neq 'E', \\ \forall i \neq j. \end{aligned}$$

El camino de respaldo de la demanda i puede compartir enlaces utilizando la misma longitud de onda que el camino de respaldo de otra demanda j de tipo 'B' o la misma longitud de onda que el camino de trabajo de otra demanda j de tipo 'E'.

8. Si $q_i = 'C'$, debe existir un τ_i y se debe recordar que $\rho_i = \emptyset$ por definición de QoS 'C', por lo que se debe atender la siguiente restricción: $\tau_i \cap \rho_j = \emptyset, \forall i \neq j$
Si ocurre una falla, la red hace el mejor esfuerzo para encontrar el camino de respaldo.
9. Si $q_i = 'D'$, debe existir un τ_i mientras que $\rho_i = \emptyset$ por definición de QoS 'D', por lo que se debe atender la siguiente restricción: $\tau_i \cap \rho_j = \emptyset, \forall i \neq j$
Solo debe existir su camino de trabajo y no existe un camino de respaldo.
10. $q_i = 'E'$ puede existir un τ_i y $\rho_i = \emptyset$ por definición de QoS 'E', por lo que se debe atender la siguiente restricción: $\tau_i \cap \rho_j = \emptyset$ si $q_j = 'A_+', \forall i \neq j.$
Puede existir un camino de trabajo que utiliza recursos libres de la red y/o comparte recursos con los caminos de respaldo de demandas de tipo 'A_1' o 'B'.

Cabe aquí enfatizar que los problemas de optimización multi-objetivo con más de 3 objetivos, como el arriba propuesto, reciben el nombre específico de *Many-Objectives Optimization Problems* (MaOP) dado que son especialmente difíciles de resolver, como se explica en Von Lücken et al. [21]. En el mejor entender de los autores, este trabajo es el primero en plantear el problema RWA con protección basada en QoS como un MaOP.

Ejemplo

Para mejor comprensión de la formulación matemática presentada, sigue un ejemplo del problema RWA teniendo en cuenta las restricciones mencionadas más arriba.

Supongamos que tenemos la red de la Figura 1, con 3 longitudes de onda (i.e. $W = \{\lambda_1; \lambda_2; \lambda_3\}$) para cada enlace y se desean establecer las demandas de conexión de la Tabla I.

Para satisfacer las demandas, se podría utilizar la solución presentada en la Tabla II con los caminos de trabajo y de respaldo ilustrados en la Figura 2.

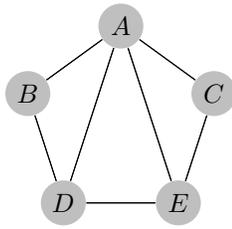


Figura 1. Red de ejemplo con 5 nodos y 7 enlaces

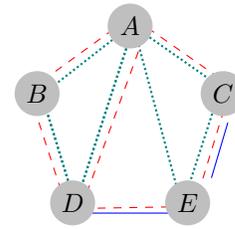


Figura 2. Red de ejemplo con las asignaciones de longitud de onda

 Tabla I
 TABLA DE DEMANDAS SOLICITADAS

Demanda	Origen	Destino	Tipo
1	A	C	'A ₊ '
2	B	C	'B'
3	B	A	'A ₁ '
4	D	C	'C'
5	A	D	'D'
6	B	A	'E'

Para la demanda $d_1(A, C, 'A_+')$, se selecciona el camino de trabajo $\tau_1=\{A, C\}$ y el camino de respaldo $\rho_1=\{A, E, C\}$, ambos con longitud de onda λ_1 . Como se puede observar en la Figura 2 y en la Tabla II, ningún otro camino de trabajo o respaldo comparte enlaces con la misma longitud de onda con el camino de trabajo ni con el respaldo de esta demanda.

Para la demanda $d_2(B, C, 'B')$, se selecciona el camino de trabajo $\tau_2=\{B, A, C\}$ y el camino de respaldo $\rho_2=\{B, D, E, C\}$, ambos con longitud de onda λ_2 .

Para la demanda $d_3(B, A, 'A_1')$, se selecciona el camino de trabajo $\tau_3=\{B, A\}$ y el camino de respaldo $\rho_3=\{B, D, A\}$ ambos con longitud de onda λ_1 .

Para las demandas $d_4(D, C, 'C')$ y $d_5(A, D, 'D')$, no son requeridos los caminos de respaldo por ser de clase 'C' y 'D' respectivamente.

En la demanda $d_6(B, A, 'E')$, para el camino de trabajo se podría establecer un nuevo enlace entre los nodos B y A con longitud de onda λ_3 , o bien se podría utilizar el camino de respaldo de la demanda d_3 . De esa manera, no se utilizan enlaces nuevos, i.e. se comparte el camino $\{B, D, A\}$ con longitud de onda λ_1 de la demanda d_3 .

Si se tuviera una nueva demanda $d_7(D, C, 'B')$, se podría hallar el camino de trabajo $\tau_7=\{D, A, C\}$ con longitud de

onda λ_3 , pero ya no se tendría una ruta con longitud de onda disponible para el camino de respaldo, por lo que no se podría atender esta demanda que quedaría así bloqueada. Alternativamente, como se puede observar en la Tabla II, la demanda d_2 de tipo 'B' sí dispone de enlaces para compartir. De esa manera, se utilizan los enlaces $\{D, E\}$ y $\{E, C\}$ con longitud de onda λ_2 para el camino de respaldo de la nueva demanda de tipo 'B' que sería entonces $\rho_7=\{D, E, C\}$.

Por último, si llegase una nueva demanda $d_8(B, C, 'A_1')$, no se podría satisfacer esta demanda ya que no se disponen de longitudes de onda libres en el enlace $\{E, C\}$ para hallar el camino de trabajo y respaldo, por lo que esta demanda quedaría sin ser satisfecha (bloqueada), i.e. $Bloq_x = \{d_8(B, C, 'A_1')\}$.

