Revista Cubana de Ciencias Informáticas
Vol. 14, No. 4, Octubre-Diciembre, 2020
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 1-19

Tipo de artículo: Artículo original
Temática: Matemática Computacional
Recibido: 01/09/2020 | Aceptado: 20/11/2020

Algoritmo de estimación de distribución basado en el aprendizaje de redes bayesianas con análisis de dependencias para problemas de optimización en enteros

Estimation of distribution algorithm based on Bayesian networks learning with dependency analysis for integer optimization problems

Julio Madera Quintana [1*] https://orcid.org/0000-0001-5551-690X
Yoan Martínez López [1] https://orcid.org/0000-0002-1950-567X
José Fernández Pardo [1] https://orcid.org/0000-0002-0167-5208

[1] Departamento de Informática, Facultad de Informática y Ciencias Exactas, Universidad de Camagüey. Circunavalación Norte km 5 1/2., Camagüey, Cuba. E-mail: {julio.madera, yoan.martinez, jose.fernenadez}@reduc.edu.cu

*Autor para la correspondencia. (julio.madera@reduc.edu.cu)

RESUMEN

A partir del estudio del algoritmo de estimación de distribuciones (EDA) basado en poliárboles se propone e investiga la clase de algoritmos EDA que utilizan pruebas de independencias en el aprendizaje de la estructura probabilística. Estos algoritmos se conocen como EDA basados en restricciones los que definen una clase de EDA llamada algoritmos de estimación de distribuciones con restricciones (CBEDA). Como resultado se

propone un nuevo algoritmo $\text{CBEDA}_{\text{TPDA}}$ que utiliza el método de detección de dependencias de tres fases para el aprendizaje de redes Bayesianas. Los resultados experimentales demuestran que la nueva propuesta exhibe adecuadas cualidades numéricas para la solución de problemas con codificación entera como son las funciones decepcionantes y el problema de la predicción de estructuras de proteínas (PSP, del inglés, Protein Structure Prediction). Los resultados son comparados con otros algoritmos del estado del arte de la computación evolutiva, incluyendo propuestas del campo de los EDA.

Palabras clave: Algoritmos de estimación de distribuciones; optimización entera; predicción de la estructura de proteínas; pruebas de independencia.

ABSTRACT

From the study of the Estimation of Distribution Algorithms (EDA) based on polytrees, we propose and evaluate the class of EDA algorithms that use independence tests for learning the probabilistic model. These algorithms are known as constraint-based EDA which define a class of EDA called constraint-based estimation of distribution algorithms (CBEDA). As a result, a new $\text{CBEDA}_{\text{TPDA}}$ algorithm is proposed using the three-phase dependence detection method for learning Bayesian networks. The experimental results show that the new proposal has adequate numerical qualities for the solution of optimization problems with integer representation such as the deceptive functions and the problem of protein structure prediction (PSP). The results are compared with other state-of-the-art algorithms in evolutionary computation, including proposals from the EDA field.

Keywords: Estimation of distribution algorithms; integer optimization; protein structure prediction; independence tests.

Revista Cubana de Ciencias Informáticas
Vol. 14, No. 4, Octubre-Diciembre, 2020
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 1-19

Introducción

En los últimos años una nueva familia de algoritmos evolutivos conocidos como Algoritmos de Estimación de Distribuciones (EDA, del inglés, Estimation of Distribution Algorithm) (Mühlenbein, y otros, 1999; Larrañaga y otros, 2002) ha despertado gran interés en la comunidad científica relacionada con la optimización, la computación evolutiva y los modelos probabilísticos. Estos algoritmos surgieron como una alternativa a los algoritmos genéticos. Los EDA están motivados por la necesidad de identificar las interrelaciones entre las variables (Madera, y otros, 2018). Las sucesivas generaciones de individuos se crean a partir de la estimación de la distribución de probabilidad observada en la población actual. Consecuentemente, el principal rasgo que distingue a los EDA de otros algoritmos evolutivos es que aprenden las interacciones entre las variables del problema (Tsagris, 2020; Dai, y otros, 2020).

Los EDA centran su funcionamiento en la construcción y simulación de distribuciones de probabilidad. La complejidad de las distribuciones está asociada con la capacidad del modelo utilizado para expresar las relaciones de dependencias que existen entre las variables del problema. A medida que aumenta la complejidad de la estructura, el EDA necesitará mayor cantidad de individuos para hacer una correcta estimación del modelo aumentando directamente el costo computacional del algoritmo.

Reducir el costo del proceso de búsqueda es una cuestión crítica en EDA. Usualmente este costo se relaciona con el número de evaluaciones de la función objetivo (costo evaluativo) y el tiempo de ejecución del algoritmo (costo en tiempo). Una reducción combinada del costo evaluativo y el costo en tiempo permitirá abordar clases de problemas de complejidad creciente que aparecen tanto en la academia como en el mundo real.

En este artículo se propone la utilización de un EDA para la solución de problemas de optimización discretos y en la predicción de la estructura terciaria de proteínas con un modelo simplificado (Santana, y otros, 2008; Bergasa-Caceres, y otros, 2020; Brower, y otros, 2020; Mazidi, y otros, 2020; Fefelova y otros, 2020; Zarges, 2020). S demuestra la utilidad de los EDA basados en restricciones para resolver estos problemas (Mühlenbein, y otros, 1999a; Mühlenbein, y otros, 2006; Krejca, y otros, 2020), resultando en un algoritmo práctico que permite no acudir a modelos de EDA más complejos como los Markovianos (Tsagris, 2020; Dai, y otros, 2020). Los resultados experimentales muestran que este método es capaz de obtener resultados similares a otros algoritmos del estado del arte a un menor costo.

