

Tipo de artículo: Artículo original  
Temática: Computación científica  
Recibido: 10/02/18 | Aceptado: 06/04/18 | Publicado: 27/04/18

## **Algoritmo de Estrategia Evolutiva y Algoritmo de Estimación de Distribuciones: Ajustes proactivos**

### *Evolution Strategies Algorithm and Estimation of Distribution Algorithm: Proactive Adjustments*

Claudia María Diez Borges<sup>1</sup>, Meidel Samuel Pérez Martín<sup>2</sup>, Mailyn Moreno Espino<sup>3</sup>

1 Universidad Tecnológica de la Habana “José Antonio Echeverría”, [cladiezbo@gmail.com](mailto:cladiezbo@gmail.com)

2 Universidad Tecnológica de la Habana “José Antonio Echeverría”, [meidelperez@gmail.com](mailto:meidelperez@gmail.com)

3 Universidad Tecnológica de la Habana “José Antonio Echeverría”, [my@ceis.cujae.edu.cu](mailto:my@ceis.cujae.edu.cu)

\* Autor para correspondencia: [cladiezbo@gmail.com](mailto:cladiezbo@gmail.com)

---

#### **Resumen**

Tomar el control para que las cosas sucedan y no esperar a que ocurran eventualmente es tener un comportamiento proactivo. Dotar a los sistemas informáticos de este comportamiento permite que se le deleguen metas al software y este trabaje para cumplirlas siempre que tenga condiciones para hacerlo.

Existen varios trabajos donde se ha incursionado en este tema y se ha demostrado a través de varios experimentos que las variantes del software con comportamiento proactivo son mejores que sus variantes no proactivas. Principalmente en las metaheurísticas se ha utilizado para escapar de óptimos locales y encontrar la mejor solución posible modificando operadores y/o parámetros de las mismas. El aporte principal de este trabajo consiste en proponer planes de ajustes proactivos para un Algoritmo de Estrategias Evolutivas y un Algoritmo de Estimación de Distribuciones, basados en la modelación del algoritmo en modelos  $i^*$ , y analizar a través de experimentos si se obtuvo mejoría con respecto a sus variantes no proactivas. Como resultado se obtuvo un plan de ajuste al operador de remplazo del Algoritmo de Estrategia Evolutiva y al Algoritmo de Estimación de Distribuciones. Se comprobó que existen cambios significativos con respecto a su variante no proactiva.

**Palabras clave:** Metaheurísticas; proactividad; ajuste proactivo; Estrategias Evolutivas; Algoritmo de Estimación de Distribuciones

#### *Abstract*

*Taking control to things happen and not waiting for them to happen eventually is to have a proactive behavior. Giving computer systems this behavior allows you to delegate goals to the software and this will work to fulfill them whenever it has conditions to do it. There are several works where this topic has been explored and it has been demonstrated through several experiments that software variants with proactive behavior are better than their non-proactive variants. Mainly in metaheuristics have been use to escape from local optima and find the best possible solution by modifying operators and / or parameters of them self. The main contribution of this work consists in proposing proactive adjustment plans for an Evolutionary Strategies and a Estimation of Distributions Algorithm, based on the modeling of the algorithm in  $i^*$  models, and analyzing them through experiments if was obtained improvement with respect to the non-proactive variants. As a result, an adjustment plan was obtained for the replacement operator of the Evolutionary Strategy Algorithm and the Distribution Estimation Algorithm. It was found that there are significant changes with respect to the non-proactive variant.*

**Keywords:** *Metaheuristics; proactivity; proactive adjustment; Evolutionary Strategies; Algorithm of Estimation of Distributions.*

---

## Introducción

Uno de los principios fundamentales del mundo es la búsqueda de un estado óptimo [1]. En el transporte [2], en la medicina [3], en la tecnología espacial e incluso en el ámbito militar [4], los ingenieros se enfrentan a problemas de optimización en los cuales tienen que tomar decisiones para optimizar costos y/o esfuerzos.

Para dar solución a los problemas de optimización existen varias técnicas que se pueden agrupar en exactas y aproximadas. Las técnicas exactas para garantizar el óptimo presentan un gran costo computacional, ya que en ocasiones el problema se cataloga como NP-completo y otros en NP-Hard que no hay método computacional matemático único que llegue al óptimo en un tiempo y en un costo razonable [5, 6]. Esto ha provocado que se tenga un mayor interés en las técnicas aproximadas, específicamente las metaheurísticas [5], ya que proveen buenas soluciones en un tiempo razonable [6-8].

