Algoritmos de estimación de distribuciones para el problema de la determinación de la cadena lateral de una proteína

Roberto Santana, Pedro Larrañaga, Jose A. Lozano

Resumen— **Este trabajo analiza diferentes mejoras a una aplicación precedente de los algoritmos de estimación de distribuciones (EDA) al problema de la determinación de la cadena lateral de una proteína. EDAs simples como el UMDA han permitido la obtención de cadenas laterales con menor valor de energía que las obtenidas con otros métodos para diferentes secuencias. Sin embargo para algunas secuencias, los resultados obtenidos con el UMDA no mejoran aquellos obtenidos por otros métodos. Por lo tanto, un problema de interés consiste en el estudio de métodos para aumentar la eficacia de los EDAs en la obtención de cadenas laterales de proteínas.**

Presentamos dos posibles alternativas utilizadas con este propósito: el uso de algoritmos de optimización local y el aprendizaje de modelos probabilísticos que tienen en cuenta las interacciones entre las variables del problema. Los algoritmos introducidos son evaluados en un conjunto de instancias difíciles. Los resultados obtenidos en este conjunto son superiores a los alcanzados con algoritmos de optimización basados en inferencia.

Palabras clave— **plegado de proteinas, algoritmos de estimacion de distribuciones, predicción de la cadena lateral**

I. Introducción

Las proteínas desempeñan un papel fundamental en los organismos vivos. Cada proteína está compuesta por un conjunto de aminoácidos o residuos que, bajo condiciones apropiadas, se pliegan para formar una estructura terciaria. La determinación de la estructura a partir de la secuencia constituye un problema de interés en Biología Molecular [1]. Los modelos computacionales de proteínas son componentes importantes para la determinación de la estructura.

El tipo de modelos utilizados para el problema de la predicción de la estructura comprende desde modelos compactos (coarse-grained models) [2], [3], [4], hasta otros que incluyen mayor nivel de detalle como aquellos basados en la posición espacial de los átomos [5], [6], [7], [8]. Estos modelos permiten la descripción de diferentes estructuras de proteínas a las cuales es posible asociar el valor de una función de energía que sirve para evaluar la calidad de la estructura [9].

Habitualmente, la búsqueda de la mejor configuración se aborda como un problema de optimización: encontrar la solución que optimiza una función de energía predeterminada. El diseño de algoritmos para predecir la estructura de una proteína a partir de su secuencia de aminoácidos es un tema que está recibiendo una creciente atención en el campo de la optimización [6], [8], [10], [11]. En este artículo se trata el problema de la determinación de la estructura, el cual se enfoca a partir del análisis de un problema relacionado, el de la predicción de la cadena lateral de la proteína [12], [13], [14].

Grupo de Sistemas Inteligentes, Departamento de Ciencias de la Computación e Inteligencia Artificial, Universidad del País Vasco. Donostia - San Sebastián, España. email: rsantana@si.ehu.es

Los métodos de optimización propuestos pertenecen a la clase de los algoritmos de estimación de distribuciones (EDAs) [15], [16], [17], que son algoritmos evolutivos que utilizan modelos probabilísticos en lugar de operadores genéticos. Los EDAs son también conocidos como algoritmos evolutivos con estimadores de densidad iterados (iterated density estimators evolutionary algorithms (IDEAS) [18]), y algoritmos genéticos con construcción de modelos (probabilistic model building genetic algorithms (PMBGA) [19]).

El algoritmo evolutivo con distribución marginal univariada (univariate marginal distribution algorithm (UMDA)) [17] es uno de los EDAs que utiliza el modelo probabilístico más simple. En este modelo univariado todas las variables son consideradas de manera independiente. Por esta razón la capacidad de representación del modelo está seriamente limitada. Sin embargo, los resultados experimentales publicados en [20] muestran que, para un conjunto de secuencias de proteínas en las cuales otros algoritmos en el estado del arte fallan, el UMDA es capaz de encontrar cadenas laterales con una mejor estructura.

En este trabajo nos proponemos analizar dos alternativas para mejorar el comportamiento de los EDAs en el problema de la determinación de la cadena lateral de una proteína. La primera alternativa consiste en aumentar la capacidad de representación del modelo probabilístico utilizado. Con este objetivo se propone la utilización de un EDA basado en el uso de árboles. La segunda alternativa se basa en la utilización de algoritmos de optimización local que son insertados en el EDA. El método de optimización local propuesto es un algoritmo de búsqueda en vecindades variables (variable neighborhood search (VNS)) [21], [22], [23], [24].

II. Problema de la cadena lateral de la proteína

Un aminoácido tiene un esqueleto formado por péctidos y una cadena lateral distintiva. Asumiendo que la posición del esqueleto está fija, y considerando que la longitud de las uniones es la misma, la posición de la proteína puede ser completamente determinada por los ángulos de unión. La Figura (1) muestra la estructura original de una proteína, su esqueleto y el conjunto de cadenas laterales.

Una de las aproximaciones al problema de la determinación de la estructura está basada en la modelación por homología. En esta aproximación, una base de datos de proteínas con estructuras conocidas es inspeccionada con

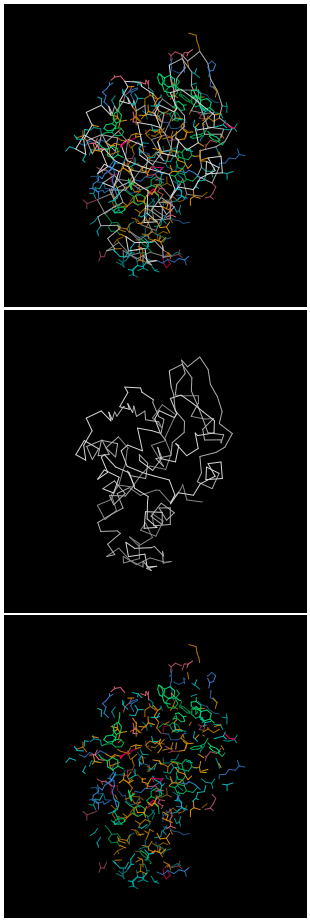


Fig. 1. De arriba hacia abajo: estructura original de la proteína, su esqueleto y el conjunto de cadenas laterales.

el objetivo de encontrar una secuencia homóloga (con un grado relevante de similaridad con la secuencia original). Una vez que se ha encontrado una estructura candidata, la misma es utilizada para buscar la estructura de la proteína deseada.

