UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS

FACULTAD DE CIENCIAS QUÍMICAS E INGENIERÍA

CENTRO DE INVESTIGACIÓN EN INGENIERÍA
Y CIENCIAS APLICADAS

ALGORITMO DE RECOCIDO SIMULADO PARA MAXIMIZAR LA RESISTENCIA MECÁNICA EN ACEROS MICROALEADOS

TESIS PROFESIONAL
PARA OBTENER EL GRADO DE:

MAESTRÍA EN INGENIERÍA Y CIENCIAS APLICADAS
OPCIÓN TERMINAL EN TECNOLOGÍA ELÉCTRICA

P R E S E N T A:

L. I. JAZMÍN YANEL JUÁREZ CHÁVEZ

ASESOR: DR. MARCO ANTONIO CRUZ CHÁVEZ.
COASESOR: DR. SERGIO ALONSO SERNA BARQUERA.

CUERNAVACA, MOR. JULIO 2011

Resumen

En este trabajo de investigación se presenta un algoritmo de Recocido Simulado para maximizar la resistencia mecánica en aceros microaleados. Para el desarrollo de este trabajo se implementó un conjunto de estructuras de vecindad del tipo doble, triple, cuádruple, quíntuple e híbrido aleatorio. Se evaluaron la eficiencia y eficacia de estas estructuras utilizando un Algoritmo de Búsqueda Local Iterada el cual se evaluó con diferentes tamaños de vecindad. En base al análisis de desempeño realizado se encontró que la estructura de vecindad que mejores resultados dio fue la estructura híbrida la cual se integró al Algoritmo de Recocido Simulado. Se realizó una comparación de desempeño entre el algoritmo propuesto de Recocido Simulado y el algoritmo de Búsqueda Local Iterada. En base al análisis de desempeño se encontró que el Algoritmo de Recocido Simulado no es adecuado para aplicarlo en el problema que requiere maximizar la resistencia mecánica en aceros microaleados porque en un intervalo grande de su ejecución, el valor de la función objetivo no mejora. Fuera de ese intervalo, el algoritmo se comporta como un Algoritmo de Búsqueda Local Iterada. De acuerdo a los resultados se observó que con respecto a Recocido Simulado, la Búsqueda Local Iterada es mejor en eficiencia y ofrece resultados competitivos en cuanto a la eficacia. Las pruebas de optimización se realizaron en los cluster Ciicap y Nopal (del CIICAp – UAEM y del ITVer, respectivamente).

Abstract

This research presents the Simulated Annealing Algorithm for Maximizing Mechanical Strength Microalloyed Steels. For the development of this work it was implemented a set of neighborhood structures like: Double Random, Triple Random, Quadruple Random, Quintuple Random and Hybrid Random. We evaluated the efficiency and effectiveness of these structures using Iterated Local Search Algorithm, which was evaluated with different neighborhood sizes. Based on performance analysis, the neighborhood structure that gave the best results was the hybrid structure that was integrated in the Simulated Annealing algorithm, by comparison with the performance between the proposed algorithm Simulated Annealing and Iterated Local Search algorithm. Based on performance analysis conducted found that the neighborhood structure that gave best results was the hybrid structure which joined the Simulated Annealing Algorithm. A comparison of performance between the algorithms proposed Simulated Annealing Algorithm and Iterated Local Search. Based on performance analysis found, the simulated annealing algorithm is not suitable for problems that needs to maximize the mechanical strength microalloyed steels because in a large range of implementation, the value of the objective function does not improve. Outside this interval, the algorithm behaves as an Iterated Local Search Algorithm**.** According to the results showed that with respect to Simulated Annealing, Iterated Local Search is improved efficiency and provides competitive results in terms of efficiency**.** The optimization tests were conducted in the cluster Ciicap and Nopal (the CIICAp - UAEM and ITVER, respectively).

Agradecimientos

A

CONACYT

Por el apoyo económico otorgado para la realización de esta maestría, sin el cual, no hubiese sido posible.

A

Fundación Telmex

Por el medio de comunicación proporcionado como complemento para la realización de mis estudios así como el apoyo económico otorgado.

Al

SITAUAEM - Sección XV de la FCAeI

Por el apoyo económico otorgado en este periodo de la maestría y a los representantes que estuvieron en el momento de mi estudio y que me apoyaron con la información para obtener los beneficios que otorga a sus integrantes. Gracias Profesores: Manuel Chávez Ramos, Raúl Martínez, Julián Rosales, Felipe Bonilla y Patricia L.

Al

Dr. Marco Antonio Cruz Chávez

Por su insistencia para que entrara a estudiar la maestría para que mis conocimientos se incrementaran y mi crecimiento académico mejorara. Gracias por insistir y por activar la chispa de la programación.

A

Mi comité tutoral y revisores de tesis

Dr. Marco Antonio Cruz Chávez (Asesor), Dr. Sergio Alfonso Serna Barquera (Coasesor), Dr. Bernardo Campillo Illanes, Dr. Martín Heriberto Cruz Rosales, M.C. Jesús del Carmen Peralta Abarca. Gracias por su tiempo invertido para que comprendiera un tema nuevo para mí. Gracias por su orientación y consejos para mejorar mi trabajo y que este se vea reflejado.

Dedicatorias

A esa energía maravillosa que nunca me abandonó en este camino a pesar de mis fallas y tropiezos, gracias por darme la fuerza para no desistir, gracias, por cuidar de lo que más amo y es mi mayor tesoro. A ti, gracias, porque para llegar a ti, estar contigo y sentirte no necesito de intermediarios. Gracias.

A mis creadores, por tener estas hermosas cualidades: sabiduría, paciencia, amor y perdón. Gracias, por guiar mis pasos a través de este largo sendero. Gracias, por convertirse en mi luz, mi fuerza y mi aliento para continuar en esta grandiosa batalla que es la vida. Gracias Simona Chávez Radilla y Teófilo Juárez Torres.

A mis compañeros de juegos, de travesuras, de aprendizajes, de alegrías y tristezas. Gracias por soportarme y por apoyarme en mis locuras. Gracias porque sé que siempre contaré con ustedes como ustedes contarán conmigo. Gracias Carlos, Ivonne y Oscar Juárez Chávez.

A la creación que considero perfecta. Gracias por permitirme conocerlos, por mostrarme lo más bello que tiene un ser humano, su inocencia, su alegría, su amor, sus sonrisas, su valor y hasta sus travesuras. Gracias por levantarme el ánimo con su risa y por hacerme sentir que todo vale la pena lograrlo. Gracias a ustedes, mis sobrinos Balam, León, Luna, Saori y a los que sigo esperando conocer para compartir su maravillosa transformación.

A mis cuñadas por ser las portadoras de lo más asombroso que he conocido. Gracias Erika y Wendy. Y a mi cuñado consentido, por ser parte de esta gran familia, mi familia y por amar a mi peor pesadilla. Gracias Uziel.

A mis mejores amigos. Gracias por su paciencia, por su tiempo, por su amistad incondicional, por las porras brindadas para incrementar mi ánimo, por compartir conmigo muchas bellas experiencias y por ser parte importante de mi vida y de mi crecimiento como ser humano. A ustedes Andrés, Gloria, Magnolia, Ivonne y Sandy, muchas gracias.

A mis compañeros de maestría. Por el gusto y el placer de conocerles, por transmitirme su energía, su coraje y fuerza, por enseñarme lo que es trabajar en un verdadero equipo, por hacerme sentir una estudiante nuevamente y por las bellas anécdotas que quedarán grabadas en mi memoria para siempre. Nos los olvidaré. Gracias Abraham, Betty, Christian E, Christian, Eduardo, Jorge, Luis y Yessi.

Y al AmoR, que es el remedio para muchos males que aquejan a los habitantes de este planeta y al cual he tenido la oportunidad de conocer a través de sus diferentes caras y que deseo de corazón que nunca falte para poder transmitirlo de igual manera. A ti, gracias.

Sólo hay un bien: el conocimiento.
Sólo hay un mal: la ignorancia.
Sócrates

Nomenclatura

Fe	Fierro
C	Carbono
Cr	Cromo
Cu	Cobre
Mn	Manganeso
Mo	Molibdeno
Ni	Níquel
V	Vanadio
N	Nitrógeno
Nb	Niobio
Ti	Titanio
P	Fósforo
S	Azufre
Si	Silicio
Sm_LS	Solucionmejor_LocalSearch (Búsqueda Local)
Sm_ILS	Solucionmejor_IteratedLocalSearch (Búsqueda Local Iterada)

Contenido

RESUMEN ---------- I

NOMENCLATURA ---------- VI

CONTENIDO ---------- VII

ÍNDICE DE FIGURAS ---------- IX

ÍNDICE DE TABLAS ---------- XIII

CAPÍTULO 1 ---------- 1

INTRODUCCIÓN ---------- 1

1.1 Teoría de la Complejidad de los Algoritmos ---------- 2

1.2 Estado del Arte ---------- 5

1.3 Objetivo de la Investigación ---------- 6

1.4 Alcance de la Investigación ---------- 7

1.5 Contribución de la Tesis ---------- 7

1.6 Organización de la Tesis ---------- 8

CAPÍTULO 2 ---------- 10

ACEROS MICROALEADOS ---------- 10

2.1 Tamaño de Grano ---------- 13

2.2 Precipitados ---------- 14

2.3 Resistencia Mecánica ---------- 15

CAPÍTULO 3 ---------- 18

ALGORITMO DE RECOCIDO SIMULADO ---------- 18

3.1 Introducción ---------- 18

3.2 Algoritmo de Recocido Simulado ---------- 19

3.3 Estructura de Vecindad Híbrida ---------- 20

3. 4 Esquema Generalizado del Algoritmo de Recocido Simulado ---------- 34

3.5 Metodología de Sintonización ---------- 39

3.6 Generación Aleatoria de Instancias de Prueba ---------- 42

3.7 Análisis de la Complejidad del Algoritmo de Recocido Simulado ---------- 44

CAPÍTULO 4 ----------47

RESULTADOS EXPERIMENTALES DEL ALGORITMO DE RECOCIDO SIMULADO ----------47

4.1 Descripción del Equipo Utilizado----------47

4.2 Análisis de la Estructura de Vecindad con el Algoritmo de Búsqueda Local Iterada ----------49

4.3 Análisis de Sensibilidad ----------58

4.4 Análisis de Eficacia y Eficiencia del Algoritmo de Recocido Simulado ----------61

CAPÍTULO 5 ----------66

CONCLUSIONES Y TRABAJOS FUTUROS----------66

5.1 Conclusiones----------66

5.2 Trabajos Futuros ----------67

REFERENCIAS----------68

APÉNDICE A ----------71

ESTRUCTURAS DE DATOS UTILIZADAS EN EL ALGORITMO DE RECOCIDO SIMULADO----------71

APÉNDICE B ----------73

FUNCIÓN TEMPORAL DE LA ESTRUCTURA DE VECINDAD DOBLE ALEATORIO ----------73

FUNCIÓN TEMPORAL DE LA ESTRUCTURA DE VECINDAD TRIPLE ALEATORIO----------74

FUNCIÓN TEMPORAL DE LA ESTRUCTURA DE VECINDAD CUÁDRUPLE ALEATORIO ----------76

FUNCIÓN TEMPORAL DE LA ESTRUCTURA DE VECINDAD QUÍNTUPLE ALEATORIO ----------77

COMPLEJIDAD DEL ALGORITMO DE RECOCIDO SIMULADO ----------79

GLOSARIO DE TÉRMINOS ----------82

Índice de Figuras

Figura 1-1 Clasificación de los Métodos de Optimización [Martínez-Oropeza,A., 2010]. *2*

Figura 1-2 Clases de Complejidad para la decisión de Problemas [Talbi,E., 2009]. *3*

Figura 2-1(a) Los átomos cercanos a los límites de los tres granos no tienen una distancia o un arreglo en equilibrio. (b) Granos y límites de grano en una muestra de acero inoxidable. (Cortesía del Dr. A. Deardo.) [Askeland y Phulé, 2004]. *13*

Figura 2-2 Consideraciones para tener un endurecimiento efectivo por dispersión: (a) La fase precipitada debe ser dura y discontinua, (b) las partículas de la fase dispersa deben ser pequeñas y numerosas, (c) las partículas de la fase dispersa deben ser redondas y no aciculares y (d) mayores cantidades de de la fase dispersa aumentan el endurecimiento. [Askeland y Phulé, 2004] *14*

Figura 2-3 Deformación localizada de un material dúctil durante un ensayo de tensión; se produce una región de cuello. En la fotografía se observa la parte del cuello en una muestra fracturada. [Askeland y Phulé, 2004] *16*

Figura 3-1 Representación de un espacio de soluciones de una estructura de vecindad. *21*

Figura 3-2 Algoritmo general de una búsqueda por vecindad *22*

Figura 3-3 Doble Aleatorio a una solución factible a partir de dos posiciones elegidas en forma aleatoria. Opción 1. ..... *23*

Figura 3-4 Doble Aleatorio a una solución factible a partir de dos posiciones elegidas en forma aleatoria. Opción 2. ..... *24*

Figura 3-5 Triple Aleatorio a una solución factible a partir de tres posiciones elegidas en forma aleatoria. Opción 1. ..... *25*

Figura 3-6 Triple Aleatorio a una solución factible a partir de tres posiciones elegidas en forma aleatoria. Opción 2. ..... *26*

Figura 3-7 Cuádruple Aleatorio a una solución factible a partir de cuatro posiciones elegidas en forma aleatoria. Opción 1. ..... *27*

Figura 3-8 Cuádruple Aleatorio a una solución factible a partir de cuatro posiciones elegidas en forma aleatoria. Opción 2. ..... *27*

Figura 3-9 Quíntuple Aleatorio a una solución factible a partir de cinco posiciones elegidas en forma aleatoria. Opción 1 ..... *28*

Figura 3-10 Quíntuple Aleatorio a una solución factible a partir de cinco posiciones elegidas en forma aleatoria. Opción 2. ..... *29*

Figura 3-11 Para realizar un cambio, el algoritmo seleccionará en forma aleatoria la estructura que aplicará. *30*

Figura 3-12 Diagrama de flujo de la Estructura Híbrida Aleatoria *31*

Figura 3-13 Representación de la Solución inicial y Nueva Solución con Búsquedas Locales (local e iterada). *33*

Figura 3-14 Algoritmo de Búsqueda Local Iterada [Cruz-Chávez, M. A, 2005]. *33*

Figura 3-15 Algoritmo de Recocido Simulado *34*

Figura 3-16 Vector (estructura) con la composición del acero para obtener la primera solución. *37*

Figura 3-17 Nombres de los elementos que forman la composición del acero. *37*

Figura 3-18 El Fe es el único elemento al cual se le podrá aumentar o disminuir su composición para balancear el acero. *38*

Figura 3-19 Rango (peso en %) de cada uno de los elementos que componen el acero. *38*

Figura 3-20 Vector con la primera solución generada y con las composiciones arriba del 100%. *38*

Figura 3-21 Vector con la primera solución generada y al 100% (balanceada). *38*

Figura 3-22 Mejor solución encontrada al final de la ejecución del programa con el Algoritmo de Recocido Simulado.....39

Figura 4-1 Clúster ciicap ubicado en el CIICAp de la UAEM.....48

Figura 4-2 Clúster NOPAL ubicado en el ITVer.....48

Figura 4-3 Gráfica con los resultados comparativos de cada una de las estructuras de vecindad.La mejor estructura es la Híbrida Aleatoria.....50

Figura 4-4 (a)Gráfica con los resultados de las ejecuciones de cada una de las cinco estructuras de vecindad.....52

Figura 4-5 Gráfica con los resultados de las ejecuciones de cada una de las cinco estructuras de vecindad (b).....53

Figura 4-6 Gráfica con los resultados comparativos de la estructura de vecindad Híbrido Aleatorio evaluadas desde un 5% hasta un 100%.....54

Figura 4-7 (a)Gráficas con los resultados de las ejecuciones de cada una de las pruebas realizadas a la estructura de vecindad Híbrido Aleatorio las cuales van desde el 5% hasta un 100%.....55

Figura 4-8 (b)Gráficas con los resultados de las ejecuciones de cada una de las pruebas realizadas a la estructura de vecindad Híbrido Aleatorio las cuales van desde el 5% hasta un 100%.....56

Figura 4-9 (c)Gráficas con los resultados de las ejecuciones de cada una de las pruebas realizadas a la estructura de vecindad Híbrido Aleatorio las cuales van desde el 5% hasta un 100%.....57

Figura 4-10 Gráfica con la ejecución de la estructura de vecindad Híbrida en el algoritmo de búsqueda iterada realizada en el clúster ciicap.....62

Figura 4-11 Gráfica con el comportamiento del Algoritmo de Recocido Simulado y la estructura de vecindad Híbrida.....63

Figura A-1 Gráfica con la función temporal de la estructura Doble Aleatorio.74

Figura A-2 Gráfica con la función temporal de la estructura Triple Aleatorio.75

Figura A-3 Gráfica con la función temporal de la estructura Cuádruple Aleatorio.....77

Figura A-4 Gráfica con la función temporal de la estructura Quíntuple Aleatorio......*79*

Figura A-5 Gráfica con la Complejidad del Algoritmo de Recocido Simulado. *80*

Índice de Tablas

Tabla 2-1 Rango de concentraciones de todos los componentes empleados en la optimización (% en peso) [Xu et al., 2009]. 12

Tabla 3-1 Instancia que presenta una solución inicial..... 43

Tabla 4-1 Resultados de la evaluación de cada una de las estructura de 50

Tabla 4-2 Resultados del Híbrido Aleatorio con pruebas desde el 5% al 100% 53

Tabla 4-3 Rangos utilizados para realizar el análisis de sensibilidad al Algoritmo de Recocido Simulado. 59

Tabla 4-4 Valor de la Longitud de la Cadena de Markov (LCM) con los incrementos y decrementos utilizados para los rangos de los parámetros de control utilizados para el análisis de sensibilidad. 60

Tabla 4-5 Valores de los parámetros de control sintonizados para el Algoritmo de Recocido Simulado de acuerdo al análisis de sensibilidad. 61

Capítulo 1

Introducción

Cualquier tipo de industria buscará siempre la forma de reducir los costos de operación, obviamente sin bajar su calidad, mediante métodos que les brinden resultados rápidamente. En algunos casos, realizar experimentos con materiales suelen demorarse y elevar su costo. Tal es el caso del acero, el cual ha estado mejorando con el paso del tiempo gracias a las técnicas que han empleado para su fortalecimiento.

Actualmente, se han estado realizando pruebas para maximizar la resistencia en aceros microaleados con modelos que pueden ser examinados por *la optimización combinatoria* quedando en espera de obtener buenos resultados [Xu y Rivera-Díaz-del-Castillo, 2008]. El objetivo principal de cualquier problema es encontrar la "mejor" solución en un conjunto de soluciones que les permita alcanzar dicho objetivo [Papadimitriou y Steiglitz, 1998]. Para los problemas de optimización combinatoria es necesario representarlos mediante modelos matemáticos para adaptar las variables del problema a dicho modelo. La función objetivo, porta el requisito principal para un problema de optimización como lo es el de maximizar o minimizar.

En este trabajo de investigación se propone hacer uso de un método de búsqueda local, como lo es el Algoritmo de Recocido Simulado, para modificar la composición química del material y así poder maximizar la resistencia en aceros microaleados tomando en cuenta únicamente la composición química del material. Dicho algoritmo está considerado como una metaheurística, la cual pertenece a los métodos de optimización (Figura

1-1), permitiéndole crear y trabajar con algoritmos híbridos [Osman y Kelly, 1996].

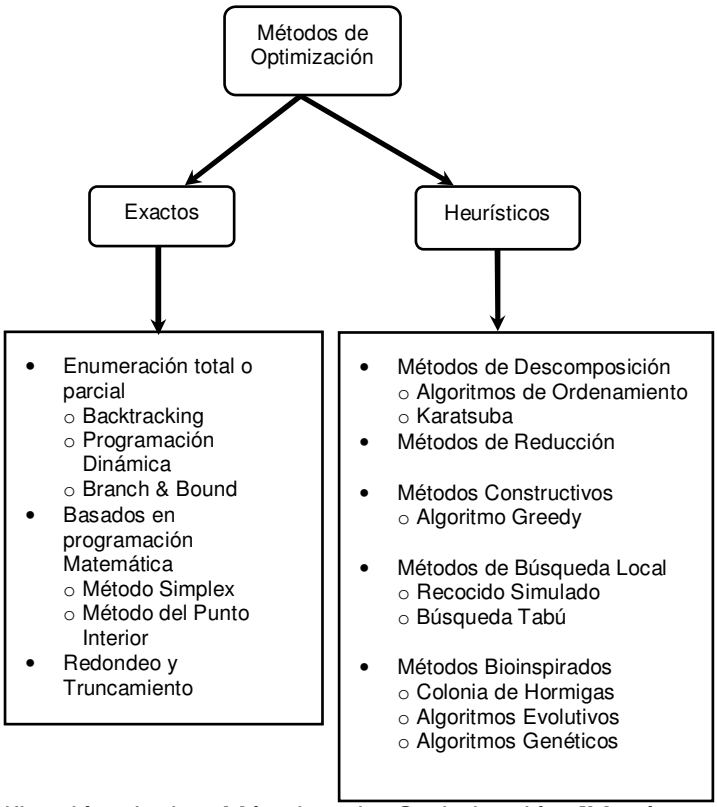


Figura 1-1 *Clasificación de los Métodos de Optimización [Martínez- Oropeza,A., 2010].*

1.1 Teoría de la Complejidad de los Algoritmos

A pesar de que las computadoras evolucionan rápidamente y de que son capaces de procesar una cantidad bastante grande de operaciones por segundo es posible encontrar problemas que pueden tardar años en terminar. Conocer el tiempo exacto que tardará el programa en arrojar resultados se vuelve complejo y en lugar de calcular el tiempo que tardará en procesarse, se calcula la cantidad de operaciones de acuerdo al tamaño que tiene de entrada.

El tiempo y los recursos computacionales que ocupará el algoritmo para resolver el problema de investigación son dos de las cosas que abarca el estudio de la complejidad de un algoritmo. Conocer su eficiencia es algo que requiere tiempo, ya que al adaptarlo al problema es necesario probarlo y si este consume mucha cantidad de memoria se considera no apto pues tiene una complejidad conocida como espacial y si a esto le agregamos que, le toma mucho tiempo arrojar resultados entonces su grado de complejidad aumenta; es decir, tiene una complejidad temporal. De ahí que, un algoritmo necesita dos recursos para resolver un problema: tiempo y espacio [Talbi E., 2009].

