INFORMÁTICA Y SISTEMAS
REVISTA DE TECNOLOGÍAS DE LA INFORMÁTICA Y LAS COMUNICACIONES

UNIVERSIDAD TÉCNICA DE MANABÍ Vol.1 N.1 Año 1 (2017)

Un marco de trabajo para las medidas de rendimiento en la optimización dinámica evolutiva

Pavel Novoa Hernández[1,*], José Luis Tubay Vergara[1,], Freddy Enrique Triana Litardo[1,], Amilkar Yudier Puris Cáceres[1,]

Resumen

Varios fenómenos reales pueden ser modelados como problemas dinámicos de optimización. Estos problemas han sido tratados eficientemente mediante métodos evolutivos durante los últimos 25 años. En este contexto, la evaluación del rendimiento de estos métodos es aún un tema en desarrollo. Sin embargo, a partir de una revisión de la literatura desarrollada en este trabajo, es posible advertir la ausencia de un marco de trabajo que se organizó convenientemente los progresos alcanzados en este campo de investigación. En consecuencia, la presente investigación tiene como objetivo proponer un marco de trabajo que permita no solo organizar los avances actuales, sino también identificar posibles medidas aún no propuestas. Se incluye además un análisis de las principales tendencias en este campo de investigación. El principal resultado obtenido a partir del marco de trabajo propuesto es el predominio de medidas de rendimiento basadas en el promedio de la calidad de la solución obtenida por el algoritmo en términos de la función objetivo.

Keywords: Medidas de rendimiento, Optimización dinámica evolutiva, Evaluación de algoritmos.

1. Introducción

En los últimos 25 años [1, 2], la solución a escenarios decisión conducentes a modelos de optimización dinámicos ha despertado un creciente interés desde el punto de vista de la

*Autor para la correspondencia
Correo-E: pnovoa@uteq.edu.ec (Pavel Novoa Hernández), jtubay@uteq.edu.ec (José Luis Tubay Vergara), ftriana@uteq.edu.ec (Freddy Enrique Triana Litardo), apuris@gmail.com (Amilkar Yudier Puris Cáceres)
[1]Universidad Estatal de Quevedo

Computación Inteligente (o Soft Computing) [3]. Diversos métodos aproximados, denominados metaheurísticos [4, 5], principalmente basados en paradigmas evolutivos, han sido aplicados con éxito a estos problemas dinámicos de optimización (PDO) [6, 7, 1, 8, 2, 9]. Sin embargo, un tema aún en desarrollo es el relacionado con la evaluación del rendimiento de estos métodos, más concretamente, cómo caracterizar cuantitativamente la ejecución de un algoritmo en ambientes dinámicos. En la actualidad existen algunos avances importantes, dentro de los cuales se destacan las revisiones realizadas en [10] y más recientemente [1]. En el primer trabajo, aunque se propone una clasificación intuitiva de las medidas, solo se analizan algunas medidas propuestas hasta el año 2002, por lo que se dejan fuera otras que aparecieron posteriormente. En el segundo caso, no se propone una clasificación explícita, pero se incluyen medidas más recientes.
No obstante los avances alcanzados, esta área del conocimiento demanda la propuesta de un marco de trabajo que permitan identificar claramente los progresos actuales y posibles trabajos futuros. En ese sentido, la presente investigación tiene por objetivo de proponer un marco de trabajo para las medidas de rendimiento en ambientes dinámicos que revele los progresos actuales y posibles líneas de investigación.

2. Metodología utilizada

En el desarrollo de la presente investigación se aplicó una búsqueda profunda de referencias bibliográficas relacionadas con el tema de las medidas de rendimiento en ambientes dinámicos. Concretamente se consultaron un total de 32 referencias en el catálogo de Scopus (www.scopus.com), las cuales después de una revisión detallada fueron consideradas solo 21. Posteriormente estas referencias fueron organizadas siguiendo un orden conceptual primero y cronológico después. En lo que sigue se detalla como fueron empleadas dichas referencias para llevar a cabo la revisión de la literatura, y la elaboración de la propuesta, esto es, un marco de trabajo para las medidas de rendimiento en ambientes dinámicos.