La estructura del artículo es la siguiente: en la próxima sección se presenta la metodología computacional propuesta. En la misma se propone un nuevo algoritmo EDA basado en restricciones. Posteriormente se presentan los problemas de optimización propuestos para la validación y comparación de los algoritmos. Una vez definida la base para la experimentación se procede al análisis de los resultados y discusión. Por último, se presentan las conclusiones y trabajo futuro de la investigación.

Metodología Computacional

Los EDA fueron introducidos por primera vez en el campo de la computación evolutiva en (Mühlenbein, y otros, 1999; Larrañaga, y otros, 2002), concebidos para sustituir los operadores de cruzamiento y mutación del algoritmo genético (GA, del inglés, Genetic Algorithm) recientes trabajos en esta área han demostrado la utilidad del operador de mutación en este tipo de algoritmo, interpretado como un operador de variación probabilística (Ochoa, y otros, 2006). De forma general los EDA utilizan una población seleccionada para estimar la distribución de probabilidad y a partir de esta se generan los puntos que formarán la nueva población. Así, las relaciones de dependencia entre las variables quedan especificadas de forma explícita a través del modelo probabilístico del conjunto seleccionado.

El algoritmo EDA es teórico, producto a que el cálculo de todos los parámetros necesarios para especificar la distribución de probabilidad es intratable. Producto a esta dificultad es que surge el algoritmo con distribución factorizada (FDA, del inglés, Factorized Distribution Algorithm) (Mühlenbein, y otros, 1999; Mühlenbein, y otros, 2006; Krejca, y otros, 2020) con el objetivo de hacer de los EDA algoritmos tratables. En la literatura especializada varios autores se refieren a EDA como la clase de los algoritmos evolutivos basados en estimación y simulación de distribuciones.

En este artículo se propone la utilización del algoritmo $CBEDA_{TPDA}$ (del inglés, Constraint Based EDA) (Madera, 2009; Madera, y Otros, 2018) basado en el algoritmo de análisis de dependencias de tres fases (TPDA, del inglés, Three - Phase Dependency Analysis algorithm) (Madera, 2009). El TPDA utiliza un

esquema híbrido para la construcción de la red Bayesiana (distribución de búsqueda del EDA). En una primera etapa construye un esqueleto a través de pruebas de independencias, luego el esqueleto es orientado utilizando un procedimiento de optimización de métricas. La utilización de este algoritmo se justifica por dos razones fundamentales: (1) por sus propiedades numéricas, dónde supera a otros algoritmos del estado del arte y (2) por la posibilidad de encontrar dependencias entre las variables, algo común en la predicción de la estructura terciaria de proteínas.

Algoritmo de estimación de distribuciones basado en restricciones

Los EDA han sido motivados por la necesidad de identificar las dependencias entre las variables, una cuestión clave para resolver problemas complejos. Por lo tanto, la principal característica que distingue a los EDA de otros algoritmos evolutivos (EA, del inglés, Evolutionary Algorithms) es que aprenden las interacciones entre las variables del problema. Los EDA se clasifican según el modelo de aprendizaje utilizado y van desde los que asumen total independencia de las variables, hasta los que toman en cuenta interacciones por pares; algunas familias de EDA también consideran modelos generales no restringidos de interacción entre variables. El presente enfoque aprovecha la detección de independencias para construir la red bayesiana (Madera, y otros, 2018; Brownlee, y otros, 2015). Formalmente, se puede tener una red bayesiana sobre el conjunto de variables aleatorias $V = \{V1, \ldots, Vn\}$.

La factorización de la distribución de probabilidad conjunta puede expresarse como:

$$\mathrm{P}(\mathrm{V}) = \mathrm{P}(\mathrm{V}_1, \mathrm{V}_2, \ldots\ldots, \mathrm{V}_n) = \prod_{i=1}^{n} \mathrm{P}(\mathrm{V}_i \,|\mathrm{Pa}_i) \qquad (1)$$

donde, Pa_i es el conjunto de padres de V_i (es decir, existe un arco de cada nodo en Pa_i a V_i).

Existen dos métodos diferentes para aprender la estructura probabilística de un conjunto de datos. El primer método concibe la utilización de un procedimiento de puntuación + búsqueda (Madera, 2009; Madera, y otros, 2018). Estos métodos definen una métrica (función de costo) que mide la idoneidad de cada red bayesiana

candidata para una base de datos. El otro enfoque se sustenta sobre pruebas de independencia para obtener una lista de relaciones en forma de aristas. Estos algoritmos construyen la estructura gráfica respetando, en la medida de lo posible, las relaciones presentes de la lista (Brownlee, y otros, 2015; Madera y Ochoa, 2018).

El algoritmo propuesto en este artículo utiliza un enfoque híbrido. En una primera fase, se construye un esqueleto a partir de pruebas de independencia, posteriormente se orientan las aristas con la aplicación de un procedimiento de puntuación + búsqueda (Madera, 2009; Madera, y otros, 2018).

