

Aplicación del método de bisección para la mejora automática del comportamiento a-todo-momento del algoritmo ASO

Juan C. Moreno-Torres, Marco A. Moreno-Armendáriz

Instituto Politécnico Nacional, Centro de Investigación en Computación,
CDMX, México

Resumen. El comportamiento a-todo-momento dota a un algoritmo de robustez al permitirle ser interrumpido en cualquier punto del proceso, mientras que se garantiza que se entregara la mejor respuesta posible cuando suceda la solicitud de paro. Este trabajo propone una metodología para identificar la relación de los parámetros de un algoritmo de optimización por colonia de hormigas con su comportamiento a-todo-momento, permitiendo implementar una técnica de mejora automática del comportamiento a-todo-momento utilizando una reducción en factores logarítmicos del espacio de búsqueda, disminuyendo la demanda computacional requerida para esta tarea.

Palabras clave: bisección, mejora automática, comportamiento a-todo-momento, algoritmo ASO.

Application of the Bisection Method for the Automatic Improvement of the All-time Behavior of the ASO Algorithm

Abstract. The all-time behavior provides a robustness algorithm by allowing it to be interrupted at any point in the process, while ensuring that the best possible response is delivered when the stop request occurs. This work proposes a methodology to identify the relationship of the parameters of an optimization algorithm by colony of ants with its behavior at-all-moment, allowing to implement a technique of automatic improvement of the behavior at-all-moment using a reduction in logarithmic factors of the search space, decreasing the computational demand required for this task.

Keywords: Bisection Method, Automatic Improvement, All-time Behavior, ASO Algorithm.

1. Introducción

El algoritmo Ant System Optimization (ASO) requiere de seis parámetros que definen su comportamiento al explorar un espacio de soluciones: m , Q , α , β , ρ y N [1]. La combinación de estos parámetros es lo que define la calidad de la optimización y su velocidad de convergencia [3].

Para su selección se han presentado propuestas iterativas, utilización de herramientas evolutivas y variación calendarizada de los parámetros durante la ejecución del algoritmo; sin embargo estas propuestas causan en incremento de la complejidad del algoritmo, dependencia al factor que fue considerado prioritario (calidad de la optimización o tiempo de convergencia en una búsqueda informada), o en el caso de las propuestas evolutivas la adición de sus propios parámetros a definir como óptimos. En 2010 se realizó una compilación que abarca la última década de investigación en este ámbito, en el cual se concluye que la correcta selección de parámetros y su completa comprensión en relación con su impacto temporal ofrece una reducción en complejidad y recursos computacionales, permitiendo eliminar el uso de herramientas adaptativas adicionales [5].

En todas las propuestas comentadas, para definir un conjunto de parámetros como “el mejor” se requiere de una suposición: que se cuenta con un tiempo indefinido para proponer la solución o que se conoce una aproximación al óptimo de antemano. López-Ibáñez en [4] propone un método de selección de parámetros basado en la optimalidad de Pareto, que define “mejor” como aquellos parámetros que deriven en conjuntos no-dominados que generen el mayor hipervolumen. El objetivo de este método de discriminación de combinaciones es generar la respuesta que converja en el menor tiempo posible y con la mayor calidad de optimización. Dicho método aplica exploración iterativa por todo el espacio de soluciones propuesto para cada parámetro; sin embargo este proceso puede resultar costoso en recursos computacionales, aún si solo requiere realizarse una vez para generar una solución para todo un conjunto de problemas de la misma clase, de manera que implementar una reducción del espacio de búsqueda resulta relevante.

El campo de soluciones que se requiere explorar es descrito en la ecuación (1), considerando *iter* como *iteración* y *param* como *parámetro*.