3. Medidas de rendimiento en ambientes dinámicos

Un problema dinámico de optimización se define formalmente de la manera siguiente:

$$\frac{max f^t}{x \in \Omega}(x) \tag{1}$$

donde f:$R^n x N_0 \longrightarrow R$ es la función objetivo, $\Omega \subseteq R^n$ es el espacio de búsqueda, y t $\in N_0$ es el tiempo. Para un algoritmo el objetivo es encontrar el conjunto de soluciones óptimas X^t en cada tiempo, tal que:

$$X^t = x^* \in \Omega | f^t(x), \forall x \in \Omega \tag{2}$$

Existen no obstante algunos autores como [11, 7] que asumen que el objetivo del algoritmo es encontrar no solo las mejores soluciones posibles en el tiempo actual, sino seguirlas después de la ocurrencia de un cambio en el ambiente. De manera similar, otros objetivos han sido adoptado en otras investigaciones, según [12] identifica los siguientes:

1. la exactitud o precisión (accuracy) del algoritmo (en función de las soluciones encontradas).
2. la estabilidad o capacidad del algoritmo de mantener un grado de exactitud aceptable a pesar de los cambios.
3. la reacción o tiempo que necesita el algoritmo para recuperar, de manera aproximada, la exactitud de etapas anteriores.

Aunque no de manera explícita, las investigaciones realizadas por [12] permiten concluir que las medidas existentes, y que intentan caracterizar a los algoritmos en función de los objetivos anteriores, se basan en el investigador que asuma como exactitud (precisión) de manera puntual, esto es, por unidad de tiempo. En nuestra opinión, estas medidas puntuales brindan fotogramas valiosos del rendimiento del algoritmo en un determinado momento, y en su conjunto permiten caracterizar el rendimiento a nivel de ejecución. En consecuencia, las diferencias entre las distintas medidas propuestas no solo está en la elección de una determinada medida puntual, sino también en el tiempo de muestreo (medición) y en la función de agregación empleada para resumir estos datos puntuales. Se expone las principales medidas que se consideran puntuales y que según algunos autores definen la exactitud de un algoritmo en ambientes dinámicos. Como se podrá advertir, la denominación brindada en esta investigación es más general que la adoptada por otros autores, dado que permite considerar otras medidas no necesariamente relacionadas con la exactitud.

4. Medidas puntuales

Son el mejor fitness (valor en la función objetivo de una solución), el fitness promedio de la población P, y la exactitud [13] (marcadas intencionalmente por un tiempo t).

$$bestFitness^t = max f_i | f_i = f^t(x_i), i = 1, ..., \|P\| \tag{3}$$

$$avgFitness^t = \frac{1}{\|P\|} \sum_{i=1}^{\|P\|} f_i \tag{4}$$

$$accuracy^t = \frac{bestFitness - f_{min}}{f_{max} - f_{min}} \tag{5}$$

donde $\|P\|$ es la cantidad de soluciones candidatas (individuos) con que cuenta el algoritmo en el tiempo t, por otro lado, f_{max} y f_{min} son los valores extremos de la función objetivo del problema.

En particular, f_{max} es el valor de la función objetivo en la solución óptima del problema, esto es, $f_{max} = f^t(x^*)$. Nótese que en esta última medida la información sobre el máximo y el mínimo del problema puede no estar disponible en determinados escenarios como algunos problemas reales. Para resolver esta dificultad, [10] propone como estimación de esta medida, la exactitud dentro de una ventana de tiempo:

$$windowAcc^t = max\frac{f(x_i) - f^w_{worst}}{f^w_{best} - f^w_{worst}} : i = 1, ..., \|P^t\| \tag{6}$$

donde w es una unidad de tiempo (evaluación o iteración) del algoritmo (después de ocurrir un cambio), además:

