Revista Cubana de Ciencias Informáticas
Vol. 10, No. 1, Enero-Marzo, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 182-193

Tipo de artículo: Artículos originales
Temática: Inteligencia artificial
Recibido: 21/04/2015 | Aceptado: 16/12/2015

# Evaluación del desempeño de la metaheurística MOVMO en funciones de prueba con restricciones.

# *Performance assessment of MOVMO metaheuristic on constrains test functions.*

**Yamisleydi Salgueiro Sicilia[1*], Jorge L. Toro Pozo[1], Rafael Bello Pérez[2]**

[1]Departamento de Informática. Universidad de Las Tunas. Ave. Carlos J. Finlay s/n. {yamisalgueiro, jorgitoltp}@gmail.com
[2]Centro de Estudios de Informática. Universidad Central "Marta Abreu"de Las Villas. rbellop@uclv.edu.cu

[*]Autor para correspondencia: yamisalgueiro@gmail.com

---

**Resumen**
Los métodos clásicos de programación matemática presentan limitaciones en la solución de problemas de optimización multiobjetivo. Estas limitaciones se evidencian fundamentalmente en problemas reales con múltiples funciones objetivo en conflicto y con espacios de soluciones complejos. En este contexto se ha extendido el uso de las metaheurísticas debido a su capacidad de lidiar con este tipo de problemas. Sin embargo, a diferencia de los métodos exactos, las metaheurísticas no garantizan encontrar la solución óptima de un problema. Por este motivo se continúan creando métodos que, ya sea mediante la incorporación de nuevas estrategias o a través de la hibridación de las existentes, permitan obtener mejores aproximaciones al frente de Pareto. Tal es el caso de la metaheurística MOVMO creada por los autores del presente trabajo, que es una versión multiobjetivo de la metaheurística VMO. Esta investigación tuvo como objetivo evaluar el desempeño de la metaheurística MOVMO en funciones con restricciones. Los estudios experimentales permitieron evaluar el desempeño de MOVMO con respecto a los métodos NSGA-II, SPEA2 y SMPSO en las funciones ConstrEx, Golinski, Osyczka, Srinivas, Tanaka y Water. Los resultados obtenidos por MOVMO en los indicadores de calidad *Epsilon* e *Hypervolume* superaron, con diferencias estadísticamente significativas, a los obtenidos por los restantes métodos en varias de las funciones de prueba. Estos resultados demuestran la competitividad de los operadores y técnicas utilizados por MOVMO en problemas de optimización multiobjetivo con restricciones.

**Palabras claves:** Metaheurísticas Multiobjetivo, Optimización Multiobjetivo, Optimización basada en Mallas Variables

**Abstract**
*Classical mathematical programming methods have limitations solving multi-objective optimization problems. These drawbacks are mainly evident in real problems with multiple functions in conflict and complex solutions spaces. That is why the use of meta-heuristics has extended a great deal at present due to its ability to deal with such problems. But as meta-heuristics do not guarantee finding the optimal solution for a problem, new methods are being created either by means of the incorporation of new strategies or by hybridization of the existing ones, to obtain better approximations to Pareto front. This is the case of MOVMO, created by the authors of this work, which is a multi-objective version of VMO meta-heuristic. The objective of this present research was to evaluate the performance of MOVMO on constrains test problems. The experimental studies*

*allowed us to assess the competence of MOVMO in comparison with NSGA-II, SPEA2 and SMPSO methods on ConstrEx, Golinski, Osyczka, Srinivas, Tanaka, and Water functions. Results achieved by MOVMO in Epsilon and Hypervolume quality indicators were higher with significant statistically differences in comparison with those results from other methods in several test functions. These results prove the competitiveness of operators and techniques used in MOVMO on constrains multi-objective optimization problems.*

**Keywords:** *Multiobjective Metaheuristics, Multiobjective Optimization, Variable Mesh Optimization*

## Introducción

El proceso de toma de decisiones en problemas del mundo real frecuentemente involucra la consideración de múltiples objetivos, como pueden ser: minimizar los costos o maximizar la calidad de un producto. Estos objetivos, que usualmente se encuentran en conflicto, deben alcanzarse teniendo en cuenta diversas restricciones de recursos o tiempo, haciendo aún más complejo escoger la mejor decisión. Este tipo de problemas es conocido como *problemas de optimización multiobjetivo* y sin pérdida de generalidad se pueden definir como:

$$\begin{array}{ll} minimize & F(x) = (f_1(x), \ldots, f_m(x))^T \\ subject\,to & x \in \Omega \end{array} \tag{1}$$

Donde $\Omega$ es el espacio no vacío de decisión y $x \in \Omega$ es el vector de decisión. $F(x)$ conformada por $m, (m \geq 2)$ funciones objetivo generalmente en conflicto $f_i : \Omega \rightarrow \mathbb{R}, i = 1, \ldots, m$ donde $\mathbb{R}^m$ es el espacio objetivo. En el caso de que se desee maximizar la función $f_i$, su equivalente es minimizar -$f_i$.