El cálculo de las funciones objetivo, considerando las primeras 6 demandas, ilustradas en la Tabla I quedaría como sigue:

$$F_1 = |Bloq_x| = |\{\emptyset\}| = 0,$$

$$F_2 = 16,$$

$$F_3 = 14,$$

$$F_4 = 0,285, \text{ y}$$

$$F_5 = 3;$$

y luego de recibir las demandas d_7 y d_8 , el cálculo de las funciones objetivo quedaría como:

$$F_1 = |Bloq_x| = |\{d_8(B, C, 'A_1')\}| = 1,$$

$$F_2 = 20,$$

$$F_3 = 16,$$

$$F_4 = 0,612, \text{ y}$$

$$F_5 = 3;$$

III. OPTIMIZACIÓN MULTI-OBJETIVO BASADA EN COLONIA DE HORMIGAS

La búsqueda de los caminos de trabajo y respaldo para cada demanda de conexión, con un tipo de servicio específico, se implementa utilizando una metaheurística de optimización basada en colonias de hormigas, conocida como ACO (*Ant Colony Optimization*) en su versión multi-objetivo (*Multi-objective Ant Colony Optimization, MOACO*) [5], para permitir optimizar varios objetivos simultáneamente.

El algoritmo utilizado fue el MOACS, definido en [3], considerando su reconocida capacidad de resolver en forma eficiente problemas de varios objetivos como fuera reconocido en [4], [5] y [6]. El algoritmo MOACS propuesto en este trabajo se describe mediante los algoritmos 1, 2, 3, 4 y 5 más abajo presentados. La principal diferencia del MOACS propuesto con respecto al MOACS definido en [3] se basa en

 Tabla II
 CAMINOS DE TRABAJO Y RESPALDO DE UNA POSIBLE SOLUCIÓN DEL EJEMPLO PRESENTADO

Demanda	Trabajo	L.Onda	Respaldo	L.Onda
1	A-C	λ_1	A-E-C	λ_1
2	B-A-C	λ_2 - - -	B-D-E-C	λ_2 - - -
3	B-A	λ_1	B-D-A	λ_1
4	D-E-C	λ_3 - - -	Sin respaldo	
5	A-D	λ_2 - - -	Sin respaldo	
6	B-D-A	λ_1	Sin respaldo	

Tabla III
ESTADO DE LA RED DE LA FIGURA 1

Origen	Destino	λ Libres
A	B	$\{\lambda_1, \lambda_3, \lambda_4\}$
A	C	$\{\lambda_3, \lambda_4\}$
A	D	$\{\emptyset\}$
A	E	$\{\lambda_1\}$
B	D	$\{\lambda_1\}$
C	E	$\{\lambda_3, \lambda_4\}$
D	E	$\{\lambda_3\}$

la utilización de la heurística H_2 , propuesta en este trabajo y explicada en la siguiente subsección, la cual busca seleccionar con mayor probabilidad a los nodos vecinos con mayor cantidad de longitudes de onda libres y comunes en todo el camino hallado.

III-A. Heurísticas consideradas

Se realizaron adaptaciones al Algoritmo MOACS, publicado por Barán y Schaerer en [3]. En el MOACS propuesto se aplicaron las siguientes heurísticas:

1. *Caminos más cortos (H_1):* Se utiliza la salida del algoritmo de Dijkstra [11] para la búsqueda del camino más corto (costo mínimo) desde un vértice origen al resto de los vértices en un grafo con pesos en cada arista. Para el problema tratado, se considera que todas las aristas tienen el mismo peso (igual a 1) y de esta forma obtenemos el camino más corto (o costo mínimo) entre todos los nodos de la red.
2. *Longitudes de onda comunes (H_2):* Algoritmo propuesto en este trabajo que calcula el porcentaje de longitudes de ondas libres y comunes desde un origen, con cada uno de los nodos vecinos. Con esto se trata de dar mayor peso a los nodos vecinos con los cuales el nodo origen mantiene la mayor cantidad de longitudes de onda comunes y así asegurar que al finalizar el camino, se pueda asignar alguna longitud de onda libre.

En el siguiente ejemplo del problema RWA, se describe el cálculo de H_2 tomando la red de la Figura 1 con $W = \{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}$. Supongamos que la red se encuentra en el estado que se muestra en la Tabla III y se requiere de una nueva conexión con origen en el nodo B y destino en el nodo C.

Inicialmente, se tiene un conjunto de longitudes de ondas comunes, $L_C = \{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}$. Se parte desde el origen B y se localizan los vecinos inmediatos, A y D. Para cada vecino, se verifica la cantidad de longitudes de ondas libres de los enlaces (A, B) y (B, D), $L_{libres}(A, B) = \{\lambda_1, \lambda_3, \lambda_4\}$ y $L_{libres}(B, D) = \{\lambda_1\}$ respectivamente. Luego se calcula la intersección entre cada conjunto L_{libres} con el conjunto L_C , entonces el valor de esta heurística se determina por $H_2 = \frac{|L_C \cap L_{libres}|}{|W|}$; por lo tanto, $H_2(A, B) = \frac{3}{4}$. Análogamente, $H_2(B, D) = \frac{1}{4}$. Estos valores calculados se

utilizan dentro en la Ecuación (7) del Algoritmo 5, para hallar el siguiente nodo. De esta forma, un nodo con mayor cantidad de longitudes de onda libres y comunes tiene una mayor probabilidad de ser seleccionado como siguiente nodo.

Una vez seleccionado el siguiente nodo, se calcula un nuevo valor de L_C como sigue: $L_C = L_C \cap L_{libres}$, siendo L_{libres} el conjunto de longitudes de onda libres del enlace seleccionado. Luego se realiza el mismo cálculo, tomando como nodo origen el nodo seleccionado en el paso anterior.

Si el nodo seleccionado fue el nodo A, el nuevo conjunto de longitudes de onda comunes queda como sigue $L_C = \{\lambda_1, \lambda_3, \lambda_4\}$. Se busca de nuevo los vecinos inmediatos aún no visitados y se vuelve a calcular H_2 con cada uno de ellos. El cálculo resultante es $H_2(A, D) = 0$, $H_2(A, E) = \frac{1}{4}$ y $H_2(A, C) = \frac{1}{2}$. Finalmente, se llegaría al nodo destino seleccionando, el nodo C. El conjunto de longitudes de onda comunes resultante queda de esta manera $L_C = \{\lambda_3, \lambda_4\}$, y cualquiera de sus valores puede ser utilizado para atender el requerimiento utilizando el camino hallado.

III-B. Técnicas de asignación de longitud de onda

Una vez encontrado un camino, ya sea de trabajo o de respaldo, se le debe asignar una longitud de onda. Se dispone de una cantidad W de longitudes de ondas en cada enlace de la red. Como se menciona en párrafos anteriores, una de las restricciones del problema es que a lo largo de un camino se mantenga la misma longitud de onda. La heurística adoptada para la resolución del problema es *Most-Used*, descrita en [15].

