

# Equipos de Algoritmos de Hormigas

## Una nueva propuesta para Optimización Multiobjetivo

Christian Lezcano, Diego Pinto y Benjamín Barán

Universidad Nacional de Asunción – Facultad Politécnica  
P.O. Box 2111 - Paraguay  
{clezcano, dpinto, bbaran}@pol.una.py  
<http://www.pol.una.py>

**Resumen** — Este trabajo propone un nuevo enfoque sobre Equipos Distribuidos de Algoritmos de Hormigas para la resolución de problemas de optimización multiobjetivos. El método propuesto ha demostrado un significativo efecto cooperativo de los algoritmos combinados en el equipo, logrando robustez en la resolución de distintos problemas combinatorios. Experimentalmente, el enfoque propuesto ha verificado un notable equilibrio entre diferentes figuras de desempeño en instancias bi-objetivas del *Traveling Salesman Problem* (TSP), del *Quadratic Assignment Problem* (QAP) y del *Vehicle Routing Problem with Time Windows* (VRPTW). La robustez y el equilibrio logrados con esta nueva propuesta se basan en una novedosa combinación de algoritmos seleccionados a partir de comparaciones multiobjetivas en el sentido Pareto de métricas de desempeño medidas en tiempo de ejecución.

**Palabras Claves** — Equipo de Algoritmos, Computación Distribuida, Optimización Multiobjetivo y Colonia de Hormigas.

## 1 Introducción

El enfoque de optimización multiobjetivo conocido como *Multiobjective Optimization Problem* (MOP) se puede definir, en palabras, como el problema de hallar una solución que satisfaciendo ciertas restricciones, optimiza varias funciones objetivos simultáneamente. Estas funciones usualmente en conflicto conforman una descripción del criterio de desempeño [1]. MOP es ampliamente tratado con distintos paradigmas tales como *Multi-Objective Evolutionary Algorithm* (MOEA) [1], *Multi-Objective Ant Colony Optimization* (MOACO) [2, 3], *Multi-Objective Particle Swarm Optimization* (MOPSO) [4] y más recientemente *Multi-Objective Adaptive Memetic Algorithm* (MOAMA) [5]. Los enfoques anteriores han sido utilizados exitosamente para resolver problemas de alta complejidad logrando soluciones óptimas o cercanas a la óptima. Por su parte, el teorema de “*No Free Lunch - NFL*” [6], establece que en promedio, todos los algoritmos tienen el mismo desempeño. En consecuencia, el desarrollo de un *Team Algorithm* (TA) [7] es una alternativa válida para alcanzar en promedio una alta robustez sobre distintos tipos de problemas de optimización.

Este trabajo propone un novedoso enfoque de TA, basado en algoritmos MOACO, el cual fue comparado con una reciente propuesta del estado del arte para TAs [7]. Se realizaron pruebas experimentales sobre diferentes *instancias bi-objetivas del Traveling Salesman Problem* (TSP), del *Quadratic Assignment Problem* (QAP) y del

*Vehicle Routing Problem with Time Windows* (VRPTW). Para medir la calidad de los algoritmos fueron consideradas distintas figuras de desempeño multiobjetivas como la distancia, la distribución, la extensión, el error y la calidad de las soluciones [8, 9].

## 2 Optimización Multiobjetivo

Un *Multiobjective Optimization Problem* (MOP) generalmente consiste en un conjunto de  $n$  variables de decisión, un conjunto de  $k$  funciones objetivos y un conjunto de  $m$  restricciones [1]. Las funciones objetivos y las restricciones son funciones de las variables de decisión. Por lo tanto, en un MOP generalmente se optimiza:

$$z = f(x) = (f_1(x), f_2(x), \dots, f_k(x)) \quad (1)$$

sujeto a

$$g(x) = (g_1(x), g_2(x), \dots, g_m(x)) \leq 0 \quad (2)$$

donde  $x = (x_1, x_2, \dots, x_n) \in X$  es un vector de decisión,  $X$  denota el espacio de decisión de  $f(x)$ ,  $z = (z_1, z_2, \dots, z_k) \in Z$  es un vector objetivo mientras que  $Z$  denota el espacio objetivo de  $f(x)$ . Un *Conjunto de Soluciones Factibles*  $\Omega \subset X$  es definido como un conjunto de vectores de decisión que satisfacen las restricciones dadas en (2). Sean dos soluciones  $u, v \in \Omega$ , se dice que  $u$  *domina* a  $v$  (denotado como  $u \succ v$ ) si  $u$  es mejor o igual que  $v$  en cada función objetivo y estrictamente mejor en al menos un objetivo. Así, se define el *Conjunto Pareto Óptimo* como  $P^* = \{u \in \Omega \mid \nexists v \in \Omega \text{ tal que } v \succ u\}$  mientras que el espacio objetivo de  $P^*$  es conocido como *Frente Pareto Óptimo*, denotado como  $F^*$ , i.e.  $F^* = f(P^*)$ .