El problema de encontrar una posición óptima para los residuos de la cadena lateral de una proteína es conocido como el problema de la determinación o la predicción de la cadena lateral [12], [13], [14], habiéndose demostrado que su versión discreta es un problema NP-completo [25]. El problema resulta importante no solo para la modelación por homología sino también para el diseño de proteínas [25], donde el objetivo está en encontrar una proteína capaz de desempeñar una función previamente definida o de satisfacer un número de propiedades estructurales.

Una manera de enfrentar el problema consiste en restringir la búsqueda al espacio discreto por medio de una discretización de las posibles configuraciones de los ángulos, conocidos como rotamers [5], [26]. La inclusión de estas configuraciones discretas implica una reducción importante del problema. Sin embargo, el problema continúa siendo exponencial y la concepción de procedimientos de búsqueda eficientes constituye un tema de interés.

En la literatura se han propuesto algoritmos deterministas y estocásticos para el problema de la determinación de la cadena lateral. En este artículo introducimos un algoritmo de optimación estocástico para la solución del problema. El algoritmo, que está basado en el uso de distribuciones de probabilidad, pertenece a la familia de los EDAs [15], [17]. Los EDAs son algoritmos evolutivos que, de forma similar a los Algoritmos Genéticos (GAs) [27], [28], están basados en poblaciones. Pero, en lugar de utilizar operadores genéticos, construyen en cada generación un modelo probabilístico del conjunto de soluciones seleccionadas y utilizan este modelo para muestrear nuevas soluciones.

III. Representación del problema

En esta sección introducimos la representación empleada en el problema de la cadena lateral de la proteína. Usaremos X_i para representar una variable aleatoria discreta. x_i denotará un posible valor de X_i. De manera similar, usamos $\mathbf{X} = (X_1, \ldots, X_n)$ para representar una variable aleatoria n-dimensional y $\mathbf{x} = (x_1, \ldots, x_n)$ para representar uno de sus posibles valores.

En el problema de la determinación de la cadena lateral, la variable X_i representa el i-ésimo residuo. x_i será interpretada como la configuración del rotamer asociada con el i-ésimo residuo. El número de valores de cada variable se corresponde con el número de posibles configuraciones del rotamer (i.e. $x_i \in \{1, \ldots, K_i\}$, donde K_i es el número de configuraciones del rotamer factibles para el residuo i).

Cuando el esqueleto está fijo la energía de una secuencia de n aminoácidos plegada puede ser expresada como [29]:

$$E(\mathbf{x}) = \sum_{i=1}^{n} E(x_i) + \sum_{i=1}^{n-1} \sum_{j>i}^{n} E(x_i, x_j), \qquad (1)$$

donde $E(x_i)$ engloba la energía de la interacción entre el rotamer y el esqueleto así como la energía intrínseca del rotamer. $E(x_i, x_j)$ es la energía de la interacción entre el par de rotamers i y j. Para dos conjuntos de átomos, la energía de la interacción es igual a la suma de las energías de las interacciones entre los mismos. Hemos adoptado la función de energía de van der Waals tal y como ha sido implementada en [30]. Esta función de energía aproxima la parte repulsiva del potencial de Lennard-Jones 12-6, penalizando los choques estéricos entre átomos. Los residuos que no interactúan tienen todos energía $E(x_i, x_j) = 0$ para cada posible configuración de los rotamers.

Diferentes algoritmos de optimización han sido propuestos para el problema de la determinación de la cadena lateral. Entre los más comunmente usados están los algoritmos dead-end elimination (DEE) [31], la aproximación de campo medio auto consistente (self consistent mean field approach (SCMF) [32]), y el método de búsqueda de cadenas laterales con librerías de rotamers (SCWRL) [5]. También se pueden utilizar métodos basados en inferencia [30], [33] para encontrar soluciones exactas y aproximadas al problema. Otros métodos son descritos en [29].

IV. Algoritmos de estimación de distribuciones

Los EDAs se caracterizan por la modelación probabilística de la información contenida en el conjunto seleccionado. En cada generación estos métodos construyen un modelo probabilístico de las soluciones seleccionadas. El modelo probabilístico tiene que ser capaz de capturar un número de dependencias estadísticas que puedan representar relaciones relevantes entre las variables. Las dependencias son entonces usadas para generar nuevas soluciones muestreando el modelo probabilístico. Cabe esperar que las soluciones generadas compartan un grupo de características con las seleccionadas. De esta manera la búsqueda se orienta hacia áreas prometedoras del espacio de búsqueda.

A. UMDA

El algoritmo evolutivo con distribución marginal univariada (Univariate Marginal Distribution Algorithm (UMDA)) usa marginales univariados para aproximar las distribuciones conjuntas. El UMDA es uno de los EDAs con el modelo probabilístico más simple al no considerar dependencias entre las variables. Resultados teóricos derivados para el UMDA exponen la relación entre los algoritmos genéticos y los EDAs [34].

El modelo probabilístico utilizado por el UMDA es descrito por la Ecuacion (2).

$$p_{UMDA}(\mathbf{x}) = \prod_{i=1}^{n} p(x_i) \tag{2}$$

La probabilidad de cada solución es igual al producto de las probabilidades univariadas. El pseudocódigo del UMDA es mostrado en el Algoritmo 1.