IV. DESCRIPCIÓN DE ALGORITMOS

El Algoritmo 1 muestra el pseudocódigo del Algoritmo MOACS, definido en [3]. La evaporación de feromonas se realiza disminuyendo el valor de cada τ_{ij} en cada iteración del algoritmo según la Ecuación (6), en donde ρ es el coeficiente de evaporación y $\Delta\tau$ se calcula como $\Delta\tau = \frac{1}{\sum_{k=1}^5 F_k}$, en donde los objetivos están debidamente normalizados.

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \rho \cdot \Delta\tau, \rho \in (0, 1] \quad (6)$$

El Algoritmo 2 presenta el pseudocódigo del algoritmo propuesto en este trabajo, en donde los caminos de trabajo y respaldo son calculados dependiendo de la calidad de servicio solicitada para cada demanda.

Para la construcción de los caminos de trabajo en el Algoritmo 3, en el cálculo de W_{C_L} de una demanda de tipo ' E ', se toman también como longitudes de onda libres todas aquellas asignadas a los caminos de respaldo ya encontrados considerando demandas de tipo ' A_1 ' y ' B '. El método *LongOndaLibres(i, j)* devuelve el conjunto de longitudes de onda libres en el enlace (i, j) que se agrega al camino.

Para la construcción de los caminos de respaldo en el Algoritmo 4, en el cálculo de W_{C_R} de una demanda de tipo ' B ', se toman como longitudes de onda libres todas

Algoritmo 1 MOACS

```

1: function MOACS( $\alpha, \beta, D, \rho, \tau_0, \omega, G(N, E, W)$ )  $\triangleright$ 
   Retorna  $Y_{true}$ 
2:    $\tau = \text{INICIALIZARFEROMONAS}(\tau_0)$ ;
3:   while not condicionDeParada do
4:      $j = j + 1$ ;  $\triangleright$  Contador de generaciones
5:     for  $\theta = 1, \omega$  do  $\triangleright \omega$  es el número de hormigas
6:        $S_D = \text{CONSTRUIRSOLUCION}(\alpha, \beta, D, \rho, G)$ ;
7:        $\text{EVALUARSOLUCION}(S_D)$ ;
8:        $Y_{true} = \text{ACTUALIZAR}Y_{known}(S_D)$ ;
9:     end for
10:     $\tau'_0 = \text{CALCULARTAUPRIMA}()$ ;
11:    if  $\tau'_0 > \tau_0$  then  $\triangleright$  Se encontró un mejor Conjunto
        Pareto
12:       $\tau_0 = \tau'_0$ ;
13:       $\tau = \text{INICIALIZARFEROMONAS}(\tau_0)$ ;
14:    else
15:       $\tau = \text{ACTUALIZARFEROMONAS}(Y_{true})$ ;
16:    end if
17:  end while
18:  return  $Y_{true}$ 
19: end function

```

Algoritmo 2 Construir Solucion - Propuesto

```

1: function CONSTRUIRSOLUCION( $\alpha, \beta, D, \rho, G$ )  $\triangleright$  Retorna
    $S_D$ 
2:    $S_D = \emptyset$ ;
3:   for cada Demanda  $d \in D$  do
4:      $L_d = \emptyset$ ;  $\triangleright$  Camino de trabajo
5:      $R_d = \emptyset$ ;  $\triangleright$  Camino de respaldo
6:      $L_d = \text{CONSTRUIRTRABAJO}(\alpha, \beta, D, \rho, G)$ ;
7:     if CALIDAD( $d$ ) in  $\{A_+, A_1, B\}$  then
8:        $R_d = \text{CONSTRUIRRESPALDO}(\alpha, \beta, D, \rho, G, L_d)$ 
9:     end if
10:     $S_D = S_D \cup \{L_d, R_d\}$ ;
11:  end for
12:  return  $S_D$ ;
13: end function

```

El valor S_D es un conjunto de caminos de trabajo y respaldo de cada demanda $d \in D$.

aquellas asignadas a los caminos de respaldo ya encontrados considerando demandas de tipo 'B' cuyos caminos de trabajo no tengan enlaces comunes con el camino de trabajo de la demanda actual.

Considerando el Algoritmo 5, la selección del siguiente estado a transitar utiliza las heurísticas mencionadas en la Sección III-A, mediante la Ecuación (7). En este algoritmo, el parámetro q_0 indica la probabilidad de seleccionar como siguiente estado a aquel que presente la mayor probabilidad en forma determinística o seleccionar en base a la distribución de probabilidades (Regla Pseudo-aleatoria)[3].

Algoritmo 3 Construir Trabajo

```

1: function CONSTRUIRTRABAJO( $\alpha, \beta, D, \rho, G$ )  $\triangleright$  Retorna
    $L_d$ 
2:    $L_d = \emptyset$ ;  $L_d = L_d \cup \{d.origen\}$ ;
3:    $W_{C_L} = W$ ;  $\triangleright$  Longitudes de onda comunes de  $L_d$ 
4:   while destinoNoAlcanzado do
5:      $j = \text{OBTENERSIGUIENTENODO}(\alpha, \beta, D, \rho, G, W_{C_L})$ ;
6:     if  $j == \emptyset$  then
7:        $d.atendida = \text{false}$ ; return;
8:     end if
9:      $L_d = L_d \cup \{j\}$ ;
10:    if onlineUpdate then
11:       $\text{ONLINEUPDATE}(i, j)$ ;
12:    end if
13:     $W_{C_L} = W_{C_L} \cap \text{LONGONDALIBRES}(i, j)$ ;
14:    if  $W_{C_L} = \emptyset$  then
15:       $d.atendida = \text{false}$ ; return;
16:    end if
17:     $i = j$ ;
18:  end while
19:  return  $L_d$ ;
20: end function

```

Algoritmo 4 Construir Respaldo

```

1: function CONSTRUIRRESPALDO( $\alpha, \beta, D, \rho, G, L_d$ )  $\triangleright$ 
   Retorna  $R_d$ 
2:    $R_d = \emptyset$ ;  $R_d = R_d \cup \{d.origen\}$ ;
3:    $W_{C_R} = W$ ;  $\triangleright$  Longitudes de onda comunes de  $R_d$ 
4:   while destinoNoAlcanzado do
5:      $j = \text{OBTENERSIGUIENTENODO}$ 
           ( $\alpha, \beta, D, \rho, G, W_{C_L}, L_d$ );
6:     if  $j == \emptyset$  then
7:        $d.atendida = \text{false}$ ; return;
8:     end if
9:      $R_d = R_d \cup \{j\}$ ;
10:    if onlineUpdate then
11:       $\text{ONLINEUPDATE}(i, j)$ ;
12:    end if
13:     $W_{C_R} = W_{C_R} \cap \text{LONGONDALIBRES}(i, j)$ ;
14:    if  $W_{C_R} = \emptyset$  then
15:       $d.atendida = \text{false}$ ; return;
16:    end if
17:     $i = j$ ;
18:  end while
19:  return  $R_d$ ;
20: end function