$$f^w_{worst} = minf(x_i) : i = 1, ..., \|P\|, t - w \leq t^{'} \leq t \tag{7}$$

$$f^w_{best} = maxf(x_i) : i = 1, ..., \|P\|, t - w \leq t^{'} \leq t \tag{8}$$

Otras medidas puntuales basadas en el valor de la función objetivo y no analizadas en [10] y [12], pero sí en investigaciones más recientes ([14],[15],[16]) son el mejor error, y el error promedio. Estas medidas vienen dadas respectivamente por las siguientes expresiones:

$$bestError^t = |f^t(x^*) - f^t(\ ^x)| \tag{9}$$

$$avgError^t = \sum_{i=1}^{\|P\|} |f^t(x^*) - f^t(x_i)| \tag{10}$$

donde x̂ es la mejor solución encontrada por el algoritmo en el tiempo t.

De manera similar a las anteriores, existen otras medidas puntuales basadas en el vector posición de las soluciones, esto es, a diferencia de las anteriores que emplean el valor de la función objetivo. Obviamente, las basadas en la posición suelen ser más exactas que las basadas en el fitness, pero suponen conocida la solución óptima del problema. En este grupo se encuentran la mejor distancia [10] y la distancia del centroide de la población [17]:

$$bestDist^t = max\frac{\delta_{max} - \delta(x^*,\ ^x)}{\delta_{max}} : i = 1, ..., \|P\| \tag{11}$$

$$centerDist^t = max\frac{\delta_{max} - \delta(x^*,\ = x)}{\delta_{max}} : = x = \frac{1}{\|P\|}\sum_{i=1}^{\|P\|} x_i \tag{12}$$

donde δ representa una función de distancia entre dos vectores. Usualmente es la distancia euclidiana. Por otra parte, δ_{max} es la máxima distancia (extensión) que puede existir entre dos puntos del espacio de búsqueda del problema.

5. Medidas de rendimiento

A partir de las medidas dadas anteriormente, diversos autores han propuesto otras más sofisticadas para caracterizar el rendimiento de los algoritmos en una ejecución. Dos aspectos son fundamentales en la definición de estas últimas: el tiempo de medición y la función de agregación para obtener el valor final de cada ejecución.[18] emplea una versión ponderada de la exactitud dada por la Expresión (3):

$$perfAcc^t = \frac{1}{G}\sum_{g=1}^{G}\alpha_g.accuracy^{(g)} \tag{13}$$

donde G es el total de iteraciones (generaciones) durante la ejecución; $\alpha_g \epsilon R^+$ es el peso de la exactitud en generación g. Obsérvese que si todos los $\alpha_g \epsilon R^+$ son igual a 1, entonces se tiene un promedio exacto de la exactitud del algoritmo a lo largo de la ejecución. Básicamente, la intención de los autores al incluir estos pesos fue ponderar determinadas generaciones durante la ejecución (ej. después de detectado el cambio o aquellas donde el óptimo se alcanza de manera exacta: $accuracy^{(t)} = 1$). Con el objetivo de obtener el rendimiento del algoritmo a lo largo de la ejecución, en [19] se propone el error y el rendimiento fuera de línea, los cuales pueden verse como promedios de las medidas puntuales bestError y bestFitness, respectivamente:

$$offlineError = \frac{1}{T}\sum_{t=1}^{T}bestError^t \tag{14}$$

$$offlinePerf = \frac{1}{T}\sum_{t=1}^{T}bestFitness^t \tag{15}$$

donde T es el total de evaluaciones en una ejecución del algoritmo sobre un instancia del problema. Basado en un objetivo similar e interesado en la adaptación de la población durante la ejecución, De [20] definió el rendimiento en línea a través de la siguiente expresión:

$$onlinePerf = \frac{1}{G}\sum_{g=1}^{G}avgFitness^g \tag{16}$$

Otras medidas similares son propuestas por [21]. La diferencia radica en que los autores consideran el rendimiento en función de las generaciones y no de las evaluaciones. Las medidas propuestas son denominadas por los autores como adaptabilidad y exactitud (Expresiones 17 y 18, respectivamente):

$$adaptability = \frac{1}{K}\sum_{k=1}^{k}\left(\frac{1}{G_k}\sum_{g=1}^{G_k}bestError^g\right) \tag{17}$$

$$stageAccuracy = \frac{1}{K}\sum_{k=1}^{k} bestError^{G_k} \tag{18}$$

donde K es la cantidad de etapas (número de cambios) del problema, y G_k es el número de generaciones dentro de la etapa k.

