METAHEURÍSTICA HÍBRIDA PARA LA IDENTIFICACIÓN DE MODELOS DE AUTÓMATA CELULAR EN TRAYECTORIAS DE SIMULACIÓN DE PLEGAMIENTO DE PROTEÍNA

Trabajo de Grado

Francisco Javier Obando Vidal

José Ricardo Gallego Garcés

Director: Mag. Néstor Milciades Díaz Mariño

Universidad del Cauca

Facultad de Ingeniería Electrónica y Telecomunicaciones

Departamento de Sistemas

Grupo de I+D en Tecnologías de la Información

Modelado y Simulación de Sistemas Complejos

Popayán, Agosto de 2012

TABLA DE CONTENIDO

INDICE DE FIGURAS

INDICE DE TABLAS

ANEXOS

LISTA DE ACRONIMOS

AA: Aminoácido.

SAA: Secuencia de aminoácidos.

AC: Autómata Celular.

EA: Algoritmos Evolutivos

AG: Algoritmo Genético

TS: Búsqueda Tabú

MA: Algoritmos Memèticos

3D: Tridimensional

C-α: Carbono alfa.

MD: Dinámica Molecular

MC: Monte Carlo

IA: Inteligencia Artificial

DM: Minería de Datos.

ILAS: Aprendizaje Iterativo con Alternación de Estrato

MCC: Coeficiente de Correlación de Matthews.

ROC: Receptor de Característica Operativa (Receiver Operating Characteristic)

FP: Falsos Positivos.

FN: Falsos Negativos.

VP: Verdaderos Positivos.

VN: Verdaderos Negativos

TPR: Tasa de Verdaderos Positivos

FPR: Tasa de Falsos Positivos.

1 INTRODUCCIÓN

El plegamiento de proteínas es el proceso por el cual una proteína alcanza su estructura terciaria, si éste es exitoso, la proteína realiza correctamente su función, de lo contrario se podrían generar diferentes patologías (Alzheimer, Parkinson, entre otras) [1] . El estudio de este fenómeno es importante, dado que un mejor entendimiento daría paso a mejores soluciones de problemas asociados al plegamiento de proteínas, tales como diseño de nuevos medicamentos, mejora de alimentos y tratamiento de enfermedades. Sin embargo, este proceso es muy complejo [2], razón por la cual uno de los enfoques que se utiliza para su estudio se basa en métodos computacionales [1-3].

En el enfoque computacional para simular el plegamiento de proteínas se aplican técnicas de simulación entre las que se encuentran la dinámica molecular [4-5] y el método Montecarlo [6-7], que son computacionalmente costosos debido al tamaño del problema del plegamiento de proteínas donde hay una gran cantidad de partículas que interactúan entre sí a nivel atómico. Los resultados obtenidos son difíciles de analizar dado que corresponden a una gran cantidad de datos que indican las configuraciones 3D de los átomos que conforman la proteína durante diversos momentos del plegamiento, por lo cual se buscan modelos que faciliten el entendimiento de este fenómeno.

Los Autómatas Celulares (AC) como técnica para la creación de modelos de simulación, son muy usados debido a la simplicidad de su arquitectura, que se caracteriza por ser discreta en tiempo y espacio [8-9]. Otra razón para su amplio uso es que pueden representar diversos fenómenos de comportamiento complejo y dinámico, como el casi del plegamiento de proteínas [1, 3]. Adicionalmente, los AC presentan la ventaja de tener un desarrollo significativo en cuanto a técnicas de aprendizaje de máquina para la identificación de los parámetros que definen su arquitectura, lo que resulta útil cuando no se conoce el detalle de la dinámica de un fenómeno complejo, pero sí existen datos que proporcionan evidencia del comportamiento global [10].

La propuesta aquí presentada busca explorar mediante el desarrollo de una nueva técnica, un enfoque computacional que permita avanzar en el estudio del fenómeno del plegamiento de proteínas, brindando una representación más simple y entendible de éste fenómeno. Surge como alternativa de solución el uso de metaheurísticas, que son capaces de realizar búsquedas concretas en un gran espacio de soluciones y lograr encontrar una solución cercana al óptimo, tal cual es el caso de la identificación de los parámetros de un modelo de AC.

La metaheurística que se plantea es un hibrido entre Algoritmos Genéticos y Búsqueda Tabú, este tipo de algoritmos híbridos suelen obtener mejores resultados que una metaheurística poblacional o de trayectoria por separado [11], es por eso que se busca aprovechar las características de la búsqueda tabú para mejorar el comportamiento del algoritmo genético planteado en [12]. En estudios realizados en [13] se han obtenido mediante algoritmos genéticos modelos de AC que brindan un 90% de precisión para la

reproducción de una trayectoria de simulación de plegamiento de proteína. A pesar que la precisión obtenida es buena, se hace necesario encontrar técnicas que permitan obtener modelos de AC que repliquen trayectorias de plegamiento de proteínas con mayor precisión.

El contexto anterior dio paso a la siguiente pregunta de investigación: ¿Cómo realizar una metaheurística hibrida entre búsqueda tabú y algoritmos genéticos que mejore los resultados del trabajo descritos en [13] obteniendo modelos de AC que representen el fenómeno del plegamiento de proteínas?.

A continuación se presenta los objetivos generales y específicos del trabajo de investigación:

- Objetivo General: Proponer una metaheurística hibrida entre AG y TS, para identificar modelos de autómatas celulares a partir de la información proporcionada por una trayectoria de simulación de plegamiento de proteínas, mejorando los resultados obtenidos por el AG propuesto en [12].

- Objetivo Especifico 1: Especificar una metaheurística hibrida entre AG - TS que mejore el proceso de búsqueda del AG propuesto en [12], teniendo en cuenta el espacio ya explorado de soluciones prometedoras del AG para determinar el punto de partida de TS.

- Objetivo Especifico 2: Implementar un prototipo de la metaheurística híbrida entre AG y TS, que recibirá como entradas el conjunto de datos de una trayectoria de plegamiento de proteínas, y un modelo AC parcialmente definido (conjunto de estados, condiciones de frontera y lattice).

- Objetivo Especifico 3: Evaluar los modelos de AC obtenidos con la metaheurística AG - TS con respecto al conjunto de datos de trayectoria de plegamiento de la proteína HP35 – NleNle[14][1].

Como resultado del trabajo de investigación se espera obtener una metaheurística hibrida entre AG y TS, que permita identificar modelos de autómatas celulares replicando de manera aproximada una trayectoria de simulación de plegamiento de proteínas, obteniendo una mayor precisión frente a otros resultados obtenidos en estudios previos.

Este Documento de tesis se estructura de la siguiente manera: El Capitulo Dos presenta el marco teórico y estudios previos relacionados con el trabajo de investigación. En el Capitulo Tres se describe el proceso de minería de datos que se siguió para la

[1] Se usaran medidas para contrastar los resultados: coeficiente de correlación de matthews , matriz de confusión (VP, VN, FN, FP), sensibilidad, especificidad, Curva ROC, precisión.

construcción de la soluciona al problema de investigación. El Capitulo Cuatro describe la técnica propuesta e implementada en forma detallada. En el Capitulo 5 se muestran los resultados obtenidos por la técnica implementada y su respectivo análisis. El Capitulo Final contiene las conclusiones del trabajo de investigación, aportes de la investigación y la descripción de las principales líneas de trabajos futuros que surgen de la tesis.

2 MARCO CONCEPTUAL

2.1 CONTEXTO GENERAL

2.1.1 Proteínas

Las proteínas son macromoléculas[2], de compuestos químicos complejos, creados y usados por las células, para su existencia y correcto funcionamiento, formadas por aminoácidos (AA) los cuales se caracterizan por tener un grupo carboxílico (-C00), un grupo amino (-NH3), una cadena lateral variable y un carbono alfa [15-16] .

La estructura de la proteína es muy compleja, por lo cual se organiza en cuatro niveles: estructura primaria, estructura secundaria, estructura terciaria y estructura cuaternaria [15, 17].

Estructura Primaria: Es Determinada por los AA y la secuencia en la que aparecen en la cadena polipeptídica[3], la cual no es única y es similar a un hilo unidimensional de AA [15-16].

Estructura Secundaria: Es la relación que guarda cada AA con respecto al AA que le sigue y al AA que le antecede en la cadena polipeptídica haciendo referencia a la forma que adopta en el espacio. Dando lugar a combinaciones periódicas y repetitivas, que son la base principal de la estructura de la proteína. Existen dos tipos de estructuras secundarias principales conocidas como hélices alfa (forma helicoidal) y hojas beta (forma zigzag) [15, 17].

Estructura Terciaria: también llamado estado nativo es un tipo de estructura completa y termodinámicamente estable en la proteína, determinada por la SAA en 3D, lo que la hace distinta para cada proteína [15-16].

Estructura Cuaternaria: Es cuando una proteína consta de más de una cadena polipeptídica [15, 17].

2.1.2 Plegamiento de Proteínas

El plegamiento de proteínas se entiende como la habilidad de todos los seres vivos de ensamblar de forma correcta los aminoácidos (AA) y llevarlos a una forma tridimensional,

[2] Macromolécula: son formadas por un gran número de átomos por lo cual poseen un alto nivel de masa molecular.

[3] Cadena polipeptídica: es la unión de muchos aminoácidos mediante la perdida de una molécula de agua entre el grupo amino de un aminoácido y el grupo carboxilo.

generando proteínas con características y funciones biológicas especificas. Si una proteína no alcanza un correcto plegado, no podría cumplir con su función biológica, a éste tipo de proteína se le conoce como prión, la cual puede generar un gran número de patologías como el Alzheimer y el síndrome de Creutz-Jakob [1, 3, 17]. Las proteínas cumplen con una gran variedad de funciones en los procesos vitales de los seres vivos, las cuales están determinadas por su estructura tridimensional y ésta a su vez por su secuencia de aminoácidos[4] (SAA) [1, 17-18] [1, 3, 17].

En el estudio del plegamiento de proteínas se busca determinar la estructura tridimensional de la proteína [1], por lo cual se han utilizado diferentes técnicas experimentales y computacionales. En las técnicas in-vitro o experimentales se han usado principalmente dos métodos: la espectroscopia por resonancia magnética nuclear (RMN) [19] y la Cristalografía de rayos X [19]. Estos dos métodos obtienen la información para identificar coordenadas en 3D de una proteína, pero a pesar de lograr grandes avances son muy costosas debido al tiempo, al personal experto y equipos necesarios para llevar a cabo el proceso de cada técnica [12, 19].

Las técnicas computacionales se dividen en dos, por homologías y ab-initio. La primera es un método comparativo en donde se tiene una SAA conocida, pero de estructura desconocida, la cual será comparada con una o varias proteínas de SAA y de estructura conocida [20-21]. Se construye un modelo de estructura 3D para la secuencia desconocida, usando estructuras conocidas de otras proteínas, como paso final se optimiza la estructura con funciones de minimización de energía hasta obtener una conformación físicamente estable [20]. El principal inconveniente de esta técnica es que no se tiene una proteína de estructura adecuada para toda proteína desconocida [20-24].

El segundo enfoque computacional, parte desde la SAA, a la cual se le aplican técnicas de simulación en busca de una aproximación razonable de su estructura nativa, explorando en un espacio de posibles conformaciones [12, 24]. Entre las técnicas de simulación se destaca la simulación Monte Carlo (MC), que es un método estadístico usado para encontrar soluciones a diferentes problemas, los cuales se pueden formular como fenómenos estocásticos y de muestreo al azar a gran escala, en el plegamiento de proteínas, se hace un muestreo que genera un conjunto de conformaciones, verificando funciones de energía, para superar mínimos locales, en donde se obtiene la información de distancia y movimiento de las partículas presentes en la conformación de la proteína [7].

Otra técnica de simulación destacada es la Dinámica Molecular (MD) el cual es un método de mecánica estadística usado para estimar el equilibrio y las propiedades dinámicas de sistemas complejos que no pueden ser calculados analíticamente, en el plegamiento de proteínas se caracteriza por aplicar ecuaciones de movimiento de la física clásica a cada uno de los átomos que conforman la SAA, obteniendo información relativa de las

[4] Secuencia de aminoácidos: es el orden en que los aminoácidos se unen a cadenas de péptidos de formulario, o polipéptidos.

propiedades estructurales de la proteína [5]. Sin embargo éstas técnicas de simulación son demasiado costosas en tiempo de cómputo, debido a que tienen que interactuar a nivel atómico, es por eso que se hace necesario investigar técnicas que simulen de forma más sencilla este tipo de fenómenos [5, 7, 12].

2.1.3 Autómatas Celulares (AC)

Los AC son hoy en día una de las técnicas para creación de modelos de simulación más populares, ya que son capaces de representar comportamientos complejos de fenómenos diversos, que se originan a partir de sus componentes y relaciones relativamente simples entre ellos [8]. Estas relaciones son de tipo local exclusivamente. La representación que estos ofrecen es discreta en tiempo y espacio. Los sistemas que son susceptibles de ser representados mediante modelos de AC, son aquellos cuyo comportamiento es dinámico, bien sea determinístico o probabilístico [9, 25].

Los parámetros necesarios para el modelado de AC son:

Lattice o Malla: Se puede representar como una matriz de n-dimensiones, compuesta por un conjunto generalmente finito de celdas o células y de geometría regular. En la mayoría de trabajos en los que son empleados se utilizan matrices de 1, 2 ó 3 dimensiones, y debe tener en consideración que entre mayor sea el número de dimensiones que se utilizan, mayor será su complejidad computacional [8].

Condición de Frontera: Es la consideración que se debe tener con las *celdas* o *células* que se encuentran en los bordes del *lattice,* Existen diferentes tipo de fronteras, entre las que se debe escoger la que mejor se ajuste al problema a tratar [9].

1. Frontera Fija: Todas las células fuera del lattice tienen un valor predeterminado (Figura 1a).

2. Frontera Reflectora: Los valores de las *células* fuera del *lattice* son el reflejo de los valores de las *células* dentro (Figura 1b).

3. Frontera Periódica: Se considera al *lattice* como si sus extremos se tocaran (Figura 1c). Así en 1D las vecinas del borde izquierdo son las *células* del borde derecho y viceversa formando un aro, en 2D los vecinos del borde superior son las *células* del borde inferior y las vecinas del borde izquierdo son las *células* del borde derecho y viceversa formando un toroide y en 3D forman una esfera.

4. Sin Frontera: Las células fuera del lattice no tienen un comportamiento determinado su comportamiento es aleatorio (Figura 1d).

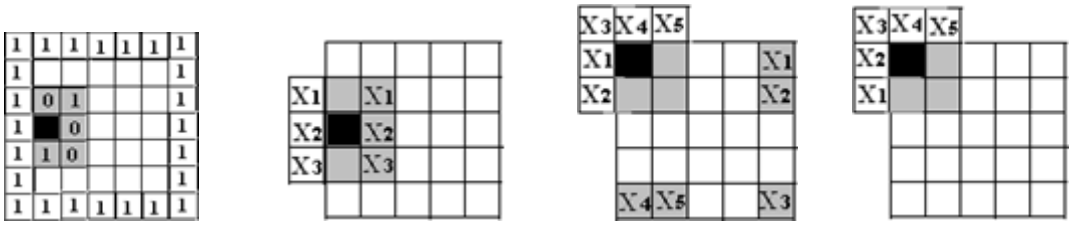


Figura 1. Condiciones de Frontera de AC en 2D con vecindad de Moore de radio 1.

Conjunto de Estados: Son un conjunto de valores o elementos que se definen para un AC, el caso más sencillo corresponde a dos estados (1s y 0s), pero el número de estados puede ser tan grande como lo requiera el problema con un tamaño determinado.

Estado: Es el valor que toma una *célula* en un instante de tiempo *t*. Este valor debe pertenecer al conjunto de estados.

Vecindario: es el entorno que rodea y afecta a una *célula* en un rango de distancia *r* y deben ser iguales para todas las células [9], entre las vecindades más comunes se encuentran La vecindad de von Neuman (Figura 2a) y la vecindad de Moore (Figura 2b). La vecindad no está limitada a una forma específica, puede tomar cualquier configuración (Figura 2c).

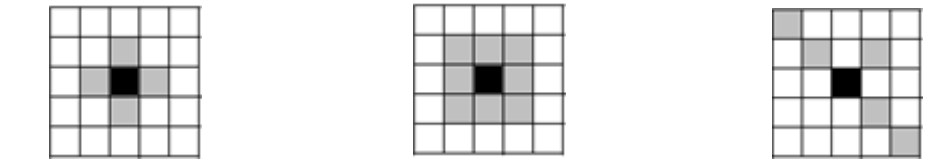


Figura 2. Vecindarios de AC en 2D

Reglas: Definen el comportamiento del sistema. Determinan el estado de una *célula* en un instante de tiempo *t + 1* teniendo en cuenta su estado y el de sus *vecinas* en un instante *t*. Pueden ser de tipo determinístico o estocástico [9].

2.1.4 Metaheurísticas

Es un término que aparece por primera vez en 1986 al introducir el concepto de búsqueda tabú (tabu search) [26-27], las metaheurísticas son métodos aproximados diseñados para resolver problemas difíciles de optimización combinatoria, donde otros métodos no son efectivos. Visto de otra manera se conciben las metaheurísticas como un método de solución a problemas en búsqueda de mejoras y estrategias de alto nivel, donde se combinan diferentes procedimientos heurísticos de tal forma que sean capaces de realizar búsquedas concretas en un gran espacio de soluciones y lograr encontrar un óptimo

cercano a la solución, en donde los procedimientos empleados tiene un grado de aleatoriedad para hallar una o varias soluciones óptimas a problemas difíciles, es decir problemas que pertenecen al tipo de complejidad NP-Hard y NP-completo [26, 28-29].

Cuando se habla de metaheurísticas hay que tener en cuenta dos características importantes, por un lado la *diversificación* que se refiere a la exploración del espacio de búsqueda, donde se evalúan las diferentes regiones distantes en un espacio de búsqueda para encontrar la solución al problema y por otro lado la *intensificación* que se refiere a la explotación del espacio de búsqueda, donde se realizan búsquedas en un espacio concreto para encontrar la mejor solución. Estas características deben estar en equilibrio ya que la búsqueda debe ser lo bastante rápidas, para identificar las regiones prometedoras con buenas soluciones en el espacio de búsqueda, de tal forma que no se malgaste tiempo en regiones ya exploradas o con soluciones de poca calidad [29-30].

Las metaheurísticas según sus características se pueden clasificar en inspiradas y no inspiradas por la naturaleza, deterministas y estocásticas, con memoria y sin memoria entre otras, pero tal vez la clasificación más importante son las basadas en trayectoria y las basadas en población.

Las basadas en trayectoria son las metaheurísticas que están enfocadas en una sola solución, se caracterizan por ser orientadas en la explotación del espacio de búsqueda, donde la solución se va actualizando constantemente formando una trayectoria. Las metaheurísticas basadas en trayectoria son [26, 31]:

1. Enfriamiento simulado (SA).
2. Búsqueda Tabú (TS).
3. Búsqueda loca Iterada (ILS).
4. Búsqueda con vecindario variable (VNS).
5. GRASP.

Las metaheurísticas basadas en población, a diferencia de las basadas en trayectoria, se enfocan en un conjunto de soluciones, recorriendo vecindarios en un espacio de búsqueda por iteración, se caracterizan por ser orientadas por la exploración. Las metaheurísticas basadas en población son:

1. Algoritmos Genéticos (AG).
2. Algoritmos de estimación de la distribución (EDA).
3. Optimización basada en colonias de hormigas (ACO).
4. Optimización basada en enjambres de partículas (PSO).
5. Búsqueda Dispersa (SS).
6. Algoritmos memèticos (MA).
7. Búsqueda Armónica (HS).

2.1.5 Algoritmos Genéticos (AG)

Los algoritmos genéticos (AG), son una de las principales estrategias para resolver problemas de búsqueda, optimización combinatoria y aprendizaje de máquina, sus principios básicos fueron consolidados por John Holland en 1975 [30]. Los AG están basados en la teoría de la evolución de Darwin, donde en cada generación se busca la supervivencia de los individuos más aptos para garantizar la supervivencia de la especie.

Los algoritmos genéticos trabajan simulando el comportamiento de la naturaleza y también toman muchos términos de estos procesos para su entendimiento y realización, primero se selecciona una población, se escogen los individuos más aptos a través de una función objetivo, se seleccionan dos individuos a los que se llaman padres, se cruzan los padres para obtener individuos para la siguiente generación a los cuales se le llamara hijos, y se tendrá en cuenta un proceso de mutación en los hijos para asegurarse que todos los puntos del espacio de búsqueda tengan probabilidad de ser examinados [30-31].

En general los algoritmos genéticos funcionan de la siguiente forma:

Generar una población inicial de soluciones: Se selecciona la población inicial garantizando la diversidad estructural para tener una mayor parte de la población.

Evaluación: A cada individuo de la población se le aplica la función objetivo para saber que tan buenas son las características de sus cromosomas.

Condición de finalización: El AG finaliza cuando se alcanza una solución óptima, pero como ésta no siempre se conoce, se tienen otros criterios de finalización como lo son un número máximo de generaciones. Mientras el algoritmo no alcance la condición de salida se deben realizar las siguientes operaciones [2, 26, 30-31]:

- Selección: Se eligen los individuos con mejores aptitudes para generar la siguiente generación.
- Cruce: Es el principal operador genético, representa la reproducción sexual, donde los cromosomas de los padres son cruzados para generar un nuevo cromosoma.
- Mutación: Generalmente modifica uno pero pueden ser más genes del cromosoma hijo.
- Reemplazo: Una vez se aplican los operadores genéticos, se seleccionan los individuos con mejores características que conformaran la siguiente generación.

Ventajas y desventajas:

- Son fáciles de usar en arquitecturas paralelas.
- Pueden tardar mucho en encontrar los máximos o mínimos óptimos, o no encontrar el máximo o mínimo absoluto, dependiendo de los parámetros seleccionados.
- Pueden converger prematuramente debido a problemas de diferentes índoles.

2.1.6 Búsqueda Tabú (TS)

Es una metaheurística propuesta por Fred Glover [32] en 1986, esta estrategia busca solucionar problemas de optimización combinatoria o problemas de tipo NP-hard. Búsqueda tabú (TS de las siglas en ingles) está basado en el principio de es mejor una mala elección basada en información a una buena elección basada en el azar, debido a que una mala elección basada en una estrategia proporciona información útil para continuar con la búsqueda, a diferencia de una buena elección al azar que no proporciona ninguna información [33-34].

TS se basa en el uso de memoria adaptativa, la cual mantiene un registro de soluciones o movimientos que ya fueron visitados. El proceso comienza con una solución inicial que puede ser elegida al azar, y a partir de ésta se genera un conjunto de soluciones conocido como vecindario, al cual se le aplica una función objetivo que permite escoger la mejor solución que no esté dentro de la memoria y se repite el proceso con la mejor solución encontrada hasta un determinado punto de parada. Esta implementación busca superar el estancamiento del optimó local, a través del uso de memoria la cual se conoce como lista tabú o memoria de corto plazo [9, 28, 35]. Además de la lista tabú y el vecindario TS, se tienen en cuenta otras características para el proceso de TS descritas a continuación:

- **Restricciones Tabú:** El concepto de tabú (Usado comúnmente como marca de prohibido) es un distintivo de TS que se usa para prevenir ciclos en el instante que no hayan mejoras en la búsqueda, una restricción tabú indica que ya se buscó en ese sitio y no arrojó ninguna información prometedora en la búsqueda, por lo general estos tabúes se almacenan en la lista tabú, estos tabúes pueden ser movimientos o soluciones anteriormente visitadas [31]. A estas restricciones tabú se les puede asociar un contador que permita verificar de nuevo un sitio de búsqueda en una determinada cantidad de iteraciones, ya que tal vez sea prometedora la búsqueda en ese lugar, se debe tener en cuenta que en ocasiones los tabúes pueden prohibir sitios prometedores de la búsqueda y hasta pueden causar estancamientos en la búsqueda [32, 34].

- **Criterio de aspiración:** Consiste en anular una restricción tabú, es usada cuando las restricciones prohíben sitios prometedores de la búsqueda. El criterio más usado es permitir un movimiento así sea tabú si este resulta ser un mejor valor de la función objetivo que el de la mejor solución conocida actualmente [32, 34].

- **Criterio de Parada:** Hay diferentes tipos de parada del proceso de búsqueda tabú, entre los más importantes tenemos [31]:

    - Terminar después de un número de iteraciones fijo.
    - Terminar después de un número de iteraciones donde no haya mejora de la función objetivo.
    - Terminar si la función objetivo alcanza un valor máximo establecido con anterioridad.

La búsqueda tabú descrita hasta el momento soluciona muchos problemas de optimización con éxito, pero hay otros elementos que en la mayoría de los casos son útiles para alcanzar un nivel de eficacia, los más importantes de estos son la intensificación y diversificación.

Intensificación: está basada en un tipo de memoria intermedia, tal como una "experiencia reciente de memoria" (Recency memory), es decir que se recuerda lo ultimo bueno conocido. Esta estrategia se basa en regresar a las regiones del espacio de búsqueda ya exploradas para ser estudiadas más a fondo, en donde se registra y se comparan las características de las mejores regiones conocidas durante el proceso de búsqueda, esta estrategia busca soluciones nuevas donde se presentan estas características. Un enfoque típico de esta estrategia es reiniciar la búsqueda en la mejor solución conocida actualmente [31, 34].

Diversificación: está basada en un tipo de memoria a largo plazo de la búsqueda, tal como una "memoria de frecuencia" (frequency memory), este tipo de estrategia se basa en visitar la regiones del espacio de búsquedas no exploradas, tomando puntos de partida a través de una proceso de búsqueda heurística para generar un punto de partida con un objetivo, en vez de uno escogido al azar.

Este tipo de estrategia permite aprender de la búsqueda desde su inicio, registrando el número total de iteraciones que contengan características con información útil de la solución. Es decir la memoria de largo plazo utiliza criterios de evaluación que indica todas aquellas soluciones con características similares al óptimo durante el proceso de ejecución de la búsqueda [31, 34].

2.2 ANTECEDENTES

2.2.1 Diseño Inverso de Autómatas Celulares

En muchos casos se conocen todas las características (configuración inicial, vecindades y reglas de transición) para la simulación de un sistema, pero en otros debido a la complejidad de los problemas o fenómenos no se conocen todas sus características, pero a partir de evidencias relacionadas con los mismos se pueden inferir dichas características, esto es lo que se conoce como diseño inverso [6] y esta será la definición asumida en el resto del documento.