El algoritmo CBEDA$_{TPDA}$

El procedimiento del CBEDATPDA (ver Algoritmo 1) utiliza el algoritmo TPDA para el aprendizaje de redes Bayesianas (Cheng, y otros, 2002). Este algoritmo permite extraer las dependencias entre las variables de los individuos que forman el conjunto seleccionado. La propuesta original se modificó para permitir su aplicación en el contexto de los EDA. Se propone un nuevo método para orientar las aristas basado en un procedimiento de puntuación + búsqueda. A los lectores interesados se les recomienda revisar el artículo original de Cheng y las referencias citadas para profundizar en los detalles del algoritmo TPDA.

Algoritmo 1 – Propuesta del algoritmo CBEDA$_{TPDA}$.

1. Poner $t \leftarrow 1$
2. Generar $N \gg 0$ individuos aleatoriamente
3. **Mientras** el criterio de parada no se alcance **hacer**
    a. Seleccionar M individuos de acuerdo a un método de selección
    b. Estimar la distribución de probabilidad $p^S(x,t)$ del conjunto seleccionado utilizando el algoritmo $TDPA$
    c. Generar N nuevos puntos de acuerdo a la distribución $p^S(x,t)$
    d. Poner $t \leftarrow t+1$
4. **Fin mientras**

Revista Cubana de Ciencias Informáticas
Vol. 14, No. 4, Octubre-Diciembre, 2020
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 1-19

En este artículo se toman en cuenta cuatro variantes del algoritmo $CBEDA_{TPDA}$:

1. Algoritmo $CBEDA_{TPDA}$: utiliza la implementación original del algoritmo $TDPA$ orientando el esqueleto mediante un procedimiento de puntuación + búsqueda.
2. $CBEDA_{TPDA\text{-}RO}$: Construye el esqueleto de la red Bayesiana de la misma forma que el algoritmo TPDA original pero las aristas son orientas de forma aleatoria.
3. $CBEDA_{TPDA\text{-}PT}$: Restringe la estructura de la red Bayesiana aprendida a un poliárbol (existe un solo camino entre cualquier par de nodos).
4. $CBEDA_{TPDA\text{-}k2}$: Restringe el número de nodos adyacentes de cualquier nodo a dos.

Problemas

Para retar a los CBEDA propuestos se utilizan las funciones decepcionantes que se descomponen aditivamente con representación binaria y no-binaria (Goldberg, y otros, 1992; Santana, y otros, 2002). También se estudia el comportamiento de los algoritmos en un problema de la Bioinformática, la predicción de estructuras de proteínas en modelos simplificados (Bergasa-Caceres, y otros, 2020; Brower, y otros, 2020; Mazidi, y otros, 2020; Fefelova, y otros, y otros, 2020; Zarges, 2020; Santana, y otros, 2008).

Función $F_{deceptive3}$

La función $F_{deceptive3}$ (Goldberg, y otros, 1992; Santana, y otros, 2002) se define de la siguiente manera:

$$F_{deceptive3}(\mathbf{x}) = \sum_{i=1}^{\frac{n}{k}} f_{dec}(x_{3\cdot i-2}, x_{3\cdot i-1}, x_{3\cdot i}) \quad (2)$$

Donde,

$$f_{dec}^{3}(u) = \begin{cases} 0.9 & for\ u = 0 \\ 0.8 & for\ u = 1 \\ 0.0 & for\ u = 2 \\ 1.0 & for\ u = 3 \end{cases} \quad (3)$$

Revista Cubana de Ciencias Informáticas
Vol. 14, No. 4, Octubre-Diciembre, 2020
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 1-19

El objetivo es maximizar la función $F_{deceptive3}$ y el óptimo global se encuentra en el punto (1,1, … ,1).

Función decepcionante con representación entera

El análisis del comportamiento de los EDA se ha centrado en el estudio de problemas binarios. Sin embargo, para estudiar la robustez de los EDA y su posible aplicación a problemas reales se hace necesario el estudio de problemas no-binarios (representación entera). Con este fin, se utilizará una función decepcionante definida para problemas con representación entera (Santana, y otros, 2002).

La función general para $F^c_{deceptivek}(\mathbf{x})$ de orden k está formada por una función aditiva compuesta por la función $f_{dec}(x_1, x_2, \cdots, x_k, k, c)$ evaluada en subcadenas de tamaño k y cardinalidad c, es decir, $x_i \in \{0, 1, \cdots c-1\}$. Esta función es una generalización de la función $F_{deceptive3}$ para variables con representación entera.

La función $F^c_{deceptivek}(\mathbf{x})$ queda definida de la siguiente manera:

$$F^c_{deceptivek}(\mathbf{x}) = \sum_{i=1}^{\frac{n}{k}} f_{dec}\left(x_{k\cdot(i-1)+1}, x_{k\cdot(i-1)+2}, \cdots, x_{k\cdot i+1}, k, c\right) \quad (4)$$

Donde, $f_{dec}(x_1, x_2, \cdots, x_k, k, c)$ se define como:

$$f_{dec}(x_1, x_2, \cdots, x_k, k, c) = \begin{cases} k\cdot(c-1), & para\ \sum_{i=1}^{k} x_i = k\cdot(c-1) \\ k\cdot(c-1) - \sum_{i=1}^{k} x_i - 1, & en\ caso\ contrario \end{cases} \quad (5)$$

El problema de la predicción de la estructura de proteínas

Las proteínas son cadenas o secuencias formadas por la combinación de 20 aminoácidos enlazados mediante enlaces peptídicos. En el espacio, dicha cadena adopta una estructura tridimensional, la que determina la funcionalidad biológica de la proteína. Esta estructura tiene cavidades y salientes que permiten el

acoplamiento con otras proteínas para formar estructuras más complejas, o para bloquear el funcionamiento de otras.

