Métodos Basados en Optimización por Colonia de Hormigas Aplicados al Modelo Hidrofóbico-Polar del Plegamiento de Proteínas

por

Margarita Rosete Montero

Tesis sometida como requisito parcial
para obtener el grado de

MAESTRA EN CIENCIAS EN EL ÁREA DE CIENCIAS COMPUTACIONALES

en el

Instituto Nacional de Astrofísica, Óptica y Electrónica
INAOE

Supervisada por:

Dr. Jesús A. González Bernal
Coordinación Ciencias Computacionales INAOE

Dr. Eduardo Morales Manzanares
Coordinación Ciencias Computacionales INAOE

Febrero 2009
Tonanzintla, Puebla

Resumen

En diversos campos del conocimiento se presentan problemas combinatorios de optimización complejos cuya resolución por medios computacionales resulta muy atractiva. Un problema dentro de la biofísica y la bioquímica es el problema del plegamiento de proteínas (PFP, por sus siglas en inglés). Este problema consiste, de manera general, en predecir la estructura tridimensional de una proteína a partir de su cadena lineal de aminoácidos. Las proteínas desempeñan un papel fundamental en las funciones de los seres vivos, y debido a que la función de una proteína está determinada por su estructura tridimensional, el estudio del PFP es muy importante. A través de los años se han propuesto varios algoritmos para intentar resolver este problema. Entre ellos encontramos métodos de Monte Carlo y recocido simulado, algoritmos genéticos y, en años recientes, optimización por colonia de hormigas (ACO), la cual se ha mostrado como una metaheurística poderosa para resolver problemas combinatorios de optimización. Si embargo, ninguno de los algoritmos en la literatura se ha mostrado completamente superior a los otros, por lo que el desarrollo de nuevas propuestas que reduzcan el tiempo de ejecución y encuentren soluciones óptimas es de gran importancia. En esta tesis se proponen varios algoritmos basados en ACO enfocados al PFP. Los dos primeros varían en dos componentes: la función heurística y el esquema de actualización de feromona. A partir de estrategias diferentes para combinar la información proporcionada por estos dos algoritmos individuales se presentan tres versiones de algoritmos ACO híbridos. Se presentan también dos nuevas técnicas que mejoran el desempeño general de los algoritmos: una técnica para evitar el estancamiento en los algoritmos y otra técnica para resolver el problema de traslape en la construcción de soluciones. Todos los algoritmos fueron evaluados utilizando secuencias estándar del modelo HP cuadrado y triangular 2D, los cuales son una simplificación del PFP original. Comparados con otros algoritmos en la literatura, los algoritmos propuestos mostraron buenos resultados en cuanto a la calidad de las soluciones encontradas y a su tiempo de ejecución.

Abstract

In many knowledge fields different complex combinatorial optimization problems arise, whose solution by computational means results very attractive. One such problem in biophysics and biochemistry is the protein folding problem (PFP). This problem consists, in a general way, in predicting the tridimensional structure of a protein from its lineal amino acids chain. Proteins play a predominant role in the biological functions of living organisms, and because of functions of proteins are determined precisely by their tridimensional structure, studying the PFP is of high importance. Throughout the years, several algorithms have been proposed to solve this problem. Among them are the Monte Carlo and simulated annealing methods, evolutionary algorithms and, in recent years, ant colony optimization (ACO). ACO has shown to be a powerful metaheuristic able to solve combinatorial optimization problems. However, any of the algorithms presented in the literature has shown to be completely superior to others. For that reason, the development of new proposals that reduce the execution time and find optimal solutions is very important. In this thesis several algorithms based in ACO and oriented to the study of PFP are proposed. The first two algorithms differ between them in two principal components: the heuristic function and the pheromone actualization scheme. From different strategies to combine the information given by these two individual algorithms, three versions of hybrid-ACO algorithms are presented. Two new techniques that improve the global performance of the algorithms are also presented: a technique to avoid stagnation in algorithms and a new method to solve overlapping problem. Every algorithm was evaluated using standard instances of the HP square and triangular model, which are simplifications of the original PFP. Compared with other algorithms in the literature, in general, the algorithms presented in this thesis showed a good performance in terms of the quality of solutions and the execution time.

Agradecimientos

Al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo económico otorgado a través de la beca para estudios de maestría.

Al Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE) y a la Coordinación de Ciencias Computaciones por todas las facilidades que me otorgaron durante mis estudios de maestría.

A mis asesores, Dr. Jesús A. González Bernal y Dr. Eduardo Morales Manzanares, por su apoyo en la realización de esta tesis.

A mis sinodales, Dr. Luis Enrique Sucar S., Dra. Angélica Muñoz M., Dr. Luis Villaseñor P., por sus observaciones y comentarios.

Al Dr. Ramón Garduño Juárez por su apoyo como consultor externo, y cuyos conocimientos y aportaciones contribuyeron en la realización de esta tesis. De igual manera, se agradece al apoyo parcial de los proyectos CONACyT 40061 y PAPIIT-UNAM IN109508.

Índice

Índice de figuras

Índice de tablas

CAPÍTULO 1

Introducción

En diversos campos del conocimiento como medicina, biología, economía e ingeniería se presentan problemas combinatorios de optimización complejos. En las ciencias computacionales este tipo de problemas han sido llamados NP-duros (no existe un algoritmo conocido capaz de resolverlos en tiempo polinomial), debido a la gran dificultad que existe para resolverlos. Dentro de los campos de la bioquímica y biofísica un problema de este tipo, que ha sido objeto de estudio desde hace ya más de medio siglo, es el llamado problema del plegamiento de proteínas.

Las proteínas son compuestos orgánicos complejos con una gran importancia biológica. En la naturaleza, las proteínas intervienen prácticamente en todas las propiedades que caracterizan a los organismos vivos. Las proteínas almacenan y transportan una gran variedad de partículas, desde macromoléculas hasta electrones; guían el flujo de electrones en el proceso vital de la fotosíntesis; y como hormonas, transmiten información entre células y órganos específicos en organismos complejos. Las proteínas son los componentes determinantes de los músculos y otros sistemas

debido a que convierten la energía química en energía mecánica. También, las proteínas son necesarias para la vista, el oído, y otros sentidos. Además, muchas proteínas funcionan como estructuras, otorgando la arquitectura filamentosa dentro de las células y los materiales que son usados en la formación de cabello, uñas, tendones y huesos de animales.

A pesar de sus funciones biológicas tan diversas, las proteínas son una clase de moléculas relativamente homogéneas. Todas están constituidas de una secuencia lineal formada de varias combinaciones de los mismos 20 aminoácidos. El secreto de su diversidad funcional recae, en parte, en la diversidad química de los aminoácidos, pero principalmente en la diversidad de las estructuras tridimensionales que estos bloques de construcción pueden formar. Las increíbles propiedades funcionales de las proteínas están determinadas por la estructura tridimensional que forman al plegarse las secuencias lineales de aminoácidos constituyentes de las proteínas. El número de posibles conformaciones que una proteína puede adoptar a partir de su secuencia de aminoácidos formando una estructura compacta biológicamente activa crece de manera exponencial conforme aumenta el número de aminoácidos en la secuencia lineal. Sin embargo, en la naturaleza este plegado se lleva a cabo en tiempos que van desde milisegundos hasta minutos.

Comprender cómo la estructura y función de las proteínas emergen a partir de su secuencia de aminoácidos es muy importante. Las motivaciones para estudiar el plegamiento de proteínas están conectadas directamente con la habilidad de deducir las funciones de las proteínas. Algunas de estas motivaciones son:

- Predecir la función de una proteína mediante la obtención de la estructura tridimensional de dicha proteína conociendo sólo su secuencia lineal de aminoácidos.

- Comprender y combatir varias enfermedades como Alzheimer, Parkinson, cataratas, enfisema pulmonar, fibrosis quística, entre otras (unas veinte conocidas hasta la fecha) que son causadas por un mal plegado de las proteínas.

- Diseñar proteínas con una estructura y función predeterminadas.

- Descifrar con éxito genomas enteros.

Además de las motivaciones biológicas para el estudio del plegado de proteínas, éste constituye un problema combinatorio de optimización importante, ya que consiste en encontrar un mínimo global en superficies de energía potencial altamente complejas.

Algunas estructuras de proteínas han sido determinadas usando técnicas como cristalografía de rayos-X y resonancia magnética nuclear (RMN). Sin embargo, para aplicar tales técnicas se utiliza equipo muy costoso, se necesita personal calificado, se requiere de la purificación y cristalización de la proteína y, en términos generales, se necesita de mucho tiempo (hasta años) y trabajo (establecer las condiciones precisas en estas técnicas experimentales es complicado) para estudiar las proteínas. Por este motivo, una técnica basada en principios algorítmicos para predecir la estructura tridimensional de las proteínas resulta muy atractiva. Debido a la complejidad que supone obtener información experimental acerca de los principios estructurales que rigen el proceso del plegado de las proteínas, se han propuesto modelos simplificados que permiten obtener información relevante de dicho proceso. Uno de estos modelos es el llamado modelo hidrofóbico-polar (HP). Este modelo reduce el alfabeto de 20 diferentes aminoácidos que una proteína puede tener, a un alfabeto binario que categoriza a los aminoácidos como H (hidrofóbicos) o P (polares). Así, la secuencia lineal de una proteína está dada como una cadena de H’s y P’s. Esta cadena es después organizada dentro de una malla como un camino en el cual cada vértice de la

malla sólo puede ser ocupado por un aminoácido a la vez, formando una determinada conformación. Las diferentes conformaciones que pueden formarse de una misma cadena lineal bajo estas condiciones son evaluadas usando una función de energía calculada en base al número de aminoácidos hidrofóbicos (H) que son adyacentes en la malla pero que no son contiguos en la secuencia lineal. De esta manera, el objetivo es encontrar una conformación con el número más grande de contactos H-H.

El problema del plegamiento de proteínas consiste en la predicción de la estructura tridimensional de una proteína conociendo sólo su secuencia lineal de aminoácidos. Para resolver el problema del plegamiento de proteínas simplificado con el modelo HP se han propuesto varios métodos a través de los años. Dentro de estos métodos encontramos la Búsqueda Tabú (Morales et al., 2000), el Recocido Simulado (Garduño et al., 1990; Morales et al., 1991), los métodos de Monte Carlo (Ramakrishnan et al., 1997; Bastolla et al., 1998; Chikenji et al., 1999; Liang et al., 2001; Zhang y Liu, 2002), los Algoritmos Evolutivos (Unger y Moult, 1993a; Unger y Moult, 1993b; Patton y Goldman, 1995; Krasnogor et al., 1998; Garduño et al., 2003) y la Optimización por Colonia de Hormigas (ACO) (Shmygelska et al., 2002; Shmygelska y Hoos, 2003; Chu et al., 2005; Shmygelska y Hoos, 2005; Fidanova, 2006; Song et al., 2006).

A pesar de que con los diferentes métodos propuestos a través de los años para resolver el problema del plegamiento de proteínas se han tenido avances en cuanto a la reducción del tiempo para la obtención de buenos resultados y se han logrado buenas aproximaciones, los diferentes métodos propuestos en la literatura trabajan con secuencias de proteínas relativamente cortas, regularmente con una longitud menor a 100 aminoácidos. Uno de los objetivos principales de estos métodos es encontrar conformaciones con niveles de energía cada vez menores, hasta alcanzar el nivel óptimo, en un tiempo de ejecución cada vez menor. De esta manera se podría conseguir trabajar con secuencias de proteínas de longitud mayor a 100 aminoácidos. Además, la investigación que conduce a la creación de algoritmos que permitan

encontrar soluciones al plegamiento de proteínas utilizando el modelo HP podrá asistir a desarrollos futuros en torno a la solución de problemas de plegamiento más complejos. El desarrollo de nuevas heurísticas y técnicas de búsqueda permite encontrar métodos que van mejorando en cuanto a las predicciones de las estructuras de las proteínas y en la disminución del tiempo de ejecución global.

Los métodos que mejor desempeño han mostrado, en cuanto a conformaciones encontradas y tiempo de ejecución, son los métodos de Monte Carlo y los métodos basados en Optimización por Colonia de Hormigas (ACO). Sin embargo, los métodos Monte Carlo utilizan modelos complejos para resolver el problema del plegamiento de proteínas y son difíciles de implementar. Además, a pesar de que los métodos Monte Carlo han sido estudiados desde hace varios años no muestran ser significativamente mejores a los métodos ACO, los cuales tienen relativamente pocos años de ser estudiados. Por esta razón, los métodos ACO son una alternativa prometedora para resolver el plegamiento de proteínas.

En la siguiente sección se introduce de manera formal la descripción del problema del plegamiento de proteínas y más adelante se dan los objetivos, aportaciones principales y retos en el desarrollo de esta tesis. Finalmente, en la última sección se presenta la organización de la tesis.

1.1 Planteamiento del problema

El problema del plegamiento de proteínas se define como la predicción de la estructura nativa (estructura tridimensional) de una proteína a partir de su secuencia lineal de aminoácidos (estructura primaria). Formalmente, dicho problema puede definirse como:

Dadas:

$s = s_1 s_2 ... s_n$, una determinada secuencia de aminoácidos.

$E(c)$, una función de energía para evaluar c (una conformación válida de s).

Encontrar una conformación de s de mínima energía, esto es:

$$c^* \in C(s) \text{ tal que } E(c^*) = min \{ E(c) \mid c \in C(s) \},$$

donde $C(s)$ es el conjunto de todas las conformaciones válidas para s.

En esta tesis el problema general está delimitado por las características del modelo HP de la siguiente manera:

- La secuencia de aminoácidos s está dada por una cadena de aminoácidos de tipo H o P.

- Las diferentes conformaciones de s están definidas dentro de una malla (cuadrada o triangular).

- Una posición en la malla sólo puede ser ocupada por un aminoácido a la vez.

- Los grados de libertad o direcciones que un aminoácido tiene en el modelo cuadrado son tres: adelante (S), izquierda (L) y derecha (R). En el modelo triangular son cinco: S (0º), A (60º), B (120º), C (240º) y D (300º). Estas direcciones se muestran en la Fig. 1.1.

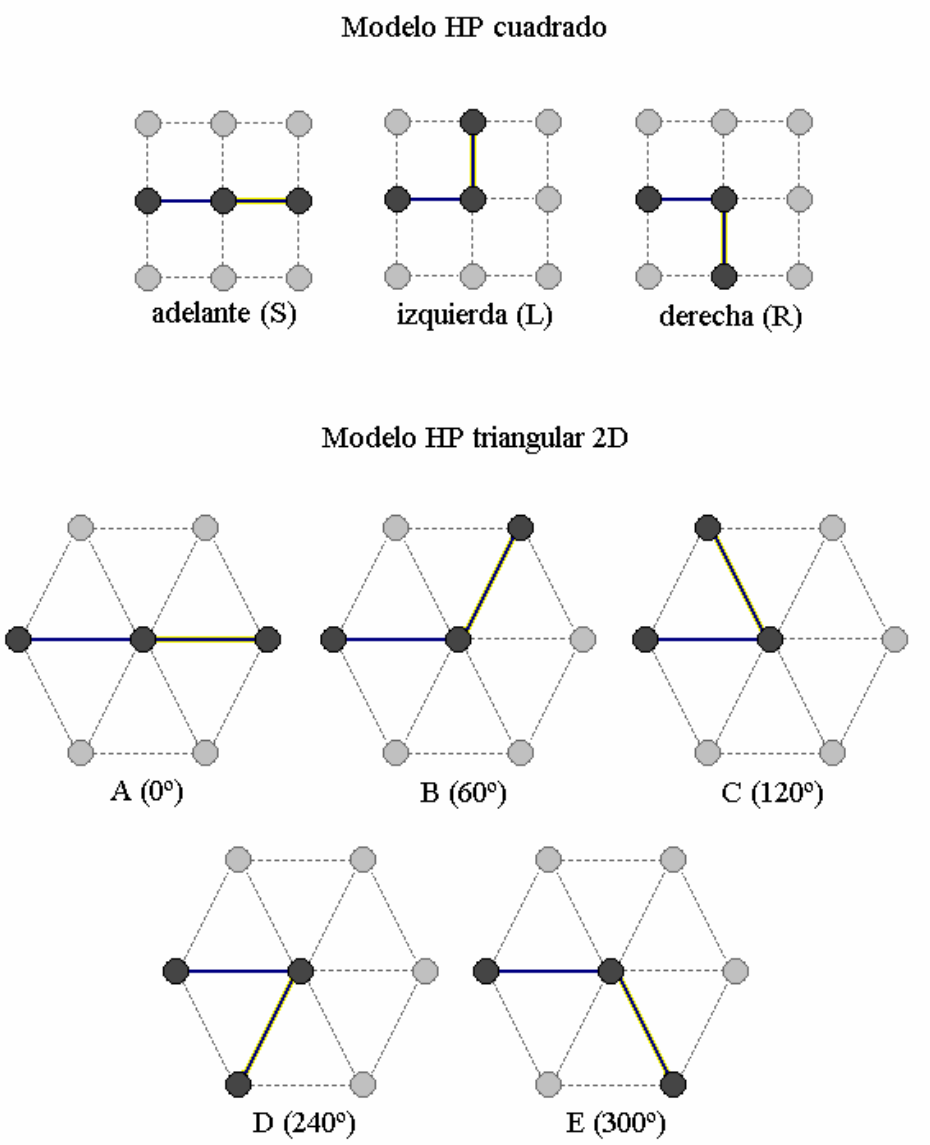


Fig. 1.1. Grados de libertad de los aminoácidos cuando se forman conformaciones a partir de la secuencia lineal de aminoácidos.

- La función de energía de una conformación está definida como:

$$E(c) = (-1)\ (\textit{Num. contactos H-H})$$

donde (*Num. contactos H-H*) se refiere al número de aminoácidos de tipo H que son adyacentes en la malla y que no son contiguos en la secuencia lineal.

1.2 Retos computacionales

Los algoritmos ACO han sido estudiados y utilizados en diferentes problemas desde su creación hace ya poco más de 15 años (Dorigo, 1992). En los algoritmos ACO varios agentes computacionales llamados hormigas construyen soluciones a un determinado problema basándose en la información obtenida de dos elementos principales en el algoritmo. Estos elementos son: una función heurística dependiente del problema; y la feromona acumulada durante la ejecución del algoritmo, la cual consiste de valores numéricos que se modifican según pasan las hormigas por las diferentes opciones en la construcción de soluciones.

La aplicación de ACO a un problema en específico implica establecer una heurística adecuada para guiar la construcción de soluciones. Sin embargo, dicha heurística no está limitada a una sola, sino que para un mismo problema pueden proponerse varias de ellas. Además, diferentes valores en los parámetros que combinan la información de la feromona y la información de la función heurística pueden dar diferentes resultados. El desempeño de los algoritmos ACO suele estar ligado fuertemente a estos factores, por lo que la correcta determinación de los mismos es importante de estudiar.

En particular para el problema del plegamiento de proteínas, el cual es el objeto de estudio de esta tesis, es muy común encontrar problemas relacionados con mínimos locales. El cómo conseguir superar o salir de estados en los que las hormigas no consiguen mejorar las soluciones que han encontrado hasta un determinado momento, normalmente es solucionado con el uso de búsquedas locales. Aunque dichas búsquedas pueden producir mejoras en el desempeño, la determinación

correcta de los operadores y el tipo de búsqueda suelen ser por lo regular difíciles de establecer. Además de que el tiempo de ejecución de dichas búsquedas también puede ser elevado y mientras más exhaustivas son las búsquedas, más tiempo consumen. Con esto, proponer alternativas para salir de estancamientos y mínimos locales es de gran interés.

Para problemas complejos de optimización combinatorios, como lo es el problema del plegamiento de proteínas, encontrar soluciones cada vez mejores en poco tiempo de ejecución es de gran importancia. Además, el funcionamiento uniforme en diferentes secuencias de un mismo problema también es deseable. Por lo mismo, identificar puntos clave que consigan un equilibrio entre las soluciones encontradas y el tiempo de ejecución de los algoritmos también es relevante.

Desde el punto de vista del problema del plegamiento de proteínas, el mayor reto consiste en encontrar la estructura tridimensional óptima de una secuencia lineal de aminoácidos en el menor tiempo posible. Sin embargo, aún utilizando un modelo simplificado, como el modelo HP, el problema del plegamiento de proteínas es de tipo NP-duro. La cantidad de estructuras tridimensionales posibles que se pueden formar a partir de una sola secuencia lineal de aminoácidos crece de manera exponencial conforme se incrementa la longitud de la cadena lineal de aminoácidos, volviéndose más complejo el problema. Por este motivo, mientras más larga sea la cadena lineal de aminoácidos, mayor será el tiempo que se requiera para encontrar la estructura tridimensional óptima asociada a dicha secuencia. Debido a lo anterior, la creación de algoritmos que consigan mejorar las soluciones y reducir el tiempo de ejecución en la búsqueda de soluciones es de gran importancia en el problema del plegamiento de proteínas.

1.3 Objetivos de la tesis

El objetivo principal de esta tesis es el diseño e implementación de un conjunto de métodos basados en la optimización por colonia de hormigas (ACO) enfocados al problema del plegamiento de proteínas simplificado con la representación del modelo hidrofóbico-polar (HP). De forma particular, dentro de este extenso problema, se busca:

- Proponer métodos que tengan un compromiso entre las soluciones encontradas y el tiempo de ejecución de los mismos.

- Desarrollar una estrategia para salir de mínimos locales en la búsqueda de soluciones de los diferentes métodos.

- Identificar puntos importantes en los métodos que sean clave para la obtención de mejores resultados.

1.4 Aportaciones de la tesis

En esta tesis se muestra un conjunto de algoritmos basados en ACO aplicados al problema del plegamiento de proteínas simplificado con el modelo HP (cuadrado y triangular). Se presentan dos algoritmos ACO individuales y tres algoritmos ACO híbridos. Los algoritmos ACO híbridos consisten en la ejecución de forma paralela de los dos algoritmos ACO individuales y combinan de diferente manera la información proporcionada por los mismos. Las aportaciones principales de esta tesis se resumen en los siguientes puntos.

- El desarrollo y evaluación de tres métodos ACO híbridos que combinan dos métodos ACO individuales diferentes.

- El desarrollo de una nueva técnica para evitar estancamiento en los métodos ACO, la cual mejora el desempeño de los mismos.

- El desarrollo de una nueva estrategia para corregir el traslape en la búsqueda de conformaciones de la secuencia lineal de una proteína.

- El desarrollo y evaluación de los primeros métodos ACO enfocados en el modelo HP triangular, los cuales superan el desempeño de otros métodos presentados en la literatura.

1.5 Organización de la tesis

El resto de este documento está organizado de la siguiente manera. En el capítulo 2 se presentan fundamentos teóricos básicos de las proteínas y del modelo HP, describiendo brevemente sus propiedades biológicas y químicas. En el capítulo 3 se da una explicación sobre la metaheurística ACO, los principios en los que está basada, una introducción al algoritmo general y las principales vertientes que hay de la misma. En el capítulo 4 se exponen algunos de los principales algoritmos que se han desarrollado en torno al problema del plegamiento de proteínas. En el capítulo 5 se presenta la descripción de los métodos propuestos en esta tesis para el problema del plegamiento de proteínas. En el capítulo 6 se describen los experimentos realizados y los resultados obtenidos en la aplicación de los distintos métodos propuestos. Finalmente, en el capítulo 7 se presentan las aportaciones del trabajo y las posibles direcciones futuras que se pueden tomar a partir del mismo.

CAPÍTULO 2

Propiedades de las proteínas

Las proteínas son compuestos orgánicos complejos con una gran importancia biológica debido a que realizan una gran diversidad de funciones en la naturaleza. Además de ser constituyentes esenciales y universales de las células, las proteínas son prácticamente la única fuente con la que el organismo puede reponer el nitrógeno que pierde. Muchas proteínas contienen grupos químicos que son esenciales para la salud porque el organismo no puede sintetizarlos; además, las proteínas pueden aprovecharse para suplir las demandas de energía ordinarias. En los animales, las proteínas contribuyen a la formación de estructuras de sostén y de protección, como huesos, cartílagos, piel, uñas y pelo, y constituyen una gran parte de los sólidos totales del cuerpo.

Este capítulo está dividido en dos secciones. En la primera sección se presenta una descripción general de las proteínas y sus características químicas principales. En la segunda sección se da una descripción del problema del plegamiento de proteínas y

del modelo hidrofóbico-polar (HP), en el que se basa esta tesis para el estudio de dicho problema.

2.1 Propiedades químicas de las proteínas

Las proteínas son polímeros lineales de α-aminoácidos (monómeros) unidos mediante enlaces peptídicos. Las proteínas suelen estar compuestas por unos veinte aminoácidos distintos y, dependiendo de su tamaño, pueden contener varias unidades de cada uno de éstos (cientos o miles), por lo que son casi infinitas las formas en que pueden combinarse los aminoácidos en la molécula de la proteína (Veale et al., 1970). Los aminoácidos son los materiales básicos en la construcción de las proteínas encontrándose en proporciones características y sin periodicidad alguna, pero encadenados en sucesión específica en cada proteína (Cantarow et al., 1965).

Como se muestra en la Fig. 2.1, los aminoácidos tienen una estructura general que incluye un átomo de carbono central llamado carbono-α (C^{α}) al cual están unidos un átomo de hidrógeno (H), un grupo amino (NH_2), un grupo carboxilo (COOH), y una cadena lateral variable (grupo *R*) que es diferente en cada tipo de aminoácido. Los aminoácidos se diferencian entre sí en las distintas estructuras y propiedades físico-químicas de sus grupos *R*. En general, los grupos *R* pueden ser clasificados en tres clases: polares con carga, polares neutros (carentes de carga), e hidrofóbicos. Se dice que los aminoácidos con cadenas laterales polares son hidrofílicos o polares (afines al agua), ya que pueden formar interacciones débiles con las moléculas de agua. Las cadenas laterales de los aminoácidos hidrofílicos contienen átomos electronegativos, como el oxígeno (O) y el nitrógeno (N). Las cadenas laterales de los aminoácidos no polares o hidrofóbicos están formadas principalmente de cadenas hidrocarbonadas. Distintas proteínas tienen estructuras y propiedades diferentes debido a la localización específica de aminoácidos hidrofílicos e hidrofóbicos en cada una de ellas (Branden y Tooze, 1999).

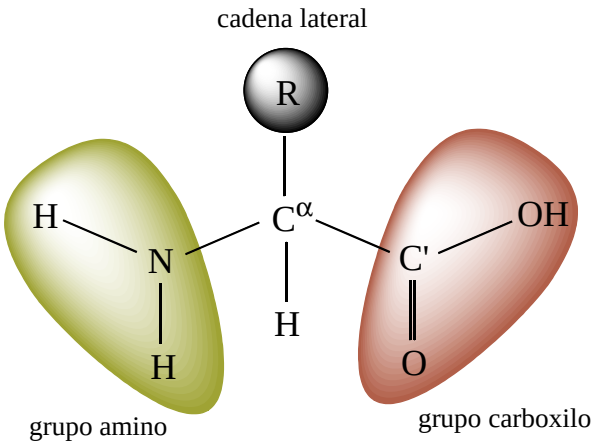


Fig 2.1. Diagrama esquemático de un aminoácido. Un átomo central de carbono (C^{α}) está unido a un grupo amino (NH_2), un grupo carboxilo (COOH), un átomo de hidrógeno (H) y una cadena lateral (*R*) con propiedades variables en los diferentes tipos de aminoácidos.

Los aminoácidos se unen de extremo a extremo durante la síntesis de la proteína mediante la formación de enlaces peptídicos. Se les llama péptidos a compuestos de dos o más aminoácidos unidos por dichos enlaces. Al referirse genéricamente a los péptidos se suele hacer mención del número de aminoácidos componentes. Así, un dipéptido consta de dos aminoácidos, un tripéptido de tres aminoácidos y un oligopéptido, de un número generalmente inferior a 10 aminoácidos. Un polipéptido designa una cadena con un número considerable de aminoácidos (mayor a 10). Se considera que el péptido constituye una molécula de proteína cuando el número de aminoácidos que lo conforman se encuentra entre 50 e incluso más de 25000.

Como ya se mencionó antes, cada aminoácido tiene un grupo amino (NH_2) y un grupo carboxilo (COOH). El enlace peptídico se forma entre el átomo de carbono (C) del grupo carboxilo y el átomo de nitrógeno (N) del grupo amino. Como producto de esta reacción, se libera una molécula de agua. El agua (HOH) se forma a partir del

OH del grupo carboxilo de uno de los aminoácidos y un hidrógeno del grupo NH_2 del otro aminoácido. Este proceso se repite mientras la cadena se expande. Una consecuencia es que el grupo amino del primer aminoácido de un péptido y el grupo carboxilo de su último aminoácido permanecen intactos, y se dice que la cadena se extiende desde su límite amino hasta su límite carboxilo. Por lo tanto, la proteína tiene una polaridad: un grupo amino libre en su lado izquierdo y un grupo carboxilo libre en su lado derecho. La formación de una sucesión de enlaces peptídicos genera una cadena principal o eje principal, desde el cual se proyectan varias cadenas laterales. A los átomos o unidades que forman la cadena principal se les llama residuos, debido a que son precisamente el residuo de las partes comunes de los aminoácidos después de que se forman los enlaces peptídicos. La unidad básica que se repite en la cadena principal desde un punto de vista bioquímico es NH–C^{α}H–C'=O, como se puede ver en la Fig. 2.2.

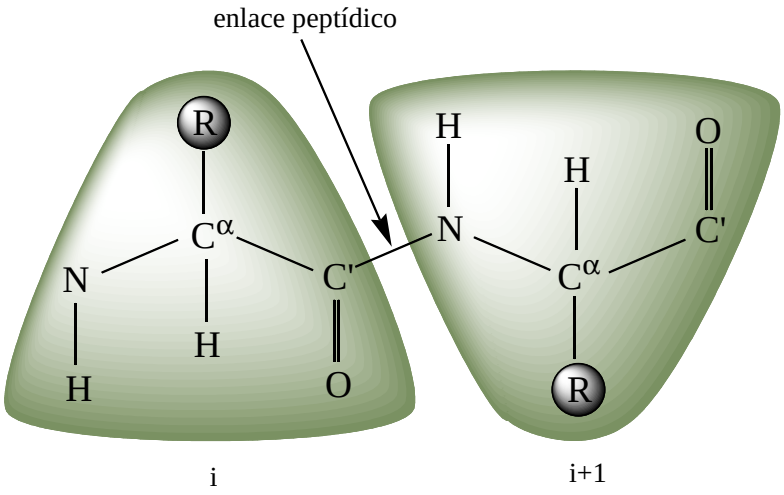


Fig. 2.2. Enlace peptídico (C-N) entre dos aminoácidos contiguos (*i, i+1*) formado entre el grupo carboxilo del aminoácido *i* y el grupo amino del aminoácido *i+1*. En la formación de dicho enlace se libera una molécula de agua. Las unidades que une dicho enlace son llamadas residuos, los cuales contienen un átomo central de C^{α} unido a un grupo NH, un grupo C'=O, y un átomo de H iguales en todos los residuos y una cadena lateral R unida al C^{α} distinta en los diferentes residuos.

El enlace peptídico es la base estructural de las proteínas y presenta una importante serie de propiedades. El enlace peptídico es planar debido a su enlace doble débil y además bastante rígido. Dos formas planares son posibles, una donde la rotación en el enlace peptídico (ángulo ω) se fija en una configuración *cis* (0º), y otra donde se fija en una configuración *trans* (180º) (Darby y Creighton, 1993). La rigidez del enlace peptídico limita las posibilidades conformacionales de los péptidos. Por lo tanto, la cadena polipeptídica tiene libertad rotacional sólo en los enlaces formados por el carbono-α. Las rotaciones en estos enlaces son descritas como ángulos de torsión o diedros, los cuales se consideran usualmente dentro del intervalo de –180º a +180º. La rotación en el enlace C^{α}–N de la cadena principal del polipéptido se denota como el ángulo de torsión *phi* (ϕ) y la rotación en el enlace C^{α}–C se denota como el ángulo de torsión *psi* (ψ). Así, una proteína de longitud *n* tiene *2n* grados de libertad de ángulos de torsión para posicionar a la cadena principal. Los ángulos diedros ϕ y ψ se describen gráficamente en la Fig. 2.3.