Las medidas de desempeño mostradas en la Tabla 1 son utilizadas en este trabajo con dos objetivos. El primero es medir la calidad de las soluciones obtenidas con los diferentes enfoques. El segundo objetivo es medir el desempeño de los diferentes algoritmos, considerando simultáneamente varias figuras de mérito (o desempeño). En función a este desempeño en un sentido Pareto, se seleccionan los algoritmos a ser utilizados en la siguiente iteración de un proceso evolutivo.

**Tabla 1.** Figuras de Desempeño

Nombre	Símbolo	Ref.	Observación
Distancia	$M_1$	[8]	Define la distancia promedio entre un Frente Pareto calculado y un Frente Pareto real $F^*$ .
Distribución	$M_2$	[8]	Estima la distribución promedio de las soluciones a lo largo de un Frente Pareto calculado.
Extensión	$M_3$	[8]	Indica la distancia Euclidiana entre las dos soluciones extremas de un Frente Pareto calculado.
Error	$M_4$	[9]	Calcula el porcentaje de soluciones de un Frente Pareto calculado que no corresponden a un Frente Pareto real $F^*$ .
Soluciones Aportadas	$M_5$	[9]	Indica el porcentaje de soluciones aportadas de un Frente Pareto calculado a un Frente Pareto real $F^*$ .

### 3 Multi-Objective Ant Colony Optimization

*Multi-Objective Ant Colony Optimization* (MOACO) es una técnica de búsqueda inspirada en el comportamiento natural de las colonias de hormigas [2, 3]. Diversos trabajos demostraron empíricamente la eficiencia de los MOACOs en la resolución de un MOP [2, 3, 7, 17-23]. En este trabajo, se consideró la utilización de distintos algoritmos MOACO, representativos del estado del arte [4, 7], los cuales son expuestos en la Tabla 2.

**Tabla 2.** Algoritmos MOACO

<i>Símbolo</i>	<i>Algoritmos</i>	<i>Ref.</i>
MAS	<i>Multiobjective Ant System</i>	[7]
BIANT	<i>Bi-criterion Ant</i>	[17]
BIMC	<i>Bi-criterion Multi Colony</i>	[17]
PACO	<i>Pareto-Ant Colony Optimization</i>	[18]
COMP	<i>COMPETants</i>	[19]
MOACS	<i>Multi-Objective Ant Colony System</i>	[20]
M3AS	<i>Multiobjective Max-Min Ant System</i>	[21]
MOAQ	<i>Multiple Objective Ant Q</i>	[22]
MOA	<i>Multiobjective Omicron ACO</i>	[23]

### 4 Team Algorithms

La computación distribuida ha ganado importancia con la proliferación de computadoras personales interconectadas en redes, proveyendo mayor potencia de cálculo a bajos costos. Básicamente, un *Team Algorithm* (TA) combina diferentes algoritmos para resolver en conjunto un mismo problema [7, 10, 11, 12]. En consecuencia, resulta natural la implementación paralela de un TA, sobre un sistema distribuido. Por ejemplo, cada algoritmo puede ser ejecutado en un procesador distinto de un sistema distribuido. En el modelo maestro-esclavo [7] adoptado para este trabajo, este mapeo *algoritmo-computador* es realizado por un proceso *Maestro*.

