

Algoritmos Evolutivos con Estimación de Distribución Celulares

Evolutionary Algorithms with Cellular Distribution Estimation

Yoan Martínez-López¹, Julio Madera Quintana¹, Ireimis Leguen de Varona¹

¹Department of Computer Sciences, Faculty of Informatics, Camagüey University, Camagüey City, 74650, Camagüey, Cuba.

Autor para correspondencia: yoan.martinez@reduc.edu.cu

Resumen

En este artículo se presenta un tipo de algoritmo evolutivo: los Algoritmos con Estimación de Distribuciones (EDA) celular, una nueva clase de algoritmos de optimización estocásticos basados en poblaciones. Como resultado de este trabajo se obtiene un algoritmo EDA con aprendizaje basado en pruebas de independencias y esquemas descentralizados para reducir el número de evaluaciones en la resolución de problemas de optimización discretos.

Palabras claves: aprendizaje, EDA celular, redes bayesianas

Abstract

This article describes a kind of evolutionary algorithm is presented: The Algorithms with cellular Estimation of Distribution (EDA), a new class of stochastic optimization algorithms based on populations. As a result of this work an EDA algorithm with learning based on independence tests and decentralized schemes to reduce the number of evaluations in solving discrete optimization problems is obtained.

Keywords: Bayesian networks, cellular EDA, learning

Introducción

La computación evolutiva (EC) es una de las ramas de la *Inteligencia Artificial* que se aplica para la resolución de problemas de *optimización combinatoria*, la cual está inspirada en los mecanismos de evolución biológica propuestos por *Darwin*, *Medel* y *Lamarck*, donde *Darwin* propuso la “*Selección natural de los más adaptados*”, *Medel* propuso la “*Teoría corpuscular de la herencia*” y *Lamarck* propuso la “*Herencia de caracteres adquiridos*”.

Dentro de la computación evolutiva, se engloban los diferentes algoritmos evolutivos (EA) o estrategias para la resolución de problemas de optimización. Estas estrategias son las siguientes:

- Procesos de Búsqueda Evolutiva: Fue propuesta por Alan Turing en el año 1948
- Estrategias Evolutivas (EE): Propuesto por Rechenberg en 1964. Representan a los individuos con Vectores reales.

- Programación Evolutiva (PE): Propuesto por Fogel en 1965. Utilizan máquinas de estado finito.
- Algoritmos Genéticos (AG): Propuesto por Holland en 1975. Representan a los individuos como cadenas binarias.
- Programación Genética (PG): Propuesto por Koza en 1992. Utilizan Árboles LISP.

Los EDA (*Estimation Distribution Algorithms*) son un grupo de algoritmos de EA que permiten ajustar el modelo a la estructura de un problema determinado, realizados por una estimación de distribuciones de probabilidades a partir de soluciones seleccionadas (Larrañaga, 2003). El modelo reflejado por el sesgo es una distribución de probabilidades. Estos algoritmos son metaheurísticas que se basan en sustituir los operadores de cruce y mutación de los individuos de los algoritmos genéticos por la estimación y posterior muestreo de una distribución de probabilidad aprendida a partir de los individuos seleccionados de una población. *Los algoritmos evolutivos celulares* son un tipo de algoritmos evolutivos de grupos discretos basados en estructuras espaciales, donde cada individuo interactúa con su vecino adyacente. Una vecindad solapada ayuda en la exploración del espacio de búsqueda, mientras que la explotación toma lugar dentro de una vecindad por operadores estocásticos. *Los algoritmos genéticos celulares* son propuestos por (Dorronsoro, 2008), los cuales manejan sus vecinos de acorde a la calidad de los individuos de una población un conjunto más razonable de estructura de la población basada en la velocidad de convergencia, entonces los parámetros de configuración para la población y las estructuras vecinas son menos necesarias. Otros investigadores (Lu, 2013), (Huang, 2015), (Alba, 2000), (Alba, 2008) propusieron una variante de algoritmos evolutivos que se basaban en la mejor capacidad de mantenimiento de la diversidad de mantenimiento de la población relativa al algoritmo genético canónico. *Los algoritmos de evaluación diferencial celular* son algoritmos propuestos por Storn y Price, son poblaciones basadas en algoritmos de optimización iterativos paralelos que superan a muchos métodos en términos de velocidad de convergencia y robustez sobre funciones de puntos de referencias comunes y en problemas reales, los cuales son influenciados por los siguientes criterios: un factor de escala F , un tamaño de población NP y una razón de recombinación (Mühlenbein, 1996; Ding, 2015).