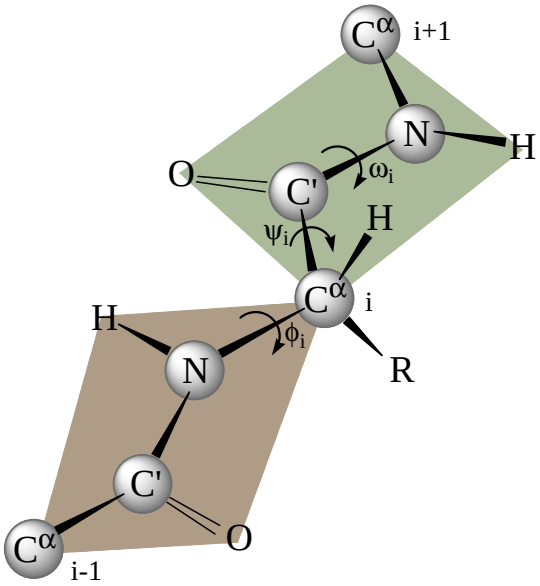


Fig. 2.3. Ángulos de torsión (ϕ, ψ, ω). Los planos rígidos en que están situados dos enlaces peptídicos contiguos pueden girar alrededor del Cα, formando los ángulos φ (en el enlace N-C^{α}) y ψ (en el enlace C^{α}-C'). La conformación de los átomos de la cadena principal está determinada por los valores de estos dos ángulos.

Los valores posibles de los ángulos ϕ y ψ, y por tanto, las conformaciones posibles de la cadena peptídica están limitadas debido a factores estéricos (posición de las moléculas en el espacio) (Macarulla y Goñi, 1993). Los valores permitidos de ϕ y ψ en las proteínas se indican usualmente en un mapa bidimensional del plano ϕ - ψ llamado "diagrama de Ramachandran" (Creighton, 1996) (Fig. 2.4).

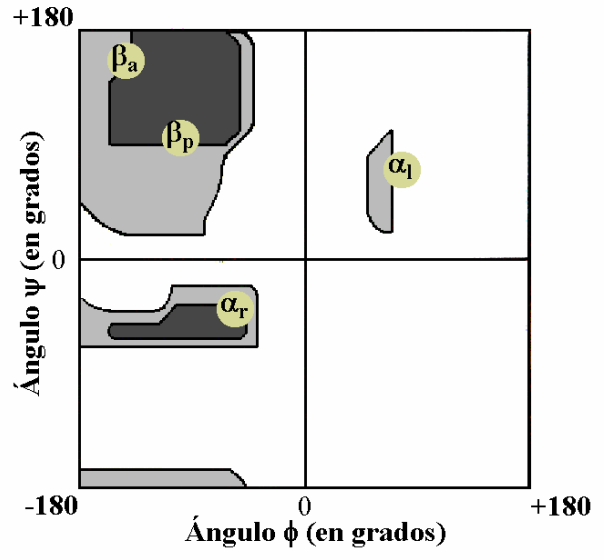


Fig. 2.4. Ejemplo de un diagrama de Ramachandran. Las regiones permitidas se muestran en gris. Las zonas más oscuras corresponden a las zonas con mayor compatibilidad estérica de los ángulos ϕ y ψ. Los círculos representan estructuras secundarias habituales en las proteínas: α_r=hélice derecha, α_l= hélice izquierda, β_p=lámina paralela y β_a=lámina antiparalela.

2.1.1 Estructura de las proteínas

Se llama conformación a los arreglos espaciales de una molécula generados por rotaciones en uniones de enlace simple, mientras que una configuración se refiere a un arreglo específico, característico y estable de átomos o grupos de átomos. De esta manera, en una proteína se llamará configuración al orden de los péptidos específicos que componen la secuencia de dicha proteína, mientras que su conformación indicará la posición en el espacio de cada uno de los péptidos que forman dicha secuencia. Esto es, una conformación será la organización de los átomos que componen de manera tridimensional la estructura de la proteína. Los dos términos, conformación y configuración, no son equivalentes y no deben confundirse.

Con el fin de sistematizar la inmensa variabilidad estructural que pueden presentar las proteínas se suelen distinguir varios niveles de organización. Clásicamente se han descrito cuatro niveles de estructuración de las proteínas: estructuras primaria, secundaria, terciaria y cuaternaria.

La *estructura primaria* se determina por la secuencia de aminoácidos en la cadena polipeptídica lineal, es decir, la especificación del número y tipo de aminoácidos que la integran y el orden en que están ensamblados (Fig. 2.5 a). Las posibilidades de estructuración a nivel primario son prácticamente ilimitadas. Como en casi todas las proteínas existen 20 aminoácidos diferentes, para una secuencia de tamaño *n*, el número de secuencias diferentes posibles está dado matemáticamente por:

$$\text{Secuencias posibles } (n) = 20^n$$

Esto es, el número de variaciones con repetición de 20 elementos tomados de *n* en *n*, siendo *n* el número de aminoácidos por molécula de proteína. Por ejemplo, para una proteína de 160 aminoácidos las posibilidades de secuencia distintas es

aproximadamente de $20^{160} \approx 10^{208}$, es decir, una cifra inmensamente superior al número total de átomos existentes en el universo ($\sim 10^{86}$).

Con esto, el número de posibilidades para secuencias de proteínas de cualquier tamaño está dado por Σ_n (20^n), esto es, la sumatoria del número de secuencias distintas de tamaño n=1, 2, 3… (Macarulla y Goñi, 1993).

La estructura primaria de la proteína determina los niveles superiores de organización, por lo que el conocimiento de la secuencia de aminoácidos es de gran interés para el estudio de la estructura y función de una proteína. La estructura lineal de aminoácidos en una secuencia determinada puede adoptar múltiples conformaciones en el espacio. La conformación de una proteína se analiza en términos de sus estructuras secundaria y terciaria.

La *estructura secundaria* hace referencia a los patrones regulares y repetidos de plegamiento de la cadena principal a nivel local. Los dos tipos de estructura secundaria más comunes son la hélice α y la lámina u hoja β. La hélice α es la conformación regular más común en las proteínas. En una hélice α la cadena principal se enrolla alrededor de un eje imaginario en la dirección de las agujas del reloj. Se obtiene por giro de la cadena en torno a los carbonos α. Las cadenas laterales de los aminoácidos en esta estructura se sitúan en la parte externa de la hélice lo que evita problemas de arreglo espacial de la molécula. La segunda estructura secundaria más común es la lámina β, cuya cadena principal se encuentra casi completamente extendida (Darby y Creighton, 1993) (Fig. 2.5b).

Es frecuente encontrar combinaciones de estructuras al azar, alfa y beta, con una disposición característica, la misma en distintas proteínas. Estas combinaciones han recibido el nombre de *estructuras supersecundarias* (Macarulla y Goñi, 1993).

La *estructura terciaria* (*estructura nativa*) se refiere al plegamiento global de la cadena polipeptídica completa que da lugar a una forma tridimensional específica. La estructura terciaria viene en cierto modo determinada por la secuencia de aminoácidos, puesto que las fuerzas que la estabilizan enlazan las cadenas laterales de los aminoácidos. Los enlaces que determinan la estructura primaria son de tipo covalente (enlaces peptídicos) mientras que la mayoría de los que fijan la conformación son de tipo no covalente. Los enlaces no covalentes son establecidos por fuerzas electrostáticas, por enlaces de hidrógeno adicionales, por enlace hidrofóbico entre cadenas laterales apolares, y por enlace polar, debido a interacciones dipolo-dipolo (Darby y Creighton, 1993). Las proteínas usualmente sólo son biológicamente activas cuando se pliegan en su estructura nativa por lo que la comprensión de sus estructuras tridimensionales es la llave para comprender cómo funcionan (Fig. 2.5c).

La *estructura cuaternaria* describe la forma en que varias moléculas de proteínas se unen entre sí conformando moléculas más grandes. Muchas proteínas están hechas de múltiples cadenas o subunidades polipeptídicas, las cuales pueden no ser idénticas. Las subunidades son, por lo regular, estructuras plegadas de forma independiente que interactúan porque tienen superficies que son complementarias en forma y en interacciones físicas (Macarulla y Goñi, 1993) (Fig. 2.5d).

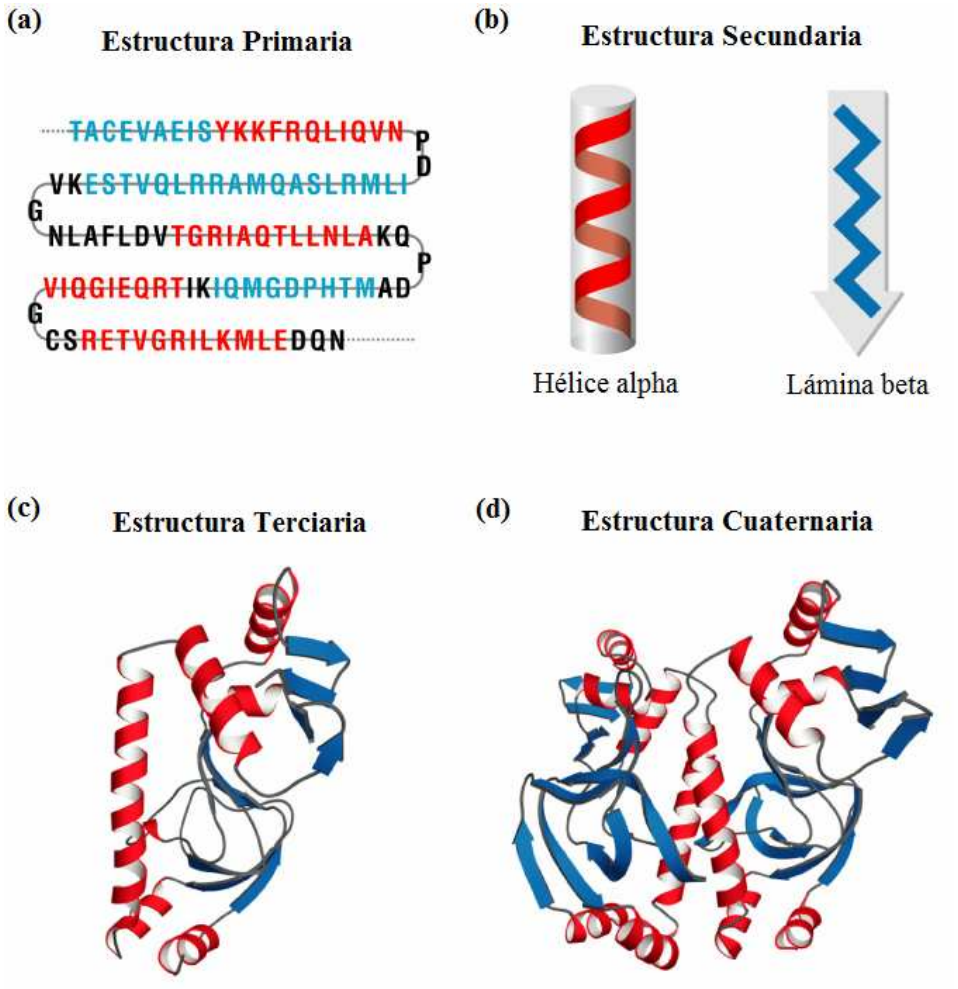


Fig. 2.5. Estructuras de las proteínas: a) Estructura primaria, dada por la cadena lineal de aminoácidos. b) Estructura secundaria, dada por estructuras regulares locales como la hélice α y la hoja β. c) Estructura terciaria, dada por la conformación tridimensional completa de una proteína. d) Estructura cuaternaria, dada por la unión de varias moléculas de proteínas.

2.2 El problema del plegamiento de proteínas

El problema del plegamiento de proteínas consiste en predecir la estructura funcional (estructura terciaria o nativa) de una proteína a partir de su secuencia lineal de aminoácidos. La idea central del estudio de las proteínas es que la estructura primaria de la proteína determina su estructura terciaria y que la función de una proteína está esencialmente determinada por su estructura terciaria (hipótesis termodinámica de Anfinsen (Anfinsen, 1973)). Esto es, las propiedades funcionales de una proteína dependen de su estructura tridimensional. Sin embargo, debido a que el número de conformaciones posibles que una cadena polipeptídica puede adoptar es exponencial y dada la escala de tiempo en la cual una proteína típica se pliega (minutos a milisegundos), para simular el plegado sólo una fracción infinitesimalmente pequeña de estas conformaciones puede ser explorada (Paradoja de Levinthal (Levinthal, 1968)). Por esta razón, la predicción de la estructura de una proteína es un reto fundamental en la biología molecular.

Obtener información experimental acerca de los principios estructurales que rigen el proceso del plegado de las proteínas es una tarea difícil que consume mucho tiempo y recursos. El modelado molecular permite obtener información relevante para ayudar a entender este complicado proceso. Existen varios enfoques en el modelado del plegamiento de proteínas: los métodos *ab initio*, los métodos basados en la energía y los métodos basados en el conocimiento. Los métodos *ab initio* realizan una búsqueda en el conjunto de todas las posibles conformaciones. Debido a que el espacio de conformaciones es exponencial, generalmente se emplean métodos simplificados, métodos determinísticos y búsquedas estocásticas. Los métodos basados en la energía están basados en la minimización de la energía de un sistema definido sobre una malla deformable. Los métodos basados en el conocimiento emplean el modelado por homología y el reconocimiento de diferentes plegados (utilizando bases de datos como el PDB) para buscar un patrón característico en las

proteínas conocidas, y a través de reglas como el potencial de fuerza media, predecir la estructura desconocida de un secuencia dada de aminoácidos.

La mayoría de los métodos *ab initio* emplean principalmente dos enfoques: el "*on-lattice*" (que es el utilizado en esta tesis) y el "*off-lattice*". Dentro del enfoque "*on-lattice*" uno de los modelos más simples y populares es el modelo hidrofóbico-polar (HP) (Dill, 1999). Este modelo se basa en una característica importante del estado nativo de una proteína descrita por la hipótesis termodinámica, la cual establece que el estado nativo de una proteína se encuentra en el estado con energía libre de Gibbs más baja. La energía libre es la energía disponible en un sistema para hacer un trabajo útil. La función de energía libre de Gibbs es una función termodinámica utilizada comúnmente en química para calcular una cantidad de energía libre.

En el modelo HP se supone que la fuerza hidrofóbica es la fuerza primaria en el plegamiento de proteínas. Se afirma que la mayor contribución a la energía libre de una conformación natural de una proteína se debe a interacciones entre aminoácidos hidrofóbicos. Esto es, los aminoácidos con un residuo hidrofóbico (no capaces de formar puentes de hidrógeno) en la secuencia de la proteína, tienden a agruparse y encontrarse en el interior de la proteína formando un núcleo hidrofóbico, mientras que aquellos aminoácidos con un residuo hidrofílico (capaces de formar puentes de hidrógeno) tienden a establecerse en el exterior o superficie de la molécula de la proteína.

En el modelo HP la estructura primaria de una proteína se representa como una cadena de H's y P's que describen los patrones de hidrofobicidad en la secuencia de aminoácidos (Hart y Newman, 2006). Así, un aminoácido hidrofóbico se representa con una H y un aminoácido polar o hidrofílico con una P. Esta cadena se organiza dentro de una malla como un camino en donde cada vértice de la malla sólo puede ser ocupado por un aminoácido a la vez, y el camino no se puede intersectar en ningún

momento. Se establece que los nodos H se atraen mutuamente mientras que los nodos P son neutrales. Debido a esto, se busca encontrar una conformación en la que se maximice el número de contactos H-H, esto es, el número de nodos H adyacentes en la malla que no sean adyacentes en la secuencia lineal. El número de estos contactos se multiplica por una constante, generalmente -1, convirtiendo el problema en el de minimizar la energía libre de las estructuras de malla.

En la Fig. 2.6 se muestran ejemplos de secuencias organizadas dentro de mallas de diferente forma. En la Fig. 2.6a se muestra la secuencia (PPHPPHHPPPPHHPPPPHHPPPPHH) organizada dentro de una malla cuadrada. El número total de residuos adyacentes en la malla cuadrada es de 8, por lo que la conformación tiene un valor de energía libre de -8. De forma similar en la Fig. 2.6b se muestra la secuencia (HHPPHPHPHPHPHP) organizada dentro de una malla triangular bidimensional, la cual tiene una energía libre de -11. En la Fig. 2.6c se muestra la secuencia (PPHPPHHHHHPPHPPHHPPHPPPHHHPHHHHPPP) organizada dentro de una malla cúbica, y con una energía libre de -17.

En el contexto de este modelo, la estructura primaria de la proteína es la secuencia lineal de residuos de aminoácidos hidrofóbicos (H) y polares (P), la estructura secundaria del modelo se puede interpretar como arreglos estructurales comunes, como zigzags y espirales, y la estructura terciaria es la estructura completamente contenida en la malla.

A pesar de la simplicidad del modelo HP, la búsqueda computacional de conformaciones para secuencias muy largas resulta muy cara y ha sido demostrado que es un problema NP-duro (Unger y Moult, 1993c), esto es, que el tiempo que se requiere para resolver una secuencia de este problema crece en el peor de los casos de manera exponencial con respecto al tamaño de la secuencia.

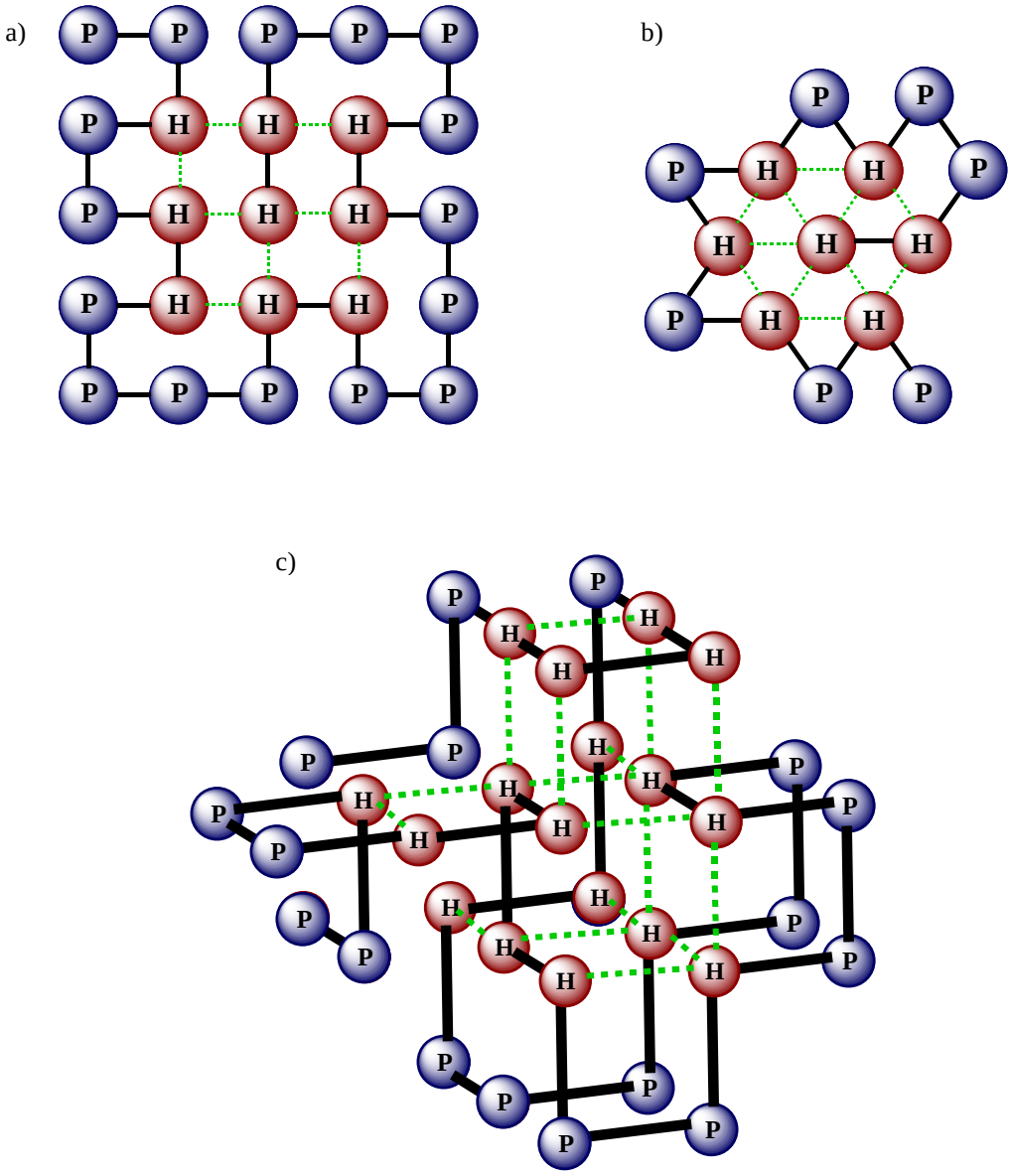


Fig. 2.6. Ejemplos de conformaciones de proteínas utilizando el modelo HP en una malla: a) cuadrada, b) triangular y c) cúbica. Las líneas oscuras muestran los enlaces de la secuencia lineal, mientras que las líneas punteadas verdes muestran los contactos H-H de residuos H adyacentes.

2.3 Resumen del capítulo 2

En este capítulo se expusieron brevemente las principales propiedades que caracterizan a las proteínas, moléculas que desempeñan una gran diversidad de funciones en el organismo de los seres vivos. A pesar de esto, las proteínas tienen una composición estructural relativamente simple. Las proteínas están compuestas por varias subunidades llamadas aminoácidos, las cuales están unidas mediante enlaces peptídicos. En la naturaleza existen alrededor de 20 aminoácidos distintos. Los aminoácidos pueden clasificarse, dependiendo de sus propiedades fisico-químicas, en tres clases: polares con carga, polares neutros e hidrofóbicos. La secuencia lineal de aminoácidos constituye la estructura primaria de la proteína. La estructura secundaria de una proteína está constituida por patrones regulares y repetidos de plegamiento dentro de la cadena de aminoácidos. La forma completa tridimensional de una proteína es lo que se conoce como estructura terciaria o nativa de la proteína, y es precisamente dicha estructura la que determina la función de la proteína. El problema del plegamiento de proteínas consiste en la predicción de la estructura funcional (estructura terciaria) de una proteína a partir de su secuencia lineal de aminoácidos (estructura primaria).

Debido a que el estudio experimental de los principios que rigen el plegamiento de proteínas resulta demasiado complejo, existen modelos simplificados que se han convertido en herramientas útiles para el estudio de las proteínas. Uno de ellos es el modelo hidrofóbico-polar (HP).

En esta tesis se muestra un conjunto de algoritmos basados en ACO aplicados a los modelos HP cuadrado y triangular. En el siguiente capítulo se dan los fundamentos generales de ACO y más adelante se explican con detalle los algoritmos desarrollados en esta tesis.

CAPÍTULO 3

Optimización por colonia de hormigas

Diferentes problemas combinatorios de optimización complejos pueden ser encontrados en diversos campos como son economía, ingeniería, biología o medicina. Este tipo de problemas han sido llamados NP-duros debido a la dificultad para resolverlos mediante las ciencias computacionales. Aun así, se han propuesto varias técnicas para intentar resolverlos. Estas técnicas existentes se pueden clasificar en algoritmos exactos y algoritmos de aproximación. Los algoritmos exactos intentan encontrar una solución óptima y probar que dicha solución es en realidad la óptima; estos algoritmos incluyen técnicas de búsqueda tales como retroceso (*backtracking*), ramificaciones y límites (*branch and bound*), programación dinámica, entre otras. Debido a que estos algoritmos muestran un bajo desempeño para muchos problemas, se han desarrollado varios tipos de algoritmos de aproximación. Los algoritmos de aproximación pueden a su vez dividirse en dos: algoritmos de construcción y algoritmos de búsqueda local. Los primeros se basan en generar soluciones agregando componentes de la solución paso a paso. Las soluciones encontradas por estos algoritmos pueden no ser óptimas con respecto a pequeños cambios locales. Por esta

razón es común intentar mejorar las soluciones así generadas aplicando una búsqueda local. Los algoritmos de búsqueda local intentan mejorar repetidamente la solución actual mediante la realización de movimientos que los conduzcan a una mejor solución dentro de una vecindad (Cordón et al., 2002).

En los últimos 20 años, la investigación en el campo de los problemas combinatorios ha prestado especial atención al diseño de técnicas de propósito general que puedan aplicarse a un amplio conjunto de problemas. Dichas técnicas han sido llamadas *metaheurísticas*. Las metaheurísticas se entienden como métodos, determinísticos o estocásticos, que pueden ser aplicados a diferentes problemas de optimización con relativamente pocas modificaciones. Las metaheurísticas incorporan conceptos de varios campos tales como genética, biología, inteligencia artificial, matemáticas, física, entre otros. Algunos ejemplos de metaheurísticas incluyen recocido simulado (Aarts et al., 1997; Kirkpatrick et al., 1983), búsqueda tabú (Glover y Laguna, 1997), búsqueda local iterativa (Lourenčo et al., 2002), algoritmos de búsqueda de vecindad variable (Hansen y Mladenovic, 1999), y algoritmos evolutivos (Bäck, 1996; Goldberg, 1989; Holland, 1975). Una metaheurística relativamente reciente es la llamada optimización por colonia de hormigas (ACO).

ACO, propuesta por Dorigo (1992), es una metaheurística inspirada en el comportamiento de hormigas reales para resolver problemas combinatorios de optimización. ACO puede ser considerada parte de la inteligencia colectiva (*Swarm Intelligence*) (Bonabeau et al., 1999), la cual toma inspiración del comportamiento social de insectos y otros animales en la naturaleza.

ACO es una metaheurística relativamente joven cuando se compara con otras metaheurísticas como los algoritmos evolutivos, búsqueda tabú o recocido simulado. Aun así, ACO ha mostrado ser un método eficiente y flexible. Ha sido aplicada a diferentes problemas combinatorios de optimización complejos obteniendo resultados

satisfactorios. Con el paso de los años se han presentado diversos trabajos en los que se proponen algunas modificaciones sobre el algoritmo original de ACO para mejorar en cierta medida su funcionamiento, y aún en nuestros días existe una amplia investigación alrededor de esta metaheurística.

En este capítulo se presenta una visión general sobre ACO, desde el fenómeno biológico en el cual fue inspirado hasta sus variantes más populares.

3.1 Hormigas naturales y hormigas artificiales

La creación de ACO fue inspirada por la observación del comportamiento de hormigas reales en la naturaleza. Está basado en el concepto de 'estigmergia', presentado por el entomólogo francés Pierre-Paul Grassé en 1946, y que podemos definir como 'comunicación por medio del ambiente'. Este tipo de comunicación tiene las características de que es una comunicación indirecta y no visual (los insectos intercambian información modificando su ambiente), además de ser local (sólo puede ser percibida por aquellos insectos que pasen por el lugar donde fue liberada por alguno de sus vecinos) (Dorigo et al., 2006). En las colonias de hormigas reales, las hormigas caminan desde y hacia una fuente de alimento depositando en la tierra una sustancia llamada feromona. Otras hormigas perciben la presencia de esta sustancia y tienden a seguir los caminos donde la concentración de la misma es mayor. Cuando las hormigas siguen cierto camino depositando feromona sobre él ocurre un proceso de refuerzo, lo cual provoca la formación de caminos marcados con altas concentraciones de feromona. Este comportamiento permite a las hormigas transportar su alimento hasta su nido de una forma bastante efectiva, recorriendo los caminos más cortos. Además, en el ambiente natural, la feromona depositada por las hormigas se evapora después de un determinado tiempo. De esta forma, los caminos menos prometedores pierden concentración de feromona progresivamente al ser visitados cada vez menos por las hormigas.

ACO está basado fuertemente en esta inspiración biológica. Por este motivo existen varias similitudes y diferencias entre las colonias de hormigas reales y artificiales. Tanto las colonias de hormigas reales como las artificiales están compuestas de un conjunto de individuos que trabajan juntos para resolver alguna tarea. Ambos conjuntos se comunican por medio del ambiente usando feromona. En el caso de las hormigas artificiales, el camino de feromona artificial consiste de valores numéricos asociados con diferentes estados del problema.

Existen varias diferencias importantes entre las hormigas reales y las artificiales. Una de ellas es que a diferencia de las hormigas reales, las hormigas artificiales se desenvuelven en un mundo discreto, moviéndose secuencialmente a través de un conjunto finito de estados de un problema. Otra diferencia es que la actualización de feromona realizada por las hormigas artificiales no es exactamente igual a la de las hormigas reales. Algunas veces sólo una parte de las hormigas artificiales es la que realiza esta actualización, y sólo después de que una solución ha sido construida en su totalidad. Finalmente, algunas implementaciones de hormigas artificiales utilizan mecanismos adicionales que no existen en la naturaleza (como búsqueda local) para mejorar la eficiencia de la búsqueda.

3.2 La metaheurística ACO

Desde el punto de vista computacional, ACO es un algoritmo iterativo de construcción. ACO lleva a cabo un cierto número de iteraciones en las cuales un grupo de hormigas artificiales o agentes computacionales buscan la solución de algún problema. Las hormigas artificiales trabajan cooperativamente a través de auto-organización, esto es, sin ninguna forma de control central que actúe sobre ellas.

La metaheurística ACO se muestra en el algoritmo 3.1. En la inicialización de parámetros se establecen el número de hormigas en la colonia, los pesos que definen el balance entre la información heurística y la información de la feromona, y otros parámetros propios del problema a tratar, como por ejemplo, la condición de término del algoritmo. En la inicialización de los caminos de feromona se estable el valor numérico inicial asociado a todos los posibles pasos de los caminos que las hormigas pueden seguir en la construcción de soluciones. Después esta inicialización, la metaheurística itera sobre tres fases: en cada iteración, un número de soluciones son construidas por las hormigas (*HormigasConstruyenSoluciones*). Estas soluciones son mejoradas a través de una búsqueda local (*AplicarBúsquedaLocal*), y finalmente los valores de la feromona son actualizados (*ActualizarFeromona*).

Algoritmo 3.1. La metaheurística ACO.

Establecer parámetros, inicializar caminos de feromona
mientras (no se cumpla la condición de término) *hacer*
    HormigasConstruyenSoluciones
    AplicarBúsquedaLocal {opcional}
    ActualizarFeromona
fin_mientras

Cada paso en la fase de construcción de soluciones de las hormigas es guiado probabilísticamente usando dos tipos de información: información heurística (la cual depende del problema) e información basada en los caminos de feromona artificial (valores numéricos). La información heurística mide el beneficio que otorga utilizar un determinado movimiento a la solución que se está construyendo. Dicha información heurística no es modificada por las hormigas durante la ejecución del algoritmo. La información de la feromona artificial mide qué tan factible ha sido un determinado movimiento en iteraciones pasadas. La información de la feromona

artificial es modificada por las hormigas dependiendo de las soluciones encontradas por las mismas.

Una vez que las soluciones han sido construidas, y antes de actualizar la feromona, es común mejorar las soluciones obtenidas por las hormigas aplicando una búsqueda local. Esta fase depende altamente del problema que se esté solucionando y es opcional.

La última fase de la metaheurística ACO es la actualización de feromona. Esta fase tiene dos propósitos. El primero es disminuir aquellos valores que estén asociados con las soluciones no prometedoras. Esto se realiza por medio de la evaporación de la feromona: todos los valores de la feromona son disminuidos por un cierto valor *p*. El segundo objetivo es reforzar los valores de la feromona asociados con las soluciones buenas o prometedoras. Esto se realiza incrementando los valores de la feromona asociados con un conjunto elegido de las mejores soluciones (Dorigo y Socha, 2007).