Estructuralmente, las proteínas pueden analizarse a diferentes escalas. Se denomina estructura primaria de una proteína, a la secuencia de aminoácidos que la componen. Estos aminoácidos se agrupan en estructuras llamadas α□hélices, laminas β□y loops, las cuales reciben el nombre de estructuras secundarias. Estas subestructuras se pliegan o "doblan" en el espacio tridimensional hasta alcanzar una configuración que se conoce como estructura terciaria o estado nativo. Es esta estructura tridimensional la que determina la funcionalidad biológica de la proteína (Branden, y otros, 1998).

Resulta aceptado que uno de los elementos que más influye en la determinación de esta estructura, es el llamado efecto hidrofóbico. Los aminoácidos pueden clasificarse en hidrofílicos o hidrofóbicos según sea su comportamiento en un medio acuoso. En el proceso de plegado los aminoácidos hidrofóbicos tienden a agruparse en el centro de la molécula formando una especie de coraza interna, mientras que los hidrofílicos tienden a quedar expuestos al solvente. Se puede definir el problema de la predicción de la estructura de una proteína como la búsqueda, a partir de la secuencia de aminoácidos, de la estructura tridimensional correspondiente.

Un modelo para PSP es relevante si refleja alguna de las propiedades del proceso de formación de estructura en el sistema real. De acuerdo con la hipótesis termodinámica, el estado nativo de una proteína se corresponde con el estado de mínima energía libre y por eso los modelos basados en energía especifican una función de "costo" que asigna un valor de energía libre a cada estructura válida. Se asume que la estructura terciaria de la proteína se corresponderá entonces, con aquella conformación que minimice la función de energía.

Dentro de este tipo de modelos basados en energía se destacan los modelos basados en retículos. Donde, cada vértice del retículo es ocupado por un aminoácido de la cadena y aminoácidos consecutivos en la secuencia se ubican en posiciones adyacentes del retículo.

Revista Cubana de Ciencias Informáticas
Vol. 14, No. 4, Octubre-Diciembre, 2020
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 1-19

Los modelos en retículos utilizan las coordenadas internas para modelar las estructuras terciarias de las proteínas, dada la posición del aminoácido i, existen δ valores para representar la posición del $i+1$ dependiendo del retículo utilizado (bidimensional, tridimensional o de diamante).

El modelo más simple de PSP en retículos, llamado modelo de Dill (Dill, 1985) solo considera dos tipos de aminoácidos, donde cada tipo representa si es hidrofóbico (representado con una H) o hidrofílico (representado con una P). Una proteína entonces, se modela como secuencia $s \in \{H, P\}^+$. En la figura 1 se muestra un ejemplo de configuración de una proteína en el modelo HP.

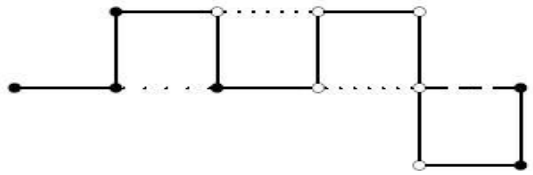

Fig. 1 - Posible configuración para la secuencia $HHHPHPPPPPH$ para el modelo HP en un retículo bidimensional. La configuración muestra un contacto HH, uno HP y dos PP.

La función de energía utilizada, solo tiene en cuenta las interacciones entre aminoácidos que sean adyacentes en el retículo, pero no consecutivos en la secuencia (vecinos topológicos). Cada interacción de este tipo se denomina bond o contacto. Dada una secuencia con n aminoácidos, $S = (s_1, s_2, \cdots, s_n)$, con $s_i \in \{H, P\}$, un plegado para S, $fold(S) = X = (x_1, x_2, \cdots x_n)$ dispuesto en un retículo L, y una matriz de interacción $\sum (s_i, s_j)$, una función de energía posible es:

$$E(S, X) = \sum_i^n \sum_{j>i+1}^n \varepsilon_{i,j} * \Delta(x_i, x_j) \quad (6)$$

Revista Cubana de Ciencias Informáticas
Vol. 14, No. 4, Octubre-Diciembre, 2020
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 1-19

Donde $\Delta(x_i, x_j) = 1$ si x_i y x_j son adyacentes en el retículo y no consecutivos en la cadena y 0 en caso contrario. El término $\varepsilon_{i,j} = \varepsilon(s_i, s_j)$ es el valor de la fila i, columna j en la matriz de interacción ε. En la Figura 2 se muestran dos matrices de interacción posibles.

	H	P
H	-1	0
P	0	0

	H	P
H	-3	-1
P	-1	0

Fig. 2 - Dos posibles matrices de interacción $\varepsilon_{i,j}$.

Con estos elementos se establece que resolver PSP en este modelo es equivalente a minimizar la función de energía $E(S, X)$ (ecuación 6).