Un EDA celular es una colección de EDA colaborativo y descentralizados, también llamados algoritmos miembros que desarrollan poblaciones solapadas (Alba, 2006). Un rasgo distintivo de esta clase de algoritmo es que se descentralizan a nivel de los algoritmos, la selección en otro algoritmo evolucionario usualmente ocurre a nivel de recombinación, ver figura 1.

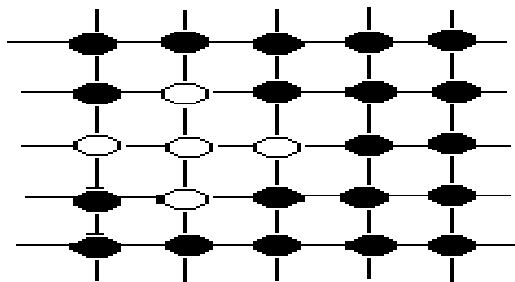


Figura 1. Representación de una malla o rejilla celular

La necesidad de encontrar el modelo que mejor se adapta a las características de un problema constituye un reto para los investigadores de este tema y da lugar a nuevas investigaciones que posibilitan que esta rama de la computación evolutiva se desarrolle cada vez más rápido. Además, un asunto crítico en estos modelos es la construcción de algoritmos que sean eficientes desde el punto de vista evaluativo. El objetivo de este trabajo es presentar una nueva metaheurística con aprendizaje basado en pruebas de independencias y esquemas descentralizados para reducir el número de evaluaciones en la resolución de problemas de optimización discretos.

Materiales y Métodos

La organización de los EDAs celulares se basan en la tradicional estructura 2D de vecinos solapados, se conocen mejor en términos de dos rejillas, esto quiere decir que una rejilla contiene cadenas y otra contiene conjuntos disjuntos de cadenas (células), ver figura 2.

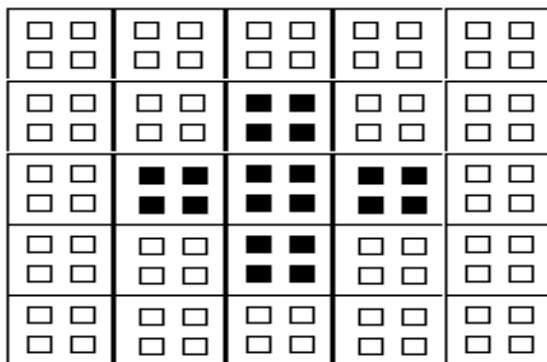


Figura 2. Representación de un EDA celular

En un EDA celular el ciclo reproductivo se desempeña dentro de cada número de población local, el cual es usualmente llamado célula, tienen sus propias poblaciones locales definidas por subpoblaciones vecinas y al mismo tiempo una célula pertenece a muchas poblaciones locales. El conjunto de todas las células define una partición de la población global

En el algoritmo se presenta el pseudocódigo del modelo de EDA celular propuesto. Cada iteración del EDA celular consiste de exactamente una iteración de todos los algoritmos miembros. Cada uno de estos algoritmos es responsable de actualizar exactamente una sub-población, y esto se realiza aplicando un modelo de EDA clásico local a la población compuesta sus individuos más los individuos de las subpoblaciones vecinas (según el vecindario definido)

Algoritmo pseudocódigo de un EDA celular simple

1. Poner $t \leftarrow 1$
2. Generar $N \gg 0$ puntos de forma aleatoria
3. **MIENTRAS** no se cumple el criterio de parada
 - a) **PARA CADA** celda
 - 1) Seleccionar localmente $M \leq \text{SizeOf}(\text{vecindario}) \times \text{SizeOf}(\text{celda})$ individuos del vecindario de acuerdo a un método de selección
 - 2) Estimar la distribución $p^s(x, t)$ de los M individuos seleccionados
 - 3) Generar $\text{SizeOf}(\text{celda})$ nuevos puntos de acuerdo a la distribución $p^s(x, t)$
 - 4) Insertar los individuos generados en la misma celda de una población auxiliar
 - b) Reemplazar la población actual por la auxiliar
 - c) Computar y actualizar las estadísticas
 - d) Poner $t \leftarrow t + 1$