La complejidad de un problema equivale a la complejidad del mejor algoritmo para resolverlo. Se dice que un problema es fácil si existe un algoritmo Polinómico que lo resuelva, de lo contrario, no tiene solución. La Teoría de la Complejidad de Problemas trata con las decisiones de dichos problemas. La respuesta para un problema de decisión es siempre sí o no [Talbi, E., 2009].

La clasificación de los problemas representa el conjunto de todos aquellos problemas que pueden resolverse con una determinada cantidad de recursos computacionales, de las cuales se conocen dos clases importantes: P y NP (Figura. 1-3).

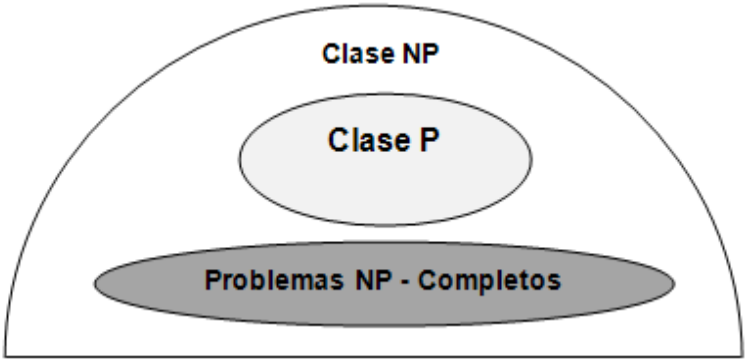


Figura 1-2 *Clases de Complejidad para la decisión de Problemas [Talbi,E., 2009].*

La clase P representa al conjunto de todos los problemas de decisión que pueden ser resueltos por una máquina determinista en tiempo polinómico. Por ende, se dice que los problemas que pertenecen a esta clase son relativamente "fáciles" de resolver. Algunos problemas son: el árbol de expansión mínima, la ruta más corta, la red de flujo máximo, emparejamiento máximo bipartito y el modelo de programación lineal continúo.

La clase NP representa a los problemas que pueden ser resueltos por un algoritmo no determinista en tiempo polinómico. Utiliza las primitivas: *elección,* propone una solución (oráculo), *comprobar*, verifica en tiempo polinomial si una propuesta de solución (certificado) da una respuesta positiva o negativa; *éxito,* cuando el algoritmo responde que sí después de la solicitud de verificación, y *falla,* cuando el algoritmo no responde "sí." Entonces, si la primitiva *elección* propone una solución que da un "sí" y el oráculo tiene la capacidad de hacerlo, entonces la complejidad computacional es polinómica.

Este trabajo de investigación utiliza el Algoritmo de Recocido Simulado y hace uso de la configuración de un acero (está catalogado como un problema NP debido a que puede ser resuelto). Antes de aplicar el Algoritmo de Recocido Simulado, se realizaron pruebas experimentales para determinar, en conjunto con la búsqueda local y local iterada, cuál de las estructuras de vecindad era la que se integraría al Algoritmo de Recocido Simulado. Para la sintonización del Algoritmo de Recocido Simulado fue necesario realizar un análisis de sensibilidad de los parámetros de control del algoritmo. La metodología de sintonización utilizada está basada en las tesis de doctorado de [Cruz-Chávez, 2005] y [Martínez-Oropeza, 2010].

1.2 Estado del Arte

Mejorar la resistencia del acero no ha sido tarea fácil. Muchos de los esfuerzos por desarrollar aceros de alta resistencia utilizando diferentes métodos ha sido llevado a cabo tanto en forma académica como industrial por lo que, el acero ideal es aquel que ofrece las mismas propiedades mecánicas que posee pero sin necesidad de que se le aplique una cubierta protectora contra la corrosión, ya que esto elevaría su costo.

Las investigaciones académicas realizadas por [Xu y Rivera-Díaz-del-Castillo, 2008a] los han llevado al uso de algoritmos de optimización para mejorar la resistencia del acero. Una vez que definieron los diferentes aspectos de una microestructura deseable y formularon los criterios de los parámetros a cumplir, diseñaron la composición de la aleación. El algoritmo de optimización empleado fue el Algoritmo Genético.

Los Algoritmos Genéticos están inspirados en la evolución biológica haciendo evolucionar a individuos de cierta población (problema) sometiéndolos a acciones aleatorias como el cruzamiento y la mutación sin olvidar que dentro de la evolución está la selección, que es donde se toman a los individuos más adaptados, los que sobreviven, de los menos adaptados, los que se eliminan. Selección, mutación y cruzamiento son los operadores probabilísticos que controlan la evolución heurística. Los Algoritmos Genéticos trabajan con las soluciones candidatas representadas con 1's y 0's llamados cromosomas.

[Xu y Rivera-Díaz-del-Castillo, 2008b] continúan innovando con el uso de este algoritmo pues al parecer se ha convertido en una herramienta que les brinda buenos resultados al combinarla con la termodinámica y con la cinética.

En su trabajo presentado en 2009 [Xu y Rivera-Díaz-del-Castillo, 2009] incorporan el uso de las redes neuronales artificiales para trabajar en conjunto con los Algoritmos Genéticos. En 2010 [Xu y Rivera-Díaz-del-Castillo 2010] vuelven a presentar un trabajo haciendo uso del Algoritmo Genético aplicado a los aceros de Ultra Alta Resistencia (Ultra High Strength - UHS) desde la termodinámica a la mecánica cuántica.

Con el presente trabajo de investigación, se pretende incursionar en la misma área de materiales para mejorar la resistencia de un acero pero con una diferente metaheurística, el Algoritmo de Recocido Simulado. En este trabajo, se hará uso únicamente de la composición química de un acero aplicándole la ecuación que será la Función Objetivo, esperando proporcione una mejor solución a las actualmente encontradas.

1.3 Objetivo de la Investigación

- Aplicar el Algoritmo de Recocido Simulado para encontrar la mejor configuración de elementos de un acero microaleado que ofrezca una mejor resistencia mecánica.

Objetivos Particulares

- Realizar pruebas con el Algoritmo de Recocido Simulado para verificar los resultados y poder comparar la resistencia obtenida en Mega Pascales.
- Evaluar una estructura de vecindad que mejore el desempeño del Algoritmo de Recocido Simulado.

- Optimizar de forma teórica la composición del acero microaleado, haciendo uso del Algoritmo de Recocido Simulado.
- Evaluar la eficiencia y complejidad del Algoritmo de Recocido Simulado.
- Sintonizar los parámetros del Algoritmo de Recocido Simulado mediante un análisis de sensibilidad.

1.4 Alcance de la Investigación

En la presente investigación, no se hará uso del tamaño de grano y de la cantidad de precipitados, solamente de la composición química del material. También, se definirán los límites máximos y mínimos para cada elemento y los elementos químicos que podrán variar en la composición.

1.5 Contribución de la Tesis

En esta investigación, se aplicó el Algoritmo de Recocido Simulado para modificar la composición química del material con el objetivo de Maximizar la Resistencia Mecánica en Aceros Microaleados, haciendo uso de una estructura de vecindad híbrida para modificar la composición química de los elementos que intervienen en el material.

Una característica del Algoritmo de Recocido Simulado es que aplica una estructura de vecindad Híbrida, la cual se comprueba y demuestra, a través de resultados experimentales, que es mucho mejor que las otras estructuras propuestas, por lo mismo, la eficacia del Recocido Simulado mejora. La estructura de vecindad se puede integrar a cualquier tipo de metaheurística para este tipo de problema y de acuerdo a los resultados experimentales, se

espera que este tipo de estructura de vecindad híbrida mejore la eficacia de estas metaheurísticas.

1.6 Organización de la Tesis

En el *capítulo uno* se presenta una introducción del trabajo de investigación realizado así como los trabajos escritos hasta el momento por otros investigadores especializados en aceros y en el estado del arte. También se presentan los objetivos para desarrollar este tema de investigación al igual que los alcances obtenidos como la contribución que aporta este trabajo.

En el *capítulo dos* se encuentra el desarrollo del problema de investigación así como sus variables. En el *capítulo tres* se da una explicación de lo que es el Algoritmo de Recocido Simulado, su comportamiento, las pruebas generadas para verificar la confiabilidad del algoritmo, la metodología empleada y el análisis de complejidad.

En el *capítulo cuatro* se presenta una búsqueda local iterada para la evaluación de las estructuras de vecindad. Posteriormente, se muestran los resultados del Algoritmo de Recocido Simulado adaptando la estructura que obtuvo mejor desempeño en las pruebas de búsqueda local iterada. Una vez adaptada la estructura al algoritmo, se genera el análisis de sensibilidad y las pruebas para examinar los resultados y emitir una opinión.

En el *capítulo cinco* se encuentran las conclusiones al igual que los posibles trabajos futuros. Finalmente se encuentran las referencias y los apéndices.

Quien siempre dice la verdad,
puede permitirse tener
mala memoria
Theodor Heuss

Capítulo 2

Aceros Microaleados

Introducción

EL uso del acero va mejorando con el paso del tiempo, especialmente aquel que ofrece la mejor resistencia a la corrosión gracias a los elementos que contiene y que le dan esa característica. La microestructura de los aceros es muy compleja y ésta es el resultado de varios tratamientos térmicos por los que el material ha pasado hasta obtener la microestructura final deseada. Un descubrimiento que permitió entender mucho mejor a la metalurgia física de las microaleaciones del acero fue el realizado por Petch [Morrison, 2000]. Este descubrimiento indicó la relación cuantitativa entre el tamaño de grano, la resistencia y las propiedades de fractura del acero. Con las primeras investigaciones de Petch, fue posible revelar que era principalmente la formación de finos carburos/nitruros los que proporcionaban el refinamiento del grano y el fortalecimiento de la precipitación. Gracias a este descubrimiento, la resistencia en el acero se convirtió en tema de estudio.

Los aceros microaleados de alta resistencia y baja aleación ó aceros HSLA (High Strength Low Alloy) se han posicionado actualmente como una clase importante de materiales estructurales de alta resistencia debido a que ha tenido una mayor expansión en su desarrollo y producción. El desarrollo de estos aceros incluye, su diseño de aleación, procesamiento y aplicaciones, por lo que esta investigación cubre las últimas cuatro décadas, haciéndose indispensables para aplicaciones estructurales. Su habilidad para obtener propiedades mecánicas finales requeridas en aplicaciones de ingeniería únicamente mediante la laminación controlada en caliente y enfriamiento

acelerado, elimina la necesidad de aplicarles costosos tratamientos térmicos posteriores. Se pueden obtener esfuerzos de cedencia entre 550 y 660 Mega Pascales a través de una adición menor al 0.1% de ciertos elementos considerados microaleantes tales como el Nb, V, y Ti que son fuertes formadores de micro-partículas (precipitados) de carbonitruros a bajo costo. Estos elementos incrementan considerablemente la resistencia del acero y son una característica de la mayoría de los aceros HSLA. Debido a su producción económica, se han vuelto atractivos para reemplazar aceros tratados térmicamente en aplicaciones tales como: camiones, rieles de trenes, estructuras para grúas incluso para la industria aeronáutica y automotriz, etc. Actualmente, se han desarrollado mejoras en la fundición del acero y su laminación reduciendo aun más su costo, por lo que han aumentado su competitividad.

En la actualidad, la termodinámica computacional se ha convertido en una herramienta poderosa para el diseño y desarrollo de las aleaciones, las cuales, se realizan en una computadora haciendo uso de criterios de optimización basados en los principios termodinámicos, cinéticos y mecánicos**.**

Algunos estudios realizados, para obtener una resistencia cercana a la óptima, se han apoyado en los métodos de optimización. Tal es el caso de los estudios de investigación realizados por el Dr. Rivera Díaz del Castillo [Xu et al., 2008] y [Xu et al., 2009] donde hace uso del Algoritmo Genético. Esta investigación se basa en una *composición química* donde el acero contiene 13 aleaciones consideradas: C, Cr, Ni, Ti, Mo, Al, Cu, Co, Nb, N, V, Mn y Si. La concentración asignada a cada uno de estos elementos está dentro de su rango y se basan en las restricciones industriales y tecnológicas relacionadas con el precio y la habilidad para fabricarlas (Tabla 2-1).

Tabla 2-1 *Rango de concentraciones de todos los componentes empleados en la optimización (% en peso) [Xu et al., 2009].*

	C	Cr	Ni	Ti	Mo	Al	Cu	Co	Nb	N	V	Mn	Si	Fe
Min	0.05	12.00	1.00	0.01	0.50	0.01	0.50	0.01	0.01	0.01	0.01	0.50	0.30	Bat.
Máx	0.20	20.00	15.00	1.50	10.00	1.00	10.00	2.00	0.10	0.01	0.20	0.50	1.00	

Factores como la composición química, el tamaño de grano, la cantidad de precipitados, el algoritmo genético y la optimización dieron pie para generar un nuevo proyecto y comparar la eficacia, eficiencia e importancia de hacer uso de un algoritmo computacional diferente a los usados actualmente. El Algoritmo de Recocido Simulado es una técnica de búsqueda local estocástica y será el actor principal para el desarrollo de este nuevo proyecto haciendo uso únicamente de la *composición química* (variable única dentro del Algoritmo de Recocido Simulado) del acero microaleado pero, para hacer uso del Algoritmo de Recocido Simulado se obtendrá, mediante el uso de la Búsqueda Local (LS - Local Search, por sus siglas en inglés), la primera solución para ser evaluada con la Búsqueda Local Iterada (ILS -Iterated Local Search, por sus siglas en inglés) que hará uso de una Estructura de Vecindad y determinará, cual de esas EV será la adecuada para implementar el Algoritmo de Recocido Simulado.

Debido a la dureza que presentan algunos problemas se ha optado por emplear técnicas heurísticas de búsqueda por vecindad, las cuáles han demostrado ser métodos muy eficientes en la búsqueda de soluciones aproximadas para dichos problemas. El tamaño y la estructura son la parte medular de una vecindad [Michaelewicz, 2004]. Si el tamaño de una vecindad es mayor, mejor será la calidad de las soluciones localmente óptimas al igual que la precisión de la solución final. Sin embargo, el tiempo que se requiere para realizar una iteración dentro de la vecindad se verá incrementado con respecto al tamaño de la vecindad, por lo que, se debe

emplear un tamaño adecuado que permita generar un mejor desempeño de la función de vecindad a emplear [Cruz-Chávez y Martínez, 2010].

2.1 Tamaño de Grano

Los límites de grano son defectos superficiales que separan un material en regiones; cada región tiene la misma estructura cristalina pero distinta orientación. La microestructura de muchos materiales cerámicos y metálicos consiste en muchos granos. Un grano es una porción del material dentro de la cual el arreglo de los átomos es casi idéntico. Sin embargo, la orientación del arreglo de átomos, o estructura cristalina, es distinta en cada grano vecino [Askeland y Phulé, 2004].

Un método para controlar las propiedades de un material es controlar el tamaño del grano. Al reducir el tamaño de grano, se aumenta la cantidad de granos y, en consecuencia, se aumenta la cantidad de superficie de límites de grano. Toda dislocación recorre solamente una distancia corta para encontrar un límite de grano y detenerse; así, la resistencia del material metálico aumenta.

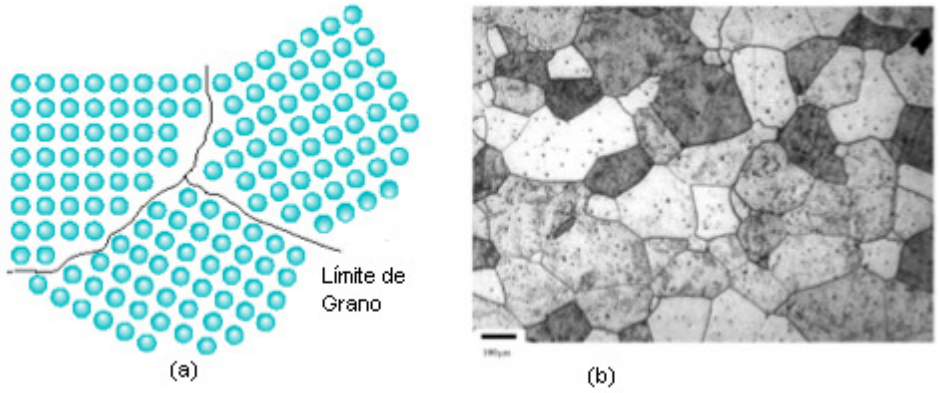


Figura 2-1*(a) Los átomos cercanos a los límites de los tres granos no tienen una distancia o un arreglo en equilibrio. (b) Granos y límites de grano en una muestra de acero inoxidable. (Cortesía del Dr. A. Deardo.) [Askeland y Phulé, 2004].*

2.2 Precipitados

La mayoría de los materiales diseñados están formados por más de una fase, y muchos de ellos están diseñados para tener mejor resistencia. En las aleaciones endurecidas por dispersión simples, se introducen partículas diminutas de una fase, en general muy fuerte y dura, en una segunda fase, que es más débil pero más dúctil. La fase endurecedora se llama **fase dispersa** o **precipitados** [Askeland y Phulé, 2004].

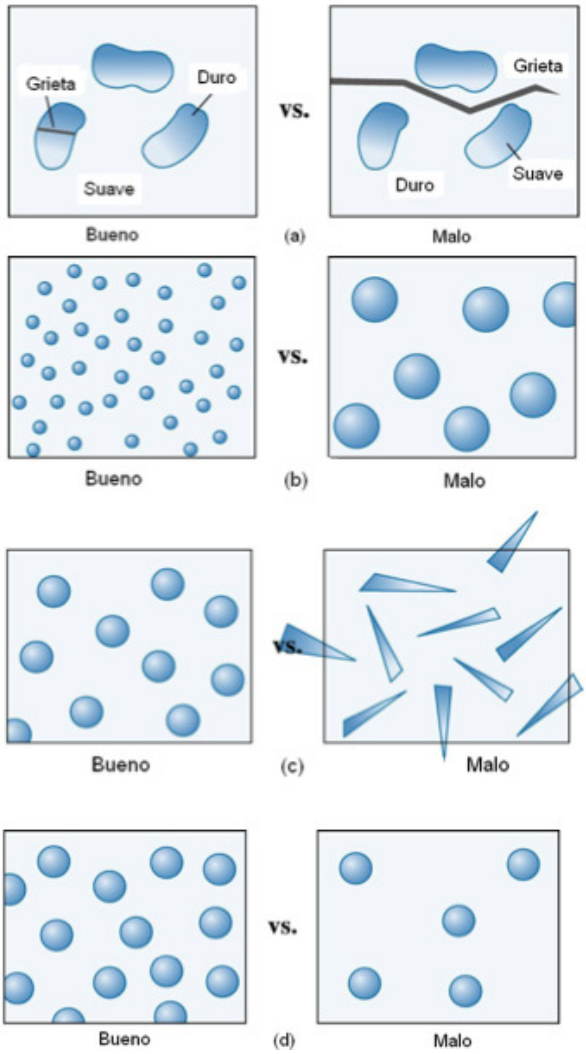


Figura 2-2 *Consideraciones para tener un endurecimiento efectivo por dispersión: (a) La fase precipitada debe ser dura y discontinua, (b) las partículas de la fase dispersa deben ser pequeñas y numerosas, (c) las partículas de la fase dispersa deben ser redondas y no*

aciculares y (d) mayores cantidades de de la fase dispersa aumentan el endurecimiento. [Askeland y Phulé, 2004].

Para que haya endurecimiento por dispersión, la fase dispersa o precipitado debe ser lo suficientemente pequeña para proporcionar obstáculos eficaces al movimiento de las dislocaciones y favorecer el mecanismo de endurecimiento.

Algunas de las consideraciones son:

- Las partículas de la fase dispersa deben ser pequeñas y numerosas para aumentar la probabilidad de inferir con el proceso de deslizamiento, ya que el área de la interfaz entre fases aumenta en forma importante.
- Las partículas de la fase dispersa deben ser redondas, y no aciculares o con aristas agudas, porque es menos probable que la forma redonda inicie una grieta o que actúe como una muesca.
- Mayores concentraciones de la fase dispersa aumentan la resistencia de una aleación.

2.3 Resistencia Mecánica

Las propiedades mecánicas de los materiales dependen de su composición y su microestructura. La composición, naturaleza de los enlaces, estructura cristalina y defectos, como dislocaciones, tamaño de grano, etc., de un material tienen una influencia profunda sobre la resistencia y ductilidad de lo materiales metálicos.

En muchas de las tecnologías emergentes en la actualidad, se hace hincapié en las propiedades mecánicas de los materiales que se usan. Ejemplo: En la fabricación de aviones, sus componentes deben ser ligeros y resistentes. En la siguiente figura 2-4 se muestra una imagen respecto a la prueba de resistencia por las que pasa un acero.

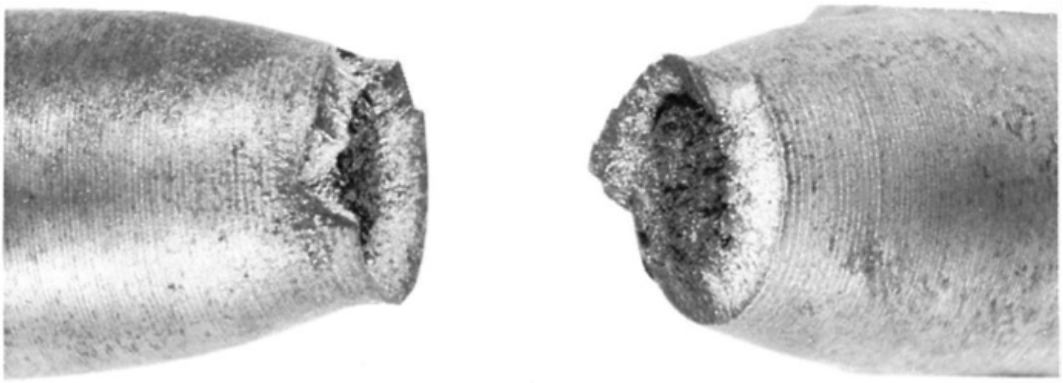


Figura 2-3 *Deformación localizada de un material dúctil durante un ensayo de tensión; se produce una región de cuello. En la fotografía se observa la parte del cuello en una muestra fracturada. [Askeland y Phulé, 2004].*

Hay distintas clases de fuerzas o "esfuerzos" que se presentan al tratar las propiedades mecánicas de los materiales. El **esfuerzo** es la fuerza que actúa sobre el área unitaria en la que se aplica y se expresa en Pa (Pascales) o en psi (libras por pulgadas cuadradas, por sus siglas en inglés). La **deformación unitaria** es el cambio de dimensión por unidad de longitud, no tiene dimensiones y se expresa en pulg / pulg o en cm / cm. Al describir el esfuerzo y la deformación unitaria, nos damos cuenta que el esfuerzo es la causa y la deformación unitaria es el efecto.