Las estructuras en estos modelos se representan a través del sistema de coordenadas internas, de las cuales aparecen dos variaciones: la codificación mediante coordenadas absolutas y coordenadas relativas, en este artículo de utiliza la última codificación.

Resultados y discusión

En esta sección se analiza el comportamiento del algoritmo cuando se varía la cardinalidad de las variables (conjunto de posibles valores que puede tomar cada variable). Para tal propósito se utiliza la función F deceptive3 c, donde c ∈ {2,3,4,5,6,7}. La metodología utilizada fue encontrar el tamaño de población crítica M para el algoritmo $\text{CBEDA}_{\text{TPDA}}$ y posteriormente ejecutar el resto de los algoritmos ($\text{CBEDA}_{\text{TPDA-RO}}$, $\text{CBEDA}_{\text{TPDA-PT}}$, $\text{CBEDA}_{\text{TPDA-k2}}$) con el tamaño de población encontrado. Para obtener el tamaño de población crítica se inicia con una población de 1000 individividuos y si el porciento de éxito es inferior al 95% se incrementa el tamaño de población en 1000 individuos y así hasta lograr el porciento de convergencia deseado. La excepción es la función Fdeceptive3 que por ser la de menos complejidad determinamos su población crítica a partir de un incremento de 100 individuos en cada fallo.

Revista Cubana de Ciencias Informáticas
Vol. 14, No. 4, Octubre-Diciembre, 2020
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 1-19

Para cada función y algoritmo se ejecutaron 50 corridas independientes.

El análisis de la tabla 1 muestra claramente una superioridad del algoritmo $CBEDA_{TPDA}$ en cuanto al porciento de éxitos, evidenciándose fluctuaciones en el resto de las propuestas. Esto es un resultado esperado pues los algoritmos $CBEDA_{TPDA\text{-}RO}$, $CBEDA_{TPDA\text{-}PT}$, $CBEDA_{TPDA\text{-}k2}$ son variantes restringidas del original $CBEDA_{TPDA}$.

Tabla 1 - Resultados para la función decepcionante discreta.

F	**M**	**$CBEDA_{TPDA}$**		**$CBEDA_{TPDA\text{-}RO}$**		**$CBEDA_{TPDA\text{-}PT}$**		**$CBEDA_{TPDA\text{-}k2}$**	
		succ	**evals**	**succ**	**evals**	**succ**	**evals**	**succ**	**evals**
$F_{deceptive3}$	1200	**98**	6372	96	**6346**	80	7952	88	6349
$F^3_{deceptive3}$	3000	**98**	16665	**98**	**15728**	**98**	17767	88	17052
$F^4_{deceptive3}$	9000	98	48508	**100**	**46584**	98	58757	96	48487
$F^5_{deceptive3}$	18000	**96**	**116888**	72	12555	88	146495	86	114447
$F^6_{deceptive3}$	40000	**96**	**270400**	70	313600	76	382947	88	263264
$F^7_{deceptive3}$	100000	96	**642250**	96	696250	98	1014690	**100**	663400

Otro resultado interesante es el comportamiento en cuanto al número de evaluaciones del CBEDATPDA-RO en los problemas con cardinalidad 2, 3 y 4, superado por el CBEDATPDA para las cardinalidades 5, 6, y 7. El peor rendimiento lo exhibe el CBEDATPDA-PT, cuya estructura no permite obtener mejores resultados en cuanto al número de evaluaciones. Este algoritmo tendrá mejor o igual comportamiento que el CBEDATPDA (debe ser capaz de aprender la estructura correcta) si las funciones estudiadas tuvieran una estructura probabilística en forma de poliárbol (Soto, 2003; Ochoa, y otros, 2006). Por último, queda demostrado que el aumento de la cardinalidad de las variables del problema es fuente de complejidad adicional como lo pueden ser el aumento de la cantidad de variables, si son decepcionantes o no, así como tener entropía univariada (o de orden superior) alta (Ochoa y Soto, 2006). El aumento de la complejidad queda reflejado en el aumento del tamaño de la población crítica a medida que aumenta la cardinalidad, así como el aumento en el número de evaluaciones realizada por el algoritmo.

Revista Cubana de Ciencias Informáticas
Vol. 14, No. 4, Octubre-Diciembre, 2020
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 1-19

Escalabilidad de los CBEDA para $F_{deceptive3}$

En esta sección se hace un estudio de escalabilidad numérica y física (tiempo de ejecución) de los diferentes CBEDA para la función Fdeceptive3. La tabla muestra los resultados de escalabilidad, a partir del incremento en el número de variables desde 15 hasta 90. Para todas las configuraciones, excepto la de 15 variables, el $CBEDA_{TPDA\text{-}RO}$ obtiene el mejor rendimiento numérico (menor número de evaluaciones) y en tres de las configuraciones también obtiene los menores tiempos de ejecución. Con respecto al tiempo de ejecución, el $CBEDA_{TPDA\text{-}RO}$ no realiza optimización de métrica por lo que se ahorra ese tiempo que es obligatorio para las demás variantes y que evidentemente es mayor que la generación de una permutación aleatoria de las variables.