En el paso de reemplazamiento (línea b), la vieja población puede ser tomada en cuenta (reemplazar un individuo si el nuevo que se generó es mejor) o no (reemplazar siempre los individuos de la vieja población por los nuevos generados).

Para inicializar y seleccionar los individuos de la población el algoritmo EDA celular se vale de operadores evolutivos los cuales determinan el modo en que el algoritmo explora el espacio de búsqueda, alguno investigadores han clasificado los esquemas de selección en dos grupos: selección proporcionada y selección ordinal, en la primera los individuos son seleccionados según el valor de función específico de cada uno, mientras que la selección ordinal solo tiene en cuenta el ranqueo del individuo dentro de la población (Echegoyen, 2012). Por otro lado, para estimar la distribución de probabilidad conjunta se recurre a las Redes Bayesianas, que es un tipo de modelo Gráfico Probabilístico.

Modelos Gráficos Probabilísticos.

Los Modelos Gráficos (MG) son herramientas que permiten representar distribuciones de probabilidad conjunta. Los Modelos Gráficos Probabilísticos (MGP) constituyen grafos en los cuales los nodos representan variables aleatorias y los arcos representan relaciones de dependencia condicional. Estos grafos proveen una forma compacta de representar la distribución de probabilidad (Madera, 2006).

Los MGP empleados por los algoritmos EDA varían en función del dominio de las variables del problema. Si estas variables son discretas se utilizan Redes Bayesianas. Si por el contrario se trata de variables continuas se utilizan Redes Gaussianas. Existe la posibilidad de generar modelos probabilísticas híbridos, adaptados para problemas con variables discretas y continuas.

Redes Bayesianas

Una red Bayesiana es un tipo de MG que utiliza Grafos Acíclicos Dirigidos (DAG), por lo que toma en consideración la dirección de los arcos. Una red Bayesiana se define mediante el par $\langle G, P \rangle$ donde G es un grafo que representa las relaciones de dependencia entre las variables y P es la factorización de la distribución de probabilidad representada por G .

Formalmente se define una red Bayesiana sobre un conjunto, $V = \{V_1, \dots, V_n\}$, de variables aleatorias. La factorización de la probabilidad conjunta puede expresarse como:

$$P(V) = P(V_1, V_2, \dots, V_n) = \prod_{i=1}^n P(V_i | Pa_i) \quad (1)$$

La expresión permite definir una Red Bayesiana con la condición de Markov (Yaramakala, 2005), cada variable (V_i) es independiente de cualquier subconjunto de las variables no descendiente de ella, condicionado en su conjunto de padres (Pa_i).

Para el aprendizaje de Redes Bayesianas (Margaritis, 2003) existen dos técnicas fundamentales: el aprendizaje basado en restricciones (*constraint based learning*), o algoritmos que detectan independencias, y el aprendizaje basado en optimización de métricas (*search-and-score based learning*), conocidos como métodos de puntuación (Scutari, 2011). Estos modelos llevan a cabo dos tareas fundamentales: primero realizan un aprendizaje estructural para identificar la topología de la red y a partir de este estiman los parámetros (aprendizaje paramétrico) representados mediante probabilidades condicionales (Madera, 2006).

El aprendizaje de Redes Bayesianas de grandes dimensiones se ha convertido en una tarea prioritaria en los últimos años, esto se debe a su aplicación en diferentes campos como es el de la Bioinformática.