```

Algoritmo 5 Obtener Siguiete Nodo

```

1: function OBTENERSIGUIENTENODO
   ( $\alpha, \beta, D, \rho, G, W_{CL}, L_d$ )           ▷ Retorna  $j$ 
2:   Seleccionar aleatoriamente  $q$          ▷  $q, q_0 \in (0, 1]$ 
3:   if  $q > q_0$  then
4:     Seleccionar el nodo  $j$  con mayor  $p_{ij}$ ;
5:   else
6:     Seleccionar  $j$  según la distribución de probabilidades  $p_{ij}$ ;
7:   end if
8:   return  $j$ ;
9: end function

```

$$p_{ij} = \begin{cases} \frac{\tau_{ij}^\alpha [(H_1^{r_{ij}})(H_2^{1-r_{ij}})]^\beta}{\sum_{v \in N_i} \tau_{iv}^\alpha [(H_1^{r_{iv}})(H_2^{1-r_{iv}})]^\beta} & \text{si } j \in N_i \\ 0 & \text{en otro caso} \end{cases} \quad (7)$$

siendo N_i el conjunto de vecinos del nodo actual, τ la matriz de feromonas, H_1 y H_2 las heurísticas 1 y 2 respectivamente, α y β parámetros que reflejan la importancia relativa de las feromonas y las visibilidades y r un valor entre $[0,1]$ que indica la importancia relativa entre las visibilidades.

V. RESULTADOS EXPERIMENTALES

En esta sección se presentan los resultados de las pruebas experimentales realizadas en el contexto de este trabajo. Se montó un ambiente de pruebas donde se realizó la ejecución del algoritmo con diferentes topologías de red y diferentes demandas aleatorias. La implementación de los algoritmos MOACS se realizaron con el lenguaje de programación Java (JDK 1.6) respectivamente. Las librerías generadas se ejecutaron en un entorno Linux. Las soluciones generadas por el algoritmo fueron evaluadas por un conjunto de métricas de desempeño, que se explican más abajo.

V-A. Ambiente de Pruebas

Para las pruebas experimentales, se utilizó un servidor con las siguientes características:

CPU: 8 núcleos, 2.5 GHz cada núcleo

Memoria: 16 GB

Sistema Operativo: Ubuntu Server 12.04 LTS 64 bits.

En la Tabla IV se resumen los parámetros utilizados para el algoritmo MOACS.

V-B. Topologías de red utilizadas

Las pruebas experimentales se realizaron sobre las siguientes topologías de Red: (a) la NSF (*National Science Foundation*) [16], (b) la red de USA [17], (c) la red de Francia [19] y (d) la red NTT de Japón [18]. En la Figura 3 se muestra de forma gráfica las topologías utilizadas.

Tabla IV
TABLA DE PARÁMETROS DEL ALGORITMO MOACS

Parámetro/Valor	Descripción
$h = 10$	Cantidad de hormigas
$gen = 5000$	Cantidad de generaciones del algoritmo
$\rho = 0,1$	Según Ecuación (6).
$\alpha = 1$	Según Ecuación (7)
$\beta = 3$	Según Ecuación (7)
$r = (h - i)/h$	Según Ecuación (7), Donde i es la hormiga actual
$\tau_0 = 0,2$	Valor inicial de la tabla de feromonas.

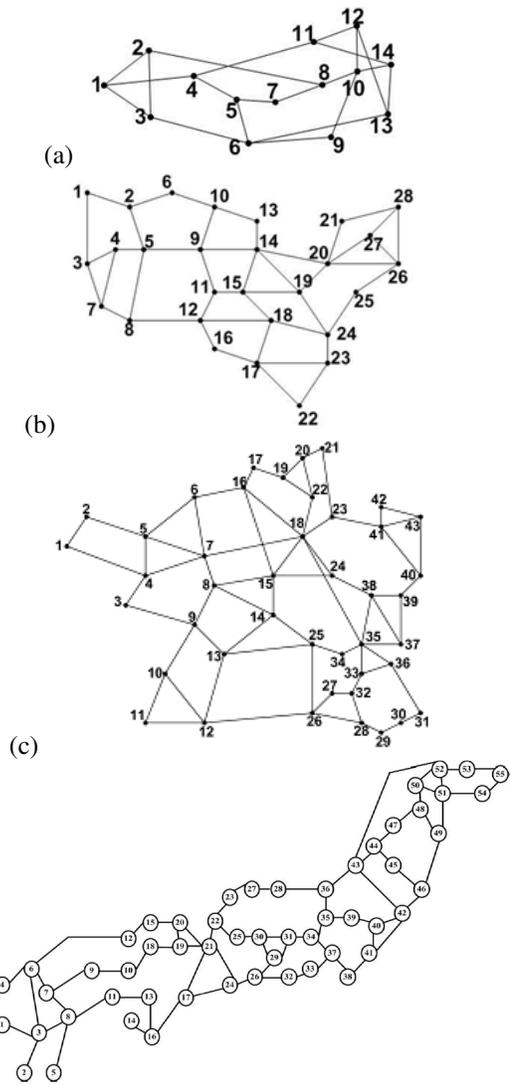


Figura 3. Topologías de Redes: (a) La red NSF con 14 nodos y 21 enlaces. (b) La red de USA con 28 nodos y 45 enlaces. (c) La red de Francia con 43 nodos y 70 enlaces. (d) La red NTT de Japón con 55 nodos y 72 enlaces.

