

**DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN**



**“METAHEURÍSTICO DE BÚSQUEDA LOCAL PARA LA  
CONFIGURACIÓN ÓPTIMA DE GRUPOS DE LENNARD-JONES”**

**PARA OBTENER EL GRADO DE:**

**MAESTRO EN CIENCIAS  
EN CIENCIAS DE LA COMPUTACIÓN**

**P R E S E N T A:**

**I.S.C. GEORGINA CASTILLO VALDEZ**

**D I R E C T O R:**

**DR. HÉCTOR JOAQUÍN FRAIRE HUACUJA**

**C O D I R E C T O R:**

**M.C. GUADALUPE CASTILLA VALDEZ**

**DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN**



**“METAHEURÍSTICO DE BÚSQUEDA LOCAL PARA LA  
CONFIGURACIÓN ÓPTIMA DE GRUPOS DE LENNARD-JONES”**

**PARA OBTENER EL GRADO DE:  
MAESTRO EN CIENCIAS EN  
CIENCIAS DE LA COMPUTACIÓN**

**P R E S E N T A:**

**I.S.C. GEORGINA CASTILLO VALDEZ**

**D I R E C T O R:**

**DR. HÉCTOR JOAQUÍN FRAIRE HUACUJA**

**C O D I R E C T O R:**

**M.C. GUADALUPE CASTILLA VALDEZ**

**J U R A D O:**

**Presidente: DR. HÉCTOR JOAQUÍN FRAIRE HUACUJA**

**Secretario: DR. JOSÉ ANTONIO MARTÍNEZ FLORES**

**Vocal: DR. ARTURO HERNÁNDEZ RAMÍREZ**

**Suplente: M.C. CLAUDIA GUADALUPE GÓMEZ SANTILLÁN**



"2006, Año del Bicentenario del Natalicio del Benemerito de las Américas, Don Benito Juárez García"

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

COORDINACIÓN DE POSGRADO EN  
COMPUTACIÓN  
U5.494/06  
AUTORIZACIÓN DE IMPRESIÓN  
DE TESIS DE GRADO

2006-NOV-24

C. ING. GEORGINA CASTILLO VALDEZ  
Presente.

Me es grato comunicarle que después de la revisión realizada por el Jurado designado para su examen de grado de Maestra en Ciencias en Ciencias de la Computación, se acordó autorizar la impresión de su tesis titulada:

**"METAHEURÍSTICO DE BÚSQUEDA LOCAL PARA LA CONFIGURACIÓN ÓPTIMA  
DE GRUPOS DE LENNARD-JONES"**

Es muy satisfactorio para la División de Estudios de Posgrado e Investigación compartir con Usted el logro de esta meta. Espero que continúe con éxito su desarrollo profesional y dedique su experiencia e inteligencia en beneficio de México.

ATENTAMENTE  
"POR MI PATRIA Y POR MI BIEN"

*Ma. Yolanda Chávez Cinco*  
M.P. MARÍA YOLANDA CHÁVEZ CINCO  
JEFA DE LA DIVISIÓN



S.E.P.  
DIVISION DE ESTUDIOS  
DE POSGRADO E  
INVESTIGACION  
I.T.C.M.

*Y*  
MYCHC ' NLCO ' cerc\*

Av. 1° de Mayo s/n Pte. Col. Los Mangos  
Cd. Madero, Tam. C. P. 89440  
Internet : [www.itcm.edu.mx](http://www.itcm.edu.mx)  
Correo Electrónico : [itcm@itcm.edu.mx](mailto:itcm@itcm.edu.mx)

Teléfono : 3 - 57 - 48 - 20 al 29  
Fax : 3 - 57 - 48 - 20 ext. 1002

## ***DEDICATORIA***

Dedico esta tesis a Dios,  
a mi mamá Dora Valdez Valdez,  
a mi papá Guadalupe Castillo Torres,  
a mis hermanos José Alfredo,  
Víctor Manuel y Dora Alicia.

## *AGRADECIMIENTOS*

Agradezco a Dios por su infinito amor, por su gran misericordia, por la vida y por la salud.

Agradezco a mis padres y hermanos por el amor, por el apoyo brindado, por su paciencia y por sus consejos.