En [20] se propone un algoritmo de optimización basado en la utilización del UMDA para la solución del problema de la determinación de la cadena lateral de la proteína.

El algoritmo comienza calculando la matriz de adyacencia que representa la estructura de las interacciones en el esqueleto de la proteína. El cálculo de la matriz se realiza utilizando un método propuesto en [33]. El uso de la matriz simplifica la evaluación de las soluciones porque permite considerar solamente las interacciones que existen entre proteínas cuyos átomos colindantes en el esqueleto.

Una vez que la matriz ha sido calculada, se determina el número de posibles configuraciones para cada residuo utilizando la librería de rotamers dependientes del esqueleto de la proteína (backbone-dependent rotamer library) [35]. Esta librería incluye las frecuencias, el promedio de los ángulos dihedros y las varianzas como una función de los ángulos internos de rotación del esqueleto de la proteína.

En el siguiente paso se aplica el criterio de eliminación de Goldstein [31]. Este criterio está basado en la Ecuación (3) la cual es utilizada en los algoritmos DEE para eliminar determinadas configuraciones de rotamers de manera iterativa.

$$E(x_i) - E(x_i') + \sum_{\substack{j=1 \\ j \neq i}}^{n} min_{x_j}(E(x_i, x_j) - E(x_i', x_j)) > 0 \tag{3}$$

La Ecuación (3) establece una condición suficiente [31] para que la configuración x_i del rotamer no esté presente en la solución óptima. Si existe una configuración x_i' para la cual se satisface la ecuación entonces x_i puede ser eliminada. El algoritmo se detiene cuando no se puede eliminar ninguna de las restantes configuraciones para las variables. En este caso determinamos cuáles son las variables que tienen más de una posible configuración. Estas variables son las únicas que serán consideradas en el proceso de optimización. Si el espacio determinado por las configuraciones restantes es lo suficientemente pequeño, las posibles soluciones pueden ser inspeccionadas utilizando una búsqueda exhaustiva. Aunque el criterio de eliminación de Goldstein contribuye considerablemente a reducir la dimensión del espacio de búsqueda, para proteínas de tamaño medio y grande la búsqueda en el espacio reducido no se puede acometer utilizando métodos exhaustivos. El criterio de eliminación de Goldstein utilizado por el método DEE es también una componente importante de otros algoritmos de optimización (e.g. SCWRL).

En [20], el UMDA es aplicado al espacio reducido de las configuraciones. Como elemento distintivo de esta aplicación del algoritmo destaca el hecho de que cada variable tiene una cantidad diferente de valores discretos. Los parámetros utilizados por el EDA presentado en [20], que son los mismos que se utilizan en los experimentos presentados en este artículo, son los siguientes.

El tamaño de población es 5000. El máximo número de generaciones es 500. Como método de selección se utiliza la selección por truncamiento con parámetro $T = 0{,}15$. En este método de selección, los mejores $T * M$ individuos de la población son seleccionados para construir el modelo probabilístico. Aplicamos una estrategia de remplazamiento en la cual, la población seleccionada en la generación t es incorporada a la población de la generación $t+1$, manteniendo de esta manera los mejores individuos encontrados hasta el momento y evitando volver a evaluar su función de energía. El algoritmo se detiene cuando el máximo número de generaciones ha sido alcanzado o cuando la población seleccionada es demasiado homogénea (menos de 10 individuos diferentes).

V. Alternativas para mejorar la eficiencia del UMDA

En esta sección consideramos como posibles alternativas para mejorar los resultados obtenidos por el UMDA el uso de modelos probabilísticos capaces de representar dependencias y el uso de algoritmos de optimización local.

A. Uso de dependencias

En términos de la teoría de propagación de bloques constructivos [27], el problema de la estructura de enlace se define por el efecto causado en la evolución por el

Algoritmo 1: UMDA

$t \Leftarrow 0$. Generar N puntos aleatoriamente.
do {
  Evaluar los puntos en la función objetivo.
  Seleccionar un conjunto S de $k \leq N$ puntos de acuerdo a un método de selección.
  Calcular los marginales univariados $p_i^s(x_i,t)$ a partir de S
  Generar N nuevos puntos de acuerdo a la distribución $p(x,t+1) = \prod_{i=1}^{n} p_i^s(x_i,t)$.
  $t \Leftarrow t+1$
} **until** Algún criterio de terminación ha sido satisfecho

rompimiento de importantes soluciones parciales conocidas como bloques constructivos (building blocks). El problema de la estructura de enlace está determinado, no solo por las características del problema de optimización, sino también por la representación usada para resolverlo. Se han propuesto diferentes alternativas [36], [37] que manipulan la representación de las soluciones con el objetivo de hacerlas menos vulnerables a los efectos destructivos de los operadores genéticos.

Otra solución ha sido la creación de operadores genéticos capaces de tratar con las interacciones entre las variables. Sin embargo, estos operadores han sido frecuentemente diseñados considerando las características específicas de los problemas, restringiendo su uso a dominios muy particulares.

La solución propuesta por los EDAs consiste en capturar las interacciones relevantes del problema por medio de dependencias probabilísticas representadas utilizando modelos gráficos probabilísticos. La complejidad de los modelos varía de acuerdo al tipo y número de dependencias utilizadas. Para nuestro trabajo utilizamos un EDA que emplea uno de los modelos probabilísticos más simples entre aquellos capaces de representar dependencias.

El algoritmo de estimación de distribuciones basado en árboles (Tree-EDA) [38], [39] utiliza como modelo probabilístico el árbol. Este modelo es capaz de representar dependencias bivariadas entre variables. El modelo probabilístico está basado en una estructura arbórea donde cada variable puede depender a lo sumo de otra variable que es llamada el padre. Dado un árbol, una distribución probabilística consistente con las relaciones de independencia marginal y condicional representadas por el árbol se define como

$$p_{Tree}(\mathbf{x}) = \prod_{i=1}^{n} p(x_i|pa(x_i)) \tag{4}$$

donde $Pa(X_i)$ es el padre de X_i en el árbol, y $p(x_i|pa(x_i)) = p(x_i)$ cuando $Pa(X_i) = \emptyset$, i.e. X_i es la raíz del árbol.