Quizás el aspecto más interesante de estas propuestas es que asumen un enfoque diferente al de [20] y [19] , que consideran o bien todas las evaluaciones o generaciones, aquí la ejecución del algoritmo es dividida en etapas. En particular, stageAccuracy establece que el objetivo no es encontrar la mejor solución en todo momento, sino antes de cada cambio, específicamente en la última generación.

Por otra parte, las medidas de estabilidad y reacción de [10, 12] se definen de la siguiente forma:

$$stability = max(0, acc^{g-1} - acc^{g}) \tag{19}$$

$$\varepsilon - reactivity = min(g^{'} - g : g < g^{'} \leq G, \frac{acc^{g^{'}}}{acc^{g}} \geq (1-\varepsilon)) \tag{20}$$

donde acc^{g-1} y acc^{g} son dos medidas puntuales (e.g. cualquiera definida en la sección anterior) correspondientes a las generaciones g-1 y g respectivamente. Además $\varepsilon \epsilon R^{+}$ y $g^{'} \epsilon N$. Nótese que stabilty tomará valores en el intervalo [0,1] de usarse la Expresión (3) para la exactitud, en cualquier caso un valor cercano a 0 significa que el algoritmo es estable. En contraste, $\varepsilon - reactivity$ no expresa un valor relativo como la anterior y está dada en unidades de tiempo, específicamente en generaciones. El significado de su valor es el tiempo que necesitó el algoritmo para alcanzar una exactitud cercana a la exactitud de una generación previa (ej. antes del cambio). En particular, un valor bajo implica una reacción alta. Estas medidas son particularmente especiales pues al depender de lo que se considere como exactitud el investigador puede obtener diferentes variantes de las mismas. Esto fue aprovechado convenientemente por Weicker, el cual condujo sendos análisis experimentales en [10, 12] para determinar cuan informativas resultan estas medidas variando la medida puntual que define a la exactitud.
Un trabajo posterior desarrollado por [22] estableció una variante generacional del rendimiento fuera de línea de Branke (Expresión 14). Esta medida denominada por su autor como Collective fitness viene dada por la expresión siguiente:

$$collectiveFitness = \frac{1}{G}\sum_{g=1}^{G} bestError^{g} \tag{21}$$

Cuando se conoce que el problema es multimodal y se tiene información sobre la posición y valor de la función objetivo de los óptimos (picos) entonces se pueden aplicar medidas como las propuestas por [11**?**] y [23]. La primera está orientada a medir cuantos picos son cubiertos por el algoritmo, mientras que la segunda mide el error en función del pico correspondiente al óptimo.

Recientemente se han propuesto otras medidas más sofisticadas como de [24] para cuantificar la degradación del fitness de la mejor solución del algoritmo después de la ocurrencia de un cambio. En particular, para definir esta medida los autores emplean regresión lineal con el objetivo de estimar la exactitud total del algoritmo. Un enfoque novedoso es el de [25], en el que se aplican múltiples pruebas de hipótesis a las series temporales de varios algoritmos. La medida puntual empleada por este tipo de análisis es el bestFitness correspondiente a cada evaluación de la ejecución.

6. Un marco de trabajo para las medidas de rendimiento en ambientes dinámicos

No obstante los progresos alcanzados anteriormente, en la actualidad es posible identificar algunos temas abiertos en este campo de investigación. En ese sentido, uno de estos temas es la carencia de un marco de trabajo que unifique y organice las medidas existentes. En nuestra opinión, dicho marco favorecería la identificación de posibles medidas aún no diseñadas, y mucho menos aplicadas en escenarios dinámicos. Hasta la fecha el intento más serio (y quizás el único) es la clasificación propuesta por [10]. En esta clasificación las medidas quedan organizadas de acuerdo a la información que necesitan estas a partir del algoritmo y del problema. En el primer caso existen dos posibilidades:

1. el fitness,
2. el genotipo (o fenotipo)

En relación a la información del problema, las medidas pueden requerir de:

1. la posición del óptimo (que es el caso de los problemas artificiales)
2. el fitness del óptimo, ninguna información.