3.3 Variantes principales de ACO

Diversas variantes de ACO han sido propuestas en la literatura. La primera implementación de ACO fue *Ant System* (AS) (Dorigo, 1992; Dorigo et al., 1996). Su característica principal fue que los valores de la feromona eran actualizados por todas las hormigas que hubieran logrado completar su solución. La primera y más conocida aplicación de AS fue en el problema del agente viajero (TSP) (Dorigo, 1992; Dorigo et al., 1996). Después fue utilizado en otros problemas: el problema de asignación cuadrática (AS-QAP) (Maniezzo y Colorni, 1999), el problema de planificación de trabajo-comercio (JSP) (Colorni et al., 1994), el problema de enrutamiento de vehículos (VRP) (Bullnheimer et al., 1998) y el problema de supersecuencia común más corta (SCS) (Michel y Middendorf, 1999).

MAX-MIN Ant System (MMAS) fue presentado como una mejora sobre la idea original de AS. Fue propuesto por Stützle y Hoos (2000) y presentó los siguientes cambios: sólo la mejor hormiga podía actualizar los caminos de feromona y los valores máximo y mínimo que la feromona podía tener estaban limitados. MMAS presentó una mejora significativa sobre el desempeño básico de AS. Aunque las primeras implementaciones de este algoritmo se enfocaron al TSP (Stützle y Hoos, 2000), más tarde fue aplicado a otros problemas combinatorios como el problema del horario de cursos universitarios (UCTP) (Socha et al., 2004), el problema de asignación generalizada (GAP) (Lourenčo y Serra, 2002), y el problema del recorrido de conjuntos (SCP) (Lessing et al., 2004).

Ant Colony System (ACS), presentado por Dorigo y Gambardella (1997), fue otra mejora hecha sobre el original AS. La mayor contribución de ACS fue la introducción de una actualización local de feromona conjuntamente con la actualización de feromona realizada al final del proceso de construcción (actualización de feromona global). Esta actualización local era realizada por todas las hormigas después de cada paso en la construcción de soluciones y su objetivo consistía en diversificar la búsqueda realizada por las hormigas. ACS fue aplicado a problemas como TSP (Dorigo y Gambardella, 1997), VRP (Bianchi et al., 2004) y UCTP (Socha et al., 2003).

Otras variantes que han surgido de ACO se listan a continuación:

Rank-based Ant System (AS_{rank}), propuesto por Bullnheimer et al. (1999), incorpora la idea de clasificación jerárquica (*ranking*) para la actualización de feromona. Mientras mejor sea la solución encontrada por una hormiga, mayor será el incremento que se le aplique al camino de feromona asociado con dicha hormiga.

Best-Worst Ant System (BWAS), propuesto por Cordón et al. (2000), incorpora conceptos de computación evolutiva incluyendo una fase en la cual se realiza una

mutación de los caminos de feromona. En la actualización de feromona, además de realizar el incremento a los caminos de las mejores hormigas y aplicar la evaporación de feromona, también se realiza un decremento en los valores de los caminos de las hormigas que encontraron las peores soluciones.

Hyper-cube ACO (HC-ACO), propuesto por Blum y Dorigo (2004), tiene como idea principal la normalización de los valores de feromona.

ACO basado en población (PB-ACO), propuesto por Guntsch y Middendorf (2002), memoriza las soluciones utilizadas para incrementar los valores de la feromona (el conjunto de estas soluciones es llamado población), y una vez que dicha población ha alcanzado una dimensión máxima, los valores de feromona asociados a los peores caminos son 'eliminados' aplicando una actualización negativa de esta feromona.

La investigación alrededor de ACO es muy extensa. Incluye la aplicación de ACO a nuevos tipos de problemas como: problemas de optimización dinámica, optimización multiobjetivo, problemas estocásticos, entre otros. Además, también se ha explorado la implementación de ACO en ambientes paralelos. También se han desarrollado técnicas híbridas combinando ideas de ACO con algoritmos exactos y otras metaheurísticas.

3.4 Problemas que se presentan al resolver un problema aplicando ACO

Abordar un problema en específico utilizando alguna implementación de ACO implica varios aspectos a tomar en cuenta, como los enlistados a continuación:

- Elegir una correcta representación de los componentes que definen el problema, como son los diferentes estados por los que las hormigas viajan para construir soluciones.

- Manipular y modificar correctamente la información proporcionada por los valores de la feromona. Existen varios enfoques, por ejemplo, para realizar la actualización de feromona se debe determinar qué hormigas van a modificar los caminos de feromona y en qué cantidad realizaran dichas modificaciones.

- Elegir una heurística apropiada para el problema a resolver. La información heurística es crucial en el desempeño del algoritmo, ya que guía las decisiones de las hormigas en el momento de construir las soluciones. Además, utilizar diferentes heurísticas en un mismo problema nos puede conducir a resultados muy distintos entre sí.

- Elegir una técnica para mejorar las soluciones encontradas por las hormigas, por ejemplo, una búsqueda local.

- Afinar los parámetros del algoritmo ACO para obtener buenos resultados.

A partir de lo anterior, podemos decir que para un mismo problema se pueden implementar diferentes enfoques de ACO obteniendo para cada uno de ellos resultados variables. En específico para el problema del plegamiento de proteínas se han implementado varios enfoques de ACO variando, sobre todo, en el tipo de heurística utilizada y en el esquema de actualización de feromona. En esta tesis se muestra cómo el uso de diferentes esquemas puede conducir a resultados variables para diferentes secuencias del problema. A partir de lo anterior, se proponen tres formas diferentes de combinar estos esquemas y generar algoritmos ACO híbridos compuestos de varias especies de hormigas. Estos algoritmos se explican con detalle en el capítulo 5.

3.5 Resumen del capítulo 3

En este capítulo se mostró brevemente el funcionamiento básico de un algoritmo ACO. Los algoritmos ACO son metaheurísticas basadas en el comportamiento de las hormigas en la naturaleza. En los algoritmos ACO un conjunto de agentes llamados hormigas construyen de forma iterativa soluciones a un determinado problema basando sus decisiones en dos fuentes de información principales: una función heurística y los valores almacenados de feromona para cada paso en la construcción de soluciones. A través de los años se han propuesto diversas modificaciones realizadas al algoritmo ACO original propuesto en 1992. Además, se han propuesto varias aplicaciones de algoritmos ACO para diferentes problemas combinatorios de optimización complejos.

En el siguiente capítulo se presentan diversas propuestas presentadas en la literatura que se han planteado para resolver el problema del plegamiento de proteínas, incluyendo algunas implementaciones de ACO para este problema. Se discute el desempeño de estos algoritmos y los resultados que reportan, así como algunas ventajas y desventajas que presentan.

CAPÍTULO 4

ESTADO DEL ARTE

El problema del plegamiento de proteínas, que consiste en la predicción de la estructura de una proteína a partir de su secuencia lineal de aminoácidos, ha sido objeto de estudio desde hace ya más de 50 años en el campo de la biología molecular y en años más recientes, en el diseño de algoritmos. A finales de los 40's e inicios de los 50's comenzó la revolución en la biología molecular moderna. Los primeros experimentos fueron llevados a cabo sólo poco tiempo después de que Sanger (1952) hubiera determinado la secuencia de la insulina y probado que la estructura covalente básica de la proteína es una cadena peptídica lineal y que la secuencia de aminoácidos de la cadena es única para una proteína dada. Hace ya casi medio siglo, Linus Pauling descubrió dos arreglos simples y regulares de aminoácidos: la hélice-α y la hoja-β, que se encuentran en casi toda proteína. Estos descubrimientos le valieron el Premio Nobel en 1954. En los inicios de los 60's, Christian Anfinsen mostró que si una proteína se despliega, entonces se pliega de nuevo en una forma adecuada a su propia naturaleza (Zagrovic et al., 2002).

El uso de técnicas computacionales para obtener la estructura terciaria de una proteína a partir de su estructura primaria resulta muy atractivo. Esto es debido a que la determinación de las estructuras de la proteínas mediante el uso de técnicas experimentales como cristalografía de rayos-X y RMN (Resonancia magnética nuclear) (Sikder y Zomaya, 2005) implican un gran costo en cuanto a tiempo y equipo y un pre-procesamiento adicional de la proteína.

Los modelos de malla (*on-lattice*) tienen ya una larga historia en el modelado de polímeros debido a su simplicidad analítica y computacional. A pesar de su simplicidad, el modelo HP (hidrofóbico-polar) es suficientemente eficaz para capturar una gran variedad de propiedades de las proteínas actuales. A través de los años varios algoritmos de optimización conocidos han sido aplicados para resolver el problema del plegamiento de proteínas. Entre estos métodos encontramos los algoritmos de Monte Carlo y Recocido Simulado, búsqueda Tabú, los algoritmos genéticos y, más recientemente, algoritmos de optimización por colonia de hormigas.

En la primera sección de este capítulo se da una breve descripción de las técnicas experimentales utilizadas para estudiar las proteínas. En las demás secciones se da una breve reseña de los algoritmos más significativos que han sido aplicados al problema del plegamiento de proteínas, y más específicamente al modelo HP.

4.1 Técnicas experimentales para determinar la estructura de las proteínas

El arreglo estructural o espacial de las proteínas es complejo y su determinación requiere de una considerable cantidad de datos experimentales. Por medio de la técnica de cristalografía de rayos-X es posible determinar la estructura exacta de una proteína. Otra técnica que puede generar estructuras menos exactas que la

cristalografía de rayos-X es la técnica RMN (Resonancia Magnética Nuclear). En las siguientes secciones se da una explicación más detallada de estas dos técnicas.

4.1.1 Cristalografía de rayos-X

La técnica de cristalografía de rayos-X ha proporcionado las bases experimentales del conocimiento actual de la estructura de las proteínas. Los cristales son una forma sólida de una sustancia en la cual las moléculas componentes se presentan en un arreglo ordenado llamado malla. El bloque básico de construcción de un cristal se llama celda unitaria. Cada celda unitaria contiene el conjunto más pequeño posible de componentes que son representativos del cristal, siendo dicho conjunto único. Los cristales de una molécula compleja, como las proteínas, producen patrones de difracción de rayos-X complejos. Cuando el cristal se coloca en el haz de rayos-X, todas las celdas unitarias presentan la misma cara al haz, por lo tanto muchas moléculas están en la misma orientación con respecto a los rayos-X entrantes. El haz de rayos-X penetra en el cristal y un número de pequeños haces emerge: cada uno en una dirección e intensidad diferentes. Este conjunto de haces difractados contiene información acerca de la estructura del cristal original. El mayor inconveniente asociado con esta técnica es que la cristalización de las proteínas es una tarea difícil. Los cristales se forman lentamente precipitando las proteínas bajo condiciones que mantienen su conformación o estructura nativa. Estas condiciones exactas sólo pueden ser descubiertas por experimentos repetitivos que requieren la variación de ciertas condiciones experimentales, una a la vez. Este es un proceso muy lento y tedioso.

4.1.2 Espectroscopia de resonancia magnética nuclear

En esta técnica se introduce una muestra dentro de un campo magnético y se bombardea con ondas de radio. Estas ondas de radio provocan que el núcleo de la molécula vibre o gire (*spin*). El núcleo vibratorio emite una señal única, la cual se recoge con un receptor de radio especial y se traduce usando una transformada de Fourier. Midiendo las frecuencias en las que diferentes núcleos giran, los científicos pueden determinar la estructura molecular, así como otras propiedades interesantes de la molécula. Los principios en los que se basa la espectroscopia de resonancia magnética nuclear hacen a esta técnica muy lenta y limitan la aplicación a moléculas pequeñas y de mediano tamaño.

4.2 Métodos de Monte Carlo

Los métodos de Monte Carlo y de Recocido Simulado se utilizan para escenarios complejos de búsqueda y muestreo. Estos algoritmos generan una trayectoria de estados para un sistema. Cada estado se evalúa utilizando una función objetivo (energía potencial). Si el valor de la función objetivo disminuye, se acepta un nuevo estado. Si el valor de la función incrementa, se acepta el nuevo estado con una probabilidad calculada con la función de probabilidad de Boltzmann dada por: $P(c) = e^{-c/kT}$, donde P(c) indica la probabilidad de un cambio c en la función objetivo, k es una constante elegida según el problema y T es una variable llamada temperatura artificial. Mientras que en los métodos Monte Carlo el valor de T se mantiene constante, en Recocido Simulado el valor de T se varía conforme avanza el algoritmo (generalmente de forma decreciente).

Entre los mejores algoritmos conocidos para el problema del plegamiento de proteínas se encuentran varios algoritmos de Monte Carlo, incluyendo PERM

(*Pruned-Enriched Rosenbluth Method*) de Bastolla et al. (1998). PERM es un algoritmo sistemático de crecimiento de cadena que evalúa conformaciones plegadas parcialmente y crea copias de aquellas conformaciones parciales que tengan un alto peso estadístico (enriquecimiento basado en la energía alcanzada y el tamaño del plegado), y elimina conformaciones parciales con un bajo peso (podado). Después de completar el plegado de la conformación actual, PERM se mueve a la siguiente conformación parcial que haya sido generada durante el enriquecimiento o comienza una nueva cadena.

Zhang y Liu (2002) presentaron un algoritmo de muestreo que es muy similar a PERM excepto que los pesos estadísticos para podado y enriquecimiento son calculados con base en la simulación de un número de cadenas en paralelo. Su algoritmo es inferior a PERM en términos de desempeño.

Otros métodos Monte Carlo (menos exitosos que PERM) desarrollados para atacar este problema incluyen el algoritmo dinámico de Monte Carlo de Ramakrishnan et al. (1997) basado en un movimiento de cambio de cuatro ciclos que desconecta la cadena. Liang y Wong (2001) presentaron un algoritmo Monte Carlo evolutivo (EMC) que trabaja con una población de individuos, donde cada individuo ejecuta una optimización de Monte Carlo. También implementaron una variante de EMC que refuerza ciertas estructuras secundarias (hélices alpha y hojas beta). Chikenji et al. (1999) presentaron el método de Monte Carlo de ensamble de multi-auto-traslape (MSOE), el cual considera configuraciones de cadena traslapadas.

Un algoritmo que ha mostrado ser de los más eficientes en el problema del plegamiento de proteínas es PERM, el cual trabaja con conformaciones parcialmente plegadas, y mantiene en cierta forma un registro de las conformaciones que ha revisado. Otros algoritmos presentados en esta sección realizan la búsqueda de soluciones tomando decisiones sobre conformaciones completamente plegadas y basan su búsqueda sólo en conformaciones generadas a partir de las conformaciones

presentes en la iteración actual. Este tipo de algoritmos se basa en un muestreo y no en construcción de conformaciones. En PERM se usa más el enfoque de construcción al utilizar conformaciones parciales, lo que muestra la eficiencia de este enfoque (similar al usado en ACO). Sin embargo, a pesar de la efectividad que muestran los métodos Monte Carlo para el problema del plegamiento de proteínas, son métodos difíciles de implementar, además de que se requiere una comprensión más profunda del problema para simularlo correctamente.

4.3 Algoritmos genéticos

De manera general, los algoritmos genéticos son métodos de optimización probabilística basados en los principios de la evolución (Bodenhofer, 2003). En los algoritmos genéticos se utiliza el término de cromosoma para referirse a una representación de alguna solución probable al problema de optimización. Una cierta posición en el cromosoma se denomina gen. A un conjunto de cromosomas en un determinado tiempo se le denomina población. Los algoritmos genéticos (AG) optimizan la población de soluciones usando operaciones inspiradas biológicamente de mutación y de recombinación entre pares de soluciones (*crossover*) de forma iterativa. Se utiliza una función de aptitud, esto es, una función objetivo que se busca maximizar en la solución final, para evaluar la calidad de cada cromosoma, y así seleccionar en cada iteración ciertos cromosomas que se utilizarán en la siguiente generación.

J. H. Holland es considerado como el pionero de los algoritmos genéticos. Desde que Holland desarrolló la idea de los AG en 1967, este campo ha sido ampliamente estudiado generando varias aplicaciones en aprendizaje computacional y optimización. Durante los últimos años, los AG han sido aplicados extensamente al problema del plegamiento de proteínas con modelos reducidos. Un equipo precursor en el campo fue el de Unger y Moult (1993a; 1993b). En su trabajo, ellos adoptaron

el modelo de Dill et al. (1993). Estos autores usaron un algoritmo que busca conformaciones representadas por direcciones relativas. Ellos presentaron un AG no estándar incorporando características de Recocido Simulado. En la operación de mutación de su algoritmo, estos autores incorporaron un paso de decisión estocástica en el cual dada una conformación con energía E_1, se le aplica un cambio aleatorio generando una nueva conformación con energía E_2. La nueva conformación es aceptada si $E_2>E_1$. En caso contrario la nueva conformación se acepta con una probabilidad dependiente del número de la iteración actual. También, una vez que se ha efectuado la operación de recombinación, las nuevas conformaciones generadas son aceptadas utilizando el criterio mencionado arriba.

Patton y Goldman (1995) presentaron un AG estándar que sobrepasa el desempeño del algoritmo presentado en (Unger y Moult, 1993b), en el sentido de que dada una cierta secuencia y un nivel de energía, su algoritmo consigue encontrar una conformación con dicho nivel de energía en una menor cantidad de evaluaciones. Además, en algunos casos, estos autores consiguieron encontrar mejores conformaciones.

Krasnogor et al. (1998) implementaron otro AG en el cual argumentan haber combinado las características más deficientes de los algoritmos presentados en (Unger y Moult, 1993b) y (Patton y Goldman, 1995) y haber conseguido un algoritmo competitivo con ambos algoritmos. En el trabajo de Krasnogor et al. encuentran conformaciones mejores a las de (Unger y Moult, 1993b) en un tiempo similar al trabajo en (Patton y Goldman, 1995).

En general, los AG aplicados al problema del plegamiento de proteínas han resultado ser menos exitosos que otros métodos como los de Monte Carlo. Los algoritmos genéticos construyen diferentes soluciones a un problema de forma iterativa. En cada iteración se genera una nueva población de individuos a partir de la aplicación de las operaciones de mutación y recombinación a los individuos de la

población generada en la iteración anterior a la actual. Debido a esto, el desempeño de los algoritmos genéticos depende fuertemente de las operaciones de mutación y recombinación definidas para el problema. Estas operaciones suelen trabajar de forma aleatoria. Debido a esto, los algoritmos genéticos no tienen realmente una guía subyacente en la búsqueda y construcción de soluciones por lo que, dependiendo de la complejidad del problema, pueden tener un elevado tiempo de ejecución.

4.4 Optimización por colonia de hormigas

En los últimos años, los algoritmos ACO han sido utilizados para resolver el problema del plegamiento de proteínas simplificado con el modelo HP en dos (HP-2D o cuadrado) y tres dimensiones (HP-3D ó cúbico).

La primera aplicación de ACO para el problema del plegamiento de proteínas fue presentada en el trabajo de Shmygelska et al. (2002). Esta aplicación trabaja con el modelo HP-2D cuadrado. En este algoritmo ACO, las hormigas construyen conformaciones candidatas para secuencias lineales de aminoácidos. Las conformaciones candidatas se representan con arreglos de estructuras locales (o direcciones de plegamiento relativas) las cuales indican la posición de cada aminoácido en la malla 2D relativa a sus predecesores directos en la secuencia dada.

En la fase de construcción, cada hormiga elige de forma aleatoria un punto de inicio p dentro de la secuencia de tamaño n dada. Desde esta posición p, cada hormiga construye la conformación parcial s_p, s_{p-1},..., s_2, s_1, y después la conformación parcial s_p, s_{p+1},..., s_{n-1}, s_n, dando como resultado la conformación de la secuencia completa s_1,..., s_n. Para elegir una dirección relativa en la construcción de las conformaciones se utilizan los valores de la heurística y de la feromona. Los valores de la heurística están dados por $\eta_{i,d} = (h_{i,d} + 1)$, donde $h_{i,d}$ se refiere al número

de nuevos contactos H-H obtenidos cuando se coloca el aminoácido s_{i+1} en la dirección relativa *d* a s_i y s_{i-1}.

Para resolver el problema del traslape, esto es, cuando se llega a un punto en donde todas las posibles direcciones que un aminoácido puede tomar ya están ocupadas por otro aminoácido, se realiza lo siguiente: primero, no se permite que un aminoácido sea colocado en un lugar en el cual todas las posiciones vecinas en la malla ya estén ocupadas. Si se encuentra un estado como el anterior entonces se deshace la mitad de la conformación construida hasta ese momento y se reinicia el proceso de construcción desde la posición correspondiente de la secuencia.

De forma adicional en la búsqueda de soluciones, después de la fase de construcción se realiza una fase de búsqueda local a las conformaciones construidas por las hormigas. Para este proceso se consideran dos tipos de vecindades. En la primera se cambian aleatoriamente todos los arreglos de estructura locales entre dos determinadas posiciones de la secuencia elegidas aleatoriamente. Todos los cambios se realizan de forma tal que las conformaciones resultantes sean válidas. Para la segunda vecindad se visitan todas las posiciones de la secuencia de forma aleatoria y se consideran las conformaciones que puedan ser formadas cambiando la dirección relativa de una sola posición.

Para prevenir estancamiento en la búsqueda de soluciones se realiza una normalización de los valores de la feromona de forma similar al método utilizado en *MAX-MIN Ant System*. Si para una posición *i* de la secuencia, la razón entre el valor mínimo (min_i) y el valor máximo (max_i) de la feromona cae bajo un umbral θ, el valor mínimo se establece en $(max_i)(\theta)$ y el valor máximo se establece en $max_i - (max_i)(\theta)$. Los resultados presentados el trabajo de Shmygelska muestran un buen desempeño del algoritmo para encontrar soluciones para las secuencias de proteínas más pequeñas (hasta 25 aminoácidos), mientras que para las secuencias más grandes

(mayores a 36 aminoácidos y hasta 64) el desempeño comenzó a degradarse, no encontrando en la mayoría de los casos la solución óptima.

En el año 2003 Shmygelska y Hoos (2003) presentaron un algoritmo mejorado que incluye algunas modificaciones sobre todo con respecto a la búsqueda local. Aquí, las hormigas construyen sus soluciones comenzando desde un punto aleatorio inicial. El lado a extender se elige con una probabilidad igual al número de residuos sin plegar al lado concerniente entre la suma del número de aminoácidos sin plegar en ambos lados. En la búsqueda local se introdujeron unos movimientos llamados "movimientos de intervalo largo" (*long range moves*), una búsqueda local selectiva, y hormigas perfeccionadoras que realizan una mejora iterativa probabilística a la mejor conformación encontrada por el algoritmo desde el inicio del mismo. Los movimientos de intervalo largo se ejecutan, primero, eligiendo aleatoriamente una posición en la secuencia desde la cual se realizará el movimiento. Después se modifica aleatoriamente la dirección del aminoácido elegido y se ajusta la dirección de los aminoácidos restantes. Si ya no es posible colocar un aminoácido en la misma posición que tenía antes del movimiento referido, entonces se elige probabilísticamente una nueva dirección utilizando los valores de la función heurística. Debido a que estos movimientos son muy caros computacionalmente, la búsqueda local sólo se aplica a las mejores conformaciones construidas en una iteración dada del algoritmo (búsqueda local selectiva). En este trabajo de Shmygelska y Hoos se reportan resultados que sobrepasan visiblemente a los resultados del trabajo anterior de los mismos autores, consiguiendo encontrar conformaciones con energía óptima para las secuencias menores a 64 aminoácidos el 100% de las veces. Además, se presentan resultados de secuencias de hasta 100 aminoácidos de longitud, para las cuales las conformaciones tienen energías cercanas a la óptima. El tiempo de ejecución reportado para encontrar dichas soluciones es, en el peor de los casos, de hasta 9 horas.

En el año 2005, Shmygelska y Hoos (2005) ampliaron su algoritmo para el modelo HP-3D cúbico. En su trabajo, Shmygelska y Hoos presentaron ligeras variaciones en cuanto a la búsqueda local del mismo. Esta búsqueda local es un simple procedimiento de búsqueda iterativa de primero el mejor, y utiliza los mismos operadores de movimiento presentados en un trabajo anterior desarrollado también por los mismos autores. También, ellos utilizaron una función heurística diferente, la cual está dada por: $\eta_{i,d} = e^{(-\gamma)(h_{i,d})}$, donde $-\gamma$ es un parámetro llamado temperatura inversa (como en (Hsu et al., 2003)), y $h_{i,d}$ es el número de nuevos contactos H-H obtenidos cuando se coloca el aminoácido i en la posición especificada por la dirección relativa d. En los experimentos que realizaron Shmygelska y Hoos se utilizó una mayor cantidad de hormigas. Los resultados que ellos obtuvieron muestran tiempos de ejecución de hasta 1 día para el modelo HP-2D y de 720 minutos para el modelo HP-3D. En cuanto a las soluciones encontradas, los autores reportan conformaciones óptimas para secuencias de hasta 85 aminoácidos y subóptimas para secuencias de 100 aminoácidos (modelo HP-2D) y conformaciones óptimas para todas las secuencias (de 48 aminoácidos) en el modelo HP-3D. Aunque este algoritmo se puede considerar como el algoritmo ACO para el plegamiento de proteínas más significativo en la literatura, aún no supera del todo a las soluciones encontradas por PERM.

Otros algoritmos desarrollados con ACO se presentan en los trabajos de Fidanova (2006), Song et al. (2006) y Chu et al. (2005), los cuales se centran en el modelo HP-3D cúbico. En el trabajo de Fidanova (2006), durante la fase de construcción, las hormigas construyen una conformación de una proteína iniciando desde el extremo izquierdo de la secuencia colocando un aminoácido a la vez. Si el aminoácido es de tipo *H*, la posición del aminoácido se elige en base a dos fuentes de información: los valores de la matriz de feromonas y la información heurística, la cual está dada por el número de nuevos contactos H-H que se forman al colocar el aminoácido correspondiente en la conformación. Cuando el aminoácido que se va a colocar es de tipo *P*, entonces se elige la dirección de plegamiento de manera

aleatoria. El algoritmo de Fidanova realiza la actualización de feromona en dos fases: una actualización local que se realiza al mismo tiempo que se construyen las conformaciones y una actualización global realizada al final de cada iteración. En el trabajo de Fidanova se presentan resultados para las secuencias en 3D que sobrepasan en cuanto a nivel de energía encontrada a otros métodos encontrados en la literatura.

En el trabajo de Song et al. (2006) se utiliza un conjunto de matrices que contienen la información necesaria para determinar las diferentes direcciones para construir las conformaciones 3D de las secuencias de proteínas. La información heurística que se utiliza es la misma que en (Shmygelska et al., 2002). Se realiza sólo una actualización global de los valores de la feromona al final de cada iteración. Para evitar el estancamiento se utilizan dos umbrales como valores mínimo y máximo que limitan los valores que la feromonas puede tener. Para tratar con el problema de traslape en la construcción de la estructura, se propone un método basado en CSMA/CD (Acceso Múltiple con Detección de Portadora y Detección de Colisiones) utilizado para resolver problemas de colisiones cuando dos computadoras envían información al mismo tiempo en una red Ethernet. Este método consiste en que cuando se llega a un punto donde no se puede seguir extendiendo la conformación, primero se despliegan dos residuos. Si esto no soluciona el problema, entonces se despliegan 4 residuos; si continúa el problema, entonces se despliegan 8 residuos, y si continúa el problema se despliegan otros 8 residuos. Si con esto no se puede solucionar el traslape, entonces se comienza la construcción de la conformación otra vez desde el punto inicial. En el algoritmo de Song et al. se realiza una fase de búsqueda local utilizando tres operadores, los cuales consisten en la realización de rotaciones en distintos puntos de la estructura de manera que se mejore el resultado en la función de energía global. En el trabajo de estos autores se reportan resultados para las secuencias 3D con nivel de energía igual a los de (Shmygelska y Hoos, 2005).

En el trabajo de Chu et al. (2005), se propone una implementación del algoritmo de ACO utilizando paralelismo. Ellos realizaron cuatro implementaciones de su algoritmo ACO-3D. El primer algoritmo es una implementación de referencia con un solo procesador, una sola colonia y sólo una matriz de feromonas. El segundo algoritmo utiliza una matriz global para todas las colonias. El tercer algoritmo y el cuarto algoritmo utilizan una matriz de feromonas para cada colonia de hormigas. Estos dos últimos algoritmos difieren en la forma de actualizar la matriz de feromonas de cada colonia de hormigas. Chu et al. reportan mejoras en cuanto al tiempo de ejecución global debido a la paralelización. Sin embargo, en el trabajo de estos autores no se reportaron soluciones óptimas en todos los casos.

Los algoritmos ACO presentados en esta sección suelen ser apoyados en el uso de búsquedas locales para asegurar su buen funcionamiento y salir de mínimos locales, lo que puede provocar un aumento considerable en el tiempo de ejecución si esta búsqueda es complicada. En esta tesis se propone un método para evitar el uso de búsquedas locales, o al menos reducir su uso. En los algoritmos mostrados en esta tesis se utilizan dos heurísticas utilizadas en los trabajos de (Fidanova, 2006) y (Shmygelska y Hoos, 2005) debido a que sus resultados son de los más significativos encontrados en la literatura.

De forma particular, a partir de la comparación del último trabajo ACO presentado en esta sección (Chu et al., 2005), con los algoritmos híbridos presentados en esta tesis, se observa que el primero se basa en la ejecución de una colonia de hormigas (o copias de la misma) en diferentes procesadores y el uso de una o varias matrices de feromonas. Sin embargo, todas las hormigas utilizan el mismo criterio para elegir sus movimientos, en cuanto a la información heurística y de la feromona. En nuestros algoritmos híbridos se utilizan diferentes especies de hormigas, las cuales utilizan diferentes enfoques en la función heurística y la actualización de la feromona. Además, para elegir sus movimientos, las hormigas pueden o no basar su decisión en

la información de la feromona proporcionada por otras especies de hormigas. Este proceso se explicará con detalle en el siguiente capítulo.

4.5 Otros algoritmos para el plegamiento de proteínas

En el trabajo de Lesh et al. (2003) se presenta una implementación genérica de búsqueda tabú usando un conjunto de transformaciones que los autores llaman "*pull moves*". En los experimentos realizados por estos autores, se encontraron soluciones a secuencias del modelo HP-2D en un tiempo de 3 a 14 horas en promedio, y se encontró la solución óptima en secuencias menores a 100 aminoácidos.

En el trabajo de Krasnogor et al. (2002) se muestra un algoritmo multi-memético para resolver el problema del plegamiento de proteínas con el modelo HP. Los algoritmos meméticos son algoritmos evolutivos que incluyen, como parte del ciclo evolutivo estándar de recombinación-mutación-selección, una fase de búsqueda local. Los algoritmos multi-meméticos utilizan un conjunto de varias búsquedas locales. En el trabajo de Krasnogor et al. se presentan experimentos realizados con secuencias del modelo HP-2D cuadrado (menores a 64 aminoácidos), y del modelo HP-2D triangular (menores a 37 aminoácidos). Estos autores encuentran conformaciones óptimas para las secuencias más pequeñas y conformaciones subóptimas para las secuencias más largas.

En el trabajo de Chen et al. (2005) se presenta un algoritmo basado en el método de ramificaciones (*branch*) y límites (*bound*) diseñado para el modelo HP-2D cuadrado. Este algoritmo no es capaz de encontrar soluciones óptimas para varias secuencias de aminoácidos, sobre todo para las mayores a 50 aminoácidos.

Por otro lado, en el trabajo de Rego et al. (2006) se presenta un método F&F (*Filter and Fan*) para estudiar el modelo HP-2D cuadrado. En los resultados que estos

autores presentan, se encuentran conformaciones con valores de energía iguales a las encontradas en (Shmygelska y Hoos, 2005), pero en menor tiempo de ejecución.

Finalmente, en el trabajo de Agarwala et al. (1997) se presenta un conjunto de reglas locales de plegamiento para el modelo HP-2D y HP-3D triangulares, las cuales pueden ser utilizadas para construir conformaciones en dichos modelos. Sin embargo, estos autores no reportan resultados de la aplicación de estas reglas en secuencias triangulares.