Mi profundo agradecimiento a los miembros del comité tutorial de esta tesis: Dr. Héctor Joaquín Fraire Huacuja, Dr. José Antonio Martínez Flores, Dr. Arturo Hernández Ramírez y M.C. Claudia Guadalupe Gómez Santillán por sus valiosas sugerencias y observaciones realizadas durante el desarrollo y revisión de esta tesis.

Doy gracias al Dr. Héctor Joaquín Fraire Huacuja y a la M.C. Guadalupe Castilla Valdez por haber dirigido esta tesis y por todo el apoyo brindado para el desarrollo de la misma.

Mi más sincero agradecimiento a todos mis hermanos de Fe, por sus oraciones, por sus consejos y por sus palabras de ánimo.

Agradezco a mis compañeros de maestría: Alberto Castro Hernández, Pedro Torres Espinosa, Carlos Alberto Camacho Andrade, Víctor Manuel Álvarez Hernández, Irma Cristina Cruz Castro y Juan Arturo Herrera Ortiz por su amistad y apoyo en esta etapa de mi vida.

Agradezco el apoyo otorgado por el Consejo Nacional de Ciencia y Tecnología (CONACYT) y por la Dirección General de Educación Superior Tecnológica (DGEST).

## RESUMEN

En este trabajo se aborda el problema de determinar la configuración de mínima energía potencial de los átomos de un grupo de Lennard-Jones. El enfoque de solución más exitoso de este problema consiste en desarrollar métodos de optimización global que incorporan diversas estructuras geométricas, un optimizador local y un algoritmo metaheurístico. Sin embargo las aplicaciones más recientes de este enfoque muestran dos importantes limitaciones: no incorporan optimizadores locales del estado del arte y en los estudios experimentales que realizan no reportan la magnitud de la variación de los resultados. En esta tesis se propone un nuevo método optimización global en el que por primera ocasión se incorporan: un optimizador local de regiones de confianza, el metaheurístico GRASP y un mecanismo de reducción de la varianza de los resultados experimentales. El método logra un 100% de eficiencia en la búsqueda de la mejor energía reportada para los grupos LJ de 13-25 átomos. En 8 de los grupos se logra una reducción del número de evaluaciones de la función objetivo con respecto a lo reportado en [Romero 1999a], en este indicador se logra una reducción global de 0.53%. El método incorpora un mecanismo de reducción de la varianza que permite incrementar la calidad de las mediciones hasta en un 48%. En el desarrollo de este proyecto se han identificado las siguientes líneas de investigación: diseño de rejillas multigeométricas, diseño e implementación de nuevos métodos de optimización global basados en otro tipo de metaheurísticos y la migración del prototipo actual a lenguajes de mayor rendimiento para experimentar con grupos de mayor tamaño.

## SUMMARY

In this work, the problem of determining the lowest potential energy of a set of atoms of Lennard-Jones is approached. The most successful solution approach of this problem consists on the development of global optimization methods which incorporate different geometric structures, a local optimization method and a metaheuristic algorithm. However, the most recent applications of this approach show two important limitations: they do not use local optimization methods from state of art; and the experimental studies they do, do not report the scale of the variation of their results.

In this thesis, a new global optimization method is proposed, where on first occasion different elements are incorporated: a local optimization method of truth regions, the GRASP metaheuristic and a mechanism of variance reduction for the experimental results. The method achieves 100% of efficiency in the search of the best reported energy for the 13-25 atoms LJ groups. Into eight groups, compared with [Romero 1999a], the reduction of the number of evaluations in the objective function is achieved; with this indicator, 0.53% of global reduction is reached. The method incorporates a mechanism of variance reduction which increases the quality of the results, up to 48%. In the development of this project, lines of investigation are identified: design of multigeometric lattice, design and implementation of new global optimization method based on other kind of metaheuristics, and the migration from the current prototype to better performance programming languages for experimenting with larger size groups.