A partir de esta clasificación es posible determinar a grosso modo en que escenarios se puede aplicar una determinada medida, en especial si se clasifica a la medida teniendo en cuenta la información del problema. No obstante los beneficios derivados de esta clasificación, en opinión de los autores de este trabajo, la misma posee algunas limitaciones importantes pues no permite identificar:

1. si la medida requiere o no conocer aspectos de la dinámica del problema (cuándo ocurren los cambios, la severidad y frecuencia de los mismos, entre otros),
2. qué unidad de tiempo se utiliza para las mediciones puntuales (evaluaciones, generaciones, entre otras medidas), y
3. qué tipo de función de agregación emplea la medida para caracterizar una ejecución (media, mediana, máximo, entre otras funciones).

El marco de trabajo (Figura 1) incluye las dimensiones propuestas por [10] pero adiciona otras igual de importantes: el tiempo de medición y la función de agregación. Además, a la dimensión información del problema se le ha añadido la dinámica del problema con la intención de incluir aquellos problemas en los que se asuman conocido este

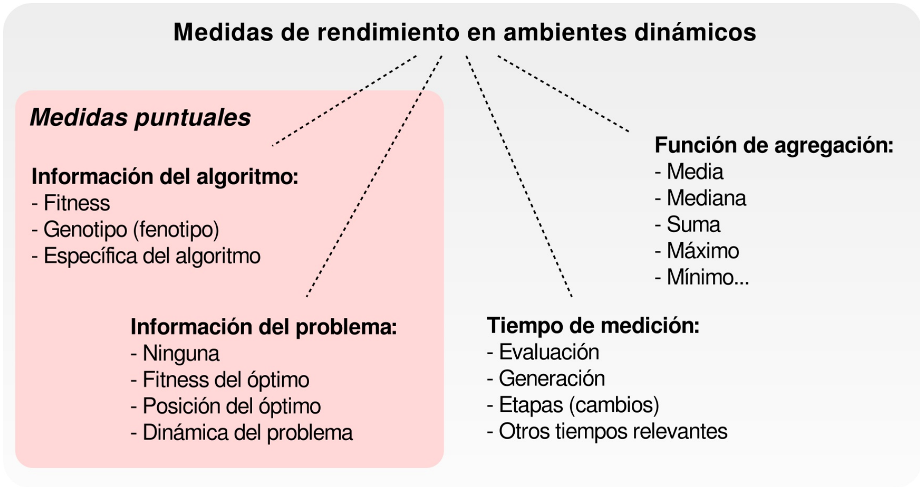

Figura 1: Marco de trabajo propuesto para las medidas de rendimiento en ambientes dinámicos

tipo de información (cuándo ocurren los cambios, cuáles y cómo varían los elementos del modelo, entre otros). Es importante advertir que esto último no implica necesariamente que se supongan conocidas otras informaciones como la posición o el fitness del óptimo global del problema, lo cual puede ser el caso de algunos escenarios reales.

Para ilustrar como utilizar este marco de trabajo para organizar las medidas existentes, el Cuadro 1 muestra cómo quedan clasificadas a partir de las dimensiones propuestas.

Tabla 3.1 Organización de las medidas de rendimiento según el marco de trabajo propuesto.