Por su parte, la Computación Evolutiva estuvo asociada al concepto de paralelismo desde sus inicios [14]. Sin embargo, solo recientemente los modelos distribuidos han sido integrados para la resolución de MOPs. Von Lucken et al. han tratado el problema de optimización de bombeo de agua utilizando asynchronous parallel MOEAs (p-MOEA) [11]. En dicho enfoque un solo algoritmo MOEA es ejecutado sobre los distintos procesadores de un sistema distribuido. Fernández et al. extienden dicho trabajo en un Team Algorithm de MOEAs (TA-MOEA), considerando diferentes algoritmos MOEAs en cada procesador [12]. Por su parte, varios enfoques paralelos de ACO fueron propuestos en [15]. Recientemente, Paciello et al. propusieron un Team Algorithm basado en algoritmos MOACO (TA-MOACO) [7]. Cabe destacar, que la implementación del TA-MOACO es similar al TA-MOEA.

#### 4.1 Método General

En forma general, cada proceso esclavo busca las mejores soluciones Pareto de alguna región particular del espacio de búsqueda. Estas soluciones son transmitidas al proceso maestro, el cual genera un conjunto de soluciones Pareto del espacio explorado por el TA.

Dependiendo del enfoque [7,12], cada algoritmo es seleccionado con algún criterio para formar parte de un proceso esclavo en la siguiente iteración. Normalmente, los criterios de desempeño son tomados como proporcionales a la calidad de las soluciones aportadas. Típicamente, la calidad de las soluciones aportadas es definida con una de las figuras de desempeño presentadas en la Tabla 1. Esta interacción entre el proceso maestro y los procesos esclavos es realizada hasta que alguna condición de parada sea satisfecha. Para facilitar el entendimiento de los algoritmos implementados la siguiente nomenclatura es presentada:

- $P$ : Conjunto de computadores disponibles en la red, i.e.  $P = \{p_i | i=1,2,\dots,/P|\}$ .
- $H$ : Conjunto de algoritmos, donde  $H = \{h_j | j=1,2,\dots,/H|\}$ .
- $W$ : Conjunto de procesos esclavos, donde  $W = \{w_k \in P \times H | w_k = (p_i, h_j), p_i \in P, h_j \in H, k=1,2,\dots,/W|\}$ ; note que  $P \times H$  es un producto cartesiano y  $W \subset P \times H$ .
- $P_{wk}$ : Conjunto Pareto calculado por un proceso esclavo  $w_k \in W$ .
- $\Gamma$ : Conjunto formado por los Conjuntos Pareto  $P_{wk}$ , i.e.:  $\Gamma = \bigcup_{w_k \in W} P_{wk}$
- $\mathcal{Y}$ : Frentes Pareto asociados a  $\Gamma$ , donde  $\mathcal{Y} = \{F_{wk} | F_{wk} = f(P_{wk})\}$ . Note que  $\mathcal{Y} = f(\Gamma)$ .
- $M_{wk}$ : Figuras de desempeño Pareto asociadas a un proceso esclavo  $w_k \in W$ , esto es,  $M_{wk} = \{M_r(F_{wk}) | r=1,2,\dots,/M_{wk}|\}$ ; note que en este trabajo  $/M_{wk}| \leq 5$  (ver Tabla 1).
- $\Psi$ : Figuras de desempeño calculadas para los procesos  $W$ , donde  $\Psi = \{M_{wk} | w_k \in W\}$ .
- $P_{known}$ : Conjunto Pareto conocido que aproxima a  $P^*$ .
- $F_{known}$ : Frente Pareto conocido que aproxima a  $F^*$ , donde  $F_{known} = f(P_{known})$

Para este trabajo, se asume que cada proceso  $w$  es ejecutado sobre un procesador  $p_i$  diferente, i.e. en todo instante  $|W| = |P|$ .

Considerando la nomenclatura anterior, el Algoritmo 1 presenta un procedimiento genérico para un proceso maestro, mientras que el Algoritmo 2 corresponde al procedimiento del  $k$ -ésimo proceso esclavo.

---

#### Algorithm 1: Generic Master Process

---

**Input:**  $P, H$

**Output:**  $P_{known}$

---

- 1: Initialize configuration parameters and  $P_{known} = \emptyset$
  - 2:  $W = \text{Initial Mapping}(P, H)$
  - 3: Initial Launch Slave Processes ( $W$ )
  - 4: **while** (stop condition is no achieved) **do**
  - 5:      $\Gamma = \text{Receive solutions from Slaves Processes}(P)$
  - 6:     Update  $P_{known}(\Gamma)$
  - 7:      $\Psi = \text{Calculated Performance of Slaves}(\mathcal{Y}, F_{known})$
  - 8:      $W = \text{Mapping}(P, H, \Psi)$
  - 9:     Send to Slave Processes ( $W, P_{known}$ )
  - 10: **end while**
  - 11: Finish Slave Process
  - 12: **return**  $P_{known}$
-