$$\begin{aligned} &\text{No. de combinaciones a explorar con el algoritmo López-Ibáñez}(X, N, Z, W, U, Y) = \\ &X \cdot N \cdot \sum_{h=1}^Z \prod_{j=1}^W \prod_{i=1}^U \left(\binom{Y(i, j, h)}{2} \right) \\ Y(i, j, h) = &\begin{cases} \text{ceil} \left(\left| \frac{(\text{param}(\text{max})_{ijh}) - \text{param}(\text{min})_{ijh}}{\Delta \text{param}_{ijh}} \right| \right) & \text{Si } Z=1, \text{ delta o none} \\ \text{ceil} \left(\left| \frac{(\text{param}(\text{max})_{ijh}) - \text{param}(\text{min})_{ijh}}{\Delta \text{param}_{ijh}} \right| \right) \cdot (\text{iter}_{ijh}) & \text{Si } Z=2, \text{ switch} \end{cases} \quad (1) \end{aligned}$$

donde:

$$\begin{aligned} &\Delta\text{parámetro, parámetro}(\max), \text{parámetro}(\min) \in \mathbb{R} \\ &\quad \text{parámetro}(\max) > \text{parámetro}(\min) \\ &\quad \text{iteración} \in \mathbb{N}, \text{iteración} \leq N \\ Z = \{1, 2\}, \text{comportamiento delta o none} = 1, \text{comportamiento switch} = 2 \\ &\quad X, Y, W, U, Y \in \mathbb{N} \end{aligned}$$

- X: número de problemas de prueba a utilizar.
- N: número de iteraciones máximas a aplicar para cada problema.
- Z: número de comportamientos a analizar, López-Ibañez definió delta, switch y none.
- W: número de parámetros en el algoritmo, para este caso de MMAS es 5, el mismo que para ASO.
- U: número de avances diferentes para cada parámetro, que dependerá de lo definido por el usuario según el algoritmo.
- Y: número de elementos en el dominio de cada parámetro, que dependerá de lo definido por el usuario según el algoritmo.
- Δ parámetro: avance considerado para el parámetro.
- parámetro(max): valor máximo considerado para el parámetro.
- parámetro(min): valor mínimo considerado para el parámetro.
- iteración: iteración en la que el parámetro realiza el cambio de valor en el comportamiento switch.

El método propuesto en este trabajo se enfoca en generar una reducción del espacio de búsqueda propuesto en [4] que se realiza de manera iterativa por todo su espacio de posibles soluciones, a través de la adaptación del algoritmo de bisección para búsqueda de raíces [2], con el fin de converger a una respuesta con una menor demanda computacional; para alcanzar este objetivo se requiere conocer la relación que mantiene cada parámetro con el comportamiento a-todo-momento del algoritmo estudiado, para lo cual se propone una metodología basada en el análisis del desempeño combinando calidad de optimización y velocidad de convergencia.

2. Preliminares

2.1. Perfil de desempeño

El perfil de desempeño P de un algoritmo a-todo-momento, denota la calidad $q_i(t)$ de la respuesta esperada en función del tiempo t_i [6].

$$P = \{(t_1, q_1), (t_2, q_2), \dots, (t_n, q_n)\}.$$

2.2. Optimalidad de Pareto en perfiles de desempeño

Dados dos perfiles de desempeño P_i y P_j , que son resultado de dos algoritmos A_i y A_j , en la misma instancia n , se dice que P_i es mejor que P_j , en términos de optimalidad de Pareto, si y solo si $P_i \neq P_j$ y $\forall (t_j, q_j) \in P_j, \exists (t_i, q_i) \in P_i$ tal que $(t_i, q_i) \preceq (t_j, q_j)$ [4].

2.3. Hipervolumen de un perfil de desempeño

El hipervolumen $H(P_i)$ de un conjunto de vectores objetivo con punto de referencia $(0, \dots, 0)$ es enunciado como: $H(P_i) = \sum_{k=(0, \dots, 0)}^{(1, \dots, 1)} \overline{P}_i \cdot \gamma(k)$, de manera que $\overline{P}_i = \{(\overline{t}_i, \overline{q}_i) | \forall (\overline{t}_i, \overline{q}_i), \exists (t_i, q_i) \in P_i\}$ [4,7], donde

$$\overline{P}_i = \begin{cases} \overline{t}_i = 1 - (0,9 \cdot \frac{t_i - t_{min}}{t_{max} - t_{min}}), \\ \overline{q}_i = 1 - (0,9 \cdot \frac{q_i - q_{min}}{q_{max} - q_{min}}), \end{cases} \quad (2)$$

$$\gamma(k) = \begin{cases} 1 & \text{si } A \succcurlyeq \{k\}, \\ 0 & \text{caso contrario.} \end{cases} \quad (3)$$

k : Vector objetivo en el perfil de desempeño.