Autores como (DEB, 2014) y (MIETTINEN et al., 2008) han resaltado las limitaciones presentes en los métodos clásicos de programación matemática para la solución de problemas de optimización multiobjetivo. Un enfoque común en este dominio ha sido convertir artificialmente los problemas de optimización multiobjetivo en problemas de un solo objetivo, para luego aplicar algún método de optimización mono-objetivo a la función integrada. Sin embargo, en esta agregación no se tiene en cuenta la correlación entre las funciones objetivo, que es usualmente compleja y depende de las alternativas disponibles. Adicionalmente, es común que las funciones objetivo no sean comparables, lo que hace más difícil agregarlas en una sola función. Todo lo anteriormente expuesto propició la inserción de métodos de otras ramas como la Inteligencia Artificial que permitieran resolver, de manera más efectiva, los problemas de optimización multiobjetivo.

Las metaheurísticas son métodos de solución de problemas que integran mejoras en procedimientos de búsqueda local y estrategias de alto nivel generando procesos capaces de escapar de los óptimos locales y desempeñar búsquedas robustas en el espacio de soluciones (SALHI, 2014). Las metaheurísticas no garantizan encontrar la solución óptima, por ello se continúan creando métodos que, mediante la incorporación de nuevas estrategias o la hibridación de las existentes, mejoren las aproximaciones al frente de Pareto obtenidas y su distribución. En este campo de investigación la tendencia en los últimos años ha sido la extensión de metaheurísticas que originalmente fueron creadas para solucionar problemas de optimización mono-objetivo, hacia métodos capaces de solucionar problemas multiobjetivo. Ejemplos de esto pueden encontrarse en la Sección 2.3 de (ZHOU et al., 2011). Otros trabajos que resumen las extensiones multiobjetivo de metaheurísticas conocidas son: (LEGUIZAMÓN and COELLO, 2011) de la metaheurística *optimización basada en colonia de hormigas* (ACO por sus siglas en inglés); (REYES-SIERRA and COELLO, 2006) para la *optimización basada en enjambre de partículas* (PSO por sus siglas en inglés) y (MEZURA-MONTES et al., 2008) donde se resumen las variantes multiobjetivo del método *evolución diferencial* (DE por sus siglas en inglés). El ejemplo clásico de extensión de un algoritmo mono-objetivo a multiobjetivo es el conocido NSGA-II, convertido en un algoritmo de referencia entre las metaheurísticas multiobjetivo.

La Optimización basada en Mallas Variables (VMO por sus siglas en inglés) es una metaheurística basada en poblaciones creada por (PURIS et al., 2012) y que mostró resultados competitivos respecto a Algoritmos Genéticos, PSO y DE. En una reciente investigación realizada por los autores del presente trabajo se comprobó la competitividad de una versión multiobjetivo de VMO llamada Optimización Multiobjetivo basada en Mallas Variables (MOVMO por sus siglas en inglés), que fue comparada contra siete algoritmos del estado del arte en las conocidas funciones de prueba sin restricciones ZDT (ZITZLER et al., 2000), DTLZ (DEB et al., 2005), WFG y LZ09.

El objetivo de la presente investigación fue comprobar el desempeño de la metaheurística MOVMO en funciones de prueba con restricciones. Para ello se comparó MOVMO contra algoritmos del estado del arte como NSGA-II (DEB et al., 2002), SPEA2 (ZITZLER and THIELE, 1999) y otro más reciente como el SMPSO (NEBRO et al., 2009). Las funciones con restricciones utilizadas fueron las conocidas ConstrEx (DEB et al., 2002), Golinski (KURPATI et al., 2002), Osyczka2 (OSYCZKA and KUNDU, 1995), Srinivas (SRINIVAS and DEB, 1995), Tanaka (TANAKA et al., 1995) y Water (RAY et al., 2001). De acuerdo a la experimentación realizada se pudo comprobar la competitividad de los mecanismos empleados en el MOVMO con respecto a los algoritmos NSGA-II, SPEA2 y SMPSO.

A continuación se muestran los conceptos básicos utilizados en la investigación y se brinda una descripción general de la metaheurística MOVMO y del framework experimental utilizado. Posteriormente en la sección dedicada a los *Resultados y discusión* se muestran los resultados experimentales obtenidos por MOVMO en las funciones de prueba con restricciones en comparación con los métodos NSGA-II, SPEA2 y SMPSO. Finalmente se brindan las conclusiones de la investigación.