En AC se han realizado trabajos de investigación con diseño inverso en diferentes áreas de investigación. En los artículos [10, 25, 33, 36-39] se pueden ver diferentes técnicas para determinar parámetros de un AC como lo son la programación evolutiva, la lógica difusa, los árboles de clasificación y decisión, junto con minería de datos, muchos de estos problemas son planteados con AC binarios de 1D y 2D donde se buscan las mejores vecindades o las mejores reglas de transición que representen el sistema.

En [10], se propone la utilización de un algoritmo genético, en donde los autores utilizan las vecindades de Moore y Von Newman en AC de dos dimensiones, y este es llevado a una representación de mapas de bits, en donde a partir de unas configuraciones iniciales se busca un conjunto de reglas que lleve al AC a una configuración objetivo preestablecida. Este tipo de enfoque es útil en fenómenos donde solo se tienen las configuraciones globales del problema correspondiente a los estados inicial y final. Sin embargo, en fenómenos complejos el uso ¿de este enfoque resulta computacionalmente costosa, debido a que no se limita el espacio de búsqueda de las reglas.

En [36] se busca una nueva metodología para la codificación de reglas de AC a través de AG y Redes Neuronales, permitiendo seleccionar el nivel de complejidad mediante restricciones del espacio de búsqueda para mejorar la calidad y el tiempo de búsqueda de las reglas. El trabajo se realiza con un AC de 1D y dos estados, se establece una clasificación jerárquica del espacio de reglas y se demuestra que casi todas las reglas de alto rendimiento estaban en un sub-espacio muy pequeño y estructurado, lo que facilitó la obtención de reglas de alto nivel. Sin embargo este modelo puede ser desventajoso, debido que si el conjunto de datos de vecindad es demasiado grande puede afectar la precisión del modelo si la vecindad del AC particular no es considerada como opción inicial. En muchos problemas de optimización se utilizan técnicas basadas en el gradiente para su solución, pero Cranny demuestra que en estos espacios de búsqueda esta no es la mejor opción.

En [33], se crea una heurística que encuentra una vecindad, la cual permite organizar la información de configuraciones globales en evidencias de configuraciones locales, representado patrones que contienen cambios. Se utilizan las evidencias para ser clasificadas mediante un AG, expresando las reglas del AC en forma arboles de decisión. Una de las debilidades de éste enfoque es la restricción de los modelos de AC en una dimensión lo cual es poco viable debido que los datos a procesar se caracterizan por contener información del espacio 3D

En [38] Se busca encontrar una técnica para detectar vecindades basándose en funciones polinomiales para representar reglas de transición local del modelo de AC, están son seleccionadas mediante una aproximación inicial que se refina a través de un algoritmo de detención de vecindades. El conjunto de reglas se infiere a través de algoritmos genéticos con restricciones enteras, donde el mayor inconveniente de esta técnica, es la baja eficiencia cuando una vecindad es demasiado grande ya que se amplía el espacio de búsqueda haciéndolo computacionalmente costoso.

En [39] Yang y Billings realizan una detección de nuevas vecindades basándose en el concepto de información mutua (IM), este método mide la dependencia entre dos variables y permite representar las reglas de transición de un AC como simples modelos polinomiales. Este enfoque de detección de vecindades produce un rango de vecindades correctas que reduce considerablemente el espacio de búsqueda, pero no siempre

garantiza la vecindad exacta, por lo que en algunos casos debe usarse el algoritmo de mínimos cuadrados ortogonales.

2.2.2 Metaheurísticas Híbridas

La combinación de una metaheurística con otra técnica o algoritmo es lo que se conoce como metaheurística híbrida, estas combinaciones pueden ser entre metaheurísticas, algoritmos de inteligencia artificial(IA), algoritmos de minería de datos(DM), arboles de búsqueda e investigación de operaciones [40-41]. Tienen como objetivo explotar las mejores características de dos o más algoritmos, teniendo en cuenta las ventajas y desventajas de cada uno de ellos, para dar una respuesta más optima a un problema.

La hibridación de metaheurísticas con otra metaheurística es la más popular debido que se combinan las ventajas de una metaheurística basada en trayectoria con las de una basada en población; comúnmente en este tipo de híbrido se utiliza la metaheurística basadas en trayectoria dentro de la basada en población para mejorar el rendimiento de esta [40-41], a este tipo de hibridación se le conoce como algoritmos Memèticos (MA) o lamarckiano.

En [11] se describe una serie de pautas para la creación de metaheurísticas hibridas para la solución a problemas de optimización combinatoria; los diseños de estos híbridos se enfocan principalmente en algoritmos evolutivos (EA) como los AG, a los cuales se le puede aplicar heurísticas, búsquedas locales u otras técnicas para generar la población inicial del EA. Otra forma de hibridación es aplicar búsqueda local (SA, TS, Etc) sobre la población del EA mejorando la descendencia de la población.

En [42], se hace énfasis en búsquedas locales dando prioridad a Búsqueda tabú(TS), describe diferentes casos de éxito de hibridación entre EA y TS. La sección 5 de este libro presenta la implementación de un algoritmo híbrido entre AG y TS en donde a la descendencia del AG es mejorada con TS, además presenta diferentes variaciones en la implementación de TS para este problema. En la sección 6 de este libro realizan la hibridación de un AG y con algunas características de TS, donde se implementa el uso de memoria de TS para evitar que el AG regrese a individuos previamente visitados; éste mecanismo junto con una mejora de la diversidad de población del AG, permiten una mejora en la convergencia del algoritmo y reduce la carga computacional del algoritmo.

2.2.3 Metodología y Framework Computacional para el Diseño Inverso de Modelos de Autómata Celulares (CAIF-PFT)

CAIF-PFT es un framework propuesto en [12] que facilita el desarrollo, evaluación y una arquitectura base para la implementación de estrategias basadas en técnicas de computación evolutiva y metaheurísticas para la identificación de modelos de AC a partir de trayectorias de plegamiento de proteínas, tomando como marco metodológico CRISP-DM.

CAIF-PFT permite el pre procesamiento de una trayectoria simulada de plegamiento de proteínas en formato PDB a mapas de contacto de tal forma que sea similar a las configuraciones globales de un AC (Malla 2D, Estados, etc.). En este framework se ha realizado con éxito la implementación de un Algoritmo Genético, presentando buenos resultados en la identificación de AC [13]. Este AG se tomará como base de implementación para el desarrollo de la técnica hibrida a desarrollar.

3 METODOLOGÍA CRISP-DM APLICADA PARA LA IDENTIFICACIÓN DE MODELOS DE AC EN TRAYECTORIAS DE PLEGAMIENTO DE PROTEÍNA

En esta sección se describe la metodología de alto nivel CRISP-DM [43] (Cross Industry Standar Process for Data Mining), que se aplicará para alcanzar los objetivos propuestos del proyecto, dado que el proceso de identificación de modelos de AC, es similar a un proceso de minería de datos.

CRISP-DM, tiene seis fases de modelado tal como lo muestra la **Figura 3**. La primera fase es el entendimiento del negocio en donde se definen los objetivos y plan de proyecto de minería de datos. La segunda fase corresponde al entendimiento de los datos en donde se realiza la recolección y familiarización de los datos. En la tercera fase se realiza la preparación de los datos, que facilita el uso de los mismos en los modelos de minería. La Cuarta Fase concierne al Modelo, aquí se selecciona la técnica más adecuada para el proyecto de minería de datos. En la Fase Cinco, se evalúan los modelos obtenidos en la fase anterior. La fase Seis corresponde al Despliegue, en donde el conocimiento obtenido en las fases anteriores se debe documentar y presentar de manera comprensible. Las características especificas de la metodología se plasmaran en el resto del capítulo, basados en la Guía de Usuarios de CRISP-DM [43].

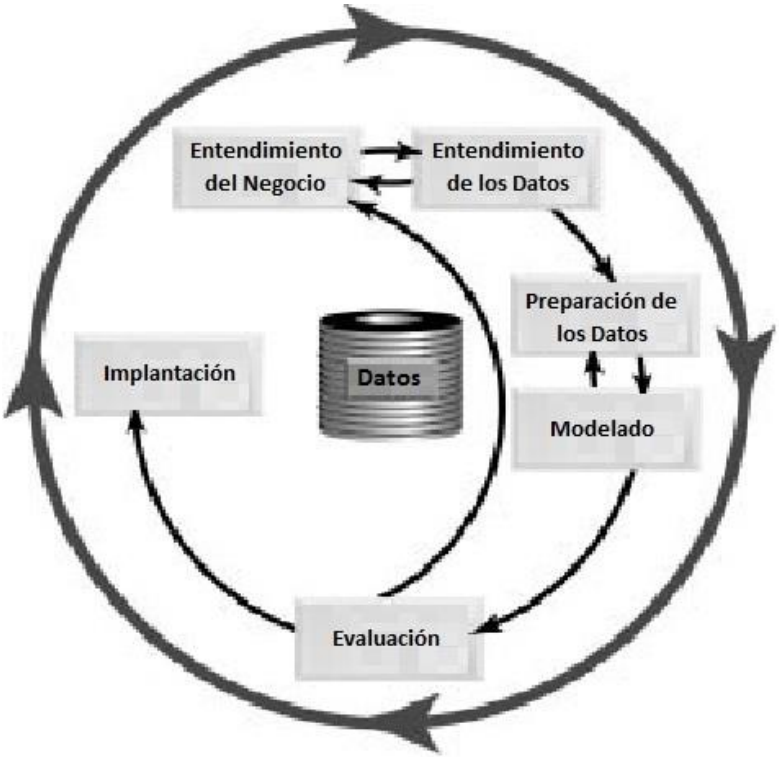


Figura 3. Modelo de CRIPS-DM (Adaptado de [43]).

3.1 FASE 1: COMPRESION DEL NEGOCIO

3.1.1 Determinar los Objetivos del Negocio

3.1.1.1 Contexto

El fenómeno de plegamiento de proteínas es centro de estudio en diversas áreas del conocimiento, las cuales buscan un mejor entendimiento de este fenómeno; en el área computacional existen dos técnicas importantes: la simulación por Montecarlo y Dinámica Molecular, estas técnicas obtienen trayectorias de plegamiento brindando información de la posición 3D de los átomos involucrados en una proteína. Estas técnicas son computacionalmente costosas debido al tiempo de CPU que requieren, es por eso que se buscan otras técnicas de simulación como los Modelos de Autómatas Celulares que computacionalmente son eficientes e idóneos para la simulación de problemas complejos, utilizando técnicas de aprendizaje maquina.

Las técnicas de aprendizaje de maquina junto con trayectorias obtenidas con simulación de dinámica molecular, serán utilizadas para obtener modelos de AC que representen el fenómeno de plegamiento de proteínas.

3.1.1.2 Objetivo del Negocio

En el contexto anterior el objetivo del negocio está enfocado en obtener un modelo de simulación basado en Autómatas Celulares, que replique con una precisión del 90% o superior una trayectoria de simulación del plegamiento de proteínas.

3.1.1.3 Criterios de Éxito del Negocio

Mejorar el proceso de búsqueda realizado por el AG propuesto en [12], realizando un algoritmo híbrido entre AG y TS, replicando la trayectoria de simulación de plegamiento de proteínas.

3.1.2 Evaluación de la Situación Actual

3.1.2.1 Inventario de Recursos

Se cuenta con el framework computacional CAIF-PFT para el diseño inverso de modelos de AC [12], que además de brindar una arquitectura adecuada para el desarrollo de técnicas o metaheurísticas enfocada al problema, permite el procesamiento de datos de una trayectoria de simulación de plegamiento de proteínas disponibles en el repositorio público del sitio web http://www.simtk.org, correspondiente al conjunto de datos de trayectoria de plegamiento de la proteína HP35-NleNle.

Además se cuenta con 15 computadores de escritorio con procesador Intel Core Duo 2.0Mhz y 2 GB de memoria RAM, que trabajarán de forma distribuida, debido a la cantidad de datos derivados de la simulación de plegamiento de proteínas.

Herramientas de minería de datos como Orange [44], que apoya diferentes tareas de minería como el modelado, procesamiento de datos, evaluación, entre otras, que cuenta además con un entorno gráfico agradable, desarrollado en python, lo que permite desarrollar scripts de minería de datos simples y de gran alcance.

3.1.2.2 Requerimientos, Asunciones y Restricciones

Simular el plegamiento de proteínas, presenta dificultades con respecto al tiempo de CPU que requiere para llevar la proteína a su estado nativo, las trayectorias utilizadas no necesariamente van a llegar hasta la estructura nativa, pero se espera que la información contenida en estas trayectorias parciales contenga el conocimiento suficiente del proceso de plegamiento.

Se espera que al finalizar el proyecto se obtengan modelos de autómatas celulares que repliquen la precisión alcanzada con el AG en [12] o superior a éste.

3.1.2.3 Riesgos y Contingencias

Tabla 1. Tabla de riesgos y contingencias

Riesgo	Descripción	Contingencia
Recursos Computacionales Limitados	La falta de disponibilidad de salas de cómputo, podría ocasionar poca experimentación de la técnica de minería de datos, esto no permitiría un buen ajuste de parámetros, lo que conlleva a una poca fiabilidad en los resultados.	Implementar un reinicio para la técnica seleccionada que permita continuar con el experimento.
Mal Entendimiento de los datos	Podría generar ocasionales retrasos en la ejecución del proyecto, debido a que una mala compresión de los datos puede llevar a falsas evaluaciones y esto implica la no consecución de los objetivos planteados.	Consultar con un experto, para un buen entendimiento de los datos.

Riesgo	Descripción	Contingencia
La técnica metaheurística no es la adecuada	Corresponde a una mala elección de la técnica de minería de datos, lo que conlleva a modelos no esperados y al no cumplimiento del objetivo de minería de datos.	Implementar diferente variantes de la técnica híbrida y hacer los afinamientos de parámetros correspondientes para la consecución de los objetivos.
Actividades incorrectamente definidas del plan del proyecto	Las actividades mal definidas podrían ocasionar retrasos en otras actividades, lo que podría ocasionar un retraso general en el proyecto.	Revisión de las actividades, y constante seguimiento de estas para evitar inconvenientes.

3.1.2.4 Costos y Beneficios

Como beneficios del proyecto se obtendría una técnica que mejore el proceso de búsqueda propuesto en [12], que nos llevaría a un mejor entendimiento de las reglas que determinan el fenómeno de plegamiento de proteínas. Los costos del proyecto se pueden ver en la tabla 2.

Tabla 2. Tabla de recursos

Tabla/Justificación	Recursos		Total
	FIET - Sistemas	Estudiantes	
Personal	1.314.720	24.651.000	25.965.720
Equipo	0	420.000	420.000
Software:Java, Python, Linux, Open Office	0	0	0
Viajes y Salidas de campo	0	0	0
Bibliografía	0	150.000	150.000
Materiales	0	200.000	200.000
Servicios técnicos	0	0	0
Publicaciones	0	0	0
Administración	0	0	0
Comunicaciones	75.000	675.000	750.000
Otros	0	0	0
TOTAL	**1.389.720**	**26.096.000**	**27.485.720**

3.1.2.5 Terminología

Los términos utilizados en el proyecto son los utilizados en [12], los cuales son conceptos relacionados con el plegamiento de proteínas y las trayectorias simuladas, los más importantes serán descritos a continuación:

Aminoácidos - AA*:* son compuestos orgánicos que se combinan para formar proteínas la secuencias de estos dan origen a las proteínas. En su estructura los AA son más o menos uniforme en el sentido de poseer un carbono alfa, un átomo de hidrógeno, un grupo amino, un grupo carboxilo y una cadena lateral [12].

Cadena lateral: Componente de un AA con mayor flexibilidad en lo relativo a tamaño, forma, capacidad de realizar enlaces de hidrógeno y distribución de carga . Todas estas características, permiten que las proteínas puedan desempeñar una gran cantidad de funciones biológicas [12].

Trayectoria de plegamiento: Es el conjunto de los estados por el cual pasa una proteína desde una configuración no plegada hasta su estructura nativa. Una trayectoria de plegamiento de proteína en formato PDB, se compone de un conjunto de modelos PDB [12].

Plegamiento de Proteínas: Habilidad de todos los seres vivos de ensamblar de forma correcta los aminoácidos (AA) y llevarlos a una forma tridimensional, generando proteínas con características y funciones específicas.

Trayectoria de simulación: Es el conjunto de configuraciones por las cuales el sistema transita en cada paso de simulación.

Mapa de contacto: Representación simplificada de la estructura 3D de los AA de una proteína. Ésta representación se basa en una matriz 2D, en la cual se registra el estado de contacto o no contacto entre cada par de AA presente en la proteína. Usualmente, el átomo que se considera es el Ca o el Cb de cada AA [12].

PDB: Banco de datos de proteína (Protein Data Bank), es un repositorio de información de estructura de proteína.

Modelo PDB: Conjunto de coordenadas en el espacio 3D de los átomos pertenecientes a un sistema proteína – solvente [12].

Modelo de simulación: Es una representación del funcionamiento de un proceso, fenómeno o sistema, generalmente ocurrente en el mundo real, sobre una escala de tiempo .

Paso de simulación: Es uno de los instantes de tiempo en los cuales se realiza la simulación de un sistema. A cada paso de simulación se le asocia un estado del sistema que representa el modelo de simulación.

Rango de Contacto: Es el rango en el cual se considera que hay contacto entre AAs de una secuencia. El rango se determina por un límite inferior y un límite superior, usualmente se toma como punto de referencia la distancia entre los C□ de cada par de AA si esta distancia se encuentra en el rango de contacto se dice que hay contacto entre el par de AA correspondiente [12].

3.1.3 Determinar Objetivos de la Minería de Datos

3.1.3.1 Objetivos de la Minería de Datos

Especificar una técnica hibrida entre AG y TS que mejore el proceso de búsqueda del AG propuesto en [12], que recibirá como entradas el conjunto de datos de una trayectoria de plegamiento de proteínas, y un modelo AC parcialmente definido (conjunto de estados, condiciones de frontera y lattice). Identificando modelos de Autómatas Celulares completos.

3.1.3.2 Criterios de Éxito de la Minería De Datos

La técnica seleccionada debe replicar los modelos de autómatas celulares con una precisión del 90% o superior, esto teniendo en cuenta los criterios de medición como especificidad [45], precisión [45], Coeficiente de correlación de Matthews [46], Curva ROC [47], Matriz de confusión; utilizados como medidas [12].

3.1.4 Construir Plan del Proyecto

3.1.4.1 Plan Del Proyecto

A continuación se presentan el plan del proyecto, el cual estará guiado por dos metodologías, la primera es la encargada de la minería de datos, en este caso la metodología CRIPS-DM que debido a su adaptabilidad a las necesidades del proyecto es ideal para la consecución de los objetivos del proyecto. Por otro lado para el desarrollo se eligió la metodología XP, para la cual se adoptaran diferentes artefactos como historias de usuario, plan de iteración, plan de pruebas y metáforas, esta metodología estará inmersas dentro de la fase de modelado de CRISP-DM. A continuación una breve descripción de las fases del plan de proyecto.

- **Fase I. Compresión del Negocio:** En esta fase es donde se establecerán objetivos del negocio y del proceso de minería de datos, la evaluación actual del problema y se definirá el plan del proyecto.

- **Fase II. Comprensión de Datos:** En esta fase del proyecto se realizaran tareas como la recolección, exploración y verificación de los datos definiendo un tratamiento y formalización para estos.

- **Fase III. Preparación de los Datos:** En esta fase se escogerá el dataset, además se realizara una descripción de éste y se procede con la preparación de los datos,

que está ligado a la selección de los mismos, haciendo limpieza, construcción y generación de nuevos datos.

- **Fase IV. Modelado:** En esta fase se inicia el proceso de modelado según la metodología CRISP DM, en donde se seleccionara una técnica para resolver el problema, en este caso la hibridación entre Algoritmo Genético y tabu search, En esta fase se utilizan las iteraciones de XP.

    - **Primera Iteración:** En esta Iteración se analizara la técnica escogida, su función objetivo y características de la metaheurística hibrida.
    - **Segunda Iteración:** En esta iteración se continúa con el proceso de modelado, en donde se definirán algunas características de tabu search, como son las estructuras de vecindad, criterios de aspiración, y lista tabú.
    - **Tercera Iteración:** En esta iteración se definirán la memoria a mediano plazo (intensificación) y la memoria a largo plazo (diversificación), de tabu search.
    - **Cuarta Iteración:** En esta iteración se inicia con el plan de pruebas y la evaluación del modelo de diseño.
    - **Quinta Iteración:** En esta iteración iniciara con la construcción del modelo y generación del mismo.
    - **Sexta Iteración:** En esta iteración se describe el modelo generado de AC, se procederá a realizar la evaluación modelo y según los resultados obtenidos al realizar la evaluación, se efectuaran las respectivas modificaciones sobre el modelo de AC.

- **Fase V. Evaluación:** En esta fase se evalúa el modelo construido, revisando cada uno de los pasos para su creación, comprobando que cumpla con los objetivos propuestos, evaluando los resultados, determinando los próximos pasos a seguir. Se revisará el proceso, listando las posibles acciones y decisiones a tomar.

- **Fase VI. Despliegue:** En esta fase se realiza el plan del despliegue de la solución, para de esta manera terminar con la aplicación y la metodología CRISP-DM.

- **Documentación:** se documenta los hallazgos más relevantes de las fases anteriores, junto con los resultados de nuestro trabajo y las experiencias ganadas a lo largo de este proyecto.

La Figura 4 presenta el cronograma en donde se reflejan las fases anteriormente mencionadas.

19	Fase II. Compresion de los Datos
20	Recolectar y describir datos iniciales
22	Explorar y verificar la calidad de los datos
24	Ajuste Compresion de de Datos
25	Fase III. Preparación de Datos
26	Selección de los datos.
27	Construcción, Limpieza y Transformación de los datos.
28	Inclusión / Exclusión de datos
29	Creación de atributos derivados a partir de otros.
30	Transformación de sus valores
31	Análisis de posibles creaciones de registros nuevos
33	Fase IV. Modelado
34	Primera Iteracion
35	Seleccionar las técnicas de modelado.
37	Supuestos del modelo
42	Segunda iteracion
43	Definir Caracteristicas Tabu
49	Tercera Iteracion
50	Definir Memoria de Mediano y largo Plazo
53	Cuarta Iteracion
54	Generar plan de Pruebas
58	Quinta Iteracion
59	Construcción y descripción de los modelos AC
62	Sexta Iteración
63	Descripción del modelo
64	Evaluación del Modelo
65	Fase V. Evaluación
66	Evaluar los resultados
67	Determinar los próximos pasos a seguir
69	Revisar el proyecto
72	Fase VI. Despliegue
73	Plan de Implementación
74	Generar informes del proyecto con cada unas de sus etapas
75	Producir reportes finales y documentar las experiencias
76	Documentación

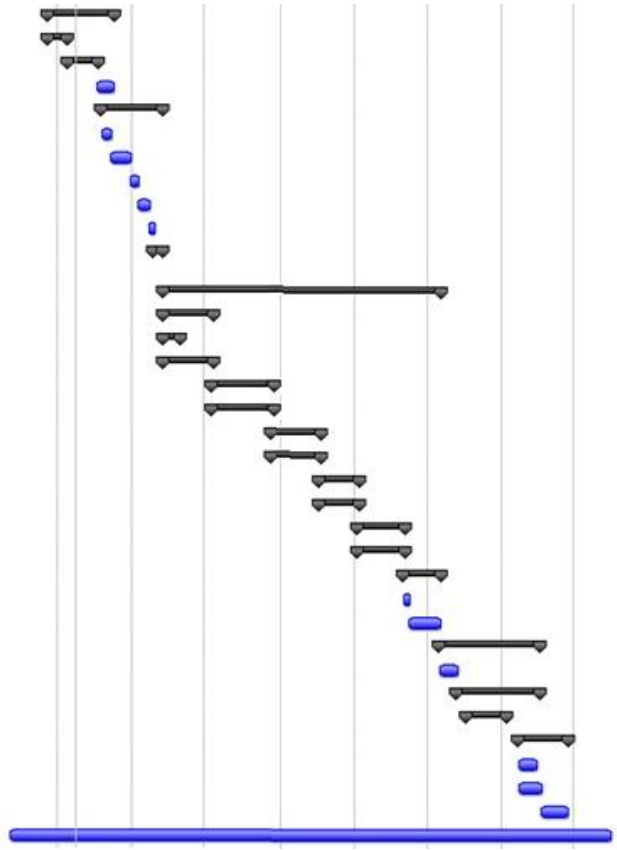


Figura 4.Cronograma de Actividades.

3.1.4.2 Evaluación Inicial de Técnicas y Herramientas

Como técnicas se tiene en cuenta el algoritmo Genético propuesto en [12] y evaluado [13], que facilita la obtención de modelos de Autómatas celulares a partir de información de trayectoria de plegamiento de proteínas.

Además se tiene un algoritmos de búsqueda local en este caso Búsqueda Tabú (TS) descrito en Anexo 2, en donde se encuentran modelos de AC con una precisión alrededor del 90%.

3.2 FASE 2: ENTENDIMIENTO DE LOS DATOS

3.2.1 Reporte Inicial de la Recolección de Datos

Los datos proporcionados para el problema están dados por la trayectoria simulada de plegamiento de proteínas conocida como Hp35-NleNle, disponibles en formato PDB, del repositorio SimTk descritas en [12], y que serán usadas en el presente proyecto; por lo cual se usara la proteína a la que se tiene acceso en el servidor del proyecto SimTk (i.e. https://simtk.org/home/foldvillin).

Para acceder a los archivos de las trayectorias simuladas es necesario registrarse como usuario, autenticarse e ingresa a la página Web del proyecto (i.e. http://archive.simtk.org/traj/project/PROJ3036?format=html), en donde se encontraran los enlaces de 10 trayectorias obtenidas por dinámica molecular, estas son nombradas como RUN0 al RUN9.

Las trayectorias simuladas que se estudiaran corresponden a la simulación 5 y 8 (RUN4 y RUN7), por ser la escogidas en [12], las cuales brindan mayor similitud con respecto a la estructura nativa de la proteína.

3.2.1.1 Descripción y exploración de los Datos

Las trayectorias simuladas se divide en marcos de simulación que en total son 100, cada marco agrupa 401 pasos de simulación, en donde el último paso de un marco corresponde al paso inicial del siguiente marco de simulación.