Tabla 2 - Resultados para la escalabilidad deceptive3 discreta.

vars	$CBEDA_{TPDA}$		$CBEDA_{TPDA\text{-}RO}$		$CBEDA_{TPDA\text{-}PT}$		$CBEDA_{TPDA\text{-}k2}$	
	evals	**tiempo**	**evals**	**tiempo**	**evals**	**tiempo**	**evals**	**tiempo**
15	**1801.22**	0.0288	1848.75	**0.0284**	2028.75	0.0353	1823.27	0.0302
30	5564.08	1.8426	**5542.5**	**1.7616**	6897	2.5785	5544.55	1.9312
45	19313	6.2820	**18725**	**3.0845**	27493.6	9.4272	19181.5	6.8397
60	31625	22.6664	**31280**	21.169	48020	29.8952	31632.7	**18.0878**
75	49685.7	**72.8218**	**49336**	74.4192	78614.3	128.768	49787.5	75.4094
90	69502	**153.315**	**69102**	157.117	110898	268.402	69171.4	158.03

Resultados para el problema de la predicción de estructuras de proteínas

En esta sección se muestran los resultados de la aplicación del $CBEDA_{TPDA}$ en la solución del problema de la predicción de estructuras de proteínas y su comparación con otros algoritmos del estado del arte de la computación evolutiva.

La tabla 3 muestras las instancias del modelo HP utilizadas en los experimentos, $H(\mathbf{x}^*)$ representa la mejor solución conocida para la secuencia. PSP se ha estudiado, utilizando las secuencias descritas.

Revista Cubana de Ciencias Informáticas
Vol. 14, No. 4, Octubre-Diciembre, 2020
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 1-19

Tabla 3 - Instancias del modelo HP utilizadas en los experimentos.

instancia	tamaño	$H(x^*)$	secuencia
s_1	20	-9	*HPHPPHHPHHPHPHHPPHPH*
s_2	24	-9	*HHPPHPPHPPHPPHPPHPPHPPHH*
s_3	25	-8	*PPHPPHHP^{4}HHP^{4}HHP^{4}HH*
s_4	36	-14	P^{3}HHPPHHP^{5}H^{7}PPHHP^{4}HHPPHPP
s_5	48	-23	*PPHPPHHPPHHP^{5}H^{10}P^{6}HHPPHHPPHPPH^5*
s_6	50	-21	*HHPHPHPHPH^{4}PHP^{3}HP^{3}HP^{4}HP^{3}HP^{3}HPH^4{PH}^{4}H*
s_7	60	-36	*PPH^{3}PH^{8}P^{3}H^{10}PHP^{3}H^{12}P^{4}H^{6}PHHPHP*
s_8	64	-42	*H^{12}PHPH{PPHH}^{2}PPH{PPHH}^{2}PPH{PPHH}^{2}PPHPHPH^{12}*

El primer experimento consiste en encontrar el óptimo para las secuencias de la tabla 3 con la aplicación de tres variantes del CBEDA$_{TPDA}$. La primera es el algoritmo CBEDA$_{TPDA}$ clásico, la segunda realiza una orientación aleatoria del esqueleto (CBEDA$_{TPDA-RO}$) y la tercera restringe el número de nodos adyacentes a dos (CBEDA$_{TPDA-k2}$).

La metodología empleada es similar a la utilizada en el estudio de otros EDA (Cotta, 2003). Todos los algoritmos tienen un tamaño de población de 5000 individuos, ejecutándose un máximo de 5000 generaciones. El truncamiento es de 0.1 con la aplicación del mejor elitismo (toda la población seleccionada pasa directamente a la próxima generación). La tabla 4 muestra los resultados obtenidos en los experimentos, donde la columna éxitos es el porciento de veces que se encontró la mejor solución en 50 corridas independientes y gen representa la generación media donde se encontró la mejor solución. Los mejores resultados de cada parámetro medido son destacados en negritas.

Los resultados muestran como todas las variantes del $CBEDA_{TPDA}$ encuentran la mejor solución para las secuencias $s_1 - s_6$ y s_8 excepto el $CBEDA_{TPDA-RO}$ en la secuencia s_5 que obtiene una solución sub-óptima. En el caso de la secuencia s_7 todos los algoritmos obtienen una solución sub-óptima, en (Cotta, 2003) se muestra empíricamente como esta instancia es decepcionante lo que aumenta su complejidad. Al igual que en (Cotta, 2003) los resultados para los CBEDA son promisorios debido a que no se utilizan técnicas de optimización local ni los parámetros se han refinado para cada instancia.

Revista Cubana de Ciencias Informáticas
Vol. 14, No. 4, Octubre-Diciembre, 2020
ISSN: 2227-1899 | RNPS: 2301
http://rcci.uci.cu
Pág. 1-19

Tabla 4 - Resultados de los algoritmos $CBEDA_{TPDA}$, $CBEDA_{TPDA-RO}$, $CBEDA_{TPDA-k2}$ en la solución de PSP.