## Metodología computacional

En problemas de optimización multiobjetivo no se cuenta con una solución óptima sino con un conjunto de soluciones. La optimalidad de Pareto es el concepto predominante para determinar las soluciones que pertenecen a este conjunto y se pueden definir de la siguiente manera:

Definición 1 *Una solución factible* $x^* \in \Omega$ *de la Ecuación 1 es denominada óptima de Pareto, ssi* $\nexists y \in \Omega$ *tal que* $F(y) < F(x^*)$*. Al conjunto de todas las soluciones óptimas de Pareto se denomina conjunto óptimo de Pareto (PS por sus siglas en inglés) y se define como:* $PS = \{x^* \in \Omega \mid \nexists y \in \Omega, F(y) < F(x^*)\}$*. La imagen del PS en el espacio objetivo es denominado frente de Pareto (PF por sus siglas en inglés) y se define como:* $PF = \{F(x) \mid x \in PS\}$*.*

Todas las soluciones que formen parte del conjunto óptimo de Pareto deben ser *no dominadas*, concepto que se muestra en la Definición 2. No obstante, los algoritmos frecuentemente encuentran soluciones que, aunque no pertenecen al conjunto óptimo de Pareto, satisfacen ciertos criterios haciéndolas importantes en aplicaciones reales Definición 3:

Definición 2 *Un vector* $u = (u_1, \ldots, u_m)^T$ *domina a otro vector* $v = (v_1, \ldots, v_m)^T$ *denotado como* $u \prec v$ *ssi* $\forall i \in \{1, \ldots, m\}, u_i \leq v_i$ *y* $u \neq v$*.*

Definición 3 *Un vector* $u = (u_1, \ldots, u_m)^T$ *débilmente domina a otro vector* $v = (v_1, \ldots, v_m)^T$ *denotado como* $u \preceq v$ *ssi* $\forall i \in \{1, \ldots, m\}, u_i \leq v_i$*.*

Las metaheurísticas no garantizan encontrar el frente de Pareto, por ello a los resultados obtenidos mediante estos métodos se les denomina *aproximaciones al frente de Pareto*. La Definición 2 solo permite establecer un orden parcial entre las aproximaciones al frente de Pareto, por lo que se necesita de otro mecanismo para determinar cuándo una es mejor que otra.
Los indicadores de calidad $I : \Psi^n \to \mathbb{R}$ permiten mapear $n$ aproximaciones al frente de Pareto a un número real $\mathbb{R}$ y realizar estadísticas de las distribuciones de los números resultantes (KNOWLES et al., 2006; ZITZLER et al., 2008). El orden que establece $I$ en $\Omega$ equivale a la calidad de la aproximación del frente de Pareto. Los indicadores de calidad utilizados en la presente investigación son *Epsilon* en su versión aditiva ($I_\varepsilon^+$) introducido en (ZITZLER et al., 2003) e *Hypervolume* ($I_{HV}$), inicialmente definido en (ZITZLER and THIELE, 1999).

## Optimización Multiobjetivo basada en Mallas Variables

En la presente sección se describen los principales componentes de MOVMO que, como se comentó con anterioridad, es una extensión multiobjetivo de la metaheurística VMO.
MOVMO posee cuatro parámetros de entrada $P, S, k$ y $C$ que se describen a continuación. $P$ es el número de nodos $(n_1, \ldots, n_P)$ de la malla, cada nodo o solución es codificado como un vector de $M$ números reales $n_i = \{v_1^i, \ldots, v_M^i\}$. $S$ es el número máximo de soluciones permitidas en el archivo de líderes. El archivo de líderes constituye la aproximación al frente de Pareto encontrada hasta el momento por la metaheurística y se

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 1, Enero-Marzo, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 182-193

almacena en el conjunto $L$. Por otra parte, $k$ es el número de nodos (soluciones) que definen la vecindad de cada nodo de la malla. Finalmente $C$ es la condición de parada, en este caso corresponde al número máximo de evaluaciones de la función objetivo.

**Algorithm 1:** Multiobjective Variable Mesh Optimization (MOVMO)

**Input**: Tamaño de la malla $P$, tamaño del archivo de líderes $S$, número de soluciones que conforman el vecindario $k$, número máximo de evaluaciones de la función objetivo $C$

**Output**: $L$ aproximación al frente de Pareto $L^*$

1 Generar la malla inicial $\mathcal{M} = \{n_i\}$ y calcular el vector objetivo $F_i$ $\forall i \in \{1, \ldots, P\}$
2 Inicializar el archivo de líderes $L$ adicionándole cada nodo o solución $n_i$ de la población inicial
3 **while** $c \leq C$ **do**
4     **foreach** $n_i$ *de la malla* $\mathcal{M}$ *actual* **do**
5         De entre los $k$-vecinos de $n_i$ seleccionar valor extremo (mejor) local $n_i^*$
6         **if** $n_i^* \prec n_i$ *(Definición 2)* **then**
7             $n_l \leftarrow$ Generar los extremos locales por la Ecuación 2
8         **else**
9             $n_l \leftarrow n_i$
10         $n_g \leftarrow$ Seleccionar aplicando Torneo Binario un líder global del archivo de líderes $L$
11         $n_x$ = Cruzar($n_l, n_g$)
12         EvaluarDesempeño($n_x$)
13         Incrementar(c)
14         Adicionar la solución $n_x$ al archivo de líderes $L$
15         **if** $n_x \preceq n_i$ *(Definición 3)* **then**
16             Remplazar $n_i$ por $n_x$ en la mall actual