Referencia	Medida de rendimiento	Inf. algoritmo	Inf. problema	T. medición	F. agregación
(De Jong, 1975)	$onlinePerf^{(G)} = \frac{1}{G}\sum_{g=1}^{G}\left(\frac{1}{\mu}\sum_{i=1}^{\mu} f_i^{(g)}\right)$	Fitness de la población	–	Generaciones	Media
(Branke, 2002)	$offlineError^{(T)} = \frac{1}{T}\sum_{t=1}^{T} bestError^{(t)}$	Fitness	Fitness, Dinámica del problema	Evaluaciones	Media
(Branke, 2002)	$offlinePerf^{(T)} = \frac{1}{T}\sum_{t=1}^{T} bestFitness^{(t)}$	Fitness	Dinámica del problema	Evaluaciones	Media
(Trojanowski y Michalewicz, 1999)	$stageAccuracy^{(K)} = \frac{1}{K}\sum_{k=1}^{K} bestError^{(Gk)}$	Fitness	Fitness, Dinámica del problema	Etapa, Generaciones	Media
(Trojanowski y Michalewicz, 1999)	$adaptability^{(K)} = \frac{1}{K}\sum_{k=1}^{K}\left[\frac{1}{G_k}\sum_{g=1}^{G_k} bestError^{(k)}\right]$	Fitness	Fitness, Dinámica del problema	Etapa, generaciones	Media
(Weicker, 2002)	$windowAcc = \frac{f_{best} - f_{W,worst}}{f_{W,best} - f_{W,worst}}$	Fitness	–	Etapa, Generaciones	Media
(Weicker, 2002)	$bestDistAccuracy^{(G)} = \frac{1}{G}\sum_{g=1}^{G} bestDist^{(g)}$	Genotipo	Posición del óptimo	Generaciones	Media
(Weicker, 2002)	$stability^{(G)} = \max\left\{0, acc^{(g-1)} - acc^{(g)}\right\}$	Depende de *acc*	Depende de *acc*	Etapa, Generaciones	Media
(Weicker, 2002)	$reactivity_{\varepsilon}^{(G)} = \min\left\{g' - g \;:\; \frac{acc(g')}{acc(g)} \geq (1-\varepsilon)\right\}$	Depende de *acc*	Depende de *acc*	Etapa, Generaciones	Media
(Morrison, 2003)	$collectiveFitness = \frac{1}{G'}\sum_{i=1}^{G'} bestFitness^{(g)}$	Fitness	–	Generaciones	Media
(Schönemann, 2007)	$bestPerf^{(G)} = \{bestFitness^{(g)} \;:\; g = 1,\ldots,G\}$	Fitness	–	Generaciones	Mediana
(Li y Yang, 2008a)	$meanAbsoluteError^{(T)} = \frac{1}{K}\sum_{k=1}^{K} bestError^{(Tk)}$	Fitness	Fitness, Dinámica del problema	Etapa, Evaluaciones	Media
(Shilane et al, 2009)	t-estadígrafos basados en $bestFitness^{(g)}$	Fitness	–	Generaciones	Media
(Alba y Sarasola, 2010)	Regresión lineal basada en $bestFitness^{(g)}$	Fitness	–	Generaciones	Media

Cuadro 1. Medidas de rendimiento en ambientes dinámicos organizadas según el marco de trabajo propuesto.

Como se aprecia en el Cuadro 1, en escenarios en los que se asume conocido el fitness del óptimo del problema es posible aplicar medidas como el error y rendimiento fuera de línea. En particular, estas medidas también necesitan conocer, al menos según la definición de [19], cuando ocurren los cambios en el ambiente, para actualizar convenientemente el fitness de la mejor solución del algoritmo. De lo contrario, tanto el error como el rendimiento fuera de línea pueden considerar valores erróneos. Esto ocurre en escenarios donde los cambios afecten directamente a la función objetivo (la altura de los picos). Lo ideal sería que la medida no empleara información alguna del problema, sin embargo, esto implica al menos para el caso de la optimización en ambientes dinámicos, una disminución de la precisión en la medición.