La determinación de la estructura del árbol a partir de datos implica la detección de las más importantes interacciones bivariadas. Con este propósito utilizamos el algoritmo propuesto por Chow y Liu [40] el cual calcula el árbol de cubrimiento de peso máximo (maximum weight spanning tree (MWST)) a partir de la matriz de información mutua entre las variables. Este algoritmo ha sido previamente utilizado en EDAs [38]. Otros métodos de aprendizaje pueden emplearse para construir el árbol [41], [42]. El pseudocódigo del EDA basado en árboles (Tree-EDA)[1] es mostrado en el Algoritmo 2.

Un problema asociado a los algoritmos utilizados por los EDAs para aprender modelos probabilísticos es la existencia de correlaciones espúreas entre las variables. Este hecho se debe, entre otros factores, al reducido tamaño de las muestras utilizadas para estimar las probabilidades marginales. Las correlaciones espúreas deterioran la exactitud de los modelos e influyen negativamente en la eficiencia de la búsqueda. Nuestros experimentos iniciales mostraron que la calidad de los árboles aprendidos puede ser mejorada cuando las interacciones representadas por el árbol se limitan a aquellas entre residuos que interactúan en el esqueleto de la proteína (la energía de la interacción entre el correspondiente par de rotamers es diferente de cero). Por lo tanto, una variante del algoritmo de aprendizaje que restringe el cálculo de los marginales bivariados y la información mutua a las variables implicadas en este tipo de interacciones puede constituir una versión más eficiente del Tree-EDA. Sin embargo, en el presente trabajo nos limitamos al análisis del Tree-EDA que no considera información sobre la estructura de la proteína durante el aprendizaje.

La complejidad computacional de los EDAs está determinada principalmente por la complejidad del algoritmo de aprendizaje pero depende también del tamaño de población y el número de generaciones que se necesitan para la convergencia. Estos valores varían de acuerdo al problema en cuestión y son difíciles de estimar. Mientras que la complejidad computacional del aprendizaje de modelo del UMDA es lineal en el número de variables, en el caso del Tree-EDA es cuadrática.

B. Algoritmos de optimización local

El VNS [21], [22] está basado en el principio del cambio sistemático de las vecindades durante la búsqueda. El algoritmo explora vecindandes cada vez más alejadas de la solución actual, saltando de una solución a otra solamente si ha obtenido una mejoría del valor de la función a optimizar. De esta forma, características favorables de la solución actual van a ser frecuentemente conservadas

[1]Implementaciones en C++ (EDAProgram) y Matlab (MATEDA) del UMDA, el Tree-EDA, y otros EDAs están respectivamente disponibles en endika@si.ehu.es y http://www.sc.ehu.es/ccwbayes/members/rsantana/software/matlab/

Algoritmo 2: Tree-EDA

$D_0 \leftarrow$ Generar N puntos aleatoriamente
$l = 1$
do {
  $D^s_{l-1} \leftarrow$ Seleccionar $N \leq M$ individuos de D_{l-1} de acuerdo a un método de selección
  Calcular los marginales univariados y bivariados $p^s_i(x_i|D^s_{l-1})$ and $p^s_{i,j}(x_i,x_j|D^s_{l-1})$ of D^s_{l-1}
  Calcular la matriz de información mutua utilizando los marginales univariados y bivariados
  Calcular el árbol de cubrimiento de peso máximo a partir de la matriz información mutua
  Calcular los parámetros del modelo
  $D_l \leftarrow$ Muestrar M individuos (la nueva población) a partir del árbol
} **until** Algún criterio de terminación ha sido satisfecho

y usadas para obtener soluciones prometedoras. Con el fin de obtener soluciones localmente óptimas, un procedimiento de búsqueda local es aplicado de forma repetida a estas soluciones vecinas.

Sea $\mathcal{N}_k, (k = 1,\ldots,k_{max})$ un conjunto finito de estructuras de vecindades previamente fijadas y $\mathcal{N}_k(\mathbf{x})$ el conjunto de soluciones en la k-ésima vecindad de $\mathbf{x}$ que representa una posible solución al problema de optimización. El pseudocódigo del algoritmo VNS se muestra en el Algoritmo 3.

Un elemento crucial del VNS es la definición de la estructura de vecindad. En nuestro caso la vecindad estará definida solamente para los puntos representados por aquellas variables y valores que permanecieron después de la aplicación del criterio de eliminación de Goldstein. Tal y como ha sido explicado en la Sección IV-A, este criterio permite reducir el número de variables y el rango de estas variables. Para el problema de la determinación de la cadena lateral, definimos la k-vecindad de una solución $\mathbf{x}$ como el conjunto de soluciones que son diferentes de la solucion $\mathbf{x}$ en exactamente k variables. De manera más formal,

$$\mathcal{N}_k(\mathbf{x}) = \{\mathbf{x}' \mid n - \sum_{i=1}^{n} I(x_i, x'_i) = k\} \tag{5}$$

donde I es la función indicador, igual a uno si ambos valores coinciden.

Claramente, dado un punto $\mathbf{x}$, para todo $j \neq k$, $\mathcal{N}_j(\mathbf{x}) \bigcap \mathcal{N}_k(\mathbf{x}) = \emptyset$. Adicionalmente, usamos información sobre la estructura de la proteína para restringir la vecindad. Usamos concretamente la información contenida en la matriz de adyacencia. En el análisis de la k-vecindad ($k > 1$), consideramos únicamente aquellos conjuntos de k variables para los cuales cada par de variables tiene una entrada diferente de cero en la matriz de adyacencia. La restricción relativa a la adyacencia de los residuos surge naturalmente del caracter aditivo a pares de variables de la función de energía. Aquellas variables cuyos correspondientes residuos no son adyacentes de acuerdo a la matriz no contribuyen de manera conjunta a la función. La contribución independiente de las variables a la función es cubierta por la 1-vecindad.