Existen varias metaheurísticas como son *Búsqueda Tabú*, *Búsqueda aleatoria*, *Algoritmo del Gran Diluvio* y *Algoritmos Evolutivos*, entre otras. No se puede demostrar que una metaheurística es mejor que otra, ya que en un problema pueden funcionar bien y en otro simplemente pueden tener un desempeño pobre, acorde al *No Free Lunch Theorem* (NFLT) [9].

Uno de los problemas que presentan las metaheurísticas es el estancamiento en óptimos locales. Con el objetivo de aminorar este problema, se han desarrollado planes de ajustes en [7, 10-13]. para dotar a las

metaheurísticas de comportamiento proactivo.

Siguiendo el mismo objetivo, en este trabajo, se presentan dos variantes proactivas, una para una Estrategia Evolutiva y otra para un Algoritmo de Estimación de Distribuciones.

## **Materiales y métodos o Metodología computacional**

Las metaheurísticas se dividen en dos grandes grupos según el número de soluciones que procesan en **S-Metaheurísticas** o metaheurísticas basadas en un punto y en **P-Metaheurísticas** o metaheurísticas poblacionales [6]. Las S-Metaheurísticas manejan una única solución en cada iteración del proceso de búsqueda y el paso de una solución a otra en cada iteración va describiendo una trayectoria en el espacio de búsqueda [6, 14] [6]. Mientras que las P-Metaheurísticas trabajan con un conjunto de soluciones que iterativamente son mejoradas a través de un proceso inteligente de exploración del espacio de búsqueda y construyen soluciones a partir de otras [5, 6, 15]. Existen varios software comerciales y bibliotecas de clases que permiten utilizar las metaheurísticas en diversos problemas [4]. Una de las bibliotecas que implementa un modelo unificado de algoritmos metaheurísticos es la Biblioteca de Clases para la Integración de Algoritmos Metaheurísticos (BiCIAM) [16].

Dentro de las P-Metaheurísticas existe un grupo de algoritmos denominados algoritmos evolutivos (EA, por sus siglas en inglés) que están basados en los mecanismos originales de evolución para seres vivos planteados por Charles Darwin, que seleccionan los mejores individuos adaptados a su ambiente, asegurando su descendencia.[17], La gran familia de los EA está compuesta por los Algoritmos Genéticos, Estrategias Evolutivas y los Algoritmos de Estimación de Distribuciones [6].

A pesar que el uso de un comportamiento proactivo permite que el software sea capaz de actuar sin la intervención de un humano [18], es una característica que ha recibido menos atención para el diseño de las metaheurísticas, aunque recientemente ha sido aplicada en algoritmos como *Record-to-Record Travel*, *Local Search* y Algoritmos Genéticos a través de planes de ajustes proactivos en algún parámetro u operador, obteniéndose un resultado en promedio mejor que sus variantes no proactivas [7, 10-13]. La proactividad consiste en tomar el control para hacer que las cosas sucedan, no esperar que ocurran eventualmente [19]. La proactividad, en el contexto informático, es cuando un software exhibe un comportamiento dirigido a metas, tomando la iniciativa con el fin de satisfacer sus metas de diseño [20], la misma se hace presente en los agentes inteligentes, rasgo que caracteriza a esta entidad de software y les permite tomar la iniciativa en orden de satisfacer sus objetivos [7, 18].

En la figura 1 se muestran los requisitos tempranos capturados para las Estrategias Evolutivas y los Algoritmos de Estimación de Distribuciones, los cuales van a ser representados por el actor "Metaheurísticas". En el diagrama se encuentran tres actores estratégicos: Metaheurísticas, Función Objetivo y Optimizador.

El "Optimizador" tiene una meta fuerte "Optimizar", para cumplimentar esta meta depende del actor "Metaheurísticas" (entidad de software) para "Encontrar mejor solución". Las "Metaheurísticas" son las encargadas de encontrar la mejor solución y para ello depende de "Función Objetivo" (software) para saber la evaluación de cada solución. La meta suave o intención del "Optimizador" es "No estancarse" en óptimos locales y para esto utiliza los planes de ajustar operadores y ajustar parámetros.