2.4. Método de bisección

Es un método numérico de búsqueda de una raíz en una función que se sabe se encuentra dentro de un intervalo conocido.

Si una raíz de una función $f(x)$ ha sido identificada dentro de los intervalos (x_1, x_2) , entonces $f(x_1) \cdot f(x_2) < 0$. Se calcula un $f(x_3)$, tal que $x_3 = (x_1 + x_2)/2$ es el punto medio del intervalo. Si $f(x_2)f(x_3) < 0$, entonces la raíz debe encontrarse entre (x_3, x_2) ; para continuar la búsqueda se sustituye x_1 por x_3 . Si $f(x_2) \cdot f(x_3) \not< 0$ entonces la raíz debe encontrarse entre (x_1, x_3) y se realiza la sustitución de x_2 por x_3 . En cualquiera de ambos casos el nuevo intervalo solo contiene la mitad de los elementos del intervalo original. Este procedimiento se continúa hasta que el intervalo es reducido a un valor ε definido como límite de búsqueda, tal que $|x_2 - x_1| \leq \varepsilon$.

El intervalo original Δx es reducido a $\Delta x/2$ tras la primer bisección, $\Delta x/2^2$ tras dos bisecciones, continuando hasta n bisecciones, donde se tiene $\Delta x/2^n = \varepsilon$. Despejando para el número de operaciones necesarias: $n = \frac{\ln(|\Delta x|/\varepsilon)}{\ln 2}$ [2].

3. Propuesta

3.1. Adecuación del método de bisección para la mejora automática del comportamiento a-todo-momento

La exploración del algoritmo heurístico en todo el dominio de cada parámetro expresado en la ecuación 1, representa el mayor impacto en cuanto al número de operaciones requeridas en la búsqueda de la mejora del comportamiento a-todo-momento; particularmente se enfocaran la atención en el comportamiento de parámetros constantes que comparte dominio con delta, pues es un caso particular cuando solo hay un Δ parámetro $_{ijh}$, mientras que el comportamiento switch solo es una multiplicidad del segundo, siendo la reducción en switch una derivación de la reducción del comportamiento delta.

El dominio de $Y(i, j, h)$ con el mayor número de elementos sucede con: $\min(\Delta$ parámetro $_{ijh})$, por lo que será considerado como caso base de estudio.

La implementación de un algoritmo de búsqueda por bisección permite reducir la búsqueda en la siguiente expresión (considerando $iter$ como *iteración* y par como *parámetro*):

$$Yb(i, j, h) = \begin{cases} \left(\frac{\text{ceil} \left(\frac{\ln \left(\left| \frac{\text{par}(\max)_{ijh} - \text{par}(\min)_{ijh}}{\Delta \text{par}_{ijh}} \right| \right)}{\ln 2} \right)}{\text{ceil} \left(\frac{\ln \left(\left| \frac{\text{par}(\max)_{ijh} - \text{par}(\min)_{ijh}}{\Delta \text{par}_{ijh}} \right| \right)}{\ln 2} \right)} \right) & \text{Si } Z=1, \text{ delta o none} \\ \left(\frac{\text{ceil} \left(\frac{\ln \left(\left| \frac{\text{par}(\max)_{ijh} - \text{par}(\min)_{ijh}}{\Delta \text{par}_{ijh}} \right| \right)}{\ln 2} \right)}{\text{ceil} \left(\frac{\ln \left(\left| \frac{\text{par}(\max)_{ijh} - \text{par}(\min)_{ijh}}{\Delta \text{par}_{ijh}} \right| \right)}{\ln 2} \right)} \right) \cdot (\text{iter}_{ijh}) & \text{Si } Z=2, \text{ switch} \end{cases} \quad (4)$$

Las reglas de decisión del algoritmo de bisección propuestas para el recorte del dominio de cada parámetro son las siguientes:

Sea un intervalo de búsqueda $[a, b]$ con un Δu , $c = a + (\text{round}(\frac{1 + \frac{b-a}{\Delta u}}{2}) \cdot \Delta u)$ es el punto medio del intervalo, donde $a < c < b$. P_i es el perfil de desempeño dada una evaluación en el valor i , y $H(P_i)$ es el hipervolumen resultante del perfil de desempeño P_i .