En la 1-vecindad de $\mathbf{x}$, se inspeccionan los valores para las variables consideradas por separado. Para $k > 1$, el algoritmo determina aleatoriamente un conjunto de k variables que cumpla que todos los pares de variables tienen una entrada diferente de cero en la matriz de adyacencia. Para el subconjunto de variables seleccionado se selecciona una asignación de valores que es diferente en cada uno de los valores del punto actual $\mathbf{x}$. Exigiendo que la solución sea diferente en las k variables se garantiza que las vecindades no se solapen. Por otro lado, al limitar la búsqueda solamente a los conjuntos de k variables que interactúan en la estructura de la proteína, el algoritmo reduce drásticamente el espacio de búsqueda.

Para esta estructura de vecindades, proponemos un esquema aleatorio del algoritmo de búsqueda local (paso 8 del Algorithmo 3). En este esquema el punto $\mathbf{x}''$ es seleccionado usando una estrategia aleatoria. La búsqueda local es realizada seleccionando aleatoriamente una solución en la vecindad y aceptando esta solución si la energía es mejorada. El parámetro *maxtries* define el máximo número de puntos de la vecindad que pueden ser visitados. El costo del algoritmo depende estrechamente del parámetro *maxtries*. El pseudocódigo del VNS-aleatorio que será aplicado a la mejor solución obtenida por el UMDA se muestra en el Algoritmo 4. En los experimentos realizados $k_{max} = 3$ y $maxtries = 5000$.

VI. Experimentos

El objetivo principal de nuestros experimentos es evaluar hasta qué punto el uso de dependencias y la adición del VNS son capaces de mejorar los resultados inicialmente obtenidos con el UMDA. Primeramente utilizamos el UMDA, UMDA+VNS y el Tree-EDA en la búsqueda de las configuraciones de las cadenas laterales con menor energía y comparamos sus resultados.

Para evaluar nuestros algoritmos escogimos un conjunto de secuencias de proteínas de un conjunto inicial[2] de 463. La base de datos original corresponde a 463 estructuras obtenidas por rayos X con una resolución igual o mejor que $2\mathring{A}$, un factor R por debajo del 20 %, y una similaridad mutua entre las secuencias menor que el 50 %. Cada proteína contiene entre una y cuatro cadenas peptídicas, cada una de las cuales puede tener hasta 1000 residuos. Como un paso de preprocesamiento determina-

[2]Estas instancias han sido obtenidas de la página web de Chen Yanover: http://www.cs.huji.ac.il/∼cheny/proteinsMRF.html

Algoritmo 3: VNS

Inicialización Seleccionar el conjunto estructuras de vecindades $\mathcal{N}_k$ $(k = 1, \ldots, k_{max})$
Encontrar una solución inicial, $\mathbf{x}$
do {
  $k \leftarrow 1$
  do {
    Agitamiento. Generar un punto $\mathbf{x}' \in \mathcal{N}_k(\mathbf{x})$ aleatoriamente
    Búsqueda local. Aplicar algún método de búsqueda local con $\mathbf{x}'$ como solución inicial
    Denotar como $\mathbf{x}''$ al óptimo local así obtenido
    Mover o no. Si el óptimo local es mejor que la solución actual , $\mathbf{x} \leftarrow \mathbf{x}''$ y $k \leftarrow 1$.
    Sino, $k \leftarrow k+1$.
  } **until** $k = k_{max}$.
} **until** Algún criterio de terminación ha sido satisfecho

Algoritmo 4: UMDA+VNS

Partir de la mejor solución $\mathbf{x}$ encontrada por el UMDA
  $k \leftarrow 1$
do {
  $j \leftarrow 1$
  do {
    Generar un punto $\mathbf{x}' \in \mathcal{N}_k(\mathbf{x})$ aleatoriamente
    if $\mathbf{x}' \leq \mathbf{x}$ Then $\mathbf{x} = \mathbf{x}'$, $j \leftarrow 1$, $k \leftarrow 1$
    Else $j \leftarrow j+1$
  } **until** $j = maxtries$
$k \leftarrow k+1$
} **until** $k = k_{max}$

mos las instancias para las cuales el criterio de Goldstein eliminó todas las configuraciones excepto una, y aquellas instancias para las cuales el algoritmo de predicción de estructura basado en inferencia (SPRINT) [30] fue capaz de converger. El número de instancias restantes fue de 50 y de éstas escogimos un conjunto de 27 para la realización de los experimentos.

A. Diseño de los experimentos

Para comparar los resultados de los algoritmos ejecutamos, en la mayor parte de los casos, 50 experimentos para cada instancia y algoritmo. Para un número de instancias más complejas el número de experimentos se redujo a 30. El comportamiento de los algoritmos fue evaluado considerando la energía de la mejor solución encontrada en cada experimento, la mejor solución encontrada en todos los experimentos, y el número de veces que esta mejor solución fue encontrada considerando todos los experimentos realizados.

En la Tabla I se muestran los resultados obtenidos para las 27 instancias. La tabla muestra el tamaño de cada instancia (tamaño), el número de experimentos realizados (exp.), el mejor valor de energía encontrado en todos los experimentos (mejor), el número de veces que la mejor solución fue alcanzada (S), y el promedio de la energía correspondiente a la mejores soluciones obtenidas en los 50 experimentos.