En esta sección se presentan varios algoritmos que no se muestran superiores en desempeño a los algoritmos presentados en secciones anteriores como PERM, el trabajo de Shmygelska y Hoos (2005), y los trabajos de Fidanova (2006), Song et al. (2006), y Chu et al. (2005). Sin embargo, los movimientos mostrados en (Lesh et al., 2003) se muestran interesantes para ser utilizados en una búsqueda local en adición a otro algoritmo.

En cuanto al modelo HP triangular, no existen muchos algoritmos que se hayan aplicado a este modelo, por lo que este problema es un tema interesante por estudiar.

4.6 Resumen y comparación de los principales métodos del capítulo

En este capítulo se expusieron los principales algoritmos encontrados en la literatura para el problema del plegamiento de proteínas. Los algoritmos más explorados y que han sido estudiados ya desde hace bastante tiempo son los algoritmos de Monte Carlo y los Algoritmos Genéticos. Un algoritmo que ha sido aplicado recientemente en el problema del plegamiento de proteínas es ACO. ACO ha mostrado ser un algoritmo eficiente para resolver el problema del plegamiento de

proteínas. ACO ha mejorado los resultados obtenidos por la mayoría de otros algoritmos en la literatura, y sólo PERM es capaz de encontrar mejores soluciones que ACO para algunas estructuras de longitud mayor del modelo cuadrado.

Se han propuesto varias aplicaciones de ACO para el modelo HP cuadrado y cúbico del plegamiento de proteínas, las cuales difieren sobre todo en la búsqueda local que utilizan, la actualización de la feromona y la función heurística utilizada. En la mayoría de los trabajos se utiliza el sistema de direcciones relativas para construir e indicar los movimientos en cada solución y son precisamente el tipo de movimientos que se utilizaron en los algoritmos desarrollados en esta tesis. Este tipo de movimientos son sencillos de entender, implementar e interpretar debido a que son rotacionalmente invariantes (sólo dependen de las dos posiciones vecinas más próximas) y no necesitan de información adicional para calcularse. Además, una ventaja importante de estos movimientos es que el cambio en una dirección en una determinada posición de la secuencia no modifica las direcciones del resto de la secuencia por lo que dichas direcciones no deben ser recalculadas.

Una parte en los algoritmos ACO que es muy enfatizada, sobre todo en los trabajos de Shmygelska et al. (2002; 2003; 2005), es la búsqueda local, la cual es utilizada sobre todo para evitar los mínimos locales. Sin embargo, muchos operadores utilizados no son suficientemente robustos para mejorar las soluciones (como varios operadores derivados de los métodos Monte Carlo) y otras propuestas necesitan mucho tiempo para ejecutarse debido a que prácticamente reconstruyen las soluciones encontradas por las hormigas. Debido a lo anterior, en los algoritmos propuestos en esta tesis no se utiliza búsqueda local. Para evitar el estancamiento en los valores de la feromona y por consiguiente evitar en cierta medida caer en mínimos locales, en esta tesis se propone una técnica de suavizado de datos, la cual se explica con detalle en el siguiente capítulo.

Para corregir el traslape durante la construcción de soluciones, los algoritmos en la literatura despliegan una parte de la solución construida y vuelven a plegarla hasta encontrar una estructura válida. Este proceso varía en la cantidad de aminoácidos que se despliegan cada vez que se encuentra el traslape y en los criterios que determinan la forma de volver a colocar dichos aminoácidos en la secuencia (muchas veces sólo se colocan aleatoriamente). Además, si durante la reconstrucción de la solución se regresa a una situación de traslape, los aminoácidos se vuelven a desplegar y plegar hasta encontrar una estructura válida, lo cual es un proceso lento. En los algoritmos presentados en esta tesis sólo se aceptan configuraciones válidas, esto es, cada vez que una hormiga llega a una situación de traslape es eliminada de la colonia. Este proceso se explica detalladamente en el siguiente capítulo.

Finalmente, para el modelo HP triangular no han sido propuestos en la literatura algoritmos basados en ACO. Para este modelo sólo existen aplicaciones con algoritmos genéticos, algoritmos exactos y algoritmos meméticos. Sin embargo, estos algoritmos no muestran un desempeño notable. Los algoritmos presentados en el siguiente capítulo han sido aplicados a este modelo obteniendo resultados favorables en cuanto a tiempo de ejecución y soluciones encontradas, como se muestra en el capítulo de resultados.

CAPITULO 5

Algoritmos ACO aplicados al modelo HP del plegamiento de proteínas

Los algoritmos ACO han sido utilizados en años recientes en una amplia variedad de diferentes problemas de optimización discreta, siendo la mayoría de estos problemas NP-duros. Dentro de este tipo de problemas encontramos el problema del agente viajero, el problema del coloreado de grafos, el problema de enrutamiento de vehículos y, uno de los más recientemente tratados, el problema del plegamiento de proteínas.

El problema del plegamiento de proteínas puede ser definido formalmente como: dada una determinada secuencia de aminoácidos $s = s_1 s_2 \ldots s_n$ y una función de energía $E(c)$, encontrar una conformación de s de mínima energía, esto es, encontrar $c^* \in C(s)$ tal que $E(c^*)=min\{E(c) \mid c \in C(s)\}$, donde $C(s)$ es el conjunto de todas las conformaciones válidas para s.

En este capítulo se presentan dos algoritmos ACO para el plegamiento de proteínas con el modelo HP triangular y cuadrado en 2D y 3D, cuyo funcionamiento se explica a detalle en las primeras secciones. En la última sección se presentan tres algoritmos ACO híbridos, cuyo objetivo es combinar los dos algoritmos de las primeras secciones de tres formas diferentes.

5.1 Descripción general de los algoritmos ACO

Los algoritmos presentados en esta tesis se desarrollan en diferentes fases: (1) Inicialización de Datos, (2) Inicialización de Hormigas, (3) Construcción de Conformaciones Candidatas, (4) Actualización de Feromona, y por último una fase llamada (5) Suavizado de Caminos de Feromona. Estas fases se observan de forma general en el Algoritmo 5.1, y se explican a continuación.

Algoritmo 5.1. Algoritmo general ACO

(1) Inicialización de datos
mientras (no se cumpla la condición de término) *hacer*
    (2) Inicialización de Hormigas
    (3) Construcción de Conformaciones Candidatas
    (4) Actualización de Feromona
    si (no hay mejora después de *n* iteraciones) *entonces*
        (5) Suavizado de Caminos de Feromona
    fin_si
fin_mientras

En *la fase de inicialización de datos*: se lee la secuencia de aminoácidos que va a ser plegada, se inicializan los parámetros del algoritmo, se inicializan los valores de la matriz de feromonas y, se inicializan variables que guardan información estadística. Esta fase sólo se lleva a cabo una vez cuando inicia el algoritmo. El resto de las fases del algoritmo se ejecutan iterativamente durante la ejecución del mismo hasta que se llegue a la condición de término, la cual puede ser un número máximo de iteraciones o encontrar una conformación con un determinado nivel de energía. En *la fase de inicialización de hormigas* se inicializa una colonia de *m* hormigas. Aquí, cada hormiga elige un punto inicial de plegamiento y coloca los dos primeros aminoácidos para comenzar a construir las soluciones. En una secuencia de tamaño *n*, este punto puede estar en una posición aleatoria que abarca el intervalo *n/4* y *3n/4*, esto es, entre las 2/4 partes centrales de la secuencia lineal. Para los algoritmos ACO híbridos, es en esta fase donde se determina la 'especie' de cada hormiga.

En el Algoritmo 5.2 se muestran los pasos que se llevan a cabo en *la fase de construcción de conformaciones.* En esta fase, las hormigas pliegan una secuencia de aminoácidos de una proteína agregando un aminoácido a la vez en la conformación. En el paso *Realizar Nuevo Movimiento*, cada hormiga determina la posición de un determinado aminoácido dentro de la conformación. La posición de cada aminoácido se determina a partir de los valores de dos fuentes de información: los valores de la matriz de feromonas τ y los valores de una función heurística η. La importancia de τ y η está determinada por los parámetros α y β respectivamente. Las distintas posiciones que un aminoácido puede tomar dentro de la conformación están basadas en el concepto de direcciones relativas, donde las posiciones son rotacionalmente invariantes. Se tienen arreglos de direcciones relativas, como se muestra en la Fig. 5.1, para el modelo HP cuadrado y triangular 2D.

Algoritmo 5.2. Procedimiento Construcción de Conformaciones Candidatas

```
para (cada paso de la secuencia) hacer
        para (cada hormiga) hacer
                Realizar Nuevo Movimiento
                si (sucede traslape)
                        Eliminar Hormiga Actual
                fin_si
        fin_para
        Ordenar Hormigas;
        Hacer Copias de las Mejores Hormigas;
fin_para
```

En estas direcciones relativas la posición de un aminoácido S_{i+1} es relativa a la posición de sus dos predecesores S_{i-1} y S_i. En el modelo HP cuadrado, al tener sólo dos dimensiones, las posibles direcciones son: S (Adelante), L (Izquierda), R (Derecha). Para el modelo triangular 2D las diferentes direcciones están determinadas por los ángulos de rotación de un triángulo equilátero, siendo las siguientes: S (0º), A (60º), B (120º), C (240º) y D (300º). Por simplicidad, al conjunto de direcciones relativas lo llamaremos *D*, ya que los algoritmos se aplican por igual a los dos modelos. En la Fig. 5.2 se muestra el sistema de coordenadas de los modelos cuadrado y triangular 2D. Se muestra un aminoácido central de referencia (en color azul), un aminoácido vecino también de referencia (en color verde) y los demás vecinos del aminoácido central (en color gris oscuro). A partir de los dos aminoácidos de referencia se pueden determinar las direcciones relativas descritas en la Fig. 5.1 moviéndose en dirección de las diferentes posiciones de los vecinos del aminoácido central. Como se puede observar en la Fig. 5.2, podemos tomar de referencia cualquier vecino del aminoácido central y las direcciones relativas se describen de igual forma. Es por eso que se dice que las direcciones relativas son rotacionalmente invariantes. De esta manera, la posición de los dos primeros aminoácidos se puede

establecer sin pérdida de generalidad, y a partir de ellos se comienza con la construcción de la conformación. Así, para una conformación de tamaño *n*, se tendrán conformaciones representadas por una secuencia de direcciones de tamaño *n-2*.

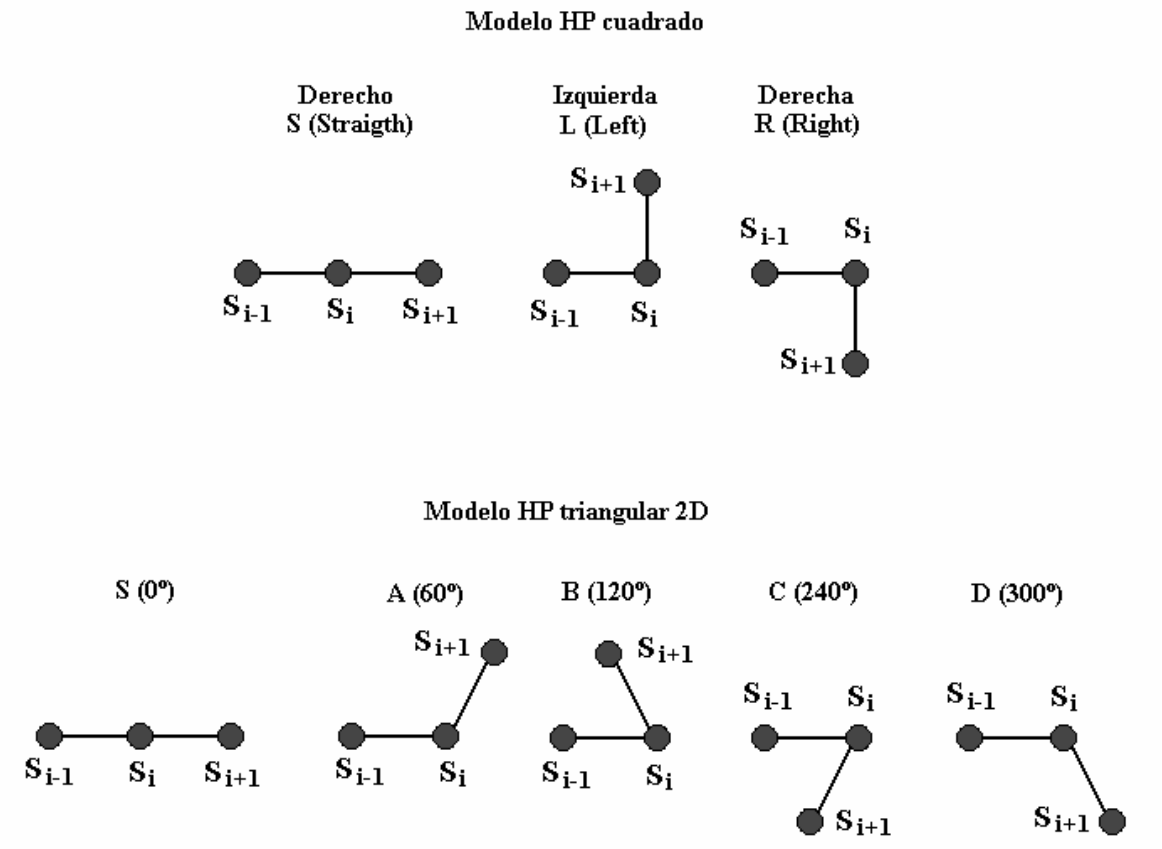


Fig. 5.1. Arreglos de direcciones relativas para el modelo HP cuadrado (S, L, R) y triangular 2D (S, A, B, C, D).

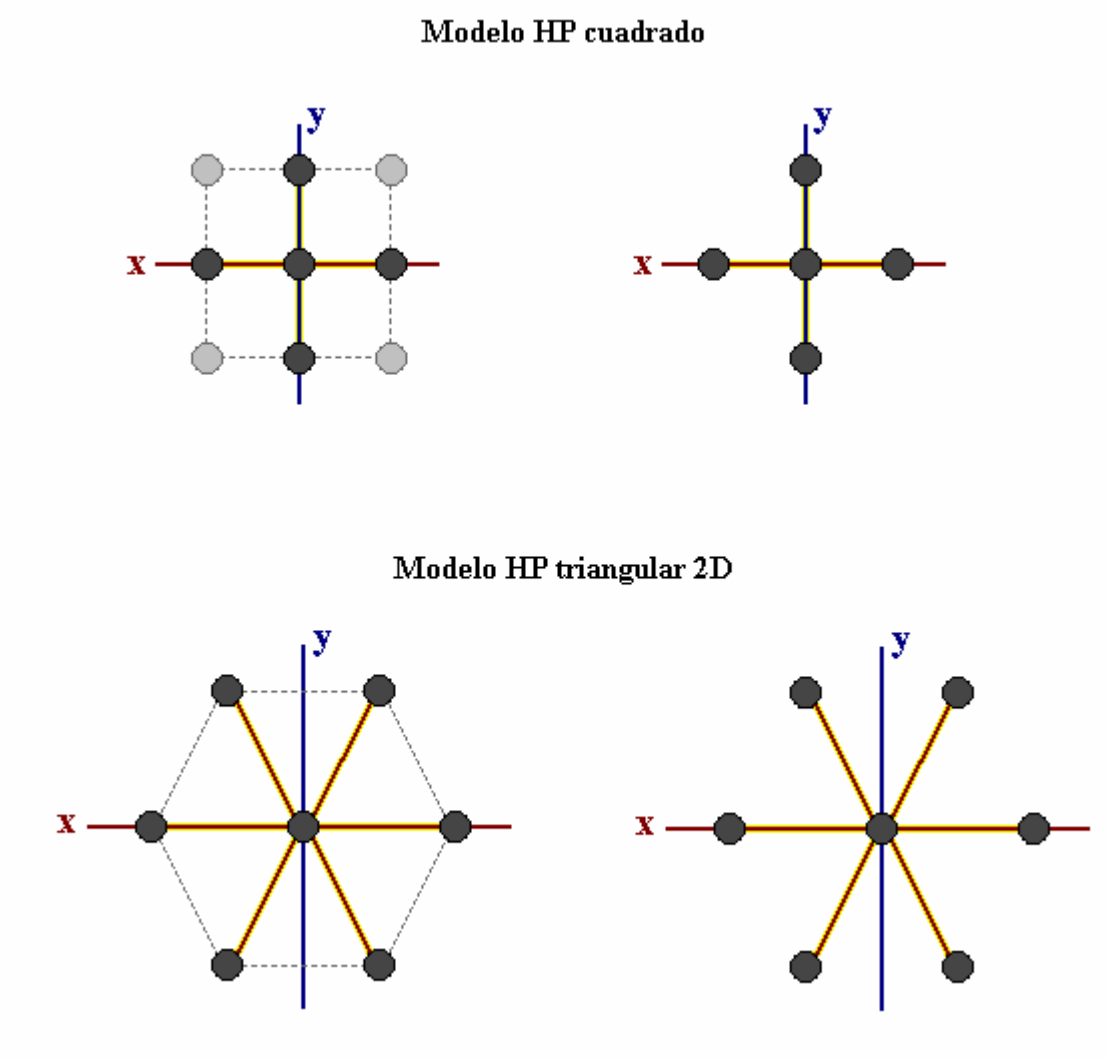


Figura 5.2. Sistema de coordenadas para el modelo HP cuadrado y triangular 2D. Se muestra el aminoácido central en color azul, un aminoácido vecino de referencia en color verde y el resto de sus posibles vecinos en color gris.

Durante la construcción de las conformaciones se puede dar el caso de que se llegue a un punto en donde un aminoácido no puede ser colocado en un espacio libre. En este caso se dice que sucede un *traslape*. Esta situación es solucionada de la siguiente forma. Primero, la estructura construida hasta el punto donde el traslape sucedió es desechada, esto es, la hormiga se elimina de la colonia temporalmente

(*Eliminar Hormiga Actual*). Las otras hormigas continúan con la construcción de la conformación y cada vez que el traslape ocurre, la hormiga correspondiente es eliminada. La eliminación de las hormigas puede justificarse como una manera de eliminar individuos defectuosos de una colonia, de forma similar a los algoritmos genéticos cuando se eligen sólo los mejores individuos para permanecer en la población. Cuando todas las hormigas han finalizado el paso de construcción, se ordenan en relación al valor de energía de la conformación que hayan encontrado hasta ese momento (*Ordenar Hormigas*). Después, para reemplazar las hormigas previamente desechadas, se realizan copias de las hormigas existentes, esto es, de aquellas que consiguieron completar el paso de construcción (*Hacer Copias de las Mejores Hormigas*).

Cuando se realizan las copias de las hormigas, se busca realizar una mayor cantidad de copias de aquellas hormigas que tengan una mejor conformación parcial, esto es, aquellas cuya conformación parcial tenga un menor nivel de energía. Para esto, utilizamos la distribución de Poisson, la cual se muestra en la Fig. 5.3. Esta distribución está dada por la ecuación 5.1.

$$p_1(k;\lambda) = \frac{\lambda^k e^{-\lambda}}{k!} \tag{5.1}$$

donde:

λ es una constante

k = 0, 1, 2, 3, …

Como podemos ver en la Fig. 5.3, para diferentes valores de λ se tienen distribuciones diferentes. Si el valor de λ es menor a 1, la gráfica tiende más rápidamente a cero. Por otro lado, si el valor de λ es mayor a uno, se obtienen distribuciones en donde para valores de k más pequeños se tienen valores menores a los valores donde k es mayor, lo cual no es deseable debido a que k=0 corresponde a la mejor hormiga.

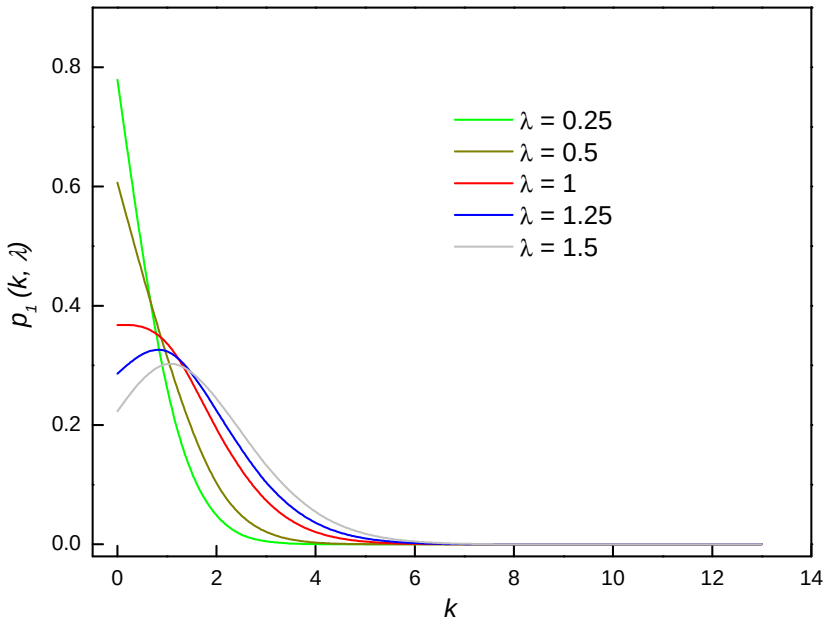


Fig. 5.3. Comportamiento de la distribución de Poisson con diferentes valores en λ.

De esta forma tenemos lo siguiente:

(1) Si el número de hormigas eliminadas es menor al número de hormigas restantes entonces se utiliza una distribución de Poisson dada por la ecuación 5.1, donde:

$\lambda = 1$

$k = 0, 1, 2, 3,...$ es la posición de la hormiga en la colonia ordenada, siendo la hormiga en la posición 0 aquella con la conformación que tenga el valor más pequeño en energía.

El número de copias de cada hormiga está determinado entonces por la ecuación 5.2.

$$\textit{Número de copias} = p_1(k; \lambda) * \textit{(Total de hormigas eliminadas)} \qquad (5.2)$$

De esta manera las copias se harán con las primeras hormigas de la colonia ordenada (las de menor energía). Debido a que es mayor la cantidad de hormigas restantes a las eliminadas conviene realizar copias sólo de las primeras mejores. Por ejemplo, si de un total de 100 hormigas se eliminaron 10, se tienen 90 hormigas restantes. Las copias se harán de acuerdo a la Tabla 5.1.

Tabla 5.1. Ejemplo del número de copias realizadas de acuerdo a la distribución de Poisson cuando el número de hormigas eliminadas es menor al número de hormigas restantes.

***k* (Posición de la hormiga)**	$p_1(k; \lambda)$	**Total de copias**
0	0.367879	4
1	0.367879	4
2	0.183939	2

Esto es, las copias se harán con las primeras 3 hormigas de la colonia. Sin embargo si tenemos el caso de que el número de hormigas eliminadas sea mayor al número de hormigas restantes, por ejemplo, de un total de 100 hormigas, se eliminaran 90 y se tuvieran 10 hormigas restantes sería deseable que las copias de las hormigas estuvieran un poco más distribuidas. Esto para evitar realizar demasiadas copias de una sola hormiga. Para esto se utiliza la distribución mostrada en la Fig. 5.4, la cual está dada por la ecuación 5.3.

$$p_2(k; \lambda) = e^{-(k/\lambda)} \tag{5.3}$$

donde:

λ es una constante

$k = 0, 1, 2, 3,...$

Como podemos ver en la Fig. 5.4 para λ = 10 se obtiene una distribución que tiende más lento a cero, lo que va a provocar que las copias de las hormigas se realicen de forma distribuida entre todas las hormigas restantes.

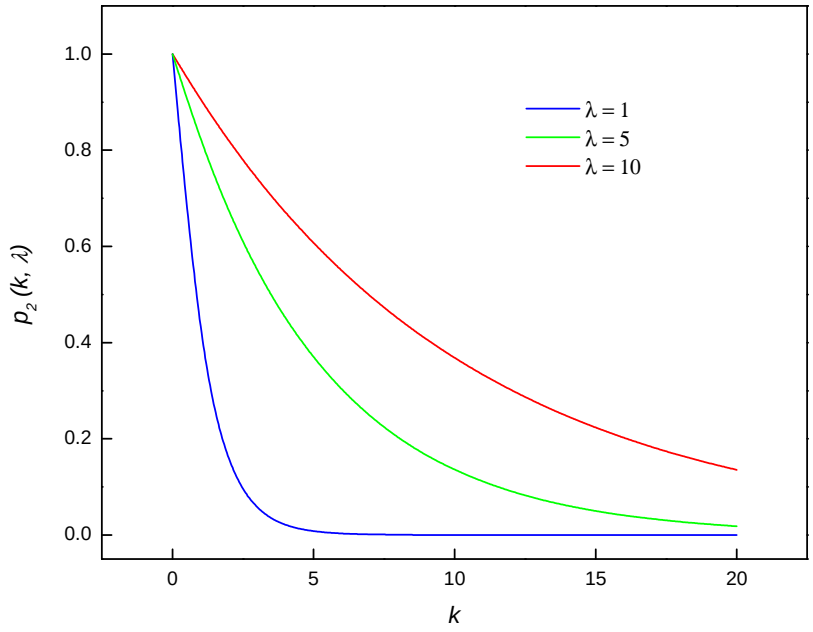


Fig. 5.4. Comportamiento de la distribución $p_2(k; \lambda) = e^{-(k/\lambda)}$ con diferentes valores en λ.

De aquí tenemos el segundo caso:

(2) Si el número de hormigas eliminadas es mayor al número de hormigas restantes, entonces se utiliza la distribución dada por la ecuación 5.3, donde:

λ = 10

k = 0, 1, 2, 3,... es la posición de la hormiga en la colonia ordenada siendo la hormiga en la posición 0 aquella con la conformación que tenga el valor más pequeño en energía.

El número de copias de cada hormiga está determinado entonces por la ecuación 5.4.

$$\textit{Número de copias} = p(k;\lambda) * \left(\frac{1}{\dfrac{\textit{No.Hormigas Restantes}}{\sum_{k=0} p_2(k;\lambda)}} \right) * \textit{(Total de hormigas eliminadas)} \qquad (5.4)$$

Es importante hacer notar la siguiente parte de la ecuación 5.4.

$$\frac{1}{\dfrac{\textit{No.Hormigas Restantes}}{\sum_{k=0} p_2(k;\lambda)}}$$

La función de este elemento es asegurar que se generen el total de hormigas eliminadas con las hormigas restantes. En la ecuación 5.2, esta parte se omite debido a que la sumatoria de los valores en la distribución de Poisson tiende a 1, por lo que esta parte se elimina automáticamente de la ecuación.

Una vez que se han generado las copias de las hormigas con cualquiera de las dos distribuciones presentadas anteriormente, dichas hormigas se colocan nuevamente dentro de la colonia (se realizan copias de conformaciones parcialmente plegadas). Una vez realizado esto, se continúa con la construcción de las conformaciones. En esta forma, las hormigas que no fueron capaces de alcanzar la estructura final continúan su búsqueda a partir de una estructura promisoria.

Una vez que las hormigas han terminado de construir las conformaciones, comienza *la fase de actualización de feromona*. En esta fase se tiene un mecanismo de evaporación de feromona, en el cual se reducen todos los valores de la matriz de feromonas por un factor constante ρ, donde $0 < \rho \leq 1$. De forma análoga con la naturaleza, este mecanismo representa una cierta pérdida de información conforme

avanza el tiempo. Además de la evaporación de feromona, también se refuerza el camino seguido por una fracción de las hormigas que hayan encontrado las conformaciones con menor nivel de energía (mayor cantidad de contactos H-H).

Después de que varias iteraciones han sido realizadas, es muy común que algunas posiciones en la matriz de feromona puedan tener valores demasiado altos y otras por el contrario, tengan un valor mínimo. Esto puede provocar que las hormigas elijan el mismo camino una y otra vez en iteraciones subsecuentes y por lo tanto las conformaciones que encuentren sean las mismas. Este problema es llamado *estancamiento*. Esta situación de estancamiento puede provocar que los algoritmos permanezcan en un mínimo local al llegar siempre a la misma solución. Para evitar esto, se propone un método para suavizar los valores de la feromona. En esto consiste la última fase del algoritmo propuesto llamada *fase de suavizado de caminos de feromona*. Ésta consiste en que si después de que en un cierto número de iteraciones (dependiendo del tamaño de la secuencia) no se observa mejora alguna, en términos de disminución en la función de energía en las soluciones encontradas, la diferencia entre los valores de los caminos de feromona es suavizada de la siguiente manera. Primero, un valor v, en el intervalo $0 < v < 1$, se suma a los valores de cada posición en la matriz de feromona de acuerdo a la ecuación 5.5.

$$\tau_{i,d} = (\tau_{i,d} + v) \tag{5.5}$$

Una vez hecho esto, los valores se normalizan de forma tal que la sumatoria de los valores de las direcciones relativas correspondientes a la misma posición de la secuencia sea 1, como se muestra en la ecuación 5.6.

$$\sum_{d \in D} \tau_{i,d} = 1 \tag{5.6}$$

donde $\tau_{i,d}$ se refiere a los valores de la feromona en la posición i de la secuencia y dirección relativa d.

Para apreciar con más detalle el efecto del suavizado de datos en los valores de la feromona, tomemos como ejemplo los siguientes datos:

0.3, 0.1, 0.1, 0.1, 0.4

Los cuales pueden ser valores de los caminos de la feromona para una determinada posición en la secuencia. Cada número representa el valor de feromona para los distintos movimientos disponibles en dicha posición. La varianza de estos datos es de 0.02. Después de aplicar el suavizado de datos, la varianza de estos datos es de 0.005 para v = 0.2, de 0.0016 para v = 0.5 y de 0.0005 para v = 1.0. Esta variación en los datos se muestra en la Fig. 5.5, en la cual se puede observar cómo se mantiene la proporción en los valores pero la varianza entre ellos se reduce.

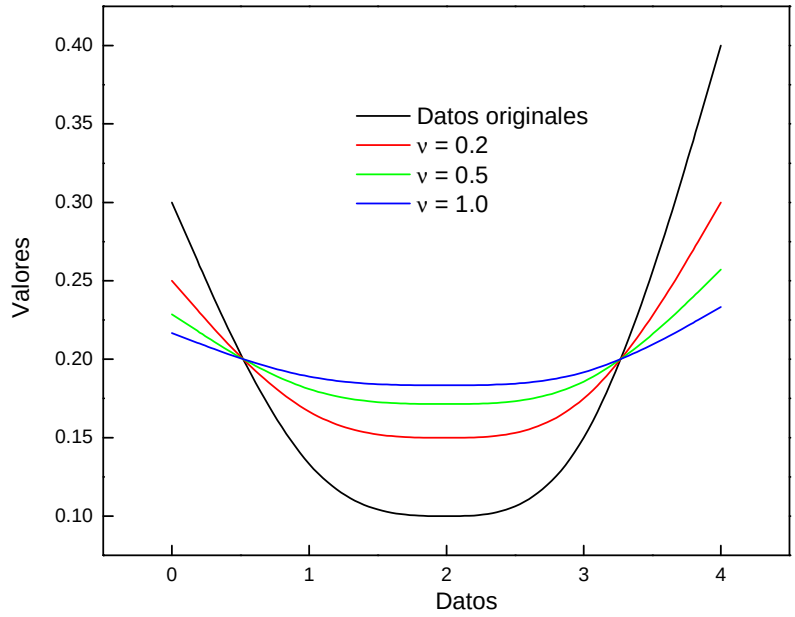


Fig. 5.5. Efecto del suavizado de datos en un conjunto de valores para diferentes valores de *v*.

Este método de suavizado incrementa la probabilidad de seguir caminos diferentes para construir nuevas soluciones sin perder totalmente la información que se ha ido almacenando hasta ese momento.

Dentro de las diferentes fases del algoritmo existen varios puntos principales a tomar en cuenta, como son:

- Función heurística utilizada.
- Fórmula para elegir probabilísticamente un nuevo movimiento.
- Actualización de los caminos de feromona.