Si $(H(P_a) > H(P_b) > H(P_c)) \vee (H(P_a) > H(P_c) > H(P_b)) \vee (H(P_c) > H(P_a) > H(P_b))$

entonces el espacio de búsqueda de la siguiente iteración sera $[a, c]$.

Si $(H(P_b) > H(P_a) > H(P_c)) \vee (H(P_b) > H(P_c) > H(P_a)) \vee (H(P_c) > H(P_b) > H(P_a))$

entonces el espacio de búsqueda de la siguiente iteración sera $[c, b]$.

Siendo la condición de paro $\frac{b-a}{\Delta u} = 0$, ó $\max(Yb(i, j, h))$ iteraciones.

A continuación se muestra en pseudocódigo el algoritmo de Bisección a-todo-momento para el comportamiento *none* con $\min(\Delta \text{parámetro})$.

```

n = No. de parametros del algoritmo
Definición del espacio de búsqueda para cada parámetro  $a_n \leq Param_n \leq b_n$ 
Definición del avance para cada parámetro  $\Delta Param_n$ 
Cálculo de  $P_n = Yb_n$  para cada parámetro
nit = max( $P_n$ ) condición de paro
Cálculo de  $c_n$  de cada parametro
while x ≤ nit do
  while y ≤ n do
    Evaluacion del algoritmo heurístico (ASO) con  $Param_y = a_y$ 
    Evaluacion del algoritmo heurístico (ASO) con  $Param_y = b_y$ 
    Evaluacion del algoritmo heurístico (ASO) con  $Param_y = c_y$ 
    Almacenamiento de perfiles de desempeño y conjunto de parámetros asociados
    Comparación del hipervolumen de entre los tres puntos evaluados
    Aplicación de las reglas de decisión para la actualización de  $c_y$ 
    y++
  end while
  x++
end while
Cálculo del hipervolumen normalizado para todo perfil de desempeño obtenido
Selección del hipervolumen normalizado más alto y extracción de sus parámetros asociados
Fin del algoritmo

```

3.2. Metodología de selección de rangos

Una característica primordial para el funcionamiento del algoritmo de bisección a-todo-momento es que se debe garantizar un par de elementos $[a_n, b_n]$

para todo parámetro, entre los cuales se sepa que se encuentra el máximo global con un comportamiento monótono creciente o decreciente, pero de los cuales se desconoce su valor exacto e interacción con otros parámetros.

La metodología propuesta a continuación permitirá identificar los umbrales adecuados para su selección:

1. Identificación del dominio de cada parámetro.
2. Definición de un conjunto de parámetros predeterminados.
3. Cálculo del hipervolumen para variaciones independientes de cada parámetro, manteniendo el resto fijos.
4. Cálculo del hipervolumen con variación en dos parámetros, manteniendo el resto fijos, explorando regiones de interés extraídas del paso anterior.
5. Selección de las regiones con hipervolumenes más altos con comportamientos monótonos donde $a_n < b_n$.

Debido a los componentes estocásticos del algoritmo ASO, los pasos 3 y 4 deben ser analizados estadísticamente.

4. Caso de estudio: Mejora automática del comportamiento a-todo-momento por bisección del algoritmo ASO para el TSP

4.1. Selección de rangos de entrada para el algoritmo de bisección a-todo-momento

El dominio de cada parámetro se encuentra definido en [1]. Para todas las pruebas realizadas en esta sección se utilizó las instancia de la librería del agente viajero TSPLIB aplicando Eil51 y KroD100 a menos que se mencione lo contrario, mientras que se realizaron 30 evaluaciones con 300 iteraciones cada una. Los parámetros predeterminados fueron seleccionados en concordancia con las recomendaciones de [1] y [4], y son mostradas en la tabla 1.

El primer análisis corresponde a recorrer por un dominio útil cada parámetro de manera independiente, manteniendo el resto en un valor fijo por todo el proceso, con la finalidad de conocer el impacto individual en el comportamiento a-todo-momento del algoritmo. Dos gráficas son presentadas para cada variable para su posterior análisis. La primera es una representación de caja de todos los resultados, la segunda el comportamiento de la media aritmética, incluyendo sus desviaciones estándar para cada punto. En las figuras 1 a 4 se puede observar el comportamiento de dichas variables.