Debido a que el UMDA+VNS parte de la mejor solución encontrada por el UMDA, los resultados son siempre iguales o mejores que los alcanzados con el UMDA. Sin embargo, esto no ocurre para en el caso del Tree-EDA el cual no es capaz de conseguir mejores resultados absolutos que el UMDA para 2 de las 27 instancias. En la Tabla I aparece subrayado el mejor valor obtenido entre el Tree-EDA y el UMDA-VNS para cada una de las instancias. De forma similar aparece en negrita el mejor promedio obtenido entre el Tree-EDA y el UMDA+VNS para cada una de las instancias. Puede apreciarse que el UMDA+VNS obtiene mejores valores promedios que el Tree-EDA.

Un conjunto de experimentos adicionales fueron realizados para investigar las diferencias estructurales entre las mejores soluciones encontradas por el UMDA y aquellas obtenidas usando el Tree-EDA. Este tipo de experimentos es ilustrado usando la proteína *pdb*1*tki* cuya estructura se muestra en la Figura 2a). Esta proteína es un dímero que contiene dos cadenas simétricas y 576 residuos. Las energías correspondientes a las mejores cadenas laterales encontradas por el UMDA y el Tree-EDA fueron respectivamente 858,67 y 856,62.

La estructura encontrada por el Tree-EDA se muestra en la Figura 2b). Las estructuras encontradas por el UM-

			UMDA			Tree-EDA			UMDA+VNS		
pdb id	tamaño	*exp.*	*mejor*	*S*	*media*	*mejor*	*S*	*media*	*mejor*	*S*	*media*
pdb1crz	75	50	626,41	1	627,25	626,12	7	627,54	626,41	1	**627,18**
pdb1ddt	146	50	754,93	1	760,02	754,30	1	760,88	753,38	1	**755,34**
pdb1dpe	185	50	727,37	2	750,51	725,83	1	739,73	725,50	7	**739,35**
pdb1dxr	353	30	1703,73	1	1722,79	1701,61	1	1716,89	1695,16	1	**1705,80**
pdb1dz4	288	30	875,77	1	884,79	868,92	1	880,96	867,01	1	**874,12**
pdb1d2e	281	30	1839,67	1	1847,65	1829,95	1	1841,83	1826,88	6	**1829,08**
pdb1e61	454	30	1936,92	1	1958,89	1942,94	1	1992,96	1926,36	1	**1934,64**
pdb1e6p	365	30	1681,67	1	1694,86	1681,93	1	1703,08	1675,07	1	**1684,93**
pdb1f60	123	30	537,42	3	540,78	536,69	1	540,68	537,04	7	**539,56**
pdb1fmj	294	30	1100,51	1	1121,42	1089,81	1	1105,23	1088,80	1	**1100,25**
pdb1fn9	239	30	989,51	3	993,92	988,82	1	1004,05	987,47	2	**991,77**
pdb1fnn	240	30	735,75	1	749,53	732,90	1	751,41	732,90	1	**733,99**
pdb1giq	265	30	806,53	1	823,58	801,38	1	815,62	800,92	2	**812,52**
pdb1gsk	208	50	939,94	1	947,77	934,79	1	945,74	935,65	1	**939,09**
pdb1h3f	206	30	785,56	1	795,11	784,95	1	794,09	782,98	3	**788,65**
pdb1h3n	318	50	1626,09	1	1639,00	1617,70	1	1653,47	1620,39	1	**1627,07**
pdb1h4r	227	30	825,64	1	830,12	816,96	1	824,09	815,84	8	**817,14**
pdb1h80	229	30	1036,90	1	1040,45	1034,96	1	1039,07	1034,77	8	**1035,05**
pdb1iqc	288	30	1538,37	1	1546,85	1531,80	1	1536,20	1530,18	1	**1534,54**
pdb1jmx	285	30	1518,10	1	1545,51	1510,21	1	1534,77	1515,11	1	**1533,47**
pdb1jy1	144	50	861,92	1	870,34	856,88	4	860,76	856,84	4	**858,39**
pdb1j3b	289	30	1600,16	1	1625,67	1592,75	1	1616,81	1586,47	1	**1602,27**
pdb1j8f	329	30	957,08	1	964,50	942,69	3	954,67	942,62	13	**943,74**
pdb1kmo	241	50	925,90	1	943,09	890,85	1	924,32	901,88	1	**918,15**
pdb1kwh	207	50	972,11	1	988,21	960,73	2	980,09	960,73	1	**972,45**
pdb1lqt	268	30	935,95	1	967,32	926,16	1	**941,13**	926,16	1	954,38
pdb1lsh	350	30	1125,04	1	1135,00	1120,77	1	1132,27	1118,95	1	**1125,89**
pdb1nqe	189	50	570,29	1	593,49	563,36	1	588,67	565,37	3	**580,76**
pdb1tki	164	30	858,67	1	867,24	856,62	1	860,97	855,56	2	**858,12**

TABLA I

RESULTADOS OBTENIDOS POR EL UMDA, EL TREE-EDA Y EL UMDA+VNS PARA EL CONJUNTO DE INSTANCIAS SELECCIONADOS.

DA y el Tree-EDA se diferencian solamente en 7 residuos. Estos residuos son identificados por sus correspondientes números de secuencia en la Figura 2b). Sus respectivas configuraciones de rotamer determinan la mejoría en la energía. En la figura, los residuos que interactúan en el esqueleto aparecen unidos. Tree-EDA es capaz de identificar el aporte a la energía de residuos que interactúan. El UMDA es capaz de encontrar configuraciones óptimas pero esto ocurre con mayor probabilidad para residuos que no interactúan o para los cuales las interacciones son débiles. Este hecho explica que para numerosos casos el uso de las dependencias probabilísticas pueda mejorar los resultados.

Por otro lado, una condición importante para el aprendizaje y conveniente explotación de las dependencias es contar con un tamaño de población adecuado. Un tamaño de población insuficiente podría provocar que los resultados obtenidos por EDAs que utilizan modelos capaces de representar dependencias no superen a aquellos obtenidos por el UMDA. Este hecho podría explicar por qué para algunas de las instancias mostradas en la Tabla I el promedio de los resultados alcanzados con el Tree-EDA no mejora los obtenidos con el UMDA.