En las secciones siguientes se da una descripción más detallada de cada una de estas partes para cada uno de los algoritmos individuales e híbridos presentados.

5.2 Algoritmo ACO-HR

Este algoritmo llamado ACO-HR (por ACO Hormigas Rojas) sigue de forma general el esquema presentado en la sección anterior (Algoritmo 5.1). Las diferentes partes que componen a este algoritmo se explican con detalle a continuación.

5.2.1 Fase de construcción de conformaciones

Una vez que han sido inicializados los primeros datos del algoritmo y se ha completado la fase de inicialización de las hormigas se tiene la fase de construcción de conformaciones candidatas (Algoritmo 5.2). En dicha fase, las hormigas construyen conformaciones para una secuencia lineal de aminoácidos H y P de una proteína dada. Comenzando desde el punto establecido en la fase de inicialización de

hormigas, éstas construyen soluciones plegando la secuencia lineal hacia ambas direcciones (izquierda y derecha), colocando un aminoácido a la vez en cada paso de construcción. De acuerdo a las expresiones 5.7 y 5.8, esta dirección es elegida probabilísticamente dependiendo del número de aminoácidos sin plegar a un respectivo lado dividido entre la suma de aminoácidos sin plegar en ambos lados.

$$P(izq) = \frac{a_{izq}}{a_{izq} + a_{der}} \tag{5.7}$$

$$P(der) = \frac{a_{der}}{a_{izq} + a_{der}} \tag{5.8}$$

donde *P(izq)* es la probabilidad de plegar hacia el lado izquierdo, *P(der)* es la probabilidad de plegar hacia el lado derecho, a_{izq} es el número de aminoácidos sin plegar del lado izquierdo y a_{der} es el número de aminoácidos sin plegar del lado derecho.

Como ya se mencionó antes, se usan direcciones relativas para colocar los aminoácidos en la conformación. Una cierta dirección relativa en cada paso de la construcción de la conformación es elegida probabilísticamente usando los valores de una función heurística $\eta_{i,d}$ y los valores de la matriz de feromonas $\tau_{i,d}$, donde *i* es una posición de la secuencia y *d* corresponde a una dirección relativa. La fórmula que combina la información de la heurística y la feromona está dada por la expresión 5.9.

$$p_{i,d} = \frac{(\tau_{i,d})^{\alpha}(\eta_{i,d})^{\beta}}{\sum_{e \in D}(\tau_{i,e})^{\alpha}(\eta_{i,e})^{\beta}} \tag{5.9}$$

Los valores heurísticos $\eta_{i,d}$ se determinan, al igual que en (Shmygelska y Hoos, 2005; Shmygelska, 2006), basándose en el número de nuevos contactos H-H que se forman al realizar un movimiento en la dirección *d*, usando la ecuación 5.10.

$$\eta_{i,d} = e^{\mu \cdot H_{i,d}} \tag{5.10}$$

donde μ es un valor constante y $H_{i,d}$ es el número de nuevos contactos H-H que se forman cuando el aminoácido *i* se coloca en la conformación usando el movimiento *d*.

5.2.2 Actualización de feromona

La fase de actualización de feromona se realiza después de cada iteración en que las hormigas han construido las soluciones. Se tienen los siguientes pasos:

- Primero se realiza la evaporación de la feromona de acuerdo a la ecuación 5.11.

$$\tau_{i,d} = \rho \, . \, \tau_{i,d} \, , \qquad 0 < \rho < 1 \tag{5.11}$$

- Luego, se refuerzan los valores de la feromona del camino de la hormiga con la mejor conformación *c* encontrada en esa iteración, de acuerdo a la ecuación 5.12.

$$\tau_{i,d} = \tau_{i,d} + \Delta_{i,d,c} \tag{5.12}$$

  En la ecuación 5.12, $\Delta_{i,d,c}$ tiene un valor constante mayor a cero que depende del tamaño de la secuencia que se está plegando, si en la posición *i* de la secuencia se eligió el movimiento *d* en la construcción de la conformación *c* (la mejor conformación de la iteración actual). En caso contrario el valor es cero.

- Finalmente se normalizan los valores de la feromona. Primero se hace una normalización tal que $\sum_{d \in D} \tau_{i,d} = 1$, para cada aminoácido *i*. De forma adicional, si el valor mínimo de feromona en la posición *i* ($\min_{d \in D} \tau_{i,d}$) es menor que un umbral θ, dicho valor es fijado en θ.

  Esta estrategia de establecer un umbral para el valor mínimo es similar al que se propuso en *MAX-MIN Ant System*, con la diferencia de que nosotros no establecemos un umbral para el valor mayor. Esto es debido a que al normalizar los valores después de cada iteración, cada vez que se termina de actualizar la feromona se está limitando el valor de los caminos de feromona a ser siempre menores a 1. Sin embargo, si algún valor se vuelve cero, aún después de realizar la normalización dicho valor permanecerá en cero. Estableciendo el umbral θ se evita que los valores de los movimientos en una cierta posición se hagan demasiado pequeños y se pierda la probabilidad de elegirlos en iteraciones futuras.

5.3 Algoritmo ACO-HN

Este algoritmo llamado ACO-HN (por ACO Hormigas Negras) sigue, de manera similar al algoritmo anterior, el esquema presentado en el Algoritmo 5.1. La diferencia entre el algoritmo anterior y el que se presenta en esta sección se encuentra en la fase de construcción de conformaciones y en la fase de actualización de feromona, las cuales se describen a continuación.

5.3.1 Fase de construcción de conformaciones

Para elegir una cierta dirección relativa en cada paso de la construcción de la conformación se utiliza la ecuación 5.13.

$$p_{i,d} = \begin{cases} \dfrac{(\tau_{i,d})^{\alpha}(\eta_{i,d})^{\beta}}{\sum_{e \in D}(\tau_{i,e})^{\alpha}(\eta_{i,e})^{\beta}} & , si\ A_i = H \\ 1/|D| & , si\ A_i = P \end{cases} \tag{5.13}$$

donde A_i se refiere al tipo de aminoácido en la posición i de la cadena lineal y $|D|$ es la cardinalidad del conjunto de posibles direcciones relativas. Si el aminoácido que se va a colocar es de tipo H, la fórmula para elegir la dirección relativa es similar a la del algoritmo anterior (5.9). Sin embargo, en este algoritmo los valores heurísticos $\eta_{i,d}$, al igual que en (Fidanova, 2006), se determinan basándose sólo en el número de nuevos contactos H-H que se forman al realizar el movimiento d en la posición i de la secuencia lineal ($h_{i,d}$), como se expresa en la ecuación 5.14.

$$\eta_{i,d} = h_{i,d} \tag{5.14}$$

Cuando el aminoácido que se va a colocar es de tipo P, la dirección relativa se elige de forma aleatoria. Esto es, que todos los movimientos válidos tendrán exactamente la misma probabilidad de ser elegidos. Al colocar los aminoácidos P de forma aleatoria se consigue tener más aleatoriedad en el momento de construir las soluciones, lo que permite explorar otros caminos.

5.3.2 Actualización de feromona

En este algoritmo, la actualización de feromona se lleva a cabo de la siguiente manera:

- Primero se efectúa la evaporación de feromona mediante la ecuación 5.11.

- Luego se actualizan los valores de las posiciones que sean parte de los mejores caminos de todas las hormigas que hayan encontrado una conformación con un nivel de energía igual o menor a las conformaciones encontradas desde el inicio del algoritmo (se toman en cuenta sólo las hormigas con conformaciones diferentes entre sí). Esto se realiza de acuerdo a la ecuación 5.12. El efecto que tiene realizar la actualización de la feromona sólo cuando se iguale o se minimice un determinado nivel de energía es evitar reforzar caminos que no consiguen mejorar las soluciones. Al tomar en cuenta todas las hormigas con buenas conformaciones se refuerzan diferentes caminos que se presentan como prometedores en la construcción de soluciones.

- Finalmente se normalizan los valores y se revisa que todos los valores sean mayores al umbral θ establecido como mínimo. De no ser así, dicho valor se establece en θ.

Esta propuesta de actualización de feromona es similar a la de (Fidanova, 2006), en el sentido de las hormigas que se toman en cuenta para reforzar los caminos de feromona. Sin embargo, a diferencia de nuestra propuesta, en (Fidanova, 2006) no se realiza la normalización de valores ni se tiene una estrategia para limitarlos y evitar que se hagan demasiado grandes o pequeños. Además de que en la propuesta de (Fidanova, 2006) se utiliza también una actualización de feromona local después de

cada paso de construcción, mientras que nosotros sólo utilizamos actualización global al término de cada iteración.

5.4 Algoritmos ACO híbridos

Como mencionamos en capítulos anteriores, un problema que se presenta al utilizar ACO para resolver algún problema es elegir correctamente la heurística a aplicar, el esquema de actualización de feromona y la forma correcta de combinar la información de ambos para la construcción de soluciones. Diferentes enfoques en estos puntos pueden funcionar mejor para algunas secuencias del mismo problema que para otras.

Al realizar diferentes experimentos con los algoritmos presentados en las secciones 5.2 y 5.3 se observó que para varias secuencias lineales de aminoácidos un cierto algoritmo funcionaba mejor que otro al momento de construir soluciones. Esto es, para los dos algoritmos existían secuencias en las que se mostraban superiores y otras secuencias en las que su desempeño caía.

Para resolver este problema se planteó la posibilidad de tener diferentes especies de hormigas ejecutándose juntas para lograr obtener buenos resultados de forma consistente. Cada especie de hormiga utilizaría enfoques diferentes en la heurística y en el esquema de actualización de feromona.

A partir de lo anterior se buscó hacer una combinación entre los dos algoritmos individuales (ACO-HR Y ACO-HN) presentados anteriormente para aprovechar las ventajas que los dos ofrecen. En esta sección se muestra la descripción de tres algoritmos ACO híbridos que combinan los dos algoritmos presentados anteriormente de diferentes maneras.

De manera general se puede decir que en los algoritmos ACO híbridos se tienen diferentes especies de hormigas que conviven en un mismo ambiente pudiendo utilizar información de hormigas de otra especie para encontrar la solución a algún problema. En nuestro caso se utilizan dos especies de hormigas, rojas (ACO-HR) y negras (ACO-HN), pero los algoritmos pueden extenderse a más de dos especies. A partir de lo anterior, tenemos de forma general dos enfoques distintos de interacción entre las especies distintas de hormigas:

- Las hormigas sólo se comunican con las hormigas de su misma especie.
- Las hormigas de una especie utilizan información de hormigas de otra especie para tomar decisiones.

Con el primer enfoque se presenta el algoritmo *ACO-Híbrido sin interacción entre especies* (H-ACO), y siguiendo el segundo enfoque se presentan los algoritmos *ACO-Híbrido con método de suma* (SH-ACO) y *ACO-Híbrido con método de trazas* (TH-ACO). El funcionamiento de estos tres algoritmos se presenta con detalle en las siguientes secciones.

De manera general cabe mencionar que en los tres algoritmos ACO híbridos cada especie de hormiga tendrá su propia matriz de feromonas la cual, de la misma forma en que se efectúa con los algoritmos individuales, se inicializa en la fase de inicialización de datos.

Durante la fase de inicialización de las hormigas se establece la especie de cada hormiga. A cada especie le corresponderá la misma cantidad de hormigas en un inicio. Conforme avanza el algoritmo, esta proporción irá variando dependiendo de la calidad de soluciones que cada especie encuentre. Después de que un determinado número de iteraciones (dependiendo del tamaño de la secuencia que se esté plegando) ha sido completado, se evalúan las soluciones que han encontrado las diferentes especies de hormigas. La especie de hormigas que tenga la mayor cantidad de

conformaciones con energía mínima será reforzada añadiendo más hormigas de dicha especie en la siguiente iteración, mientras que la cantidad de las otras especies será reducida en la misma proporción. La cantidad de hormigas de una especie no debe ser menor a un umbral mínimo establecido. Si al reducir la población de hormigas de una especie se rebasara ese umbral, la cantidad de hormigas de dicha especie se mantendrá en el valor mínimo. Conforme se repite este proceso a lo largo del algoritmo se consigue tener una mayor cantidad de hormigas de la especie que presenta un mejor desempeño sin eliminar completamente a las hormigas de otras especies.

5.4.1 ACO Híbrido sin interacciones entre especies (H-ACO)

La primera versión de ACO híbrido, llamada H-ACO, se basa en el principio de que las hormigas sólo se comunican con las hormigas de su misma especie. Dicho de otro modo, el algoritmo H-ACO consiste en la ejecución en paralelo de los algoritmos que representan una especie distinta de hormigas. En este algoritmo las hormigas construyen soluciones basándose sólo en el conocimiento generado por las hormigas de su mismo tipo (en su propia matriz de feromonas) y utilizando la heurística propia de la especie. Además, las hormigas sólo pueden modificar su propia matriz de feromonas y no pueden acceder en ningún momento a la matriz de feromonas de la otra especie de hormigas.

La ventaja que ofrece este algoritmo con respecto a los algoritmos individuales es que se logran conjuntar las características de dichos algoritmos sin modificar su funcionamiento básico.

5.4.2 ACO Híbrido con método de suma (SH-ACO)

La idea principal de este algoritmo, llamado SH-ACO, y del que se presenta en la siguiente sección, es que todas las hormigas basan sus decisiones combinando de alguna manera la información propia de su especie junto con la información que hormigas de especies diferentes a la suya han recopilado. El método que se propone para realizar esta combinación es utilizar una fórmula de evaluación diferente para elegir una cierta dirección relativa en la construcción de las conformaciones.

En esta versión se utiliza la fórmula de evaluación dada por la expresión 5.15.

$$p_{i,j} = \frac{\left(\sum_{k=1}^{NH} \tau_{i,j,k}\right)^{\alpha} \left(\eta_{i,j,k}\right)^{\beta}}{\sum_{h \in D}\left(\left(\sum_{k=1}^{NH} \tau_{i,h,k}\right)^{\alpha} \left(\eta_{i,h,k}\right)^{\beta}\right)} \qquad (5.15)$$

donde NH es el número de especies distintas de hormigas, $\tau_{i,j,k}$ es el valor de la feromona de la especie de hormiga k en la posición i de la secuencia y la dirección relativa j; y $\eta_{i,j,k}$ es el valor de la heurística correspondiente al tipo de hormiga k. Aquí, una hormiga de tipo k elige plegar el aminoácido i de la secuencia lineal en una dirección relativa j con una probabilidad determinada por la expresión 5.15, utilizando la información de la heurística utilizada por las hormigas de especie k, y una sumatoria de los valores de la feromona depositada en dicha posición por las hormigas de todas las especies.

En esta versión del algoritmo híbrido, todas las hormigas tienen una interacción aún si son de especies distintas al tomar en cuenta la feromona depositada en el ambiente por todas las especies de hormigas.

La principal aportación de este algoritmo y el que se presenta en la siguiente sección es la forma de combinar la información de diferentes especies de hormigas (diferentes enfoques en la función heurística y actualización de feromona) en la fórmula de evaluación y no sólo ejecutando las especies en paralelo.

5.4.3 ACO Híbrido con método de trazas (TH-ACO)

En este algoritmo, llamado TH-ACO, las hormigas construyen las conformaciones tomando en cuenta la información de la matriz de feromonas de las hormigas de especie distinta a la suya. Sin embargo, en este algoritmo se utiliza además la información proporcionada por una matriz adicional llamada 'matriz de trazas'. La función de dicha matriz de trazas es almacenar aquellos puntos por donde han pasado las hormigas al momento de construir las conformaciones. Cada vez que una hormiga pasa por un cierto punto en la construcción de la conformación, esto es, cada vez que una hormiga de especie *k* coloca en la conformación al aminoácido *i* de la secuencia usando la dirección relativa *j*, dicha posición $p_{i,j,k}$ es marcada en la matriz de trazas. De esta manera una hormiga de especie k_1 tomará en cuenta la información de la feromona de las hormigas de especie $k_2,\ldots,k_n$ sólo si alguna hormiga de esas especies ya ha pasado por la posición en la que se encuentra actualmente dicha hormiga, esto es, si alguna otra hormiga ya ha dejado su traza en la matriz de trazas. En caso contrario sólo se toma en cuenta la matriz de feromona de las hormigas de su misma especie. De esta manera, tenemos la fórmula de evaluación dada en la expresión 5.16.

$$p_{i,j} = \frac{\left(\sum_{k=1}^{NH} \tau_{i,j,k} * \sigma_{i,j,k}\right)^{\alpha} \left(\eta_{i,j,k}\right)^{\beta}}{\sum_{h \in D}\left(\left(\sum_{k=1}^{NH} \tau_{i,h,k} * \sigma_{i,j,k}\right)^{\alpha} \left(\eta_{i,h,k}\right)^{\beta}\right)} \qquad (5.16)$$

donde *NH* es el número de especies distintas de hormigas, $\tau_{i,j,k}$ es el valor de la feromona de la especie de hormiga *k* en la posición *i* de la secuencia y la dirección relativa *j*; y $\eta_{i,j,k}$ es el valor de la heurística correspondiente al tipo de hormiga *k*. $\sigma_{i,j,k}$ es el valor de la matriz de trazas de la especie de hormiga *k* en la posición *i* de la secuencia y la dirección relativa *j* y nos indica si alguna hormiga de la especie *k* ya ha colocado el aminoácido *i* de la secuencia utilizando la dirección relativa *j*.

La principal diferencia de este algoritmo con el presentado en la sección anterior es el uso de la matriz de trazas, la cual discrimina cuándo tomar en cuenta la información de otras especies de hormigas y cuando no. En el algoritmo SH-ACO siempre se toma en cuenta la información proporcionada por otras especies de hormigas. En el algoritmo TH-ACO se obtiene información generada en la iteración actual (matriz de trazas) que nos dice qué tan factible es que las hormigas de una especie sigan los mismos caminos de las hormigas de otra especie. En TH-ACO sólo se toma en cuenta la información de otras especies de hormigas si en algún determinado momento en la construcción de soluciones de la iteración actual, el camino de las hormigas se intersecta.

5.5 Resumen del capítulo 5

En este capítulo se describieron diferentes métodos basados en ACO para aplicarlos en el problema del plegamiento de proteínas. Los métodos propuestos en esta tesis utilizan dos heurísticas presentadas en la literatura. El primer algoritmo presentado (ACO-HR) utiliza la heurística presentada en (Shmygelska y Hoos, 2005) y el segundo algoritmo presentado (ACO-HN) utiliza la heurística mostrada en (Fidanova, 2006). Además, los movimientos utilizados por las hormigas para la construcción de las diferentes conformaciones son los mismos utilizados en la mayoría de los trabajos presentados en la literatura.

Sin embargo, los métodos descritos en esta sección no utilizan búsqueda local para mejorar las soluciones encontradas, a diferencia de otros métodos ACO presentados en la literatura.

Otra diferencia consiste en los diferentes esquemas de actualización de feromona utilizados. De forma similar al esquema utilizado en *MAX-MIN Ant System*, se tiene un límite en los valores que pueden tomar los valores de feromona. Sin embargo, en *MAX-MIN Ant System* se trabaja con un límite inferior y en uno superior. En los métodos propuestos sólo se tiene un límite inferior para evitar que algún valor se pudiera volver cero.

Se incluye además una función en los métodos propuestos que no se ha propuesto en otros algoritmos ACO. Esta función, llamada suavizado de feromona, permite evitar el estancamiento en los métodos propuestos cuando los valores de la feromona tienden a favorecer demasiado sólo a algunos movimientos.

También, de forma particular en el problema del plegamiento de proteínas, se propone un nuevo esquema para lidiar con el traslape en la construcción de conformaciones diferente a todos los esquemas propuestos en la literatura.

Por otro lado, en este capítulo se describieron tres métodos ACO híbridos en donde se combinan los dos métodos ACO individuales presentados en las primeras secciones. Los métodos ACO híbridos trabajan con diferentes especies de hormigas: de especie ACO-HR o de especie ACO-HN. De esta manera, el método H-ACO trabaja con dos especies de hormigas ejecutándose de forma paralela sin compartir información. Los métodos SH-ACO y TH-ACO trabajan con dos especies de hormigas que comparten información cuando se elige algún movimiento para la construcción de soluciones. La información que las hormigas de distinta especie comparten es la información proporcionada por los valores de feromona, y la combinan en la fórmula de evaluación para elegir un nuevo movimiento. El método

SH-ACO combina la información de todas las especies de hormigas de forma general y el método TH-ACO lo hace de forma selectiva.

En el siguiente capítulo se presentan diferentes pruebas que muestran los resultados que se obtuvieron con los diferentes algoritmos presentados en este capítulo sobre secuencias estándar de los modelos cuadrado y triangular.

CAPÍTULO 6

Experimentos y resultados

Para comparar los resultados de los diferentes algoritmos ACO presentados en el capítulo 5 entre ellos mismos y con los resultados obtenidos por otros algoritmos encontrados en la literatura se realizaron varias pruebas en diferentes secuencias estándar. Los experimentos consistieron en ejecuciones independientes de cada secuencia y con cada uno de los algoritmos presentados en el capítulo anterior para el modelo HP cuadrado y el modelo HP triangular 2D. Los valores de feromona se inicializaron usando números aleatorios en el intervalo 0.1 a 1.0. Después, estos valores se normalizaron usando la ecuación 5.6 y se les aplicó el método de suavizado de feromona mostrado en el capítulo anterior. Se utilizó una población de 100 hormigas y la condición de término de los algoritmos fue encontrar la solución óptima de la secuencia o concluir un máximo de iteraciones dependiente de la secuencia a procesar. En los algoritmos ACO híbridos, la población de 100 hormigas se repartió en 50 hormigas para cada una de las dos especies en la inicialización.

También se realizaron varias pruebas para mostrar la eficacia de la estrategia para evitar el estancamiento y de la estrategia para solucionar el traslape. Se muestran también otros experimentos sobre el efecto que tiene variar el tamaño de la colonia de

hormigas y finalmente, se muestran varias pruebas sobre el efecto de la variación de parámetros en los algoritmos. Todos los experimentos se realizaron en una máquina con procesador AMD Athlon a 1.53 GHz, con 256 MB de memoria RAM bajo Windows. Los algoritmos se implementaron en Java y aunque usan múltiples hormigas en cada iteración, fueron implementados secuencialmente.

6.1 Resultados de las secuencias estándar del modelo HP-cuadrado

Para evaluar los algoritmos presentados en el capítulo 5 para el modelo HP-cuadrado se utilizaron las secuencias mostradas en la Tabla 6.1, las cuales son secuencias estándar utilizadas ampliamente en la literatura (Chikenji et al., 1999; Liang y Wong, 2001; Shmygelska y Hoos, 2005; Unger y Moult, 1993a).

Tabla 6.1. Secuencias del modelo HP-2D.

ID	Tamaño	Energía óptima	Secuencia proteínica
SQ-1	18	-9	$PHP(PH)_2(H_2P)_2H_5$
SQ-2	18	-8	$(HP)_2H_3P_3H_4P_2H_2$
SQ-3	18	-4	$H_2P_5H(HP_2)_2PHP$
SQ-4	20	-9	$(HP)_2PH_2PHP_2HPH_2P_2HPH$
SQ-5	24	-9	$H_2(P_2H)_7H$
SQ-6	25	-8	$P_2HP_2(H_2P_4)_3H_2$
SQ-7	20	-10	$H_3P_2(HP)_3(PH)_3P_2H$
SQ-8	36	-14	$P_3H_2P_2H_2P_5H_7P_2H_2P_4H_2P_2HP_2$
SQ-9	48	-23	$P_2H(P_2H_2)_2P_5H_{10}P_6(H_2P_2)_2HP_2H_5$
SQ-10	50	-21	$H_2(PH)_3PH_4PH(P_3H)_2P_4H(P_3H)_2PHPH_4(HP)_3H_2$
SQ-11	64	-42	$H_{12}(PH)_2(P_2H_2)_2P_2HP_2H_2PPH_2P_2HP_2(H_2P_2)_2(HP)_2H_{12}$
SQ-12	85	-53	$H_4P_4H_{12}P_6(H_{12}P_3)_3HP_2(H_2P_2)_2HPH$
SQ-13	100	-48	$P_6HPH_2P_5H_3PH_5PH_2P_4H_2P_2H_2PH_5PH_{10}PH_2PH_7P_{11}H_7P_2HP$ $H_3P_6HPH_2$
SQ-14	100	-50	$P_3H_2P_2H_4P_2H_3(PH_2)_2PH_4P_8H_6P_2H_6P_9HPH_2PH_{11}P_2H_3PH_2P$ $HP_2HPH_3P_6H_3$

Los experimentos realizados sobre las secuencias de la Tabla 6.1 consistieron en ejecuciones independientes para cada secuencia (100 para las secuencias SQ-1 a SQ-8, 50 para las secuencias SQ-9 a SQ-11, y 20 para las secuencias SQ-12 a SQ-14) y para cada algoritmo. Los parámetros de los algoritmos utilizados en todos los experimentos fueron: $\alpha = 1$, $\beta = 2$, $\rho = 0.85$, $\theta = 0.05$ y $\mu = 0.2$.

En las tablas 6.2, 6.3 y 6.4 se muestran los resultados obtenidos por los diferentes algoritmos ACO híbridos (H-ACO, SH-ACO y TH-ACO) así como los resultados de los algoritmos ACO individuales (ACO-HR y ACO-HN). En la Tabla 6.2 y en la Tabla 6.3 se reportan el número de iteraciones promedio y el tiempo de ejecución promedio, respectivamente, que le toma al algoritmo encontrar estructuras con un mínimo nivel de energía (también reportado en las tablas). En la Tabla 6.4 se muestra el porcentaje de ejecuciones en las que se encontró el nivel de energía reportado, llamado porcentaje de éxito.

En estas tablas podemos observar que los algoritmos ACO individuales tienen un desempeño variable en las diferentes secuencias, esto es, para algunas se desempeñan mejor que para otras. Por ejemplo, para la secuencia SQ-8, el algoritmo ACO-HR tiene un bajo porcentaje de ejecuciones en las que encuentra una solución de mínima energía, mientras que el algoritmo ACO-HN se desempeña mejor para esta secuencia. De manera similar, el algoritmo ACO-HN no consiguió encontrar estructuras de mínima energía para las secuencias SQ-10 y SQ-11, mientras que el algoritmo ACO-HR encuentra dichas estructuras de manera bastante eficiente. De forma general, el algoritmo individual que mejor desempeño presenta es el algoritmo ACO-HR.

Tabla 6.2. Comparación del desempeño de los algoritmos ACO individuales (ACO-HR y ACO-HN) y las versiones híbridas (H-ACO, SH-ACO y TH-ACO) de ACO. *E* se refiere al mínimo nivel de energía que los algoritmos fueron capaces de alcanzar, I_{avg} es el número promedio de iteraciones requeridas por los algoritmos para alcanzar *E*.

ID	ACO-HR		ACO-HN		H-ACO		SH-ACO		TH-ACO	
	E	**$I_{avg.}$**	**E**	**$I_{avg.}$**	**E**	**$I_{avg.}$**	**E**	**$I_{avg.}$**	**E**	**$I_{avg.}$**
SQ-1	-9	**60.41**	-9	125.11	-9	73.20	-9	**60.48**	-9	74.29
SQ-2	-8	25.72	-8	**12.61**	-8	**13.54**	-8	15.05	-8	13.75
SQ-3	-4	**586.65**	-4	690.92	-4	811.118	-4	646.83	-4	**633.36**
SQ-4	-9	70.02	-9	**13.82**	-9	**17.90**	-9	19.82	-9	20.53
SQ-5	-9	60.88	-9	**30.15**	-9	**27.56**	-9	28.0	-9	27.59
SQ-6	-8	85.33	-8	**26.85**	-8	34.01	-8	35.2	-8	**28.16**
SQ-7	-10	**414.19**	-10	1092.94	-10	**427.49**	-10	558.85	-10	487.93
SQ-8	-14	1100.57	-14	**1006.4**	-14	1138.78	-14	1101.66	-14	**863.32**
SQ-9	-23	**1489.58**	-23	1893.65	-23	**1410.47**	-23	1622.33	-23	1776.7
SQ-10	-21	**642.1**	-20	2775.94	-21	887.30	-21	**883.9**	-21	1009.6
SQ-11	-42	**995.93**	-38	3250.2	-42	852.738	-42	**812.47**	-42	852.73
SQ-12	-51	**2358.5**	-51	5694	-51	4595	-51	**3248**	-51	4633
SQ-13	-46	**5378.23**	-45	10648	-46	15908.6	-46	**9250.9**	-46	9643.67
SQ-14	-46	**6543.76**	-46	28996	-46	1844.5	-46	2245	-46	**1817**
PROMEDIO		1415.13		4018.33		2003.01		1466.32		1562.97

A primera vista, cuando observamos el promedio del desempeño de los algoritmos ACO individuales e híbridos en cuanto al número de iteraciones requeridas para encontrar conformaciones óptimas (Tabla 6.2), podemos observar que el algoritmo con un mejor desempeño es el algoritmo ACO-HR. Sin embargo, cuando realizamos una comparación del desempeño de los algoritmos en cuanto al tiempo de ejecución (Tabla 6.3) podemos observar que a pesar de que el algoritmo ACO-HR realiza en promedio menos iteraciones para hallar conformaciones óptimas, el algoritmo SH-ACO es más rápido que el anterior. Además, el porcentaje de éxito en los algoritmos ACO híbridos es más elevado que el de los algoritmos ACO individuales. Si observamos la Tabla 6.4, podemos ver que el porcentaje de éxito de los algoritmos ACO híbridos es mayor en casi todos los casos (menos en la secuencia

SQ-11) que el porcentaje de éxito que presentan los algoritmos ACO individuales. Una posible razón por la cual el porcentaje de éxito de la secuencia SQ-11 es menor en los algoritmos ACO híbridos que el que presenta el algoritmo ACO-HR es que al combinar los dos algoritmos individuales, el bajo desempeño del algoritmo ACO-HN provoca que los algoritmos ACO híbridos necesiten un mayor número de iteraciones para contrarrestar el mal desempeño de las hormigas de especie ACO-HN.

Tabla 6.3. Comparación del desempeño de los algoritmos ACO individuales (ACO-HR y ACO-HN) y las versiones híbridas (H-ACO, SH-ACO y TH-ACO) de ACO. *E* se refiere al mínimo nivel de energía que los algoritmos fueron capaces de alcanzar, T_{avg} es el tiempo promedio de CPU, dado en segundos, requerido por los algoritmos para alcanzar *E*.