**Algorithm 2:** *Generic k-th Slave Process***Input:**  $H$ **Output:**  $P_k$ 


---

```

1: Receive initial configuration from Master Process ( $h_k$ )
2:  $P_{known} = \emptyset$ 
3: while (stop condition is no achieved) do
4:    $P_k = \text{run algorithm}(h_k, P_{known})$ 
5:   Send best solutions to Master Process ( $P_k$ )
6:   Receive information from Master Process ( $h_k, P_{known}$ )
7: end while

```

---

El método propuesto en [7] considera una clasificación de los procesos esclavos utilizando una sola figura de mérito. En este trabajo llamamos a dicha propuesta TA-MOACO o sencillamente *simple-TA* (*s-TA*). Básicamente, en cada iteración del maestro, el algoritmo del peor proceso esclavo es sustituido por el algoritmo del mejor proceso esclavo. Note que se trata de un enfoque de muerte súbita, donde un algoritmo con mal desempeño al inicio de la corrida nunca más es seleccionado.

Es claro que los TA recientemente propuestos en la literatura [7, 12] no consideran simultáneamente los diversos factores que inciden en la convergencia, distribución y extensión de la calidad de las soluciones asociadas a un conjunto de soluciones Pareto Óptimas. La utilización de una sola figura de mérito es una apreciación parcial del verdadero desempeño de los procesos esclavos. Por ende, es de esperar que considerando varias figuras de mérito en forma simultánea, se potenciaría la capacidad de búsqueda del TA intentando equilibrar las distintas figuras de méritos (Tabla 1). En consecuencia, este trabajo propone por primera vez la utilización simultánea de varias figuras de desempeño (en un contexto multiobjetivo) asociadas a la selección de los algoritmos de un TA. De esta forma, en cada iteración es obtenida una clasificación en el sentido Pareto de los procesos esclavos, lo cual define qué algoritmos serán utilizados en la siguiente iteración. Básicamente, una clasificación  $R$  es realizada sobre el conjunto de los procesos esclavos  $w \in W$  considerando sus figuras de desempeños, i.e.  $R = \{R_l \mid l=1, 2, \dots, |R|\}$  donde  $R_1 \succ R_2 \succ \dots \succ R_{|R|}$ . La clasificación por frentes Pareto es realizada utilizando el conocido algoritmo *fast non-dominated sorting* propuesto en [24].

## 4.2 TA Elitista

El *elitist-TA* (*e-TA*) es una extensión natural del *s-TA* en la que se consideran procesadores de iguales características. En este enfoque, el proceso maestro selecciona aleatoriamente un proceso esclavo  $w_i$ , perteneciendo al mejor frente  $R_1$  y otro proceso esclavo  $w'$  correspondiente al peor frente  $R_{|R|}$ . Seguidamente, el algoritmo asociado a  $w_i$  reemplaza al algoritmo de  $w'$ . De esta forma, los algoritmos con mejor desempeño van reemplazando paulatinamente a algoritmos con peores desempeños. En cada iteración, un solo algoritmo es reemplazado. Este procedimiento del *e-TA* es presentado en el siguiente Algoritmo 3, el cual corresponde a la rutina “*Mapping*” del proceso maestro dado en el Algoritmo 1. Note que si todos los procesos logran el mismo desempeño entonces  $|R|=1$ .

**Algorithm 3:** Mapping for *e-TA***Input:**  $P, H, \Psi$ **Output:**  $W$ 


---

```

1:  $R =$  Pareto Classification of results from Slaves ( $\Psi$ )
2: if  $|R| > 1$  then
3:   Randomly select a slave process  $w_i \in R_l$ 
4:   Randomly select a slave process  $w' \in R_{|R|}$ 
5:   Replace algorithm  $h_{w'}$  with  $h_{w_i}$ 
6: end if
7: return  $W$ ;

```

---

**4.3 TA Probabilista**

Es claro que el enfoque *e-TA* presenta una alta presión elitista en el método de selección de los algoritmos. Además, un algoritmo reemplazado pierde toda posibilidad de ser seleccionado en una iteración futura. Lo anterior nos sugiere que el *e-TA* puede tener un estancamiento con algún algoritmo cuyo desempeño inicial fue bueno. Sin embargo, estos algoritmos podrían en general ser malos en el largo plazo. Claramente esto podría provocar un pobre desempeño del TA.