El ensayo de tensión mide la resistencia de un material a una fuerza elástica. Las propiedades que se obtienen del ensayo de tensión pueden aplicarse en el diseño de distintos componentes [Askeland y Phulé, 2004].

Para este problema de esta investigación, como parte de la función objetivo, únicamente se tomó en cuenta la composición química del acero y la fórmula para medir la resistencia propuesta por [Steve, 1956] (Ecuación 2-1). Las demás variables, tamaño de grano y precipitados, serán tomados en cuenta para una nueva función objetivo.

La fórmula que se muestra a continuación, devuelve la cedencia con las cantidades de los elementos de la composición química. Esta fórmula es la función objetivo que mostrará el valor máximo obtenido por el Algoritmo de Recocido Simulado [Steve, 1956].

$$\sigma_y = 170 + 1300(wt.\%C) + 160(wt.\%Mn) + 160(wt.\%Cr) + 130(wt.\%Mo) + 88(wt.\%Ni) + 45(wt.\%Cu) + 270(wt.\%V)$$

Ecuación 2-1

La condición que se tiene que cumplir en este trabajo de investigación, es maximizar; es decir, obtener el mejor resultado. Normalmente, la función del Algoritmo de Recocido Simulado se utiliza para minimizar, pero para este problema, se ocupará para realizar lo contrario. La forma de representar la función objetivo se muestra en el capítulo siguiente.

No hay verdad que al
nacer no haya sido perseguida
Voltaire

Capítulo 3

Algoritmo de Recocido Simulado

3.1 Introducción

El Algoritmo de Recocido Simulado es un método de optimización que simula el proceso físico del recocido de sólidos. Es este mismo proceso el que se simula computacionalmente para obtener buenas soluciones de acuerdo al problema planteado. Para mejorar el desempeño del Algoritmo de Recocido Simulado se propone la evaluación de cinco estructuras de vecindad. En el presente capítulo se presentan las cinco estructuras de vecindad incluyendo el procedimiento utilizado para desarrollar la estructura de vecindad híbrida. Se presenta el Algoritmo de Recocido Simulado y se define la Función Objetivo que utiliza el algoritmo para el problema de maximizar la resistencia mecánica de aceros microaleados.

Las estructuras de vecindad están adaptadas de acuerdo a la necesidad del problema. Su desarrollo se basó en las estructuras presentadas por [Cruz-Chávez, M. A, Martínez-Oropeza, A., 2010] en su artículo *Estructura Híbrida de Vecindad para Problemas de Optimización Discreta*. Las estructuras de vecindad son consideradas técnicas para mejorar una solución, el único requisito que tienen es que se debe avanzar poco a poco a través de una vecindad desde una solución inicial hasta lograr encontrar el valor que cumpla con las expectativas de la Función Objetivo.

3.2 Algoritmo de Recocido Simulado

El recocido de sólidos es un proceso físico que consiste en incrementar la temperatura al valor máximo en donde todas las partículas del sólido se ordenan aleatoriamente en la fase líquida seguida de un enfriamiento muy lento.

El comportamiento que sigue es: Máxima temperatura en calentamiento, Reducir lentamente mediante enfriamiento [Laarhoven et al., 1992].

El Algoritmo de Recocido Simulado es una metaheurística basado en el recocido de sólidos y fue propuesta por primera vez por [Kirkpatrick et al., 1983]. Esta metaheurística es una técnica de búsqueda local estocástica que aproxima el valor mínimo de la función de costo $f:S \rightarrow R$ sobre un conjunto finito de soluciones S [Martínez-Rangel M. G., 2008].

El Algoritmo de Recocido Simulado fue originalmente diseñado para minimizar el costo de la función objetivo. Para este trabajo de investigación, se requiere hacer lo contrario para lo que fue creado, esto es, el problema de la resistencia mecánica de aceros microaleados requiere de maximizar esta resistencia. El Algoritmo de Recocido Simulado está considerado como un algoritmo de búsqueda local inteligente y se diferencia de la búsqueda local iterada por la forma en que utiliza el criterio de aceptación de nuevas soluciones. La búsqueda local iterada sólo aceptará buenas soluciones; sin embargo, el Algoritmo de Recocido Simulado inicia aceptando todas las soluciones, sean estas buenas o malas, para después ir aceptando soluciones buenas y rara o esporádicamente alguna solución mala. Esto se debe gracias a que cuenta con la función de Boltzmann, que al tomar cualquier solución, las evalúa aceptando las soluciones que cumplan cierta probabilidad de aceptación.

3.3 Estructura de Vecindad Híbrida

Las técnicas aplicadas a búsqueda local iterada mejor conocida como estructuras de vecindad, tienen un comportamiento en forma iterativa, esto con la finalidad de que permita, al algoritmo, realizar una buena explotación del espacio de soluciones [Martínez-Oropeza, A., 2010]. Los métodos heurísticos, en algunos problemas considerados como intratables, han implementado las estructuras de vecindad para optimizar a través de una búsqueda local de una manera rápida la función objetivo de un problema.

Para llegar a obtener una mejor solución, es importante moverse dentro de una vecindad, partiendo desde la solución inicial hacia una solución vecina, misma que deberá ser evaluada por la Función Objetivo (maximizar), hasta encontrar una solución que mejore la solución anterior.

La forma en la cual se determina el espacio de soluciones será a través del problema con el cuál se esté tratando, pero, para decidir ese espacio de soluciones se debe conocer y entender, primero, el problema [Joyanes y Zahonero, 2000]. Es así como podremos definir el espacio de soluciones que tendrá un vecindario, sin olvidar cuál será nuestro objetivo: *Maximizar*.

Una *Vecindad* es el conjunto de soluciones las cuales se pueden alcanzar a partir de una solución ***s*** por medio de un movimiento σ/**s** [Papadimitriou y Steiglitz, 1998] y [Martínez, 2006], que puede ser un intercambio, inserción o eliminación entre los elementos de una solución *s*. Basándonos en el concepto anterior, tomemos la primera solución como un punto, al cual representaremos con ***s***, todas las soluciones que estén cerca de ese punto definidas por una estructura de vecindad serán consideradas el *vecindario* representado por ***N(s)***. Nuestro espacio de soluciones queda representado por ***S*** [Moreno, A. 2008] (Figura 3-1).

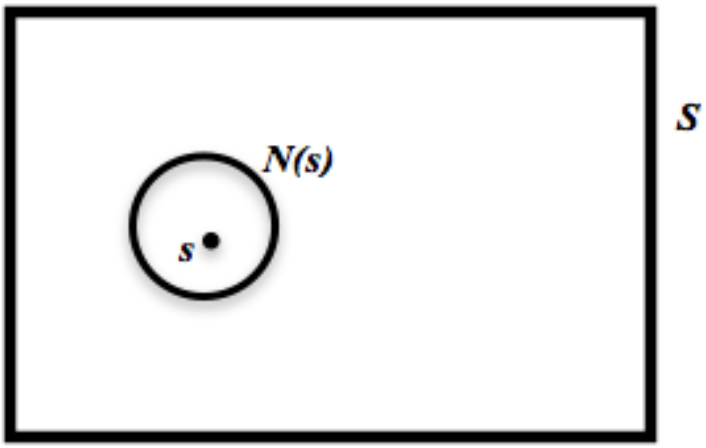


Figura 3-1 *Representación de un espacio de soluciones de una estructura de vecindad.*

Matemáticamente, tenemos que $s \in S$, el vecindario N(**s**) de **s** se define como un conjunto de posibles soluciones cerca de **s**, representado por la función $N(s) = \{s' \in S : s \xrightarrow{\sigma} s'\}$. Si a partir de este punto generamos otro que mejore el primero, éste será representado por ***s'***. De esta manera, nos iremos moviendo paso a paso desde una solución inicial hacia una solución que mejore la anterior, esto es, *f(s')>=f(s)* donde *f(s')* y *f(s)* son funciones de costo ***s'*** y ***s*** respectivamente, y sea la seleccionada que de una máxima resistencia de acuerdo a la Función Objetivo.

Para elegir y determinar la estructura de vecindad que mejor soluciones de al problema se debe definir el movimiento que permita alcanzar una solución *s'* desde una solución *s,* siempre y cuando cumpla con el criterio de selección, el proceso se repetirá hasta que pasado un determinado número de ciclos, la solución encontrada no pueda ser mejorada, si este es el caso, significa que se ha llegado a un *óptimo local* (Figura 3-2).

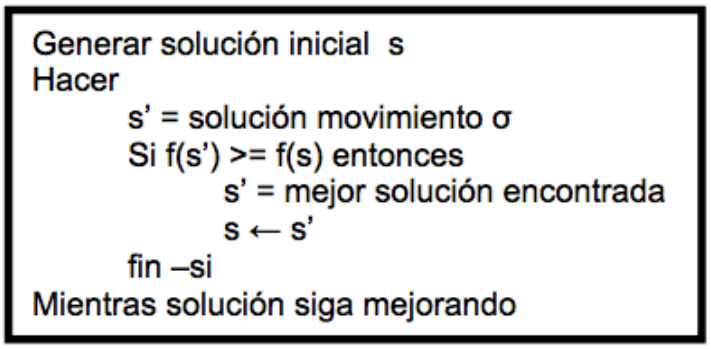


Figura 3-2 *Algoritmo general de una búsqueda por vecindad.*

Se describe, a continuación, el funcionamiento de cada una de las estructuras de vecindad creadas para este problema de investigación. Pero antes, se explicará el procedimiento para realizar los movimientos dentro del espacio de soluciones, los pasos que se siguen son:

1.- Determinar el tamaño del vecindario, tomando el incremento (Δ) y el decremento (-Δ) que se le puede hacer a siete de los elementos de la composición del acero microaleado. El Fe, que es el elemento con el porcentaje mayor, contiene un rango sobre el cual puede realizarse la modificación. El límite superior es de **98.9** con un límite inferior de **0**. El límite superior será dividido entre **0.0001** (cantidad que afecta en lo mínimo la composición de los elementos). Al realizar esta operación, obtenemos el tamaño de la vecindad: **989000**.

2.- Seleccionar en forma aleatoria a cuál de los elementos se les hará el incremento o decremento en composición porcentual. Hacer el cambio en composición.

- **Estructura de Vecindad Doble Aleatorio**

Para la estructura de vecindad Doble Aleatorio, se genera una solución ***s*** inicial factible a partir de la cuál se eligen dos números aleatorios *iposic1* e *iposic2*, mismos que corresponden a una posición del vector donde está almacenada la solución inicial y los cuales se eligen de la siguiente forma:

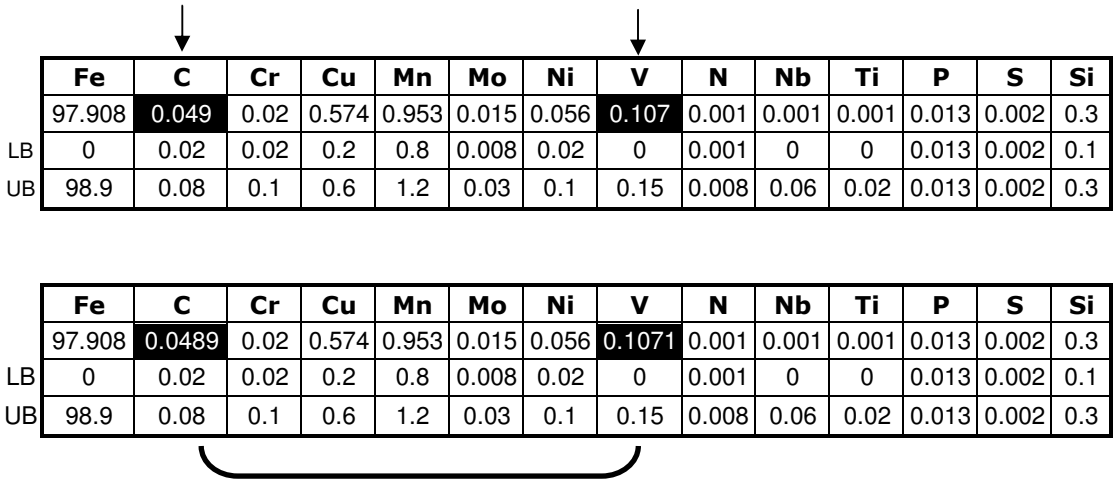
- El primer caso: incrementar el primer elemento (generado en iposic1) y decrementar el segundo elemento (iposic2) Figura 3-3.

- El segundo caso: decrementar el primer elemento (generado en iposic1) e incrementar el segundo elemento (iposic3) Figura 3-4.

	Fe	C	Cr	Cu	Mn	Mo	Ni	V	N	Nb	Ti	P	S	Si
	97.908	0.049	0.02	0.574	0.953	0.015	0.056	0.107	0.001	0.001	0.001	0.013	0.002	0.3
LB	0	0.02	0.02	0.2	0.8	0.008	0.02	0	0.001	0	0	0.013	0.002	0.1
UB	98.9	0.08	0.1	0.6	1.2	0.03	0.1	0.15	0.008	0.06	0.02	0.013	0.002	0.3

	Fe	C	Cr	Cu	Mn	Mo	Ni	V	N	Nb	Ti	P	S	Si
	97.908	0.0491	0.02	0.574	0.953	0.015	0.056	0.1069	0.001	0.001	0.001	0.013	0.002	0.3
LB	0	0.02	0.02	0.2	0.8	0.008	0.02	0	0.001	0	0	0.013	0.002	0.1
UB	98.9	0.08	0.1	0.6	1.2	0.03	0.1	0.15	0.008	0.06	0.02	0.013	0.002	0.3

Figura 3-3 *Doble Aleatorio a una solución factible a partir de dos posiciones elegidas en forma aleatoria. Opción 1.*



	Fe	C	Cr	Cu	Mn	Mo	Ni	V	N	Nb	Ti	P	S	Si
	97.908	0.049	0.02	0.574	0.953	0.015	0.056	0.107	0.001	0.001	0.001	0.013	0.002	0.3
LB	0	0.02	0.02	0.2	0.8	0.008	0.02	0	0.001	0	0	0.013	0.002	0.1
UB	98.9	0.08	0.1	0.6	1.2	0.03	0.1	0.15	0.008	0.06	0.02	0.013	0.002	0.3

	Fe	C	Cr	Cu	Mn	Mo	Ni	V	N	Nb	Ti	P	S	Si
	97.908	0.0489	0.02	0.574	0.953	0.015	0.056	0.1071	0.001	0.001	0.001	0.013	0.002	0.3
LB	0	0.02	0.02	0.2	0.8	0.008	0.02	0	0.001	0	0	0.013	0.002	0.1
UB	98.9	0.08	0.1	0.6	1.2	0.03	0.1	0.15	0.008	0.06	0.02	0.013	0.002	0.3

Figura 3-4 *Doble Aleatorio a una solución factible a partir de dos posiciones elegidas en forma aleatoria. Opción 2.*

Los números aleatorios generados serán validados ya que no a todos los componentes se les puede hacer el incremento/decremento en el porcentaje de su composición química. Los elementos que se verán afectados por el incremento (Δ) ó decremento (-Δ) van del *Fe* hasta el *V*, es decir, se hará este cambio siempre y cuando, las posiciones generadas aleatoriamente no sean las mismas, si es este el caso, se generan nuevamente hasta que cumplan con la condición. Una vez cumplida dicha condición, se obtiene una nueva solución vecinal ***s***'. El desarrollo de la complejidad del algoritmo se explicará en el apéndice B.

- **Estructura de Vecindad del Triple Aleatorio**

Para realizar una búsqueda por vecindad con un triple aleatorio, se lleva a cabo el mismo procedimiento explicado en la estructura anterior, generar una solución ***s*** inicial factible hasta obtener una nueva solución vecinal ***s***'; la diferencia en esta estructura es que aquí se generan tres números aleatorios:

iposic1, *iposic2* e *iposic3*. Estos números generados indicarán las posiciones en la estructura y también tienen el mismo requisito del doble aleatorio. Para la estructura de vecindad Triple Aleatorio se generan dos casos para el cambio del porcentaje de sus elementos seleccionados.

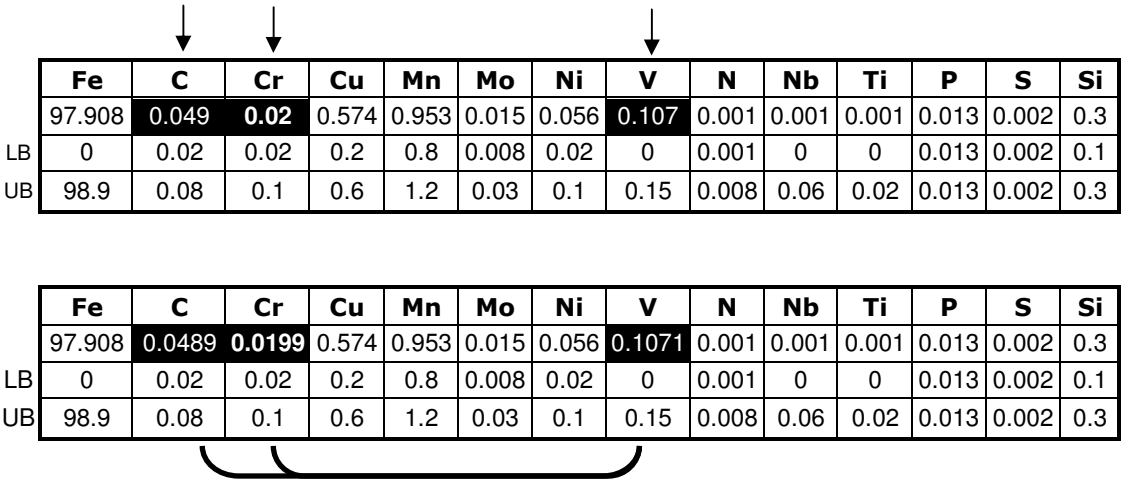
- El primer caso: incrementar dos elementos (generados en iposic1 e iposic2) y decrementar un elemento (iposic3) Figura 3-5.

- El segundo caso: decrementar dos elementos (generados en iposic1 e iposic2) e incrementar un elemento (iposic3) Figura 3-6.

	Fe	C	Cr	Cu	Mn	Mo	Ni	V	N	Nb	Ti	P	S	Si
	97.908	0.049	**0.02**	0.574	0.953	0.015	0.056	0.107	0.001	0.001	0.001	0.013	0.002	0.3
LB	0	0.02	0.02	0.2	0.8	0.008	0.02	0	0.001	0	0	0.013	0.002	0.1
UB	98.9	0.08	0.1	0.6	1.2	0.03	0.1	0.15	0.008	0.06	0.02	0.013	0.002	0.3

	Fe	C	Cr	Cu	Mn	Mo	Ni	V	N	Nb	Ti	P	S	Si
	97.908	0.0491	**0.0201**	0.574	0.953	0.015	0.056	0.1069	0.001	0.001	0.001	0.013	0.002	0.3
LB	0	0.02	0.02	0.2	0.8	0.008	0.02	0	0.001	0	0	0.013	0.002	0.1
UB	98.9	0.08	0.1	0.6	1.2	0.03	0.1	0.15	0.008	0.06	0.02	0.013	0.002	0.3

Figura 3-5 *Triple Aleatorio a una solución factible a partir de tres posiciones elegidas en forma aleatoria. Opción 1.*



	Fe	C	Cr	Cu	Mn	Mo	Ni	V	N	Nb	Ti	P	S	Si
	97.908	0.049	**0.02**	0.574	0.953	0.015	0.056	0.107	0.001	0.001	0.001	0.013	0.002	0.3
LB	0	0.02	0.02	0.2	0.8	0.008	0.02	0	0.001	0	0	0.013	0.002	0.1
UB	98.9	0.08	0.1	0.6	1.2	0.03	0.1	0.15	0.008	0.06	0.02	0.013	0.002	0.3

	Fe	C	Cr	Cu	Mn	Mo	Ni	V	N	Nb	Ti	P	S	Si
	97.908	0.0489	**0.0199**	0.574	0.953	0.015	0.056	0.1071	0.001	0.001	0.001	0.013	0.002	0.3
LB	0	0.02	0.02	0.2	0.8	0.008	0.02	0	0.001	0	0	0.013	0.002	0.1
UB	98.9	0.08	0.1	0.6	1.2	0.03	0.1	0.15	0.008	0.06	0.02	0.013	0.002	0.3

Figura 3-6 *Triple Aleatorio a una solución factible a partir de tres posiciones elegidas en forma aleatoria. Opción 2.*

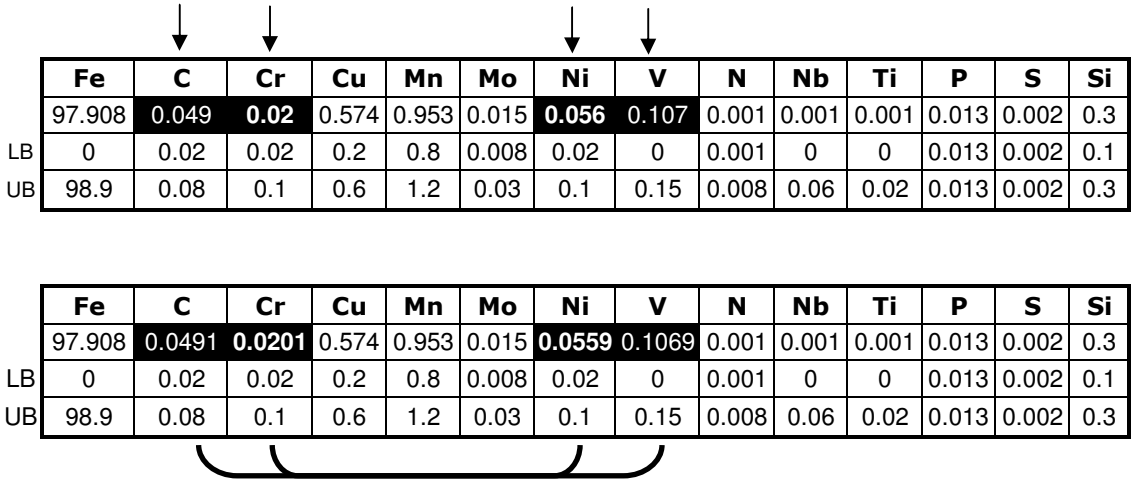
El desarrollo de la complejidad del algoritmo se explicará en el apéndice B.