ID	ACO-HR		ACO-HN		H-ACO		SH-ACO		TH-ACO	
	E	**T_{avg}**	**E**	**T_{avg}**	**E**	**T_{avg}**	**E**	**$T_{avg.}$**	**E**	**$T_{avg.}$**
SQ-1	-9	**0.4319**	-9	0.8103	-9	0.4883	-9	**0.4084**	-9	0.4917
SQ-2	-8	0.1725	-8	**0.0815**	-8	0.0969	-8	0.1096	-8	**0.0918**
SQ-3	-4	**3.6817**	-4	4.1149	-4	4.9016	-4	**3.9527**	-4	3.9581
SQ-4	-9	0.4955	-9	**0.1019**	-9	**0.1305**	-9	0.157	-9	0.1568
SQ-5	-9	0.5592	-9	**0.2686**	-9	**0.2385**	-9	0.2695	-9	0.2511
SQ-6	-8	0.8313	-8	**0.2486**	-8	0.3379	-8	0.3383	-8	**0.2617**
SQ-7	-10	**3.0174**	-10	7.4938	-10	**3.0314**	-10	3.991	-10	3.6244
SQ-8	-14	16.4331	-14	**14.149**	-14	16.7108	-14	16.008	-14	**12.892**
SQ-9	-23	**35.1235**	-23	40.4940	-23	**31.5324**	-23	36.118	-23	41.535
SQ-10	-21	**16.2622**	-20	59.490	-21	**21.1052**	-21	21.177	-21	24.770
SQ-11	-42	**36.2809**	-38	110.518	-42	30.9691	-42	**29.061**	-42	30.969
SQ-12	-51	**134.211**	-51	306.391	-51	265.312	-51	**176.734**	-51	260.375
SQ-13	-46	**392.896**	-45	740.287	-46	1118.87	-46	**648.984**	-46	697.135
SQ-14	-46	**510.239**	-46	2030.86	-46	133.707	-46	163.859	-46	**128.516**
PROMEDIO		82.1882		236.807		116.245		78.6548		86.0734

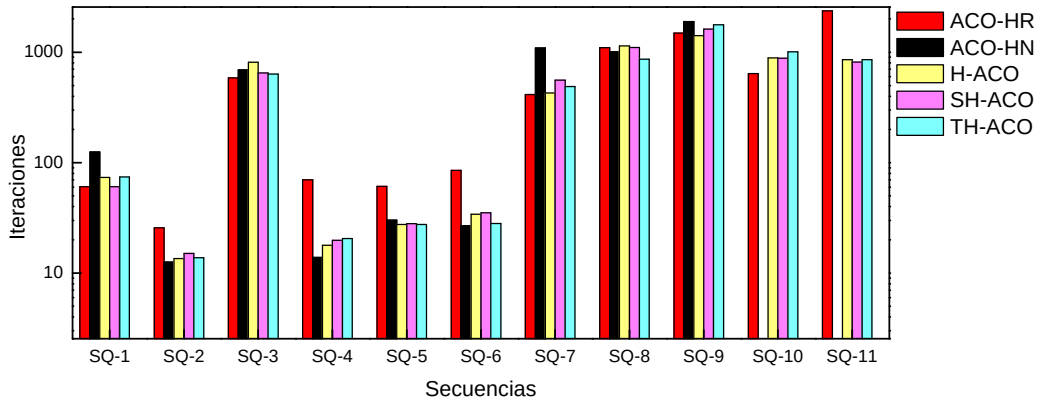
Tabla 6.4. Comparación de los algoritmos ACO individuales (ACO-HR y ACO-HN) y las versiones híbridas (H-ACO, SH-ACO y TH-ACO) de ACO. *%ex.* se refiere al porcentaje de veces que los algoritmos fueron capaces de encontrar el valor de *E*, mostrado en la tabla, del total de ejecuciones (100 ejecuciones para SQ-1 a SQ-8, 50 ejecuciones para SQ-9 a SQ-11 y 25 ejecuciones para SQ-12 a SQ-14). It_{max} es el número máximo de iteraciones por cada ejecución.

ID	It_{max}	ACO-HR		ACO-HN		H-ACO		SH-ACO		TH-ACO	
		E	**%ex.**	**E**	**%ex.**	**E**	**%ex.**	**E**	**%ex.**	**E**	**%ex.**
SQ-1	1000	-9	**100%**	-9	**100%**	-9	**100%**	-9	**100%**	-9	**100%**
SQ-2	1000	-8	**100%**	-8	**100%**	-8	**100%**	-8	**100%**	-8	**100%**
SQ-3	3000	-4	68%	-4	**82%**	-4	93%	-4	**100%**	-4	**100%**
SQ-4	1000	-9	**100%**	-9	**100%**	-9	**100%**	-9	**100%**	-9	**100%**
SQ-5	1000	-9	**100%**	-9	**100%**	-9	**100%**	-9	**100%**	-9	**100%**
SQ-6	1000	-8	**100%**	-8	**100%**	-8	**100%**	-8	**100%**	-8	**100%**
SQ-7	3000	-10	**98%**	-10	**82%**	-10	**100%**	-10	**100%**	-10	**100%**
SQ-8	3500	-14	78%	-14	**92%**	-14	96%	-14	**100%**	-14	98%
SQ-9	3500	-23	**80%**	-23	60%	-23	80%	-23	**88%**	-23	**88%**
SQ-10	3500	-21	**96%**	*-20*	60%	-21	98%	-21	**100%**	-21	98%
SQ-11	4000	-42	**98%**	*-38*	10%	-42	84%	-42	**96%**	-42	94%
SQ-12	20000	-51	**50%**	-51	46%	-51	52%	-51	**56%**	-51	**56%**
SQ-13	20000	-46	**46%**	*-45*	60%	-46	48%	-46	**52%**	-46	48%
SQ-14	20000	-46	40%	-46	**60%**	-46	56%	-46	**60%**	-46	**60%**
PROMEDIO			82.4%		75.1%		86.2%		89.4%		88.7%

Al observar el desempeño general de los algoritmos ACO híbridos podemos notar que presentan resultados más uniformes en todas las secuencias. Alcanzan el mínimo nivel de energía que los algoritmos individuales son capaces de encontrar en un número de iteraciones menor o intermedio (Fig. 6.1a). El tiempo de ejecución por iteración de los algoritmos híbridos no se ve afectado en comparación a los algoritmos individuales. Además, el porcentaje de éxito (cuando en una ejecución se encuentra la conformación de mínima energía) aumenta considerablemente en los algoritmos híbridos, como se muestra en la Fig. 6.1b. Los algoritmos SH-ACO y TH-ACO tienen un desempeño muy similar y, en comparación con H-ACO, estos dos

algoritmos tienen un mejor desempeño. Aunque en algunas secuencias H-ACO encontró soluciones en menos iteraciones que los otros dos algoritmos híbridos, SH-ACO y TH-ACO tuvieron el porcentaje de éxito más elevado. A partir de esto podemos concluir que se tiene un mejor desempeño con una estrategia de combinación de información de feromona que sólo correr los algoritmos de forma paralela sin que compartan información de este tipo.

a)



b)

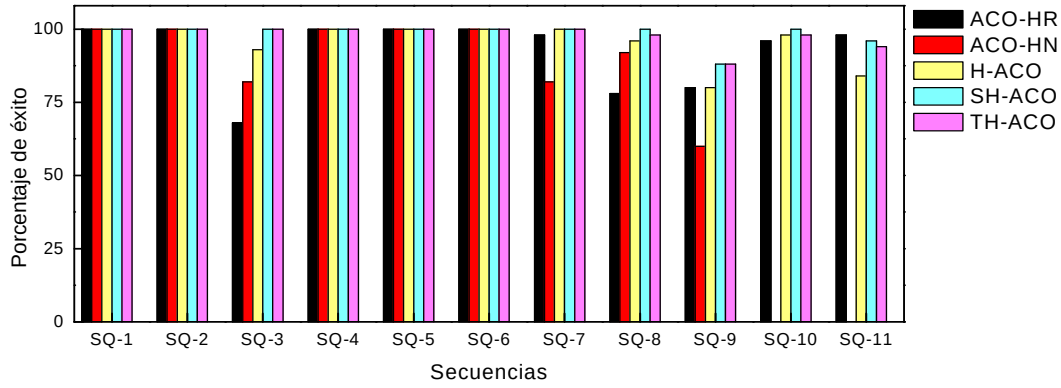


Fig. 6.1. a) Número de iteraciones promedio que les toma a los algoritmos encontrar una conformación de mínima energía para las secuencias SQ-1 a SQ-11 de la Tabla 6.2. b) Porcentaje de éxito de los algoritmos para encontrar una conformación de mínima energía.

Por otro lado, es difícil realizar una comparación exacta en cuanto al desempeño de los algoritmos presentados en la literatura con los presentados en esta tesis. El tiempo de ejecución puede variar dependiendo de las características de la máquina usada para realizar los experimentos y del lenguaje de programación que se utilice así como de la codificación. En varios trabajos sobre algoritmos genéticos no se reporta el tiempo de ejecución sino el número de conformaciones válidas revisadas durante cada búsqueda.

Algunos resultados mostrados en la literatura de distintos algoritmos aplicados a algunas secuencias de la Tabla 6.1 se muestran en la Tabla 6.5. Se muestran el algoritmo genético (GA) de Unger y Moult (1993a), el algoritmo Monte Carlo evolutivo (EMC) de Liang y Wong (2001), el método Rosenbluth de podado y enriquecimiento (PERM) presentado en (Ramakrishnan et al., 1997) y evaluado en (Shmygelska y Hoos, 2005), y el método ACO (ACO-HPPFP-3) de Shmygelska y Hoos (2005). En esta tabla podemos observar cómo el algoritmo genético (GA) y el Monte Carlo evolutivo (EMC) obtienen conformaciones óptimas para secuencias menores a 50 aminoácidos pero no así para secuencias de longitud mayor. PERM tiene un buen desempeño global, encontrando la energía óptima para todas las secuencias, aunque tarda un tiempo considerable en encontrar la solución para la secuencia SQ-11. ACO-HPPFP-3 también muestra un buen desempeño para la mayoría de las secuencias, encontrando la energía óptima para las secuencias de longitud menor a 100 aminoácidos.

La secuencia SQ-11 tiene una conformación óptima muy simétrica como se observa en la Fig. 6.2d. En (Hsu et al., 2003) se plantea que este tipo de conformaciones son difíciles de encontrar para cualquier algoritmo de crecimiento de cadena. Los algoritmos presentados en la literatura encuentran la conformación óptima para esta secuencia en 1.5 hrs. como mínimo (ACO-HPPFP-3) y otros no son capaces de encontrar tal conformación. PERM encuentra esta conformación en 78 horas. Los algoritmos presentados en esta tesis (ACO-HR y ACO híbridos) son

capaces de encontrar una conformación de mínima energía para esta secuencia en un tiempo promedio de 36 segundos o menos, lo que sobrepasa notablemente a otros algoritmos.

Tabla 6.5. Comparación del desempeño de varios algoritmos presentados en la literatura y los algoritmos presentados en esta tesis para el modelo HP-cuadrado. *E* se refiere al mínimo nivel de energía que los algoritmos fueron capaces de alcanzar, T_{avg} es el tiempo de CPU promedio requerido por los algoritmos para alcanzar *E*. Para GA y EMC, N_{vc} es el número de conformaciones válidas analizadas antes de que una conformación con valor *E* fuera encontrada. Los resultados presentados en PERM y ACO-HPPFP-3 están basados en 100-200 ejecuciones por secuencia realizadas en una máquina con procesador Pentium IV a 2.4 GHz, y 1 GB en RAM bajo Linux. En ACO-HPPFP-3 utilizan 100 hormigas en sus ejecuciones.

ID	GA		EMC		PERM		ACO-HPPFP-3	
	E	N_{vc}	E	N_{vc}	E	T_{avg}	E	$T_{avg.}$
SQ-4	-9	30492	-9	9374	-9	<1 seg.	-9	<1 seg.
SQ-5	-9	30491	-9	6929	-9	<1 seg.	-9	<1 seg.
SQ-6	-8	20400	-8	7202	-8	2 seg.	-8	<1 seg.
SQ-8	-14	301339	-14	12447	-14	<1 seg.	-14	4 seg.
SQ-9	-23	126547	-23	165791	-23	2 seg.	-23	1 min.
SQ-10	-21	592887	-21	74613	-21	3 seg.	-21	15 seg.
SQ-11	-37	187393	-39	564809	-42	78 hrs.	-42	1.5 hrs.
SQ-12			-52	44029	-53	1 min.	-53	24 hrs*
SQ-13					-48	8 min.	-47	10 hrs.
SQ-14					-50	1 hr.	-49	12 hrs.

* sólo el 20% de ejecuciones (Mencionado por los autores en (Shmygelska y Hoos, 2005))

ID	ACO-HR		ACO-HN		H-ACO		SH-ACO		TH-ACO	
	E	T_{avg}	E	T_{avg}	E	T_{avg}	E	T_{avg}	E	T_{avg}
SQ-4	-9	<1 seg	-9	<1 seg	-9	<1 seg	-9	<1 seg	-9	<1 seg
SQ-5	-9	<1 seg	-9	<1 seg	-9	<1 seg	-9	<1 seg	-9	<1 seg
SQ-6	-8	<1 seg	-8	<1 seg	-8	<1 seg	-8	<1 seg	-8	<1 seg
SQ-8	-14	16 seg.	-14	14 seg.	-14	16 seg.	-14	16 seg.	-14	13 seg.
SQ-9	-23	35 seg.	-23	40 seg.	-23	31 seg.	-23	36 seg.	-23	41 seg.
SQ-10	-21	16 seg.	-20	59 seg.	-21	21 seg.	-21	21 seg.	-21	24 seg.
SQ-11	-42	36 seg.	-38	2 min..	-42	31 seg.	-42	29 seg.	-42	31 seg.
SQ-12	-51	2 min.	-51	5 min.	-51	5 min.	-51	3 min.	-51	4 min.
SQ-13	-46	6 min.	-45	12 min.	-46	19 min.	-46	11 min.	-46	12 min.
SQ-14	-46	8 min.	-46	3 min.	-46	2 min.	-46	3 min.	-46	2 min.

En cuanto a las estructuras de menor longitud a SQ-11, el desempeño de los algoritmos presentados en esta tesis tienen un excelente desempeño comparado con el desempeño de los algoritmos de la Tabla 6.5, tomando en cuenta que el tiempo de ejecución de dichos algoritmos se midió en una computadora con un procesador a 2.4 GHz, mientras que el procesador de nuestra máquina de referencia tiene solo 1.5 GHz. Además, la memoria de la computadora con la que se midieron los algoritmos de la Tabla 6.5 tiene 4 veces mayor capacidad que la memoria de nuestra máquina de referencia. Para las secuencias de tamaño mayor (SQ-12, SQ-13 y SQ-14) los algoritmos presentados en esta tesis no consiguieron encontrar las conformaciones de mínima energía conocidas, aunque obtienen conformaciones sub-óptimas en una cantidad de iteraciones razonable, desde 2000, en el mejor de los casos, hasta 29000 en el peor de los casos.

Algunas de las conformaciones encontradas por los algoritmos presentados en esta tesis se muestran en la Fig. 6.2. En estas figuras podemos apreciar cómo los aminoácidos *H* tienden a agruparse mientras que los aminoácidos *P* permanecen en las orillas, lo que muestra la propiedad mencionada en el capítulo 2, la cual nos menciona que los aminoácidos con un residuo hidrofóbico (H) tienden a agruparse y encontrarse en el interior de la proteína formando un núcleo, mientras que aquellos aminoácidos con un residuo hidrofílico (P) tienden a establecerse en el exterior o superficie de la molécula de la proteína.

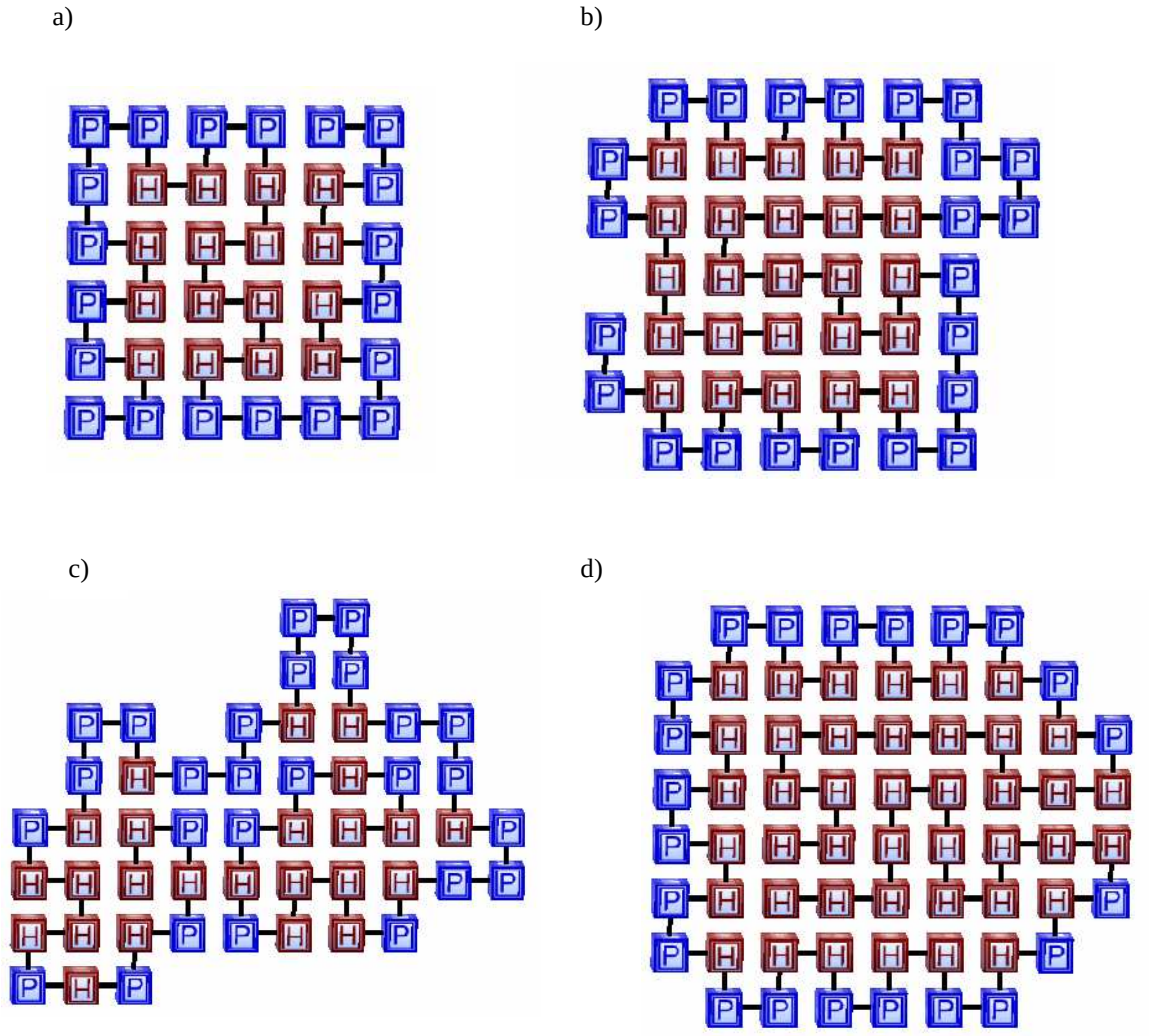


Fig. 6.2. Ejemplo de conformaciones encontradas por los algoritmos presentados en esta tesis. a) SQ-8, b) SQ-9, c) SQ-10 y d) SQ-11.

6.2 Resultados de las secuencias estándar del modelo HP triangular 2D

Para evaluar los algoritmos presentados en el capítulo 5 para el modelo HP triangular 2D se utilizaron las secuencias mostradas en la Tabla 6.6, las cuales también han sido utilizadas en la literatura (Krasnogor et al., 2002; Santos y Santos, 2004).

Los espacios vacíos en la columna de energía óptima de la Tabla 6.6 indican que para dichas secuencias no se conoce el valor de energía óptimo todavía.

Los experimentos realizados sobre estas secuencias consistieron en ejecuciones independientes para cada secuencia (100 para las secuencias ST-1 a ST-17, 50 para las secuencias ST-18 a ST-26 y 20 para las secuencias ST-27 a ST-30) y para cada algoritmo. Los parámetros de los algoritmos utilizados en todos los experimentos fueron: $\alpha = 1$, $\beta = 2$, $\rho = 0.8$, $\theta = 0.05$ y $\mu = 0.2$.

Tabla 6.6. Secuencias del modelo HP triangular 2D.

ID	Tamaño	Energía óptima	Secuencia proteínica
ST-1	12	-11	HHPHPHPHPHPH
ST-2	14	-11	HHPPHPHPHPHPHP
ST-3	14	-11	HHPPHPPHPHPHPH
ST-4	16	-11	HHPHPPHPPHPPHPPH
ST-5	16	-11	HHPPHPPHPHPHPPHP
ST-6	17	-11	HHPPHPPHPPHPPHPPH
ST-7	17	-17	HHPHPHPHPHPHPHPHH
ST-8	20	-15	HPHPPHHPHPPHPHHPPHPH
ST-9	20	-17	HHHPPHPHPHPPHPHPHPPH
ST-10	20	-17	HHPPHPPHPHPHPPHPHPHH
ST-11	20	-17	HHPHPHPHPHPPHPPHPPHH
ST-12	21	-17	HHPPHPPHPHPPHPHPPHPHH
ST-13	21	-17	HHPHPPHPPHPHPHPPHPPHH
ST-14	21	-17	HHPPHPHPHPPHPHPPHPPHH
ST-15	22	-17	HHPPHPPHPHPHPPHPPHPPHH
ST-16	24	-17	HHPPHPPHPPHPPHPPHPPHPPHH
ST-17	24	-17	HHPPHPPHPPHPPHPPHPPHPPHH
ST-18	23	-25	HHHPHPHPHPHPHPHPHPHPHHH
ST-19	24	-25	HHHPHPHPPHPHPHPHPHPHPHHH
ST-20	24	-25	HHHPHPHPHPPHPHPHPHPHPHHH
ST-21	30	-25	HHHPPHPPHPPHPPHPHPPHPHPPHPPHHH
ST-22	30	-25	HHHPPHPPHPPHPHPPHPHPPHPPHPPHHH
ST-23	36		PPPHHPPHHPPPPPHHHHHHHPPHHPPPPHHPPHPP
ST-24	37	-29	HHHPPHPPHPPHPHPHPPHPPHPPHPPPPPHPHPHHH
ST-25	45		PHHHPHHHPPPHPHHPHHPPHPHHHHPHPPHHHHHP HPHHPPHHP
ST-26	48		PPHPPHHPPHHPPPPPHHHHHHHHHHPPPPPPHHPPH HPPHPPHHHHH
ST-27	50		HHPHPHPHPHHHHPHPPPHPPPHPPPPHPPPHPPPHP HHHHPHPHPHPHH
ST-28	57		HPHHHPHHHPPHHPHPHHPHHHPHPHPHHPPHHHPP HPHPPPPHPPHPPHHPPHPPH
ST-29	60		PPHHHPHHHHHHHHPPPHHHHHHHHHHPHPPPHHH HHHHHHHHHPPPPHHHHHHPHHPHP
ST-30	64		HHHHHHHHHHHHPHPHPPHHPPHHPPHPPHHPPHH PPHPPHHPPHHPPHPHPHHHHHHHHHHHH

En los resultados de las tablas 6.7, 6.8 y 6.9 podemos observar que los diferentes algoritmos tienen un desempeño similar para las secuencias de longitud menor. El algoritmo ACO-HR tiene problemas en encontrar una conformación de mínima energía para las secuencias ST-24 y ST-29 mientras que algoritmo ACO-HN encuentra la solución a dichas secuencias de manera efectiva. En contraparte, el algoritmo ACO-HN tiene problemas en encontrar una conformación de mínima energía para las secuencias ST-28 y ST-30, mientras que el algoritmo ACO-HR las encuentra de forma eficiente.

También podemos ver cómo los algoritmos ACO híbridos equilibran su desempeño con todas las secuencias, esto es, son capaces de encontrar una conformación óptima para todas las secuencias de la Tabla 6.6. Este comportamiento es similar a aquel observado en los experimentos con el modelo HP cuadrado. Si observamos las Tablas 6.7 y 6.8, podemos observar que en promedio, el algoritmo que tiene el número de iteraciones y el tiempo de ejecución es el algoritmo ACO-HR. Sin embargo, si tomamos en cuenta que el algoritmo ACO-HR no fue capaz de encontrar la solución óptima para dos secuencias, y además, observamos el porcentaje de éxito mostrado en la Tabla 6.9, podemos decir que los algoritmos híbridos tienen mejor desempeño que el algoritmo ACO-HR, sobre todo el algoritmo SH-ACO.

A partir de lo anterior podemos decir que al combinar los dos algoritmos individuales se consiguen combatir las deficiencias de cada uno de estos produciendo un mejor desempeño global en la búsqueda de soluciones en términos de menor cantidad de iteraciones necesarias para encontrar soluciones de mínima energía un mayor porcentaje de veces.

Tabla 6.7. Comparación del desempeño de los algoritmos ACO individuales (ACO-HR y ACO-HN) y las versiones híbridas (H-ACO, SH-ACO y TH-ACO) de ACO para el modelo triangular. *E* se refiere al mínimo nivel de energía que los algoritmos fueron capaces de alcanzar, I_{avg} es el número promedio de iteraciones requeridas por los algoritmos para alcanzar *E*.

ID	ACO-HR		ACO-HN		H-ACO		SH-ACO		TH-ACO	
	E	**$I_{avg.}$**	**E**	**$I_{avg.}$**	**E**	**$I_{avg.}$**	**E**	**$I_{avg.}$**	**E**	**$I_{avg.}$**
ST-1	-11	97.45	-11	**19.1**	-11	66.19	-11	**26.65**	-11	34.07
ST-2	-11	174.86	-11	**41.76**	-11	166.58	-11	82.72	-11	**77.86**
ST-3	-11	353.709	-11	**39.86**	-11	151.65	-11	**85.31**	-11	92.84
ST-4	-11	38.55	-11	**4.74**	-11	19.28	-11	9.28	-11	**7.98**
ST-5	-11	318.53	-11	**39.55**	-11	168.63	-11	**54.85**	-11	73.39
ST-6	-11	97.22	-11	**7.28**	-11	24.93	-11	**14.14**	-11	16.59
ST-7	-17	877.56	-17	**129.28**	-17	646.95	-17	217.35	-17	**197.80**
ST-8	-15	12.08	-15	**5.93**	-15	11.83	-15	**9.69**	-15	10.69
ST-9	-17	1250.06	-17	**244.38**	-17	545.39	-17	396.52	-17	**316.32**
ST-10	-17	312.92	-17	**197.96**	-17	290.31	-17	287.15	-17	**188.59**
ST-11	-17	641.13	-17	**293.98**	-17	748.01	-17	**570.65**	-17	596.45
ST-12	-17	**44.45**	-17	47.92	-17	49.97	-17	**45.57**	-17	52.33
ST-13	-17	**40.82**	-17	43.71	-17	69.03	-17	**47.04**	-17	48.39
ST-14	-17	**67.8**	-17	133.86	-17	118.68	-17	120.36	-17	**109.80**
ST-15	-17	**35.95**	-17	55.27	-17	56.30	-17	52.07	-17	**51.57**
ST-16	-17	**47.36**	-17	71.69	-17	79.47	-17	70.88	-17	**56.84**
ST-17	-17	**57.83**	-17	75.02	-17	69.56	-17	65.02	-17	**52.83**
ST-18	-25	1915	-25	**992.96**	-25	**1044.89**	-25	1176.5	-25	1249.51
ST-19	-25	1905.5	-25	**792.57**	-25	1450.25	-25	**1021.5**	-25	1200.51
ST-20	-25	1833	-25	**1268.67**	-25	1436.38	-25	1201.95	-25	**1090.67**
ST-21	-25	**427.24**	-25	820.94	-25	**586.66**	-25	587	-25	638.875
ST-22	-25	**367.12**	-25	620.95	-25	508.84	-25	**354.46**	-25	504.58
ST-23	-24	3454	-24	**2743**	-24	3411.22	-24	**3230.5**	-24	3371.72
ST-24	*-28*	*255.272*	-29	**3812**	-29	3729.5	-29	**3145.8**	-29	3368.52
ST-25	-42	**663.5**	-42	**1372.19**	-42	1346.6	-42	**1332.04**	-42	1338.17
ST-26	-40	1947.88	-40	**1256.47**	-40	1523.99	-40	**1325.6**	-40	1763.5
ST-27	-41	**2733**	-41	4967.43	-41	4978.4	-41	4495	-41	**4378.9**
ST-28	-49	**1568.4**	*-48*	*4533*	-49	3548.7	-49	2362	-49	**2178.22**
ST-29	*-70*	*1071.8*	-71	**2667**	-71	3150.6	-71	2950.6	-71	**2845.33**
ST-30	-74	**1500.5**	*-71*	*4834.91*	-74	2865.8	-74	**683**	-74	2398.6
PROMEDIO		803.683		1071.11		1095.49		867.37		943.715

Tabla 6.8. Comparación del desempeño de los algoritmos ACO individuales (ACO-HR y ACO-HN) y las versiones híbridas (H-ACO, SH-ACO y TH-ACO) de ACO para el modelo triangular. *E* se refiere al mínimo nivel de energía que los algoritmos fueron capaces de alcanzar, T_{avg} es el tiempo promedio de CPU, dado en segundos, requerido por los algoritmos para alcanzar *E*.

ID	ACO-HR		ACO-HN		H-ACO		SH-ACO		TH-ACO	
	E	T_{avg}	E	T_{avg}	E	T_{avg}	E	$T_{avg.}$	E	$T_{avg.}$
ST-1	-11	0.5426	-11	**0.1091**	-11	0.3655	-11	**0.1431**	-11	0.2001
ST-2	-11	**1.1545**	-11	0.2617	-11	1.0259	-11	0.5416	-11	**0.5073**
ST-3	-11	2.3462	-11	**0.2571**	-11	0.9389	-11	**0.5485**	-11	0.6071
ST-4	-11	0.3027	-11	**0.0367**	-11	0.1472	-11	0.0818	-11	**0.0628**
ST-5	-11	2.4511	-11	**0.2902**	-11	1.2265	-11	**0.4267**	-11	0.5640
ST-6	-11	0.7995	-11	**0.0592**	-11	0.1989	-11	**0.1227**	-11	0.1426
ST-7	-17	7.8303	-17	**1.0586**	-17	5.2632	-17	1.8176	-17	**1.6893**
ST-8	-15	0.1329	-15	**0.0663**	-15	0.1198	-15	**0.1026**	-15	0.1111
ST-9	-17	13.0885	-17	**2.3997**	-17	5.2988	-17	4.0172	-17	**3.2426**
ST-10	-17	3.3077	-17	**1.965**	-17	2.83069	-17	2.9121	-17	**1.9299**
ST-11	-17	6.6786	-17	**2.9407**	-17	7.3571	-17	**5.7398**	-17	6.1537
ST-12	-17	**0.5045**	-17	0.5287	-17	0.5239	-17	**0.4894**	-17	0.5665
ST-13	-17	0.4553	-17	**0.4547**	-17	0.7256	-17	**0.5075**	-17	0.5229
ST-14	-17	**0.7502**	-17	1.3779	-17	1.2491	-17	1.2869	-17	**1.1909**
ST-15	-17	**0.4229**	-17	0.5987	-17	0.6343	-17	0.5906	-17	**0.5886**
ST-16	-17	**0.613**	-17	0.8695	-17	0.9893	-17	0.8786	-17	**0.7204**
ST-17	-17	**0.7489**	-17	0.9163	-17	0.8556	-17	0.8264	-17	**0.6675**
ST-18	-25	24.1715	-25	**11.8003**	-25	**12.6506**	-25	14.4552	-25	15.7513
ST-19	-25	25.1797	-25	**10.127**	-25	18.3455	-25	**13.2726**	-25	23.865
ST-20	-25	24.284	-25	**16.1179**	-25	18.1466	-25	15.4798	-25	**14.3478**
ST-21	-25	**6.8782**	-25	13.4395	-25	**9.6498**	-25	9.8522	-25	10.9591
ST-22	-25	**4.7943**	-25	10.2898	-25	8.4149	-25	**5.9849**	-25	8.5972
ST-23	-24	**27.469**	-24	56.8373	-24	68.9543	-24	**66.504**	-24	71.8835
ST-24	*-28*	*5.6646*	-29	**80.734**	-29	76.937	-29	**70.359**	-29	74.938
ST-25	-42	**20.0729**	-42	39.1898	-42	39.109	-42	**38.12**	-42	39.145
ST-26	-40	62.6978	-40	**38.792**	-40	47.047	-40	**41.797**	-40	53.725
ST-27	-41	**93.2209**	-41	162.481	-41	163.574	-41	**148.77**	-41	157.185
ST-28	-49	**67.157**	*-48*	*180.719*	-49	143.834	-49	96.828	-49	**92.750**
ST-29	*-70*	*51.0278*	-71	**119.907**	-71	144.072	-71	137.282	-71	**136.546**
ST-30	-74	**79.1473**	*-71*	*236.503*	-74	143.886	-74	**34.828**	-74	126.766
PROMEDIO		17.7965		33.0376		30.8123		23.8189		28.1975

Tabla 6.9. Comparación de los algoritmos ACO individuales (ACO-HR y ACO-HN) y las versiones ACO-híbridas (H-ACO, SH-ACO y TH-ACO) para el modelo triangular. *%ex.* Se refiere al porcentaje de veces que los algoritmos fueron capaces de encontrar *E* del total de ejecuciones (100 ejecuciones para ST-1 a ST-17, 50 ejecuciones para ST-18 a ST-30). It_{max} es el número máximo de iteraciones por cada ejecución.