Estrategia de aprendizaje

Un asunto crítico en un EDA es el uso de una estrategia que aprendan del modelo probabilístico porque usualmente no son eficientes desde el punto de vista evaluativo, lo cual puede afectar el rendimiento del algoritmo, por lo que el aprendizaje de la estructura y los parámetros a partir de poblaciones locales, puede ser unas de las alternativas para dar solución este problema. Este aprendizaje se realiza al generarse varios puntos, con los que se construyen el grafo de dependencia en una Red Bayesiana (aprendizaje del grafo de dependencia de las Redes Bayesianas es

decir de la estructura) y luego se estiman sus probabilidades a partir de ese grafo de dependencia (aprendizaje de los parámetros) (Alba, 2006), (Madera, 2006), (Madera, 2007).

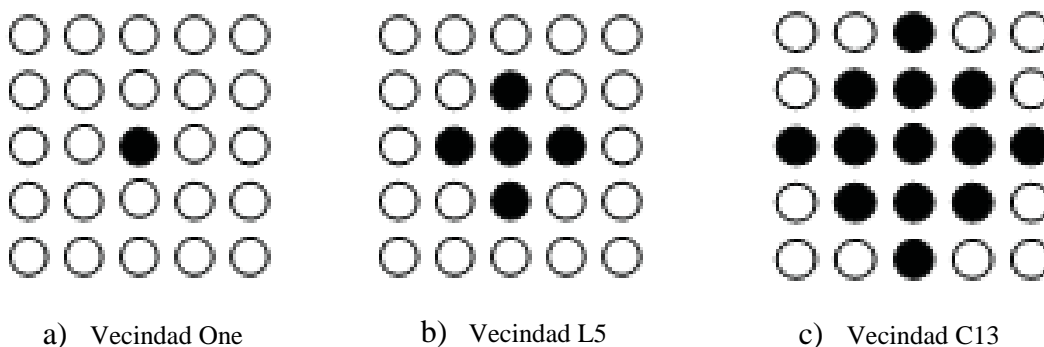
Operadores de aprendizaje

Los operadores de aprendizaje de los EDA se basan en los algoritmos híbridos (Tsamardinos, 2005):

- **Max-Min Hill-Climbing (MMHC):** Algoritmo híbrido que combina los algoritmos Max-Min Parents and Children (restringe el espacio de búsqueda) y Hill-Climbing (para encontrar la estructura de la red óptima en el espacio restringido) (Madera, 2006).
- **Restricted Maximization (RSMAX2):** Implementación más general del algoritmo Max-Min Hill-Climbing que puede usar cualquier combinación de algoritmos basados en restricciones y en optimización de métricas (Madera, 2006).

Vecindades

Un vecindario es un conjunto de individuos vecinos a uno dado, es decir, que están situados próximos a él en la población según una topología espacial dada de la rejilla (Dorronsoro, 2008). El vecindario de 5 individuos, denominado comúnmente NEWS (North, East, West, South), considera el individuo central y los inmediatamente superior, inferior, izquierdo y derecho. Existen otras vecindades, como es el caso de One, C13 o compactos de C25 y C41, el usar un vecindario de menor radio hace que las soluciones se extiendan más lentamente por la población, induciendo una menor presión selectiva global y manteniendo mayor diversidad genética que al usar vecindarios mayores, como se muestra en la figura 3.



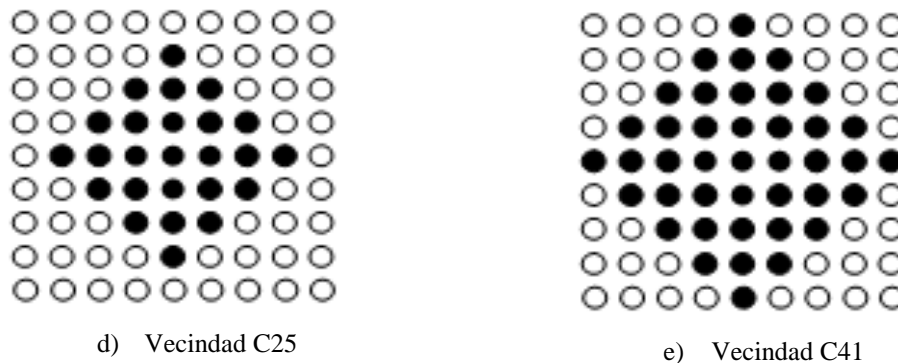


Figura 3. Representación de diferentes vecindades

Problemas de optimización

Los autores (Pelikan, Hauschild et al., 2012) plantean que un problema de optimización puede definirse como un conjunto de soluciones potenciales al problema y un procedimiento para evaluar la calidad de estas soluciones. La tarea es encontrar una solución a partir del conjunto de soluciones potenciales que aumenta al máximo la calidad definida por el procedimiento de evaluación.