Los archivos contienen modelos PDB por cada paso de simulación, estos modelos a su vez contienen información como el método de generación, tiempo de simulación, estructura de simulación, el número de modelo y la información 3D de los átomos presentes en la simulación. La **Figura 5** muestra un ejemplo de un modelo PDB del repositorio SimTk [12].

```
REMARK    GENERATED BY TRJCONV
TITLE     SMP-ensv-03 t=   0.00000
REMARK    THIS IS A SIMULATION BOX
CRYST1   42.262   49.547   46.483  90.00  90.00  90.00 P 1           1
MODEL        0
ATOM      1  CA  NLE   1      24.660  30.640  36.370  1.00  0.00
ATOM      2  CA  SER   2      22.250  28.460  38.450  1.00  0.00
ATOM      3  CA  ASP   3      20.580  25.660  36.590  1.00  0.00
                         ...
ATOM     33  CA  GLY  33      16.770  17.990  21.780  1.00  0.00
ATOM     34  CA  LEU  34      14.340  19.600  24.530  1.00  0.00
ATOM     35  CA  CPH  35      13.820  20.720  28.200  1.00  0.00
TER
```

Figura 5. Ejemplo de un modelo PDB (Tomado de [12]).

3.2.1.2 Verificar la Calidad de los Datos

Para verificar la calidad de los datos se realizó una rutina para cada marco de simulación el cual cuenta los 401 modelos PDB y los carbonos alfas de cada modelo, se lanza como resultado el siguiente reporte que se ilustra en la Figura 6.

```
C:\CAlpha\models\frame95.pdb
Model 0000 - numero de atomos 35
Model 0001 - numero de atomos 35
Model 0002 - numero de atomos 35
Model 0003 - numero de atomos 35
.
.
.
Model 0399 - numero de atomos 35
Model 0400 - numero de atomos 35

Cantidad de modelos 401
```

Figura 6. Reporte Calidad de los datos.

Este reporte se lanza para cada uno de los 100 marcos de simulación, y se verifica que los datos contengan los 35 átomos de la proteína, además que cada uno de los archivos contenga los 401 pasos de simulación, si el reporte no cumple con estas condiciones significa que la trayectoria está incompleta y los datos a usar no serian confiables.

3.3 FASE 3: PREPARACIÓN DE LOS DATOS

3.3.1 Selección de Los Datos

Dado que el proyecto busca replicar y mejorar los resultados obtenidos en [12], se utilizaron los datos usados en éste, los cuales son el RUN4 y RUN7, que llegan a una estructura nativa al final de la simulación. En el proyecto base se consideró la no inclusión de las moléculas de agua, reduciendo el sistema a solo los átomos de los AA, esto se hace para reducir la cantidad de datos y de esta forma analizarlos de forma razonable.

En la Figura 7 se observa un modelo PDB con los campos que este contiene y se considera la exclusión de los datos de Ocupación y factor de temperatura, debido a que sus valores no varían y no afecta la dinámica del sistema.

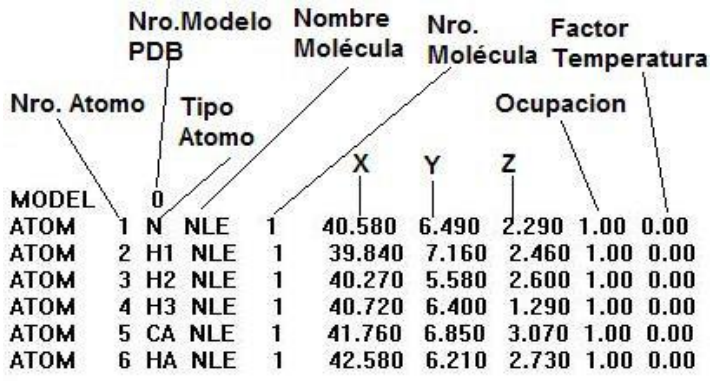


Figura 7. Ejemplo de campos contenidos en un Modelo PDB (Tomado de [12]).

3.3.2 Construcción, Limpieza y Transformación de Datos

Para el proceso de limpieza, construcción y formateo de datos el framework CAIF-PFT propuesto en [12], facilita este proceso, en consola se escribe la instrucción *python gacai-pft.py -p PDBDir,* En donde *-p* indica la opción para convertir la trayectoria simulada en formato PDB, y *PDBDir* es la dirección donde se encuentran los archivos de la trayectorias, La **Figura 8** muestra un ejemplo del uso de esta instrucción.

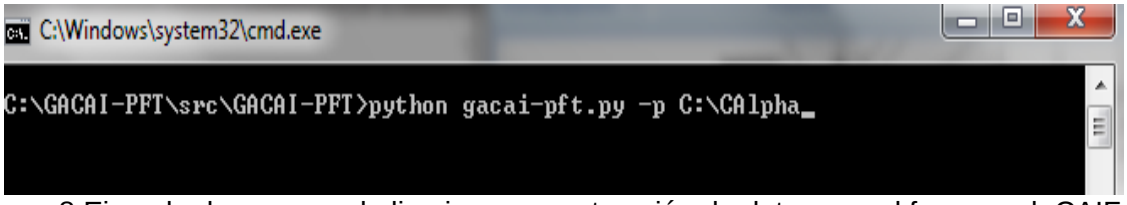


Figura 8.Ejemplo de proceso de limpieza y construcción de datos con el framework CAIF-PFT.

El proceso se ejecuta en dos fases, la primera realiza la limpieza de datos creando una carpeta llamada *models* en la cual por cada archivos de la trayectoria elimina los átomos que no hacen parte de la SAA y se excluyen las columnas de datos correspondiente a factor de ocupación y de temperatura, dado que estos atributos usualmente no varían y toman valores de 1.0 y 0.0, respectivamente, además el comportamiento constante de estos atributos, no son tenidos en cuenta por tener asociadas características dinámicas, al menos para el conjunto de datos considerado.

Los datos quedan organizados en modelos individuales, lo que facilita la construcción y formateo de los datos. Cada modelo PDB separado es equivalente a un paso de simulación. La **Figura 9** muestra el nuevo formato de los archivos, en donde se muestra el nombre del átomo, nombre de la AA y las coordenadas X, Y, Z de cada átomo.

ATOM	NAME	RES	NAME	X	Y	Z
	CA		NLE	24.660	30.640	36.370
	CA		SER	22.250	28.460	38.450
	CA		ASP	20.580	25.660	36.590
	CA		GLU	22.170	26.610	33.320
	CA		ASP	24.720	29.310	32.390
	CA		PHE	23.220	30.160	28.940
	CA		LYP	19.960	31.140	30.840
	CA		ALA	21.190	34.890	31.200
	CA		VAL	21.190	35.040	27.310
	CA		PHE	17.800	33.090	27.280
	CA		GLY	15.630	32.020	30.290
	CA		MET	14.090	29.250	28.400
	CA		THR	15.770	26.010	27.240
	CA		ARG	19.020	26.450	25.230
	CA		SER	17.510	24.380	22.320
	CA		ALA	14.650	26.940	21.680
	CA		PHE	17.120	29.920	21.160
	CA		ALA	18.530	29.240	17.680

Figura 9.Formato de modelo PDB después de la limpieza de datos.

La segunda fase realiza la construcción y formateo de datos, en primer lugar crea una carpeta que es llamada *models-ContactMaps* que contiene los mapas de contacto[5] por cada modelo PDB. Este proceso busca llevar los datos de cada paso de simulación a una forma discreta análoga a una configuración global de un modelo AC, con el fin de representar la estructura 3D en una matriz 2D, para manipular de manera más fácil los datos, además no presenta una pérdida definitiva de información, ya que es posible reconstruir la proteína a partir de un mapa de contacto y la SAA a través de algoritmos como los planteados en [48-49].

La configuración global tiene las siguientes características: Malla 2D, conjunto de estados, geometría de celdas y frontera del modelo AC. Al final del proceso se obtiene 40.000 mapas de contacto de 35 x 35 bits tal como muestra en la Figura 10.

4	3	1	1	0	0	0	0	0	0	0	0	0	0
3	4	3	1	1	1	0	0	0	1	0	0	0	0
1	3	4	3	1	1	1	1	0	0	0	0	0	0
1	1	3	4	3	1	1	0	0	0	0	0	0	0
0	1	1	3	4	3	1	0	0	0	0	0	0	0
0	1	1	1	3	4	3	1	0	0	0	0	1	0
0	0	1	1	1	3	4	3	1	0	0	0	0	0
0	0	1	0	0	1	3	4	3	1	0	0	0	0
0	0	0	0	0	0	1	3	4	3	1	1	0	0
0	1	0	0	0	0	0	1	3	4	3	1	1	0
0	0	0	0	0	0	0	0	1	3	4	3	1	1
0	0	0	0	0	0	0	0	1	1	3	4	3	1
0	0	0	0	0	1	0	0	0	1	1	3	4	3
0	0	0	0	0	0	0	0	0	0	1	1	3	4

Figura 10.Segmento de un mapa de contacto, después del proceso de construcción y formateo de datos.

3.4 FASE 4: MODELADO

En esta fase se propone una metaheurística híbrida basada en el Algoritmo Genético (AG) propuesto en [12], combinada con la Búsqueda Tabú (TS) propuesta en el Anexo 1. Se buscó aprovechar las características de intensificación y diversificación de estas dos Metaheurísticas con el fin de obtener una metaheurística hibrida que supere los resultados obtenidos en [12-13] y conseguir los objetivos de este trabajo.

[5] Mapa de Contacto: Matriz cuadrada NxN, donde N es la cantidad de AA en la Secuencia. Cada fila de esta matriz corresponde a un AA de la secuencia, y cada columna forma uno de dos posibles estados (0, 1, 3 ó 4), dependiendo si los Carbonos alfa se interceptan. Dicha intercepción está dada por la distancia euclidiana calculada por las posiciones3D de cada C-α, El rango de contacto definido para este caso es [4Á, 7Á].

3.4.1 Selección de la Técnica de Modelado

Se consideran las técnicas resumidas en el estado del arte para la identificación de modelos AC [10, 25, 36-39], sin embargo estas técnicas no se adaptan de forma directa para conseguir el objetivo de minería de datos que se busca.

En [12] con el framework CAIF-PFT se realiza un AG que identifica modelos de AC a partir de configuraciones globales obtenidas mediante simulación por dinámica molecular de plegamiento de proteínas con una precisión alrededor del 93% y deja abierta la investigación hacia nuevas técnicas que mejoren los resultados obtenidos o se exploren nuevas características hacia la solución del problema. En anexo 1, se realizó una Búsqueda tabú (TS), en aras de superar los resultados obtenidos en AG, realizando diferentes experimentos con TS con parámetros similares a los de AG como son el volumen y el manejo de datos, representación de la solución y condiciones de parada, manejo de estratos. Se obtienen resultados cercanos a los obtenidos en AG, pero TS no logra superarlos.

Dado que TS usa parámetros similares de AG, se explora la posibilidad de realizar una híbridación con estas dos técnicas, con el fin de obtener precisiones superiores a las obtenidas en el AG.

3.4.2 Metaheurística Hibrida entre Algoritmo Genético y Búsqueda Tabú

En esta sección se describen las técnicas híbridas que alcanzan una precisión superior del 90% en la identificación de modelos AC en trayectorias de simulación de plegamiento de proteínas.

La híbridación se considera desde dos puntos de vista, el primero se denomina *Hibridación por intensificación* que consiste en ejecutar el AG hasta un determinado número de iteraciones, se escogen los mejores individuos de esta ejecución y el espacio de búsqueda en donde se hallaron dichos individuos; estos individuos serán tomados como entradas en el TS intensificando sobre su espacio de búsqueda, con la generación de un vecindario en amplitud, tal como se indica en Anexo 1.

La segunda se denomina *Hibridación por Mejora de Hijos* puede ser visto como un Algoritmo Memetico (MA) [50], en donde se aprovechan las características de AG y TS para potencializar las búsquedas. El híbrido mejora un porcentaje de los hijos de una población generación a generación a través de TS, sobre los mejores individuos de la generación, se aplica TS con vecindario en profundidad, garantizando así una mejor descendencia, posteriormente se realiza mutación, selección y cruce, repitiendo el proceso hasta que se cumpla la condición de parada.

Los modelos de AC están constituidos por estados, lattice, condición de frontera, vecindad y reglas, Se definen como entradas para las técnicas híbridas modelos parcialmente definidos con los siguientes elementos:

- Lattice: NxN (N es numero de AA de la proteína).
- Condición de Frontera: Finita.
- Estados: (0, 1), 0 para no contacto y 1 para contacto.

Las técnicas híbridas completaran modelos parcialmente definidos, buscando vecindades para extraer las reglas de AC que complementaran los modelos de AC, por lo cual se define una vecindad de Moore como la máxima vecindad correspondiente al espacio de búsqueda definido, esta vecindad es de 49 celdas, como muestra la Figura 11.

V1	V2	V3	V4	V5	V6	V7
V8	V9	V10	V11	V12	V13	V14
V15	V16	V17	V18	V19	V20	V21
V22	V23	V24	V25	V26	V27	V28
V29	V30	V31	V32	V33	V34	V35
V36	V37	V38	V39	V40	V41	V42
V43	V44	V45	V46	V47	V48	V49

Figura 11. Vecindad de Moore (Adaptado de [12])

Las características que a continuación se describen pertenecen al proceso de AG.

3.4.2.1 Descripción de los Individuos:

Dado que los híbridos tienen características similares a AG, se representaran los individuos por fenotipo y genotipo como se plantea en [12].

El genotipo representa la vecindad de un individuo, la cual es una cadena binaria de longitud 49, en donde se denomina gen a cada posición de la cadena, las cuales pueden tomar un valor de 1 ó 0, indicando la inclusión o no inclusión de la celda en la vecindad. Cada posición de la cadena le corresponde una celda en la vecindad Moore de la Figura 11.

El fenotipo para un individuo se determina a partir de la posición de los genes en el genotipo cuyo alelo tome el valor binario de 1, según la posición que ocupa en la cadena correspondiente Para el caso de estudio, el fenotipo representará la vecindad.

La Figura 12 da un ejemplo, en el cual se muestra la correspondencia entre genotipo y fenotipo para el caso de que el individuo sea uno con la vecindad de Moore

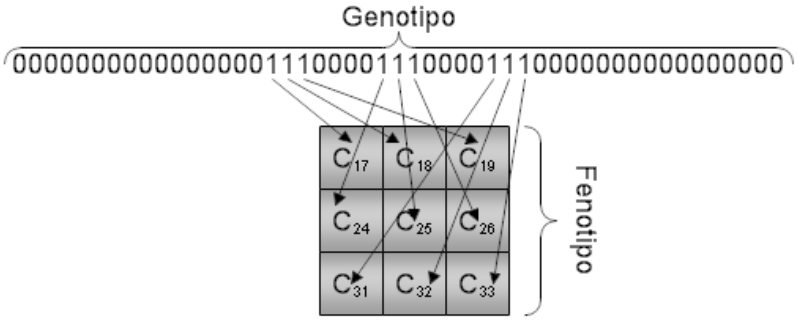


Figura 12. Representación de genotipo y fenotipo (tomado de [12]).

3.4.2.2 Función de Evaluación:

Los individuos del híbrido serán evaluados de la misma forma que en [12], en donde se usa el Coeficiente de Correlación de Matthews (MCC), la cual es una medida que no se ve afectada por sesgos de clase. MCC toma valores entre -1 y 1, donde los valores más cercanos a 1 indican buenas predicciones, es decir los verdaderos positivos (TP) y los verdaderos negativos (TN) son clasificados correctamente. En la Figura 13 se observa la ecuación del cálculo para MCC. (FP y FN indica los falsos negativos y positivos respectivamente.)

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{((TP + FN)(TP + FP)(TN + FN)(TN + FP))}}$$

Figura 13. Ecuación para el cálculo MCC.

3.4.2.3 Operadores Genético de Selección, Mutación y Cruce

Selección: Indica los mejores individuos a escoger para la siguiente generación. Se utilizará una selección por torneo como la usada en [12], la aplicación tiene asociado un parámetro de probabilidad que escoger al ganador, otorgando a los mejores individuos la posibilidad de ser seleccionados para el proceso de reproducción. Además se usa una estrategia elitista para conservar el mejor individuo de una generación.

Cruce: Éste operador determina cómo se genera un nuevo individuo a partir de dos individuos de la generación actual, en la aplicación el operador de cruce será a dos puntos y se tiene como parámetro la probabilidad de que el operador sea aplicado a los individuos que sean seleccionados para generar nuevos individuos [12].

Mutación: Éste operador determina cambios en los individuos resultantes del cruce, en donde se cambia uno o dos genes del individuo. La probabilidad de mutación de este operador indica la frecuencia con la que los genes de un individuo son cambiados [12].

A continuación se describen las características necesarias de TS para las técnicas híbridas, tomadas de Anexo 1.

3.4.2.4 Movimiento Tabú

Se denomina movimiento tabú al cambio de un gen del mejor individuo encontrado hasta el momento, dirigiendo la búsqueda a encontrar mejores individuos, en donde se almacena la posición de dicho movimiento.

3.4.2.5 Lista Tabú

Es una lista donde se guardan estructuras individuos o movimientos a vetar evitando que se repitan y se caiga en ciclos dentro de la búsqueda, se debe tener cuidado al seleccionar el tamaño de la lista tabú, ya que un tamaño pequeño no cumpliría el objetivo principal de la lista tabú que es la de evitar los ciclos dentro de la búsqueda y si es grande crearía podas que puedan dejar por fuera buenas soluciones.

Se definieron dos tipos de lista tabú

- Lista tabú estática con un tamaño comprendido entre el 10% y el 20% del tamaño del individuo, para el presente trabajo se escogió como tamaño de la lista tabú estática de 7, el cual es la raíz cuadrada del tamaño de los individuos, que durante la experimentación de TS lanzo los mejores resultados.

- Lista tabú dinámica inicialmente se empieza con un tamaño del 10% del tamaño del individuo, luego en cada iteración se incrementa el tamaño de la lista en uno hasta que la lista dinámica tenga un tamaño igual al 20% del tamaño del individuo ó cuando en el proceso de la búsqueda tabú se encuentre una solución mejor a la solución que se tiene hasta ese momento, en este caso la lista dinámica se volverá a asignar el valor del 10% del tamaño de la solución. La lista dinámica se incrementara máximo hasta alcanzar el tamaño del 20% del tamaño de la solución.

3.4.2.6 Criterio de Aspiración

Sirve para sacar un elemento de la lista tabú a partir de un criterio establecido en el proceso de TS, para la aplicación se estableció como criterio de aspiración los individuos de la lista tabú que al ser evaluados superen el valor de la función de evaluación del mejor individuo encontrado hasta el momento, aquellos elementos que cumplen con esta

condición serán sacados de lista tabú o aquellos que cumplan un número máximo de iteraciones en la lista, este número de iteraciones esta dado por el tamaño de la lista tabú.

3.4.2.7 Diversificación

Es el proceso por el cual TS trata de escapar de óptimos locales dándole prioridad a soluciones poco exploradas, en la aplicación si la función de evaluación actual no es mejor a la global, se penaliza la función de evaluación en 10 iteraciones, penalizando aquellos movimientos tabú que se han repetido mucho durante el proceso.

3.4.2.8 Intensificación

Es el proceso en el cual se intensifica o se explota la búsqueda en espacios que son muy prometedores, en anexo1, terminada una iteración ILAS se verifican los individuos con los mejores valores de la función de evaluación y se intensifica en el estrato en el cual se encontró dicho individuo, si al terminar la intensificación se encuentra un individuo con una valor en la función del evaluación mejor al individuo global se hace el cambio y se sigue explorando mas caminos a partir de dicha individuo, en caso contrario el proceso sigue su curso normal.

3.4.2.9 Criterio de parada

Determina el momento en el cual ya no continuará ejecutando el híbrido, como condición de parada se han definido los siguientes criterios:

- Máximo número de iteraciones para el proyecto de 200 iteraciones.
- Máximo número de iteraciones sin que la solución global sea cambiada en caso del TS máximo de 40 iteraciones
- Máximo valor en la función de evaluación sea 1.0

3.4.2.10 Vecindario

Un vecindario o entorno es un conjunto de soluciones vecinas que se generan a partir del mejor individuo de cada iteración, realizando transformaciones locales (movimientos Tabú) sobre el individuo (Anexo 1).

En las técnicas hibridas se aplican 2 tipos de vecindarios:

- **Vecindario en Amplitud**: Dada una solución actual (cadena binaria de 49 bits) se realiza un cambio de cada bit de la solución actual, explorando todas las posibles combinaciones de esta, en donde si la celda contiene 1 se cambia por 0, o si es 0

se cambia por 1, esto quiere decir que el vecindario estaría compuesto por 49 posibles soluciones.

La Figura 19 muestra la generación de un vecindario a partir de una solución actual (en este caso la marcada en color rojo), en donde una nueva solución se genera por el cambio de un bit (bit de color diferente) y la línea punteada gruesa representa todo vecindario de la solución actual. El paso siguiente es escoger la mejor solución de vecindario para este ejemplo está representada por la línea punteada roja, a partir de la cual se genera el nuevo vecindario. la línea roja indica que se puede regresar a la solución anterior, pero en el concepto de TS ese movimiento es tabú, lo que evitaría regresar a esa solución anterior ya explorada y de esta forma prevenir los ciclos en la búsqueda.

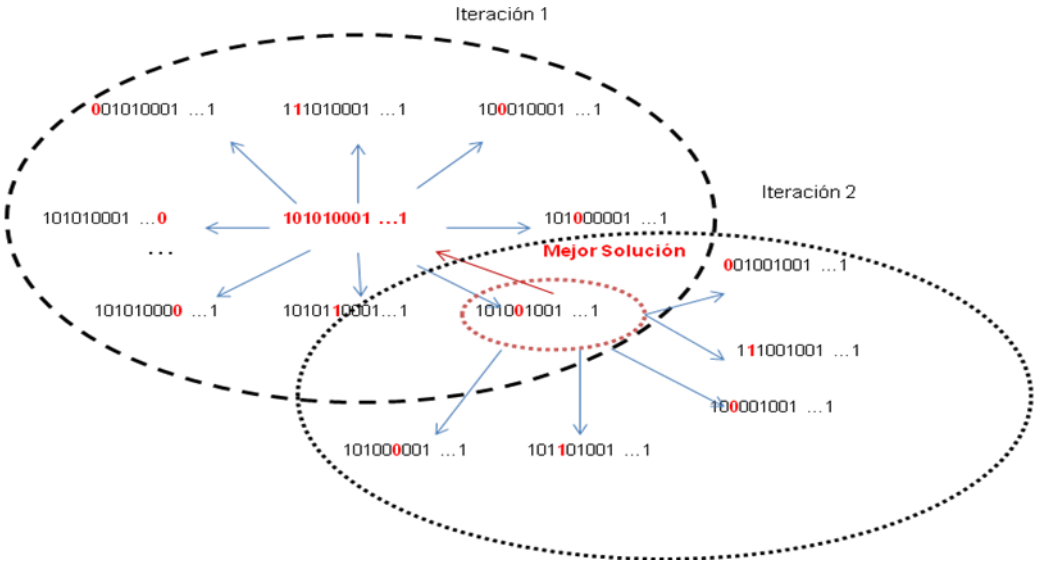


Figura 14. Vecindario en amplitud TS (Tomado de Anexo 1).

- **Vecindario en Profundidad**: Se define un número N de genes a cambiar del mejor individuo, se realizan un cambio aleatorio de genes dependiendo del tamaño N en donde si el gen es 1 se cambia por 0 y viceversa. Esto quiere decir que si el numero N es 5 el vecindario estaría compuesto por 5 nuevos individuos. Además se define un número de rondas R, que indica el número de veces que se repite este proceso, evitando aquellos movimientos ya escogidos.

  La Figura 15 muestra un ejemplo del vecindario en amplitud en donde se realizaran 5 rondas con 3 individuos a evaluar, el proceso inicia generando el vecindario del individuo escogido, se evalúa el vecindario y se escoge el mejor vecino (círculo rojo), esto me indica una ronda y se repite este proceso hasta llegar al máximo número de rondas a realizar.

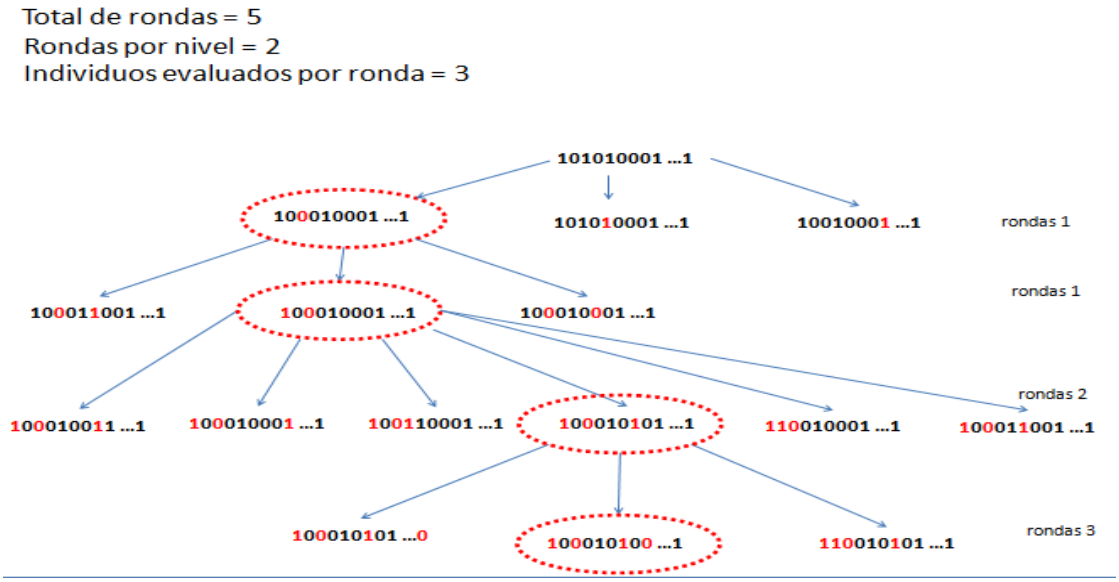


Figura 15.Vecindario en Amplitud de TS (Tomado de Anexo 1).