ID	It_{max}	ACO-HR		ACO-HN		H-ACO		SH-ACO		TH-ACO	
		E	%ex.	E	%ex.	E	%ex.	E	%ex.	E	%ex.
ST-1	1000	-11	**100%**	-11	**100%**	-11	**100%**	-11	**100%**	-11	**100%**
ST-2	1000	-11	**100%**	-11	**100%**	-11	**100%**	-11	**100%**	-11	**100%**
ST-3	1000	-11	**100%**	-11	**100%**	-11	**100%**	-11	**100%**	-11	**100%**
ST-4	1000	-11	**100%**	-11	**100%**	-11	**100%**	-11	**100%**	-11	**100%**
ST-5	1000	-11	**100%**	-11	**100%**	-11	**100%**	-11	**100%**	-11	**100%**
ST-6	1000	-11	**100%**	-11	**100%**	-11	**100%**	-11	**100%**	-11	**100%**
ST-7	3000	-17	85%	-17	**100%**	-17	**100%**	-17	**100%**	-17	**100%**
ST-8	1000	-15	**100%**	-15	**100%**	-15	**100%**	-15	**100%**	-15	**100%**
ST-9	3000	-17	85%	-17	**100%**	-17	98%	-17	**99%**	-17	**99%**
ST-10	3000	-17	82%	-17	**100%**	-17	**100%**	-17	**100%**	-17	**100%**
ST-11	3000	-17	83%	-17	**99%**	-17	**99%**	-17	**99%**	-17	98%
ST-12	3000	-17	**100%**	-17	**100%**	-17	**100%**	-17	**100%**	-17	**100%**
ST-13	3000	-17	**100%**	-17	**100%**	-17	**100%**	-17	**100%**	-17	**100%**
ST-14	3000	-17	92%	-17	**100%**	-17	**100%**	-17	**100%**	-17	**100%**
ST-15	3000	-17	98%	-17	**100%**	-17	**100%**	-17	**100%**	-17	**100%**
ST-16	3000	-17	98%	-17	**100%**	-17	**100%**	-17	**100%**	-17	**100%**
ST-17	3000	-17	98%	-17	**100%**	-17	**100%**	-17	**100%**	-17	**100%**
ST-18	3500	-25	4%	-25	**96%**	-25	76%	-25	**96%**	-25	84%
ST-19	3500	-25	8%	-25	**86%**	-25	76%	-25	80%	-25	**82%**
ST-20	3500	-25	12%	-25	**84%**	-25	74%	-25	**82%**	-25	**82%**
ST-21	3500	-25	**100%**	-25	90%	-25	**100%**	-25	**100%**	-25	**100%**
ST-22	3500	-25	**100%**	-25	90%	-25	**100%**	-25	**100%**	-25	**100%**
ST-23	3500	-24	8%	-24	**60%**	-24	**100%**	-24	**100%**	-24	**100%**
ST-24	4000	*-28*	*98%*	-29	**6%**	-29	4%	-29	**6%**	-29	**6%**
ST-25	5000	-42	90%	-42	**100%**	-42	**100%**	-42	**100%**	-42	**100%**
ST-26	5000	-40	90%	-40	**100%**	-40	**100%**	-40	**100%**	-40	**100%**
ST-27	5000	-41	8%	-41	**48%**	-41	46%	-41	**50%**	-41	**50%**
ST-28	7000	-49	**75%**	*-48*	*60%*	-49	82%	-49	**85%**	-49	84%
ST-29	7000	*-70*	*6%*	-71	**12%**	-71	8%	-71	**12%**	-71	**12%**
ST-30	7000	-74	**60%**	*-71*	*85%*	-74	56%	-74	**60%**	-74	58%
PROMEDIO			76%		87.2%		87.2%		88.7%		88.5%

El modelo HP triangular 2D no ha sido tan ampliamente estudiado como lo ha sido el modelo HP cuadrado para el que existen numerosos algoritmos en la literatura.

Dos algoritmos que han sido aplicados al modelo HP triangular son el algoritmo multimemético de Krasnogor et al. (2002) y el algoritmo genético de Santos y Santos (2004). Los resultados presentados en estos trabajos se muestran en la Tabla 6.10.

Aunque Krasnogor et al. y Santos et al. sólo reportan la energía mínima que sus algoritmos fueron capaces de encontrar para diferentes secuencias, podemos observar que los algoritmos presentados en esta tesis consiguen encontrar en todos los casos conformaciones con un nivel de energía igual o menor que la que se había reportado hasta ese momento. Conforme la longitud de las secuencias es mayor, esta diferencia se vuelve más visible. Por ejemplo, para la secuencia ST-30 se consigue encontrar una conformación con un nivel de energía de -74 mientras que el nivel de energía que había sido reportado en la literatura era de sólo -54.

Estos resultados muestran a ACO como una opción prometedora para aplicarse en el problema del plegamiento de proteínas con modelos simplificados como los presentados en esta tesis, o incluso más complejos: con más grados de libertad o con más tipos de aminoácidos.

Tabla 6.10. Comparación del desempeño de varios algoritmos para el modelo HP triangular 2D presentados en la literatura y los algoritmos presentados en esta tesis. Para cada algoritmo se muestra el mínimo nivel de energía que dichos algoritmos fueron capaces de encontrar para las diferentes secuencias.

ID	MMA	GA*	ACO -HR	ACO -HN	H-ACO	SH-ACO	TH-ACO
ST-1	**-11**	**-11**	**-11**	**-11**	**-11**	**-11**	**-11**
ST-2	**-11**	**-11**	**-11**	**-11**	**-11**	**-11**	**-11**
ST-3	**-11**	**-11**	**-11**	**-11**	**-11**	**-11**	**-11**
ST-4	**-11**	**-11**	**-11**	**-11**	**-11**	**-11**	**-11**
ST-5	**-11**	**-11**	**-11**	**-11**	**-11**	**-11**	**-11**
ST-6	**-11**	**-11**	**-11**	**-11**	**-11**	**-11**	**-11**
ST-7	**-17**	**-17**	**-17**	**-17**	**-17**	**-17**	**-17**
ST-8		**-15**	**-15**	**-15**	**-15**	**-15**	**-15**
ST-9		**-17**	**-17**	**-17**	**-17**	**-17**	**-17**
ST-10	**-17**	-16	**-17**	**-17**	**-17**	**-17**	**-17**
ST-11	**-17**	-15	**-17**	**-17**	**-17**	**-17**	**-17**
ST-12	**-17**	-15	**-17**	**-17**	**-17**	**-17**	**-17**
ST-13	**-17**	-15	**-17**	**-17**	**-17**	**-17**	**-17**
ST-14	**-17**	-16	**-17**	**-17**	**-17**	**-17**	**-17**
ST-15	**-17**	-16	**-17**	**-17**	**-17**	**-17**	**-17**
ST-16	-16	-15	**-17**	**-17**	**-17**	**-17**	**-17**
ST-17		-14	**-17**	**-17**	**-17**	**-17**	**-17**
ST-18	**-25**	-24	**-25**	**-25**	**-25**	**-25**	**-25**
ST-19	**-25**	-21	**-25**	**-25**	**-25**	**-25**	**-25**
ST-20	**-25**	-24	**-25**	**-25**	**-25**	**-25**	**-25**
ST-21	-24	-18	**-25**	**-25**	**-25**	**-25**	**-25**
ST-22	-24	-20	**-25**	**-25**	**-25**	**-25**	**-25**
ST-23		-23	**-24**	**-24**	**-24**	**-24**	**-24**
ST-24	-26	-21	-28	**-29**	**-29**	**-29**	**-29**
ST-25		-36	**-42**	**-42**	**-42**	**-42**	**-42**
ST-26		-37	**-40**	**-40**	**-40**	**-40**	**-40**
ST-27		-32	**-41**	**-41**	**-41**	**-41**	**-41**
ST-28		-38	**-49**	-48	**-49**	**-49**	**-49**
ST-29		-62	-70	**-71**	**-71**	**-71**	**-71**
ST-30		-54	**-74**	-71	**-74**	**-74**	**-74**

*ejecuciones con un máximo de 100-200 generaciones.

Algunos ejemplos de las conformaciones encontradas por los algoritmos presentados en esta tesis para las secuencias del modelo HP triangular se muestran en la Fig. 6.3.

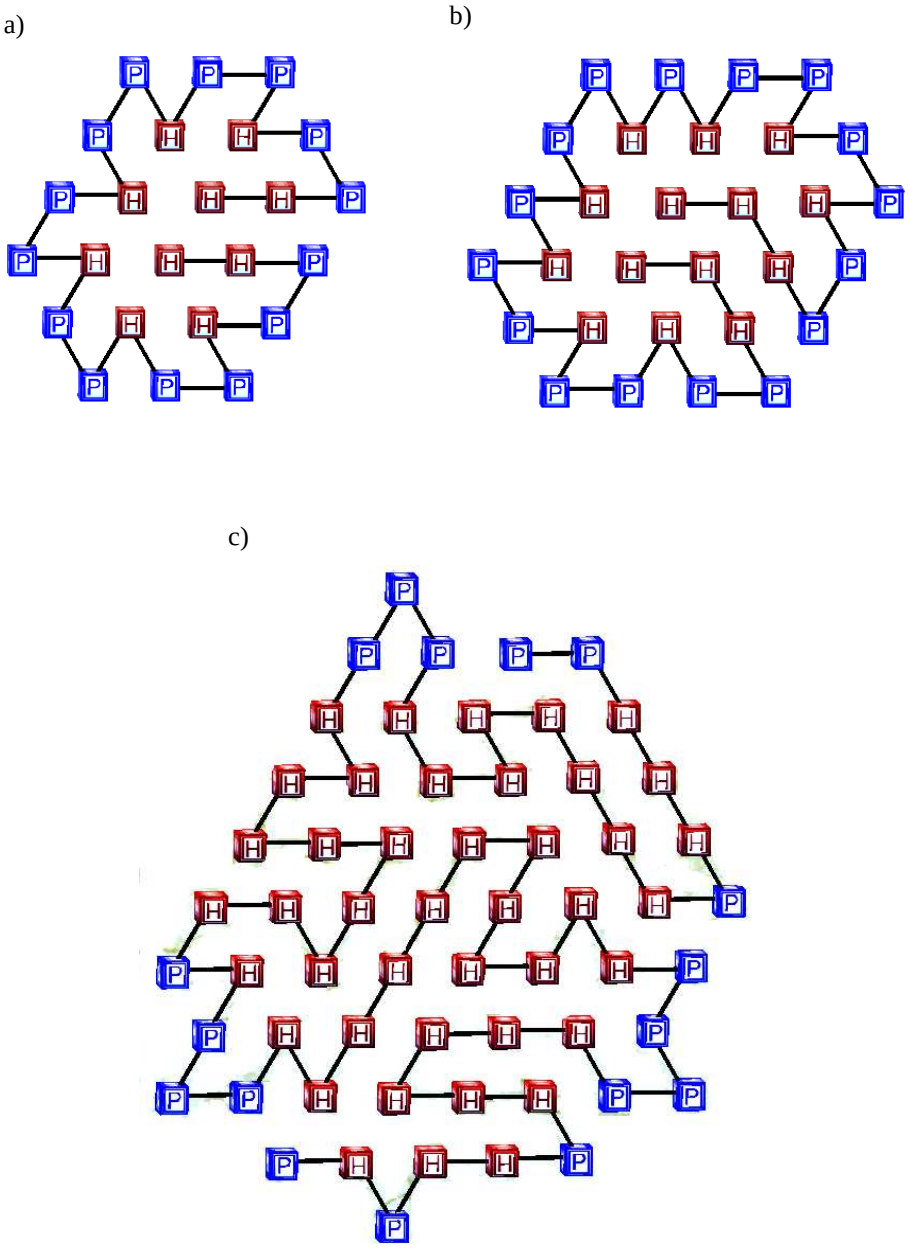


Fig. 6.3. Ejemplo de conformaciones encontradas por los algoritmos presentados en esta tesis para secuencias del modelo HP triangular: a) ST-17, b) ST-22, y c) ST-29.

6.3 Pruebas en el suavizado de feromona

Un elemento importante en los algoritmos presentados es la parte de suavizado de feromona empleada para evitar estancamiento en los valores de la feromona y por lo tanto caer en mínimos locales. Para mostrar la eficacia de dicho método se realizaron experimentos en diferentes secuencias de los modelos HP cuadrado y triangular 2D. Dichos experimentos consistieron en ejecutar los algoritmos eliminando la parte correspondiente al método de suavizado y comparar los resultados obtenidos sin el uso de suavizado con los resultados que se obtienen cuando se ejecutan los algoritmos usando el suavizado. Se realizaron 20 ejecuciones independientes para cada secuencia y en los diferentes algoritmos utilizados para evaluar la estrategia de suavizado. Los resultados de dichos experimentos se muestran en la Fig. 6.4 y Fig. 6.5.

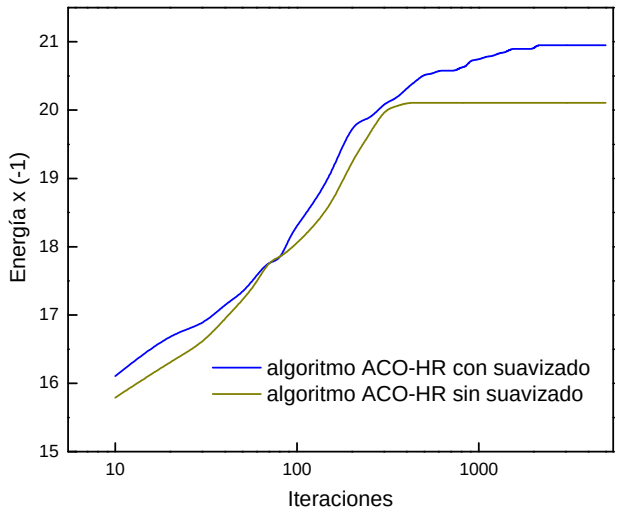


Fig. 6.4 Valor de la función de energía con el aumento de las iteraciones de la secuencia SQ-10 entre el algoritmo ACO-HR y la variante del mismo algoritmo que no utiliza el método de suavizado de feromona.

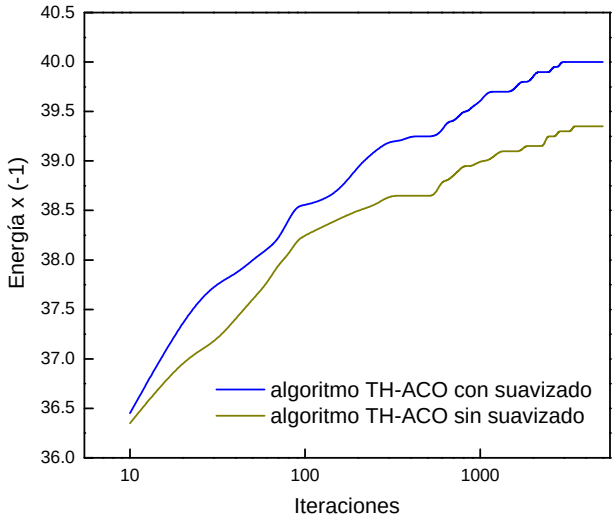


Fig. 6.5 Valor de la función de energía con el aumento de las iteraciones de la secuencia ST-26 entre el algoritmo TH-ACO y la variante del mismo algoritmo que no utiliza el método de suavizado de feromona.

Como podemos observar en las gráficas, el desempeño de los algoritmos es mejor cuando se incluye el paso de suavizado de feromona que cuando se omite dicho paso. En la Fig. 6.4 se nota más claramente que cuando en el algoritmo se omite el suavizado de feromona se llega a un punto en el cual aunque el algoritmo siga iterando no existe mejoría alguna en las soluciones, es decir hay estancamiento, y al continuar reforzando una y otra vez los mismos caminos, las hormigas tienden a construir una y otra vez las mismas conformaciones (mínimos locales) resultando muy difícil que salgan de tal estado. Cuando se aplica el método de suavizado de feromona el algoritmo consigue salir de dicho estancamiento y continuar con la búsqueda de nuevas conformaciones.

El método de suavizado de feromona, debido a su simplicidad, no afecta el tiempo de ejecución de los algoritmos. Esta es una gran ventaja en comparación con el uso de búsquedas locales, las cuales suelen ser muy lentas en ejecución.

6.4 Pruebas en la estrategia para resolver el traslape

Otra parte importante que mejora el desempeño de los algoritmos presentados en esta tesis es la estrategia para resolver el traslape. Para probar cómo la estrategia propuesta disminuye el tiempo de ejecución y ayuda con la búsqueda de soluciones se realizó la siguiente prueba. Utilizando el algoritmo ACO-HR con el modelo HP cuadrado, se modificó el módulo de traslape propuesto en esta tesis y se implementó la estrategia propuesta por Shmygelska y Hoos (2005). Esta propuesta consiste en que en el momento de encontrar el traslape, se despliegue la mitad de la conformación que se ha plegado hasta ese momento y después se vuelva a plegar de forma completamente aleatoria hasta encontrar una conformación parcial válida. Se realizaron 20 ejecuciones independientes utilizando la secuencia SQ-8. En la Fig. 6.6 se muestran los resultados obtenidos en estas pruebas que muestran la calidad de la solución encontrada utilizando en el algoritmo las dos diferentes estrategias para solucionar el traslape.

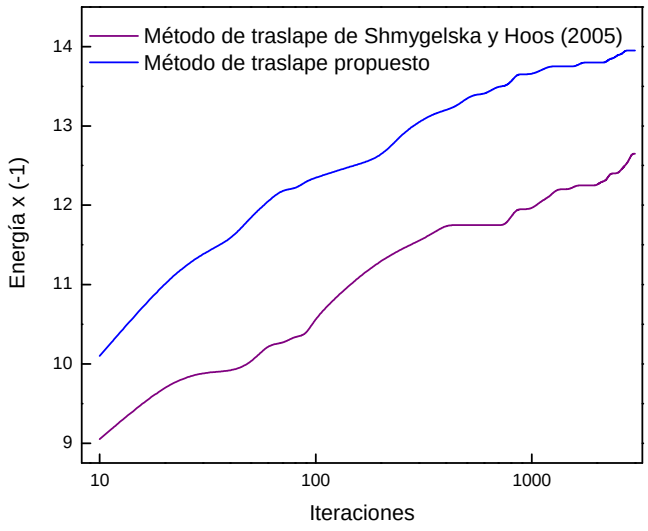


Fig. 6.6. Valor de la función de energía con el aumento de iteraciones para la secuencia SQ-8 evaluada con el algoritmo ACO-HR y diferentes estrategias para solucionar traslape.

De la Fig. 6.6 podemos ver cómo al utilizar la estrategia que nosotros proponemos se consigue aumentar la calidad de las conformaciones encontradas. Además, el tiempo de ejecución se reduce considerablemente, como podemos observar en la Fig. 6.7.

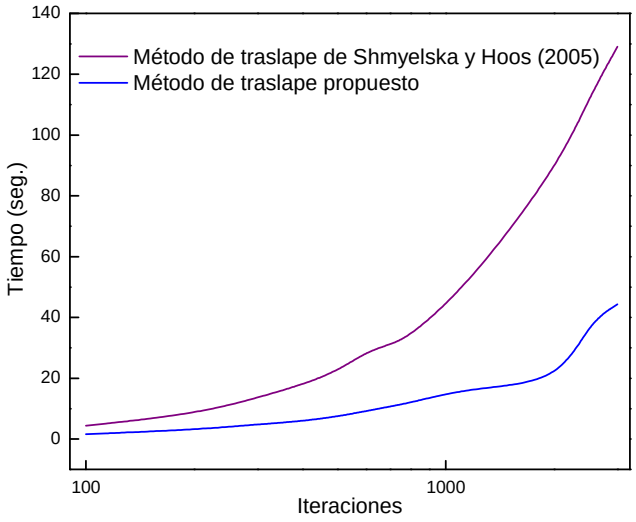


Fig. 6.7. Tiempo de ejecución para la secuencia SQ-8 evaluada con el algoritmo ACO-HR y diferentes estrategias para solucionar traslape.

6.5 Pruebas variando el tamaño de la colonia de hormigas

Otro factor importante en el desempeño de los algoritmos es el tamaño de la colonia de hormigas, es decir el número de hormigas utilizadas en cada iteración para la construcción de soluciones. Para estudiar el efecto de la variación del tamaño de la colonia de hormigas en la búsqueda de soluciones se realizaron varios experimentos fijando la cantidad de hormigas en 50, 100 y 200 hormigas. Los experimentos se realizaron en la secuencia SQ-10 del modelo cuadrado utilizando el algoritmo ACO-HR y en la secuencia ST-26 del modelo triangular 2D utilizando el algoritmo TH-ACO.

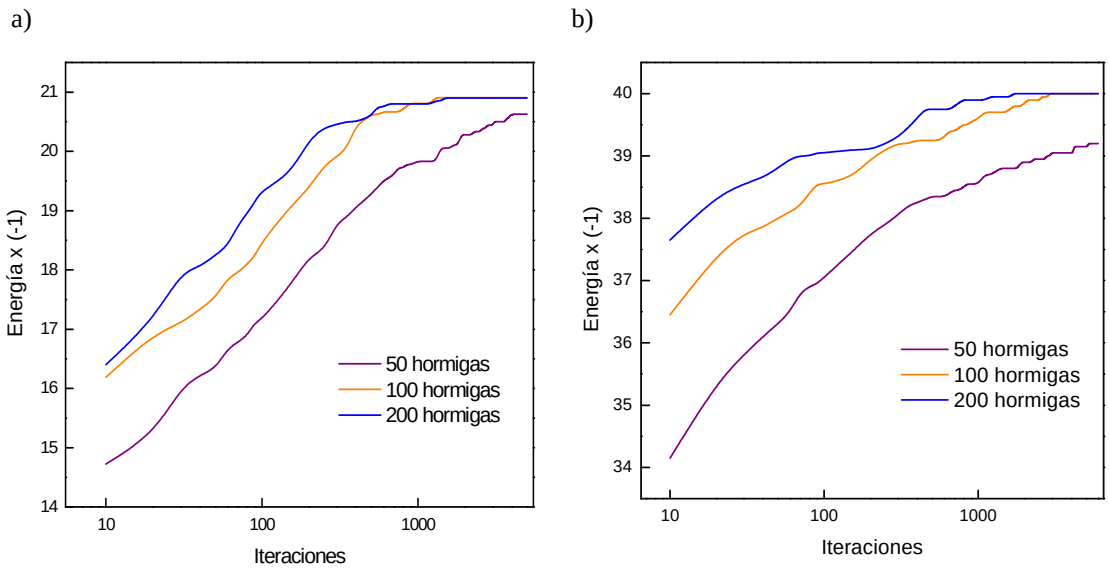


Fig. 6.8. Pruebas en el desempeño de los algoritmos variando la cantidad de hormigas de la colonia: a) algoritmo ACO-HR y secuencia SQ-10 del modelo HP cuadrado y b) algoritmo TH-ACO y secuencia ST-26 del modelo HP triangular 2D.

Los resultados de estos experimentos se muestran en la Fig. 6.8 y la Fig. 6.9. Como se puede observar en las figuras, al utilizar un menor número de hormigas se requiere una mayor cantidad de iteraciones para que los algoritmos encuentren una conformación de mínima energía, y conforme se aumenta la cantidad de hormigas se encuentran conformaciones con menor nivel de energía en las primeras iteraciones. Sin embargo, el tiempo de ejecución por cada iteración aumenta considerablemente al aumentar la cantidad de hormigas. En la Fig. 6.8 y Fig 6.9 se puede observar que con 100 hormigas se obtiene un buen equilibrio entre el tiempo de ejecución y el número de iteraciones necesarias para encontrar soluciones de mínima energía en los algoritmos.

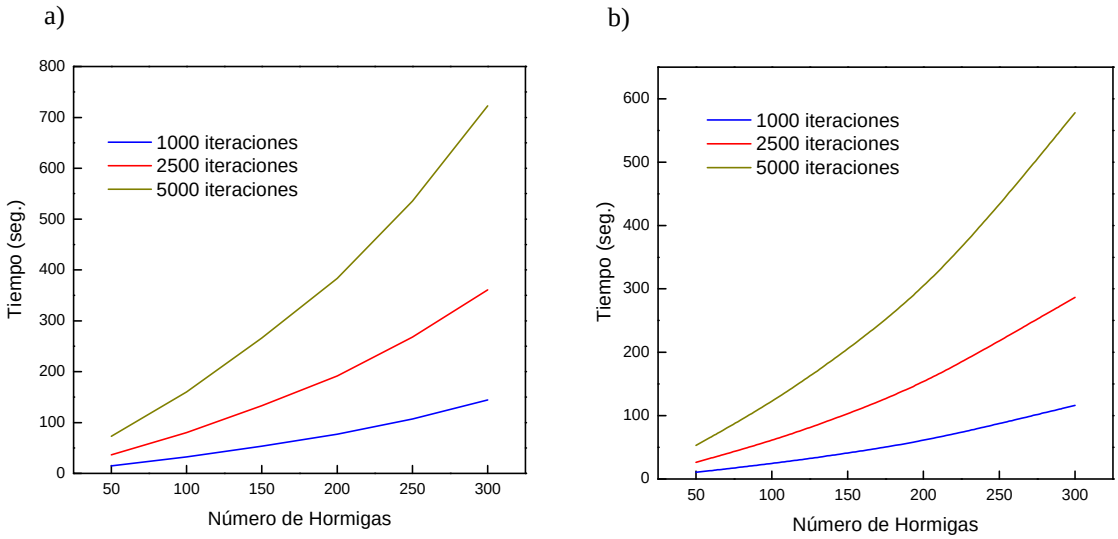


Fig. 6.9. Efectos en el tiempo de ejecución de los algoritmos durante 1000, 2500 y 5000 iteraciones cuando se varía la cantidad de hormigas de la colonia para: a) el algoritmo ACO-HR y la secuencia SQ-10 del modelo HP cuadrado y b) el algoritmo TH-ACO y la secuencia ST-26 del modelo HP triangular 2D.

6.6 Efecto de la variación de parámetros en ACO

Dos componentes importantes en los algoritmos ACO son los valores de la feromona y la función heurística. La primera representa qué tanto ha sido elegido un determinado movimiento durante las iteraciones del algoritmo, y la segunda representa el beneficio que otorga utilizar un cierto componente de la solución en la fase de construcción. Los parámetros que controlan la influencia de los valores de la feromona y de la función heurística son α, el peso de la información de la feromona; β, el peso de la información de la función heurística; y ρ, la persistencia de la feromona.

Para mostrar el efecto que tienen estos parámetros en el desempeño de los algoritmos presentados se realizaron 20 ejecuciones independientes sobre las

secuencias SQ-11 del modelo HP cuadrado y ST-26 del modelo triangular 2D. Se probaron los algoritmos ACO-HN del modelo cuadrado y TH-ACO del modelo triangular 2D. En la Fig. 6.10 se muestran los efectos de la variación en los parámetros α y β. Como podemos observar en estas figuras tanto el valor de la feromona como de la heurística son importantes para obtener resultados favorables en la búsqueda de soluciones. Cuando uno de estos valores se ignora ($\alpha=0$ o $\beta=0$) y se toma en cuenta sólo la feromona o sólo la heurística, no se producen los mejores resultados. Los valores con los que los algoritmos se desempeñan mejor, tanto en el modelo cuadrado como en el triangular son $\alpha=1$ y $\beta=2$, dando una mayor importancia a los valores de la información heurística.

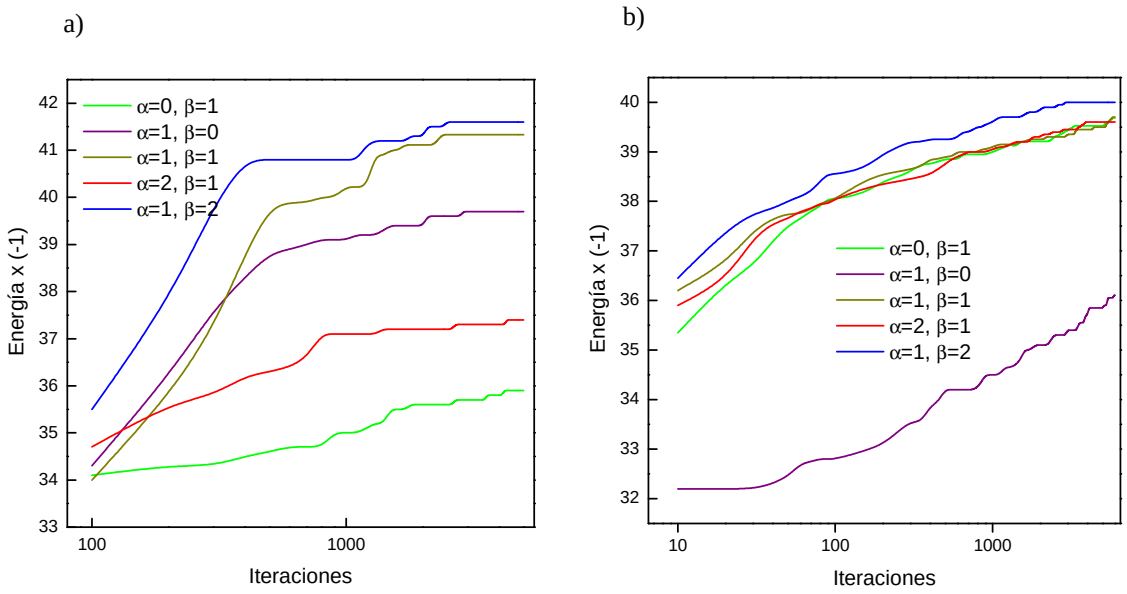


Fig. 6.10. Efectos de la variación de parámetros α y β para: a) la secuencia SQ-11 del modelo HP cuadrado con el algoritmo ACO-HR y b) ST-26 del modelo HP triangular 2D con el algoritmo TH-ACO.

También se realizaron pruebas variando el valor de la persistencia de la feromona (ρ) manteniendo otros parámetros estáticos. Los resultados de estos experimentos se muestran en la Fig. 6.11. En estas figuras podemos observar que tanto en el modelo cuadrado como en el modelo triangular es importante utilizar la información de iteraciones pasadas ya que si el valor de ρ es muy bajo (ρ=0.25), los algoritmos tienen un bajo desempeño. Sin embargo, si no hay ninguna pérdida de información de iteraciones pasadas (ρ=1) tampoco se tiene un buen desempeño. De lo anterior podemos observar que los valores de ρ en los que se tiene un mejor desempeño es en el intervalo de $0.5<\rho<0.85$, teniendo un mejor desempeño para ρ=0.85.