Un problema de optimización puede definirse de la siguiente manera:

$$X_{\text{opt}} = \operatorname{argmax}_{x \in D^n} f(x) \quad (2)$$

En la función anterior $x_i = (x_1, x_2, \dots, x_n)$ denota un vector de dominio discreto o continuo de variables aleatorias.

Para el caso discreto, cada x_i toma valores de 1 a r_i , en decir, la variable x_i puede tomar $r_i + 1$ valores.

Para el caso continuo cada x_i toma valores en un intervalo, la variable x_i toma, de forma continua, todos los posibles valores comprendidos en el intervalo.

La solución a este problema consiste en encontrar el punto máximo (óptimo) de la función $f(x) \Rightarrow R$. A partir de este momento las soluciones candidatas serán tratadas como individuos pertenecientes a la población con la cual se trabaja.

Resultados y Discusión

Para el estudio del algoritmo EDA celular se seleccionó la función objetivo discreta OneMax, cuya función tiene $(n + 1)$ valores diferentes de aptitud, los cuales son multinomialmente distribuidos, donde el objetivo es maximizar la

función OneMax. El óptimo global se alcanza en el punto [1.....1] y su valor es n. Además, es una función que se descompone aditivamente donde el valor del óptimo global [1.....1] es igual al número de variables n.

$$Onemax(x) = \sum_{i=1}^n x_i \quad (3)$$

Para comprobar el correcto funcionamiento del algoritmo se realizaron una serie de experimentos consistentes en realizar ejecuciones a las diferentes vecindades, para esto se escogieron un conjunto de parámetros comunes, la población seleccionada siempre será el 30 % de la población global, el método de selección utilizado fue el truncamiento donde se seleccionan los mejores individuos de la población, en todos los experimentos se utilizó el elitismo igual a 1. Para todos los algoritmos se realizaron 100 ejecuciones, el criterio de parada consiste en encontrar el óptimo o realizar un número fijo de iteraciones, en el caso de las funciones discretas es de 30. en la tabla 1 se observa la configuración de los parámetros para cada una de esta función.

Tabla 1 Configuración del EDA celular

<i>Parámetro</i>	<i>Vecindades: One/ L5/ C13/ C25/ C41</i>			
Tamaño del problema	10	30	50	60
Óptimo	10	30	50	60
Tamaño de la estructura(GRID)	(5x5x2x2) = 100 individuos	(5x5x2x2) = 100 individuos	(5x5x2x2) = 100 individuos	(5x5x2x2) = 100 individuos
Umbral de truncamiento	0.3	0.3	0.3	0.3
Elitismo	1	1	1	1

La figura 4 muestra el porcentaje de las ejecuciones que encontraron el óptimo de la función OneMax, por cada una de las vecindades, para 10, 30, 50 y 60 variables.