3.4.3 Diseño de Prueba del Hibrido.

Las características de diseño de prueba usados en [12] , serán aplicadas de forma similar para las técnicas híbridas. Se dividirán las evidencias en 10 conjuntos de 4.000 pasos de simulación por dinámica molecular (Una trayectoria tiene 40.000 pasos de simulación), donde un conjunto de datos será usado para la ejecución del Híbrido, como muestra la Figura 16.

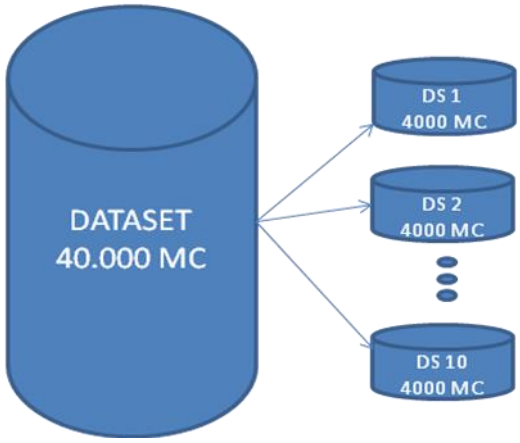


Figura 16.Ejemplo de conjunto de datos.

Se usara el enfoque de aprendizaje iterativo con estratos alternantes (ILAS), que evita un sobre entrenamiento de los individuos del híbrido al tener que enfrentarse a menor cantidad de datos y de esta forma reducir el costo computacional para cada generación del híbrido. Con el enfoque ILAS se divide el conjunto de datos en *S* estratos de igual tamaño, es decir que para un conjunto de datos (4.000 mapas de contacto), si el estrato tiene un tamaño de 200 mapas de contacto se obtendrían 20 estratos, en donde el último estrato de datos le sigue al primero como muestra la Figura 17.

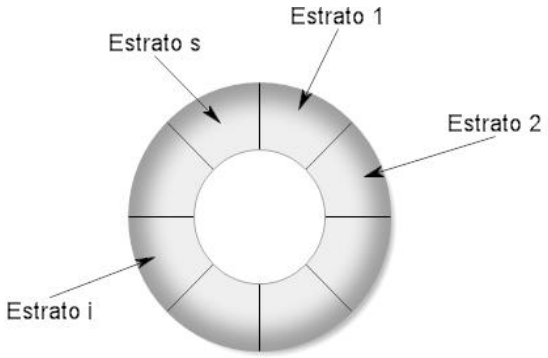


Figura 17.Ejemplo de ILAS (Tomado de [12]).

Para verificar el desempeño de los modelos obtenidos para cada versión del algoritmo híbrido, se usara un grafico ROC [47], el cual se caracteriza por no verse afectado por sesgo de clase, la Figura 18 muestra la representación de la tasa de falsos positivos (TFP)[6] de un predictor versus la tasa de verdaderos positivos (TVP)[7], la utilización del Grafico ROC, es explicado en [12].

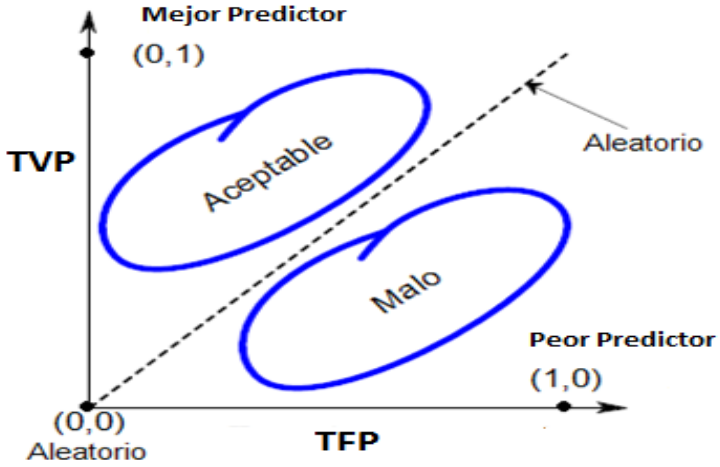


Figura 18.Grafico ROC (Tomado de [12]).

[6] Tasa de Falsos Positivos: ésta dada por FP/(FP+VN); donde FP son los Falsos Positivos y VN los verdaderos Negativos, del predictor.

[7] Tasa de Verdaderos Positivos: ésta dada por VP/(VP+FN); donde VP los Verdaderos Positivos y FN son los Falsos Negativos, del predictor.

Adicional se generara un reporte de las medidas sensitividad o VFP, especificidad o TFP y precisión[8], para verificar la calidad de los modelos obtenidos por el híbrido y se mostrara el fenotipo, genotipo y las reglas del modelo AC, que se destaque del híbrido.

3.4.4 Construcción del Modelo

En esta sección se describen los parámetros correspondientes a las técnicas híbridas y los modelos obtenidos en cada una de ellas.

3.4.4.1 Construcción del Modelo - Hibridación por intensificación

Los parámetros a usar para AG son los usados en [12] y de igual forma los usados en el TS con vecindario en Amplitud de Anexo 1, que serán descritos a continuación:

Los parámetros similares para AG y TS son:

Longitud del Cromosoma: 49.

- Alelos: (0,1)
- Función de Evaluación (Fitness): Coeficiente de Correlación de Matthews – Penalización por cantidad de reglas.
- Tamaño del estrato ILAS: 200 pasos de simulación.

Parámetros AG:

Los parámetros usados para el AG fueron usados en [12], en donde se escogió una selección por torneo de tamaño 8, un operador de cruce de dos a dos puntos con una probabilidad de cruce 0.9 y se tiene una mutación con una probabilidad de 2% que es aproximadamente el inverso del tamaño del individuo(49 bits). Estos los parámetros ajustados para obtener mejores resultado de modelos AC presentados en [12].

Los parámetros son los siguiente:

- Operador de Selección: Torneo.
- Tamaño del Torneo: 8
- Tasa de Cruce: 0.9
- Operador de Cruce: Dos puntos.
- Elitismo: 5 % de los mejores individuos.
- Operador de Mutación: Inversión de bit.
- Tasa de mutación: 2%.
- Tamaño de la Población: 75

[8] Precisión: esta dada por (VP + VN) / (P + N), en donde VP son los verdaderos Positivos, VN los verdaderos Negativos, P las clases positivas y N las clases Negativas.

Parámetros TS

- Tamaño de Vecindario: 49.
- Generador de Vecindario: Por amplitud (Generado a partir de mejor individuo AG por iteración ILAS).
- Lista Tabú: 7
- Número de Iteraciones: 200.
- Diversificación Inicial: 20 (número máximo de iteraciones sin cambios en la solución global).
- Diversificación final: 7 (número máximo de iteraciones en el cual se aplica el criterio de diversificación)
- Condición de parada forzosa: 40

En la Figura 19 se observa el comportamiento del AG para 60 generaciones, que corresponde a 3 iteraciones ILAS, los individuos escogidos de cada una de la iteraciones ILAS son los obtenidos en las generaciones 13, 33 y 53; que son aquellos individuos que obtuvieron una función de evaluación alta y serán intensificados con TS.

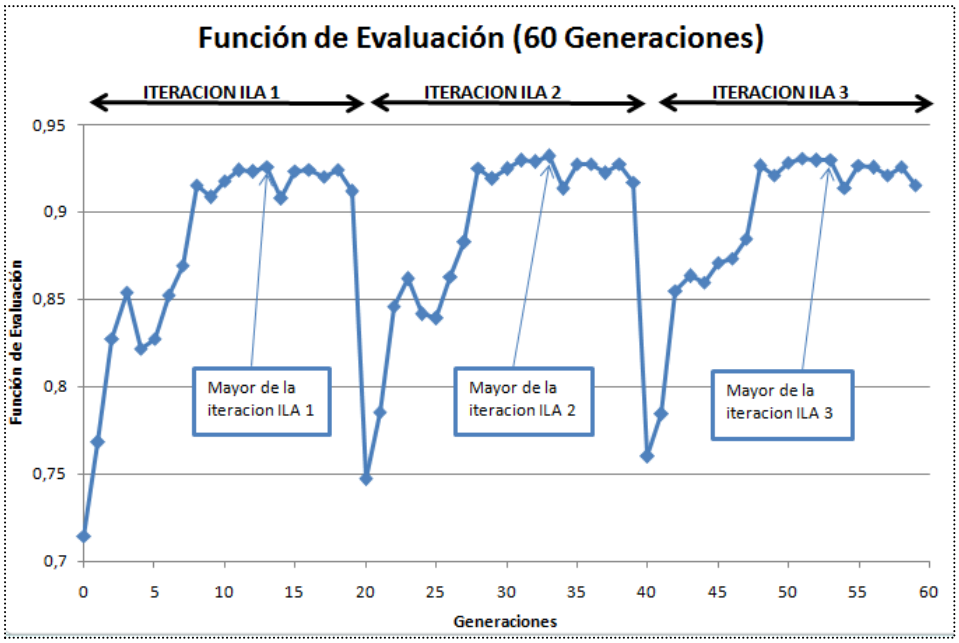


Figura 19.Ejemplo de ejecución del AG

Adicional a los individuos a intensificar, se obtiene el estrato en donde se encontraron estos, el cual será el espacio de búsqueda para TS, que a diferencia de AG que utiliza un conjunto de datos de 4.000 pasos de simulación (1 iteración ILAS), TS solo usara los 200 pasos de simulación en donde se encontraron los individuos. Las figuras 20, 21 y 22,

muestra los gráficos de dispersión de resultados de función de evaluación Global y Actual del comportamiento de TS para los individuos 13, 33 y 53 respectivamente.

En la gráfica de la Figura 20 se puede observar que el TS sube rápidamente en la función de evaluación actual, además se alcanza un criterio de aspiración, esto quiere decir que un movimiento que era tabú (prohibido) alcanzo un valor más alto en la función de evaluación actual con respecto a la global y esta última fue reemplazada, obteniendo mejores modelos. El mejor individuo es obtenido en la iteración 26, para este estrato.

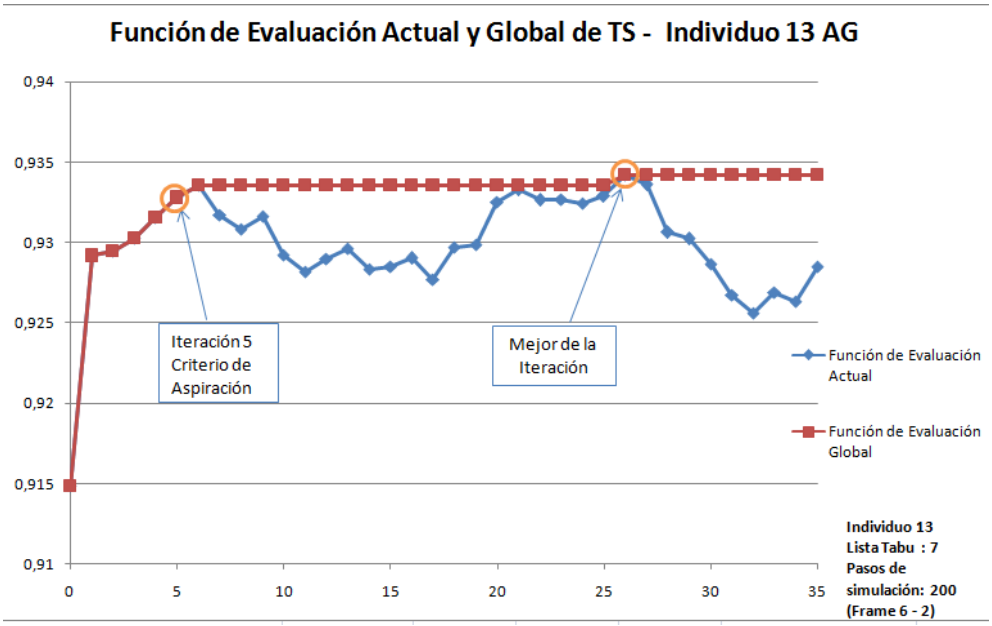


Figura 20.Ejemplo Ejecución del TS Individuo 13.

Al igual que el comportamiento del individuo 13 el gráfico de comportamiento de TS de la

Figura 21, para individuo 33 converge con mucha rapidez en las primeras iteraciones, se alcanza un criterio de aspiración en la iteración 3, y se obtiene el mejor individuo para el estrato en la iteración 7. El comportamiento de la función global indica que no se alcanzó un mejor individuo después de la iteración 7, sin embargo los individuos con la función de evaluación actual no se descartan por completo, debido a que alcanzan un tamaño de vecindad menor con respecto al individuo de la iteración 7.

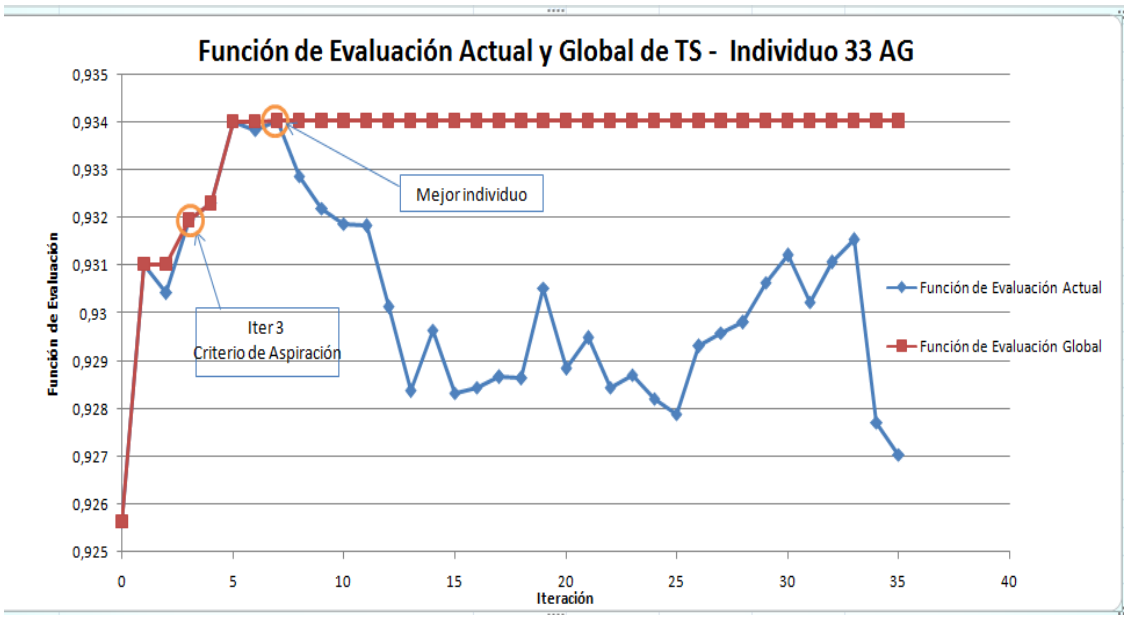
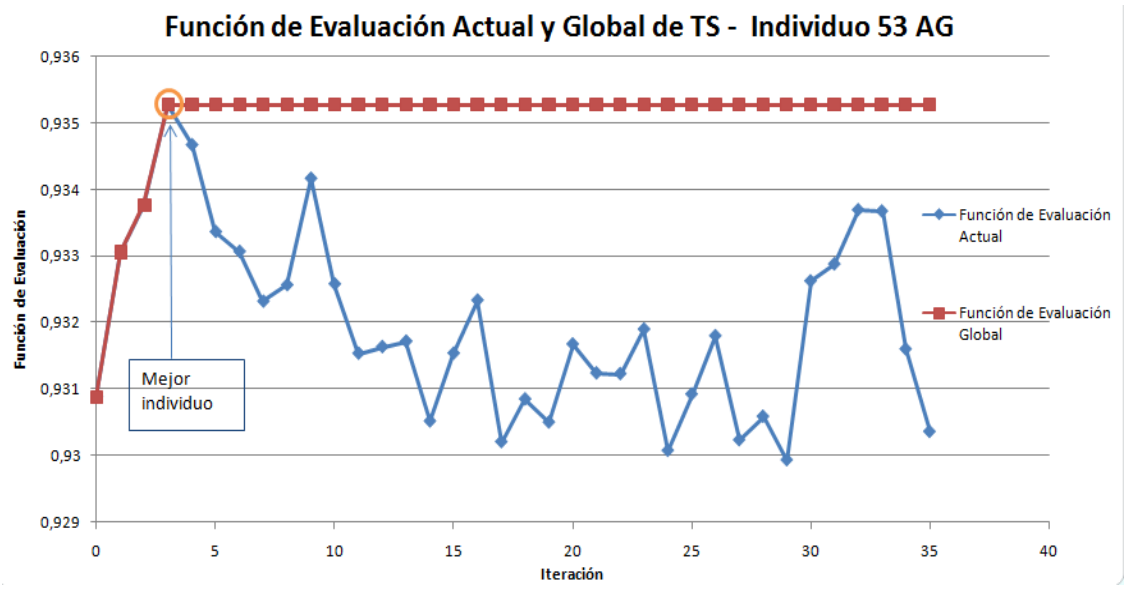


Figura 21.Ejemplo Ejecución del TS Individuo 33

El comportamiento de la gráfica para el individuo 53 intensificado con TS de la Figura 22, converge más rápido que los anteriores y no aplica un criterio de aspiración, alcanzando el mejor individuo para ese estrato en la iteración 3. Se observa además que la función de evaluación global se mantiene constante después de la iteración 3, pero al igual que el grafico del individuo 33, no se descartan los individuos intensificados encontrados después de la iteración 3, ya que algunos de estos individuos tienen un tamaño de



vecindad menor con respecto al mejor individuo.

Figura 22.Ejemplo Ejecución del TS Individuo 53.

Este tipo de comportamiento en los individuos 33 y 53 nos indica que a medida que pasan las generaciones en AG, los individuos son mejores generación a generación, por lo cual TS alcanza el óptimo más cercano en las primeras iteraciones pero se pueden encontrar soluciones con vecindarios pequeños que los hacen interésateme para su análisis.

3.4.4.2 Construcción del Modelo - Hibridación por Mejora de Hijos

Los parámetros a usar para la hibridación AG-TS son los siguientes:

- Longitud del Cromosoma: 49
- Alelos: (0,1)
- Función de Evaluación (Fitness): Coeficiente de Correlación de Matthews – Penalización por cantidad de reglas.
- Tamaño de la Población: 75
- Hijos a Mejorar: 8 (10% de la población, lo mejores de la evaluación).
- Generador de Vecindario: Por Profundidad (Generado a partir de mejores individuos de la Evaluación)
- Máximo Evaluaciones de TS: 30
- Lista Tabú: 4
- Operador de Selección: Torneo.
- Tamaño del Torneo: 8
- Tasa de Cruce: 0.9
- Operador de Cruce: Dos puntos.
- Elitismo: 5 % de los mejores individuos.
- Operador de Mutación: Inversión de bit.
- Tasa de mutación: 2%..
- Tamaño del estrato ILAS: 200 pasos de simulación.
- Numero de Iteraciones:200

En la Figura 23 se observa el gráfico de comportamiento de la técnica Híbrida AG-TS para 60 generaciones, que corresponde a 3 iteraciones ILAS, el comportamiento del híbrido es similar al AG de la figura 14, debido a que se maneja el mismo conjunto de datos, pero adicional a esto el comportamiento oscilatorio observado se debe a que los valores de la función de evaluación en cada generación utilizan estratos diferentes y cada 20 generaciones se repite el mismo estrato. Este comportamiento es similar al visto en la construcción de modelos de [12], en la gráfica de ejemplo de ejecución del AG (Figura 36).

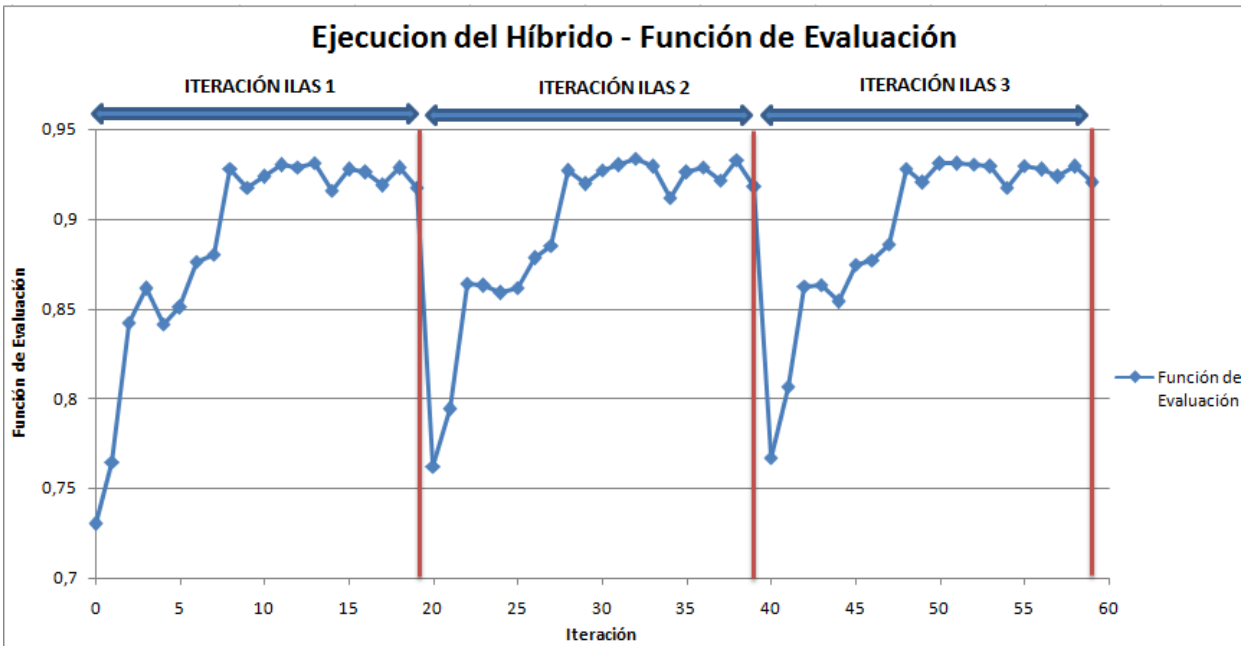


Figura 23.Ejemplo Ejecución del Híbrido AG-TS.

El híbrido supera el comportamiento del AG como se muestra en el gráfico de la Figura 24, se escogen los individuos de la generación 13, 33 y 53 para comparar los individuos obtenidos con el AG y con la técnica híbrida por intensificación.

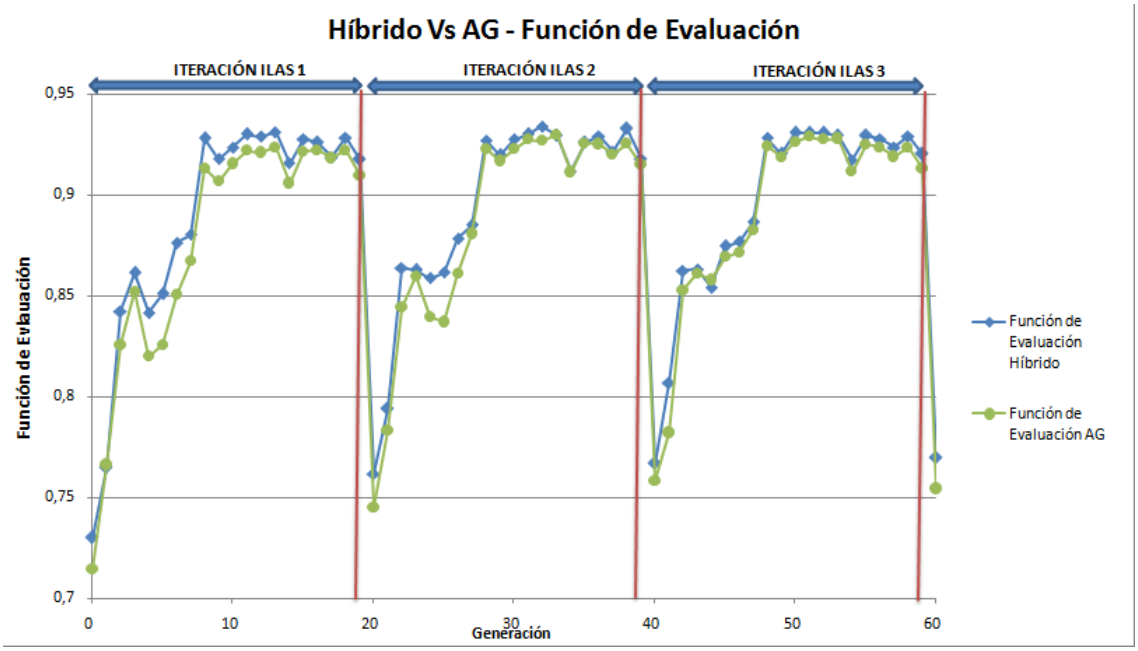


Figura 24.Ejecución del Híbrido AG-TS Versus AG.

En la grafica de la Figura 25 se analizan los resultados entre el número de evaluaciones realizadas con respecto a la función objetivo (MCC). Se analizan los puntos en los cuales las evaluaciones son múltiplos de 75 y 315 que son el numero de evaluaciones generadas por iteración en el AG y en el hibrido por mejora de hijos respectivamente.

En esta grafica se observa como el híbrido supera al AG durante más de 19.000 evaluaciones mostrando un mejor comportamiento que el AG en la función de evaluación.