- **Estructura de Vecindad Cuádruple Aleatorio**

La diferencia de esta estructura con las otras es que, aquí se generan cuatro números aleatorios: *iposic1*, *iposic2*, *iposic3* e *iposic4*. Estos números generados indicarán las posiciones en la estructura y tienen el mismo requisito del doble y triple aleatorio. Para la estructura de vecindad Cuádruple Aleatorio se generan dos casos para llevar a cabo el cambio del porcentaje de sus elementos seleccionados.

- El primer caso: incrementar dos elementos (generados en iposic1 e iposic2) y disminuir dos elementos (iposic3, iposic4) Figura 3-7.

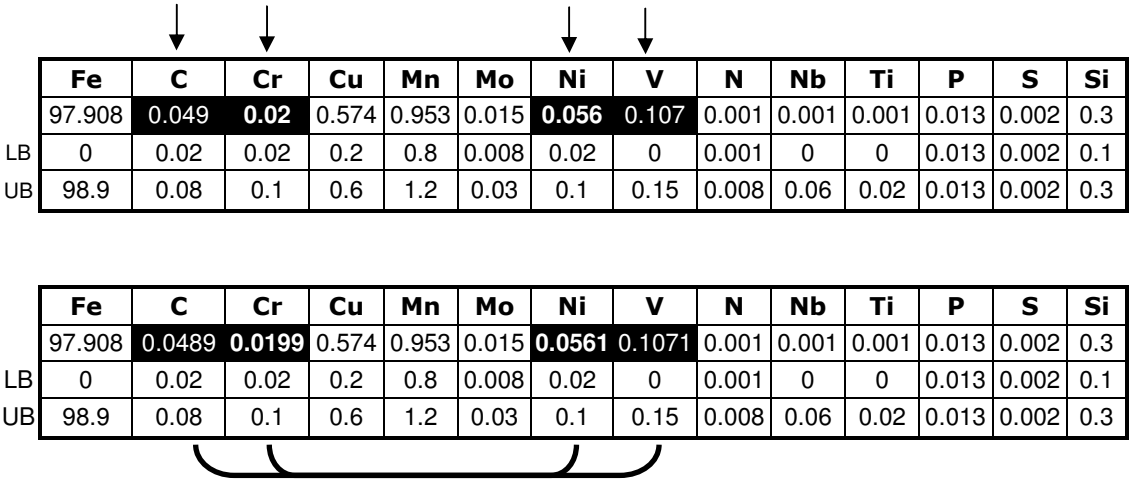
- El segundo caso: decrementar dos elementos (generados en iposic1 e iposic2) e incrementar dos elementos (iposic3 e iposic4) Figura 3-8.



	Fe	C	Cr	Cu	Mn	Mo	Ni	V	N	Nb	Ti	P	S	Si
	97.908	0.049	**0.02**	0.574	0.953	0.015	**0.056**	0.107	0.001	0.001	0.001	0.013	0.002	0.3
LB	0	0.02	0.02	0.2	0.8	0.008	0.02	0	0.001	0	0	0.013	0.002	0.1
UB	98.9	0.08	0.1	0.6	1.2	0.03	0.1	0.15	0.008	0.06	0.02	0.013	0.002	0.3

	Fe	C	Cr	Cu	Mn	Mo	Ni	V	N	Nb	Ti	P	S	Si
	97.908	0.0491	**0.0201**	0.574	0.953	0.015	**0.0559**	0.1069	0.001	0.001	0.001	0.013	0.002	0.3
LB	0	0.02	0.02	0.2	0.8	0.008	0.02	0	0.001	0	0	0.013	0.002	0.1
UB	98.9	0.08	0.1	0.6	1.2	0.03	0.1	0.15	0.008	0.06	0.02	0.013	0.002	0.3

Figura 3-7 *Cuádruple Aleatorio a una solución factible a partir de cuatro posiciones elegidas en forma aleatoria. Opción 1.*



	Fe	C	Cr	Cu	Mn	Mo	Ni	V	N	Nb	Ti	P	S	Si
	97.908	0.049	**0.02**	0.574	0.953	0.015	**0.056**	0.107	0.001	0.001	0.001	0.013	0.002	0.3
LB	0	0.02	0.02	0.2	0.8	0.008	0.02	0	0.001	0	0	0.013	0.002	0.1
UB	98.9	0.08	0.1	0.6	1.2	0.03	0.1	0.15	0.008	0.06	0.02	0.013	0.002	0.3

	Fe	C	Cr	Cu	Mn	Mo	Ni	V	N	Nb	Ti	P	S	Si
	97.908	0.0489	**0.0199**	0.574	0.953	0.015	**0.0561**	0.1071	0.001	0.001	0.001	0.013	0.002	0.3
LB	0	0.02	0.02	0.2	0.8	0.008	0.02	0	0.001	0	0	0.013	0.002	0.1
UB	98.9	0.08	0.1	0.6	1.2	0.03	0.1	0.15	0.008	0.06	0.02	0.013	0.002	0.3

Figura 3-8 *Cuádruple Aleatorio a una solución factible a partir de cuatro posiciones elegidas en forma aleatoria. Opción 2.*

El desarrollo de la complejidad del algoritmo se explicará en el apéndice B.

- **Estructura de Vecindad del Quíntuple Aleatorio**

La diferencia con el resto de las otras estructuras, radica en que aquí se generan cinco números aleatorios: *iposic1*, *iposic2*, *iposic3, iposic4 e iposic5*. Los números aleatorios generados indicarán las posiciones en la estructura y tienen el mismo requisito del doble, triple y cuádruple aleatorio. Para la estructura de vecindad Quíntuple Aleatorio se generan dos casos para llevar a cabo el cambio del porcentaje de sus elementos seleccionados.

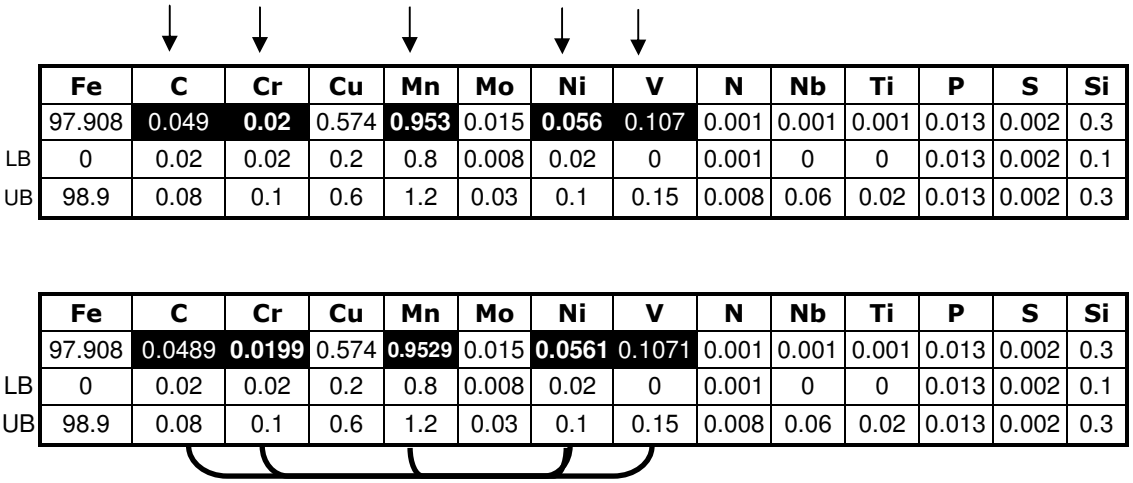
- El primer caso: incrementar tres elementos (generados en iposic1, iposic2 e iposic3) y disminuir dos elementos (iposic4 e iposic5) Figura 3-9.

- El segundo caso: decrementar tres elementos (generados en iposic1, iposic2 e iposic3) e incrementar dos elementos (iposic4 e iposic5) Figura. 3-10.

	Fe	C	Cr	Cu	Mn	Mo	Ni	V	N	Nb	Ti	P	S	Si
	97.908	0.049	**0.02**	0.574	**0.953**	0.015	**0.056**	0.107	0.001	0.001	0.001	0.013	0.002	0.3
LB	0	0.02	0.02	0.2	0.8	0.008	0.02	0	0.001	0	0	0.013	0.002	0.1
UB	98.9	0.08	0.1	0.6	1.2	0.03	0.1	0.15	0.008	0.06	0.02	0.013	0.002	0.3

	Fe	C	Cr	Cu	Mn	Mo	Ni	V	N	Nb	Ti	P	S	Si
	97.908	0.0491	**0.0201**	0.574	0.9531	0.015	**0.0559**	0.1069	0.001	0.001	0.001	0.013	0.002	0.3
LB	0	0.02	0.02	0.2	0.8	0.008	0.02	0	0.001	0	0	0.013	0.002	0.1
UB	98.9	0.08	0.1	0.6	1.2	0.03	0.1	0.15	0.008	0.06	0.02	0.013	0.002	0.3

Figura 3-9 *Quíntuple Aleatorio a una solución factible a partir de cinco posiciones elegidas en forma aleatoria. Opción 1*



	Fe	C	Cr	Cu	Mn	Mo	Ni	V	N	Nb	Ti	P	S	Si
	97.908	0.049	**0.02**	0.574	**0.953**	0.015	**0.056**	0.107	0.001	0.001	0.001	0.013	0.002	0.3
LB	0	0.02	0.02	0.2	0.8	0.008	0.02	0	0.001	0	0	0.013	0.002	0.1
UB	98.9	0.08	0.1	0.6	1.2	0.03	0.1	0.15	0.008	0.06	0.02	0.013	0.002	0.3

	Fe	C	Cr	Cu	Mn	Mo	Ni	V	N	Nb	Ti	P	S	Si
	97.908	0.0489	**0.0199**	0.574	**0.9529**	0.015	**0.0561**	0.1071	0.001	0.001	0.001	0.013	0.002	0.3
LB	0	0.02	0.02	0.2	0.8	0.008	0.02	0	0.001	0	0	0.013	0.002	0.1
UB	98.9	0.08	0.1	0.6	1.2	0.03	0.1	0.15	0.008	0.06	0.02	0.013	0.002	0.3

Figura 3-10 *Quíntuple Aleatorio a una solución factible a partir de cinco posiciones elegidas en forma aleatoria. Opción 2.*

El desarrollo de la complejidad del algoritmo se explicará en el apéndice B.

- **Estructura de Vecindad Híbrida Aleatorio**

Esta estructura de vecindad se encuentra formada por las cuatro estructuras anteriores. Aquí, se selecciona en forma aleatoria la estructura de vecindad que se ejecutará en cada iteración del ciclo de Búsqueda Local Iterada Figura 3-11.

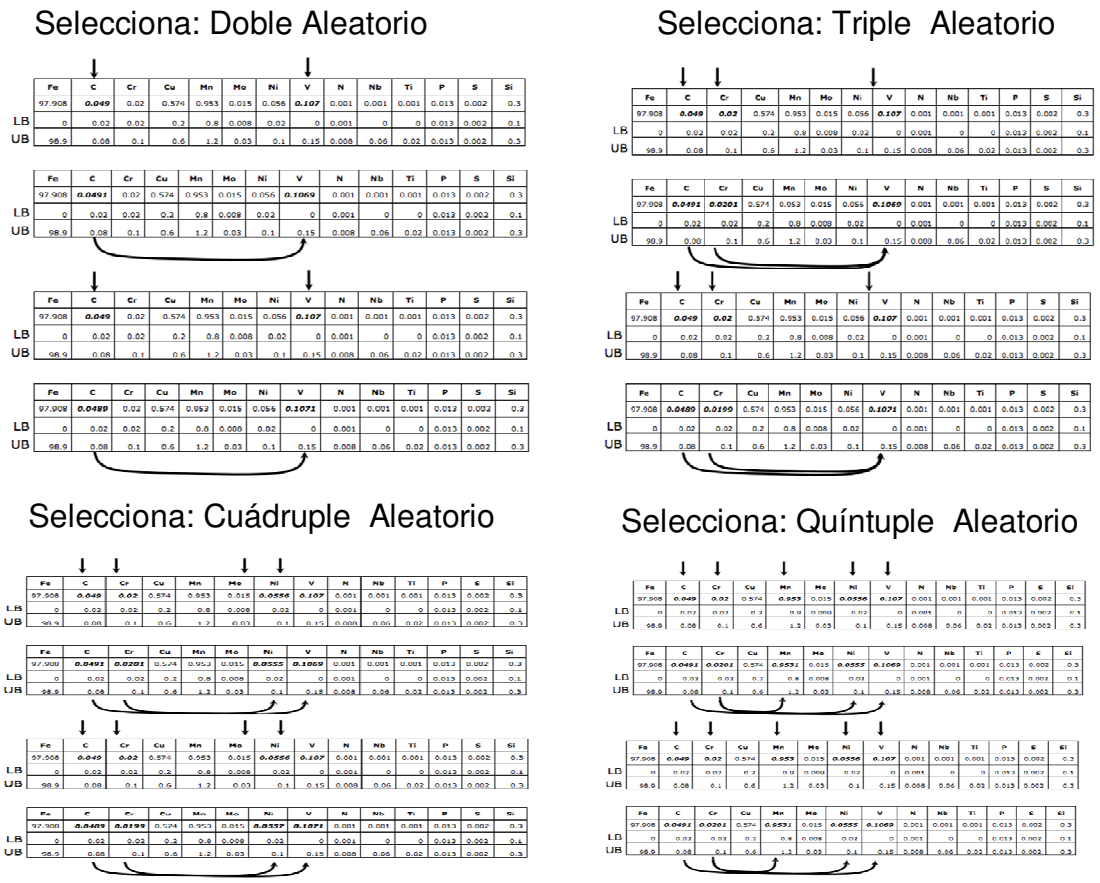


Figura 3-11 *Para realizar un cambio, el algoritmo seleccionará en forma aleatoria la estructura que aplicará.*

El comportamiento de la estructura de vecindad híbrida depende de la selección que se realice, en conjunto con la búsqueda local, en forma aleatoria de las cuatro estructuras de vecindad sencillas mencionadas anteriormente. La selección de cada una de las estructuras de vecindad estará ejecutándose hasta que se cumpla con el criterio de paro (400 iteraciones es el valor del criterio de paro, tomado de la literatura) que no es otra cosa que el número máximo de iteraciones que realizará el algoritmo [Cruz-Chávez et al., 2010]. En la Figura 3-12 se muestra el diagrama de flujo

con la estructura híbrida. El desarrollo de la complejidad del algoritmo se explicará en el apéndice B.

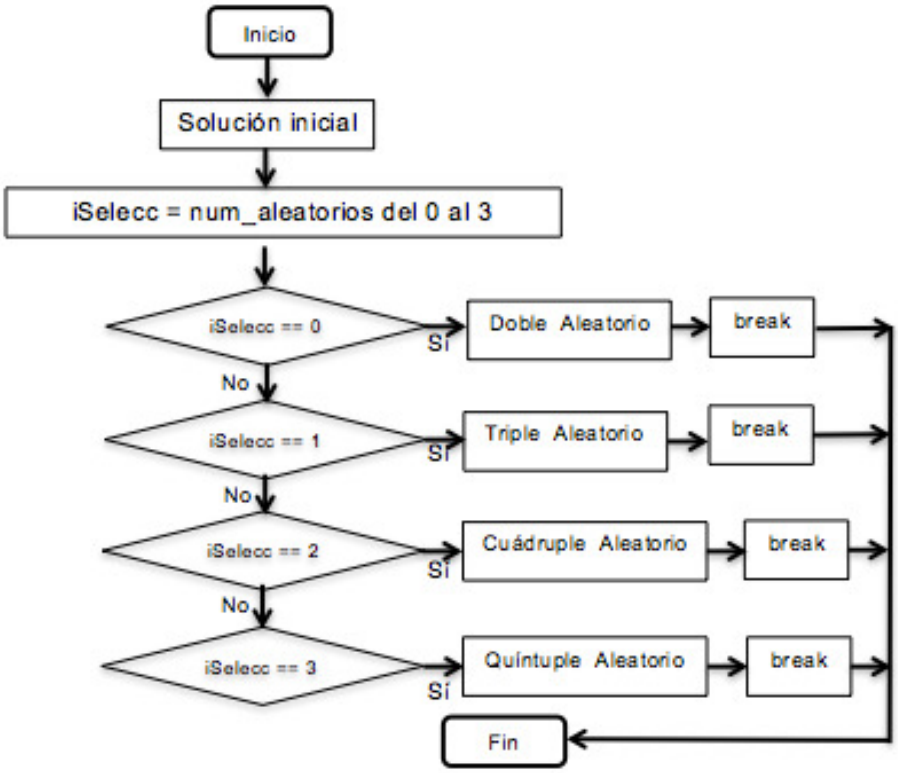


Figura 3-12 *Diagrama de flujo de la Estructura Híbrida Aleatoria.*

- **Búsqueda Local Iterada**

Existen diferentes métodos que nos pueden acercar a una solución mucho mejor que con la que contamos actualmente. Algunos métodos llamados de optimización tienen un submétodo llamado *Búsqueda Local Iterada* (*ILS – Iterated Local Search*) el cual está basado en una heurística no determinística.

En [Glover et al., 2002] se menciona que es un método estocástico de Búsqueda Local que iterativamente aplica búsquedas locales a perturbaciones de la solución inicial de una manera aleatoria en un espacio de soluciones óptimas. Este método trabaja en conjunto con la *Búsqueda*

Local (*LS – Local Search*). Y, para comprender mejor el funcionamiento de la búsqueda local iterada, es necesario saber, en primer lugar, como funciona la búsqueda local:

1. Con la búsqueda local iterada generamos la solución inicial ***s.***
2. Con la búsqueda local y con la estructura de vecindad obtenemos la solución vecina ***s'***.
3. Evaluamos, mediante la función objetivo, a ***s*** y ***s'*** [Michaelewicz y Fogel, 2004].
4. Si esta nueva solución ***s'*** es mejor que la actual ***s*** será sustituida, de lo contrario permanecerá igual; es decir:

   Si ***s'*** *>=* ***s*** *entonces* ***s*** ← ***s'***.
5. Dentro de la búsqueda local se repiten los pasos 2 y 3 hasta que se cumpla con el criterio de paro (400 iteraciones).

La búsqueda local iterada permite salir de las soluciones óptimas locales. El valor óptimo local que se obtiene es tomado como el mejor e iniciará la siguiente iteración. Si en las siguientes iteraciones se encuentra uno mucho mejor que el actual, lo reemplaza, sino es así, se genera un nuevo cambio (con la búsqueda local) hasta encontrar uno mejor que el actual. Es decir:

s'_LS pasa a ser s_ILS si f(s'_LS) >= f(s_ILS)

El cambio que se le aplique a ***s***, mediante alguna estructura de vecindad, será el que determine el tipo de solución obtenida. Para interpretarlo mejor, observe la Figura 3-13. En la figura 3-14 se encuentra desarrollado el Algoritmo de Búsqueda Local Iterada.

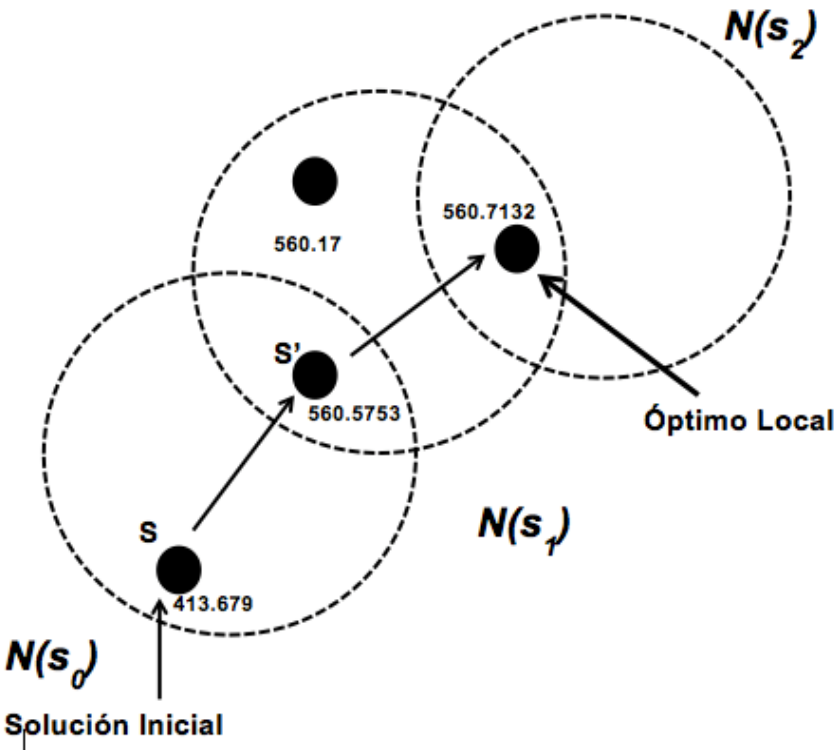


Figura 3-13 *Representación de la Solución inicial y Nueva Solución con Búsquedas Locales (local e iterada).*

```
Hacer
    Generar solución inicial s
    Hacer
        s'=Estructura_de_Vecindad(s)
        Si (f (s') >= f (s)) entonces
            s = s'
            Si (f(s') > f(Sm-LS)) entonces
                Sm-LS=s'
            Fin-si
        Fin-si
    Hasta CP-LS
    Si (f (Sm-LS) > f (Sm-ILS)) entonces
        Sm-ILS = Sm-LS
    Fin-si
Mientras CP_ILS
```

Figura 3-14 *Algoritmo de Búsqueda Local Iterada [Cruz-Chávez, M. A, 2005].*

3. 4 Esquema Generalizado del Algoritmo de Recocido Simulado

En este apartado, se describe la forma en la que trabaja el Algoritmo de Recocido Simulado con la estructura de vecindad Híbrida, más la composición química, partiendo de la primera solución hasta la mejor solución encontrada. La figura 3-15 muestra el Algoritmo de Recocido Simulado:

Inicializar $(T_0, s_0, T_f, \alpha, L_k)$
$k = 1$
$T_k = T_0$
$s = s_0$
Repetir
  Repetir
    s' = Estructura_de_vecindad_Híbrida(s)
    si $f(s') \geq f(s)$ entonces
      s = s'
    fin-si
    sino
      $Paceptar = \exp\left(\frac{f(s') - f(s)}{T_k}\right)$
      ρ = número aleatorio de [0,1)
      si $\rho < Paceptar$ **entonces**
        s = s'
      fin-si
  Hasta alcanzar el equilibrio (L_k)
  $T_{k+1} = \alpha T_k$
  $k = k + 1$
Hasta $T_k \leq T_f$

Figura 3-15 *Algoritmo de Recocido Simulado*

La forma en la que trabaja el Algoritmo de Recocido Simulado se describe a continuación:

1.- Inicializar los parámetros T_0 con un valor inicial, T_f con un valor final (este parámetro permitirá decrementar T_0) y s_0 con una configuración inicial.