Con el objeto de evitar la pérdida de diversidad de algoritmos en los procesos esclavos, se propone un enfoque cuya selección de algoritmos se basa en una distribución de probabilidades acorde con la clasificación de procesos esclavos  $R$ . Este enfoque es llamado *probabilistic-TA* (*p-TA*) y la probabilidad de seleccionar un proceso esclavo puede ser calculada conforme a la siguiente ecuación (3).

$$p(R_l) = \frac{2 * (|R| - l + 2)}{(|R| + 1) * (|R| + 2)} \text{ con } l=1, 2, \dots, |R|+1 \quad (3)$$

Todos los procesos esclavos pertenecientes a una clasificación  $R_l$  ( $l=1, 2, \dots, |R|$ ) reciben la misma probabilidad de selección. En este enfoque, los algoritmos no asignados a ningún proceso esclavo son clasificados como  $R_{|R|+1}$ . De esta forma, se garantiza que todo algoritmo tenga una probabilidad de selección mayor a cero en cualquier iteración, independientemente de su desempeño en iteraciones pasadas. El esquema de un *p-TA* es presentado en el siguiente Algoritmo 4.

**Algorithm 4:** Mapping for *p-TA***Input:**  $P, H, \Psi$ **Output:**  $W$ 


---

```

1:  $R =$  Pareto Classification of results from Slaves ( $\Psi$ )
2: Calculates  $p(R_l)$  using equation (3)
3: for ( $i = 1$  to  $|W|$ ) do
4:    $w' =$  Select a Slave Process of  $R$  according to  $p(R_l)$ 
5:   Replace algorithm  $h_{w_i}$  with  $h_{w'}$ 
6: end for
7: return  $W$ ;

```

---

## 5 Procedimiento Experimental

Los experimentos realizados muestran el comportamiento de los enfoques propuestos *e-TA* y *p-TA* cuando son aplicados a distintos problemas combinatorios en un contexto multiobjetivo. Se han seleccionado las siguientes instancias bi-objetivas: a) *kroab100* y *kroac100*, instancias del TSP, b) *qapUni.75.0.1* y *qapUni.75.p75.1*, instancias del QAP y c) *c101* y *rc101*, instancias del VRPTW [7].

El conjunto de algoritmos MOACO considerados fueron  $H = \{\text{BIANT, BIMC, COMP, MOACS, M3AS, MOAQ, MOA, PACO, MAS}\}$ , mencionados en la Tabla 2. Para facilitar la comparación, los parámetros utilizados para los algoritmos MOACO fueron considerados como similares a los presentados en [7].

Los TA fueron implementados en C++ (Compiler GNU GCC) con PVM library 3.4.5 [13], sobre un sistema operativo Linux, distribución Fedora 4. Los experimentos fueron ejecutados en una red de 10 estaciones de trabajo homogéneas (9 computadoras esclavas y 1 computadora maestra), con procesadores de 2 GHz y 512 MB de memoria RAM, i.e.  $P = \{p_1, p_2, \dots, p_9 \mid p_i \equiv p_j \forall i, j\}$ .

Para la clasificación Pareto de los procesos esclavos ( $R$ ), solo se han considerado como figuras de desempeño la Extensión y el número de Soluciones Aportadas, i.e.  $|M_w|=2$ . Dado que el *s-TA* solo considera una sola métrica ( $|M_w|=1$ ), se ha decidido utilizar el número de Soluciones Aportadas acorde a la propuesta original [7].