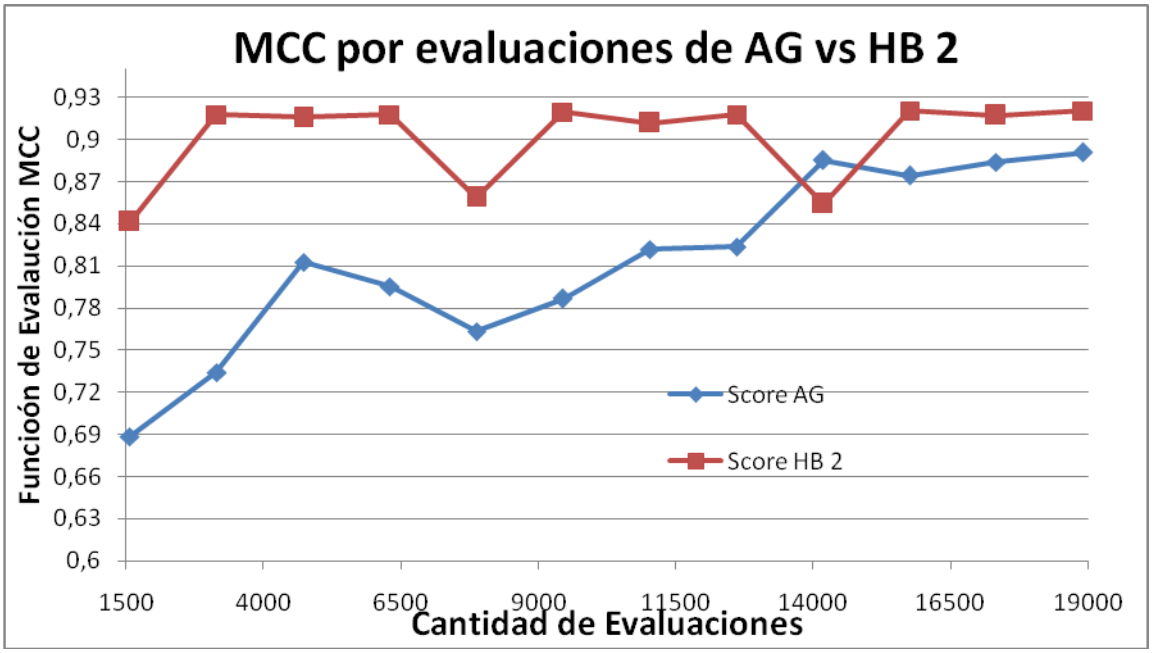


Figura 25.Cantidad de evaluaciones Vs Función Objetivo AG Vs HB2.

La tabla 3 muestra la cantidad de evaluaciones, la iteración en donde se alcanzaron el número de evaluaciones y el score de las técnicas.

Tabla 3. Cantidad de evaluaciones AG y HB 2.

Evaluaciones	Iteraciones AG (75)	Score AG	Iteraciones HB2(315)	Score HB 2
1575	21	0,68858799	5	0,84158781
3150	42	0,73397861	10	0,91775887
4725	63	0,81269055	15	0,91568549
6300	84	0,79525118	20	0,91772883
7875	105	0,76337557	25	0,85871325
9450	126	0,78677062	30	0,91955411
11025	147	0,82175381	35	0,91165068
12600	168	0,82363822	40	0,91783874
14175	189	0,88512806	45	0,85399626
15750	210	0,87453701	50	0,92038649
17325	231	0,88394498	55	0,91731067
18900	252	0,89056404	60	0,92040376

3.4.4.3 Análisis de los Modelos Obtenidos con las Técnicas Híbridas

Para el conjunto de datos 1, que contiene los mapas de contactos correspondientes a los primeros 4.000 pasos de simulación, se destacan los individuos 13, 33 y 53 del AG, como se menciono anteriormente. A continuación en las tablas 4, 5 y 6 se muestra la precisión, especificidad y sensitividad de la trayectoria simulada con el AG para estos Individuos, adicionalmente se mostraran los tamaños de la vecindad de los modelo de AC.

Tabla 4. Medidas de la Evaluación de los individuos AG

Individuo	Precisión	especificidad	Sensitividad	Tamaños de la vecindad
13 AG	0.923284898	0.97446942	0.6075894	**23**
33 AG	0.930964285	0.977215069	0.6458495	**19**
53 AG	0.93561265	0.98317582	0.6424076	**20**

Al intensificar los individuos 13, 33 y 53 con TS se obtienen los siguientes resultados (tabla 5).

Tabla 5. Medidas de la Evaluación del Híbrido por intensificación.

Individuo	Precisión	especificidad	Sensitividad	Tamaños de la vecindad
13 INT	0.93165551	0.977828910	0.64701779	**17**
33 INT	0.92344210	0.96044070	0.69521174	**18**
53 INT	0.91956938	0.95808946	0.6992165	**17**

Se puede observar en la tabla 4 y 5 que el individuo 13 INT es mejor que el 13 AG, sin embargo los individuos 33 INT y 53 INT son superados por los individuos 33 AG y 53 AG, pero se observa que los tamaños de los vecindarios del hibrido son menores que los obtenidos por AG, lo que hace a estos individuos interesantes para analizar, debido a que menor cantidad de vecinos menor es el numero de reglas a interpretar y eso hace que sea un poco más sencillo explicar el fenómeno.

Tabla 6. Medidas de la Evaluación del Híbrido AG-TS por Mejora de Hijos

Individuo	Precisión	especificidad	Sensitividad	Tamaños de la vecindad
13 HB	0.93628204	0.98615608	0.62883156	**20**
33 HB	0.93144469	0.98623174	0.59397077	**22**
53 HB	0.93775306	0.9807989	0.67239488	**20**

La tabla 6 muestra mejores resultados en las medidas de precisión para la técnica híbrida por mejora de hijos, superando los resultados del AG y del híbrido por intensificación (tabla 4 y 5). Y los tamaños de vecindarios de esta técnica son menores a los obtenidos en el AG.

Los modelos candidatos para representar la dinámica de toda la sección de la trayectoria (4000 pasos de simulación - conjunto de datos 1) es el individuo 53 en ambos híbridos.

La vecindad del modelo 53 INT hallado con la técnica híbrida por intensificación se presenta en la Figura 26 en donde se muestra una vecindad de 17 celdas distribuidas, la representación genotipo y el fenotipo para este individuo.

a) 00000010010000010100100001010001110010011

b)

0	0	0	0	0	0	1
0	0	1	0	0	0	0
0	1	0	1	0	0	1
0	0	0	0	1	0	1
0	0	0	1	1	1	0
0	1	0	0	1	0	1
1	0	1	0	0	1	1

c)

V1	V2	V3	V4	V5	V6	V7
V8	V9	V10	V11	V12	V13	V14
V15	V16	V17	V18	V19	V20	V21
V22	V23	V24	V25	V26	V27	V28
V29	V30	V31	V32	V33	V34	V35
V36	V37	V38	V39	V40	V41	V42
V43	V44	V45	V46	V47	V48	V49

Figura 26.Mejor individuo Hibrido por intensificación - conjunto de datos 1 a) Genotipo. b) Fenotipo. c) Fenotipo.

La vecindad del modelo 53 HB hallado con la técnica híbrida por mejora de hijos, se presenta en la Figura 27 en donde se muestra una vecindad de 20 celdas distribuidas junto con la representación de su genotipo y el fenotipo para este individuo.

a) 00001011010000001010101001010011010010011010100 11

b)

0	0	0	0	1	0	1
1	0	1	0	0	0	0
0	0	1	0	1	0	1
0	1	0	0	1	0	1
0	0	1	1	0	1	0
0	1	0	0	1	1	0
1	0	1	0	0	1	1

c)

V1	V2	V3	V4	V5	V6	V7
V8	V9	V10	V11	V12	V13	V14
V15	V16	V17	V18	V19	V20	V21
V22	V23	V24	V25	V26	V27	V28
V29	V30	V31	V32	V33	V34	V35
V36	V37	V38	V39	V40	V41	V42
V43	V44	V45	V46	V47	V48	V49

Figura 27.Mejor individuo Hibrido por mejora de Hijos - conjunto de datos 1 a) Genotipo. b) Fenotipo. c) Fenotipo.

Estos individuos encontrados sobre el conjunto de datos 1, ofrecen precisiones mayores al 90%, además al ser evaluados sobre otros conjuntos de datos se obtienen precisiones superiores al 96% como muestra las tablas 7 y 8.

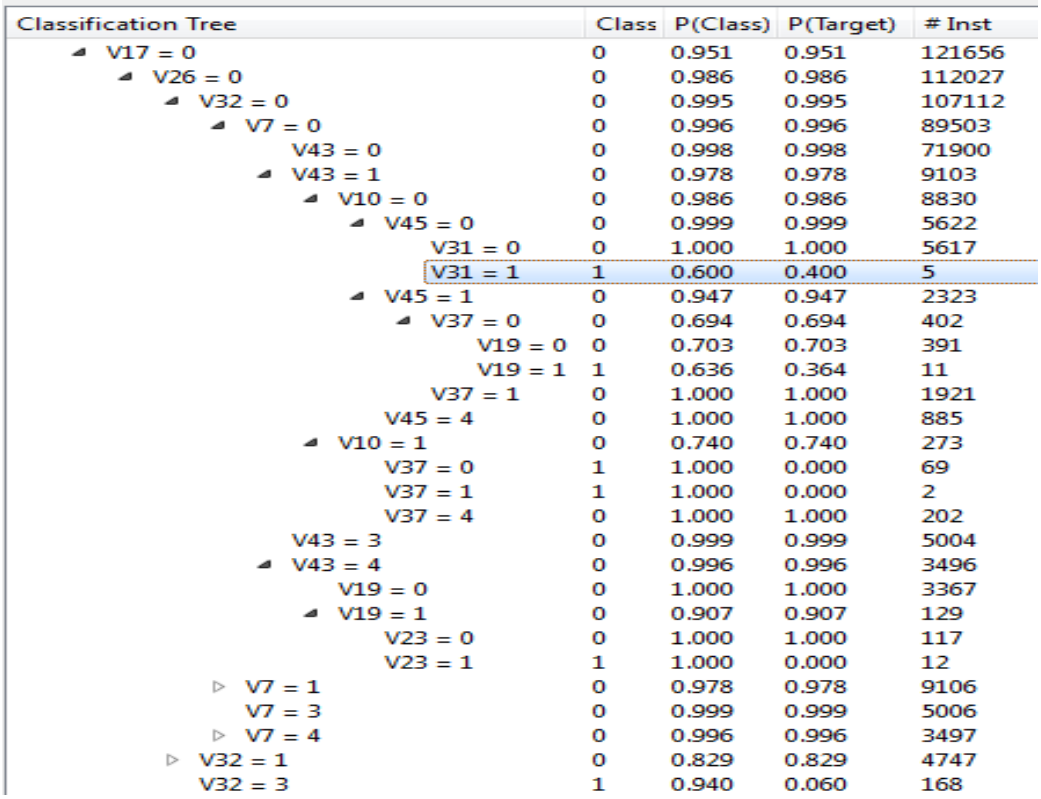
Tabla 7. Medidas de la Evaluación del Híbrido AG-TS por Hijos sobre diferentes conjuntos de datos

DATASET	TPR o TVP	FPR 0 TFP	MCC	PRECISION
Conjunto de Datos 2	0,94007995	0,01210694	0,92384374	0,980906939
Conjunto de Datos 4	0,93260655	0,01521718	0,90878439	0,977263878
Conjunto de Datos 7	0,93396932	0,01257309	0,91811145	0,979722449

Tabla 8. Medidas de la Evaluación del Híbrido por Intensificación sobre diferentes conjuntos de datos

DATASET	TPR o TVP	FPR 0 TFP	MCC	PRECISION
Conjunto de Datos 2	0,91082667	0,03486386	0,83797512	0,957200816
Conjunto de Datos 4	0,90667618	0,04136887	0,81661356	0,951144082
Conjunto de Datos 7	0,94839154	0,02436836	0,8908069	0,971705714

La Figura 28 muestra el árbol de decisión de las reglas obtenidas para el individuo 53 de la hibridación por mejora de hijos que se destaca sobre las otras técnicas, este árbol es generado con Orage Data Mining[44]. En la parte superior de la figura se muestra las etiquetas class, P (Class), P (Target) y #Inst; donde class son lo objetos en este caso 0, 1, 3 y 4, P (class) es la probabilidad de clase mayor, es decir la clase predicha correcta para un nodo, P (Target) es la clase destino positiva, #Int es numero de instancias para una clase en un nodo.



Classification Tree	Class	P(Class)	P(Target)	# Inst
V17 = 0	0	0.951	0.951	121656
V26 = 0	0	0.986	0.986	112027
V32 = 0	0	0.995	0.995	107112
V7 = 0	0	0.996	0.996	89503
V43 = 0	0	0.998	0.998	71900
V43 = 1	0	0.978	0.978	9103
V10 = 0	0	0.986	0.986	8830
V45 = 0	0	0.999	0.999	5622
V31 = 0	0	1.000	1.000	5617
V31 = 1	1	0.600	0.400	5
V45 = 1	0	0.947	0.947	2323
V37 = 0	0	0.694	0.694	402
V19 = 0	0	0.703	0.703	391
V19 = 1	1	0.636	0.364	11
V37 = 1	0	1.000	1.000	1921
V45 = 4	0	1.000	1.000	885
V10 = 1	0	0.740	0.740	273
V37 = 0	1	1.000	0.000	69
V37 = 1	1	1.000	0.000	2
V37 = 4	0	1.000	1.000	202
V43 = 3	0	0.999	0.999	5004
V43 = 4	0	0.996	0.996	3496
V19 = 0	0	1.000	1.000	3367
V19 = 1	0	0.907	0.907	129
V23 = 0	0	1.000	1.000	117
V23 = 1	1	1.000	0.000	12
V7 = 1	0	0.978	0.978	9106
V7 = 3	0	0.999	0.999	5006
V7 = 4	0	0.996	0.996	3497
V32 = 1	0	0.829	0.829	4747
V32 = 3	1	0.940	0.060	168

Figura 28. Árbol de decisión mejor modelo.

En la Figura 29, se observa la reglas compuesta por V17, V25, V43, V7 Y V31 con sus estados en 0 (cero), y su siguiente estado en el paso de simulación es 0, esto nos indica que los aminoácidos más cercanos a la diagonal tiene contacto entre ellos. En la figura se observa como la regla presentada no marca la diagonal principal, indicando la regla de solo contacto de 0.

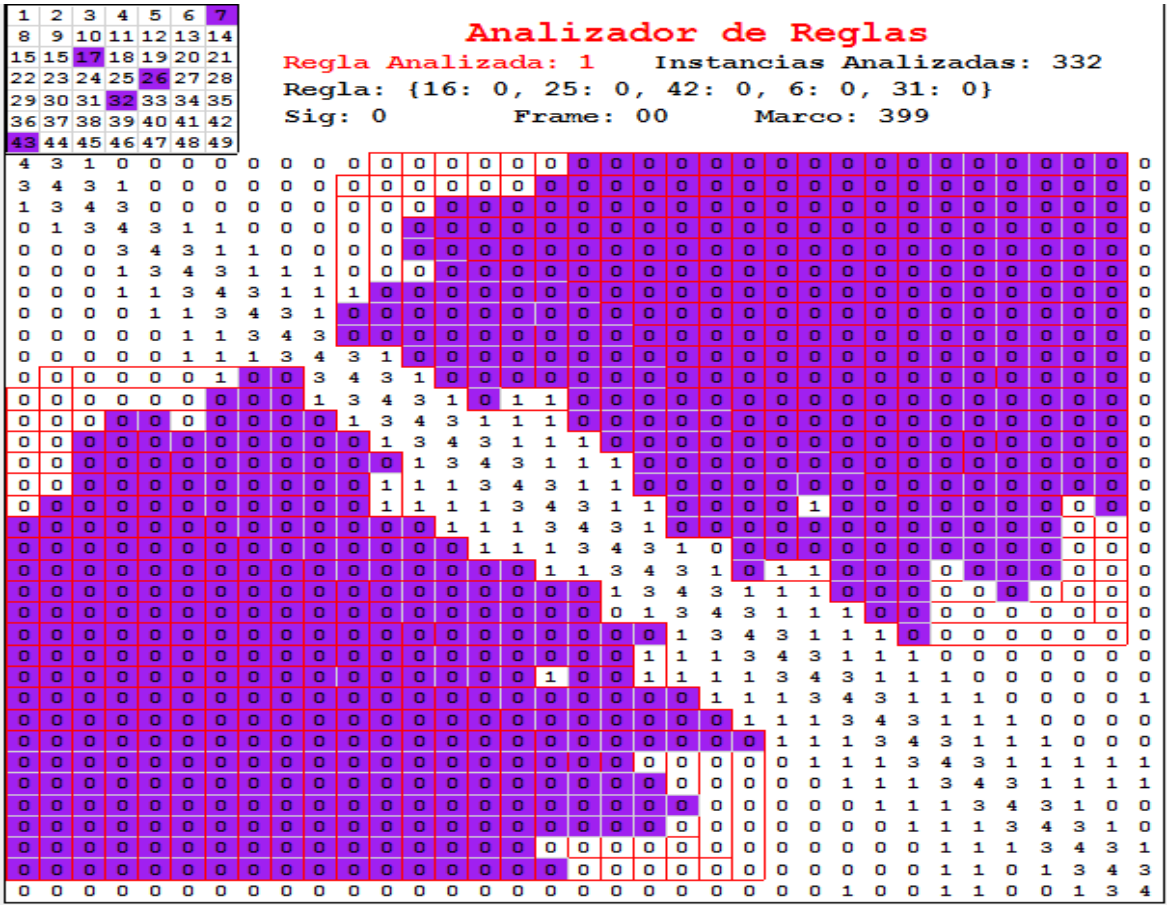


Figura 29. Validador regla del 0 (cero).

En la Figura 320 se observa la presentación de la regla V17, V41,V43 Y V26 con sus estados en 1 (uno) y su siguiente estado en el paso de simulación de 1, esta regla representa una de las características importantes del plegamiento de proteínas que son las hélices, la figura indica que la existencia de 2 hélices. En la Figura 31 se observa la proteína con la 2 hélices, tal como representa la regla analizada de la figura 30.

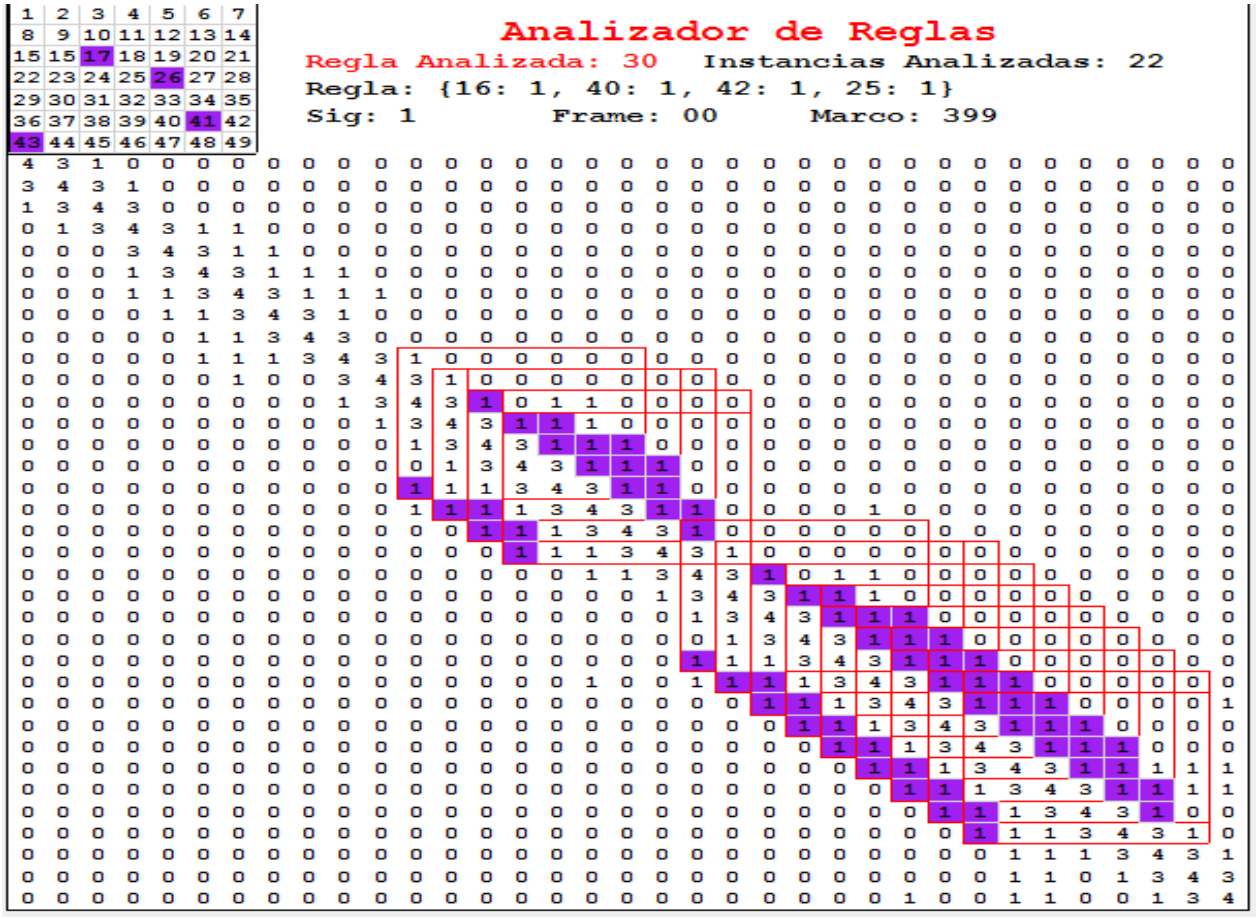


Figura 30. Validador regla del 1 (Hélices).

.

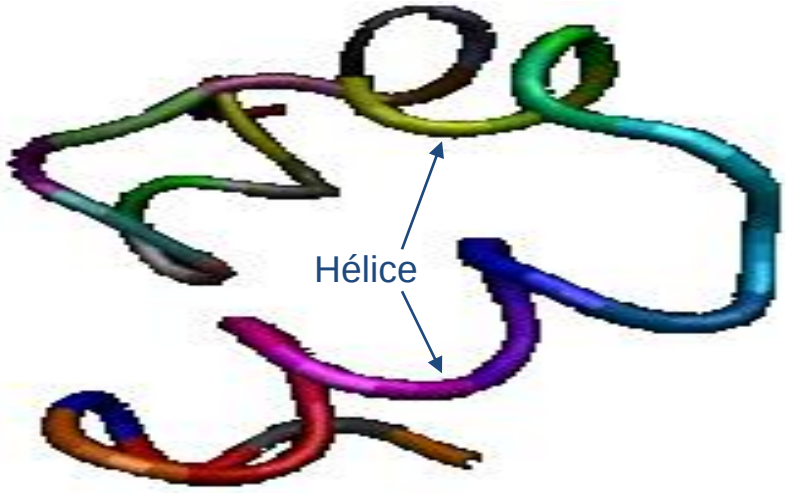


Figura 31. Proteína con dos Hélices.

En la Figura 32 se observa la representación de la regla V17, V26, V43 Y V20 en 0(cero) con V32 en 1 (uno), y su siguiente estado en el paso de simulación de 0, se muestra otra de las características importantes del plegamiento de las proteínas que son los giros, la figura representa la existencia de un giro bien definido entre las dos hélices de la proteína que se forman en este marco de simulación.

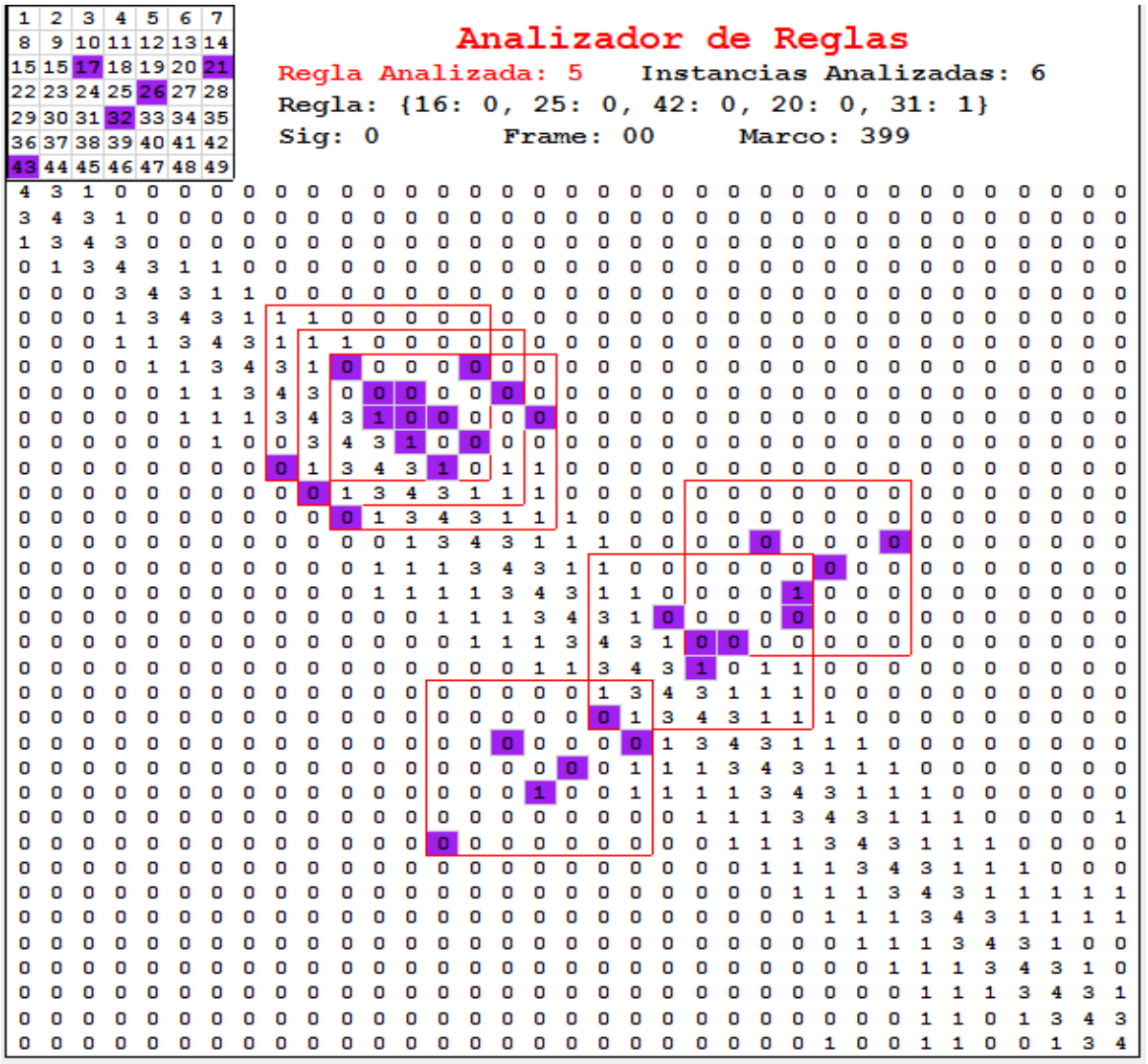
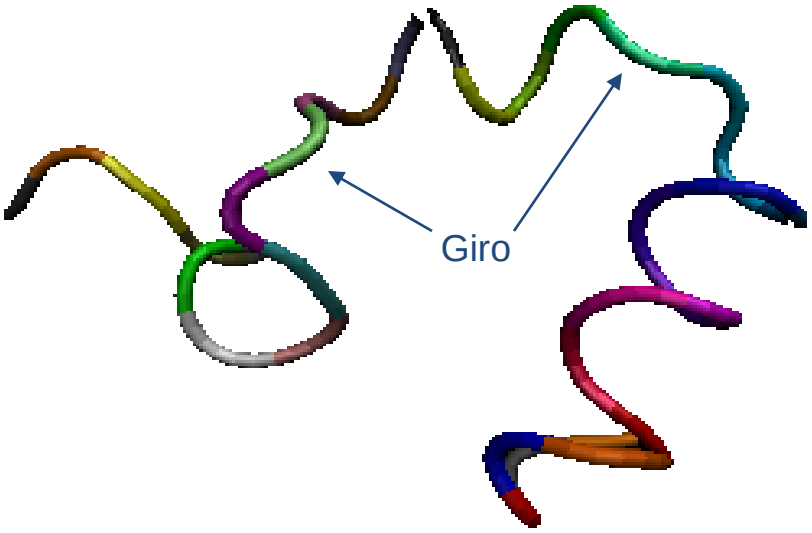


Figura 32. Validador regla del 0 y 1 (Giros).