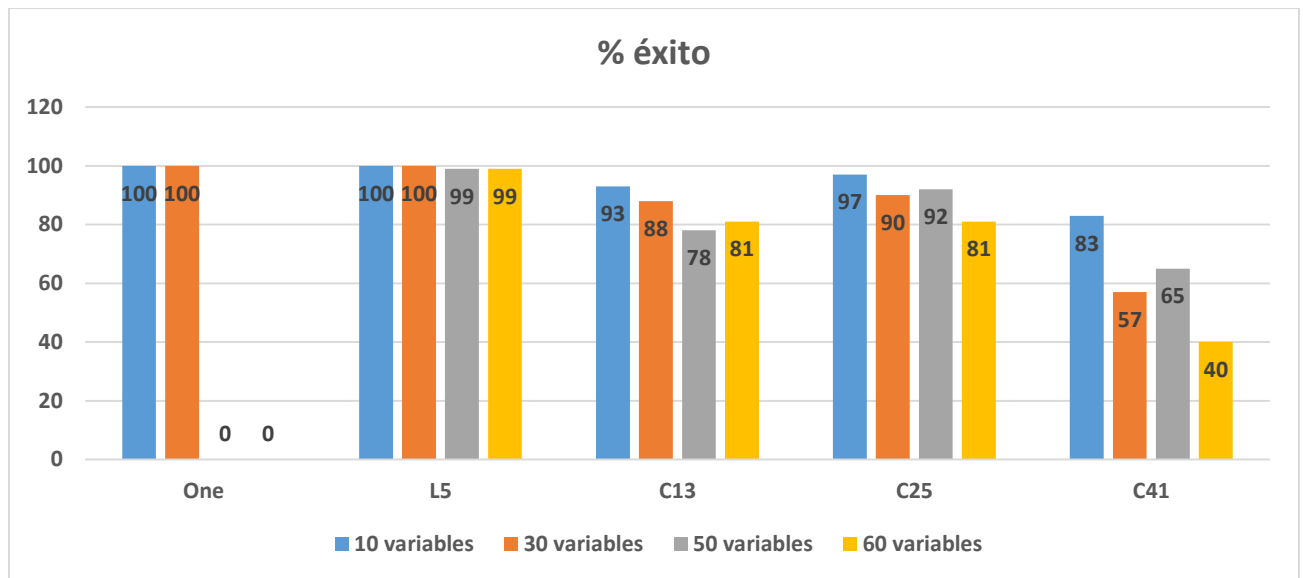


Figura 4. Porcentaje de éxitos en la resolución del problema OneMax.

Como se visualiza en la figura anterior, el algoritmo EDA celular con vecindad L5 tiene mayor eficiencia evaluativa para las diferentes cantidades de variables que con el resto de las vecindades. La de vecindad One es la de peor eficiencia, ya que para 50 y 60 variables el %Éxito su valor es 0, por no encontrar el óptimo en ninguna ejecución. Además, las vecindades C13 y C25 tuvieron una buena eficiencia evaluativa.

En la figura 5 se visualizan los valores Mayor reducción del número de evaluaciones de la función objetivo que en el caso discreto.

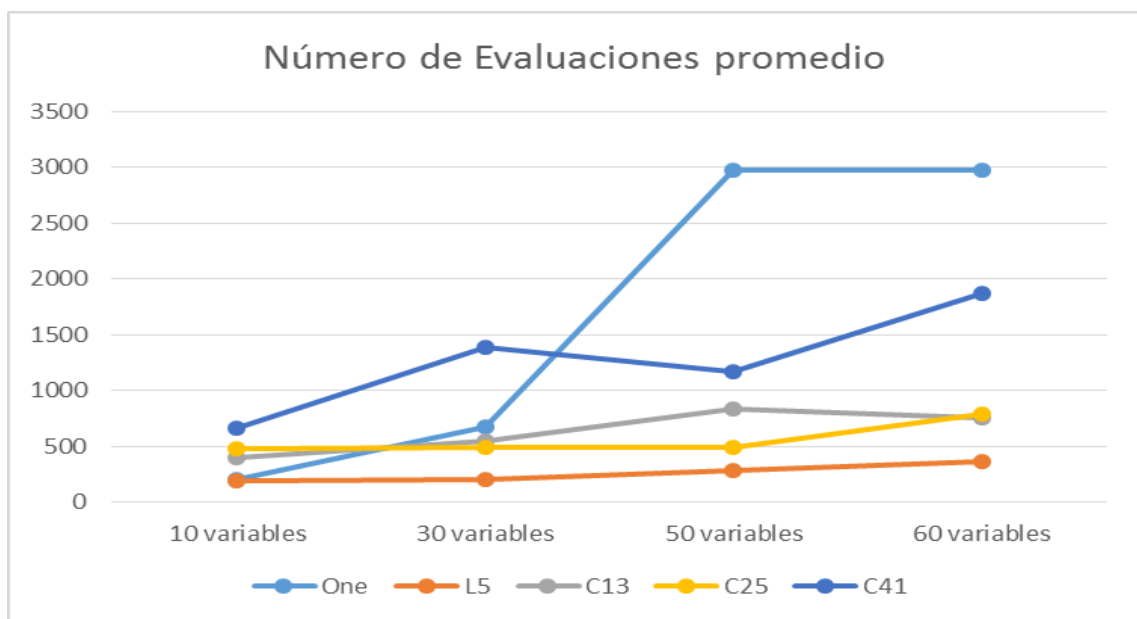


Figura 5. Número promedio de evaluaciones para converger al óptimo en la resolución del problema OneMax.