### 5.1 Diseño Experimental

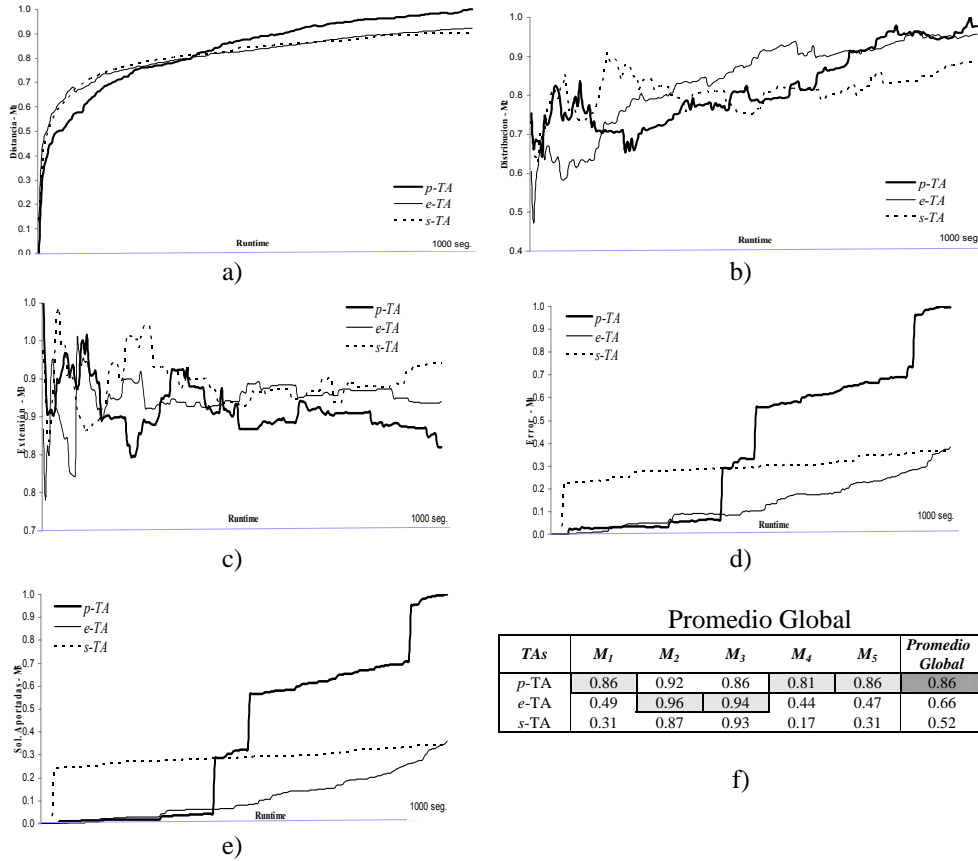
Definamos el conjunto  $A = \{s\text{-TA}, e\text{-TA}, p\text{-TA}\}$  de algoritmos implementados en la pruebas. Los siguientes pasos describen el esquema de las pruebas experimentales aplicadas a cada instancia:

1. Fueron realizadas 10 corridas para cada algoritmo  $h_i \in A$ , obteniéndose los siguientes Frentes Pareto  $F_i = \{F_{i1}, F_{i2}, \dots, F_{i10}\}$ .
2. Un conjunto Pareto Óptimo conocido  $F_{\text{known}}$  fue calculado mediante la eliminación de las soluciones dominadas del conjunto unión de todas las soluciones encontradas.
3. Cada Frente Pareto  $F_{ij}$  fue comparado con el Frente Pareto conocido  $F_{\text{known}}$ , obteniéndose un conjunto de figuras desempeño (ver Tabla 1).

El criterio de parada para cada corrida fue de 1000 segundos de corrida, adoptado experimentalmente.

### 5.2 Convergencia y Resultados Globales

En este análisis se presentan los resultados transitorios para cada enfoque. Básicamente, las Figuras 1.a a 1.e representan el comportamiento de las distintas figuras de desempeño muestreadas cada 10 segundos. Los resultados se encuentran promediados y normalizados, de forma tal que el peor valor corresponde a cero y el mejor valor corresponde a 1. Estos resultados corresponden a promedios generales sobre todas las instancias de pruebas utilizadas, considerando todas las corridas.



**Fig. 1.** Convergencia de las figuras de desempeños asociadas a los algoritmos. Convergencia de: a) Distancia, b) Distribución, c) Extensión, d) Error y e) Soluciones Aportadas. En f) se presenta los promedios globales logrados por cada enfoque.

En la Figura 1.a puede observarse que la Distancia promedio  $M_1$  al  $F_{known}$  inicialmente es pobre y a medida que avanza el ciclo evolutivo esta converge a mejores valores. Claramente el *p-TA* presenta un pobre desempeño en comparación a *e-TA* y *s-TA*, para tiempos cortos de ejecución. Sin embargo, cuando la búsqueda avanza el *p-TA* logra acortar su distancia a  $F_{known}$ , siendo finalmente el de mejor desempeño promedio. Esto indica que el *e-TA* y el *s-TA* presentan convergencia prematura a óptimos locales.