$T_k = T_0$

$k = 1$

$s = s_0$

2.- Para cada iteración $k, k = 1 \ldots k_f$ hacer lo siguiente:

a. Repetir hasta alcanzar el equilibrio (L_k);
   - Calcular el valor de la solución actual *s* mediante la función de Costo *f(s);*
   - Generar una nueva solución *s'* utilizando una estructura de vecindad híbrida (incremento/decremento) *s' = N (s);*
   - Calcular el valor de la nueva solución *s'* mediante la función de costo *f(s').*
   - Evaluar la función objetivo donde la función de costo de la solución vecina *f(s')* sea mayor que la función de costo de la solución actual *f(s),* de cumplirse se reemplaza *s = s'* de lo contrario, se acepta o rechaza de acuerdo al criterio de aceptación de la función de probabilidad de Boltzmann *P(s)*.

b. Decrementar T_0 para *k+1* utilizando el coeficiente del parámetro de control α , $T_{k+1} = \alpha T_k$, donde α es una constante tal que $0 < \alpha < 1$.

3.- Cuando $T_{k+1} \leq T_f$, terminar.

Los parámetros importantes del Algoritmo de Recocido Simulado se definen a continuación:

F(s): Función de costo de la solución actual a maximizar, en este trabajo equivale a la composición química del acero.

F(s'): Función de costo de la solución vecina.

N(s): Es la estructura de vecindad (genera nuevos vecinos).

P(s): Es la función que define el criterio de aceptación e indica si el nuevo estado se acepta o se rechaza, esta función la da la distribución de probabilidad de Boltzmann.

f(s) se define como la función de costo, para el problema de maximizar la resistencia del acero microaleado. La vecindad *N(s)* de *S* se define como el conjunto de soluciones factibles que pueden ser generadas a partir de *s* mediante un movimiento, este puede ser, una perturbación, inserción o eliminación pero para este problema en específico se trata de incrementos/decrementos.

Es importante mencionar que el algoritmo de Recocido Simulado utiliza el algoritmo de Metrópolis para optimización combinatoria. Donde se desea maximizar una función de costo *f* haciendo las siguientes substituciones. Se toma el estado ***s*** como una solución posible del problema de optimización; *f(s)* como el costo del estado *s*, se genera un nuevo estado *s'* y *f(s)* como el costo de estado *s'* aplicando una pequeña perturbación a *s'* (estructura de vecindad) y se define un parámetro de control $c = T_0$ cuyo valor controla en su mayor parte el criterio de aceptación de malas soluciones esto es, si T_0 es muy grande la probabilidad de aceptación será mayor, y si T_0 es muy pequeño, la probabilidad de aceptación será prácticamente cercana a cero.

El criterio de aceptación es:

g

$$P\{Aceptar \quad nueva \quad solución\} = \begin{cases} 1, & f(s') \geq f(s) \\ e^{\frac{f(s')-f(s)}{c}} & f(s') < f(s) \end{cases}$$

Una vez mencionado el funcionamiento del Algoritmo de Recocido Simulado, se explica la forma en la que trabaja la estructura de vecindad híbrida, y es la siguiente:

1.- La estructura donde se genera la primera solución se encuentra de la siguiente forma (Figura 3-16):

	Fe	C	Cr	Cu	Mn	Mo	Ni	V	N	Nb	Ti	P	S	Si
	98.9	0	0	0	0	0	0	0	0.001	0.001	0.001	0.013	0.002	0.3
LB	0	0.02	0.02	0.2	0.8	0.008	0.02	0	0.001	0	0	0.013	0.002	0.1
UB	98.9	0.08	0.1	0.6	1.2	0.03	0.1	0.15	0.008	0.06	0.02	0.013	0.002	0.3

Figura 3-16 *Vector (estructura) con la composición del acero para obtener la primera solución.*

En la parte superior del vector, Figura 3-17, se encuentran señalados, con los signos abreviados, los nombres de los catorce elementos químicos que componen el acero microaleado. Los elementos a los que se les podrá asignar o cambiar su composición van desde el Fe hasta el V, del N hasta el Si permanecerá constante. A partir del C hasta el V, el programa se encargará de generar la cantidad (peso en %) de los elementos ya que es con ellos con los que se tratará de obtener la resistencia. El Fe es el único elemento que contiene una cantidad asignada (Figura 3-18) y al cual se le podrá incrementar (Δ) o decrementar (-Δ) su composición, respetando su rango, para balancear al 100% la cantidad total de todos los elementos.
LB (Lower Bound) y UB (Upper Bound) indican los límites de la composición de cada elemento dentro del cual podrán realizar los cambios (Figura 3-19).

Fe	C	Cr	Cu	Mn

Figura 3-17 *Nombres de los elementos que forman la composición del acero.*

Fe
98.9

Figura 3-18 *El Fe es el único elemento al cual se le podrá aumentar o disminuir su composición para balancear el acero.*

LB	0	0.02	0.02	0.2	
UB	98.9	0.08	0.1	0.6	

Figura 3-19 *Rango (peso en %) de cada uno de los elementos que componen el acero.*

2.- Una vez que se generan las cantidades del Carbono al Vanadio, se procede a realizar una suma para corroborar si se ha rebasado o no el 100%, en caso de que así sea, será al Fe al único elemento que se le podrá aplicar los cambios para balancear la cantidad de los elementos (Figura 3-20).

	Fe	C	Cr	Cu	Mn	Mo	Ni	V	N	Nb	Ti	P	S	Si	suma
	97.647	0.0581	0.059	0.5259	1.195	0.024	0.093	0.084	0.001	0.001	0.001	0.013	0.002	0.3	100.004
LB	0	0.02	0.02	0.2	0.8	0.008	0.02	0	0.001	0	0	0.013	0.002	0.1	
UB	98.9	0.08	0.1	0.6	1.2	0.03	0.1	0.15	0.008	0.06	0.02	0.013	0.002	0.3	

Figura 3-20 *Vector con la primera solución generada y con las composiciones arriba del 100%.*

Para comprobar la suma, existe dentro del programa una función encargada de realizarla, así como una de balancear. El vector, una vez balanceada la suma, se ve así (Figura 3-21):

	Fe	C	Cr	Cu	Mn	Mo	Ni	V	N	Nb	Ti	P	S	Si	suma
	97.643	0.0581	0.059	0.5259	1.195	0.024	0.093	0.084	0.001	0.001	0.001	0.013	0.002	0.3	100
LB	0	0.02	0.02	0.2	0.8	0.008	0.02	0	0.001	0	0	0.013	0.002	0.1	
UB	98.9	0.08	0.1	0.6	1.2	0.03	0.1	0.15	0.008	0.06	0.02	0.013	0.002	0.3	

Figura 3-21 *Vector con la primera solución generada y al 100% (balanceada).*

3.- Una vez que se obtiene la primera solución, comienza la función del Algoritmo de Recocido Simulado. Esta configuración se copia en otra estructura para que se realice el incremento (Δ) o decremento (-Δ) de acuerdo a la estructura de vecindad Híbrida.

4.- Después de iterar dentro del Algoritmo de Recocido Simulado hasta cumplir con el criterio de paro (T_f), obtenemos la mejor solución encontrada. Para esto, cada vez que se genera una solución, pasa por la función objetivo para corroborar si es o no mejor que la anterior.

5.- Finalmente, la mejor solución obtenida se almacena en la estructura que guarda las mejores soluciones encontradas (Figura 3-22).

	Fe	C	Cr	Cu	Mn	Mo	Ni	V	N	Nb	Ti	P	S	Si	suma
	97.6429	0.0583	0.0589	0.5258	1.195	0.0241	0.093	0.084	0.001	0.001	0.001	0.013	0.002	0.3	100
LB	0	0.02	0.02	0.2	0.8	0.008	0.02	0	0.001	0	0	0.013	0.002	0.1	
UB	98.9	0.08	0.1	0.6	1.2	0.03	0.1	0.15	0.008	0.06	0.02	0.013	0.002	0.3	

Figura 3-22 *Mejor solución encontrada al final de la ejecución del programa con el Algoritmo de Recocido Simulado.*

3.5 Metodología de Sintonización

Para lograr una buena sintonización de los parámetros de control de un algoritmo, se debe de encontrar un rango numérico. El parámetro de control inicial tiene un rango de 10019 a 1000, lo dividimos entre 30 (cantidad mínima de pruebas) y obtenemos el valor que será el decremento o incremento. Para el caso del Algoritmo de Recocido Simulado este parámetro no debe ser tan grande de tal forma que el Algoritmo no se tarde mucho o

que este tarde muy poco. Por ejemplo, la longitud de la Cadena de Markov tiene un rango numérico de 1 a 4 veces el tamaño de la estructura de vecindad, este parámetro no tiene un valor medio. El resto de los otros parámetros cuenta con un valor medio del cual partimos y son 15 incrementos y 15 decrementos, es decir, tenemos un valor arriba de la solución inicial con la cual partimos. Con el Algoritmo de Recocido Simulado partimos con una solución inicial, lo que hacemos es incrementar ese valor a un número muy grande, ejemplo: si es 500 lo incrementamos a diez mil porque sería un valor muy grande. El coeficiente de control cuenta con un valor medio el cual fue obtenido de la literatura, ese valor es 0.98 y a partir de aquí son 15 incrementos y 15 decrementos. El parámetro de control final también cuenta con un valor intermedio y es de 0.001. Partiendo de este número, realizamos lo mismo con el parámetro anterior, 15 incrementos y decrementos.

El objetivo del análisis de sensibilidad, que se verá en el capítulo 4.3, es encontrar la sintonización adecuada de los parámetros de control del algoritmo para lograr una mejora en la eficiencia y eficacia.

En la siguiente sección, se explica una metodología de sintonización, la cual fue tomada de la tesis de [Cruz-Chávez, 2005] y de [Martínez-Oropeza, 2010], para encontrar la adecuada proporción en los valores correspondientes a los parámetros de control.

1. Selección de los Parámetros de Control.

[Martínez-Oropeza, 2010] menciona que existen dos formas para poder determinar la sintonización: llevar a cabo una revisión de publicaciones que se relacionen con el algoritmo implementado, de esta manera es posible

analizar los parámetros que fueron tomados en cuenta en investigaciones anteriores. Y, una segunda opción consiste en realizar un análisis tanto del problema como del algoritmo, para identificar los parámetros críticos que influyen de cierta manera en la calidad de las soluciones.

2. Establecer Rangos de Evaluación.

Una vez identificados los parámetros de control, se deben establecer los rangos para realizar el análisis de sensibilidad a cada uno de ellos, por lo que se recomienda revisar en la literatura para identificar y analizar los valores utilizados por diferentes autores para el algoritmo propuesto. Realizar una adecuada sintonización permitirá identificar la proporción adecuada de los parámetros de control, dentro de la cuál, el algoritmo obtiene buenas soluciones.

3. Pruebas a Rangos de Evaluación.

Tomando en cuenta los rangos que permitirán evaluar el comportamiento del algoritmo cuando los parámetros de control tomen distintos valores.

Se recomienda realizar conjuntos, mínimo, de 30 pruebas para cada una de las muestras calculadas. Para una adecuada sintonización de los parámetros de control, se debe realizar un barrido de los valores correspondientes a una de las variables, manteniendo fijos los demás, hasta identificar aquel valor que mejore la calidad de las soluciones. Una vez encontrado el valor para la variable, se fija y se comienza con la variación de otro parámetro, realizando

el mismo proceso hasta obtener el conjunto de valores que permitan que el algoritmo mejore en eficacia y eficiencia.

4. Sintonización de Parámetros.

Todos los valores obtenidos para cada uno de los parámetros de control al terminar de evaluar cada una de las muestras, son considerados como los valores de sintonización. Dicha sintonización se lleva a cabo con la finalidad de identificar los valores correspondientes a las variables de control que influirán, de manera positiva, en la calidad de la solución, esto permitirá obtener una mejora en el desempeño del algoritmo [Martínez-Oropeza, 2010].

3.6 Generación Aleatoria de Instancias de Prueba

Por ser un problema relativamente nuevo en el que se han usado metaheurísticas (genéticos) y a la fecha no se encontró en la literatura un problema del mismo tipo que trabajara únicamente con la composición química del material utilizando para esto el algoritmo de recocido simulado. Debido a lo anterior las instancias para este problema fueron generadas en forma aleatoria.

Para poder generar la instancia en estudio, se creo un programa en Xcode, con C y gcc, que nos permite seleccionar una instancia, a partir de esa instancia se genera una solución de forma aleatoria (Tabla 3-1).

Tabla 3-1 *Instancia que presenta una solución inicial.*

Elemento	Lím. Inferior	Lím. Superior	Número generado
Fe	0	98.9	97.4226
C	0.02	0.08	0.0799
Cr	0.02	0.2	0.0999
Cu	0.02	0.6	0.5999
Mn	0.8	1.2	1.1999
Mo	0.008	0.03	0.03
N	0.02	0.1	0.0999
V	0	0.15	0.1499
N	0	0.06	0.001
T	0	0.02	0.001
P	0.013	0.013	0.013
S	0.002	0.002	0.003
Si	0.1	0.3	0.3
			100

La instancia para este problema se compone de los elementos químicos del acero (primera columna), los cuales cuentan con un rango establecido dentro del límite superior y del límite inferior (segunda y tercera columna). En la cuarta columna se encuentran los números generados aleatoriamente. Cada uno de los valores está dentro del rango correspondiente a cada elemento. La suma total de la composición química debe dar siempre el 100% para que sea considerada como una solución factible. De esta forma, se genera una instancia de prueba de forma aleatoria para este tipo de problema.

3.7 Análisis de la Complejidad del Algoritmo de Recocido Simulado

Cuando se tiene un algoritmo que ya está funcionando, se debe realizar un estudio para conocer su comportamiento y medir su rendimiento, especialmente el uso eficiente de los recursos así como su simplicidad.

La complejidad computacional estudia la manera de clasificar algoritmos como buenos y malos y de acuerdo a la dificultad inherente de resolverlos. Para ello, podemos basarnos en un algoritmo donde pueda ser medido en función de los siguientes parámetros:

· **Temporal o Tiempo:** Es el tiempo requerido para la ejecución del algoritmo para encontrar una solución.

· **Espacial o Espacio:** Es el almacenamiento requerido por un algoritmo para encontrar la solución, se le considera también como la cantidad de memoria utilizada durante la ejecución de dicho algoritmo.

Los parámetros, mencionados arriba, representan el costo que requiere un algoritmo para encontrar una solución al problema tratado.

El tiempo de ejecución de un algoritmo o la complejidad temporal se describe como una función de entrada *T(n)*, donde se toma el tamaño de la entrada, representado por ***n***, que se encuentra en función de los siguientes parámetros: Datos de entrada, calidad del código objeto compilado, naturaleza y rapidez del procesador así como la complejidad del algoritmo.

Este análisis de complejidad permite determinar la eficiencia de un algoritmo para poder compararlo con otros algoritmos que resuelvan un mismo problema. T(n) representa el número de instrucciones simples (asignaciones,

comparaciones, operaciones aritméticas, etc.) que son ejecutadas en el algoritmo. Siempre se considera la complejidad del algoritmo en el peor de los casos, aunque también puede ser analizado el mejor caso y el caso promedio.

Para determinar si un algoritmo es considerado bueno o malo, nos basamos en las siguientes convenciones:

1. Todos los algoritmos, desde constantes hasta polinomiales, son polinomiales y son buenos algoritmos.

2. Todos los algoritmos exponenciales y factoriales, son exponenciales y son malos algoritmos.

3. Los algoritmos polinomiales son buenos algoritmos.

4. Los algoritms exponenciales son malos algoritmos.

La complejidad del Algoritmo de Recocido Simulado se calculó realizando primero un análisis del número de instrucciones requeridas por el algoritmo para encontrar una solución al problema tratado obteniendo la función temporal del algoritmo. Dentro de la función la letra **e** se interpreta como el número de ejecuciones que se realizarán.

Las siguientes funciones temporales pertenecen a cada una de las estructuras de vecindad con las cuales se trabajó. Se toma en cuenta, para el peor de los casos, a la estructura de vecindad quíntuple, debido a que contiene la función con los valores más altos, lo que indica que es la que ejecuta más instrucciones y la que hace que la eficiencia disminuya cuando forma parte de la estructura de vecindad Híbrida.

Doble A: $f(e) = 15e + 149$

Triple A: $f(e) = 28e + 220$

Cuádruple A: $f(e) = 41e + 295$

Quíntuple A: $f(e) = 56e + 367$

Híbrida A: $T(e) = 140e + 1031$

Para el caso del Recocido Simulado, ***m*** representa el número de iteraciones de un ciclo; ***n*** representa los elementos químicos de la composición del acero a los cuales se les cambiará su composición. Con esto se concluye que la complejidad del Algoritmo de Recocido Simulado con una estructura de vecindad Híbrida para el peor de los casos, tiene una complejidad de:

$$O\left(n + nm(n-7)^2 \ln m\right)$$

El desarrollo de la complejidad del algoritmo se explicará en el apéndice B.

Los científicos no persiguen la verdad, es ésta quien los persigue a ellos.
Karl Schlechta

Capítulo 4

Resultados Experimentales del Algoritmo de Recocido Simulado

Se presentan resultados de la evaluación de cinco estructuras de vecindad utilizando un algoritmo de búsqueda local iterada para implementar la estructura más eficiente en Recocido Simulado con sus parámetros sintonizados. Se presenta el análisis del desempeño del algoritmo de Recocido Simulado.

4.1 Descripción del Equipo Utilizado

Para determinar la estructura de vecindad a la cual se le aplicará el Algoritmo de Recocido Simulado fue necesario realizar pruebas experimentales a cada una de las estructura de vecindad propuestas. Estas pruebas se llevaron a cabo en:

Cluster Tarantula CIICAp-UAEM (Figura 4-1).
Proc Max 3.20GHz
RAM 7GB
HD 1.12TB
: : FRONTEND
Proc Intel(R) Pentium(R) D CPU 2.80GHz
RAM 1GB
HD 160GB
: : 6 NODOS Procs Intel(R) Celeron(R) CPU E1400 @ 2.00GHz
RAM 1GB
HD 160GB
Grid Morelos

Figura 4-1 *Clúster ciicap ubicado en el CIICAp de la UAEM.*

Figura 4-2 *Clúster NOPAL ubicado en el ITVer.*

CLUSTER NOPAL (Figura 4-2)
Proc Max 3.20GHz
RAM 57GB
HD 1.2TB

: : FRONTEND
Proc Dual Intel(R) Pentium(R) 4 CPU 3.20GHz
RAM 1GB
HD 80GB

: : 5 NODOS
Procs Quad Intel(R) Core(TM)2 Quad CPU Q8200 @ 2.33GHz
RAM 4GB
HD 80GB

: : 9 NODOS
Procs Dual Intel(R) Core(TM)2 Duo CPU E8200 @ 2.66GHz
RAM 4GB
HD 80GB

4.2 Análisis de la Estructura de Vecindad con el Algoritmo de Búsqueda Local Iterada

Las pruebas que se realizaron en el clúster con las diferentes estructuras de vecindad (Doble, Triple, Cuádruple, Quíntuple e Híbrido Aleatorio) servirán para determinar cual de ellas es la más eficiente y cual usaremos para aplicarla en el Algoritmo de Recocido Simulado.

Los resultados arrojados por el clúster para la evaluación de las estructuras son (Tabla 4-1):

Tabla 4-1 *Resultados de la evaluación de cada una de las estructura de Vecindad.*

ESTRUCTURA	MEJOR	PEOR	PROM. MEJOR	σ PROM
Doble A	560.7226	560.5687	560.71195	0.038837532
Triple A	560.723	559.261	560.7093767	0.041466125
Cuádruple A	560.7272	560.5694	560.7110967	0.031658928
Quíntuple A	560.7181	554.7926	560.69919	0.29037762
Híbrido A	**560.7351**	**560.5701**	**560.7122467**	0.039025684

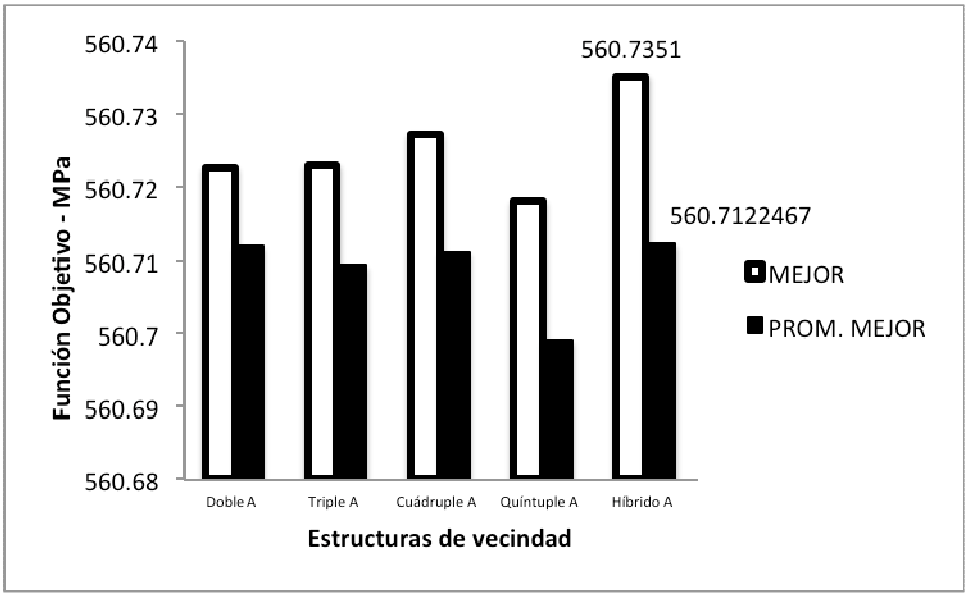


Figura 4-3 *Gráfica con los resultados comparativos de cada una de las estructuras de vecindad.La mejor estructura es la Híbrida Aleatoria.*

La tabla y la gráfica mostradas son el resumen de cada una de las estructuras de vecindad, las cuales tienen un tamaño de **989000** (que corresponde al tamaño de la vecindad) y un criterio de paro de **400**

iteraciones; además, en cada una de las estructuras se realizaron 30 pruebas. De los valores generados, desde la solución inicial hasta la mejor solución obtenida, tomamos solamente el mejor y el peor valor de cada una de las estructuras; de igual manera, obtenemos el promedio así como la desviación estándar promedio.