En la Figura 2 se representa la captura de requisitos tardíos a través de los patrones Hardgoal why Dependency y Resource why Dependency. A diferencia de la Figura 1 el "Optimizador" es solo responsable de optimizar y darle a las "Metaheurísticas" los recursos: "Operadores iniciales" y "Parámetros iniciales". Mientras, las "Metaheurísticas" asumen la meta suave "No estancarse" del actor "Optimizador", la cual depende de dos tareas "Ajustar Parámetros" y "Ajustar Operadores" que a su vez dependen de dos recursos "Operadores" y "Parámetros".

Las tareas "Ajustar Parámetros" y "Ajustar Operadores" son las encargadas de realizar los cambios necesarios en los recursos "Operadores" y "Parámetros" para garantizar que se cumpla la meta suave. Estos cambios pueden ser realizados en algún parámetro, operador o en ambos, lo cual depende de la decisión que se tome a la hora de realizar la captura de requisitos tardíos.

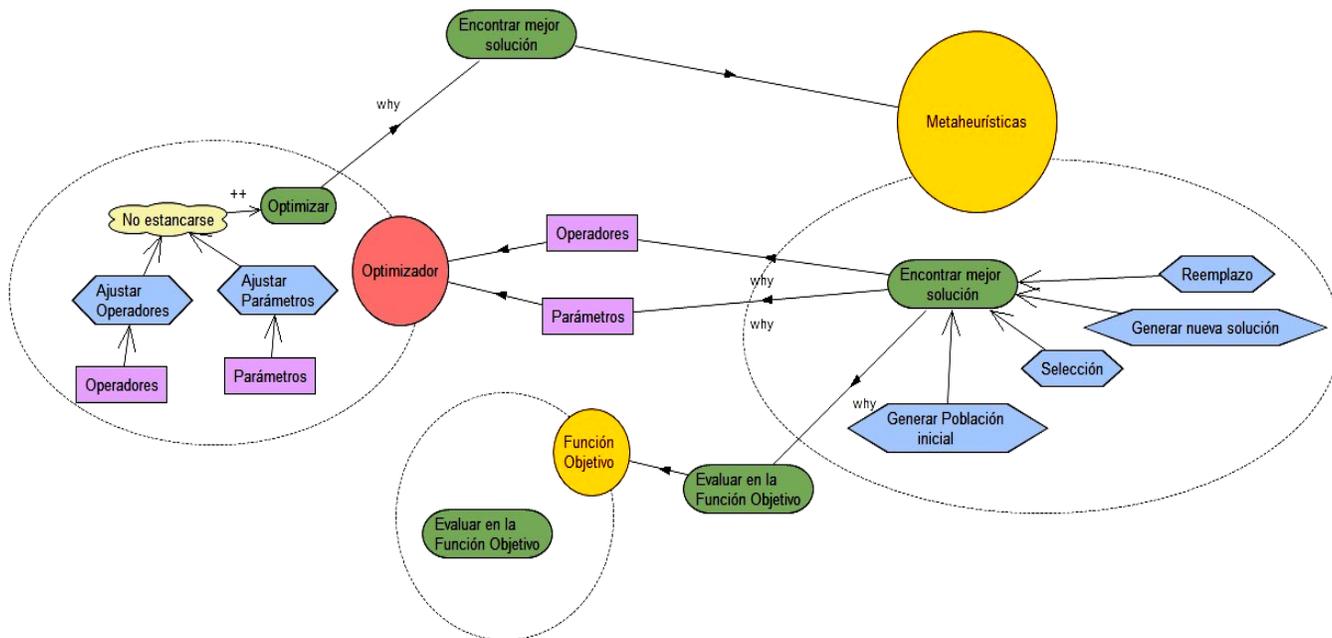


Figura 1: Requisitos tempranos de las Metaheurísticas

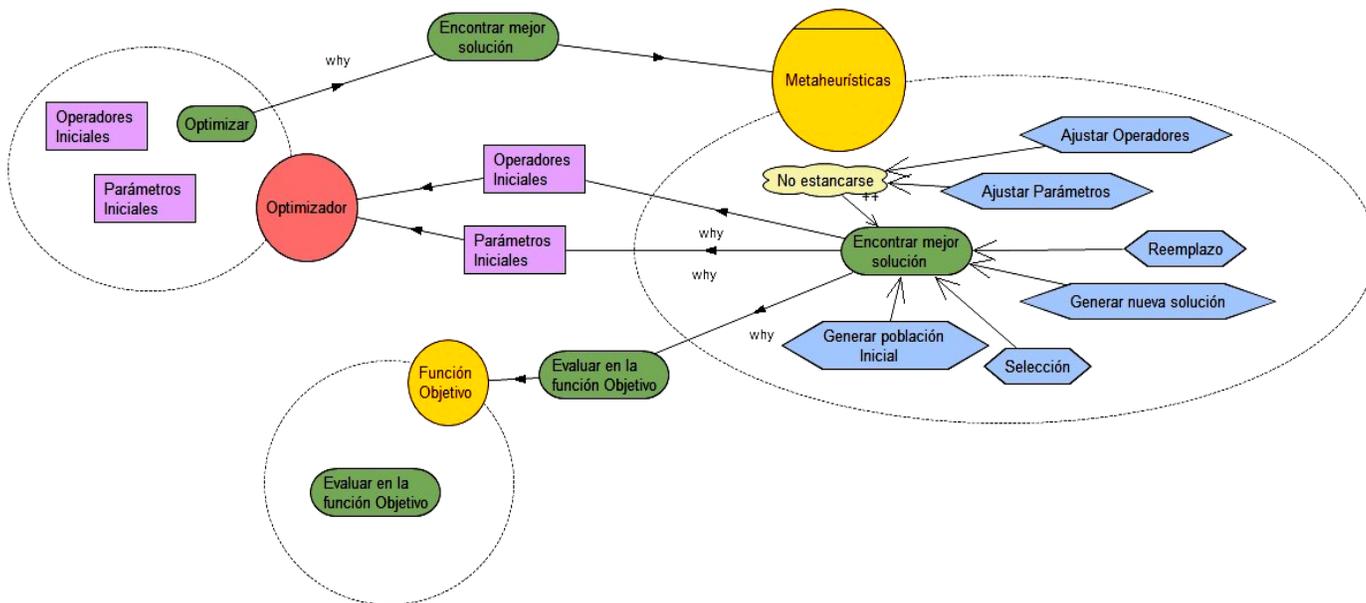


Figura 2: Requisitos tardíos de las Metaheurísticas

A partir de esta captura de requisitos se proponen los siguientes planes de ajuste:

Para las EE's se propuso un plan de ajuste enfocado en la generación de la nueva población, es decir se crea una especie de remplazo combinando los remplazos Estado Estable y Generacional. Este plan se aplicará cuando se obtenga una repetición  $x$  (definida por el usuario) del mejor valor en una población.