F		CBEDA$_{TPDA}$			CBEDA$_{TPDA\text{-}RO}$			CBEDA$_{TPDA\text{-}k2}$		
	$H(x^*)$	$H(x)$	éxitos	gen	$H(x)$	éxitos	gen	$H(x)$	éxitos	gen
s_1	-9	**-9**	100	**3.32**	**-9**	**100**	**3.32**	**-9**	**100**	3.34
s_2	-9	**-9**	100	**3.66**	**-9**	**100**	3.98	**-9**	**100**	3.74
s_3	-8	**-8**	100	**4.52**	**-8**	**100**	4.54	**-8**	96	5.81
s_4	-14	**-14**	4	12.5	**-14**	4	12.5	**-14**	**10**	**12.4**
s_5	-23	**-23**	6	**21.0**	-22	**8**	53.0	**-23**	4	27.5
s_6	-21	**-21**	88	13.86	**-21**	62	16.22	**-21**	76	**13.76**
s_7	-36	-35	16	**54.37**	-35	**20**	79.7	-35	4	144.5
s_8	-42	**-42**	10	**28.0**	**-42**	**24**	50.58	**-42**	4	94.0

El siguiente experimento tiene como motivación la siguiente pregunta: ¿Es necesario acudir a modelos de aprendizaje más complejos como las redes de Markov para la solución de PSP tridimensional con EDA? Para dar respuesta a la pregunta seguimos un experimento similar al utilizado por Santana y otros (Santana, y otros, 2008), configurando el $CBEDA_{TPDA}$ con un tamaño de población de 5000 individuos y un máximo de 1000 generaciones. El truncamiento es de 0.1 con la estrategia del mejor elitismo al igual que en los experimentos para retículos en dos dimensiones. La comparación se realiza entre el GA híbrido (Cutello, y otros, 2007), el $MK - EDA_2$ (el subíndice dos significa que cada nodo puede tener a lo sumo dos nodos adyacentes) y el $CBEDA_{TPDA}$.

Tabla 5 - Resultados del *GA híbrido*, $MK - EDA_2$, $CBEDA_{TPDA}$ en retículos tridimensionales.

S	*GA híbrido*		$MK - EDA_2$		$CBEDA_{TPDA}$	
	$H(x)$	$media \pm \delta$	$H(x)$	$media \pm \delta$	$H(x)$	$media \pm \delta$
s_1	-11	-10.52 ± 0.54	-11	-10.82 ± 0.38	**-11**	**-11.0 ± 0.0**
s_2	-13	-11.28 ± 0.90	-13	-12.02 ± 0.94	**-13**	**-13.0 ± 0.0**
s_3	-9	-8.54 ± 0.64	-9	-8.96 ± 0.19	**-9**	**-9.0 ± 0.0**
s_4	-18	-15.76 ± 1.05	-18	-16.40 ± 0.80	**-18**	**-17.96 ± 0.06**
s_5	-28	-24.60 ± 1.57	**-29**	**-27.24 ± 0.92**	-29	-25.36 ± 0.37
s_6	-26	-23.02 ± 1.48	-29	-25.70 ± 1.26	**-31**	**-28.72 ± 0.19**
s_7	-49	-41.18 ± 2.75	**-49**	**-46.30 ± 2.04**	-49	-41.86 ± 0.36
s_8	-46	-40.40 ± 2.50	**-52**	**-46.78 ± 2.28**	-50	-41.2 ± 0.64

Los resultados se muestran en la tabla 5. Se puede apreciar que el $CBEDA_{TPDA}$ supera al GA híbrido y al $MK-EDA_2$ en el promedio del óptimo obtenido para las instancias de la s_1-s_4 y para la instancia s_6. En el caso de las instancias s_5 y s_7, ambos EDA obtienen el mismo valor, aunque el $MK-EDA_2$ supera en el promedio del óptimo al $CBEDA_{TPDA}$. Para la instancia s_8 el $MK-EDA_2$ mejora al GA híbrido y al $CBEDA_{TPDA}$, tanto en el óptimo encontrado como en el promedio. Una observación importante es que los EDA superaron siempre los resultados obtenidos por el GA híbrido, aun cuando este último utiliza técnicas de optimización local para la resolución del problema PSP.

Cómo respuesta a la pregunta planteada podemos decir que no es necesario acudir a modelos complejos de EDA para la solución de PSP en las secuencias estudiadas. Los modelos Bayesianos obtienen resultados similares a los modelos Markovianos con la ventaja de no estar restringida la estructura (en el $MK-EDA_2$ cada nodo puede tener a lo sumo dos nodos adyacentes). En una futura investigación se puede comprobar si la inclusión de un optimizador local mejora las soluciones obtenidas con el $CBEDA_{TPDA}$.

Conclusiones

En este artículo se propuso la utilización de algoritmos EDA para resolver problemas de dominio entero, específicamente la función $F_{deceptive}$ y el problema de la predicción de la estructura terciaria de proteínas. Se utilizó el algoritmo $CBEDA_{TPDA}$ por su posibilidad de detectar dependencias entre las variables durante la optimización. Los resultados experimentales demostraron, que las soluciones encontradas son comparables con los algoritmos del estado del arte, incluso para algunas instancias los superan. En el caso del problema PSP dos dimensiones el algoritmo $CBEDA_{TPDA}$, sin modificaciones, encuentra mejores soluciones que otras variantes del mismo. Para tres dimensiones en PSP se demuestra que acudir a modelos más complejos ($MK-EDA_2$) de EDA resulta innecesario.

Como trabajo futuro se recomienda incorporar al $CBEDA_{TPDA}$ un optimizador local para mejorar el rendimiento del mismo, así como un reparador de secuencias con cualidades superiores. Una mejora posible al algoritmo propuesta es estructurar la población espacialmente, utilizando algoritmos celulares y distribuidos. Se propone extender los experimentos a secuencias más complejas, así como su comparación con otros algoritmos fuera del campo de los EDA y los algoritmos genéticos.