Una vez agrupados los valores obtenidos, verificamos cuál de las estructuras es la que mejores resultados arroja y, para visualizarlos de una forma mucho más rápida, graficamos esos valores. A simple vista, podemos observar en la gráfica que la mejor estructura es la Híbrida, ya que tanto en el mejor resultado como en el promedio son los valores más altos y en el caso del peor valor queda en cuarto lugar. Con la desviación estándar podemos observar que no está muy retirado del valor de su promedio obtenido.

Las gráficas de la figura 4-4, son los resultados arrojados en la ejecución del programa, el cual consta, cada uno de ellos de 400 iteraciones (que corresponde al criterio de paro) ejecutadas en 30 pruebas. Lo que se grafica es sólo el mejor valor obtenido de las iteraciones de cada una de las estructuras de vecindad.

Una vez determinada la mejor estructura de vecindad, se probará con diferentes tamaños de vecindad para realizar el ajuste y determinar el correcto. La siguiente prueba realizada fue únicamente con la estructura de vecindad híbrida Aleatoria, la cuál resultó ser la mejor. Los porcentajes manejados son: 5%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% y 100% el tamaño de la vecindad. Estas pruebas servirán para determinar el valor correcto del tamaño de la vecindad y ajustarlo para iniciar la sintonización de los parámetros de control del Algoritmo de Recocido Simulado.

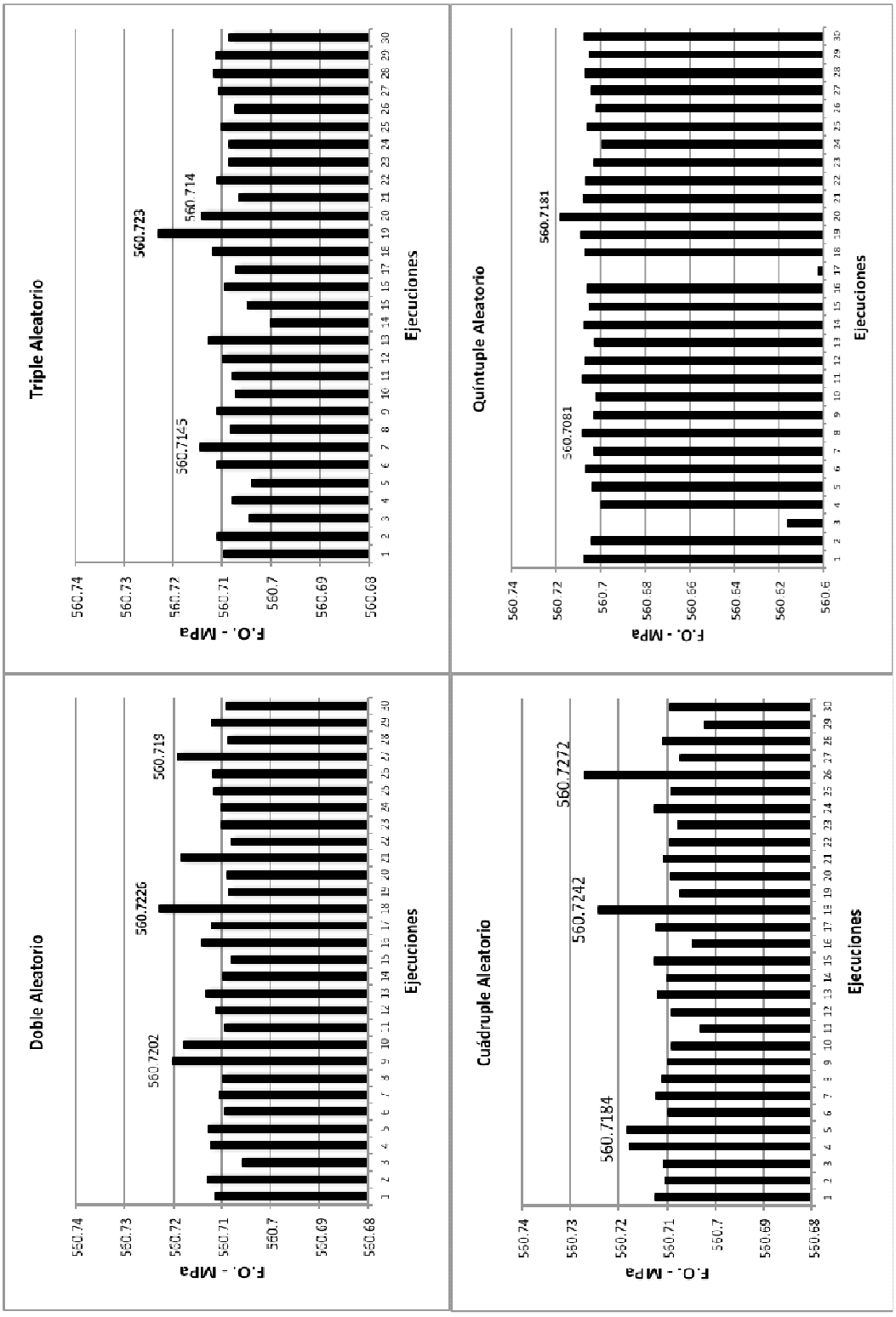


Figura 4-4 (a)Gráfica con los resultados de las ejecuciones de cada una de las cinco estructuras de vecindad.

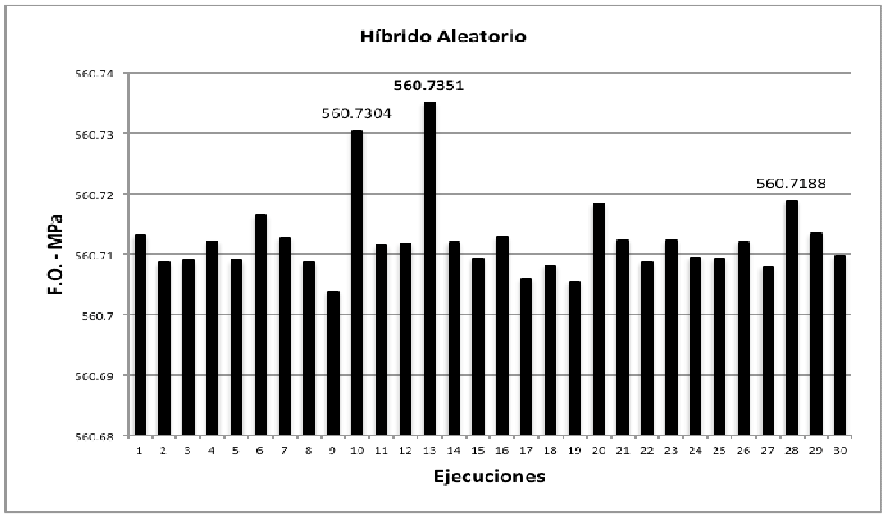


Figura 4-5 Gráfica con los resultados de las ejecuciones de cada una de las cinco estructuras de vecindad (b).

***Tabla 4-2** Resultados del Híbrido Aleatorio con pruebas desde el 5% al 100%*

ESTRUCTURA	MEJOR	PEOR	PROM. MEJOR	σ PROM
Híbrido A 5%	560.5685	**540.5428**	559.5534767	1.332514519
Híbrido A 10%	560.7272	544.0187	560.56363	1.355543374
Híbrido A 20%	560.7178	547.6995	560.6444533	0.730190119
Híbrido A 30%	**560.7346**	559.7758	560.6943067	0.248209605
Híbrido A 40%	560.733	554.9217	560.7091967	0.16541446
Híbrido A 50%	560.7333	559.1024	560.7089333	0.054319031
Híbrido A 60%	560.727	560.569	560.7116867	0.030341118
Híbrido A 70%	**560.7346**	560.5694	560.7120367	0.031377934
Híbrido A 80%	560.7247	560.5698	**560.71228**	0.036617864
Híbrido A 90%	560.7212	560.57	560.7114067	0.031654708
Híbrido A 100%	**560.7351**	560.5701	560.7122467	0.039025684

La tabla 4-2 contiene el resumen del mejor, peor, promedio mejor y desvación promedio de las pruebas generadas a la mejor estructura de vecindad seleccionada, la híbrida aleatoria en sus diferenes tamaños de vecindad.

Podemos observar que los mejores resultados obtenidos son los equivalentes, respecto al tamaño de vecindad, a un 30%, 70% y 100%. Los que nos indica que podemos trabajar con cualquiera de los tres ya que la diferencia no es mucha.

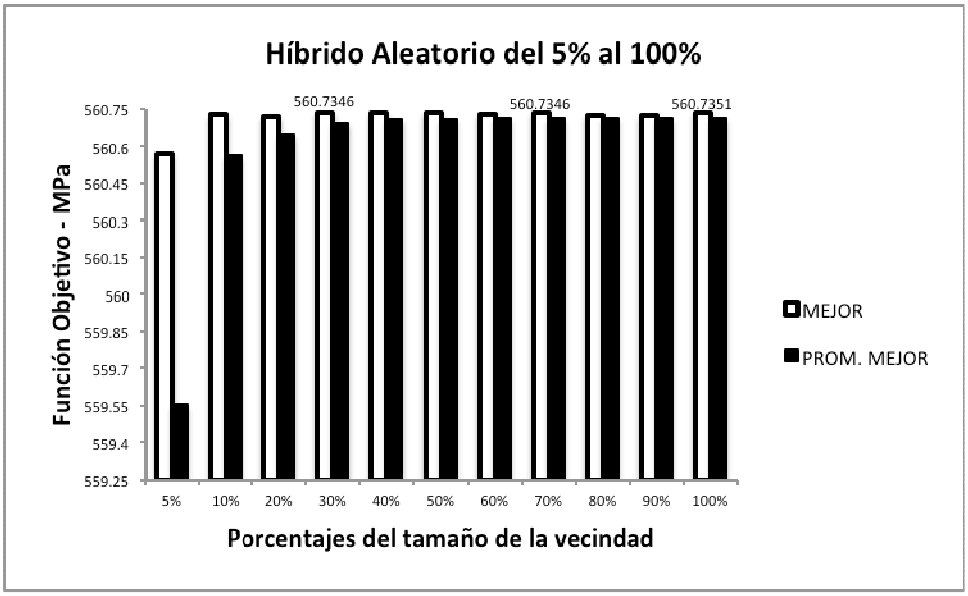


Figura 4-6 *Gráfica con los resultados comparativos de la estructura de vecindad Híbrido Aleatorio evaluadas desde un 5% hasta un 100%.*

En la figura 4-5 observamos que los resultados arrojados en los diferentes tamaños de vecindad no están tan alejados unos de otros, a excepción del tamaño que equivale al 5%. Sin embargo, vistos los resultados desde la gráfica, se decidió trabajar con el tamaño de vecindad al 100%.

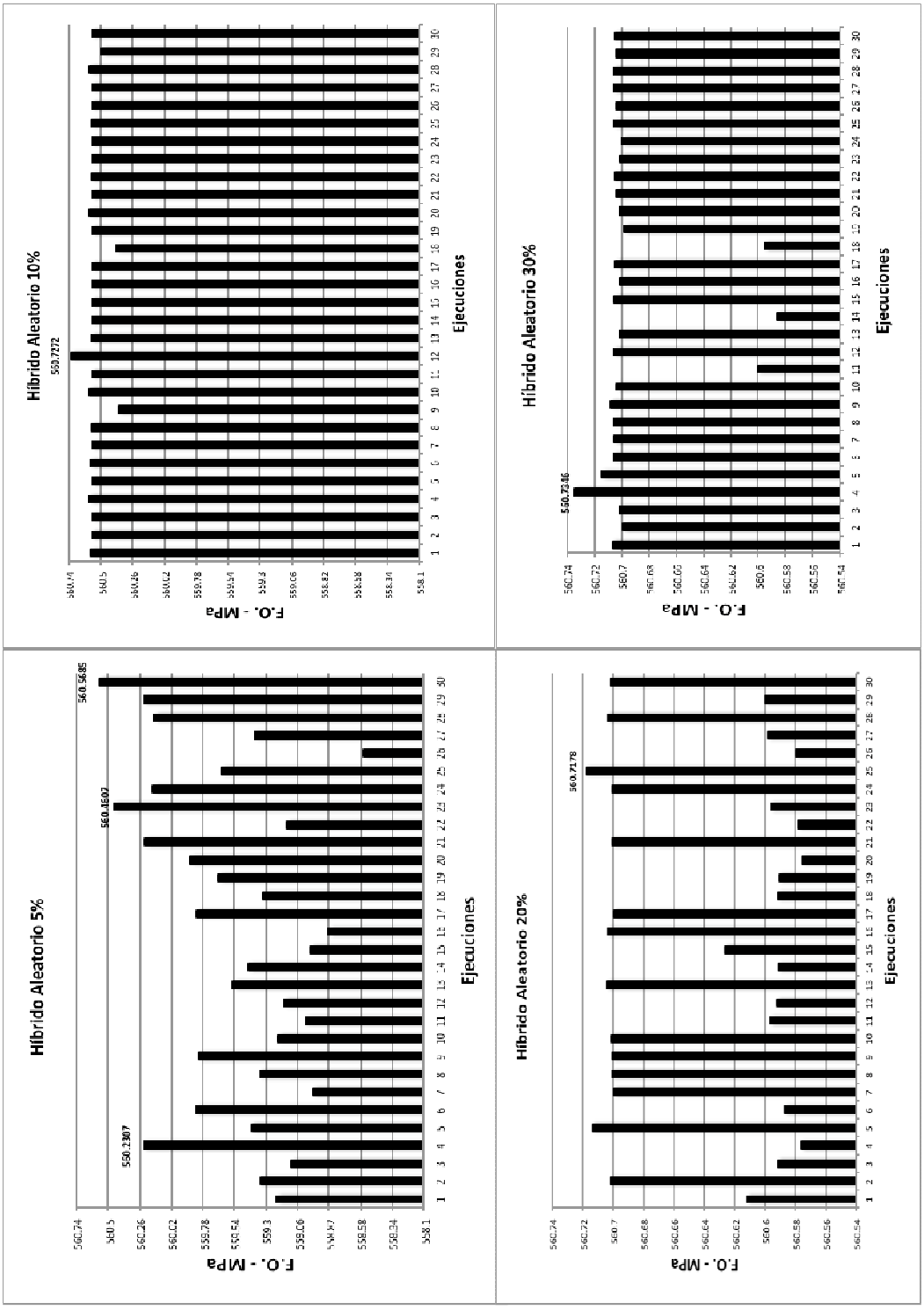


Figura 4-7 *(a)Gráficas con los resultados de las ejecuciones de cada una de las pruebas realizadas a la estructura de vecindad Híbrido Aleatorio las cuales van desde el 5% hasta un 100%.*

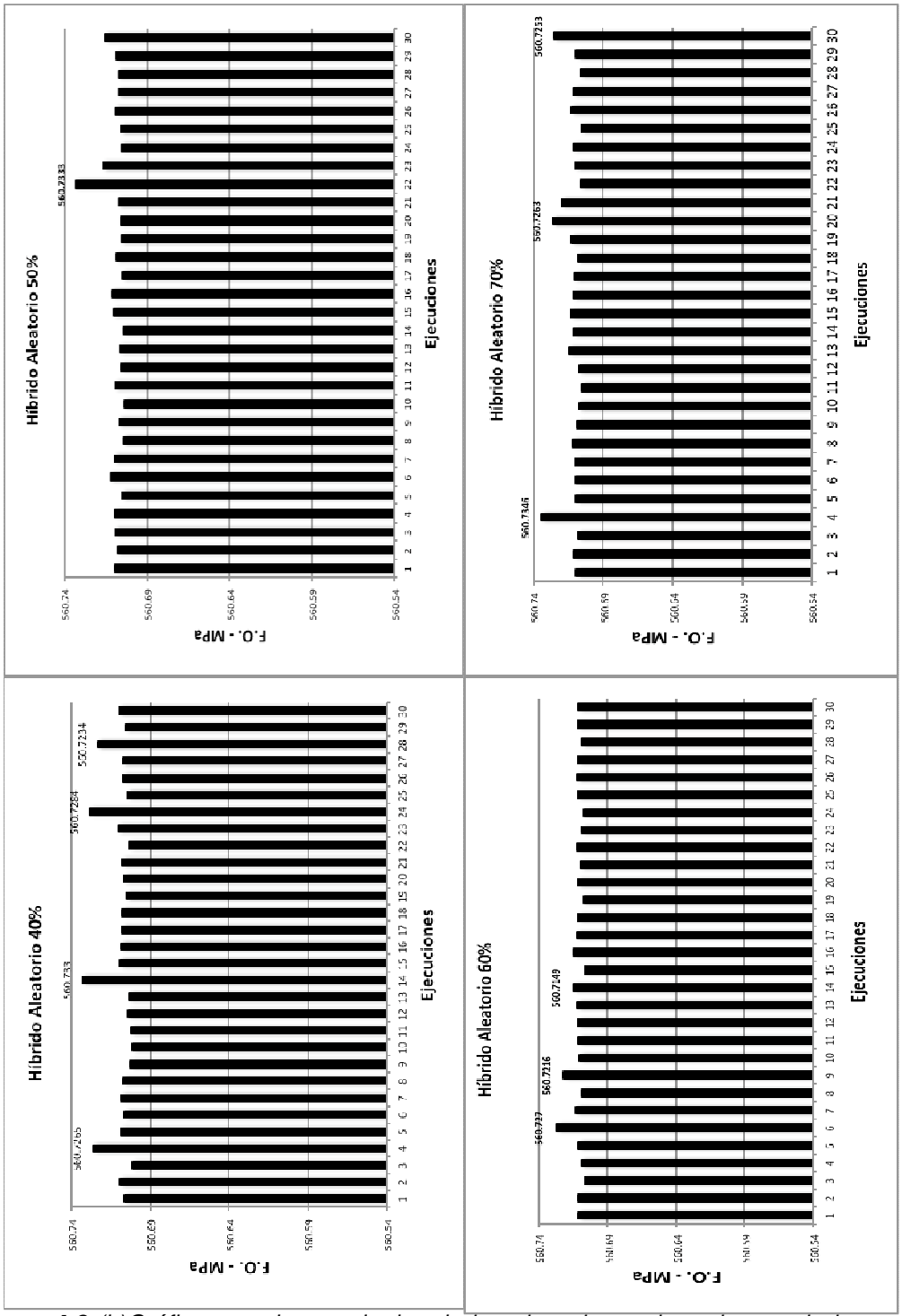


Figura 4-8 *(b)Gráficas con los resultados de las ejecuciones de cada una de las pruebas realizadas a la estructura de vecindad Híbrido Aleatorio las cuales van desde el 5% hasta un 100%.*

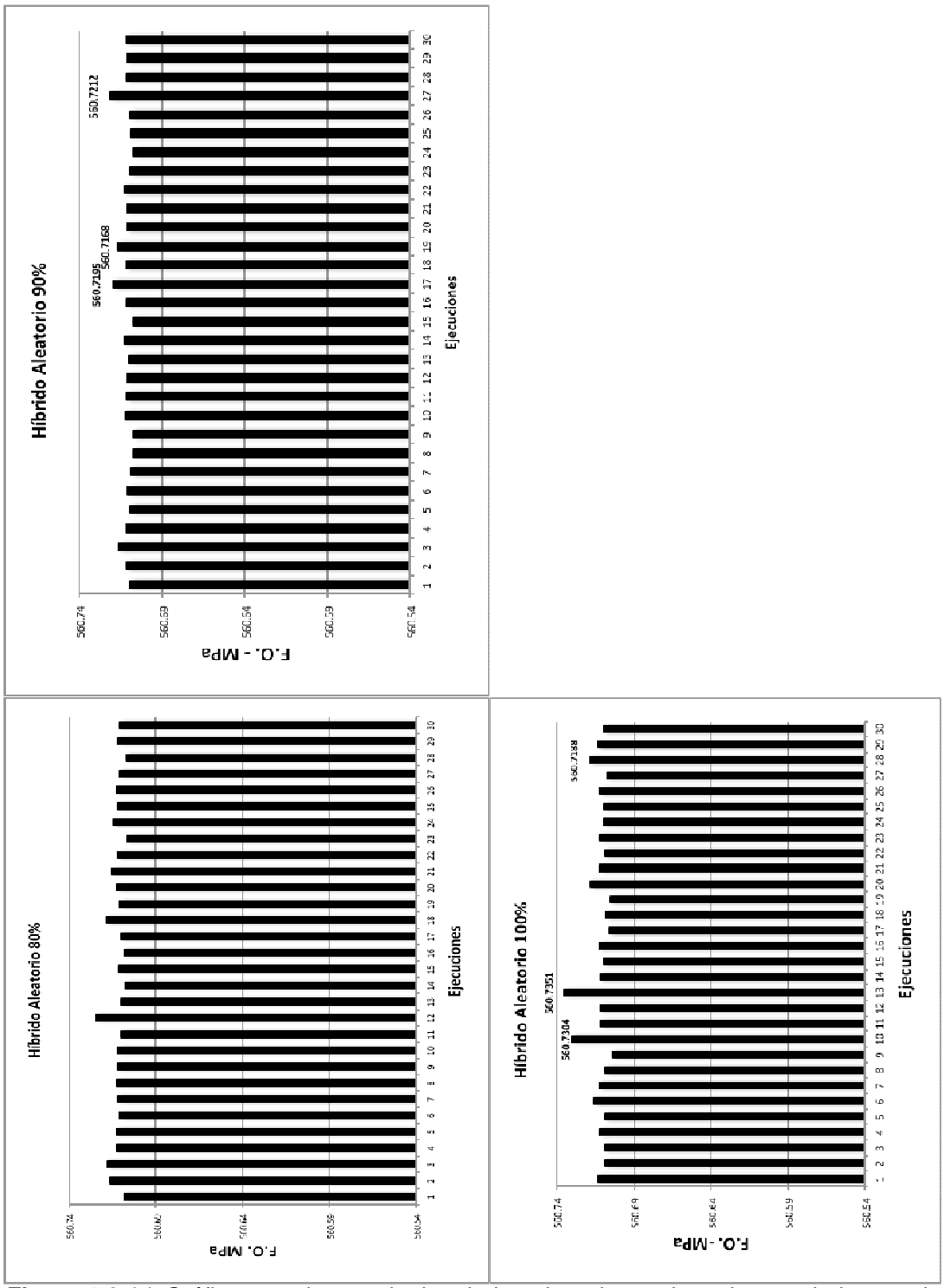


Figura 4-9 *(c) Gráficas con los resultados de las ejecuciones de cada una de las pruebas realizadas a la estructura de vecindad Híbrido Aleatorio las cuales van desde el 5% hasta un 100%.*

Las figuras 4-6 (a, b y c) muestran los valores que se generaron en cada uno de los diferentes tamaños de vecindad, desde el 5% hasta el 100%. Ahí podemos observar los mejores valores que arrojan cada uno de ellos y así como la semejanza que muestran los tamaños de vecindad al 10%, 40% y 50% en adelante. Podemos ver también la disparidad que existe en los tamaños del 5%, 20% y 30%.