VII. CONCLUSIONES Y TRABAJO FUTURO

En este artículo hemos presentado dos posibles alternativas para la mejora del comportamiento del UMDA en el problema de la determinación de la cadena lateral de una proteína. Tanto el uso del algoritmo UMDA+VNS como el del Tree-EDA han mostrado que es posible obtener cadenas laterales de mejor energía que aquellas encontradas con el UMDA.

Como posibles líneas de investigación futura se encuentra la evaluación de otros métodos de optimización local como la Búsqueda Tabú [43], y de EDAs con modelos probabilísticos más complejos como las mezclas de árboles [44]. Así mismo el uso de información sobre la estructura de la proteína es un camino prometedor para aumentar la eficiencia del Tree-EDA.

AGRADECIMIENTOS

Los autores agradecen a Chen Yanover por aportar el conjunto de instancias usadas en nuestros experimentos. Este trabajo ha sido subvencionado por los proyectos de investigación SAIOTEK-Autoinmune (II) Etortek 2006 del Gobierno Vasco, y el proyecto TIN 2005-03824 del Ministerio Español de Educación y Ciencia.

REFERENCIAS

[1] B. Al-Lazikani, J. Jung, Z. Xiang, and B. Honig, "Protein structure prediction," *Current Opinion in Chemical Biology*, vol. 5, no. 1, pp. 51–56, 2001.

[2] K. A. Dill, "Theory for the folding and stability of globular proteins," *Biochemistry*, vol. 24, no. 6, pp. 1501–1509, 1985.

[3] Andrzej Kolinski and Jeffrey Skolnick, "Reduced models of proteins and their applications," *Polymer*, vol. 45, no. 2, pp. 511 – 524, 2004.

[4] A. Mongea, E. J. P. Lathropa, J. R. Gunna, P. S. Shenkina, and R. A. Freisnera, "Computer modeling of protein folding: Conformational and energetic analysis of reduced and detailed protein models," *Journal of Molecular Biology*, vol. 247, no. 5, pp. 995–1012, 1995.

[5] R. L. Dunbrack, "Rotamer libraries in the 21st century," *Current Opinion in Structural Biology*, vol. 12, pp. 431–440, 2002.

[6] C. M. Kraemer-Pecore, A. M. Wollacott, and J. R. Desjarlais, "Computational protein design," *Current Opinion in Chemical Biology*, vol. 5, pp. 690–695, 2001.

[7] G. A. Lazar, J. R. Desjarlais, and T. M. Handel, "De *novo* protein design of the hydrophobic core of ubiquitin," *Protein Science*, vol. 6, pp. 1167–1178, 1997.

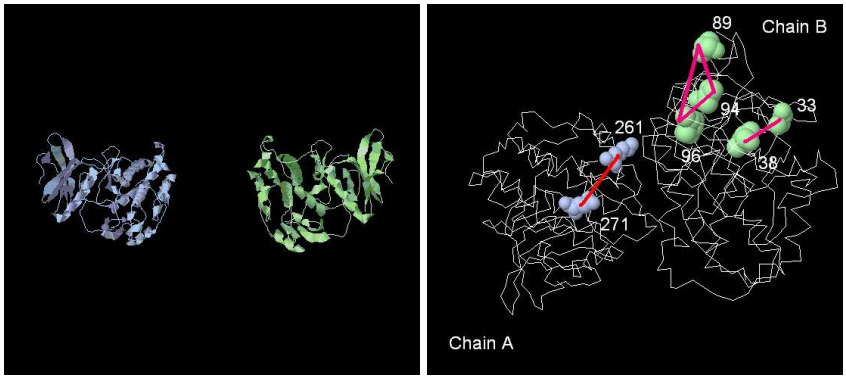
[8] C. A. Rohl, C. E. M. Strauss, K. Misura, and D. Baker, "Protein structure prediction using Rosetta," *Methods in Enzymology*, vol. 383, pp. 66–93, 2004.

[9] T. Dandekar and R. Köenig, "Computational methods for the prediction of protein folds," *Biochimica et Biophysica Acta (BBA) - Protein Structure and Molecular Enzymology*, vol. 1343, no. 1, pp. 1–15, 1997.

[10] A. A. Canutescu, A. A. Shelenkov, and R. L. Dunbrack, "A graph-theory algorithm for rapid protein side-chain prediction," *Protein Science*, vol. 12, pp. 2001–2014, 2003.

[11] S. Liang and N. V. Grishin, "Side-chain modeling with an optimized scoring function," *Protein Science*, vol. 11, pp. 322–331, 2002.

[12] C. Lee and S. Subbiah, "Prediction of protein side-chain conformation by packing optimization," *Journal of Molecular Biology*, vol. 217, pp. 373–388, 1991.



a) b)

Fig. 2. Proteina *pdb*1*tki* (a) Estructura original. (b) Mejor estructura aprendida por el Tree-EDA.

[13] P. S. Shenkin, H. Farid, and J. S. Fetrow, "Prediction and evaluation of side-chain conformations for protein backbone structures," *Proteins: Structure, Function, and Genetics*, vol. 26, no. 3, pp. 323–352, 1998.

[14] M. Vasquez, "Modeling side-chain conformation," *Current Opinion in Structural Biology*, vol. 6, no. 2, pp. 217–221, 1996.

[15] P. Larrañaga and J. A. Lozano, Eds., *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*, Kluwer Academic Publishers, Boston/Dordrecht/London, 2002.

[16] J. A. Lozano, P. Larrañaga, I. Inza, and E. Bengoetxea, Eds., *Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms*, Springer-Verlag, 2006.