Tabla V
CANTIDAD DE DEMANDAS UTILIZADAS PARA CADA RED

Red	Demanda Alta	Demanda Media
NSF	45	25
USA	25	15
Francia	20	10
NTT	10	5

V-C. Ajustes de parámetros

Para la selección de demandas, se realizó una prueba con cada topología de red. El valor de bloqueo b utilizado fue de cero, lo que significa que el algoritmo debía atender todas las demandas del conjunto. El valor de W utilizado en las pruebas fue de 20. Los pasos en cada prueba fueron los siguientes:

1. generar 50 demandas D aleatorias y utilizarlas en el algoritmo propuesto;
2. si no se obtienen soluciones, se restan 5 demandas al conjunto D y se vuelve al paso 1;
3. si se obtienen soluciones, se vuelve a resolver el algoritmo con ese mismo conjunto D por 10 veces para conocer el nivel de aleatoriedad;
4. si en el paso 3 se encuentran soluciones en cada una de las 10 corridas, se toma el conjunto D como el conjunto de Demanda Alta para la instancia de red. En caso contrario, se vuelve al paso 2.

Para el conjunto de Demanda Media, se seleccionó de forma aleatoria aproximadamente 50 % de las demandas del conjunto de Demanda Alta. En la Tabla V se muestra el número de demandas consideradas para cada topología de red en los experimentos que se reportan a continuación.

V-D. Métricas de desempeño

Para medir el rendimiento de los algoritmos, se utilizaron dos métricas de desempeño: El Hipervolumen (HV) del frente Pareto y la Distancia (DI) al frente Pareto Óptimo. La métrica del hipervolumen fue propuesta originalmente en el trabajo de [12], que lo define como "el tamaño del espacio objetivo cubierto por las soluciones no nominadas". En este trabajo, será mejor el conjunto Pareto cuyo frente calculado tenga un mayor valor de HV. La segunda métrica que se utilizó es la de Distancia (DI) [13]. La métrica DI calcula la distancia promedio al Frente Pareto aproximado para cada instancia de prueba considerada. Será mejor el conjunto Pareto cuyo frente calculado tenga un menor valor en esta métrica DI.

V-E. Cálculo de Y_{true} para cada red

Para el problema en cuestión, aun no son conocidas las soluciones óptimas (Y_{true}), por lo tanto, se realizaron varias corridas previas del algoritmo. Los resultados se guardaron y se procesaron para luego utilizarse en comparaciones posteriores.

Así, de la unión de todas las soluciones obtenidas considerando todas las corridas realizadas, se eliminaron las soluciones dominadas quedando en el conjunto las mejores

Tabla VI
TABLA COMPARATIVA ENTRE MOACS PROPUESTO MONO-OBJETIVO Y EL MILP

	MILP	Tiempo(s)	MOACS	Tiempo(s)	Demandas
NSF	1,335	1290	1,586	25	12
USA	0,744	15151	0,942	90	10
Francia	0,649	16935	0,693	195	8
NTT	0,390	18740	0,425	330	8

soluciones que conforman así la mejor aproximación conocida, la Y_{true} .

V-F. Comparación entre el MOACS propuesto y el MILP

Para validar los resultados obtenidos con el algoritmo MOACS propuesto, se realizó la implementación del problema con Programación Lineal Entera Mixta (MILP) [14]. Para el efecto, se utilizó el *IBM ILOG CPLEX Optimization Studio* en la versión 12.6.0². Como el MILP resuelve problemas mono-objetivos, para comparar ambas técnicas se realizó una suma ponderada de los objetivos normalizados del MOACS propuesto, de manera de volverlo mono-objetivo, como lo requiere MILP. La Ecuación (8) muestra el cálculo de la suma ponderada, utilizada en el MILP y en el MOACS propuesto mono-objetivo, en donde F_1, F_2, F_3, F_4 y F_5 ya se encuentran normalizados (Pueden tomar valores del intervalo $[0,1]$); por lo tanto F puede tomar un valor entre 0 y 5.

$$F = \sum_{k=1}^5 F_k \quad (8)$$

En la Tabla VI se muestra el resultado al utilizar el algoritmo propuesto mono-objetivo versus el MILP, considerando la cantidad de demandas mostradas en la última columna. El valor de la columna MILP corresponde al resultado de aplicar la Ecuación 8. El valor de la columna MOACS corresponde al promedio de 30 corridas del algoritmo MOACS propuesto mono-objetivo. Se estableció el valor de $|W|$ igual a 5. Como era de esperarse, en la Tabla VI se observa que en todos los casos el tiempo de ejecución del MILP sobrepasa ampliamente el tiempo del algoritmo propuesto (miles de segundos vs. algunos pocos segundos), llegando a tiempos muy grandes a medida que el tamaño de la red crece.

El MILP para instancias pequeñas y pocas demandas logra encontrar la solución óptima. De los experimentos se comprobó que los resultados obtenidos por el MOACS propuesto se aproximan considerablemente al óptimo obtenido con el MILP, con una destacable disminución del tiempo de ejecución. Cabe resaltar que en la medida en que crece la cantidad de demandas o el tamaño de la red, el MILP ya no logra terminar su cómputo en un tiempo razonable

²<http://www-03.ibm.com/software/products/en/ibmilogcpleoptistud>

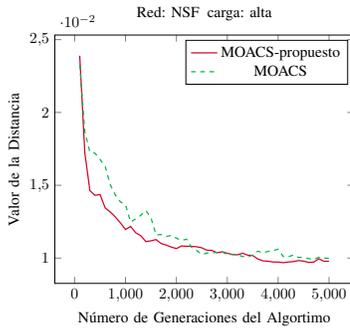


Figura 4. Gráfico comparativo entre el Algoritmo MOACS propuesto y el MOACS original para la red NSF con carga alta utilizando la métrica de Distancia

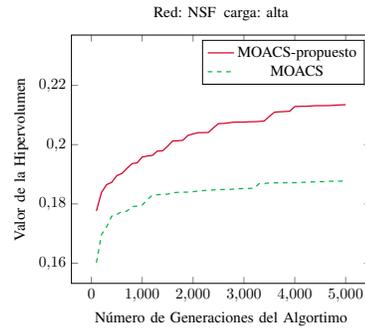


Figura 6. Gráfico comparativo entre el Algoritmo MOACS propuesto y el MOACS original para la red NSF con carga alta utilizando la métrica de Hipervolumen

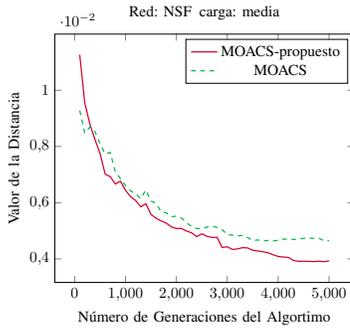


Figura 5. Gráfico comparativo entre el Algoritmo MOACS propuesto y el MOACS original para la red NSF con carga media utilizando la métrica de Distancia