Referencias

Branden, c., Tooze, j. Introduction to Protein Structure. Garland, New York, 1998.

Bergasa-Caceres, F., Rabitz, H. A. Interdiction of Protein Folding for Therapeutic Drug Development in sars cov-2. *The journal of physical chemistry b*, *124*(38), 2020, p. 8201-8208.

Brower, r. C., delisi, c. Impact of massively parallel computation on protein structure determination. *High-performance computing in biomedical research*, 2020, p. 447-463.

Brownlee, a. E., wright, j. A. Constrained, mixed-integer and multi-objective optimization of building designs by nsga-ii with fitness approximation. *Applied soft computing*, *33*, 2015, p. 114-126.

Cheng, j., greiner, r., kelly, j., bell, d., liu, w. Learning bayesian networks from data: an information-theory based approach. *Artificial intelligence*, 137(1-2), 2002, p. 43-90.

Cotta, c. Protein structure prediction using evolutionary algorithms hybridized with backtracking. En j. Mira y j. R. Alvarez, editores, *artificial neural nets problem solving methods*, tomo 2687 de *lecture notes in computer science*, 2003, p. 321–328. Berlin, germany: springer verlag,

Cutello, v., nicosia, g., pavone, m., y timmis, j. An immune algorithm for protein structure prediction on lattice models. *Ieee trans. Evol. Comput.*, tomo 11(1), 2007, p. 101–117.

Dai, j., ren, j., du, w. Decomposition-based bayesian network structure learning algorithm using local topology information. *Knowledge-based systems*, 105602, 2020.

Dill, k. A. Dominant forces in protein folding. Biochemistry, 24(1501), 1985.

Fefelova, i., fefelov, a., voronenko, m., kornelyuk, a., sachenko, a., ryzhkov, e., lytvynenko, v. Predicting the protein tertiary structure by hybrid clonal selection algorithms on 3d Square Lattice. In *2020 IEEE 15th*

International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET), 2020, p. 965-968, IEEE.

Goldberg, d. E., deb, k., clark, j. H. Genetic algorithms, noise, and the sizing of populations. *Complex systems-champaign*, *6*, 1992.

Krejca, m. S., y witt, c. Theory of estimation-of-distribution algorithms. In *theory of evolutionary computation* (p. 405-442). Springer, cham, 2020.

Larrañaga, p., lozano, j. A. *Estimation of distribution algorithms. A new tool for evolutionary computation*. Kluwer academic publishers, 2002.

Madera, j., ochoa, a. Evaluating the max-min hill-climbing estimation of distribution algorithm on b-functions. In *international workshop on artificial intelligence and pattern recognition* (p. 26-33). 2018, springer, cham.

Madera, j. Hacia una generación más eficiente de algoritmos evolutivos con estimación de distribuciones: pruebas de independencia + paralelismo, tesis doctoral, 2009, universidad de la habana.

Mazidi, a., roshanfar, f. Pspga: a new method for protein structure prediction based on genetic algorithm. *Journal of applied dynamic systems and control*, *3*(1), 2020, p. 9-16.

Mühlenbein, h., mahnig, t., ochoa, a. Schemata, distributions and graphical models in evolutionary optimization. *Journal of heuristics*, 5, 1999, p. 215-247.

Mühlenbein, h., mahnig, t. Convergence theory and applications of the factorized distribution algorithm. *Journal of computing and information technology*, *7*(1), 1999a, p. 19-32.

Mühlenbein, h., höns, r. The factorized distribution algorithm and the minimum relative entropy principle. In *scalable optimization via probabilistic modeling*. 2006, p. 11-37, springer, berlin, heidelberg.

Ochoa, a., soto, m. Linking entropy to estimation of distribution algorithms. In i. Inza j.a. Lozano, p. Larrañaga and e. Bengoetxea, editors, *towards a new evolutionary computation. Advances on estimation of distribution algorithms*. 2006, springer.

Santana, r., ochoa, a., soto, m. R. Solving problems with integer representation using a tree based factorized distribution algorithm. In *electronic proceedings of the first international naiso congress on neuro fuzzy technologies*, 2002, academic press.

Santana, r., Larrañaga, p., y Lozano, j. A. Protein folding in simplified models with estimation of distribution algorithms. Ieee *transactions on evolutionary computation, 12(4),* 2008, p. 418 – 438.

Soto, m. A single connected factorize distribution algorithm and its cost of evaluation. Phd thesis, 2003, university of havana, havana, cuba.

Tsagris, m. A new scalable bayesian network learning algorithm with applications to economics. *Computational economics*, 2020, p. 1-27.

Zarges, c. Theoretical foundations of immune-inspired randomized search heuristics for optimization. In *theory of evolutionary computation*, 2020, p. 443-474, springer, cham.

Conflicto de interés

Los autores autorizan la distribución y uso de su artículo.

Contribuciones de los autores

Julio Madera Quintana: Trabajó en la Conceptualización, Adquisición de fondos, Investigación, Metodología, Administración del proyecto, Recursos, Software, Supervisión, Visualización, Redacción – revisión y edición, Análisis formal.

Yoan Martínez López: Curación de datos, Análisis formal, Investigación, Recursos, Software, Validación, Visualización, Redacción – borrador original.

José Fernández Pardo: Curación de datos, Investigación, Recursos, Software, Validación, Redacción– borrador original.