El Algoritmo 1 muestra un seudo-código de la metaheurística MOVMO que comienza inicializando la población de nodos de la malla $\mathcal{M}_0$ y, con ellas, inicializa el archivo de líderes $L_0$. Luego, por cada nodo $n_i$ de la población actual (Líneas 5 a 9) se selecciona el extremo local (nodo con mejor valor en las funciones objetivo) de entre los $k$-vecinos más cercanos de acuerdo a la distancia Euclidiana. Si el extremo local domina a $n_i$ un nuevo nodo es generado por la Ecuación 2. Seguidamente, (Líneas 10 a 16) se selecciona por Torneo Binario un líder global $n_g$ del archivo de líderes $L$ y se le aplica el operador de cruzamiento SBX entre este y el obtenido del paso anterior $n_l$. El nuevo nodo o solución generada $n_x$ es adicionada al archivo de líderes $L$. Finalmente, si en el proceso anterior se genera un nodo o solución que débilmente domine al nodo actual, este último es remplazado dinámicamente en la población. Siempre que se realice la evaluación de una solución se determinará también el número de restricciones que la misma viola siguiendo el procedimiento propuesto en (DEB et al., 2002).

$$n_z = F(n_i, n_i^*, Pr(n_i, n_i^*)) \tag{2}$$

Donde $Pr$ es llamado el factor de cercanía y representa la relación entre el valor de la función objetivo del nodo actual y el de su extremo local, este factor es calculado por la Ecuación 3.

$$Pr(n_i, n_i^*) = \frac{1}{1 + \sqrt{\sum_{j=1}^{m} \left(f_j(n_i) - f_j(n_i^*)\right)^2}} \tag{3}$$

La función $F$ se describe como:

$$v_j^z = \begin{cases} \overline{m_j}, & \text{if} \left|\overline{m_j} - v_j^{i^*}\right| > \xi_j \text{ and} \\ & U[0,1] \leq Pr(n_i, n_i^*) \\ v_j^{i^*} + U[-\xi_j, \xi_j], & \text{if} \left|\overline{m_j} - v_j^{i^*}\right| \leq \xi_j \\ U[v_j^i, \overline{m_j}], & \text{other case} \end{cases} \tag{4}$$

Donde $\overline{m_j} = average(v_j^i, v_j^{i^*}), U[x, y]$ denota un valor aleatorio uniformemente distribuido en el intervalo $[x, y]$. Y $\xi_j$ define la distancia mínima permitida por cada componente. Su valor decrece durante la corrida del método, calculado por la Ecuación 5,

$$\xi_j = \begin{cases} \frac{range(a_j,b_j)}{4} & \text{if} \quad c < 0{,}15\,\%C \\ \frac{range(a_j,b_j)}{8} & \text{if} \quad 0{,}15\,\%C \leq c < 0{,}3\,\%C \\ \frac{range(a_j,b_j)}{16} & \text{if} \quad 0{,}3\,\%C \leq c < 0{,}6\,\%C \\ \frac{range(a_j,b_j)}{50} & \text{if} \quad 0{,}6\,\%C \leq c < 0{,}8\,\%C \\ \frac{range(a_j,b_j)}{100} & \text{if} \quad c \geq 0{,}8\,\%C \end{cases} \tag{5}$$

donde $C$ y $c$ son el número máximo de evaluaciones de la función objetivo y su valor actual respectivamente. Por otro, lado $range(a_j, b_j)$ denota el dominio de amplitud $(a_j, b_j)$ de cada componente.
El procedimiento empleado para la actualización y mantenimiento del archivo de líderes $L$ dada una solución $n_x$ puede verse en detalle en (DEB et al., 2002). En su primer paso se comprueba si la solución $n_x$ domina a alguna de las solución pertenecientes al archivo de líderes $L$, en caso afirmativo dichas soluciones son eliminadas. Sin embargo, si la solución que se desea insertar es dominada o igual a algún miembro del archivo de líderes, entonces esta se descartada y el algoritmo finaliza. Una vez que el conjunto $L$ tenga el máximo de soluciones permitidas y se desee insertar una nueva solución $n_x$, aquella solución que peor *Crowding Distance* aporte será eliminada. Finalmente, se actualiza el valor del *Crowding Distance* de cada solución en $L$, puesto que este valor se utiliza en el proceso de selección por Torneo Binario.