Tabla 1: Parámetros predeterminados y dominios útiles considerados.

m	α	β	ρ	Q
0.8*No. de ciudades	1.0	2.0	0.2	1.0
$1 < m < 120$	$0,1 < \alpha < 5,0$	$0,1 < \beta < 5,0$	$0,1 < \rho < 0,9$	$1 < Q < 10$

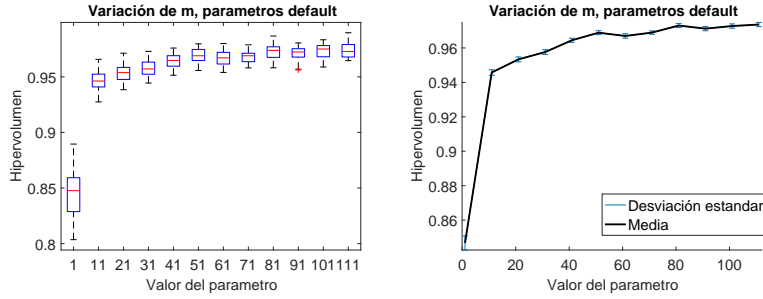


Fig. 1: Comportamiento a-todo-momento con variación de m.

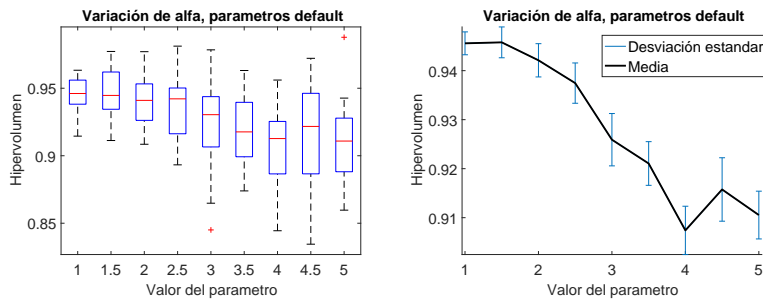


Fig. 2: Comportamiento a-todo-momento con variación de alfa.

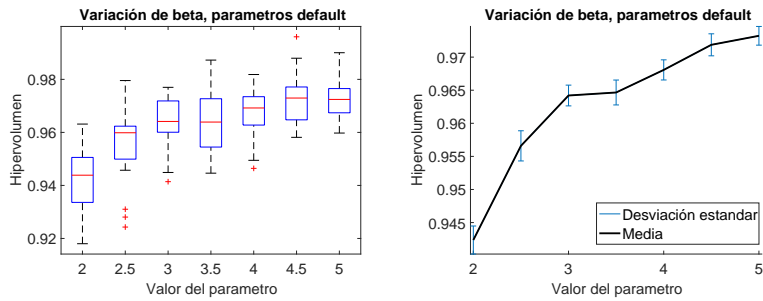


Fig. 3: Comportamiento a-todo-momento con variación de beta.

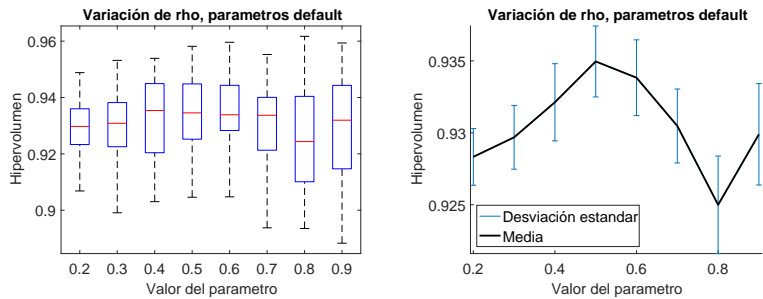


Fig. 4: Comportamiento a-todo-momento con variación de rho.

Con la finalidad de identificar relaciones entre pares de parámetros un enfoque similar se aplicó, excepto que esta vez realizando variaciones sobre dos parámetros de manera secuencial. Los mismos valores predeterminados se aplicaron para los parámetros fijos, así como el mismo número de iteraciones y evaluaciones. El parámetro de comparación elegido para esta tarea fue α , ya que de las gráficas de comportamiento individual mostró la mayor influencia sobre el comportamiento a-todo-momento, además de encontrarse directamente influenciada por 3 de los 4 otros parámetros en las ecuaciones del algoritmo. En las figuras 5 y 6 se observa la respuesta obtenida.