La figura 3 muestra los cambios de dirección o giros de la proteína, que representa la regla analizada en la figura 32.



--

Figura 33. Giros de la Proteína.

3.5 FASE 5: EVALUACIÓN

3.5.1 Evaluación de Resultados

Los híbridos implementados son capaces de identificar modelos de AC en trayectorias de plegamiento de proteínas, con una precisión superior al 93%, para los primeros 4.000 pasos de simulación, que es un segmento de la trayectoria, que es el más pesado debido a que apenas se está plegando la proteína, las diferentes evaluaciones sobre otros individuos y diferentes conjuntos de datos, serán mostrados a en la capítulo 5 de este libro.

Los resultados permiten concluir que el objetivo de la minería de datos ha sido logrado por los Híbridos propuestos dado que las medidas para la evaluación de los resultados son buenas. (Tabla 7 y 8). Sin embargo se debe explorar soluciones sobre otros conjuntos de datos, ya que no fue posible realizar experimentos sobre conjuntos de datos diferentes al 1, debido al tiempo de ejecución de cada una de las técnicas (AG, TS, Hibridación por intensificación y mejora de hijos), adicionalmente la poca disponibilidad de equipos limito el tiempo de experimentación.

3.5.2 Revisión del Proceso

En la ejecución de la técnica híbrida por intensificación, se hace necesario verificar los parámetros de TS para obtener mejores resultados, debido a que la experimentación sobre el mismo estrato no arrojo buenos resultados, es por eso que se decide realizar un

cambio en los parámetros, en donde se encontraron resultados buenos en las primera iteraciones de TS.

En la técnica híbrida por mejora de hijos, se hizo necesario cambiar el enfoque de la parte distribuida del proceso de evaluación, por lo cual se decide tomar el enfoque de vecindario por profundidad para TS, que evalúa una menor cantidad de hijos a diferencia del vecindario en amplitud que evalúa todas las combinaciones de un vecino. El cambio consistió en realizar las evaluaciones de los 75 individuos de AG de forma distribuida y en TS solo 6 individuos de forma distribuida por cada iteración TS.

Se realizaron cambios en el porcentaje de hijos a evaluar, ya que a mayor cantidad de hijos a evaluar, mayor es la cantidad de cómputo, por lo cual se decidió dejar solo el 10% de los individuos a evaluar por TS, ya que al experimentar los resultados obtenidos eran mejores cuando se intensificaba entre 10% y el 20% la población de AG, sin embargo se escogió el 10% ya que la intensificar con un porcentaje mayor de la población resultaba muy costoso en tiempo de computo debido a la cantidad de individuos a evaluar.

A pesar que la técnica híbrida por mejora de hijos obtiene buenos resultados, es necesario realizar más ejecuciones ya que esta técnica requiere mucho tiempo en cómputo para cada generación.

3.5.3 Determinar Pasos a Seguir

Dado que en el proceso de evaluación se identificaron patrones en las vecindades generadas por las técnicas, se observó que hay vecinos que podrían ser fijos y que generan buenas evaluaciones, por lo cual se pude experimentar nuevas técnicas que generen vecindades por prioridad de vecinos, en aras de encontrar mejores modelos a los obtenidos en este trabajo.

En el anexo 2 se presentan los modelos AC evaluados con representación de la vecindad y medidas de calidad, además se realizó un experimento que consistía en cambiar algún bit de la diagonal (si era 0 a 1 y viceversa) verificando como afectaba la evaluación de los modelos AC, esto nos dio paso a definir un vecindario en prioridad que será explicado en el capítulo 5 de este libro.

3.6 FASE 6: DESPLIEGUE

3.6.1 Reporte Final

Entendimiento del Negocio

Los modelos obtenidos cumplen con los objetivos del negocio propuestos, pero se hace necesario mas experimentación para encontrar modelos aun mejores a los encontrados en el proceso.

Proceso de Minería

El framework caif-pft es una herramienta útil para la limpieza, construcción y formateo de datos de la fase de preparación de datos y facilitar el proceso de este.

La arquitectura que brindo el framework caif-pft fue de gran ayuda para la definición de las técnicas híbridas en la fase de modelado.

Resultados de Minería de Datos

Los resultados obtenidos en el proceso de minería de datos justifican los objetivos propuestos en este trabajo al obtener modelos de AC con buena precisión, además se encontraron en las vecindades, patrones que lleven a modificar solo algunos vecinos para encontrar modelos de AC con mejores precisiones.

Despliegue de Resultados

Los resultados del proceso de minería de datos serán desplegados mediante una herramienta software apoyada en el framework CAIF – PFT, que implementa dos metaheurísticas híbridas que permite identificar modelos de AC para trayectorias de plegamiento de proteínas.

Las reglas de los modelos AC obtenida por las técnicas serán presentadas con una herramienta grafica que permite visualizar en un mapa de contacto el comportamiento de los aminoácidos de la proteína en un marco de simulación.

4 ALGORITMO HÍBRIDO AG-TS

Se implementaron dos técnicas híbridas entre una metaheurística poblacional (AG) y una metaheurística de trayectoria (TS), soportadas por la arquitectura del framework CAIF-PFT [12], las técnicas buscan identificar las vecindades más adecuadas para extraer reglas de evolución de un AC a partir de información proporcionada por una trayectoria de plegamiento de proteínas obtenida mediante dinámica molecular.

El diseño de las características de la hibridación de AG y TS fueron descritas en el capítulo 3 de este libro en la sección de modelado de la técnica, en donde se presentan los detalles del proceso de adaptación del framework para la implementación de las técnicas híbridas.

4.1 DESCRIPCIÓN DE LAS TÉCNICAS HÍBRIDAS SOPORTADA POR EL FRAMEWOK CAIF-PFT

Hibridación por Intensificación:

La hibridación por intensificación, consiste en ejecutar por separado un Algoritmo Genético (AG) un determinado número de generaciones, al terminar la ejecución de éste, se escogen aquellos individuos que tienen los mejores valores en su función de evaluación y el estrato en donde fueron encontrados dichos individuos.

El AG usado para esta hibridación, están descritas en el capítulo 3 de este libro, en donde se describen las características del diseño de AG como individuos, fenotipo, genotipo, selección, mutación, cruce y función de evaluación. La Figura 34 muestra el diagrama de flujo que representa la implementación de AG en forma general.

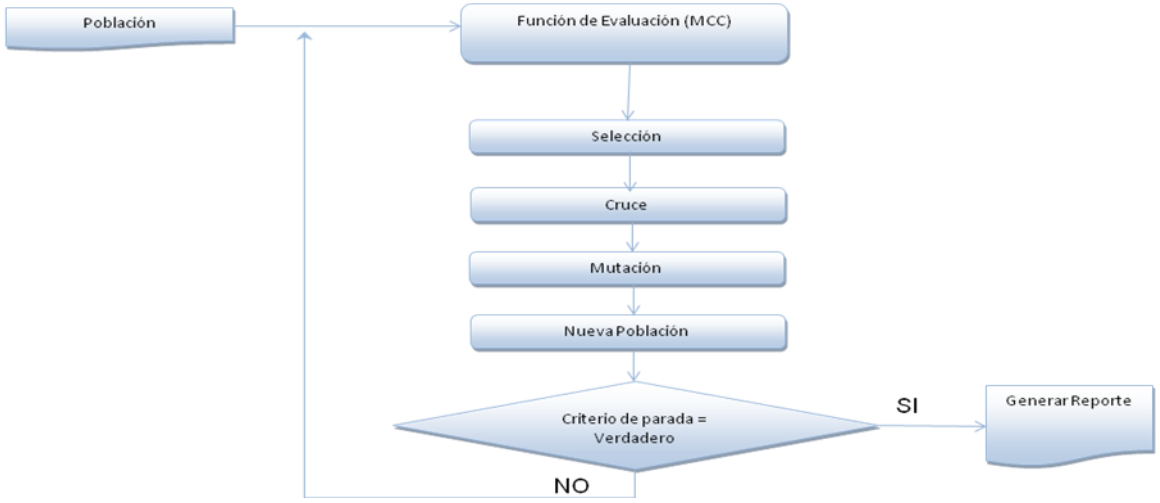


Figura 34.Esquema conceptual: Algoritmo Genético.

Los mejores individuos encontrados se les aplicara búsqueda tabú (TS), con generación de vecindario en amplitud, con el fin de explotar el espacio de búsqueda (Estrato de cada individuo), en que fueron encontrados y de esta forma mejorar los individuos encontrados en AG.

De igual forma que AG, las características de diseño de TS están descritas en el capítulo 3 de este libro, se describe el diseño del vecindario en amplitud, lista tabú, criterios de aspiración y otros conceptos de diseño de TS. La Figura 35 muestra un diagrama de flujo de TS.

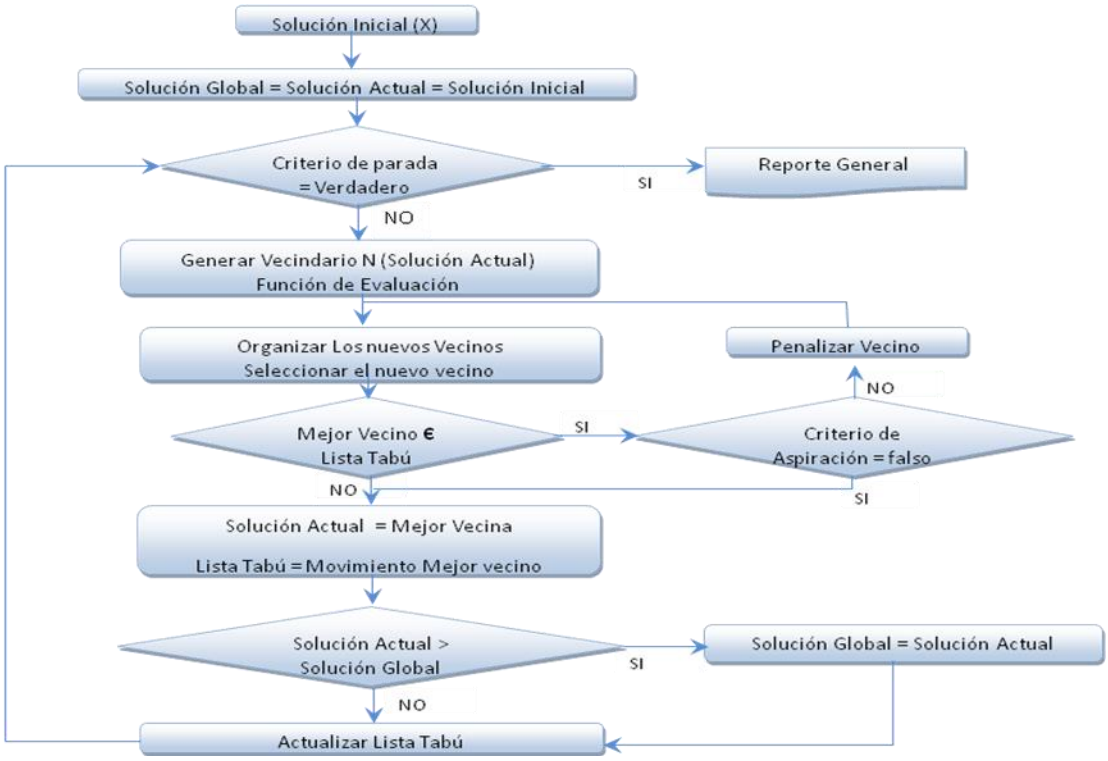


Figura 35.Esquema conceptual: Búsqueda Tabú.

La Figura 36 Muestra un esquema conceptual de la hibridación por intensificación en donde se tiene una población inicial la cual puede ser generada aleatoriamente o preestablecida, se le realiza el proceso de AG, seguido se escogen los mejores individuos de una iteración ILAS del AG y se le aplica TS mejorando los individuos.

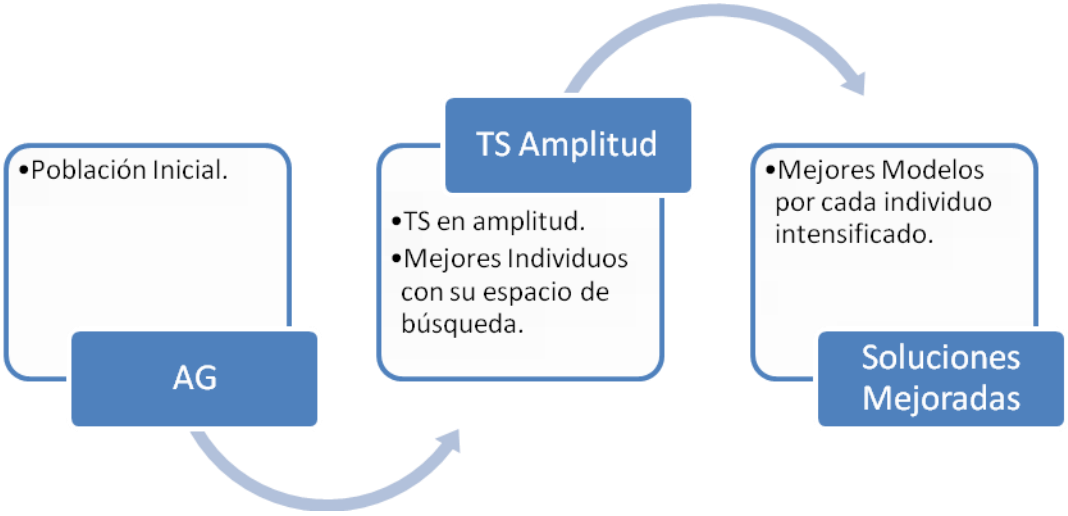


Figura 36.Esquema conceptual: Hibridación por Intensificación.

Hibridación por mejora de Hijos

La Hibridación por mejora de hijos consiste en mejorar un porcentaje de descendencia de AG en cada generación con la ayuda de TS, en primer lugar se genera la población de AG y se realiza el proceso de evaluación de los individuo; después se escogen aquellos individuos que obtuvieron un valor alto en su función de evaluación en la generación, en este caso del 10% del tamaño de la población como se menciono en el capítulo 3.

Con los mejores individuos escogidos se procede a realizar TS con vecindario en profundidad en el estrato de la generación, explotando la búsqueda en ese estrato, de tal forma que se garantiza una mejor descendencia de AG. Seguido de este proceso se ejecuta selección, mutación y cruce, repitiendo el proceso hasta cumplir un criterio de parada, Figura 37 muestra un esquema conceptual general.

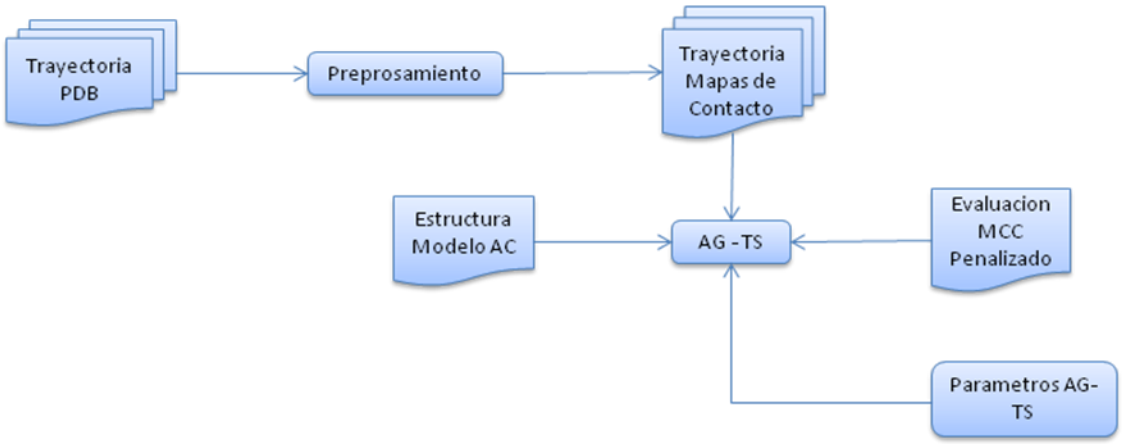


Figura 37.Esquema conceptual General: Hibridación Por mejora de Hijos.

La muestra Figura 38 el esquema conceptual detallado de la hibridación por mejora de hijos.

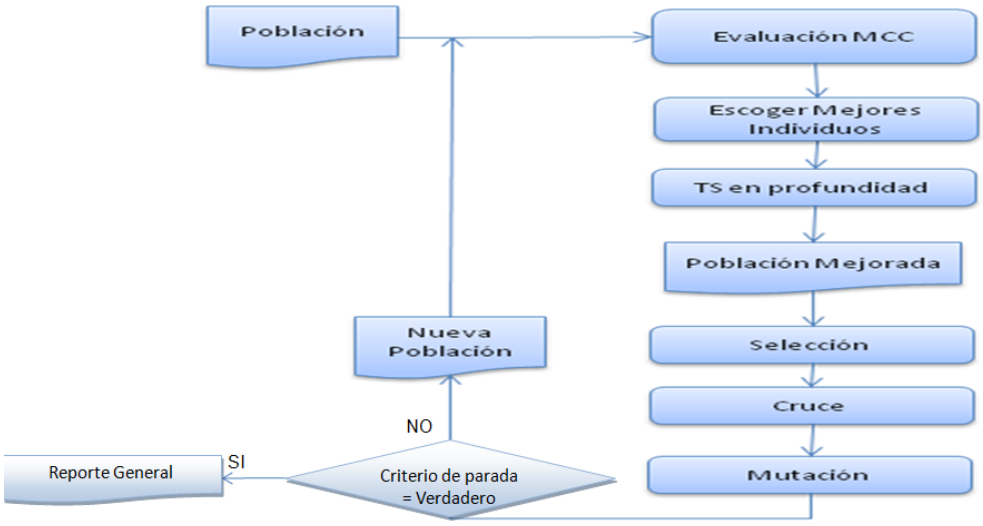


Figura 38.Esquema conceptual Detallado: Hibridación por mejora de hijos.

Para escoger que individuos de la población total son los ideales para mejorar con TS se realizo una prueba piloto del híbrido en donde se organiza la los individuos de la población de mayor a menor por su función de evaluación y se experimentó tomando un porcentaje de la población al inicio, al medio, al final y aleatoria, con el objetivo de observar que porcentaje de la población permitía un mejor comportamiento.

La Figura 39 muestra el comportamiento del híbrido con los porcentajes de población mencionados. Se observa que el mejor comportamiento es realizado por el hibrido con la población mejorada por TS al inicio (Línea con Cuadro), seguido por la mejora con TS de la población al final (Línea con rombo), la mejora de la población con TS para el medio y aleatorio no son una buena opción según lo observado. Adicional se presenta el comportamiento de AG (Línea con Circulo), en donde se puede observar que el hibrido tiene un mejor comportamiento.

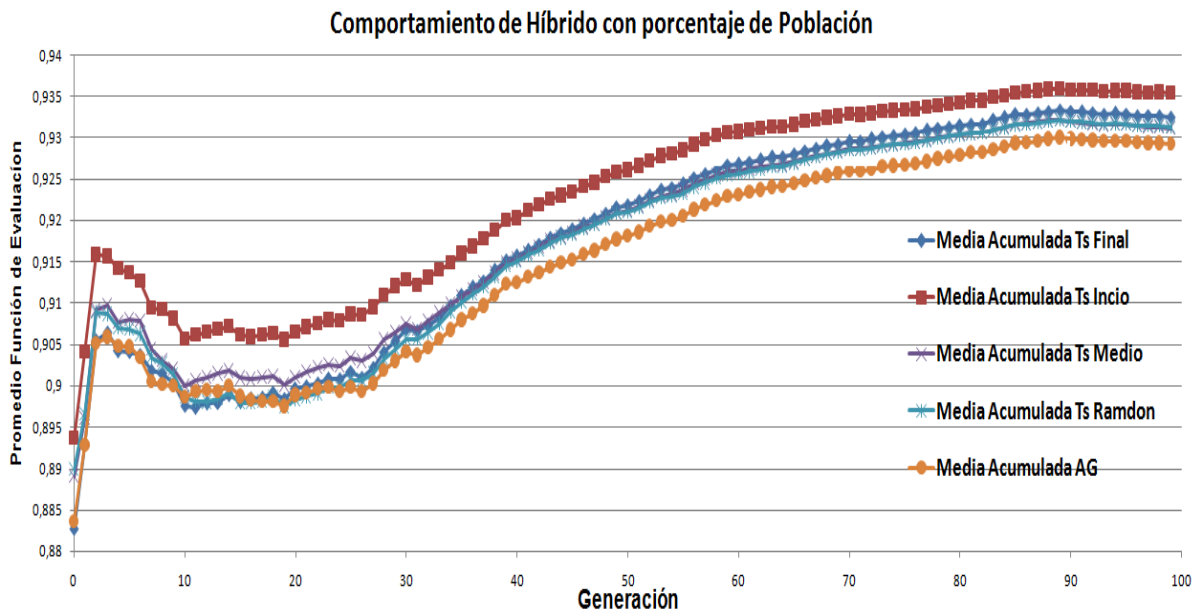


Figura 39.Comportamiento del híbrido por mejora de Hijos con porcentaje de población al inicio, al medio, al final, aleatorio y AG.

4.2 PLAN DE ITERACION

A continuación se presenta el plan de iteración que soporta las fases de Extreme Programming [51-52] para la implementación de la técnica Hibrida por mejora de hijos, las tablas 9, 10, 11 y 12 muestra el plan de iteración para la planificación, diseño, desarrollo, y pruebas.

Tabla 9. Fase de Planificación Híbrido por Mejora de Hijos.

Código de la Iteración	01HB
Fases a la que pertenece	Planificación
Fecha de Inicio	Mayo 22 de 2012
Fecha de Cierre	Mayo 25 de 2012
Comentarios	Ninguno
Descripción de iteración	• Modelo conceptual (Metáfora). • Definir historias de usuario. • Batería de pruebas. ○ Pruebas Unitarias. • Desarrollo de la Técnica. • Puntos de Hibridación. • Pruebas de modelos.

Tabla 10. Fase de Diseño Híbrido por Mejora de Hijos.

Código de la Iteración	02HB
Fases a la que pertenece	Diseño
Fecha de Inicio	Mayo 26 de 2012
Fecha de Cierre	Junio 3 de 2012
Comentarios	Se debe tener en cuenta los parámetros usados en la técnica AG planteada en anexo 1 y la técnica Tabu Search (TS), planteada en el Anexo 2, en donde se encuentran los parámetros de diseño para esta fase.

Descripción de iteración	• Definir los modelos de AC (Parcialmente definidos)
	  ○ Malla (2D)
	  ○ Frontera (finita): los datos fuera de la frontera tendrán un valor fijo igual a cero (0)
	  ○ Estados (0, 1, 3, 4)
	• Definir Función Objetivo.
	• Tamaño de los individuos.
	• Definir los parámetros de entrada de AG
	  ○ Espacio de Búsqueda (Evidencias).
	  ○ Número máximo de Iteraciones.
	  ○ Probabilidad de selección.
	  ○ Operador Genético de cruce.
	  ○ Tamaño de cruce
	  ○ Probabilidad de Mutación.
	  ○ Tamaño de la población.
	• Ordenar población.
	• Escoger los mejores Individuos de la población de la generación.
	• parámetros de ajuste de TS
	  ○ Mejores Individuos de la generación.
	  ○ Estrato para el Espacio de búsqueda Individuos.
	  ○ Tamaño de vecindario en profundidad.
	  ○ Número máximo de profundidad.
	  ○ Lista tabú.
	  ○ Tenencia tabú: Numero de iteraciones en las que una solución permanecerá en la lista tabú
	  ○ Función Objetivo.

Tabla 11. Fase de Desarrollo Híbrido por Mejora de Hijos.

Código de la Iteración	03HB
Fases a la que pertenece	Desarrollo
Fecha de Inicio	Junio 6 de 2012
Fecha de Cierre	Junio 15 de 2012
Comentarios	Se tiene en cuenta el proceso de diseño, los parámetros de entrada de AG y TS.

Descripción de iteración	• Implementar la estrategia
	• Parámetros de entrada
	  ○ Individuos
	  ○ Modelos AC
	  ○ Número máximo de iteraciones
	  ○ Evidencias (Mapas de Contacto)
	• Función de evaluación por MCC.
	• Generar vecindario en Profundidad.
	• Seleccionar el siguiente mejor individuo
	  ○ Evaluar los individuos y seleccionar el que tiene mejor evaluación.
	  ○ El movimiento tabú no debe pertenecer a la lista tabú o debe tener un criterio de aspiración para poder ser seleccionado.
	• Criterios de parada
	  ○ Cumplir número máximo de iteraciones.
	  ○ La media móvil es menor a 40 generaciones.
	  ○ Número máximo de iteraciones en la cual la mejor solución no cambia.
	  ○ La mejor solución alcanza o supera el criterio de evaluación de la función objetivo.
	• Generar reportes: Reporte de comportamiento del Híbrido.
	• Generar reporte de medidas: Terminado el proceso de ejecución de la técnica, escoger los mejores individuos según su función de evaluación y generar las medidas de precisión, MCC, tasa de falsos Positivos, tasas de Falsos Negativos, sensitividad y sensibilidad.

Tabla 12. Fase de Pruebas Híbrido por Mejora de Hijos.