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 1, Enero-Marzo, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 182-193

## Framework experimental

La validación de nuevas metaheurísticas frecuentemente requiere la definición de un marco experimental exhaustivo que incluye problemas y algoritmos del estado del arte. La parte crítica de estas comparaciones recae en la validación estadística de los resultados que permiten contrastar las diferencias encontradas entre los resultados obtenidos por cada uno de los métodos (DERRAC et al., 2011).

Tanto la implementación de la metaheurística MOVMO como los experimentos conducidos en la presente investigación fueron realizados utilizando el framework de computación evolutiva jMetal v4.5 (DURILLO and NEBRO, 2011). Con el objetivo de asegurar una justa comparación entre MOVMO y los métodos anteriormente descritos se utilizaron las siguientes configuraciones de los parámetros: tamaño de población de 100 individuos e igual número para el tamaño del archivo de líderes para los métodos SMPSO y MOVMO. En todos los algoritmos de optimización multiobjetivo basados en estrategias genéticas se utiliza como operador de cruzamiento el SBX con un índice de distribución de $n_c = 20$ y $p_c = 0{,}9$ como valor de probabilidad. El operador de mutación utilizado fue polinomial con índice de distribución de $n_m = 20$ y probabilidad de $p_m = \frac{1}{D}$, donde $D$ es el número de variables de decisión. Finalmente, el operador de selección escogido es el de Torneo Binario. La configuración utilizada en los problemas de prueba se resumen en la tabla 1.

Tabla 1. Configuración de la cantidad de variables, restricciones y número de objetivos de cada función

| Nombre de la Función | Variables | Restricciones | Objetivos |
|---|---|---|---|
| ConstrEx | 2 | 2 | 2 |
| Golinski | 7 | 11 | 2 |
| Osyczka | 6 | 6 | 2 |
| Srinivas | 2 | 2 | 2 |
| Tanaka | 2 | 2 | 2 |
| Water | 3 | 7 | 3 |

Para cada método se ejecutaron un total de 30 corridas independientes y el número máximo de evaluaciones de la función objetivo fue de 25000. De estas ejecuciones se obtuvieron la media $\tilde{x}$ y el rango inter-quartile (IQR), como medidas de tendencia central y dispersión estadística, respectivamente. A estos resultados se le realizó la prueba de la suma de rangos de Wilcoxon que realiza una prueba por pares que permite detectar la existencia de diferencias significativas entre las medias de dos muestras. En este caso los resultados obtenidos por dos algoritmos (DERRAC et al., 2011). Para la ejecución del test se utilizó el lenguaje *R* y la herramienta *RStudio*.

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 1, Enero-Marzo, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 182-193

## Resultados y discusión

En esta sección se muestran los resultados experimentales obtenidos por los métodos MOVMO, NSGA-II, SPEA2 y SMPSO en las funciones de prueba con restricciones ConstrEx, Golinski, Osyczka, Srinivas, Tanaka y Water. Inicialmente se analiza los resultados obtenidos de acuerdo al indicador de calidad *Epsilon* y posteriormente los correspondientes al indicador $Hypervolume$.

En las tablas 2 y 4, correspondientes a la media y rango inter-quartil (IQR) de los indicadores $I_{\varepsilon}^{+}$ e $I_{HV}$ el resultado del método que obtuvo el mejor desempeño se resalta con un fondo gris oscuro, mientras que el resultado del segundo método se resalta con un fondo de un gris más claro. Por otra parte, las tablas 3 y 5 corresponden a los resultados de la prueba de la suma de rangos de Wilcoxon de los indicadores de calidad $I_{\varepsilon}^{+}$ e $I_{HV}$. En estas tablas el símbolo "▲" implica que el resultado obtenido por el método de la fila supera al obtenido por el de la columna con diferencia estadísticamente significativas. Por el contrario el símbolo "▽" implica que el resultado obtenido por el método de la columna supera al de la fila con diferencias estadísticamente significativas. Finalmente, el símbolo "–" representa la ausencia de diferencias significativas entre el resultado del método de la fila y el de la columna. El orden en que se muestran cada uno de los símbolos coincide con el orden de los problemas de prueba: el primer símbolo corresponde al problema de prueba ConstrEx, el segundo a Golinski y así sucesivamente hasta el sexto que corresponde a Water.

Es importante resaltar que en el caso del indicador de calidad *Epsilon* ($I_{\varepsilon}^{+}$) menores valores implican un mejor desempeño, mientras que en el $Hypervolume$ ($I_{HV}$) ocurre lo contrario; mayores valores implican mejores desempeños. En el caso de los resultados de la prueba de suma de rangos de Wilcoxon se consideró un *p-value*$< 0{,}05$.

Las tablas 2 y 3 corresponden a los resultados experimentales obtenidos en el indicador de calidad *Epsilon* ($I_{\varepsilon}^{+}$).