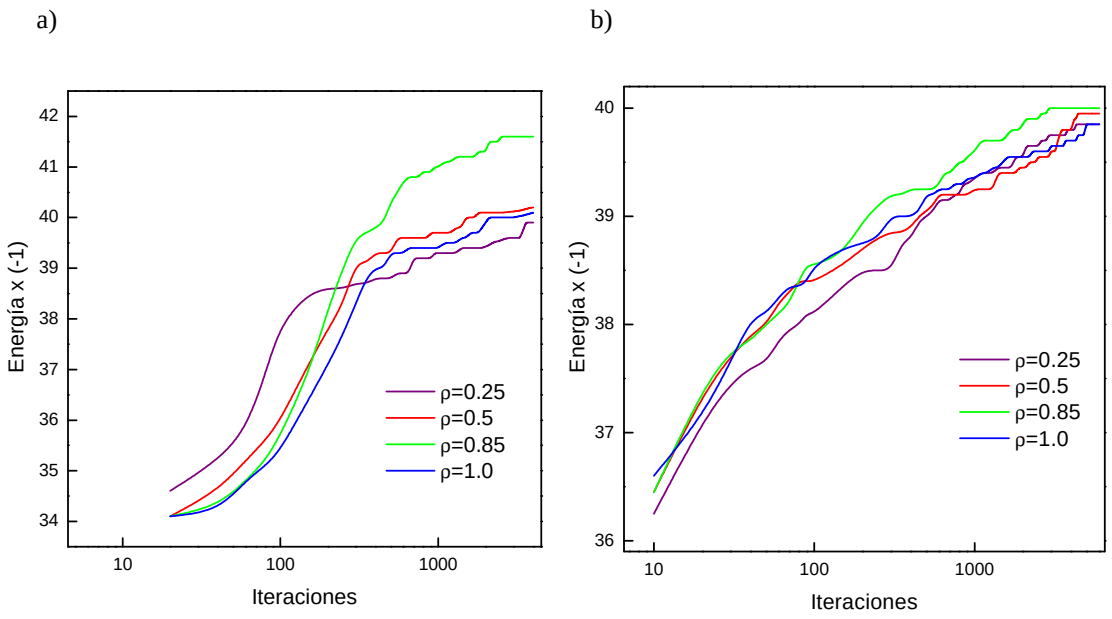


Fig. 6.11. Efectos de la variación de la persistencia de la feromona ρ para: a) la secuencia SQ-11 del modelo HP cuadrado con el algoritmo ACO-HR y b) para la secuencia ST-26 del modelo HP triangular 2D con el algoritmo TH-ACO.

Para evaluar qué tan sensibles son nuestros algoritmos cuando se varían los valores de los parámetros α y β se llevaron a cabo las siguientes pruebas. En la primera se mantuvo fijo el valor de β en 2 y se varió el valor de α desde 0.8 hasta 1.2, con incrementos de 0.1. En la segunda se mantuvo fijo el valor de α en 1 y se varió el valor de β desde 1.8 hasta 2.2 con incrementos de 0.1.

Los resultados obtenidos en estas pruebas se muestran en la Fig. 6.12, para el modelo cuadrado, y en la Fig. 6.13, para el modelo triangular. En estas figuras no se aprecia un cambio significativo en los resultados obtenidos con los diferentes valores en α y β descritos. En las últimas iteraciones los algoritmos consiguen obtener conformaciones con niveles de energía similares, como se observa en las gráficas cuando las líneas tienen a tocarse arriba de las 1000 iteraciones.

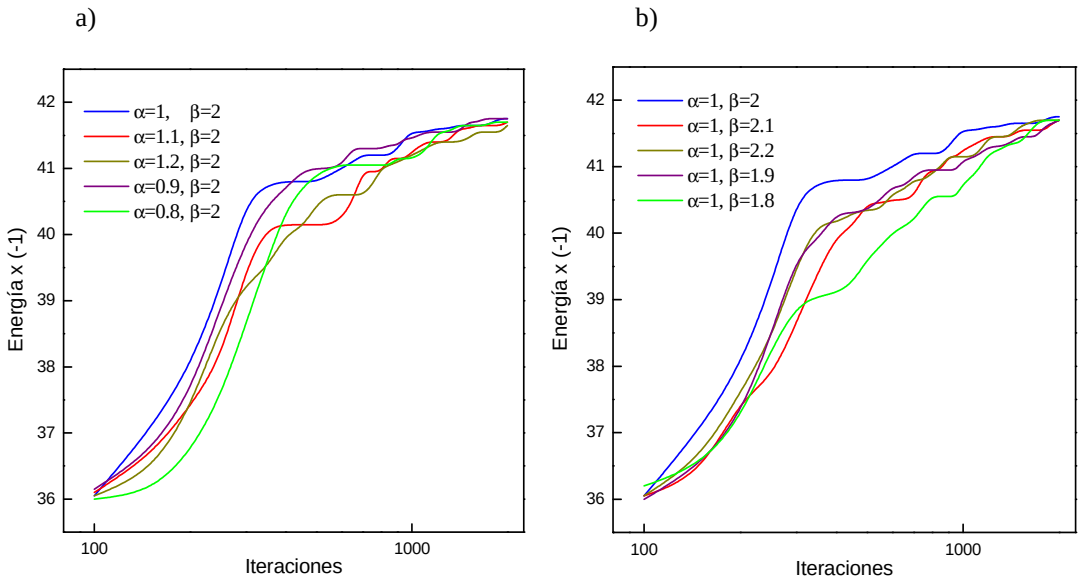


Fig. 6.12. Comportamiento del algoritmo ACO-HR para la secuencia SQ-11 del modelo HP cuadrado al variar levemente los valores de: a) α y b) β.

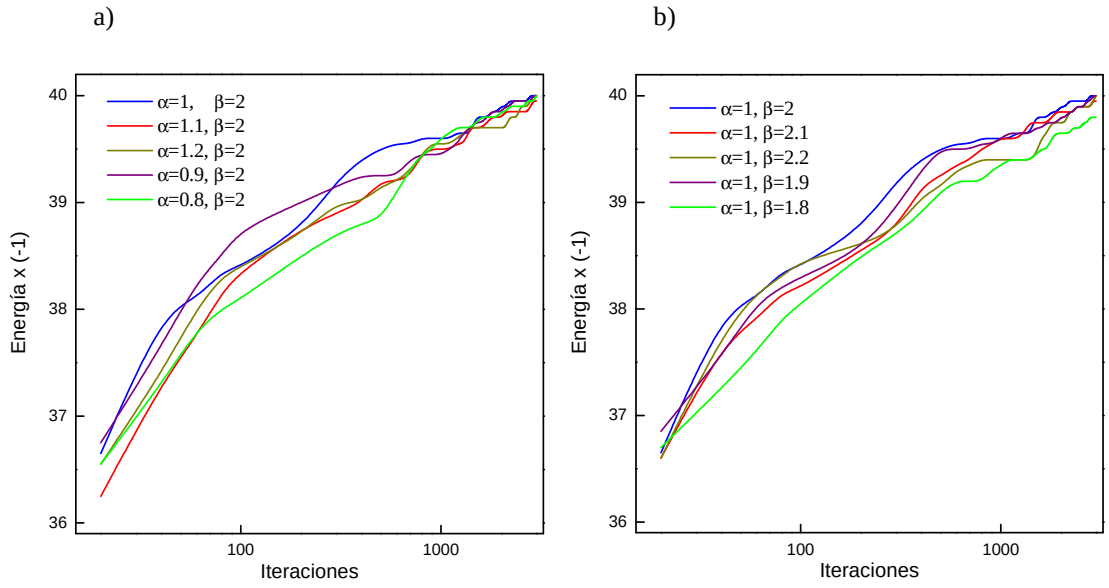


Fig. 6.12. Comportamiento del algoritmo TH-ACO para la secuencia ST-26 del modelo HP triangular 2D al variar levemente los valores de: a) α y b) β.

6.7 Análisis y discusión de resultados

A partir de los resultados presentados en las secciones anteriores, tenemos varios puntos importantes que hacer notar en los algoritmos ACO desarrollados:

- Cuando se utiliza el método de suavizado de feromona se consigue mejorar el desempeño de los algoritmos. Después de que los algoritmos ACO han realizado varias iteraciones, es común que determinados valores en la matriz de feromonas sean muy elevados mientras que otros sean muy bajos. Al utilizar el método de suavizado de feromona se aumenta la probabilidad de que las hormigas elijan los movimientos cuyo valor de feromona casi había desaparecido (valores que tienden a cero). Sin embargo, no se pierde la información que había sido almacenada con el avance de las iteraciones (los caminos que tenían valores de feromona mayores continúan teniendo los valores mayores aún después de realizar el suavizado). Esto permite que las hormigas puedan elegir caminos diferentes que las lleven a

encontrar mejores soluciones. Además, la ventaja del método de suavizado de feromona utilizado es que es su tiempo de ejecución es menor que el tiempo que consumen las búsquedas locales.

- La estrategia propuesta en esta tesis para resolver el traslape mejora el desempeño de los algoritmos en dos sentidos. El primero es en el tiempo de ejecución. El segundo es en la disminución de la energía encontrada en un número menor de iteraciones. El método de traslape propuesto no pierde tiempo desplegando la secuencia a e intentando volver a plegarla. Otros métodos propuestos en la literatura despliegan las conformaciones que presentan traslape e intentan volver a plegarlas. Sin embargo, cuando se intenta plegar nuevamente la secuencia se puede volver a caer en traslape varias veces, lo cual puede consumir un elevado tiempo de ejecución. Además de que durante la construcción de conformaciones se puede tener traslape en varias conformaciones y a lo largo de varias iteraciones, lo cual también aumenta el tiempo de ejecución. El método propuesto sólo elimina las conformaciones que presentan traslape (al considerarlas como defectuosas) y las reemplaza por otras, por lo que no presenta un elevado tiempo de ejecución. Por otro lado, debido a que se realizan copias de las conformaciones parcialmente plegadas que presentan un bajo nivel de energía, se aumenta la probabilidad de encontrar otras conformaciones con un bajo nivel de energía en un número menor de iteraciones.

- Los algoritmos ACO híbridos (H-ACO, SH-ACO, TH-ACO) consiguen un mayor porcentaje de éxito que los algoritmos ACO individuales (ACO-HR y ACO-HN). Los algoritmos ACO híbridos tienen hormigas de especie ACO-HR y de especie ACO-HN buscando soluciones de forma paralela. De esta manera, si una especie no es capaz de encontrar una conformación de baja energía de una secuencia, la otra especie se encarga de encontrarla. Además, conforme aumenta el número de iteraciones, la cantidad de hormigas de la especie que muestre el mejor desempeño va aumentando, mientras que la cantidad de hormigas de la especie

que no muestre buen desempeño va disminuyendo. De esta manera, los algoritmos tienden a favorecer a la especie de hormigas que tenga el mejor desempeño.

- En los algoritmos híbridos, cuando se combina la información de feromona para que las hormigas construyan soluciones (SH-ACO y TH-ACO) se tienen mejores resultados que cuando sólo se ejecutan las dos especies de hormigas en paralelo (H-ACO). En los algoritmos SH-ACO y TH-ACO, si las hormigas de una especie encuentran una conformación de mínima energía entonces la información de los valores de feromona de esta especie le proporciona información útil a la otra especie para guiar su camino y aumentar su posibilidad de encontrar conformaciones también con mínima energía. Lo anterior no sucede cuando sólo se ejecutan en paralelo.

- Un punto importante en el desempeño de los algoritmos en general, es que cuando la secuencia a plegar es de mayor longitud el tiempo requerido para encontrar conformaciones de mínima energía es mayor. En algunos casos, los algoritmos desarrollados no encontraron conformaciones con el mínimo nivel de energía reportado en la literatura, sobre todo en el modelo HP cuadrado. Lo anterior muestra una limitante en los algoritmos desarrollados. Para solucionar este problema y conseguir mejorar el desempeño de los algoritmos se requiere buscar formas alternativas de mejorar las soluciones encontradas por los mismos. El uso de una búsqueda local, el desarrollo de nuevas heurísticas o la creación de nuevos esquemas de combinación para los algoritmos híbridos podrían ser alternativas posibles para mejorar el desempeño de los algoritmos.

CAPÍTULO 7

Conclusiones y trabajo futuro

El estudio del plegamiento de las proteínas en el campo de la bioquímica tiene una gran importancia debido a que al comprender este complicado proceso se pueden llegar a comprender también diferentes funciones de las proteínas en los seres vivos. Como consecuencia, se podrían diseñar proteínas con una función específica, estudiar diversas enfermedades que tienen que ver con un mal plegado de las proteínas y definir la función de las proteínas a partir sólo del conocimiento de su estructura lineal de aminoácidos.

El estudio del problema del plegamiento de proteínas en las ciencias computacionales se puede ver como el resolver un problema de optimización complejo en donde se requiere buscar una solución óptima dentro de un conjunto enorme de posibles soluciones en un tiempo razonable. Por otra parte, en este problema se puede caer en mínimos locales muy fácilmente. Por este motivo, proponer diferentes técnicas que consigan mejorar el desempeño de los trabajos realizados hasta el momento es de gran importancia.

Varios métodos han sido propuestos a través de los años para intentar resolver el problema del plegamiento de proteínas. Uno de estos métodos es ACO.

En esta tesis se probaron dos algoritmos ACO cuyas principales diferencias consistieron en la función heurística utilizada, en el esquema de actualización de feromona y en la fórmula de evaluación que utilizan las hormigas para elegir sus movimientos en la construcción de soluciones. Además, se planteó la posibilidad de combinar diferentes funciones heurísticas y esquemas de actualización de feromona. Para esto se propusieron tres formas diferentes de combinar dos algoritmos ACO individuales (llamadas especies de hormigas). Estas combinaciones, llamadas algoritmos ACO híbridos utilizan diferentes estrategias para combinar información. Dos de ellas se centran principalmente en la fórmula de evaluación que utilizan las hormigas para elegir sus movimientos en la construcción de soluciones.

En este capítulo se presentan las conclusiones finales de esta tesis y el trabajo futuro que puede derivarse a partir de lo que se ha presentado en la misma.

7.1 Conclusiones

Las contribuciones realizadas en esta tesis se resumen en los siguientes puntos:

- Se mostró que al variar la función heurística y el esquema de actualización de feromona (ACO-HR y ACO-HN) en los algoritmos ACO se pueden obtener diferentes resultados en secuencias de un mismo problema.

- Los algoritmos ACO híbridos mostraron un mejor desempeño global que los algoritmos ACO individuales, siendo capaces de encontrar la energía más baja

encontrada por los algoritmos individuales en un tiempo medio o menor al de los mismos.

- Los algoritmos ACO híbridos lograron incrementar el porcentaje de éxito notablemente, obteniendo un porcentaje mayor o igual al porcentaje mayor presentado por los algoritmos individuales. Arriba del 75% de los casos, los algoritmos híbridos consiguieron obtener porcentajes de éxito del 100% o valores cercanos al mismo.

- La estrategia incluida en los algoritmos ACO de suavizado de feromona evitó el estancamiento de los mismos y permitió salir en muchos casos de mínimos locales, sin incrementar el tiempo de ejecución de los algoritmos como sucede cuando se utiliza una búsqueda local robusta.

- En esta tesis se mostró que los algoritmos ACO pueden ser aplicados de forma eficiente al problema del plegamiento de proteínas. A pesar de que ACO está basado en operaciones simples se consiguieron obtener buenos resultados en comparación a otros algoritmos presentados en la literatura.

- Además, la nueva estrategia para resolver el problema de traslape consiguió disminuir el tiempo de ejecución de los algoritmos, mostrándose como una estrategia para evitar estados no permitidos en las hormigas y fortalecer la búsqueda de soluciones óptimas.

- Se propuso la primera aplicación de un algoritmo ACO para el modelo HP triangular y se mostró la efectividad de este algoritmo en secuencias del modelo HP triangular, encontrando mejores soluciones a las reportadas en la literatura. De esta manera, ACO se presenta como una técnica efectiva para resolver este problema.

7.2 Trabajo futuro

El trabajo futuro que puede surgir a partir de esta tesis se resume en los siguientes puntos:

- Explorar otros modelos del plegamiento de proteínas más complejos como el modelo HP triangular 3D, hexagonal, diamante, el modelo HZ (*hydrophobic zipper*), etc.

- Explorar alternativas de búsqueda local que mejore las conformaciones que presenten los más bajos niveles de energía. Esta búsqueda local podría trabajar de forma conjunta con el método de suavizado de feromona, ejecutándose sólo después de que las hormigas no consigan mejorar las soluciones después de un determinado número de iteraciones. De esta manera se buscaría no incrementar el tiempo de ejecución de los algoritmos considerablemente y mejorar las soluciones.

- Realizar implementaciones de un algoritmo ACO híbrido con más de dos especies de hormigas y probar con otros esquemas de combinación a los explorados en esta tesis.

- Otra extensión atractiva del presente trabajo sería probar los algoritmos híbridos propuestos en un problema de optimización combinatoria diferente al problema del plegamiento de proteínas.

Referencias

1. Aarts E. H., Korst J. H., Laarhoven P. J. (1997). “Simulated Annealing”. En E. Aarts y J. Lenstra, (Eds.), *Local Search in Combinatorial Optimization* (pp. 91-120), Chichester: John Wiley and Sons.

2. Agarwala R., Batzoglou S., Dančík V., Decatur S.E., Hannenhalli S., Farach M., Muthukrishnan S., Skiena S. (1997). “Local Rules for Protein Folding on a Triangular Lattice and Generalized Hydrophobicity in the HP Model”. *Journal of Computational Biology*, 4(3), 275-296.

3. Anfinsen C. (1973). “Principles that Govern the Folding of Protein Chains”. *Science*, 181(96), 223-230.

4. Bäck T. (1996). *Evolutionary Algorithms in Theory and Practice*. New York, NY: Oxford University Press.

5. Bastolla U., Frauenkron H., Gerstner E., Grassberger P., Nadler W. (1998). “Testing a New Monte Carlo Algorithm for the Protein Folding Problem”. *Proteins: Structure, Function, and Genetics*, 32(1), 52-66.

6. Bianchi L., Birattari M., Chiarandini M., Manfrin M., Mastrolilli M., Paquete L., Rossi-Doria O., Schiavinotto T. (2004). “Metaheuristics for the Vehicle Routing Problem with Stochastic Demands”. En X. Yao *et al.*, (Eds.), *Proceedings of Parallel Problem Solving from Nature–PPSN VII, 8th International Conference*, Vol. 3242 de LNCS (pp. 450-460). Berlin, Germany: Springer-Verlag.

7. Blum C., Dorigo M. (2004). “The Hyper-Cube Framework for Ant Colony Optimization”. *IEEE Transactions on Systems, Man, and Cybernetics – Part B*, 34(2), 1161-1172.

8. Bodenhofer U. (2003). *Genetic Algorithms: Theory and Applications* (3a. ed.). Lecture Notes, Fuzzy Logic Laboratorium Linz-Hagenberg, Winter 2003/2004.

9. Bonabeau E., Dorigo M., Théraulaz G. (1999). *Swarm Intelligence: From Natural to Artificial Systems*. New York, NY: Oxford University Press.

10. Branden C., Tooze J. (1999). *Introduction to Protein Structure* (2ª ed.). New York, NY: Garland Publishing Incorporated.

11. Bullnheimer B., Hartl R. F., Strauss C. (1998). “Applying the Ant System to the Vehicle Routing Problem”. In I. H. Osman, S. Voβ, S. Martello y C. Roucairol, (Eds.), *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization* (pp. 109-120). Boston, MA: Kluwer Academic Publishers.

12. Bullnheimer B., Hartl R. F., Strauss C. (1999). “A New Rank-Based Version of the Ant System: a Computational Study”. *Central European Journal for Operations Research and Economics*, 7(1), 25-38.

13. Cantarow A., Schepartz B. (1965). *Bioquímica* (3ª. ed.). México: Interamericana.

14. Chen M., Huang W. (2005). “A Branch and Bound Algorithm for the Protein Folding Problem in the HP Lattice Model”. *Genomics, Proteomics and Bioinformatics*, 3(4), 225-230.

15. Chikenji G., Kikuchi M., Iba Y. (1999). "Multi-Self-Overlap Ensemble for Protein Folding: Ground State Search and Thermodynamics". *Physical Review Letters*, 83(9), 886-1889.

16. Chu D., Till M., Zomaya A. (2005). "Parallel Ant Colony Optimization for 3D Protein Structure Prediction Using the HP Lattice Model". En Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05) Workshop 6, Vol. 7, (193b).

17. Colorni A., Dorigo M., Maniezzo V., Trubian M. (1994). "Ant System for Job-Shop Scheduling". *Belgian Journal of Operations Research, Statistics and Computer Science*, 34(1), 39-53.

18. Cordón O., Fernández de Viana I., Herrera F., Moreno L. (2000). "A New ACO Model Integrating Evolutionary Computation Concepts: The Best-Worst Ant System". En M. Dorigo, M. Middendorf y T. Stützle, (Eds.), *Abstracts Proceedings of ANTS2000 – From Ant Colonies to Artificial Ants: A Series of International Workshop on Ant Algorithms* (pp. 22-29). Université Libre de Bruxelles, Belgium: IRIDIA.

19. Cordón O., Herrera F., Stützle T. (2002). "A Review on the Ant Colony Optimization Metaheuristic: Basis, Models and New Trends". *Mathware and Soft Computing*, 9(3), 141-175.

20. Creighton T.E. (1996). *Proteins, Structures and Molecular Properties* (2ª. ed.). New York, NY: W.H. Freeman and Company.

21. Darby N.J., Creighton T.E. (1993). *Protein Structure*. Oxford, UK: Oxford University Press.

22. Dill K.A., Fiebig K.M., Chan H.S. (1993). “Cooperativity in Protein Folding Kinetics”. *Proceedings of National Academy of Sciences of the USA*, 90(5) 1942-1946.

23. Dill K. A. (1999). “Polymer Principles and Protein Folding”. *Protein Science*, 8(6), 1166-1180.

24. Dorigo M. (1992). “Optimization, Learning and Natural Algorithms”. PhD Thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy.

25. Dorigo M., Socha K. (2007). “An Introduction to Ant Colony Optimization”. Por aparecer en T. F. Gonzalez, (Ed.), *Approximation Algorithms and Metaheuristics*, CRC Press.

26. Dorigo M., Birattari M., Stützle T. (2006). “Ant Colony Optimization - Artificial Ants as a Computational Intelligence Technique”. *IEEE Computational Intelligence Magazine*, 1(4), 28-39.

27. Dorigo M., Maniezzo V., Colorni A. (1996). “Ant System: Optimization by a Colony of Cooperating Agents”. *IEEE Transactions on Systems, Man, and Cybernetics – Part B*, 26(1), 29-41.

28. Dorigo M., Gambardella L.M. (1997). “Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem”. *IEEE Transactions on Evolutionary Computation*, 1(1), 53-66.

29. Fidanova S. (2006). “3D HP Protein Folding Problem Using Ant Algorithm”, En *Proceedings of Bioprocess of Systems 2006*, Vol. 3 (pp. 19-26). Sofía, Bulgaria: III.

30. Garduño R., Morales L.B., Pérez-Neri F. (1990). "Global Minimum Energy Conformations of the Thyrotropin Releasing Hormone and its Analogs". *Journal of Molecular Structures (THEOCHEM)*, 208(3-4), 279-300.

31. Garduño R., Morales L.B. (2003). "A Genetic Algorithm with Conformational Memories for Structure Prediction of Polypeptides". *Journal of Biomolecular Structures and Dynamics*, 21(1), 41-63.

32. Glover F., Laguna M. (1997). *Tabu Search*. Boston, MA: Kluwer Academic Publishers.

33. Goldberg D.E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Boston, MA: Addison-Wesley Longman Publishing Company, Inc.

34. Guntsch M., Middendorf M. (2002). "A Population Based Approach for ACO". In S. Cagnoni, J. Gottlieb, E. Hart, M. Middendorf y G. Raidl, (Eds.), *Applications of Evolutionary Computing, Proceedings of Evo Workshops 2002: EvoCOP, EvoIASP, EvoSTim*, Vol. 2279 de LNCS (pp. 71-80). Berlin, Germany: Springer-Verlag.

35. Hansen P., Mladenovic N. (1999). "An Introduction to Variable Neighborhood Search". In S. Voss, S. Martello, I. H. Osman y C. Roucairol, (Eds.), *Meta-Heuristics – Advances and Trends in Local Search Paradigms for Optimization* (pp. 433-458). Dordrecht, The Netherlands: Kluwer Academic Publishers.

36. Hart W.E., Newman A. (2006). "Protein Structure Prediction with Lattice Models". En Srinivas Aluru, (Ed.), *Handbook of Computational Molecular Biology*. Chapman & Hall CRC Computer and Information Science Series.

37. Holland J. (1975). *Adaptation in Natural and Artificial Systems*, Ann Arbor, MI:University of Michigan Press.

38. Hsu H.P., Mehra V., Nadler W., Grassberger P. (2003). "Growth Algorithm for Lattice Heteropolymers at Low Temperatures". *Journal of Chemical Physics*, 118(1), 444-451.

39. Kirkpatrick S., Gelatt C., Vecchi M. (1983). "Optimization by Simulated Annealing". *Science*, 220(4598), 671-680.

40. Krasnogor N., Pelta D., López P.M., de la Canal E. (1998). "Genetic Algorithms for the Protein Folding Problem, a Critical View". En C. Alpaydin, (Ed.), *Proceedings of Engineering of Intelligent Systems*, (pp. 353-360). La Laguna, Tenerife: ICSC Academic Press.

41. Krasnogor N., Blackburne B.P., Burke E.K., Hirst J.D. (2002). "Multimeme Algorithms for Protein Structure Prediction". *Lecture Notes in Computer Science*, 2439, 769-778.

42. Lesh N., Mitzenmacher M., Whitesides S. (2003). "A Complete and Effective Move Set for Simplified Protein Folding". En *Proceedings of Research in Computational Molecular Biology (RECOMB'03), 7th Annual International Conference on Computational Biology* (pp. 188-195). New York, NY: ACM Press.

43. Lessing L., Dumitrescu I., Stützle T. (2004). "A Comparison Between ACO Algorithms for the Set Covering Problem". En M. Dorigo, L. Gambardella, F. Mondada, T. Stützle y C. Blum, (Eds.), *ANTS'2004, 4th International Workshop on Ant Algorithms and Swarm Intelligence*, Vol. 3172 de LNCS (pp. 1-12). Berlin, Germany: Springer-Verlag.

44. Levinthal, C. (1968). “Are there Pathways for Protein Folding?”. *Journal of Chemical Physics*, 65, 44-45.

45. Liang F., Wong W.H. (2001). “Evolutionary Monte Carlo for Protein Folding Simulations”, *Journal of Chemical Physics*, 115(7), 3374-3380.

46. Lourenčo H. R., Martin O., Stützle T. (2002). “Iterated Local Search”. Technical Report AIDA-00-06, FG Intellektik, FB Informatik, TU Darmstadt, Germany, November 2000. Por aparecer en F. Glover y G. Kochenberger, (Eds.), *Handbook of Metaheuristics*. Kluwer Academic Publishers.

47. Lourenčo H. R., Serra D. (2002). “Adaptive Approach Heuristics for the Generalized Assignment Problem”. *Mathware and Soft Computing*, 9(3), 209-234.

48. Macarulla J.M., Goñi F.M. (1993). *Biomoléculas, Lecciones de Química Estructural* (3ª ed.). Barcelona: Reverté.

49. Maniezzo V., Colorni A. (1999). “The Ant System Applied to the Quadratic Assignment Problem”. *IEEE Transactions on Knowledge and Data Engineering*, 11(5), 769-778.

50. Michel R., Middendorf M. (1999). “An ACO Algorithm for the Shortest Common Supersequence Problem”. In D. Corne, M. Dorigo y F. Glover, (Eds.), *New Methods in Optimization*. Boston, MA: McGraw Hill.

51. Morales L.B., Garduño R., Aguilar J.M., Riveros F.J. (2000). “A Parallel Tabu Search for Conformational Energy Optimization of Oligopeptides”. *Journal of Chemical Physics*, 21(2), 147-156.

52. Morales L.B., Garduño R., Romero D. (1991). "Applications of Simulated Annealing to the Multiple-Minima Problem in Small Peptides". *Journal of Biomolecular Structure and Dynamics*, 8(4), 721-735.

53. Patton A., Goldman E. (1995). "A Standard GA Approach to Native Protein Conformation Prediction". En L. Eshelman, (Ed.), *Proceedings of the 6th International Conference on Genetic Algorithms* (pp. 574-581). San Francisco, CA: Morgan Kauffman.

54. Ramakrishnan R., Ramachandran B., Pekny J.F. (1997). "A Dynamic Monte Carlo Algorithm for Exploration of Dense Conformational Spaces in Heteropolymers". *Journal of Chemical Physics*, 106(6), 2418-2424.

55. Rego C., Li H., Glover F. (2006). "A Filter-and-Fan Approach to the 2D Lattice Model of the Protein Folding Problem". Manuscrito bajo revision.

56. Sanger, F. (1952). "The Arrangement of Amino Acids in Proteins". *Advances in Protein Chemistry*, 7, 1-67.

57. Santos E. E., Santos E. (2004). "Reducing the Computational Load of Energy Evaluations for Protein Folding". Fourth IEEE Symposium of Bioinformatics and Bioengineering 2004, pp. 79-86.

58. Shmygelska A., Hernández R., Hoos H.H. (2002). "An Ant Colony Optimization Algorithm for the 2D HP Protein Folding Problem". En *Proceedings of the 3rd International Workshop on Ant Algorithms*, Vol. 2463 de LNCS (pp. 40-52). Springer Verlag.

59. Shmygelska A., Hoos H.H. (2003). "An Improved Ant Colony Optimization Algorithm for the 2D HP Protein Folding Problem". En *Proceedings of the 16th Canadian Conference on Artificial Intelligence (AI'2003)* Vol. 2671 (pp. 400-417). Springer Verlag.

60. Shmygelska A., Hoos H.H. (2005). "An Ant Colony Optimization Algorithm for the 2D and 3D Hydrophopic Polar Protein Folding Problem". *BMC Bioinformatics*, 6(30).

61. Shmygelska A. (2006). "Novel Heuristic Search Methods for Protein Folding and Identification of Folding Pathways". PhD. Thesis, Department of Computer Science, The University of British Columbia, Vancouver, Canada.

62. Sikder A.R., Zomaya A.Y. (2005). "An Overview of Protein-Folding Techniques: Issues and Perspectives". *International Journal of Bioinformatics Research and Applications*, 1(1), 121-143.

63. Socha K., Sampels M., Manfrin M. (2003). "Ant Algorithms for the University Course Timetabling Problem with Regard to the State-of-the-Art". En G. Raidl *et al.*, (Eds.), *Proceedings of EvoCOP 2003 – 3rd European Workshop on Evolutionary Computation in Combinatorial Optimization*, Vol. 2611 de LNCS (pp. 334-345). Berlin, Germany: Springer-Verlag.

64. Socha K., Knowles J., Sampels M. (2004). "A MAX-MIN Ant System for the University Timetabling Problem". En M. Dorigo, G. Di Caro y M Sampels, (Eds.), *Proceedings of ANTS 2002 – 3rd International Workshop on Ant Algorithms*, Vol. 2463 de LNCS (pp. 1-13). Berlin, Germany: Springer-Verlag.

65. Song J., Cheng J., Zheng T. (2006). "Protein 3D HP Model Folding Simulation Based on ACO". En *Proceedings of the 6th International Conference on Intelligent Systems Design and Applications* (pp. 410-415). Jinan, China.

66. Stützle T., Hoos H.H. (2000). "MAX-MIN Ant System". *Future Generation Computer System*, 16(8), 889-914.

67. Unger R., Moult J. (1993). "Genetic Algorithms for Protein Folding Simulations". *Journal of Molecular Biology*, 231(1), 75-81.

68. Unger R., Moult J. (1993). "A Genetic Algorithm for Three Dimensional Protein Folding Simulations". En *Proceedings of the 5th International Conference on Genetic Algorithms* (pp. 581-588). San Francisco, CA: Morgan Kauffman Publishers Inc.

69. Unger R., Moult J. (1993). "Finding the Lowest Free-Energy Conformation of a Protein is an NP-Hard Problem – Proof and Implications". *Bulletin of Mathematical Biology*, 55(6), 1183-1198.

70. Veale T. W., Bray H. G., James S.P. (1970). *Biochemistry for Medical Students* (9ª ed.). Cía. Editorial Continental.

71. Zagrovic B., Christopher D.S., Khaliq S., Michael R.S.,Vijay S.P. (2002). "Native-Like Mean Structure in the Unfolded Ensemble of Small Proteins". *Journal of Molecular Biology*, 323(1), 153–164.

72. Zhang J.L., Liu J.S. (2002). "A New Sequential Importance Sampling Method and its Application to the Two-Dimensional Hydrophobic-Hydrophilic Model". *Journal of Chemical Physics*, 117(7), 3492-3498.