Una vez analizada esta información y con la firme convicción de que será usada la estructura de vecindad Híbrida aleatoria con un tamaño de vecindad al 100%, procedemos a realizar el análisis de sensibilidad para aplicar todos estos datos al Algoritmo de Recocido Simulado.

4.3 Análisis de Sensibilidad

La metodología utilizada para el análisis de sensibilidad realizado en el Algoritmo de Recocido Simulado, fue explicada en el capítulo anterior. Recapitulando tenemos:

1. Selección de los Parámetros de Control.

Las variables utilizadas para llevar a cabo el análisis de sensibilidad del Algoritmo de Recocido Simulado son las siguientes:

- Parámetro de control valor inicial (T_0)
- Parámetro de control valor final o criterio de paro (T_f)
- Coeficiente de control (α)
- Longitud de la Cadena de Markov (LCM)

2. Establecer Rangos de Evaluación.

Una vez identificados los parámetros de control, se deben establecer los rangos para realizar el análisis de sensibilidad a cada una de las variables mostradas en el paso 1.

En el análisis realizado a cada una de las variables se observó que si α tiene un valor cercano a 1, es decir 0.994 o 0.999, el parámetro de control T_0 disminuirá lentamente pero, si este es muy próximo a 0, como 0.965, T_0 disminuirá rápidamente alcanzando, de la misma forma, el criterio de paro que es T_f. En la revisión realizada en la literatura donde se aplica el Algoritmo de Recocido Simulado, [Van Laarhoven et al., 1992] indica que, se tome como valor de la longitud de la cadena de Markov L_k, el tamaño de la vecindad.

Tabla 4-3 *Rangos utilizados para realizar el análisis de sensibilidad al Algoritmo de Recocido Simulado.*

Parámetros de Control	Límite Superior	Límite Inferior
T_0	**100019**	**1000**
α	**0.994**	**0.965**
T_f	**2**	**0.0001**
LCM	**989000**	

3. Pruebas a Rangos de Evaluación.

Tomando en cuenta los rangos que permitirán evaluar el comportamiento del algoritmo cuando los parámetros de control tomen distintos valores, se calcularon, para cada uno de los rangos mostrados en la tabla anterior, un

mínimo de 30 pruebas.

Una vez evaluado el valor encontrado para la primera variable, se fija y se comienza con la variación de otro parámetro. El primer valor que se sometió a prueba fue T_0.

Con la primer variable fija, la segunda variable a sintonizar es α partiendo con un valor de 0.98 [Cruz-Chávez, 2005], con los resultados obtenidos, se selecciona el mejor valor encontrado en las 30 pruebas y se fija, como la anterior, para encontrar el siguiente valor.

Para encontrar el valor del parámetro T_f se parte de 0.001 con los incrementos por arriba y por debajo de este valor. Se realizan las 30 pruebas y se obtiene el valor con el que se ejecutará (Tabla 4-2).

La Longitud de la Cadena de Markov representa el tamaño de la vecindad que usará la estructura de vecindad con el Algoritmo de Recocido Simulado. Este valor fue el primero en asignarse.

Tabla 4-4 *Valor de la Longitud de la Cadena de Markov (LCM) con los incrementos y decrementos utilizados para los rangos de los parámetros de control utilizados para el análisis de sensibilidad.*

Parámetros de Control	Incremento	Decremento
T_0	**311**	**311**
α	**0.001**	**0.001**
T_f	**0.143**	**0.00006**
LCM	**989000**	

Una vez fijados todos los valores para los parámetros de control, se generan, nuevamente, 30 pruebas más.

4. Sintonización de Parámetros.

A los valores obtenidos para cada uno de los parámetros de control al terminar la evaluación para cada una de las muestras, serán considerados como los valores de sintonización. La sintonización se lleva a cabo con la finalidad de identificar los valores correspondientes a las variables de control que influirán (Tabla 4-5), de manera positiva, en la calidad de la solución, esto permitirá obtener una mejora en el desempeño del algoritmo [Martínez-Oropeza, 2010].

Tabla 4-5 *Valores de los parámetros de control sintonizados para el Algoritmo de Recocido Simulado de acuerdo al análisis de sensibilidad.*

Parámetros de Control	Valor Sintonizado
T_0	**8775**
α	**0.971**
T_f	**0.00052**
LCM	**989000**

4.4 Análisis de Eficacia y Eficiencia del Algoritmo de Recocido Simulado

Este análisis se lleva a cabo para evaluar la calidad de las soluciones obtenidas por el algoritmo, así como los recursos y tiempo que necesitaron

para su ejecución.

Con el análisis realizado, utilizando Búsqueda Local Iterada y Algoritmo de Recocido Simulado, se demuestra que en un intervalo de la ejecución del Algoritmo de Recocido Simulado se comporta como un Algoritmo de Búsqueda Local Iterada quedando demostrado que, para este tipo de problemas el Algoritmo de Recocido Simulado no es apto ya que es mucho mejor el Algoritmo de Búsqueda Local Iterada. Ambos utilizan la estructura de vecindad híbrida aleatoria, la cual rindió los mejores resultados.

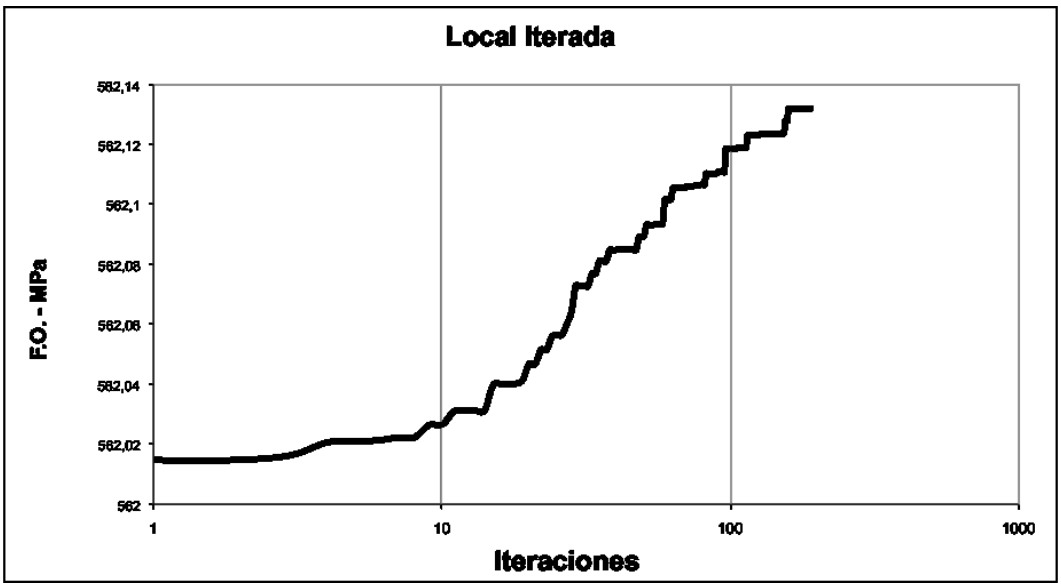


Figura 4-10 *Gráfica con la ejecución de la estructura de vecindad Híbrida en el algoritmo de búsqueda iterada realizada en el clúster ciicap.*

En la gráfica de la figura 4-6 se muestra el comportamiento de la Búsqueda Local Iterada. La base de las x's se encuentra en forma logarítmica para una mejor interpretación ya que la gráfica de la figura 4-8 también se encuentra con esa misma base.

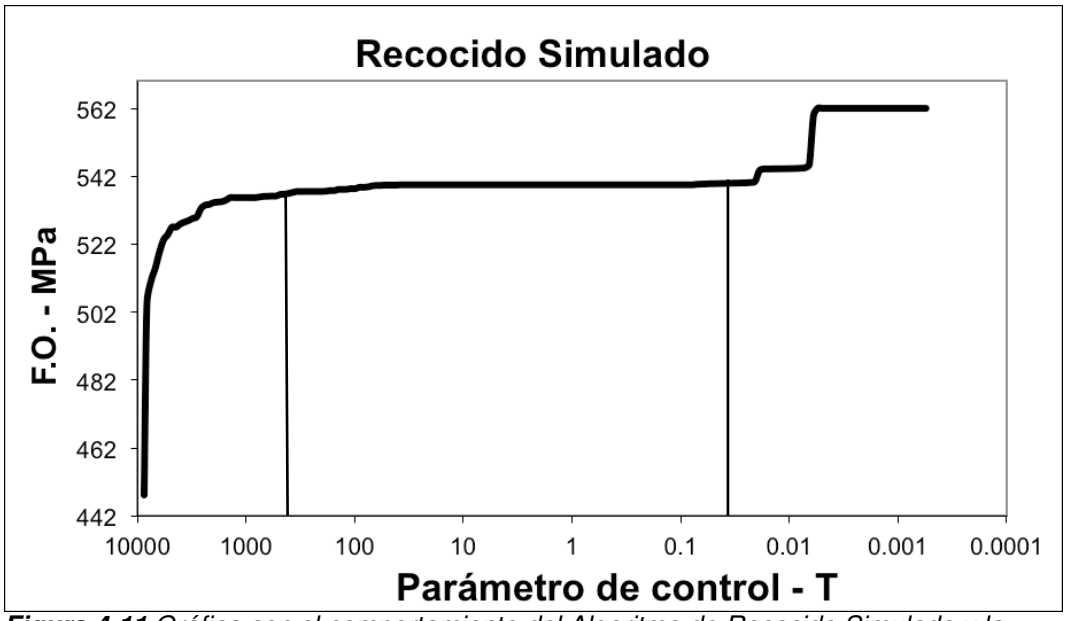


Figura 4-11 *Gráfica con el comportamiento del Algoritmo de Recocido Simulado y la estructura de vecindad Híbrida.*

La parte logarítmica que se muestra en las gráficas representa el número de búsquedas locales que se están haciendo (para el caso de la Búsqueda Local Iterada), cada iteración equivale a una de metrópolis. La ventaja para el Algoritmo de Recocido Simulado es que inicia con un valor muy elevado del parámetro de control.

Observamos en la gráfica el intervalo del parámetro de control donde el valor del Algoritmo de Recocido Simulado en las 30 pruebas realizadas es constante. Donde más aumenta el valor de la función objetivo es a partir de 0.001 hasta el final y en la parte donde se dispara está dando prácticamente el 99% de aceptación de soluciones buenas.

Este comportamiento es muy semejante a lo que es el Algoritmo de Búsqueda Local Iterada y es en esta parte también que se nota que el Algoritmo de Recocido Simulado está aceptando soluciones malas y buenas

al 50% y al hacerlo tiene una mejoría rapidísimo de 440 a 540 en la función objetivo pero hay un intervalo donde no mejora. Ese intervalo se puede quitar y al hacerlo, el Algoritmo de Recocido Simulado estaría mejorando su desempeño notablemente y sólo así se podría decir que el Algoritmo de Recocido Simulado es apto para este tipo de problemas.

El intervalo que se muestra en la gráfica 4-11 no es necesario pero para quitar ese intervalo debemos basarnos en el tiempo de ejecución del algoritmo. Es decir, necesitamos calcular el tiempo que se lleva, en las 30 pruebas, ese intervalo constante y de esta forma al quitar ese tiempo el Algoritmo de Recocido sería más eficiente porque tardaría menos y daría lo mismo (en MPa). Son 30 pruebas, casi es la mitad o más de la mitad el tiempo que se le quitaría y al hacerlo haría mucho más eficiente el Algoritmo de Recocido Simulado.

La ventaja del Algoritmo de Recocido Simulado es que puede detectar esa parte y entonces podemos quitar ese segmento del parámetro de control y pasar de 1000 a 50 e inmediatamente reducir el valor cerca a 0.01, es entonces que estaríamos en ese punto, lo haríamos de forma experimental de manera que cuando llegue a 50 se pase a un valor antes de 0.01, si es así, el tiempo se reduce a la mitad y entonces sería más eficiente el Algoritmo de Recocido Simulado que el Algoritmo de Búsqueda Local Iterada.

La mejor configuración obtenida al realizar las pruebas experimentales (Tabla 4-6), fue con el Algoritmo de Búsqueda Local Iterada (ILS). Es por eso que para este trabajo de investigación, se propone hacer uso de dicho algoritmo.

Tabla 4-6 *Mejor configuración obtenida y el resultado de la resistencia en MPa.*

SOLUCION MEJOR-Sm:	
Fe:	97.4223
C:	0.08
Cr:	0.0999
Cu:	0.6
Mn:	1.2
Mo:	0.03
Ni:	0.0999
V:	0.1499
N:	0.001
Nb:	0.001
Ti:	0.001
P:	0.013
S:	0.002
Si:	0.3

$\sigma_y MPa = 562.1482$

La ciencia humana consiste más en destruir errores que en descubrir verdades
Sócrates

Capítulo 5

Conclusiones y Trabajos Futuros

5.1 Conclusiones

De acuerdo a los resultados experimentales obtenidos, el Algoritmo de Recocido Simulado no funcionó como se esperaba ya que en todo el intervalo del parámetro de control, no mejora sustancialmente la función objetivo. Esto puede indicar que la función de Boltzmann no está trabajando como debe en un intervalo del parámetro de control y al eliminar dicha función nos queda un algoritmo de búsqueda local iterada. Se comprueba que la búsqueda local iterada no mejora sustancialmente los resultados realizados con respecto a Recocido Simulado, pero si es mejor en 30 pruebas. Podemos observar que en un intervalo grande del parámetro de control (50 a 0.01), el Algoritmo de Recocido Simulado no mejora la solución que se tiene y en el resto del intervalo del parámetro de control (10,000 a 100 y de 0.01 a 0.0001) se comporta como un algoritmo de búsqueda local iterada.

Por lo tanto, para este tipo de problemas, el Algoritmo de Recocido Simulado no es recomendable. Lo que sí se recomienda, para problemas de este tipo, es hacer uso del algoritmo de Búsqueda Local Iterada debido a que en eficacia son casi iguales pero en eficiencia el algoritmo de Búsqueda Local Iterada le gana al Algoritmo de Recocido Simulado precisamente porque no hace uso de una función exponencial y eso hace que el Algoritmo de Recocido Simulado sea menos eficiente que la Búsqueda Local Iterada.

5.2 Trabajos Futuros

Aplicar la estructura de vecindad Híbrida, utilizada en el Algoritmo de Recocido Simulado, en alguna otra metaheurística como genéticos pero, en un ambiente Grid (Tecnología que permite utilizar de forma coordinada todo tipo de recursos como cómputo, almacenamiento y aplicaciones específicas) haciendo uso del Algoritmo de Búsqueda Local Iterada. Paralelizar el algoritmo con programación en GPU's. Además, mejorar la Función Objetivo del problema para maximizar la resistencia mecánica en aceros microaleados incluyendo un número mayor de variables como lo es el tamaño de grano, la cantidad de precipitados y la composición química. Finalmente, aplicar este mismo problema con diferentes instancias.

Cree a aquellos que buscan
la verdad; duda de los
que la han encontrado
André Gide

Referencias

Aarts, E.H.L. and Van Laarhoven, P.J.M. 1985. Statistical cooling: A general approach to combinatorial optimization problems. Philips Journal of Research, 40(4), 193-226.

Askeland R. D., Phulé P.P. (2004). "*Ciencia e Ingeniería de los materiales*". 4a. Edición. ISBN 970-686-361-3.

Cruz-Chávez, M. A, Martínez O. A. (2010). "*Estructura Híbrida de Vecindad para Problemas de Optimización Discreta".* CERMA2010, IEEE-Computer Society, ISBN 978-0-7695-4204-1, pp, Septiembre 28 – Octubre 1, México, 2010.

Cruz-Chávez, M. A. (2005). *"Cooperación de Procesos para el Problema de Job Shop Scheduling Aplicando Recocido Simulado".* Tesis de Doctorado. Marzo 2005.

Glover F., Kochenberger G. (2002), Editors, "*Handbook of metaheuristics; International Series in Operations Research and Management Science".* vol. 57; Kluwer Academic Publishers, Norwell, MA, 321-353.

Joyanes, A. L., Zahonero, M. I. (2000) "*Programación en C: Metodología, algoritmos y estructura de datos".* ISBN: 8448124871. McGraw-Hill, Primera Edición 2000.

Kirkpatrick S., Gelatt S. D. Jr., Vecchi M. P. (1983). *"Optimization by Simulated Annealing"*. Science, 220 (4598), 13 May, pp. 671-680, 1983.

Martínez-Oropeza. A. (2010). *"Solución al Problema de Máquinas en Paralelo No Relacionadas Mediante un Algoritmo de Colonia de Hormigas"*. Tesis de Maestría. Agosto 2010.

Martínez-Rangel. M. G. (2008). *"Algoritmo de Recocido Simulado Paralelizado Aplicado al Problema de Asignación de Recursos en un*

Taller de Manufactura Flexible Sujeto a Disposiciones de Tiempo". Tesis de Doctorado. Diciembre 2008.

Michaelewicz Z., Fogel D. B. (2004). "*How to Solve it: Modern Heuristics"*. Springer –Verlag Berlin Heidelberg 2000, 2004. Germany.

Morrison, W. B. (2000) "*Past and Future Development of HSLA Steels"*. The Fourth International Conference on HSLA Steels. October 30- November 2, 2000. Xi'an, China.

Osman, I.H., Kelly, J.P. (1996). "*Meta-Heuristics: Theory and Applications*", 39 Kluwer Academic Publishers. ISBN: 0792397002. USA, 1996.

Papadimitriou C. H., Steiglitz K. (1998). "*Combinatorial Optimization: Algorithms and Complexity*". ISBN. 0-486-40258-4. USA. 1998. Mineola NY.

Steve, W., Haynes, A. G. (1956), J. Iron Steel Inst. London 183, 349. Book: Material Science and Technology Vol. 7 Constitution and Properties of Steel. Weinheim; New York; Basel; Cambridge. 1992. ISBN: 3-527-26813-8 (Weinheim); ISBN: 1-56081-190-0 (New York).

Talbi, El-Ghazali. (2009). "*Metaheuristics, From Design To Implementation".* University of Lille – CNRS – INRIA ISBN 978-0-470-27858-1. Published by John Wiley & Sons, Inc., Hoboken, New Jersey.

Xu W., Rivera-Díaz-del-Castillo P. E. J., Van der Zwaag S. (2008a). "*Designing nanoprecipitation strengthened UHS stainless steels combining genetic algorithms and thermodynamics*". Computational Materials Science 44 (2008) 678–689.

Xu W., Rivera-Díaz-del-Castillo P. E. J., Van der Zwaag S. (2008b). "*Genetic alloy design based on thermodynamics and kinetics*", Philosophical Magazine, 88: 12, 1825 — 1833.

Xu W., Rivera-Díaz-del-Castillo P. E. J., Van der Zwaag S. (2009). "*A combined optimization of alloy composition and aging temperatura in designing new UHS precipitation hardenable stainless steels*". Computational Materials Science 45 (2009) 467–473.

Moreno, Alfredo. "*Programar es fácil (o no)".* http://profeblog.es/blog/alfredo/2008/03/24/funciones-y-paso-de parametros-en-c/. *Un blog de Alfredo Moreno sobre programación de ordenadores para humanos y similares.*

Van Laarhoven, P.J.M., Aarts, E.H.L. and Lenstra, J.K., (1992). Job shop scheduling by simulated annealing. Operations Research, 40, 113–126.

Apéndice A

Estructuras de Datos Utilizadas en el Algoritmo de Recocido Simulado

Una estructura contiene múltiples variables, que pueden ser de tipos diferentes. La estructura es importante para la creación de programas potentes y que requieran grandes cantidades de datos. Un tipo de dato enumerado es una colección de miembros con nombre que tienen valores enteros equivalentes. Un typedef es de hecho no un nuevo tipo de dato sino simplemente un sinónimo de un tipo existente [Joyanes, Zahonero, 2000].