Tabla 2. Media e IQR de los resultados obtenidos por NSGA-II, SPEA2, SMPSO y MOVMO en el indicador de calidad $I_{\varepsilon}^{+}$

| | NSGAII | SPEA2 | SMPSO | MOVMO |
|---|---|---|---|---|
| ConstrEx | $1{,}54e-02_{4{,}3e-03}$ | $2{,}62e-02_{7{,}6e-03}$ | $8{,}26e-03_{1{,}3e-03}$ | $8{,}35e-03_{1{,}3e-02}$ |
| Golinski | $8{,}62e+00_{2{,}8e+00}$ | $1{,}58e+01_{4{,}9e+00}$ | $1{,}38e+01_{2{,}3e+00}$ | $5{,}76e+00_{1{,}5e+00}$ |
| Osyczka2 | $1{,}52e+01_{1{,}4e+01}$ | $1{,}61e+01_{9{,}3e+00}$ | $3{,}37e+01_{5{,}7e+00}$ | $1{,}66e+01_{1{,}7e+01}$ |
| Srinivas | $3{,}21e+00_{7{,}6e-01}$ | $1{,}90e+00_{1{,}8e-01}$ | $1{,}43e+00_{1{,}6e-01}$ | $2{,}09e+00_{9{,}2e-01}$ |
| Tanaka | $8{,}54e-03_{1{,}1e-03}$ | $7{,}96e-03_{1{,}8e-03}$ | $1{,}54e-02_{5{,}8e-03}$ | $7{,}67e-03_{5{,}1e-03}$ |
| Water | $7{,}06e+04_{1{,}6e+04}$ | $6{,}50e+04_{1{,}9e+04}$ | $1{,}11e+05_{4{,}0e+04}$ | $2{,}42e+04_{3{,}2e+03}$ |

Como se puede observar en la tabla 2, MOVMO obtuvo el primer lugar en los problemas de prueba Golinski, Tanaka y Water, y el segundo lugar en ConstrEx; resumiendo tres primeros lugares y un segundo lugar de un total de seis problemas de prueba. En el segundo lugar se colocó el método SMPSO con dos primeros lugares,

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 1, Enero-Marzo, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 182-193

seguido por NSGA-II con un primer lugar, y un segundo lugar y el SPEA2 con cuatro segundos lugares. Partiendo de estos resultados y luego de aplicar la prueba de la suma de rangos de Wilcoxon, se obtuvieron los resultados mostrados en la tabla 3.

Tabla 3. Prueba de la suma de rangos de Wilcoxon realizado entre los resultados obtenidos por el algoritmo de la fila y el de la columna respecto al indicador $I_\epsilon^+$ en los seis problemas de prueba

| | SPEA2 | SMPSO | MOVMO |
|---|---|---|---|
| NSGAII | ▲ ▲ ▲ ▽ ▽ ▽ | ▽ ▲ ▲ ▽ ▲ ▲ | ▽ ▽ ▲ ▽ – ▽ |
| SPEA2 | | ▽ ▽ ▲ ▽ ▲ ▲ | ▽ ▽ – – – ▽ |
| SMPSO | | | – ▽ ▽ ▲ ▽ ▽ |

En la última columna de la tabla 3 correspondiente a los resultados obtenidos por el MOVMO podemos observar seis columnas de símbolos correspondientes a los problemas de prueba con restricciones ConstrEx, Golinski, Osyczka, Srinivas, Tanaka y Water respectivamente. La columna uno, dos y seis presentan el símbolo "▽ "cuyo significado es que el método de la columna (MOVMO) supera con diferencias estadísticamente significativas a los métodos de la fila (NSGA-II, SPEA2 y SMPSO). Los resultados más discretos del MOVMO se evidencian en los problemas de prueba Osyczka2, Srinivas y Tanaka. Podemos concluir que en los problemas de prueba ConstrEx, Golinski y Water, MOVMO superó al resto de los métodos de acuerdo al indicador de calidad $I_\epsilon^+$. Los resultados del test de Wilcoxon para el indicador de calidad *Epsilon* de un total de dieciocho comparaciones MOVMO es superado en dos, no posee diferencias significativas en cinco, y en las restantes once supera al resto de los métodos con diferencias estadísticamente significativas.
Las tablas 4 y 5 corresponden a los resultados experimentales obtenidos en el indicador de calidad *Hypervolume* ($I_{HV}$).