Para el EDA se propuso un plan de ajuste enfocado en modificar la población existente, es decir, se escogen de las anteriores mejores soluciones  $n$  elementos y se sustituyen por  $n$  elementos de la actual población. Este plan se aplicará cuando se repita  $x$  veces el mejor resultado de la población.

## **Resultados y discusión**

En los experimentos se comparan los algoritmos propuestos con sus variantes no proactivas, En el presente trabajo el problema de prueba utilizado fue el problema de los bloques, que consiste en dividir la cadena de bits de entrada en grupos consecutivos de 4 bits (bloques). Para evaluar las soluciones en este problema se utiliza como función objetivo OneMax donde se cuenta la cantidad de bits con valor 1 que hay en el bloque. Se le asigna un valor al bloque, dependiendo de esta cantidad, y se suman los valores de cada uno de los bloques para obtener el valor total de la cadena evaluada en la función objetivo.

Se realizó un experimento con las siguientes características:

- Se construye la población inicial asignando a cada uno de los 100 bits de la cadena un valor aleatorio de 0 o 1.
- El método de reemplazo usado fue el reemplazo Estado Estable.
- El tipo de selección utilizado fue la selección por Ruleta.
- Se trabajó con población de 100 individuos.
- En el caso de la variante proactiva, el ajuste proactivo se ejecuta cuando se obtenga de forma consecutiva 150 veces el mismo mejor valor.
- Para el EDA se utilizó como distribución aleatoria la invariante marginal.
- Para las EE's se empleó el operador de mutación en dos puntos.

Al ejecutar la prueba de Mann-Whitney con  $H_0: EDA = P\_EDA$  y  $H_1: EDA < P\_EDA$  con una muestra de 1000 elementos y  $\alpha = 0.99$  se obtuvo como resultado que se acepta  $H_1$ .