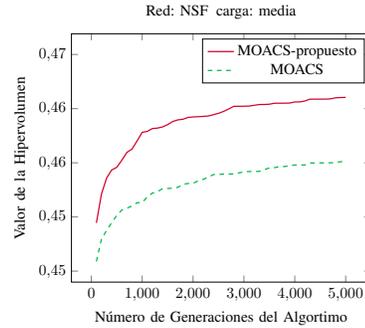


Figura 7. Gráfico comparativo entre el Algoritmo MOACS propuesto y el MOACS original para la red NSF con carga media utilizando la métrica de Hipervolumen

V-G. Comparación entre el MOACS original y el MOACS propuesto

Para validar los resultados, se realizaron unas pruebas comparativas entre el algoritmo MOACS original propuesto en [3] y el MOACS propuesto en este trabajo. Las pruebas fueron realizadas sobre cada una de las 4 redes de la Figura 3. Para cada topología de red y cada tipo de demanda (Alta y Media), se realizaron 30 corridas en total, 15 para calcular el Ytrue y otras 15 para calcular las métricas de desempeño. Los valores de las métricas en los gráficos de resultados (Figuras 4 al 19) corresponden al promedio de todas las corridas realizadas. El tiempo promedio de ejecución depende de la instancia y carga de la red, teniéndose un total aproximado de 338 horas de pruebas experimentales.

En las figuras 4 y 5 se presenta la evolución de la distancia DI en función de las generaciones del algoritmo, teniendo en cuenta una carga de red alta para la Figura 4, una carga de red media para la Figura 5 y considerando en ambos casos la instancia de red NSF de la Figura 3. Se observa que el MOACS propuesto encuentra soluciones cercanas al óptimo en menor tiempo que el MOACS original. Esto se debe a que la heurística H_2 (*Longitud de ondas comunes*) busca favorecer estados del problema que permitan hallar una longitud de onda factible una vez encontrado todo el camino. Es decir, asegura la continuidad de una longitud de onda a lo largo del

camino, lo cual permite a la hormiga encontrar una solución factible. Además, cabe resaltar que casi en la totalidad de las generaciones el algoritmo propuesto mejora al MOACS en su versión original, obteniendo un mejor valor para la métrica DI.

La evolución del Hipervolumen en función de las generaciones del algoritmo, para la instancia de red NSF, se presenta en las Figuras 6 y 7. A diferencia de la métrica de distancia que disminuye a medida que se encuentran mejores soluciones, el hipervolumen HV mejora cuando su valor aumenta. Para esta instancia de red, el valor del hipervolumen de las soluciones halladas por el MOACS propuesto crece con mayor rapidez que para el MOACS y supera en gran medida al algoritmo original propuesto en [3].

Teniendo en cuenta la instancia de red USA de la Figura 3, la evolución de la distancia DI se presenta en las Figuras 8 y 9. Se observa un comportamiento similar a la red NSF, donde el MOACS propuesto tiene mejor evaluación. En esta instancia de red, para ambas cargas alta y media, se puede apreciar que el algoritmo propuesto supera durante todas las generaciones al algoritmo original.

En las Figuras 10 y 11 se observa la evolución del hipervolumen HV para una carga de red alta y media respectivamente, en ambos casos para la instancia de red USA. Una vez más se observa que el hipervolumen en el MOACS propuesto

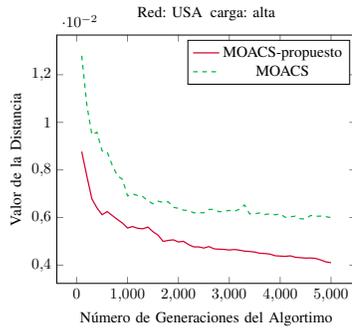


Figura 8. Gráfico comparativo entre el Algoritmo MOACS propuesto y el MOACS para la red de USA con carga alta utilizando la métrica de Distancia

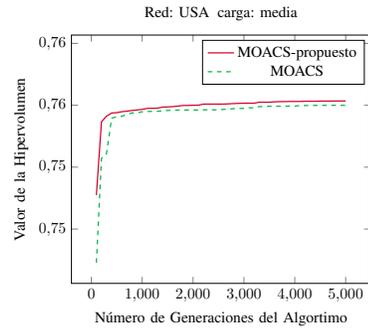


Figura 11. Gráfico comparativo entre el Algoritmo MOACS propuesto y el MOACS para la red de USA con carga media utilizando la métrica de Hipervolumen

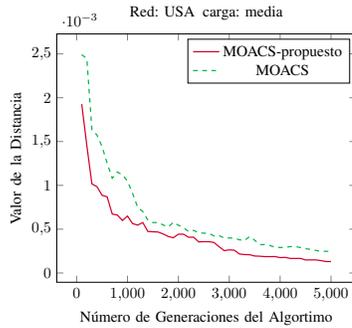


Figura 9. Gráfico comparativo entre el Algoritmo MOACS propuesto y el MOACS para la red de USA con carga media utilizando la métrica de Distancia

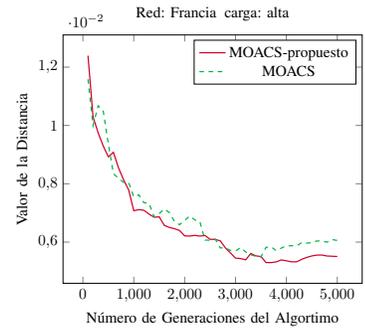


Figura 12. Gráfico comparativo entre el Algoritmo MOACS propuesto y el MOACS para la red de Francia con carga alta utilizando la métrica de Distancia

toma valores altos en pocas generaciones del algoritmo, en comparación con el MOACS original, que requiere de más generaciones para hallar un Conjunto Pareto que evalúe mejor en esta métrica.

En las Figuras 12 y 13 se muestra la evolución de la métrica distancia DI para cargas alta y media respectivamente, tomando la instancia de red de Francia de la Figura 3. Nuevamente vemos que el MOACS propuesto tiene mejor evaluación que el MOACS original en la métrica DI al terminar las generaciones;

no obstante, los resultados de ambos algoritmos son similares para esta instancia.