Código de la Iteración	04INT
Fases a la que pertenece	Desarrollo – pruebas
Fecha de Inicio	Junio 17 de 2012
Fecha de Cierre	Junio 30 de 2012
Comentarios	Tener en cuenta el proceso de evaluación realizado, reportes generados por la técnica.

	• Pruebas de ejecución (Graficas de comportamiento).
	• Simular el modelo de AC
	  ○ Se realiza la simulación correspondiente a una solución
	  ○ Los modelos de AC ya están completamente definidos.
	• Generar Reportes
	  ○ Pruebas sobre los reportes Estadísticas de la ejecución del Hibrido,

4.3 DIAGRAMA DE CLASES.

El diagrama de clase que se utilizó en la implementación de los híbridos por intensificación y mejora de hijos se muestra en la Figura 40. en donde se observar la clase HBContactMapStrategy que es la clase que implementa las estrategias híbridas junto con la clase HBStrategyIteration. Las clases Evidences, CAmodel, HBStrategy, Evaluator, Parser, Simulator, son clases que proporciona el framework CAIF – PFT que dan soporte a la aplicación.

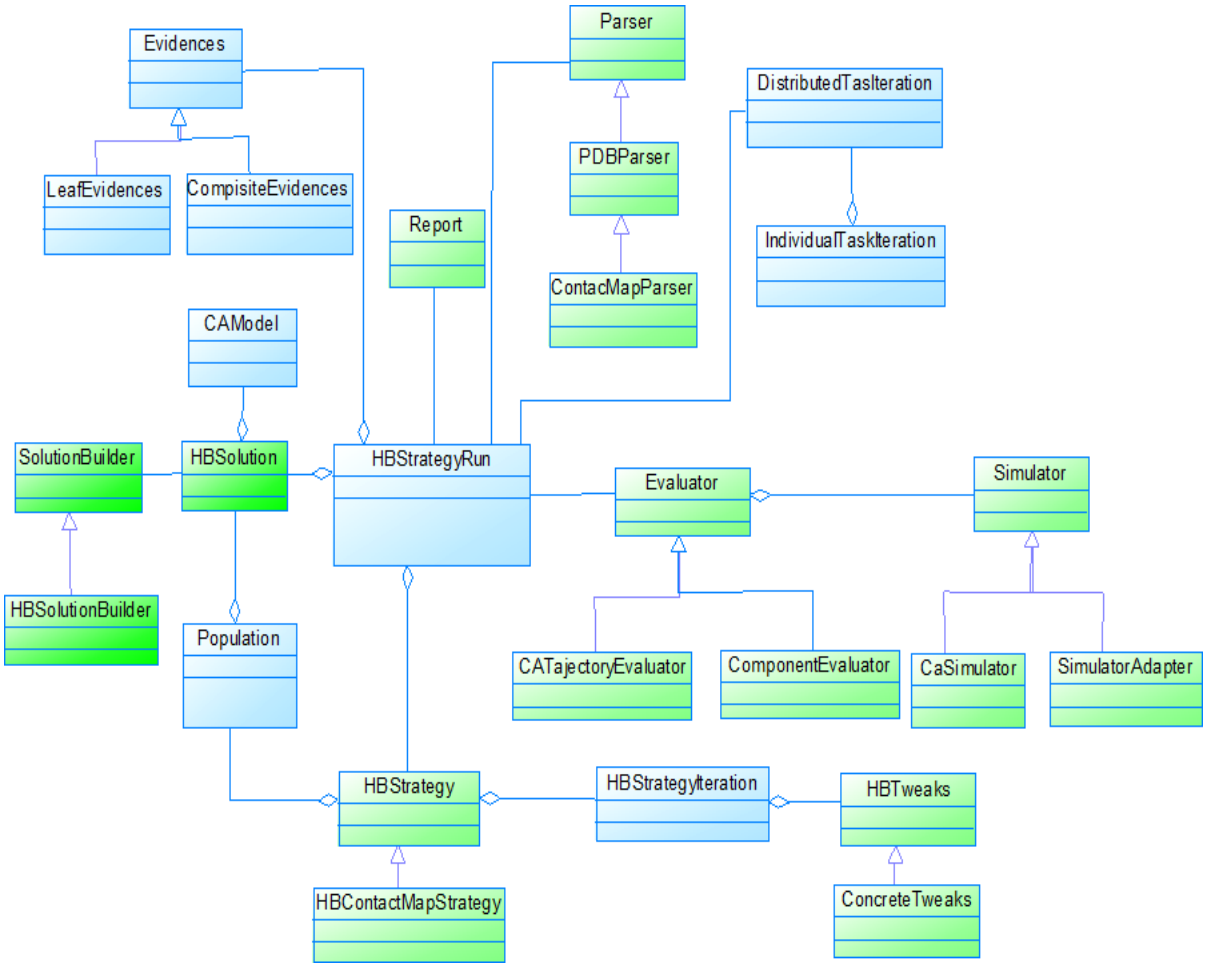


Figura 40.Diagrama de Clases.

4.4 HISTORIAS DE USUARIO DEL ALGORITMO HÍBRIDO.

Las historias de usuario que a continuación se presentan describen las funcionalidades más importantes de la técnica híbrida y los puntos de variación a tener en cuenta de la arquitectura del framework CAIF-PFT. las plantillas usadas son una adaptación de [53].

Tabla 13. Historia de Usuario Función de Evaluación [53].

NOMBRE HISTORIA DE USUARIO: Función de Evaluación individuos			
Fecha: 24/Mayo/2012	Tipo de Actividad: Implementación.	Nueva: X	Modificación:
Historia Numero: 01	Prioridad: Alta	Usuario:	Técnica: X

Descripción:	Se debe tener las trayectorias de mapa de contacto almacenados en disco (Evidencias), con permisos de lectura. El usuario proporcionara la ubicación de los mapas de contacto, si la ubicación es errada la aplicación deberá informar "error de ruta", si la ubicación es correcta se inicia la hibridación hasta cumplir con los criterios de parada. Las funciones de evaluación serán ejecutadas en un entorno distribuido con un esquema maestro esclavo, en donde el servidor (maestro) distribuye los individuos de la generación en celdas de procesamiento. La función de evaluación estará dada por Coeficiente de Correlación de Mathews.
Nota: **Clases Involucradas:** Evidences, StartegyRun, Strategy, HBconcreteContactMap, HBStrategyIteration, HBTweaks, HBConcretTweaks, HBDistributedTaskIteration, Evaluator.	
Pre condiciones: Para la evaluación de los individuos del Híbrido se debe tener la trayectoria pre formateada a mapas de contacto. Para la mejora de los individuos con TS se debe tener el estrato (200 pasos de simulación), de la generación.	
Recetas relacionadas de framework: • Receta 1 para manejo de evidencias • Receta 2 de pre procesamiento de la trayectoria • Receta 3 para configuración de las características para la ejecución de las	

Tabla 14. Historia de Usuario Ejecutar estrategia [53].

NOMBRE HISTORIA DE USUARIO: Ejecutar Estrategia.			
Fecha: 28/Maya/2012	Tipo de Actividad	Nueva: X	Modificación:___
Historia Numero: 02	Prioridad: Alta	Usuario:	Técnica: X
Descripción:	Se debe tener las trayectorias de mapa de contacto almacenados en disco (Evidencias) con permisos de lectura. Se inicia Híbrido hasta cumplir los criterios de parada: máximo de iteraciones, función objetivo igual a 1, estancamiento en un óptimo después de un número determinado de iteraciones. Se toman los mejores individuos de AG y serán Mejorados por TS Generación a generación. Reporte por generación del comportamiento de Híbrido y por separado reporte de los individuos mejorados con TS.		
Nota: **Clases Involucradas:** Evidences, HBStartegyRun, Strategy, HBconcreteContactMap,			

HBStrategyIteration, HBTweaks, HBConcretTweaks, TSconcreteStrategy. **Pre condiciones:** Mapas de contacto pre-procesados. Lanzar entorno distribuido. **Recetas relacionadas de framework:** • Receta 1 para manejo de evidencias • Receta 2 de pre procesamiento de la trayectoria • Receta 3 para configuración de las características para la ejecución de las técnicas. Receta 4 definir estrategia completa (AG y TS).

Tabla 15. Historia de Usuario Escoger Individuos a mejorar con TS [53].

NOMBRE HISTORIA DE USUARIO: Mejorar individuos TS.			
Fecha: 28/Maya/2012	Tipo de Actividad	Nueva: X	Modificación:___
Historia Numero: 02	Prioridad: Alta	Usuario:	Técnica: X
Descripción:	Se debe ejecutar la estrategia y realizar al menos 1 función de evaluación sobre la población. Se organiza la población evaluada de mayor a menor según su función de evaluación, se escogen los mejores 8 individuos que serán mejorados con TS, se aplica vecindario en profundidad, terminando el proceso de TS al cumplir con el máximo numero de evaluaciones por individuo, se devuelven los mejores individuos explotados sobre el estrato de la generación actual.		
Clases Involucradas: Evidences, HBStartegyRun, Strategy, HBconcreteContactMap, HBStrategyIteration, HBTweaks, HBConcretTweaks, TSconcreteStrategy, HBdistributedTaskIteration, TSConcreteStrategy. **Pre condiciones:** Mapas de contacto pre-procesados. Lanzar entorno distribuido. Realizar una función de evaluación. **Recetas relacionadas de framework:** • Receta 1 para manejo de evidencias • Receta 2 de pre procesamiento de la trayectoria • Receta 3 para configuración de las características para la ejecución de las técnicas. • Receta 4 definir estrategia completa (AG y TS). • Receta 5: Definir representación de la solución.			

Tabla 16. Historia de Usuario Configurar Parámetros Híbrido [53].

<table>
<tr><th colspan="4">NOMBRE HISTORIA DE USUARIO: Configurar parámetros Híbrido</th></tr>
<tr><td>Fecha: 25/Mar/2012</td><td>Tipo de Actividad</td><td>Nueva: _</td><td>Modificación:</td></tr>
<tr><td>Historia Numero: 03</td><td>Prioridad:</td><td>Usuario: X</td><td>Técnica:</td></tr>
<tr><td>Descripción:</td><td colspan="3">La aplicación deberá permitir la configuración de parámetros para la ejecución de HB, donde se encuentran parámetros como:
Máximo de Iteración de HB: que el usuario podrá definir entre 1 a 200.
Tamaño de estratos: el usuario define un número entero entre 1 y 40.000.
Operador Genético de Selección: Es un numero entre 1 y 10, que indique la probabilidad de escoger el mejor individuo, el valor por defecto es 8 si no se proporciona este.
Operador Genético de Cruce: es un numero entre 0 y 1, que el usuario proporciona, si este no es proporcionado por defecto se inicia en 0.9.
Operador Genético de Mutación: es un numero entero que proporciona el usuario, si este no es proporcionado, por defecto será 0.02.
Tamaño lista tabú: un número entre 3 y 10, para guardar los movimientos, por defecto su valor es de 7.
Vecindario Tabú: Vecindario en profundidad, que permita explorar todas las soluciones. Número máximo de genes a cambiar por el vecindario entre 5 y 8.</td></tr>
<tr><td colspan="4">Nota:
Clases Involucradas: StrategyRun, HBconcreteContactMap, TSStrategyIteration, TsTweaks, HBConcretTweaks, TsTweaks.

Pre condiciones: El usuario debe proporcionar lo parámetros.
Los mapas de contacto están en disco duro.

Recetas relacionadas de framework:
Receta 1 para manejo de evidencias
Receta 2 de pre procesamiento de la trayectoria
Receta 3 para configuración de las características para la ejecución de las técnicas.
Receta 4 definir estrategia completa (AG y TS).
Receta 7. definir ajustes de la estrategia</td></tr>
</table>

Tabla 17. Historia de Usuario Evaluación de Soluciones.

NOMBRE HISTORIA DE USUARIO: Evaluación de soluciones			
Fecha: 26/Mar/2012	Tipo de Actividad	Nueva: X	Modificación:___
Historia Numero: 04	Prioridad:	Usuario: X	Técnica:
Descripción:	La aplicación deberá evaluar la mejor solución encontrada por TS, El usuario debe proporcionar la solución a evaluar junto con la trayectoria de de mapas de contacto, la aplicación debe permitir cargar el archivo de solución y de mapa de contacto almacenados en disco, si no se puede cargar los archivos correspondientes, se mostrara un mensaje de "error al cargar archivos". Si carga los archivos, se inicia la simulación del modelo AC del archivo de solución, se deben generar las medidas TP, TN, Precisión, sensitividad, Especificidad, Coeficiente de correlación de Matthews.		
Nota: **Clases Involucradas: Evidences, StartegyRun, Strategy, TsconcreteContactMap, TSStrategyIteration, Evalution, Report** **Pre condiciones:** Para AG se debe tener la trayectoria pre formateada a mapas de contacto. Para TS se debe tener el estrato (200 pasos de simulación), del individuo a mejorar. **Recetas relacionadas de framework:** • Receta 1 para manejo de evidencias • Receta 2 de pre procesamiento de la trayectoria • Receta 3 para configuración de las características para la ejecución de las técnicas. • Receta 7 definir ajustes de la estrategia. • Receta 8 Definir reportes de la estrategia			

4.5 PLAN DE PRUEBAS

Las pruebas están enfocadas hacia los reportes generados por la técnica híbrida y hacia los reportes estadísticos generados para cada individuo.

Algunas de las pruebas fueron realizadas con PyUnit [54] y su código fuente se encuentra comentado en los módulos realizados en el proyecto, la Figura 41 muestra un ejemplo realizado con PyUnit con la clase de prueba TestPyunit en donde se prueba si el tamaño del vecindario es el adecuado, usamos función assert para comprobar si la variable que se usa es la correcta.

```python
class TestPyUnit(unittest.TestCase):

    tstweaks=TSTweaks(Tweaks)

    def setUp(self):
        print "Preparando contexto"
        self.representation = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
        , 1, 1, 1, 1, 1, 1]
        self.position = 2

    def test(self):
        print "Ejecutando prueba ts_movement"

        r = TSTweaks.ts_movement(self.tstweaks,self.representation, self.position)
        self.assertEqual(r, [1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
        1, 1, 1, 1, 1, 1])

        self.assertNotEqual(r, [1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
        1, 1, 1, 1, 1, 1, 1])

        self.assertTrue((self.position<len(self.representation)),
                        'Error: el valor de position es demasiado grande')

        print "Ejecutando prueba ts_generate_neighborhood"

    def tearDown(self):
        print "Desconstruyendo contexto"
        del self.representation
        del self.position

if __name__ == "__main__":
    unittest.main()
```

Figura 41.Ejemplo Pyunit.

Por otra lado algunas funciones realizadas arrojan resultados probabilísticos, lo que dificultad la realización de pruebas de caja negra, por lo cual se opto por generar archivos planos con los cuales podíamos hacer un seguimiento para verificar el correcto funcionamiento de las funciones. Un ejemplo de esto es la generación del vecindario en profundidad a partir de un individuo, se muestra en la Figura 42.

```python
def ts_generate_neighborhood(self,population):
        next_population = []
        representation = population[0]
        representation = representation.getRepresentation()
        for i in range(0,len(representation)):
            if (i%5)==0:
                solution = self._solutionBuilder.construct()
                next_representation = self.ts_movement(representation, i)
                solution.setRepresentation(next_representation)
                self._repair(solution)

                # Crea un objeto TSSolution y llena sus atributos
                tsSolution = TSSolution()
                tsSolution.setRepresentation(next_representation)
                tsSolution.setModel(solution.getModel())
                tsSolution.setMov(i)
                # va agregando todos los vecinos(TSSolution) a la siguiente poblacion
                next_population.append(tsSolution)
        return next_population
```

Figura 42.Función de vecindario en Profundidad.

Se indica pos en 4, el cual será el tamaño del vecindario, el código devuelve la población (next_population) generada a partir del mejor individuo (population[0]) y del tamaño de la población, utilizando el movimiento tabú de la función ts_movement (representation, pos), la Figura 43 muestra los resultados correctos de la función.

```
          Vecindario Ordenado

Tamanio de la Poblacion:    4

    00100000100111001010101101011111100100001011111111
    00100000100101001010101101011111100100001011011111
    00100000100101101010101101011111100100001011111111
    00100000100101011010101101011111100100001011111111
```

Figura 43.Resultados Vecindario en Profundidad.

5 ANÁLISIS Y RESULTADOS

Los modelos obtenidos como resultado de una serie de experimentos realizados sobre el conjunto de datos 1 serán analizados en este capítulo.

5.1 CONJUNTOS DE DATOS

Los datos de la trayectoria de la proteína se organizan en 10 conjuntos de datos, como fue propuesto en [12], en donde se organizan en 10 secciones de 4000 pasos de simulación como se ilustra la figura 44.

Dataset	Marcos PDB	Pasos de simulación por marco	Total (pasos de simulación)
1	0 a 9	400	4000
2	10 a 19	400	4000
3	20 a 29	400	4000
4	30 a 39	400	4000
5	40 a 49	400	4000
6	50 a 59	400	4000
7	60 a 69	400	4000
8	70 a 79	400	4000
9	80 a 89	400	4000
10	90 a 99	400	4000

Figura 44.Conjunto de datos.(Tomado de [12]).

En este proyecto se consideran los conjuntos de datos de las secciones 1, 2, 4 y 7 para realizar evaluaciones por exclusión de datos de los individuos 13, 33, 53.

5.2 OBTENCION DE MODELOS

En el conjunto de datos 1 se realizaron las diferentes ejecuciones de las técnicas híbridas como fue observado en la sección 3.4.4 de construcción del modelo, el comportamiento del algoritmo híbrido por mejora de hijos tiene un mejor comportamiento frente a las demás técnicas.

La Figura 45 muestra una comparación de las dos primeras iteraciones ILAS para el AG frente a la primera iteración ILAS del Híbrido. En donde se puede observar que el comportamiento del híbrido (Línea azul) es similar al comportamiento del AG para ILAS 2

(Línea verde), esto quiere decir que el híbrido mejora hasta en dos iteraciones ILAS al AG, sin embargo el tiempo de ejecución que requiere el híbrido es superior al de AG.

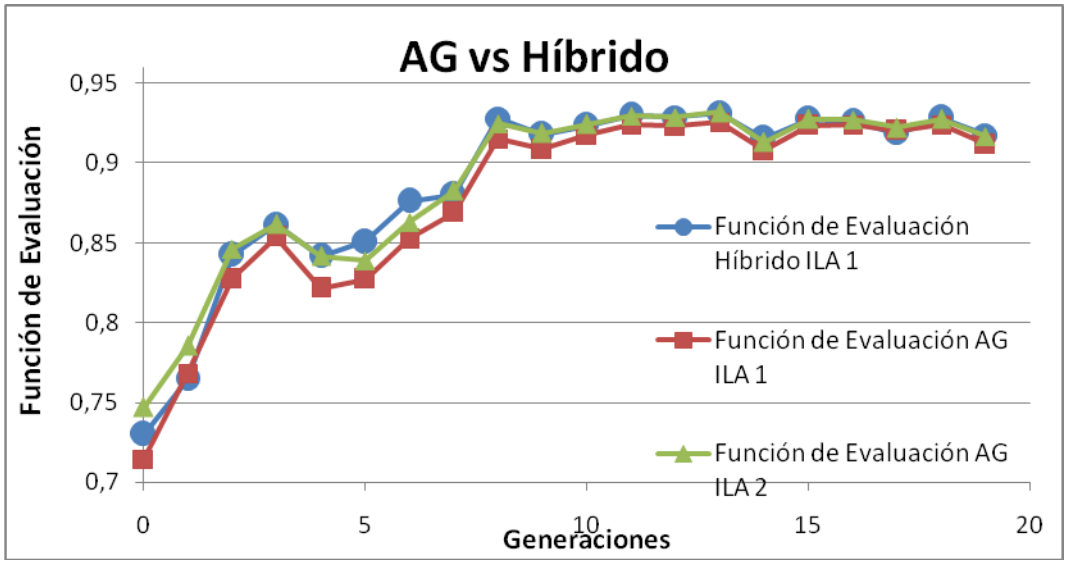


Figura 45.Grafica AG VS Híbrido por ILAS.

La Figura 46 muestra el promedio de entre AG y el Híbrido, en donde se observa claramente como el hibrido (Línea con triangulo) supera al AG (Línea con rombo) generación a generación. Además se ratifica el hecho que en ambas técnicas se mejora la descendencia de los individuos tras el paso del tiempo.

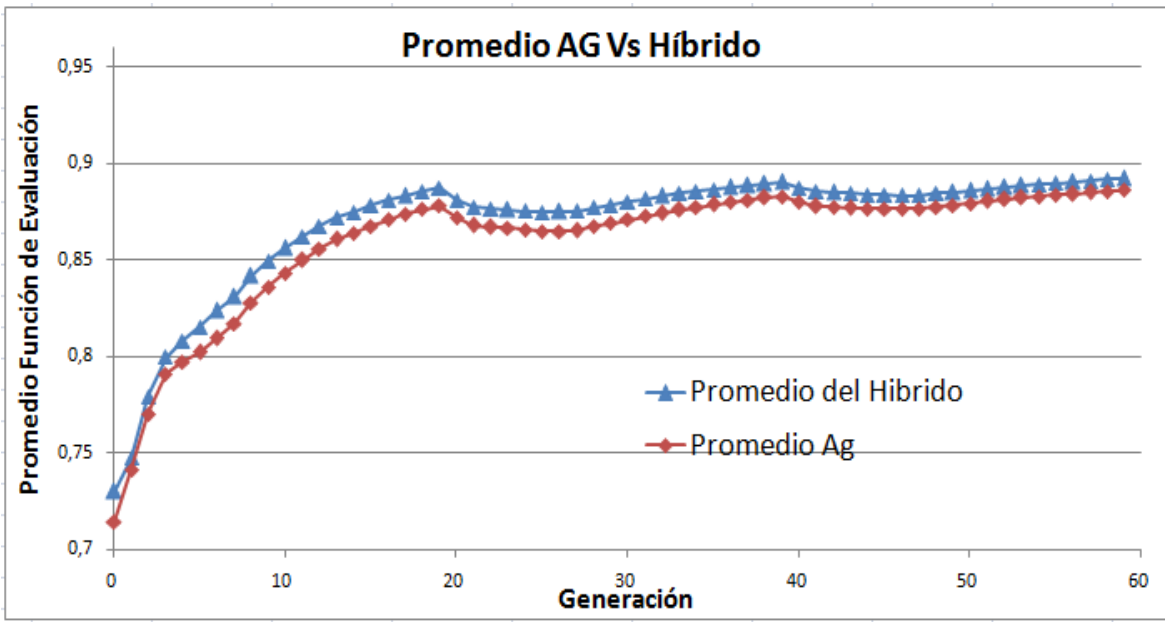


Figura 46.Grafica AG VS Híbrido Promedios.

Para los conjuntos de datos 1, 2, 4 y 7 se realizó evaluación por exclusión para los individuos 13, 33 y 53 para cada uno de los híbridos. En la Figura 47 se observa que el individuo 13 obtenido con el híbrido por mejora de hijos, en donde el conjunto de datos 4 y 7 obtiene un mejor desempeño en espacio ROC, el comportamiento decae un poco para el datase 2.

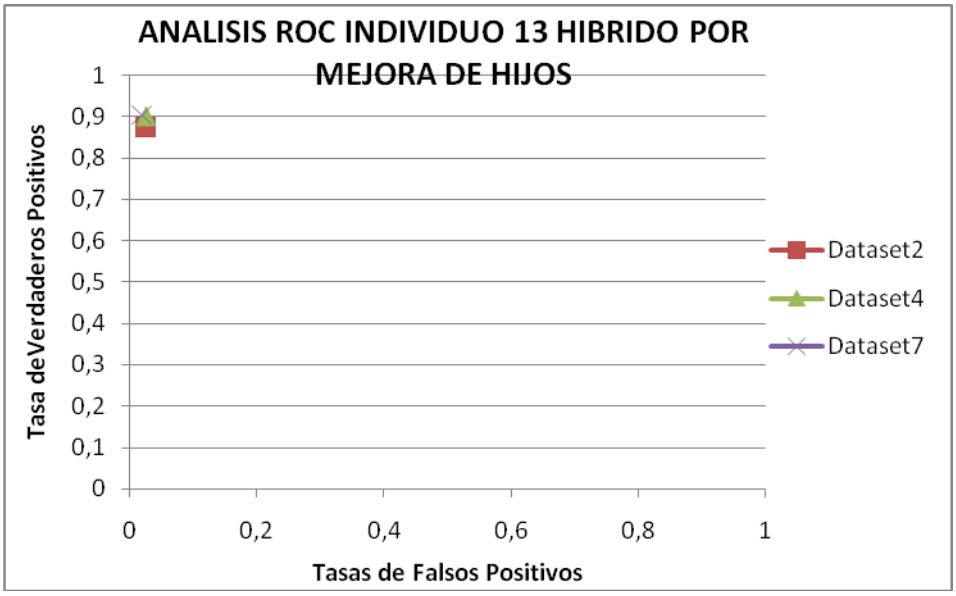


Figura 47.Grafica Espacio ROC individuo 13 - Híbrido por mejora de Hijos.

Para el híbrido por intensificación, el comportamiento es similar al anterior, pero se aleja un poco de la tasa de verdaderos positivos como lo muestra la Figura 48

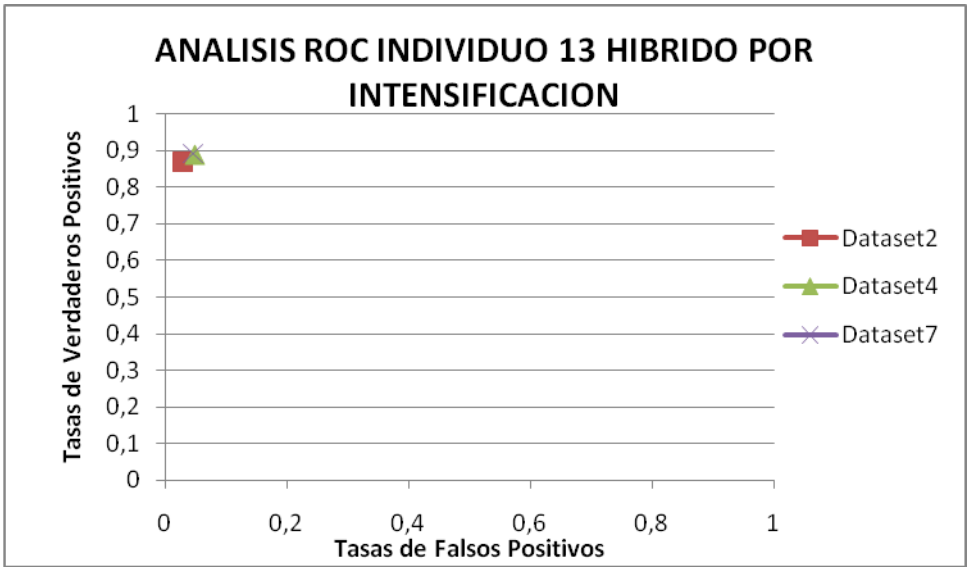


Figura 48.Grafica Espacio ROC individuo 13 Híbrido por intensificación.