Por otro lado, nótese que existen casos en que la información que necesita las medidas depende de la requerida por la medida puntual utilizada. En este caso se encuentran la estabilidad y la reacción que dependen de la definición de la medida puntual acc. En cualquier caso, ambas necesitan conocer cuando ocurren los cambios para cada etapa del problema. Otro aspecto destacable en cuanto a la información utilizada por las medidas es el predominio de medidas basadas en el fitness en relación con las basadas en la posición. En nuestra opinión, esta tendencia se justifica por una razón fundamental: la posición del óptimo del problema solo está disponible en escenarios artificiales, por lo que la aplicación de estas medidas se verá limitada a este contexto. En cuanto al resto de las dimensiones de la clasificación propuesta, estas nos brindan otros detalles técnicos importantes. La dimensión agregación expresa el tipo de función empleada para resumir los datos de las mediciones. En ese sentido se puede ver que existe una tendencia a emplear la media para caracterizar una ejecución [20, 19, 21, 10], sin embargo, otros autores argumentan que este estadígrafo es sensible a valores muy dispersos entre sí. En contraste, emplean otros más robustos como la mediana, tal es el caso de [**?**] que aplica esta función de agregación no para caracterizar una ejecución en particular, sino para obtener una serie temporal partir de varias ejecuciones. La dimensión tiempo de medición nos permite definir el nivel de la medición.

En ese sentido, como se aprecia en la clasificación propuesta existen medidas que se toman los datos provenientes de todas las evaluaciones o generaciones de la ejecución, como es el caso del error y el rendimiento fuera de línea [19], mientras que otras medidas como stageAccuracy [21] y meanAbsoluteError [26] solo emplean la última generación (evaluación) antes de la ocurrencia de cada cambio.

Nótese que ambas medidas utilizan expresiones similares, pero con tiempos de medición distintos. A diferencia de las fuera de línea de [19], estas medidas asumen que el objetivo del algoritmo es encontrar la mejor solución en cada etapa (problema estacionario) k de manera secuencial. Otras medidas más sofisticadas quedan organizadas fácilmente por la clasificación propuesta, como se aprecia en los casos de las hipótesis múltiples de [25] y la degradación del fitness de [**?**] que se basan en el bestFitness de cada generación.

Es de notar también que la clasificación permite identificar posibles medidas aún no propuestas y que pudieran ser quizás más representativas en determinados escenarios. La mayoría de las medidas analizadas emplean la media como función de agregación, ¿qué sucedería si se empleara la mediana? Asimismo, la tendencia es realizar las mediciones en cada evaluación o generación de la ejecución, en ese sentido diferentes medidas pueden obtenerse si se realizan en las evaluaciones o generaciones antes del cambio.

Finalmente, es necesario destacar que otra clasificación es posible si se tiene en cuenta lo que investigador define como el objetivo de un algoritmo en ambientes dinámicos. En ese sentido, pudieran emplearse las categorías propuestas por [10]: exactitud, estabilidad y reacción.

Sin embargo, en nuestra opinión estas categorías son subjetivas y resultan en ocasiones ambiguas, note la diferencia de los nombres dados a medidas con expresiones similares y que han sido definidas por distintos autores. Ese es el caso de stageAccuracy [21] y de meanAbsoluteError [26] (Cuadro 1). A esto se le añade el hecho de que otros objetivos son posibles, y no están contemplados de manera explícita por las categorías de Weicker, pudiera ser de interés saber cuán eficiente pudiera ser, en términos de tiempo, un algoritmo en ambientes dinámicos.

7. Conclusiones

A partir de la aplicación del marco de trabajo se pudo determinar que, como tendencia general, predomina el empleo de medidas basadas en el fitness (valores de la función objetivo) y que emplean como función de agregación la media de los datos. Asimismo, se puede apreciar que existen medidas con nombres diferentes y que tienen expresiones matemáticas similares. Finalmente, otra observación importante derivada de esta clasificación, es que existen medidas propuestas recientemente que son más sofisticadas y que incluyen algún tipo de procesamiento estadístico de los datos.

Referencias

[1] T. Nguyen, S. Yang, J. Branke, Evolutionary dynamic optimization: A survey of the state of the art, Swarm and Evolutionary Computation.
URL http://doi.org/http://dx.doi.org/10.1016/j.swevo.2012.05.001

[2] T. Nguyen, S. Yang, J. Branke, X. Yao, Evolutionary Dynamic Optimization: Methodologies", publisher = Evolutionary Computation for Dynamic Optimization Problems, editor = En S. Yang and X. Yao, address = Berlin, Heidelberg, publisher = Springer Berlin Heidelberg, pages = 39–64, year = 2013, url = http://doi.org/10.1007/978-3-642-38416-5_2.