En los gráficos del hipervolumen HV de la red de Francia, (Figura 14 para carga alta y Figura 15 para carga media), nuevamente se observa una mejor evaluación en la métrica para el MOACS propuesto. Por ejemplo, en el gráfico de carga media observamos que, aunque ambos algoritmos terminan con los mismos valores en la métrica, el MOACS propuesto tiene mejores valores en menor número de generaciones, lo

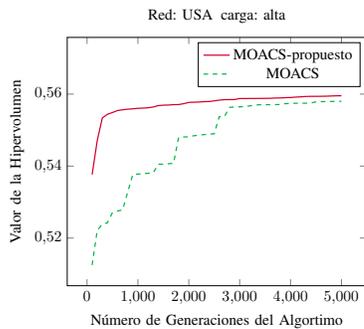


Figura 10. Gráfico comparativo entre el Algoritmo MOACS propuesto y el MOACS para la red de USA con carga alta utilizando la métrica de Hipervolumen

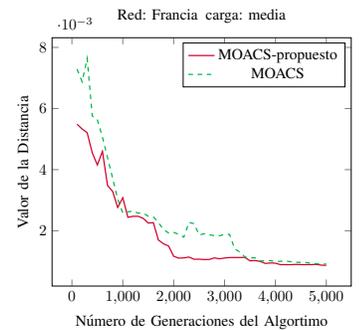


Figura 13. Gráfico comparativo entre el Algoritmo MOACS propuesto y el MOACS para la red de Francia con carga media utilizando la métrica de Distancia

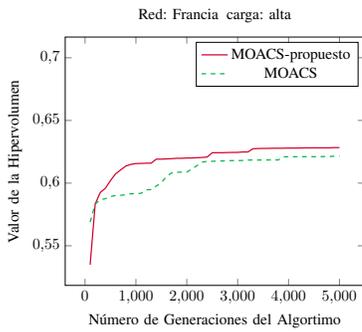


Figura 14. Gráfico comparativo entre el Algoritmo MOACS propuesto y el MOACS para la red de Francia con carga alta utilizando la métrica de Hipervolumen

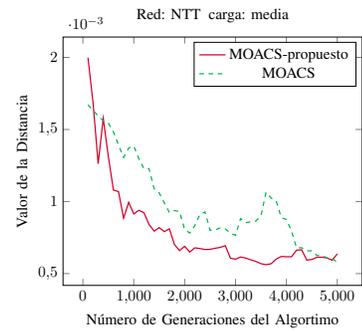


Figura 17. Gráfico comparativo entre el Algoritmo MOACS propuesto y el MOACS para la red NTT con carga media utilizando la métrica de Distancia

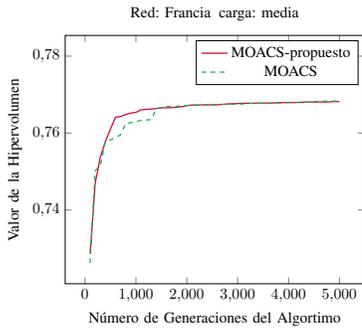


Figura 15. Gráfico comparativo entre el Algoritmo MOACS propuesto y el MOACS para la red de Francia con carga media utilizando la métrica de Hipervolumen

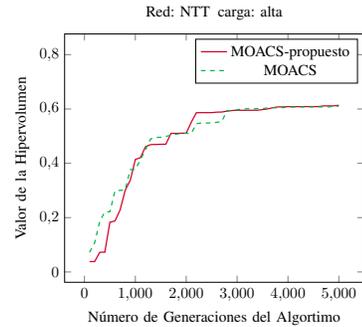


Figura 18. Gráfico comparativo entre el Algoritmo MOACS propuesto y el MOACS para la red NTT con carga alta utilizando la métrica de Hipervolumen

cual es una propiedad interesante en caso de no contarse con suficiente tiempo de cálculo y se desean buenas aproximaciones.

En la Figura 16 se observa la evolución de la métrica de distancia DI en función de las generaciones del algoritmo, para una carga de red alta, y en la Figura 17 para una carga de red media. También se observa una mejor evaluación para el MOACS propuesto, en donde se observa de hecho que sus valores se aproximan más al Y_{true} estimado conforme se

explica en la Sección V-E.

Por último, se observa en las Figuras 18 y 19 un comportamiento similar al ya descrito en la evolución de la métrica de hipervolumen HV en ambos algoritmos con carga alta, y una mejor evolución del MOACS propuesto con carga media.

Teniendo en cuenta la métrica de distancia DI, en líneas generales, el MOACS propuesto logra aproximarse al Y_{true} en pocas generaciones. Esto implica que se requiere menor tiempo de ejecución del algoritmo para obtener resultados próximos al óptimo calculado. En la métrica hipervolumen

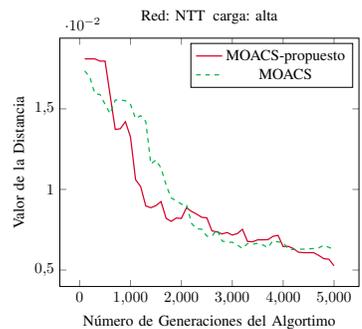


Figura 16. Gráfico comparativo entre el Algoritmo MOACS propuesto y el MOACS para la red NTT con carga alta utilizando la métrica de Distancia

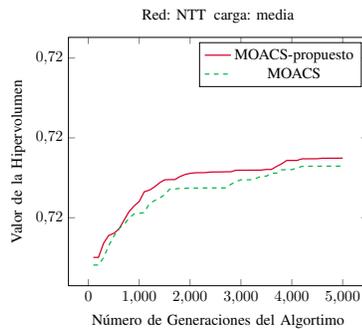


Figura 19. Gráfico comparativo entre el Algoritmo MOACS propuesto y el MOACS para la red NTT con carga media utilizando la métrica de Hipervolumen

HV, el MOACS propuesto supera en casi todos los casos al algoritmo MOACS propuesto en [3]. Así mismo, en casi todos los casos se observa un crecimiento rápido del valor de la métrica de hipervolumen HV al utilizar la propuesta de este trabajo. Con esto, podemos afirmar que el algoritmo propuesto obtiene resultados prometedores a la hora de compararlo con la propuesta original del MOACS.

VI. CONCLUSIÓN

En este trabajo se ha implementado un algoritmo de optimización multi-objetivo basado en colonia de hormigas para resolver por primera vez el problema RWA en redes WDM con 5 objetivos (*Many-Objective Problem*), teniendo en cuenta la calidad de servicio (QoS) solicitada por el cliente, minimizando simultáneamente los siguientes 5 objetivos: demandas no atendidas, cantidad de saltos, carga total de la red, desviación media de carga de la red y carga máxima de enlace, constituyéndose así este trabajo en la primera propuesta *many-objective* del problema RWA con protección basado en calidad de servicio.