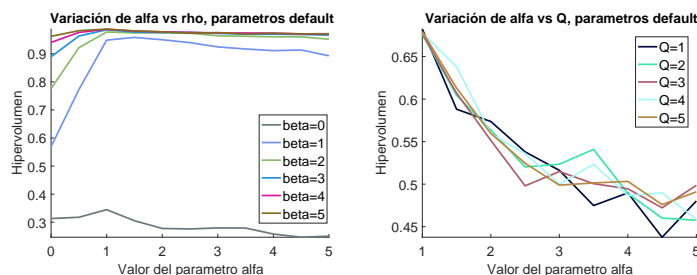


Fig. 5: Comportamiento a-todo-momento con variación de alfa y beta (izquierda), comportamiento a-todo-momento con variación de alfa y Q (derecha).

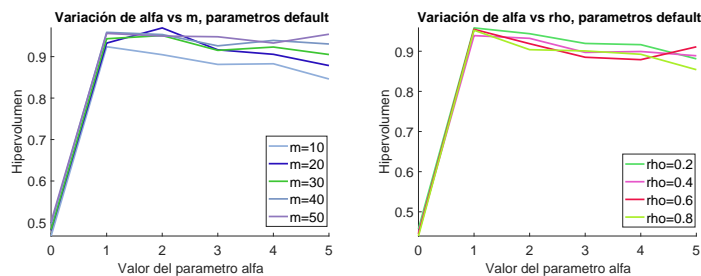


Fig. 6: Comportamiento a-todo-momento con variación de alfa y m (izquierda), comportamiento a-todo-momento con variación de alfa y rho (derecha).

Finalmente, se seleccionaron las regiones que cumplen con las restricciones del algoritmo de bisección a-todo-momento y se procedió a implementar una comparación del algoritmo de mejora del comportamiento a-todo-momento de López-Ibáñez según el pseudocódigo descrito en [4] y el propuesto en este trabajo.

Los algoritmos fueron evaluados 5 veces consecutivas independientes por 200 iteraciones bajo los dominios de la tabla 2:

Tabla 2: Rangos a explorar con los algoritmos de mejora automática del comportamiento a-todo-momento.

$20 \leq m \leq 80$	$0,5 \leq \alpha \leq 3,5$	$2 \leq \beta \leq 5$	$0,4 \leq \rho \leq 0,7$	$1 \leq Q \leq 3$
$\Delta m = 10$	$\Delta \alpha = 0,5$	$\Delta \beta = 0,5$	$\Delta \rho = 0,1$	$\Delta Q = 1$

Para el algoritmo López-Ibáñez el número de elementos a estudiar en ASO con los rangos mostrado arriba corresponde a $Y(i, j, h) = 1944$ por evaluación;

por otro lado para el algoritmo de bisección a-todo-momento los elementos por evaluación corresponden a $Yb(i, j, h) = 108$. La media resultante para cada método se muestra en la tabla 3:

Tabla 3: Media de los parámetros resultantes de la mejora automática del comportamiento a-todo-momento.

	m	α	β	ρ	Q
Bisección Completo	54	1.3	4.6	0.68	1.8
Algoritmo López-Ibáñez	78	1.6	4.6	0.7	1.6

El comportamiento a-todo-momento con los dos conjuntos de parámetros obtenidos es evaluado a través del hipervolumen para 30 evaluaciones con 300 iteraciones cada una con la finalidad de apreciar las respuestas generadas de manera estadística. La media aritmética y desviación estándar servirán como comparativas, y pueden se pueden ver en la tabla 4.

Tabla 4: Comparación de desempeño del hipervolumen entre algoritmos de mejora automática del comportamiento a-todo-momento.

	Media aritmética del Hipervolumen generado	Desviación Estándar del Hipervolumen generado	Mejor respuesta del Hipervolumen generado
Bisección Completo	0.8521	0.0581	0.9588
Algoritmo López-Ibáñez	0.8178	0.0693	0.9621

Se registró el tiempo de procesamiento para las 5 pruebas de cada algoritmo, a continuación se presenta el resultado promedio. Todas las iteraciones se realizaron sobre un procesador Inter i5 de 2.5MHz programadas en MATLAB.