[3] J. Verdegay, R. Yager, P. Bonissone, (2008), On heuristics as a fundamental constituent of soft computing. Fuzzy Sets and Systems 159 846–855.

[4] B. Melián, M. Pérez, M. V. JA., JM., (2003), Metaheurísticas: Una visión global. Revista Iberoamericana de Inteligencia Artificial.

[5] I. Boussaïd, J. Lepagnot, P. Siarry, A survey on optimization metaheuristics, Information Sciences 237 (2013) 82–117.

[6] P. Novoa-Hernández, D. Pelta, C. Corona, (2010), Studies in Computational Intelligence 284 371–383.

[7] C. Cruz, J. González, D. Pelta, (2011), Optimization in dynamic environments: a survey on problems, methods and measures. Soft Computing.

[8] M. Plessis, A. Engelbrecht, (2013), Metaheuristics for Dynamic Optimization 433 117–145.
URL http://doi.org/10.1007/978-3-642-30665-5_7

[9] P. Novoa-Hernández, C. Corona, P. C., DA., (2016), Self-adaptation in dynamic environments - a survey and open issues. International Journal of Bio-inspired Computation.

[10] K. Weicker, Performance measures for dynamic environments, Parallel Problem Solving from Nature - PPSN VII 2439 (2002) 64–73.
URL http://doi.org/10.1007/3-540-45712-7_7

[11] J. Branke, Evolutionary optimization in dynamic environments, Genetic Algorithms and Evolutionary Computation (Vol.

[12] K. Weicker, Evolutionary Algorithms and Dynamic Optimization Problems, Institut für Formale Methoden der Informatik der Universität Stuttgart, 2003.

[13] W. Feng, T. Brune, L. Chan, M. Chowdhury, C. Kuek, Y. Li, Benchmarks for testing evolutionary algorithms (Tech, Rep. No. CSC-97006), 1997.

[14] P. Novoa-Hernández, C. Corona, D. Pelta, (2011), Efficient multi-swarm PSO algorithms for dynamic environments. Memetic Computing.

[15] P. Novoa-Hernández, C. Corona, D. Pelta, (2013), Self-adaptive, multipopulation differential evolution in dynamic environments. Soft Computing.
URL http://doi.org/10.1007/s00500-013-1022-x

[16] P. Novoa-Hernández, C. Corona, D. Pelta, (2015), A software tool for assisting experimentation in dynamic environments. Applied Computational Intelligence and Soft Computing 5.

[17] S. R., P. Eggenberger, Adaptation on the evolutionary time scale: A working hypothesis and basic experiments, Artificial Evolution: Third European Conf., AE'97, Berlin, 1997.

[18] N. Mori, H. Kita, Y. Nishikawa, Adaptation to a changing environment by means of the thermodynamical genetic algorithm, Parallel Problem Solving from Nature, 1996.

[19] J. Branke, Memory enhanced evolutionary algorithms for changing optimization problems, Proceedings of the Congress on Evolutionary Computation (Vol) 3 (1999) 1875–1882.

[20] D. Jong, K., An analysis of the behavior of a class of genetic adaptive systems, University of Michigan, 1975.

[21] K. Trojanowski, Z. Michalewicz, Searching for optima in non-stationary environments, En Evolutionary Computation 99. (1999) 2348.
URL http://doi.org/10.1109/CEC.1999.785498

[22] R. Morrison, Performance Measurement in Dynamic Environments, GECCO Workshop on Evolutionary Algorithms for Dynamic Optimization Problems, 2003.

[23] S. Bird, X. Li, Informative performance metrics for dynamic optimization problems, En 9th Conf, 2007.

[24] E. Alba, B. Sarasola, Measuring fitness degradation in dynamic optimization problems, En EvoApplications 1 (2010) 572–581.

[25] D. Shilane, J. Martikainen, S. Ovaska, (2009), ICANNGA 2009.

[26] C. Li, S. Yang, T. Nguyen, E. Yu, X. Yao, Y. Jin, H. G. Beyer, Suganthan, PN, 2008.