Considerando los resultados obtenidos en la prueba comparativa con el MILP, se verifica que el algoritmo propuesto resuelve el problema en tiempos de ejecución lógicamente menores, y la calidad de las soluciones se acerca considerablemente a la solución óptima calculada por el MILP. Esto permite concluir que el algoritmo propuesto representa una alternativa factible para resolver el problema RWA con QoS en grafos con mayor cantidad de nodos y atendiendo una tasa elevada de demandas, instancias en las que ya no se podría utilizar el algoritmo MILP.

En los escenarios en los cuales la resolución con MILP era inviable, se compararon las soluciones calculadas por el algoritmo propuesto contra las mejores soluciones proveídas por el algoritmo MOACS de Barán y Schaefer [3] demostrando una clara superioridad del algoritmo propuesto en este trabajo.

Como trabajo futuro se plantea implementar otras estrategias evolutivas de manera a poder comparar el algoritmo basado en colonias de hormigas para el problema planteado con otras alternativas.

Así mismo, analizando la métrica de hipervolumen HV, se observó que el algoritmo converge a sus mejores soluciones aproximadamente desde la generación 2000, lo cual nos indica que se podrían considerar operadores de preservación de la diversidad que permitan explorar otras zonas del espacio de búsqueda de manera a evitar una convergencia prematura. Se plantea también como trabajo futuro la implementación de estrategias de exploración como la degradación o reinicialización de los niveles de feromonas, de manera a evitar el estancamiento del proceso de búsqueda.

Por último debemos mencionar que este trabajo aporta al estado del arte una nueva heurística (longitudes de onda comunes) para el RWA, con cuya implementación se obtuvo un buen desempeño como lo demuestran los resultados experimentales presentados.

REFERENCIAS

- [1] Son Ngo, S. N., Jiang, X., y Horiguchi, S. (2005, Setiembre). *Dynamic lightpath protection in wdm optical networks using ant-based mobile agents*. In Intelligent Agent Technology, IEEE/WIC/ACM International Conference on (pág. 51-57). IEEE.
- [2] Ramamurthy, S., Sahasrabudhe, L., y Mukherjee, B. (2003). *Survivable WDM mesh networks*. Journal of Lightwave Technology, 21(4), 870.
- [3] Barán, B., y Schaefer, M. (2003, Febrero). *A Multiobjective Ant Colony System for Vehicle Routing Problem with Time Windows*. In Applied Informatics (pág. 97-102).
- [4] García-Martínez, C., Cerdón, O., y Herrera, F. (2004). *An empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP*. In Ant Colony Optimization y Swarm Intelligence (pág. 61-72). Springer Berlin Heidelberg.
- [5] Paciello, J., Martínez, H., Lezcano, C., y Barán, B. (2006). *Algoritmos de Optimización multi-objetivos basados en colonias de hormigas*. Proceedings of CLEI (Vol. 6).
- [6] Arévalo S. (2011). *Comunicaciones Ópticas Enrutamiento Multicast Multiobjetivo bajo restricciones de Capa Óptica*. (Tesis de grado). Universidad Nacional de Asunción. Paraguay.
- [7] Sridharan, M., y Somani, A. K. (2000, Setiembre). *Revenue maximization in survivable WDM networks*. In Opticom 2000 (pág. 291-302). International Society for Optics y Photonics.
- [8] Zang, H., Ou, C., y Mukherjee, B. (2003). *Path-protection routing y wavelength assignment (RWA) in WDM mesh networks under duct-layer constraints*. IEEE/ACM Transactions on Networking (TON), 11(2), 248-258.
- [9] Arteta, A., Barán, B., y Pinto, D. (2007, Octubre). *Routing and wavelength assignment over WDM optical networks: a comparison between MOACOs and classical approaches*. In Proceedings of the 4th international IFIP/ACM Latin American conference on Networking (pág. 53-63). ACM.
- [10] Kim, S. I., y Lumetta, S. S. (2006). *Protection Based QoS in WDM Mesh Networks*. UIUC Center for Reliable y High-Performance Computing Technical Report CRHC-06-01.
- [11] Dijkstra, E. W. (1959). *A note on two problems in connexion with graphs*. Numerische mathematik, 1(1), 269-271.
- [12] Zitzler, E., y Thiele, L. (1998, Enero). *Multiobjective optimization using evolutionary algorithms: a comparative case study*. In Parallel problem solving from nature PPSN V (pág. 292-301). Springer Berlin Heidelberg.
- [13] Zitzler, E., Deb, K., y Thiele, L. (2000). *Comparison of multiobjective evolutionary algorithms: Empirical results*. Evolutionary computation, 8(2), 173-195.
- [14] Garrido, R. S. (1993). *Programación lineal: Metodología y problemas*. Editorial Tebar.
- [15] Zang, H., Jue, J. P., y Mukherjee, B. (2000). *A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks*. Optical Networks Magazine, 1(1), 47-60.
- [16] Cui, X., Lin, C., y Wei, Y. (2003, Octubre). *A multiobjective model for QoS multicast routing based on genetic algorithm*. In Computer Networks y Mobile Computing, 2003. ICCNMC 2003. 2003 International Conference on (pág. 49-53). IEEE.
- [17] Gao, X. (2009). *Design and analysis of effective routing and channel scheduling for wavelength division multiplexing optical networks*. (Doctoral dissertation, University of Central Florida Orlando, Florida).
- [18] Baran, B., y Sosa, R. (2000). *A new approach for AntiNet routing*. In Computer Communications y Networks, 2000. Proceedings. Ninth International Conference on (pág. 303-308). IEEE.
- [19] Pinto-Roa, D. P., Barán, B., y Brizuela, C. A. (2011). *Routing and wavelength converter allocation in WDM networks: a multi-objective evolutionary optimization approach*. Photonic Network Communications, 22(1), 23-45.
- [20] Rodas-Britez, M. D., y Pinto-Roa, D. P. (2014, September). *Quality of protection on WDM networks: A quantitative paradigm based on recovery probability*. In Computing Conference (CLEI), 2014 XL Latin American (pág. 1-11). IEEE.
- [21] Von Lücken, C., Barán, B., y Brizuela, C. (2014). *A survey on multi-objective evolutionary algorithms for many-objective problems*. Computational Optimization y Applications, 58(3), 707-756.