Tabla 4. Media e IQR de los resultados obtenidos por NSGA-II, SPEA2, SMPSO y MOVMO en el indicador de calidad $I_{HV}$

| | NSGAII | SPEA2 | SMPSO | MOVMO |
|---|---|---|---|---|
| ConstrEx | $7{,}74e-01_{3{,}5e-04}$ | $7{,}76e-01_{3{,}2e-04}$ | $7{,}76e-01_{1{,}8e-04}$ | $7{,}77e-01_{1{,}4e-03}$ |
| Golinski | $9{,}69e-01_{1{,}3e-04}$ | $9{,}67e-01_{5{,}5e-04}$ | $9{,}67e-01_{1{,}1e-03}$ | $9{,}69e-01_{2{,}1e-04}$ |
| Osyczka2 | $7{,}45e-01_{7{,}8e-03}$ | $7{,}35e-01_{2{,}6e-02}$ | $6{,}94e-01_{1{,}1e-02}$ | $7{,}35e-01_{3{,}8e-02}$ |
| Srinivas | $5{,}38e-01_{3{,}8e-04}$ | $5{,}40e-01_{2{,}6e-04}$ | $5{,}41e-01_{9{,}0e-05}$ | $5{,}41e-01_{2{,}5e-04}$ |
| Tanaka | $3{,}08e-01_{3{,}4e-04}$ | $3{,}09e-01_{4{,}4e-04}$ | $3{,}04e-01_{1{,}6e-03}$ | $3{,}09e-01_{4{,}3e-04}$ |
| Water | $3{,}96e-01_{1{,}1e-02}$ | $4{,}05e-01_{1{,}0e-02}$ | $3{,}39e-01_{4{,}9e-02}$ | $4{,}28e-01_{2{,}8e-03}$ |

Como se puede observar en la tabla 4, MOVMO obtuvo el primer lugar en los problemas de prueba ConstrEx, Golinski, Tanaka y Water. En resumen, obtuvo el primer lugar en cuatro de los seis problemas de prueba. En el segundo lugar se colocaron los algoritmos NSGA-II y SMPSO, ya que en ambos casos alcanzaron un primer lugar y un segundo lugar. En la última posición se ubicó el SPEA2, con tres segundos lugares. Partiendo de

Revista Cubana de Ciencias Informáticas
Vol. 10, No. 1, Enero-Marzo, 2016
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 182-193

estos resultados, y luego de aplicar la prueba de la suma de rangos de Wilcoxon, se obtuvieron los resultados mostrados en la tabla 5.

Tabla 5. Prueba de la suma de rangos de Wilcoxon realizado entre los resultados obtenidos por el algoritmo de la fila y el de la columna respecto al indicador $I_{HV}$ en los seis problemas de prueba

| | SPEA2 | | | | | | SMPSO | | | | | | MOVMO | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| NSGAII | ▽ | ▲ | ▲ | ▽ | ▽ | ▽ | ▽ | ▲ | ▲ | ▽ | ▲ | ▲ | ▽ | ▽ | ▲ | ▽ | ▽ | ▽ |
| SPEA2 | | | | | | | ▽ | ▲ | ▲ | ▽ | ▲ | ▲ | ▽ | ▽ | – | ▽ | ▽ | ▽ |
| SMPSO | | | | | | | | | | | | | ▽ | ▽ | ▽ | – | ▽ | ▽ |

La última columna de la tabla 5 contiene los resultados obtenidos por MOVMO. En ella podemos observar seis columnas de símbolos correspondientes a los problemas de prueba con restricciones ConstrEx, Golinski, Osyczka, Srinivas, Tanaka y Water, respectivamente. Las columnas de símbolos uno, dos, cinco y seis presentan el símbolo "▽ "cuyo significado es que el método de la columna (MOVMO) supera con diferencias estadísticamente significativas a los métodos de la fila (NSGA-II, SPEA2 y SMPSO). De forma individual MOVMO es superado exclusivamente por NSGA-II en la función Osyczka y no muestra diferencias significativas en esta misma función con respecto al algoritmo SPEA2, tampoco presenta diferencias significativas en la función Srinivas en la comparación con el algoritmo SMPSO. Podemos concluir que en los problemas de prueba ConstrEx, Golinski, Tanaka y Water, MOVMO superó al resto de los métodos de acuerdo al indicador de calidad $I_{HV}$. Los resultados de la prueba de la suma de rangos de Wilcoxon para el indicador de calidad *Hypervolume* muestran que de un total de dieciocho comparaciones MOVMO es superado en uno, no posee diferencias significativas en dos, y en las restantes quince supera al resto de los métodos con diferencias estadísticamente significativas.

## Conclusiones

En la presente investigación se comprobó, de acuerdo a los indicadores de calidad *Epsilon* e *Hypervolume*, la competitividad de la metaheurística multiobjetivo MOVMO. Las comparaciones se realizaron con respecto a los métodos NSGA-II, SPEA2 y SMPSO en los problemas de prueba con restricciones ConstrEx, Golinski, Osyczka, Srinivas, Tanaka y Water. De manera general MOVMO superó al resto de los métodos de acuerdo a los indicadores de calidad *Epsilon* e *Hypervolume* en los problemas ConstrEx, Golinski y Water. En ambos indicadores los resultados más discretos fueron en Osyczka y Srinivas. Los resultados obtenidos demuestran la efectividad de los mecanismos de búsqueda empleados en MOVMO, así como la factibilidad del uso de este método en la solución de problemas de optimización multiobjetivo con restricciones.