Para las EE's se evaluaron los resultados con la prueba Mann-Whitney quedando  $H_0: EE = P\_EE$  y  $H_1: EE < P\_EE$  con una muestra de 1000 elementos y  $\alpha = 0.95$  se obtuvo como resultado q se acepta  $H_1$ .

## Conclusiones

Con este trabajo se puede concluir que los patrones de requisitos formulados, basados en los modelos de  $i^*$ , permiten detectar requisitos proactivos para las Metaheurísticas, teniendo en cuenta estos requisitos capturados, se propuso un plan de ajuste proactivo para un EDA y una EE a partir del ajuste del operador de remplazo. Se obtuvo que las variantes proactivas propuestas son superiores, en cuanto a sus resultados, que sus versiones no proactivas.

## Referencias

- [1] T. Weise. (2009). "*Global Optimization Algorithms – Theory and Application – (Second ed.)*".
- [2] X.-S. Yang, *et al.*, "*Metaheuristics in Water, Geotechnical and Transport Engineering*". Teddington, Middlesex, TW11 0LW, UK: Elsevier, 2012.
- [3] E. Al-Daoud, *et al.*, "Enhanced Metaheuristic Algorithms for the Identification of Cancer MDPs", *International Journal of Intelligent Systems and Applications(IJISA)* vol. 6, p. 9, 2014.
- [4] S. Rao, "*Engineering Optimization Theory and Practice*". New Jersey, USA, 2009.
- [5] F. Glover, *et al.*, "*Handbook of Metaheuristics*". London, UK: Kluwer Academic Publishers, 2003.
- [6] E.-G. Talbi, "*Metaheuristics: From desing to implementation*". New Jersey, USA: Jon Wiley & Sons .INC., 2009.
- [7] M. Moreno, *et al.*, "An agent based approach for the implementation of cooperative proactive S-Metaheuristics", *Expert Systems With Applications*, vol. 63, p. 30, 2016.
- [8] Y. Donoso, *et al.*, "*Multi-Objective Optimization in Computer Networks Using Metaheuristics*". Florida, US: Auerbach Publications, 2007.
- [9] D. H. Wolpert, *et al.*, "No Free Lunch Theorems for Optimization", *IEEE Transactions on Evolutionary Computation*, vol. 1, p. 16, 1996.
- [10] M. Moreno, *et al.*, "An agent based implementation of proactive S-Metaheuristics", *8th International conference hybrid artificial intellligent systems*, vol. 8073, p. 9, 2013.
- [11] L. Bofill, "Ajustes proactivos para un Algoritmo Genético", Tesis de grado, Instituto Superior Politécnico "José Antonio Echeverría", La Habana, Cuba, 2015.

- [12] M. Moreno, "Patrones para incorporar proactividad en sistemas informáticos", Tesis de doctorado, Instituto Superior Politécnico "José Antonio Echeverría", La Habana, Cuba, 2013.
- [13] U. N. d. Colombia, "Proactive local search based on FDC", *DYNA*, vol. 81, p. 8, 2014.
- [14] Y. Noa, "Estrategias para mejorar el balance entre Exploración y Exportación en optimización de Enjambre de Partículas", Tesis de maestría, Universidad de La Habana, La Habana, Cuba, 2011.
- [15] L. Lanzarini, *et al.*, "Metaheurísticas poblacionales aplicadas a la resolución de problemas complejos", *XI Workshop de Investigadores en Ciencias de la Computación*, vol. 1, p. 5, 2009.
- [16] R. Díaz, "Nueva versión de la biblioteca de clases BiCIAM para solucionar problemas multiobjetivo", Tesis de diploma, Instituto Superior Politécnico "José Antonio Echeverría", La Habana, Cuba, 2014.
- [17] J. Dréo, *et al.*, "*Metaheuristics for Hard Optimization*". Berlin, Alemania: Springer, 2006.
- [18] M. Wooldridge, "*An Introduction to MultiAgent Systems*". Liverpool, UK: John Wiley & R Sons Ltd, 2002.
- [19] A. Grant, *et al.*, "The dynamics of proactivity at work", *Research in Organizational Behavior*, vol. 28, p. 32, 2008.
- [20] N. Jennings, "An agent-based approach for building complex software systems", *ACM*, vol. 44(4), p. 6, 2001.
- [21] E. Yu, "Modelling Strategic Relationships for Process Reengineering", Tesis de Doctorado, Computer Science, University of Toronto, Toronto, Canada, 1995.