El espacio ROC para el individuo 33 presenta un mejor comportamiento con respecto al individuo 13 en ambas técnicas híbridas. La grafica de la Figura 49, el individuo presenta un mejor comportamiento en el conjunto de datos 7 para la técnica híbrida por mejora de hijos, con respecto a los demás conjuntos de datos, a diferencia de la técnica por intensificación, en donde el individuo 33 tiene un mejor comportamiento en el conjunto de datos 4 y 2 como muestra la gráfica de la Figura 50.

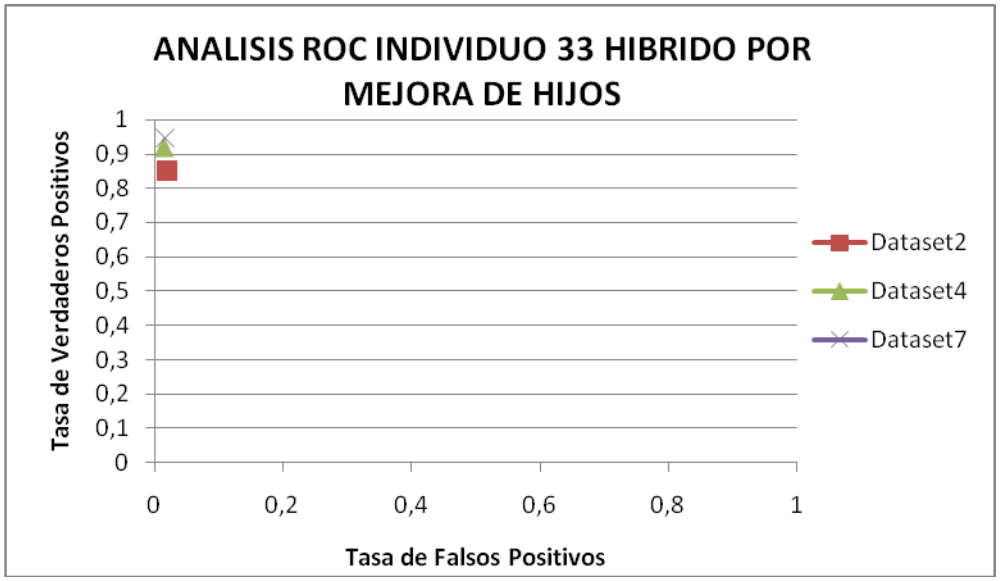


Figura 49.Grafica Espacio ROC individuo 33 - Híbrido por mejora de Hijos.

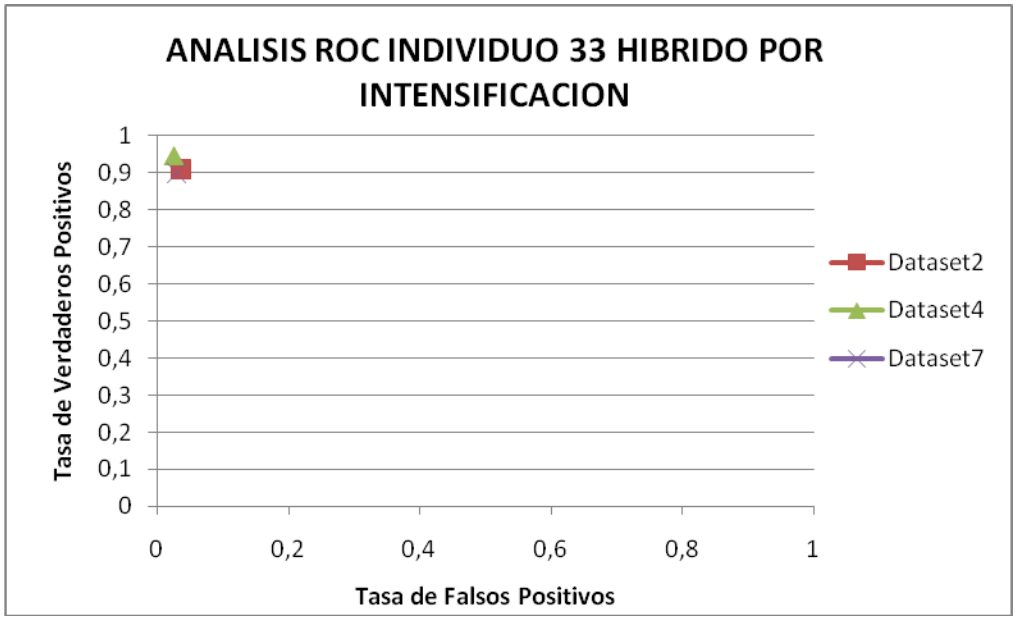


Figura 50.Grafica Espacio ROC individuo 33 - Híbrido por Intensificación.

Para el individuo 53 en la técnica híbrida por mejora de hijos el comportamiento del individuo en los conjuntos de datos es similar, se mantiene muy cerca uno del otro, superando los comportamientos observados en los otros individuos. La grafica de la Figura 51 muestra el comportamiento de este individuo.

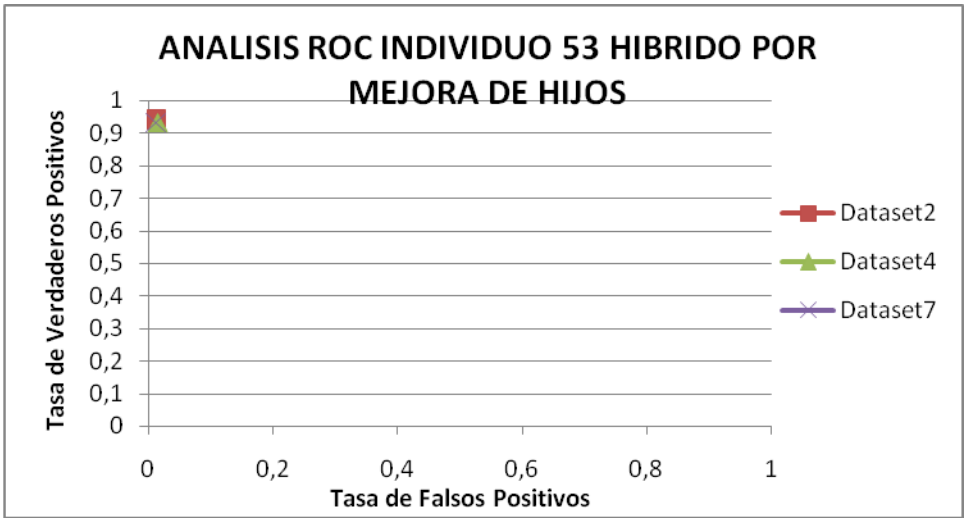


Figura 51.Grafica Espacio ROC individuo 33 - Híbrido por mejora de Hijos.

En la gráfica de la Figura 52, la técnica híbrida por intensificación para el individuo 53, se nota una mejora con respecto a los otros individuos, en donde el comportamiento del individuo en el conjunto de datos 2 y 7 es mejor con respecto al conjunto de datos 4, comportamiento diferente al visto en los otros individuos el comportamiento del individuo en el conjunto de datos 4 fue mejor.

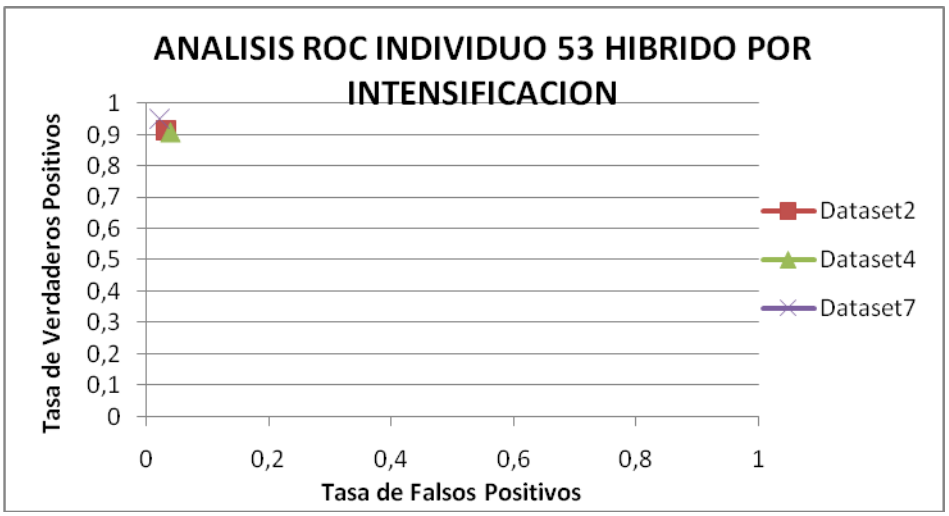


Figura 52.Grafica Espacio ROC individuo 53 - Híbrido por Intensificación.

En los individuos obtenidos en ambas técnicas, al observar la representación de fenotipo se observaron ciertos patrones en los individuos que se muestran en la figura 52.

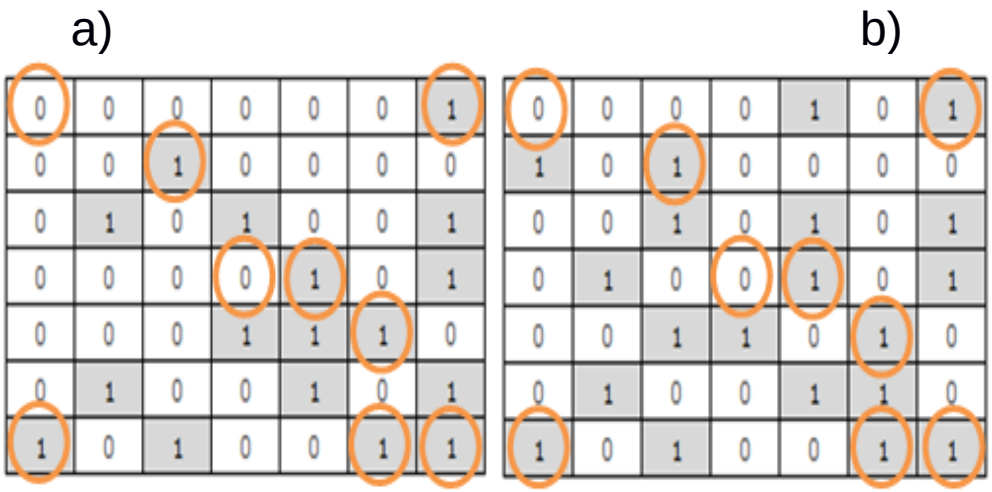


Figura 53.Patrones en los individuos a) Individuo por Intensificación b) Individuo por Mejora de Hijos.

Dados estos patrones se podría pensar en hacer un nuevo vecindario para obtener mejores modelos de AC en donde los bits de las celdas marcadas con el circulo quedarían fijos y los demás bits de las celdas no marcadas pueden variar. En el anexo 2 se muestran diferentes modelos de AC, algunos de estos modelos se les realizaron algunos cambios de bits, y se observaron algunas características que no habían sido tenido en cuenta en el [12], como algunos patrones dentro de la vecindad que generan mejores modelos:

- El bit V7, V10, V26, V34, V39, V42, V43, V48 y V49 este siempre en 1
- El bit V1 y v25 este en 0

V1	V2	V3	V4	V5	V6	V7
V8	V9	V10	V11	V12	V13	V14
V15	V16	V17	V18	V19	V20	V21
V22	V23	V24	V25	V26	V27	V28
V29	V30	V31	V32	V33	V34	V35
V36	V37	V38	V39	V40	V41	V42
V43	V44	V45	V46	V47	V48	V49

Figura 54. Vecindario Modelo AC.

Dadas estas condiciones se plantea un esquema de vecindario en prioridad, que puede ser usado en AG, TS y en cualquiera de los híbridos, que reduciría el número de vecinos a

evaluar, además disminuir la carga de procesamiento ya que se haría un menor número de evaluaciones de individuos. La Figura 55 muestra un esquema de vecindario en prioridad en donde las celdas de color gris con bits 0 y 1 son valore fijos observados en los diferentes modelos AC obtenidos, las celdas de color rojo con signo de interrogación (?) son los candidatos a ser modificados (pasar de 1 a 0 ó viceversa) dándole mayor prioridad para generar el vecindario de AC, y las celdas de color amarillo marcados con la X tendrían una prioridad menor para ser cambiados.

0	?	?	X	X	X	1
X	?	1	?	X	X	X
X	?	?	?	?	X	X
X	X	?	0	1	X	X
X	X	?	?	?	1	X
X	?	X	1	?	?	1
1	?	X	X	?	1	1

Figura 55. Esquema de vecindario en Prioridad.

6 CONCLUSIONES Y TRABAJOS FUTUROS

6.1 CONCLUSIONES

En el trabajo se pudieron aprovechar las características del framework CAIF-PFT, siendo una herramienta que se puede utilizar para el desarrollo de técnicas computación evolutiva y metaheurísticas que busquen soluciones al plegamiento de proteínas como las planteadas en este proyecto de investigación, debido a que proporciona una arquitectura adecuada para el desarrollo y una serie de características que facilitan la evaluación de modelos de AC.

En el trabajo se analizó el comportamiento de las metaheurísticas AG, TS y las combinaciones de estas (hibridación por intensificación, hibridación por mejora de hijos), se observó que la metaheurística hibrida por mejora de hijos presenta un mejor comportamiento frente a las otras técnicas, obteniendo mejores valores en la función de evaluación y modelos con precisiones superiores al 93% para el dataset 1 y superiores al 96% para otros dataset; satisfaciendo la medidas de calidad propuestas como objetivos del proyecto.

Se puede concluir que la diferencia entre el comportamiento de AG y la hibridación por mejora de hijos, está dada por la mejora que realiza TS a la descendencia de AG, sin embargo el tiempo de procesamiento del híbrido es mayor al tiempo de AG como se observó en la gráfica 25 debido a que el híbrido debe hacer un mayor número de evaluaciones que las realizadas por AG.

El entorno distribuido basado en la arquitectura maestro - esclavo propuesto en [12] usado en este proyecto resulta de gran utilidad para la evaluación de los individuos generados por las diferentes técnicas, ahorrando tiempo en la ejecución de las mismas .

Una mala selección de los parámetros de las técnicas híbridas (AG y TS) puede generar problemas en el comportamiento de los algoritmos haciendo que los resultados no alcancen el umbral propuesto para este trabajo de investigación.

Los modelos de AC obtenidos por las técnicas híbridas alcanzan una precisión superior al 90% para los primeros 4.000 pasos de simulación siendo este el segmento más complejo en la trayectoria de plegamiento de proteínas, debido a que allí se presenta un proceso de estabilización inicial de la proteína.

Otra característica observada es el tamaño de las vecindades de los modelos AC, el AG tiende a tener vecindades con mayor tamaño para las primeras iteraciones, a diferencia de lo híbridos que con ayuda de TS bajan el tamaño de la vecindad sin perder la buena precisión en los modelos. El tamaño de la vecindad implica un mayor número de reglas a analizar de tal forma que a mayor cantidad de vecinos mayor el número de reglas y a menor cantidad de vecinos, menor es el número de reglas.

6.2 TRABAJOS FUTUROS

Se puede considerar ejecutar los híbridos en un entorno distribuido más amplio de tal manera que se pueda explorar más el espacio de búsqueda.

Una de las mejoras que se propone es una heurística diferente a las planteadas para la búsqueda de los vecindarios de las técnicas híbridas donde se busquen en niveles de profundidad mayores a uno y además que busque cambiar más de un bit a la vez, tratando de explorar mejores espacios de búsqueda para encontrar mejores individuos en un menor tiempo de ejecución. Por lo cual se propone realizar un vecindario con algoritmos greedy como lo son algoritmos de kruskal y de Prim con los cuales se han encontrado buenos resultados para problemas binarios de optimización combinatoria, esperando disminuir las evaluaciones realizadas y obtener mejores resultados que los encontrados en este trabajo.

Se podría experimentar guardando en la lista tabú individuos y no movimientos como fue planteado en este trabajo, esto se realizaría con el fin de explorar un mayor espacio de búsqueda y de esta forma encontrar mejores soluciones en menor tiempo de ejecución.

En el presente trabajo se mostraron los resultado de un validador de reglas que muestra las reglas en un mapa de contacto, como trabajo futuro se plantea que el validador de reglas pueda acoplarse con un visualizador de proteínas, de tal forma que al ver una regla esta se vaya visualizando en 3D y resulte más fácil su entendimiento.

Se propone experimentar con otras técnicas de Búsqueda Local (enfriamiento simulado, GRAPS, Vecindario Variable), que al ser hibridadas con AG, guíen el proceso hacia la búsqueda de modelos de AC con mejores precisiones.

Como trabajo futuro se propones una hibridación con otros algoritmos poblacionales como CHC(Cross generational elitist selection, Heterogeneous recombination and Cataclism mutation) que es una variación del AG tradicional y que en problemas binarios de optimización combinatoria obtiene mejores resultados que un AG estándar, según se reporte en la literatura del área.

7 REFERENCIAS Y BIBLIOGRAFIA

[1] K. A. Dill*, et al.*, "The protein folding problem," *Annual review of biophysics,* vol. 37, p. 289, 2008.

[2] C. D. Snow*, et al.*, "Absolute comparison of simulated and experimental protein-folding dynamics," *Nature,* vol. 420, pp. 102-106, 2002.

[3] P. Xicohtencatl*, et al.*, "Plegamiento de las proteínas," *BUAP Cuerpo Académico de Adaptación, Escuela de Medicina Veterinaria y Zootecnia,Universidad Autónoma Metropolitana- Xochimilco,* p. 6, 2006.

[4] F. R. M. Antonio*, et al.*, "Sistema predictor de estructuras de proteínas utilizando Dinámica Molecular (MODYPP)," *RISCE Revista Internacional de Sistemas Computacionales y Electrónicos,* p. 63, 2009.

[5] T. Schlick, "Molecular Dynamics: Basics," in *Molecular modeling and simulation: an interdisciplinary guide*. vol. 21, ed: Springer Verlag, 2010, pp. 18-22.

[6] A. Nicastro and S. Sferco, "Simulación con el método de Monte Carlo para estudiar la estabilidad de factores estructurales de una hornilla B," *FABICIB,* vol. 9, pp. 88-99, 2005.

[7] T. Schlick, "Monte Carlo Techniques," in *Molecular modeling and simulation: an interdisciplinary guide*. vol. 21, ed: Springer Verlag, 2010, pp. 18-22.

[8] T. de Camino Beck, "Un Lenguaje para la Especificación de Autómatas Celulares con Aplicaciones en Biología," *Master's thesis, Instituto Tecnologico de Costa Rica,* pp. 12-45, 2000.

[9] J. Kari, "Theory of cellular automata: A survey," *Theoretical computer science,* vol. 334, pp. 3-33, 2005.

[10] T. Bäck*, et al.*, "Inverse design of cellular automata by genetic algorithms: an unconventional programming paradigm," *Unconventional Programming Paradigms,* pp. 161-172, 2005.

[11] I. Fister*, et al.*, "Hybridization of Evolutionary Algorithms," p. 15, 2011.

[12] N. Diaz, "Metodología y framework computacional para el diseño inverso de modelos de autómatas celulares de secuencias cortas de aminoácidos soportando en un proceso de minería de datos," *Tesis de Maestría, Universidad del Valle,* p. 56, 2010.

[13] N. Diaz and I. Tischer, "Minería de modelos de autómatas celular en trayectoria de plegamiento de proteína," *Sexto congreso colombiano de computación,* pp. 1-6, 2011.

[14] D. L. Ensign*, et al.*, "Heterogeneity even at the speed limit of folding: large-scale molecular dynamics study of a fast-folding variant of the villin headpiece," *Journal of molecular biology,* vol. 374, pp. 806-816, 2007.

[15] D. Martín*, et al.*, "Bioquimica de Harper," *Editorial El Manual Moderno,SA de CV México, DF,* vol. 15, pp. 29-58, 2001.

[16] A. V. Blanco, "Química Biológica " *Blanco A editor. ed. Buenos Aires: El Ateneo,* vol. 8, pp. 21-42, 2006.

[17] L. Olivares-Quiroz and L. G. C. Scherer, "Plegamiento de las proteínas: Un problema interdisciplinario," *Rev. Soc. Quím. Méx,* vol. 48, pp. 95-105, 2004.

[18] P. F. Xicohtencatl, *Plegamiento de las proteínas*, 2006.

[19] T. Schlick, "The Technique of X-ray Crystallography And The Technique of NMR Spectroscopy," in *Molecular modeling and simulation: an interdisciplinary guide*. vol. 21, ed: Springer Verlag, 2010, pp. 18-22.
[20] K. Ginalski, "Comparative modeling for protein structure prediction," *Current opinion in structural biology,* vol. 16, pp. 172-177, 2006.
[21] M. A. Martí-Renom*, et al.*, "Comparative protein structure modeling of genes and genomes," *Annual review of biophysics and biomolecular structure,* vol. 29, pp. 291-325, 2000.
[22] S. A. Adcock and J. A. McCammon, "Molecular dynamics: survey of methods for simulating the activity of proteins," *Chemical reviews,* vol. 106, pp. 1589-1615, 2006.
[23] D. M. Webster, *Protein structure prediction: methods and protocols* vol. 143: Humana Pr Inc, 2000.
[24] Y. Xu and J. Wooley, *Computational methods for protein structure prediction and modeling: basic characterization* vol. 1: Springer Verlag, 2007.
[25] G. Terrazas*, et al.*, "An evolutionary methodology for the automated design of cellular automaton-based complex systems," *Journal of Cellular Automata,* vol. 2, pp. 77–102, 2007.
[26] J. B. Santana*, et al.*, "Metaheurísticas: Una revisión actualizada.," *La Laguna, España: Departamento de Estadística, Investigación Operativa y Computación. Universidad de La Laguna,* pp. 4-31, 2004.
[27] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Computers & Operations Research,* vol. 13, pp. 533-549, 1986.
[28] S. Luke, *Essentials of metaheuristics*, 2010.
[29] C. García, "Metaheurísticas e ingeniería del software," *Tesis Doctoral,* pp. 59-83, 2007.
[30] E. G. Talbi, *Metaheuristics:From design to implementation*: Wiley Online Library, 2009.
[31] M. Gendreau and J.-Y. Potvin, *Handbook of Metaheuristics* vol. 146: Springer, 2010.
[32] B. M. Batista and F. Glover, "Introducción Búsqueda Tabú," *Revista Iberoamericana de Inteligencia Artificial. No.19,* p. 35, 2003.
[33] K. E. N. I. Maeda and C. Sakama, "Identifying cellular automata rules," *Journal of Cellular Automata,* vol. 2, pp. 1–20, 2007.
[34] F. Glover, "Tabu Search-Part I,II," *Orsa Journal on Computing,* vol. 3, p. 17, 1989.
[35] O. Lozano, "Búsqueda Tabú (tabu search)," *http://dis.unal.edu.co/~fgonza/courses/2003/pmge/present/tabusearch.pdf:Accesado,* p. 5, 2003.
[36] T. Bossomaier*, et al.*, "A new paradigm for evolving cellular automata rules," 1999, pp. 169-176.
[37] E. Sapin*, et al.*, "A new universal cellular automaton discovered by evolutionary algorithms," 2004, pp. 175-187.
[38] S. A. Billings and Y. Yang, "Identification of probabilistic cellular automata," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on,* vol. 33, pp. 225-236, 2003.
[39] Y. Zhao and S. Billings, "Neighborhood detection using mutual information for the identification of cellular automata," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on,* vol. 36, pp. 473-479, 2006.
[40] C. Blum*, et al.*, "Hybrid metaheuristics," *Hybrid Optimization,* pp. 305-335, 2011.
[41] C. Blum*, et al.*, "A brief survey on hybrid metaheuristics," *ALBCOM research group,* p. 16, 2010.

[42] W. Jaziri, *Local Search Techniques: Focus on Tabu Search* vol. 1: In-Teh, 2008.
[43] J. C. Pete Chapman, Randy Kerber*, et al.*, "CRISP-DM Step-by-step data mining guide," p. 78, 2000.
[44] J. Demšar*, et al.*, "Orange: From experimental machine learning to interactive data mining," *Knowledge discovery in databases: PKDD 2004,* pp. 537-539, 2004.
[45] O. Carugo, "Detailed estimation of bioinformatics prediction reliability through the Fragmented Prediction Performance Plots," *BMC bioinformatics,* vol. 8, p. 380, 2007.
[46] P. Baldi*, et al.*, "Assessing the accuracy of prediction algorithms for classification: an overview," *Bioinformatics,* vol. 16, p. 412, 2000.
[47] M. H. Zweig and G. Campbell, "Receiver-operating characteristic (ROC) plots: a fundamental evaluation tool in clinical medicine," *Clin Chem,* vol. 39, pp. 561-577, 1993.
[48] J. J. Moré and Z. Wu, "Distance geometry optimization for protein structures," *Journal of Global Optimization,* vol. 15, pp. 219-234, 1999.
[49] M. Vendruscolo*, et al.*, "Recovery of protein structure from contact maps," *Folding and Design,* vol. 2, pp. 295-306, 1997.
[50] P. Moscato and C. Cotta, "Una introducción a los algoritmos meméticos," *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial,* 2003.
[51] G. F. Escribano, "Introducción a Extreme Porgraming," *Ingenieria del Sofware,* vol. 1, p. 14, 2002.
[52] D. Wells. (2001, *Extreme programming*.
[53] E. Sanchez*, et al.*, "Mejorando la gestión de historias de usuario en eXtreme Programming," *Departamento de Sistemas Informaticos y Computacion Universidad Politecnica de Valencia,* vol. 1, p. 10, 2008.
[54] S. Purcell. (2002, *Python unit testing framework*.