## Referencias

DEB, K. (2014). Multi-objective optimization. In Burke, E. K. and Kendall, G., editors, *Search Methodologies*, pages 403–449. Springer US.

DEB, K., PRATAP, A., AGARWAL, S., and MEYARIVAN, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.

DEB, K., THIELE, L., LAUMANNS, M., and ZITZLER, E. (2005). Scalable test problems for evolutionary multiobjective optimization. In Abraham, A., Jain, L., and Goldberg, R., editors, *Evolutionary Multiobjective Optimization*, Advanced Information and Knowledge Processing, pages 105–145. Springer London.

DERRAC, J., GARCÍA, S., MOLINA, D., and HERRERA, F. (2011). A practical tutorial on the use of non-parametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, 1(1):3 – 18.

DURILLO, J. J. and NEBRO, A. J. (2011). jmetal: A java framework for multi-objective optimization. *Advances in Engineering Software*, 42(10):760 – 771.

KNOWLES, J., THIELE, L., and ZITZLER, E. (2006). A tutorial on the performance assessment of stochastic multiobjective optimizers. Technical report, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Switzerland.

KURPATI, A., AZARM, S., and WU, J. (2002). Constraint handling improvements for multi-objective genetic algorithms. *Structural and Multidisciplinary Optimization*, 23(3):204–213.

LEGUIZAMÓN, G. and COELLO, C. A. C. (2011). Multi-objective ant colony optimization: A taxonomy and review of approaches. *Integration of Swarm Intelligence and Artificial Neural Network*, pages 67–94.

MEZURA-MONTES, E., REYES-SIERRA, M., and COELLO, C. (2008). Multi-objective optimization using differential evolution: A survey of the state-of-the-art. In Chakraborty, U., editor, *Advances in Differential Evolution*, volume 143 of *Studies in Computational Intelligence*, pages 173–196. Springer Berlin Heidelberg.

MIETTINEN, K., RUIZ, F., and WIERZBICKI, A. P. (2008). Introduction to multiobjective optimization: Interactive approaches. In Branke, J., Deb, K., Miettinen, K., and Slowinski, R., editors, *Multiobjective Optimization*, volume 5252 of *Lecture Notes in Computer Science*, pages 27–57. Springer Berlin Heidelberg.

NEBRO, A., DURILLO, J., GARCIA-NIETO, J., COELLO COELLO, C., LUNA, F., and ALBA, E. (2009). Smpso: A new pso-based metaheuristic for multi-objective optimization. In *IEEE Symposium on Computational intelligence in Multi-criteria Decision-Making, 2009. mcdm '09*, pages 66–73.

OSYCZKA, A. and KUNDU, S. (1995). A new method to solve generalized multicriteria optimization problems using a simple genetic algorithm. *Structural Optimization*, 10(2):94–99.

PURIS, A., BELLO, R., MOLINA, D., and HERRERA, F. (2012). Variable mesh optimization for continuous optimization problems. *Soft Computing*, 16(3):511–525.

RAY, T., TAI, K., and CHYE SEOW, K. (2001). Multiobjective design optimization by an evolutionary algorithm. *Engineering Optimization*, 33(4):399–424.

REYES-SIERRA, M. and COELLO, C. C. (2006). Multi-objective particle swarm optimizers: A survey of the state-of-the-art. *International Journal of Computational Intelligence Research*, 2(3):287–308.

SALHI, S. (2014). Handbook of metaheuristics. *Journal of the Operational Research Society*, 65(2):320–320.

SRINIVAS, N. and DEB, K. (1995). Multiobjective function optimization using nondominated sorting genetic algorithms. *Evolutionary Computation*, 2(3):221–248.

TANAKA, M., WATANABE, H., FURUKAWA, Y., and TANINO, T. (1995). Ga-based decision support system for multicriteria optimization. In *IEEE International Conference on Systems, Man, and Cybernetics*, volume 2, pages 1556–1561.

ZHOU, A., QU, B.-Y., LI, H., ZHAO, S.-Z., SUGANTHAN, P. N., and ZHANG, Q. (2011). Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1):32 – 49.

ZITZLER, E., KALYANMOY, D., and LOTHAR, T. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation. MIT Press*, 8(2):173–195.

ZITZLER, E., KNOWLES, J., and THIELE, L. (2008). Quality assessment of pareto set approximations. In Branke, J., Deb, K., Miettinen, K., and Slowinski, R., editors, *Multiobjective Optimization*, volume 5252 of *Lecture Notes in Computer Science*, pages 373–404. Springer Berlin Heidelberg.

ZITZLER, E. and THIELE, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271.

ZITZLER, E., THIELE, L., LAUMANNS, M., FONSECA, C., and DA FONSECA, V. (2003). Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132.