Como se muestra en la figura anterior, de las vecindades que usa el EDA celular con aprendizaje local de las estructuras y los parámetros, que aprenden las Redes Bayesianas, el de mejor resultado es la L5 que realiza menos de 500 evaluaciones por cada cantidad de variables.

El algoritmo EDA celular con vecindades L5, C25, C13 necesitan menos generaciones para encontrar óptimo de la función OneMax, ver figura 6.

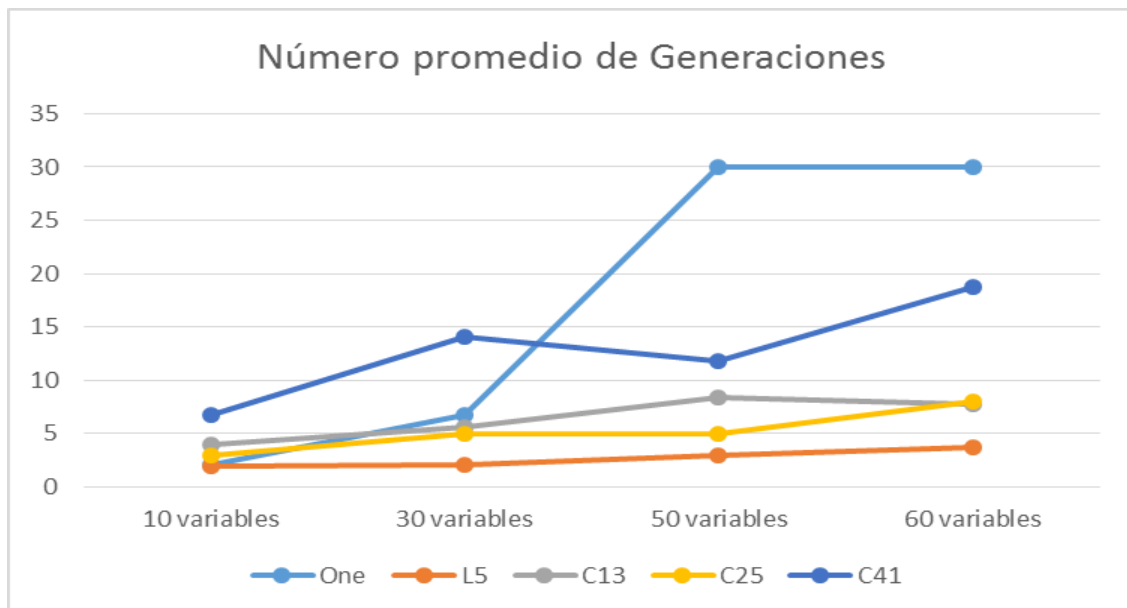


Figura 6. Número promedio de generaciones para converger al óptimo en la resolución del problema OneMax.

Conclusiones

Los EDA descentralizados con aprendizaje basado en pruebas de independencias pueden aumentar la eficiencia evaluativa del EDA por la forma de estructurar la población y la interacción entre los diferentes algoritmos. Estas metaheurísticas pueden ser aplicadas en la resolución de problemas de optimización como: problemas de las rutas en el transporte, problemas en la predicción del arribo de turistas a un destino en específico, problemas de selección de atributos en bases de datos de alta dimensionalidad, configuración de pesos en una red neuronal, entre otros.

Bibliografía

- ECHEGOYEN, C., MENDIBURU, A., SANTANA, R., LOZANO, J. A. (2012). Clases de Equivalencia en Algoritmos de Estimación de Distribuciones. San Sebastián: Universidad del País Vasco.
- LARRAÑAGA, P., PUERTA, J. M. (2003). Algoritmos de Estimación de Distribuciones.