La medida de Distribución  $M_2$  nos indica que no existe una clara superioridad de algún enfoque para intervalos cortos de corridas (ver Figura 1.b). Sin embargo, para tiempos largos de ejecución el *p-TA* y el *e-TA* presentan una clara superioridad sobre el *s-TA*. Note que las tendencias indican que todos los enfoques propuestos tienden a mejorar sus distribuciones, mientras que el *s-TA* en promedio mantiene el mismo valor de distribución, durante la mayor parte del ciclo evolutivo.

En cuanto a la Extensión  $M_3$ , se verifica una muy leve superioridad del *s-TA* con respecto a los enfoques propuestos. Esto se encuentra asociado al hecho de que el *s-*

*TA* no utiliza la sinergia de las soluciones globales, de  $F_{known}$ , generadas por el *TA* en su búsqueda de nuevas soluciones.

El Error  $M_4$  y las Soluciones Aportadas  $M_5$  presentados (Figura 1.d y 1.e), nos indican un estancamiento del *s-TA* y el *e-TA*. Note que el *s-TA* tiene un rápido ascenso inicial pero con un estancamiento prematuro. Por su parte el *p-TA* presenta una lenta convergencia superando ampliamente el desempeño alcanzado por los otros enfoques, considerando tiempos largos de ejecución.

Por otra parte, una tabla sobre el ranking obtenido considerando el promedio alcanzado sobre todas las corridas en todas las instancias de problemas resueltos es dado en la Figura 1.f. Puede observarse la clara robustez del *p-TA* en comparación con los demás algoritmos. Esto se condice con un equilibrio alcanzado sobre las distintas figuras de desempeño. Note que para la Distancia, la Distribución y la Extensión, el *p-TA* ha obtenido valores significativos. Incluso considerando al Error y las Soluciones Aportadas, los mejores valores corresponden al *p-TA*.

## 6 Conclusiones

La primera propuesta para combinar algoritmos en un equipo (*e-TA*) es una extensión natural del estado del arte, donde los algoritmos con malos desempeños son descartados en la medida que avanza la resolución de un problema. Por su parte, el *p-TA* relaja el enfoque anterior dando una probabilidad no nula de selección a todos los algoritmos. Esta probabilidad asociada a los algoritmos es dependiente de la clasificación Pareto de los procesos esclavos a la que pertenecen.

Los resultados presentados indican que las nuevas propuestas son enfoques prometedores, al lograr una mayor robustez considerando los distintos tipos de problemas combinatorios abordados en este trabajo. Específicamente, el *p-TA* ha logrado demostrar un mejor equilibrio entre las distintas figuras de mérito multiobjetivo al resolver el conjunto de problemas bi-objetivos propuestos. Esta robustez se justifica debido al uso combinado de diversos algoritmos disponibles, siendo estos clasificados y seleccionados de forma justa desde el punto de vista de la optimización Pareto.

Como trabajo futuro se pretende realizar más pruebas experimentales sobre más instancias así como también con problemas reales de ingeniería, buscando refinar el enfoque propuesto hacia métodos cada vez más “*inteligentes*” para combinar diferentes algoritmos sobre un sistema distribuido.

## Referencias

1. Coello, C., Lamont, G., Van Veldhuizen, D.: Evolutionary Algorithms for Solving Multi-Objective Problems. Genetic and Evolutionary Computation Series, Editors: David E. Goldberg and John R. Koza. Second Edition, Springer (2007)
2. Iredi, S., Merkle, D., Middendorf, M.: Bi-Criterion Optimization with Multi Colony Ant Algorithms. Proc. First International Conference on Evolutionary Multi-criterion Optimization (EMO'01), Lecture Notes in Computer Science 1993, Págs. 359-372 (2001)