Tabla 5: Comparación de desempeño temporal entre algoritmos de mejora automática del comportamiento a-todo-momento.

	Tiempo de ejecución promedio por prueba (segundos)
Bisección Completo	1857
Algoritmo López-Ibáñez	34,991

4.2. Análisis de los datos

Las gráficas del comportamiento de α y ρ nos permiten identificar un rango claro donde el comportamiento genera el hipervolumen mas alto y los cuales cumplen las características exigidas por el algoritmo propuesto.

Los parámetros m , Q y β permiten como domino todos los números naturales, por lo que el estudio de su comportamiento se ve limitado a los rangos seleccionados que son considerados como factibles a utilizarse. Este tipo de análisis refleja que Q no mantiene relación con el comportamiento a-todo-momento del algoritmo, ya que su variación no afecta significativamente la respuesta del hipervolumen, debido a esta razón y la limitante de espacio sus gráficas son omitidas del documento. El parámetro m por otro lado mantiene una relación directamente proporcional con el comportamiento a-todo-momento del algoritmo. Se puede identificar que un valor demasiado alto de β reduciría el elemento heurístico del algoritmo, por lo que debe ser delimitado de manera que no evite la integración de la búsqueda al azar.

El tiempo de ejecución para el caso de estudio utilizando Bisección a-todo-momento en el peor de los casos resultó ser 18.8 veces menor que el algoritmo López-Ibáñez, esto debido al método de búsqueda que redujo las iteraciones necesarias de igual manera 18 veces. Analizando los perfiles de desempeño de los tres algoritmos bajo el criterio de optimalidad de Pareto y teniendo en cuenta la reducción en tiempo de ejecución, el algoritmo de Bisección Completo se perfila como el óptimo.

5. Conclusiones

Este trabajo presenta dos principales aportes al análisis y mejora del comportamiento a-todo-momento del algoritmo ASO. El primero propone una metodología para analizar el impacto de cada parámetro con respecto a la respuesta a-todo-momento, el cual no se limita a este caso de estudio específico, sino que es aplicable sobre cualquier algoritmo heurístico que genere un perfil de desempeño con elementos no-dominados y su comportamiento dependa de la selección de un conjunto de parámetros. El segundo aporte radica en la propuesta de la implementación del algoritmo de bisección en la mejora automática del comportamiento a-todo-momento desarrollada en [4], los resultados para la instancia analizada mostraron una reducción de 18.8 veces el tiempo de ejecución del algoritmo original, mientras que convergen a parámetros que ofrecen resultados similares.

Agradecimientos. Los autores extendemos un agradecimiento al gobierno Mexicano (Instituto Politécnico Nacional, EDI-IPN, COFAA-IPN, SIP-IPN, BEIFI-IPN, CONACYT, SNI), por proporcionar el apoyo necesario para el desarrollo de este trabajo; los resultados aquí presentados son posibles gracias a estas instituciones.

Referencias

1. Dorigo, M., Maniezzo, V., Coloni, A.: Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 26(1), 29–41 (1996)
2. Kiusalaas, J.: *Numerical methods in engineering with MATLAB*, vol. 1. Cambridge University Press (2005)
3. Lobo, F., Lima, C.F., Michalewicz, Z.: *Parameter setting in evolutionary algorithms*, vol. 54. Springer Science and Business Media (2007)
4. López-Ibáñez, M., Stützle, T.: Automatically improving the anytime behaviour of optimisation algorithms. *European Journal of Operational Research* 235(3), 569–582 (2014)
5. Stützle, T., López-Ibáñez, M., Pellegrini, P., Maur, M., De Oca, M.M., Birattari, M., Dorigo, M.: Parameter adaptation in ant colony optimization. *Autonomous search* pp. 191–215 (2011)

6. Zilberstein, S.: Using anytime algorithms in intelligent systems. *AI magazine* 17(3), 73 (1996)
7. Zitzler, E., Brockhoff, D., Thiele, L.: The hypervolume indicator revisited: On the design of pareto-compliant indicators via weighted integration. In: *Evolutionary multi-criterion optimization*. pp. 862–876. Springer (2007)