- MARGARITIS, D. (2003). Learning Bayesian Network Model Structure from Data. Carnegie-Mellon University.
- MÜHLENBEIN, H., PAAß, G. (1996). From Recombination of Genes to the Estimation of Distributions. *Computer Science 1411: Parallel Problem Solving from Nature - PPSN IV*, 178–187.
- PELIKAN, M., HAUSCHILD, M. W., LOBO, F. G. (2012). Introduction to Estimation of Distribution Algorithms (No. MEDAL Report No. 2012003). St. Louis: University of Missouri.
- SCUTARI, M. (2011). Bayesian network structure learning, parameter learning and inference.
- TSAMARDINOS, I., ALIFERIS, C., STATNIKOV, A. (2003). "Algorithms for Large Scale Markov Blanket Discovery". Paper presented at the Sixteenth International Florida Artificial Intelligence Research Society Conference.
- YARAMAKALA, S., MARGARITIS, D. (2005). "Speculative Markov Blanket Discovery for Optimal Feature Selection". Paper presented at the Fifth IEEE International Conference on Data Mining.
- E. ALBA, J. MADERA, B. DORRONSORO, A. OCHOA, y M. SOTO (2006). Theory and practice of cellular UMDA for discrete optimization. Springer-Verlag Berlin Heidelberg, pp. 242–251
- B. DORRONSORO, E. ALBA, G. LUQUE y P. BOUVRY(2008). A Self-Adaptive Cellular Memetic Algorithm for the DNA Fragment Assembly Problem, IEEE Congress on Evolutionary Computation (CEC2008)
- E. ALBA and J. TROYA, "Cellular evolutionary algorithms: Evaluating the influence of ratio," in Parallel problem solving from nature PPSN-6, ser. LNCS, M. Schoenauer, Ed., vol. 1917. Springer-Verlag, Heidelberg, 2000, pp. 29–38.
- E. ALBA and B. DORRONSORO, Cellular Genetic Algorithms, ser. Operations Research/Computer Science Interfaces. Springer-Verlag Heidelberg, 2008.
- I. TSAMARDINOS, L. E. BROWN, C. F. ALIFERIS (2005). The Max-Min Hill-Climbing Bayesian Network Structure Learning Algorithm. Kluwer Academic Publishers. Printed in the Netherlands.
- J. MADERA Y A. OCHOA. Un EDA basado en aprendizaje MAX-MIN con escalador de colinas. Informe Técnico 2006-396, Instituto de Cibernética, Matemática y Física (ICIMAF), 2006. (Actualizado en 2008).
- J. MADERA, E. ALBA, G. LUQUE. Performance Evaluation of the Parallel Polytree Approximation Distribution Algorithm on Three Network Technologies. *Revista Iberoamericana de Inteligencia Artificial*, vol. 11 (35), 2007.
- J. MADERA, A. OCHOA. Una versión paralela del algoritmo MMHCEDA. Instituto de Cibernética, Matemática y Física (ICIMAF), 2006. (Actualizado en 2008).

J. MADERA, E. ALBA, A. OCHOA, A parallel island model for estimation of distribution algorithms, en: J.A. Lozano, P. Larrañaga, I. Inza y E. Bengoetxea (Eds.), Towards a new evolutionary computation. Advances on estimation of distribution algorithms, Springer, 2006.

K. LU, D. M. LI and J. YU, "Intelligent test paper construction method based on cellular genetic algorithm", Computer Engineering and Applications, vol. 49, no. 16, (2013), pp. 57-60.

ANKUN HUANG, DONGMEI LI1, JIAJIA HOU and TAO BI (2015). An Adaptive Cellular Genetic Algorithm Based on Selection Strategy for Test Sheet Generation. International Journal of Hybrid Information Technology, Vol.8, No.9, pp.33-42, <http://dx.doi.org/10.14257/ijhit.2015.8.9.04>.

QINGFENG DING, GUOXIN ZHENG (2015). The Cellular Differential Evolution Based on Chaotic Local Search, Mathematical Problems in Engineering, Volume 2015 <http://dx.doi.org/10.1155/2015/128902>.