Para el desarrollo del programa Algoritmo de Recocido Simulado fue necesario anticiparse creando una estructura de datos con los componentes de los elementos químicos. La composición química presentada en este trabajo de investigación, contiene en su estructura los elementos que componen un acero microaleado. El contenido de cada uno de los elementos servirá de base para calcular la resistencia en Mega Pascales y para analizar y comparar el mejor resultado devuelto por la Función Objetivo.

```
typedef struct {
        char *cNom_Elem_Quimico;
        float fPorc_Elem;
        float fLB;
        float fUB;
}tconfig_elemen;
```

char *cNom_Elem_Quimico: contiene los nombres de cada uno de los elementos químicos que componen el acero. Es un apuntador de tipo carácter.

float fPorc_Elem: contiene el porcentaje asignado a cada elemento químico. Existen algunos valores que fueron declarados constantes dentro de la estructura. El tipo de dato es flotante ya que las cantidades
se manejan en porcentajes de acuerdo a su peso.

float fLB y fUB: se utiliza este campo para manejar el rango de cada uno de los elementos y, para que en el momento de asignarlo en forma aleatoria, no rebase o se salga del rango establecido.

tconfig_elemen: es el nombre del tipo de dato creado a partir de la estructura. Con él, se crean los cuatro arreglos que se necesitarán para el desarrollo del Algoritmo de Recocido Simulado.

Apéndice B

Función Temporal de la estructura de vecindad Doble Aleatorio

```
for(;;)
        {
          iposic1= rand()%8;
          iposic2= rand()%8;
          icomp1=iComprobar_Celdas(iposic1);
          icomp2=iComprobar_Celdas(iposic2);

          if(iposic1!=iposic2)
          {
                    iSelecc=rand()%2
                    if (iSelecc==1)
                    {
                              isigno=1;          /* isigno =  0: Dec  1:Inc */
                              icomp1=iAumDism(tAc_Mic2, iposic1,isigno);
                              isigno=0;
                              icomp2=iAumDism(tAc_Mic2, iposic2,isigno);
                    }
                    else
                    {
                              isigno=0;          /* isigno =  0: Dec  1:Inc */
                              icomp1=iAumDism(tAc_Mic2, iposic1,isigno);
                              isigno=1;
                              icomp2=iAumDism(tAc_Mic2, iposic2,isigno);
                    }
                    if((icomp1==icomp2)&&(icomp1&&icomp2!=0))
                              break;
          }
        }/*Fin for*/
```

$$T(e)=15e+149$$

La función temporal $T(e)=15e+149$ para la estructura de vecindad Doble Aleatorio, es de tipo Polinomial y consta de una cota superior en la cual la constante está determinada por **c**. La premisa de la función temporal para el Peor de los casos es:

$$O(g(e)) = \{ f(e) \mid \exists c, e_0, \forall e \geq e_0, c_1 g(e) \leq f(e) \leq c_2 g(e) \}$$

$$f(e) = 15e + 149 = O(e) si \exists c: \quad 0 \leq 15e + 149 \leq ce^2 \qquad e \geq e_0 = 20$$

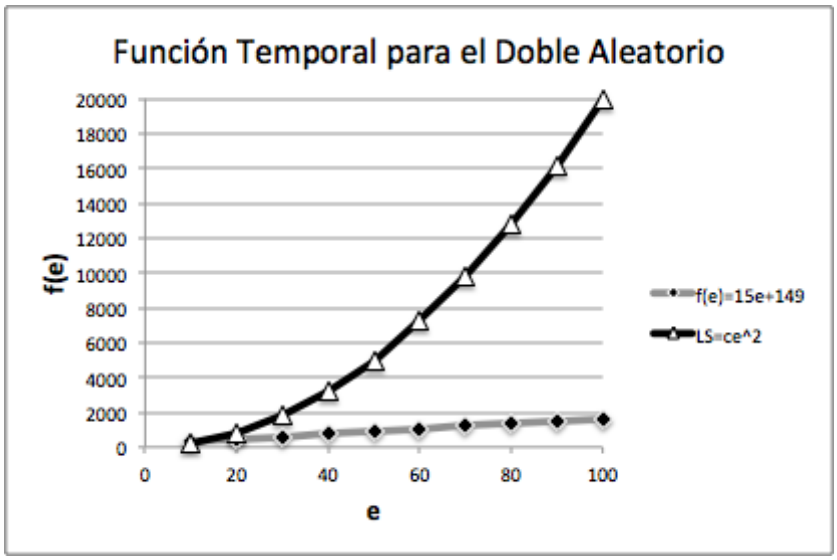


Figura A-1 *Gráfica con la función temporal de la estructura Doble Aleatorio.*

Función Temporal de la estructura de vecindad Triple Aleatorio

```
for(;;)
        {
          iposic1= rand()%8;
          iposic2= rand()%8;
          iposic3= rand()%8;

          icomp1=iComprobar_Celdas(iposic1);
          icomp2=iComprobar_Celdas(iposic2);
          icomp3=iComprobar_Celdas(iposic2);

          expr=(iposic1!=iposic2);
          expr1=(iposic1!=iposic3);
          expr2=(iposic2!=iposic3);

          if(expr&&expr1&&expr2==1)
          {
                    iSelecc=rand()%2;

                    if (iSelecc==1)
                    {
                              isigno=1;  /* 2 INC - 1 DEC */
```

```
                    icomp1=iAumDism(tAc_Mic2, iposic1,isigno);
                    icomp2=iAumDism(tAc_Mic2, iposic2,isigno);
                    isigno=0;
                    icomp3=iAumDism(tAc_Mic2, iposic3,isigno);
            }
            else
            {
                    isigno=0;    /* 2 DEC - 1 INC */
                    icomp1=iAumDism(tAc_Mic2, iposic1,isigno);
                    icomp2=iAumDism(tAc_Mic2, iposic2,isigno);
                    isigno=1;
                    icomp3=iAumDism(tAc_Mic2, iposic3,isigno);
            }
            if((icomp1==icomp2==icomp3)&&(icomp1&&icomp2&&icomp3!=0))
                    break;
    }
}/*Fin for*/
```

$$T(e) = 28e + 220$$

La premisa de la función temporal para el Peor de los casos es:

$$O(g(e)) = \{f(e) \mid \exists c, e_0, \forall e \geq e_0, c_1 g(e) \leq f(e) \leq c_2 g(e)\}$$

$$f(e) = 28e + 220 = O(e) si \exists c: \ 0 \leq 28e + 220 \leq ce^2 \qquad e \geq e_0 = 30$$

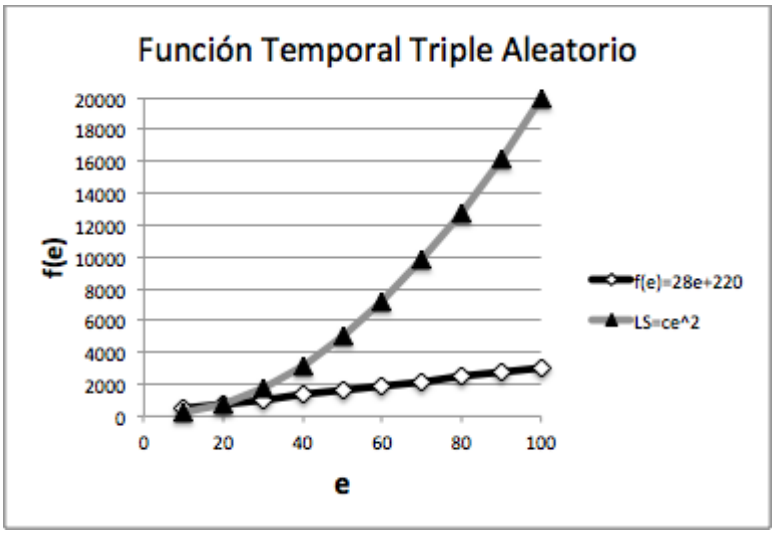


Figura A-2 *Gráfica con la función temporal de la estructura Triple Aleatorio.*

Función Temporal de la estructura de vecindad Cuádruple Aleatorio

```
for(;;)
        {
          iposic1= rand()%8;
          iposic2= rand()%8;
          iposic3= rand()%8;
          iposic4= rand()%8;

          icomp1=iComprobar_Celdas(iposic1);
          icomp2=iComprobar_Celdas(iposic2);
          icomp3=iComprobar_Celdas(iposic3);
          icomp4=iComprobar_Celdas(iposic4);

          expr= (iposic1!=iposic2);
          expr1=(iposic1!=iposic3);
          expr2=(iposic1!=iposic4);
          expr3=(iposic2!=iposic3);
          expr4=(iposic2!=iposic4);
          expr5=(iposic3!=iposic4);

          if(expr&&expr1&&expr2&&expr3&&expr4&&expr5==1)
          {
                    iSelecc=rand()%2;

                    if (iSelecc==1)
                    {
                    isigno=1;                    /* 2 INC - 2 DEC */
                    icomp1=iAumDism(tAc_Mic2, iposic1,isigno);
                    icomp2=iAumDism(tAc_Mic2, iposic2,isigno);
                    isigno=0;
                    icomp3=iAumDism(tAc_Mic2, iposic3,isigno);
                    icomp4=iAumDism(tAc_Mic2, iposic3,isigno);
                    }
          else
                    {
                              isigno=0;                    /* 2 DEC - 2 INC */
                              icomp1=iAumDism(tAc_Mic2, iposic1,isigno);
                              icomp2=iAumDism(tAc_Mic2, iposic2,isigno);

                              isigno=1;
                              icomp3=iAumDism(tAc_Mic2, iposic3,isigno);
                              icomp4=iAumDism(tAc_Mic2, iposic3,isigno);
                    }

          if((icomp1==icomp2==icomp3==icomp4)&&(icomp1&&icomp2&&icomp3&&icomp4!=
        0))
          break;
          }
        }/*Fin for*/
}/*Fin func*/
```

$$T(e) = 41e + 295$$

La premisa de la función temporal para el Peor de los casos es:

$$O(g(e)) = \{f(e) \mid \exists c, e_0, \forall e \geq e_0, c_1 g(e) \leq f(e) \leq c_2 g(e)\}$$

$$f(e) = 41e + 295 = O(e) si \exists c: \ 0 \leq 41e + 295 \leq ce^2 \qquad e \geq e_0 = 40$$

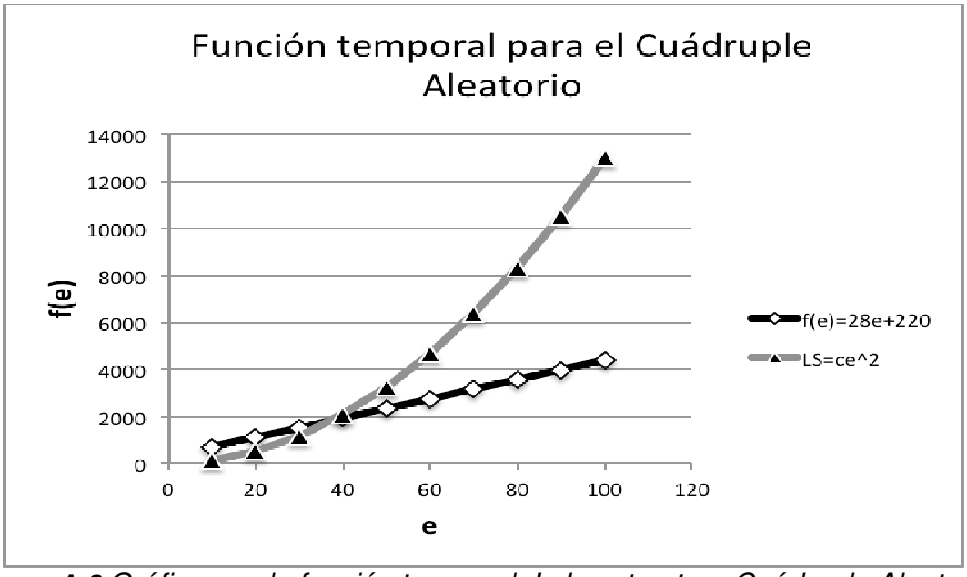


Figura A-3 *Gráfica con la función temporal de la estructura Cuádruple Aleatorio.*

Función Temporal de la estructura de vecindad Quíntuple Aleatorio

```
for(;;)
        {
          iposic1= rand()%8;
          iposic2= rand()%8;
          iposic3= rand()%8;
          iposic4= rand()%8;
          iposic5= rand()%8;

          icomp1=iComprobar_Celdas(iposic1);
          icomp2=iComprobar_Celdas(iposic2);
          icomp3=iComprobar_Celdas(iposic2);
          icomp4=iComprobar_Celdas(iposic2);
```

```
        icomp5=iComprobar_Celdas(iposic2);

        expr= (iposic1!=iposic2);
        expr1=(iposic1!=iposic3);
        expr2=(iposic1!=iposic4);
        expr3=(iposic1!=iposic5);
        expr4=(iposic2!=iposic3);
        expr5=(iposic2!=iposic4);
        expr6=(iposic2!=iposic5);
        expr7=(iposic3!=iposic4);
        expr8=(iposic3!=iposic5);
        expr9=(iposic4!=iposic5);

      if(expr&&expr1&&expr2&&expr3&&expr4&&expr5&&expr6&&expr7&&expr8&&expr9==
1)
        {
                iSelecc=rand()%2;
                if (iSelecc==1)
                {
                        isigno=1;   /* 3 INC -  2 DEC */
                        icomp1=iAumDism(tAc_Mic2, iposic1,isigno);
                        icomp2=iAumDism(tAc_Mic2, iposic2,isigno);
                        icomp3=iAumDism(tAc_Mic2, iposic2,isigno);
                        isigno=0;
                        icomp4=iAumDism(tAc_Mic2, iposic3,isigno);
                        icomp5=iAumDism(tAc_Mic2, iposic3,isigno);
                }
                else
                {
                        isigno=0;          /* 3 DEC -  2 INC */
                        icomp1=iAumDism(tAc_Mic2, iposic1,isigno);
                        icomp2=iAumDism(tAc_Mic2, iposic2,isigno);
                        icomp3=iAumDism(tAc_Mic2, iposic3,isigno);
                        isigno=1;
                        icomp4=iAumDism(tAc_Mic2, iposic3,isigno);
                        icomp5=iAumDism(tAc_Mic2, iposic3,isigno);
                }

      if((icomp1==icomp2==icomp3==icomp4==icomp5)&&(icomp1&&icomp2&&icomp3&&ic
omp4&&icomp5!=0))
                        break;
        }
      }/*Fin for*/
```

$$T(e) = 56e + 367$$

La premisa de la función temporal para el Peor de los casos es:

$$O(g(e)) = \{f(e) \mid \exists c, e_0, \forall e \geq e_0, c_1 g(e) \leq f(e) \leq c_2 g(e)\}$$

$$f(e) = 56e + 367 = O(e) si \exists c: \ 0 \leq 56e + 367 \leq ce^2 \qquad e \geq e_0 = 50$$

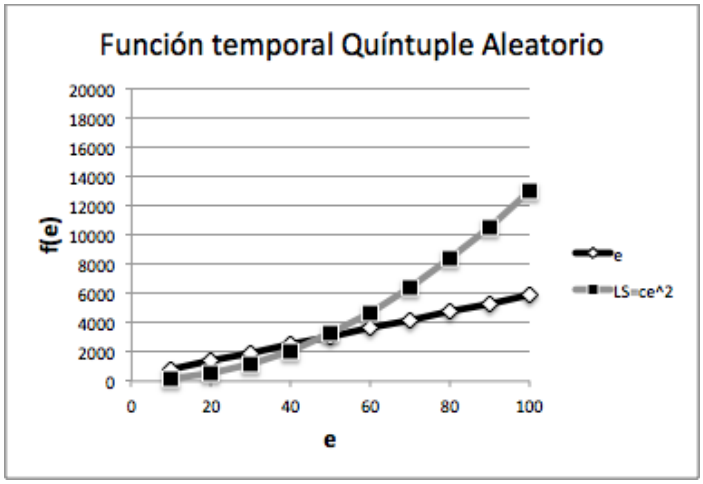


Figura A-4 *Gráfica con la función temporal de la estructura Quíntuple Aleatorio.*

Complejidad del Algoritmo de Recocido Simulado

De acuerdo con Aart y Van Laarhoven (1985), la complejidad computacional de un algoritmo de recocido simulado presenta la ecuación:

$$O(\tau L \ln R)$$

las variables, para este problema de investigación, quedan de la siguiente forma:

τ = **e** que representa el número de los elementos de la composición química del acero.

$L = (n-7)m$ representa el número de incrementos que se le hace a cada elemento y equivale al tamaño de la vecindad.

$R = m^{n-7}$ es el espacio de soluciones.

Por lo tanto, la complejidad computacional del algoritmo de recocido simulado para el problema de Maximizar la Resistencia de Aceros Microaleados, queda como sigue:

$$O\left(n + nm(n-7)^2 \ln m\right)$$

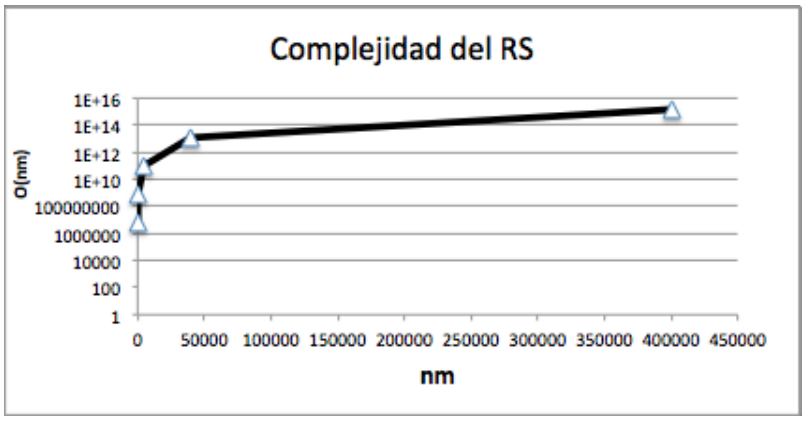


Figura A-5 *Gráfica con la Complejidad del Algoritmo de Recocido Simulado.*

La complejidad del algoritmo es de tipo polinomial, en la gráfica se observa su comportamiento.

A continuación, se muestra únicamente la función con el Algoritmo de Recocido Simulado.

```
float f_ARS(tconfig_elemen *tAc_Mic, tconfig_elemen *tAc_Mic1,
tconfig_elemen *tAc_Mic2, tconfig_elemen *tSm_ARS, tconfig_elemen
*tSm_LS)

int  j, iLM=989000, contador=0, iterM=0;
double dS=0, dSp=0, dResta=0, dPaceptar=0, dPotencia=0, dalfa=0;
float fSm_ARS=0;
double dBeta=0.971, dT=0, dTf=0.00052;

vCopy_Struct(tAc_Mic1,tAc_Mic);

for(dT=8775; dT>=dTf; dT *= dBeta )
```

```
{
      for (j=0; j<=iLM; j++)
      {
        fBusq_Hibrida(tAc_Mic1,tAc_Mic2);

        dS=fEval_Func_Objetivo(tAc_Mic1);
         dSp=fEval_Func_Objetivo(tAc_Mic2);

       if((dSp-dS) >= 0)
        {
                vCopy_Struct(tAc_Mic1,tAc_Mic2);

                if(dSp>fSm_ARS)
                {
                        fSm_ARS=dSp;
                        vCopy_Struct(tSm_ARS,tAc_Mic2);  /*** ARS ***/
                }
        }
         else
        {
                dResta = dSp - dS;
                dPotencia = dResta / dT;
                dPaceptar=exp(dPotencia);
                iMetropolis(iterM, dT, fSm_ARS);
                contador=0;
        }
}
iArchConfig_Sm(tSm_ARS);
return fSm_ARS;}
```

Glosario de Términos

Algoritmo Determinístico: Es un algoritmo en el que una misma entrada obtiene siempre el mismo resultado.

Algoritmo no Determinístico: Algoritmo en el que a una misma entrada se obtienen diferentes salidas, de modo que no es posible saber de manera previa, el resultado que de la ejecución de un algoritmo de este tipo.

Análisis de Sensibilidad: Evaluación del comportamiento de las variables críticas de un problema con la finalidad de establecer un rango numérico, dentro del cuál, la solución obtenida por el algoritmo sigue siendo buena, además de que permite conocer que tan sensible es el algoritmo a cambios en los valores de ciertas variables propias del problema.

Búsqueda Local: Técnica iterativa de que permite explorar el espacio de soluciones de un problema dado, a partir de una solución inicial, por medio de movimientos, de modo que vaya mejorando la solución obtenida de acuerdo a la función objetivo. Este tipo de técnicas son utilizadas para mejorar la calidad de las soluciones obtenidas por un algoritmo, así como para reducir el tiempo en que se obtienen dichas soluciones.

Complejidad Espacial: Es la cantidad de memoria requerida por el algoritmo durante la ejecución.

Complejidad Temporal: Es el número de pasos necesarios (tiempo) para obtener una solución a una instancia de un problema dado.

Costo: Para la investigación del ARS, el costo se refiere al valor devuelto por la Función Objetivo. El valor se toma en la medida de los Mega Pascales

para verificar la resistencia.

Estructura de Vecindad: Tipo de movimiento utilizado para explorar el espacio de soluciones de un problema dado.

Heurística: Procedimiento intuitivo bien definido que proporciona buenas soluciones aproximadas a problemas difíciles de resolver sin garantizar la optimalidad, en un tiempo computacional razonable.

Metaheurísticas: Métodos aproximados que mejoran procedimientos heurísticos, los cuales son diseñados para ser aplicados a problemas considerados difíciles de resolver, donde las heurísticas no son eficientes.

Óptimo Global: Es la mejor solución de un espacio de soluciones f(x).

Óptimo Local: Representa la mejor solución de f(x) en un entorno x.

Recocido Simulado (SA): Algoritmo que realiza una búsqueda aleatoria orientada, el cuál hace una analogía del proceso simulado de fundición de metal, de modo que la temperatura va descendiendo gradualmente hasta que su energía mínima se alcanza. De modo que permite escapar de óptimos locales al aceptar algunas soluciones consideradas malas, obteniendo muy buenos resultados en diversos problemas de optimización a los que se ha aplicado.

Sintonización: Es la proporción adecuada en cuanto a los valores obtenidos mediante el análisis de sensibilidad aplicado a los parámetros de control, tomando en cuenta el problema y el método de optimización utilizado, de modo que el algoritmo muestre una mejora tanto en eficiencia como en eficacia.

Tiempo polinomial: En las ciencias computacionales, el tiempo viene expresado generalmente en función del tamaño de la entrada. Se considera que un algoritmo puede ser resuelto en tiempo polinomial sí su función de complejidad es de orden O(p(n)), es decir, que puede ser representado por un polinomio.

Vecindad: Es el conjunto de soluciones que pueden ser alcanzadas desde una solución dada por medio de un movimiento. Para el problema de la resistencia son los incrementos y decrementos de los elementos de la composición química.