3. García-Martínez, C., Cerdón, O., Herrera, F.: An Empirical Analysis of Multiple Objective Ant Colony Optimization Algorithms for the Bi-criteria TSP. ANTS Workshop Págs. 61-72 (2004)
4. Moore, J., Chapman, R.: Application of Particle Swarm to Multiobjective Optimization. Department of Computer Science and Software Engineering, Auburn University (1999)
5. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*. Vol. 1, Issue 1, Págs. 67 – 82, Abril (1997)
6. Baez, M., Zarate, D., Barán, B.: Multi-Objective Adaptive Memetic Algorithm. XXIII Informatics Congress Latin-American 2007, San José, Costa Rica, Octubre (2007)
7. Paciello, J., Martínez, H., Barán, B.: Team Algorithms for Ant Colony based Multi-objective Problems. Argentine Symposium Artificial Intelligent ASAI 2006, Mendoza, Argentina, Septiembre, (2006)
8. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms. Empirical result. *Evolutionary Computation Journal* 8: 2, Págs. 173-195 (2000)
9. Van Veldhuizen, D. A.: Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations. PhD thesis, Graduate School of Engineering of the Air Force Institute of Technology, Air University, Junio (1999)
10. Barán, B., Kaszkurewicz, E., Bhaya, A.: Parallel Asynchronous Team Algorithms: Convergence and Performance Analysis. *IEEE Transactions on Parallel & Distributed Systems*, Vol. 7, No. 7, Págs.677-688, (1996)
11. von Lucken, C., Sotelo, A., Barán, B.: Pump Scheduling Optimization using Asynchronous Parallel Evolutionary Algorithms. *CLEI Electronic Journal* (ISSN 0717-500) (2005)
12. Fernandez, J., Barán, B.: Multiobjective Evolutionary Elitist Team Algorithm. XXXI Conference Latino-American of Informatics, CLEI, Cali, Colombia, (2005)
13. A. G. et al.: PVM: Parallel Virtual machine - A user's guide and Tutorial for Networked parallel Computing. M.I.T. press, Cambridge, MA, (1994)
14. Cantu-Paz, E.: Designing efficient and accurate parallel genetic algorithms. Technical Report 2108, Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana-Illinois, (1999)
15. Stützle, T.: Parallelization strategies for ant colony optimization. In Eiben, A. *et al.* (Eds.) Proceedings of PPSN-V, Fifth International Conference on Parallel Problem Solving from Nature, Págs.722-731, Springer-Verlag (1998)
16. Deb, K., Goel, T.: Controlled Elitist Non-dominated Sorting Genetic Algorithms for Better Convergence. In E. Zitzler, K. Deb, L. Thiele, C. A. Coello Coello, and D. Corne, editors, First International Conference on Evolutionary Multi-Criterion Optimization, Págs. 67–81. Springer-Verlag. Lecture Notes in Computer Science No. 1993 (2001)
17. Iredi, S., Merkle, D., Middendorf, M.: Bi-Criterion Optimization with Multi Colony Ant Algorithms. Proc. First International Conference on Evolutionary Multi-criterion Optimization (EMO'01), Lecture Notes in Computer Science 1993, Págs. 359-372 (1993)
18. Doerner, K., Gutjahr, W., Hartl, R., Strauss, C.: Pareto Ant Colony Optimization: A Metaheuristic Approach to Multiobjective Portfolio Selection. Proceedings of the 4th. Metaheuristics International Conference. Porto, 243-248 (2004)
19. Doerner, K., Hartl, R., Reimann, M.: Are COMPETants more competent for problem solving? – the case of a multiple objective transportation problem. *Central European Journal of Operations Research*, 11:2, Págs. 115-141 (2003)
20. Schaefer, M., Baran, B.: A multiobjective Ant Colony System for Vehicle Routing Problems with Time Windows. Proc. Twenty first IASTED International Conference on Applied Informatics, Innsbruck, Austria, Págs. 97-102 (2003)
21. Pinto, D., Barán, B.: Solving Multiobjective Multicast Routing Problem with a new Ant Colony Optimization approach. II IFIP/ACM Latin-American Networking Conference, October, Cali, Colombia (2005)
22. Mariano, C., Morales, E.: A Multiple Objective Ant-Q Algorithm for the Design of Water Distribution Irrigation Networks". First International Workshop on Ant Colony Optimization ANTS (1998)
23. Gardel, P., Barán, B., Estigarribia, H., Fernandez, U.: Applications of Omicrom ACO to Reactive Compensation Problem in Multiobjective context. Congress Argentine of Computer Science 2005 CACIC 2005. Concordia – Argentina (2005)
24. Deb, K., Agrawal, S., Pratab, A., Meyarivan, T.: A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. KanGAL report 200001, Indian Institute of Technology, Kanpur, India, (2000)