[17] H. Mühlenbein and G. Paaß, "From recombination of genes to the estimation of distributions I. Binary parameters," in *Parallel Problem Solving from Nature - PPSN IV*, Hans-Michael Voigt, Werner Ebeling, Ingo Rechenberg, and Hans-Paul Schwefel, Eds., Berlin, 1996, pp. 178–187, Springer Verlag, LNCS 1141.

[18] P. A. Bosman and D. Thierens, "Linkage information processing in distribution estimation algorithms," in *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-1999*, Wolfgang Banzhaf, Jason Daida, Agoston E. Eiben, Max H. Garzon, Vasant Honavar, Mark Jakiela, and Robert E. Smith, Eds., Orlando, FL, 1999, vol. I, pp. 60–67, Morgan Kaufmann Publishers, San Francisco, CA.

[19] M. Pelikan, D. E. Goldberg, and F. Lobo, "A survey of optimization by building and using probabilistic models," *Computational Optimization and Applications*, vol. 21, no. 1, pp. 5–20, 2002.

[20] R. Santana, P. Larrañaga, and J. A. Lozano, "Side chain placement using estimation of distribution algorithms," *Artificial Intelligence in Medicine*, 2006, In Press. Available online 18 July 2006.

[21] N. Mladenović, "A variable neighborhood algorithm – a new metaheuristics for combinatorial optimization," in *Abstracts of Papers Presented at Optimization Days. Montréal*, 1995, p. 112.

[22] N. Mladenović and P. Hansen, "Variable neighborhood search," *Computers and Operation Research*, vol. 24, pp. 1097–1100, 1997.

[23] P. Hansen and N. Mladenović, "Variable neighborhood search," in *Handbook of Applied Optimization*, P. Pardalos and M. Resende, Eds., pp. 221–234. Oxford University Press, 2002.

[24] P. Hansen and N. Mladenović, "Variable neighborhood search," in *Handbook of Metaheuristics*, F. Glover and G. Kochenberger, Eds., pp. 145–184. Kluwer Academic Publisher, 2003.

[25] N. Pokala and T. M. Handel, "Review: Protein design–where we were, where we are, where we're going," *Journal of Structural Biology*, vol. 134, pp. 269–281, 2001.

[26] J. W. Ponder and F. M. Richard, "Tertiary templates for proteins. Use of packing criteria in the enumeration of allowed sequence for different structure classes," *Journal of Molecular Biology*, vol. 193, pp. 775–791, 1987.

[27] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA, 1989.

[28] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI, 1975.

[29] C. A. Voigt, D. B. Gordon, and S. L. Mayo, "Trading accuracy for speed: A quantitative comparison of search algorithms in protein sequence design," *Journal of Molecular Biology*, vol. 299, no. 3, pp. 799–803, 2000.

[30] C. Yanover and Y. Weiss, "Approximate inference and protein-folding.," in *Advances in Neural Information Processing Systems 15*, S. Becker, S. Thrun, and K. Obermayer, Eds., pp. 1457–1464. MIT Press, Cambridge, MA, 2003.

[31] M. De Maeyer, J. Desmet, and I. Lasters, "The dead-end elimination theorem: Mathematical aspects, implementation, optimization, evaluation, and performance," *Methods in Molecular Biology*, vol. 143, pp. 265–304, 2000.

[32] P. Koehl and M. Delarue, "Building protein lattice models using self consistent mean field theory," *Journal of Chemical Physics*, vol. 108, pp. 9540–9549, 1998.

[33] C. Yanover and Y. Weiss, "Approximate inference for side-chain prediction," Submitted for publication, 2004.

[34] H. Mühlenbein and T. Mahnig, "Evolutionary computation and beyond," in *Foundations of Real-World Intelligence*, Y. Uesaka, P. Kanerva, and H. Asoh, Eds., pp. 123–188. CSLI Publications, Stanford, California, 2001.

[35] R. L. Dunbrack and F. E. Cohen, "Bayesian statistical analysis of protein side-chain rotamer preferences," *Protein Science*, vol. 6, no. 8, pp. 1661–1681, 1997.

[36] K. Deb, *Binary and floating-point function optimization using messy genetic algorithms*, Ph.D. thesis, University of Illinois at Urbana-Champaign, Urbana, Illinois, 1991.

[37] C. H. M. van Kemenade, "Explicit filtering of building blocks for genetic algorithms," in *90*, p. 9. Centrum voor Wiskunde en Informatica (CWI), ISSN 0169-118X, 31 1996.

[38] S. Baluja and S. Davies, "Using optimal dependency-trees for combinatorial optimization: Learning the structure of the search space," in *Proceedings of the 14th International Conference on Machine Learning*. 1997, pp. 30–38, Morgan Kaufmann.

[39] R. Santana, E. Ponce de León, and A. Ochoa, "The edge incident model," in *Proceedings of the Second Symposium on Artificial Intelligence (CIMAF-99)*, Habana, Cuba, March 1999, pp. 352–359.

[40] C. K. Chow and C.Ñ. Liu, "Approximating discrete probability distributions with dependence trees," *IEEE Transactions on Information Theory*, vol. 14, no. 3, pp. 462–467, 1968.

[41] M. Pelikan and H. Mühlenbein, "The bivariate marginal distribution algorithm," in *Advances in Soft Computing - Engineering Design and Manufacturing*, R. Roy, T. Furuhashi, and P.K. Chawdhry, Eds., London, 1999, pp. 521–535, Springer-Verlag.

[42] M. R. Soto, *A Single Connected Factorized Distribution Algorithm and Its Cost of Evaluation*, Ph.D. thesis, University of Havana, Havana, Cuba, July 2003, In Spanish.

[43] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Computers and Operations Research*, vol. 5, pp. 533–549, 1986.

[44] R. Santana, A. Ochoa, and M. R. Soto, "The mixture of trees factorized distribution algorithm," in *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-2001*, San Francisco, CA, 2001, pp. 543–550, Morgan Kaufmann Publishers.