# CONTENIDO

RESUMEN	
CONTENIDO .....	i
LISTA DE FIGURAS .....	iii
LISTA DE TABLAS .....	iv
CAPÍTULO 1. INTRODUCCIÓN .....	1
1.1 Antecedentes.....	1
1.2 Descripción del problema de investigación.....	4
1.3 Justificación.....	4
1.4 Objetivos.....	5
1.4.1 Objetivo general .....	5
1.4.2 Objetivos específicos.....	5
1.5 Hipótesis.....	6
1.6 Organización de la tesis.....	6
CAPÍTULO 2. MARCO TEÓRICO .....	7
2.1 Complejidad del problema de Lennard Jones (LJ).....	8
2.1.1 Conceptos preliminares .....	8
2.1.2 Problema de optimización de LJ .....	10
2.1.3 Versión de decisión $LJ_d$ ( $LJ_{dd}$ ).....	10
2.1.4 El problema de $LJ_{dd}$ está en $NP$ .....	11
2.1.5 Problema de $LJ_{dd}$ es $NP$ -completo.....	12
2.2 Optimización matemática .....	16
2.2.1 Problema de optimización .....	16
2.2.2 Clasificación .....	17
2.2.3 Optimización no lineal .....	18
2.2.4 Optimización local.....	19
2.2.5 Optimización global .....	19
2.3 Fundamentos matemáticos .....	20
2.4 Algoritmos metaheurísticos.....	26
2.4.1 GRASP .....	27
2.4.2 Estructura de vecindades .....	28
2.4.3 Lista restringida de candidatos .....	30
2.5 Optimización numérica.....	31
2.6 Rejillas.....	34
2.6.1 Rejilla FCC.....	35
2.6.2 Rejilla icosaedra.....	36
2.7 Rebanada esférica.....	39
2.8 Coordenadas esféricas .....	41
2.9 Técnica de variables antitéticas.....	42
2.9.1 Técnicas de reducción de la varianza .....	43
2.9.2 Técnica de variables antitéticas.....	43
2.10 Trabajos relacionados .....	43



2.10.1 Análisis experimental de algoritmos .....	44
2.10.2 Solución del problema de LJ .....	45
CAPÍTULO 3. MÉTODO DE SOLUCIÓN.....	48
3.1 Enfoque de solución .....	48
3.2 Procedimiento de búsqueda miope, aleatorizado y adaptativo.....	51
3.2.1 Construcción miope aleatorizada.....	53
3.2.1.1 Método numérico de búsqueda local .....	56
3.2.1.2 Lista restringida de candidatos .....	57
3.2.2 Búsqueda local.....	57
3.2.2.1 Método numérico de búsqueda local .....	63
3.2.2.2 Cruzamiento .....	63
3.2.2.3 Rotación.....	65
3.2.2.4 Maquillaje.....	65
3.3 Método para reducir la varianza de los resultados en experimentos con algoritmos metaheurísticos .....	66
CAPÍTULO 4. RESULTADOS EXPERIMENTALES .....	73
4.1 Plan de pruebas.....	73
4.2 Plataforma experimental.....	75
4.3 Experimento 1: Evaluación de métodos de búsqueda local .....	75
4.4 Experimento 2: Discriminación entre los métodos de búsqueda local BFGS y Regiones de Confianza .....	79
4.5 Experimento 3: Evaluación de la eficiencia del algoritmo desarrollado.....	81
4.6 Experimento 4: Complejidad del algoritmo .....	84
4.7 Experimento 5: Impacto de la función de maquillaje.....	87
4.8 Experimento 6: Impacto del número de configuraciones optimizadas.....	92
4.9 Experimento 7: Evaluación del método de reducción de la varianza.....	96
CAPÍTULO 5. CONCLUSIONES Y TRABAJOS FUTUROS.....	100
REFERENCIAS BIBLIOGRÁFICAS .....	100

## LISTA DE FIGURAS

<b>Figura 1.1</b> Modelo de la energía potencial de un grupo de LJ con 3 átomos.....	3
<b>Figura 1.2</b> Modelo de la energía potencial de un grupo de LJ con 4 átomos.....	3
<b>Figura 2.1</b> Generación no determinista de una solución candidata.....	11
<b>Figura 2.2</b> Validación determinista de una solución candidata.....	12
<b>Figura 2.3</b> Grafo no dirigido.....	14
<b>Figura 2.4</b> Clasificación de la optimización.....	18
<b>Figura 2.5</b> Gráfica de la función $f(x_1, x_2) = 2x_1^2 + 2x_1x_2 + x_2^2 + 2x_1 - 3$ .....	23
<b>Figura 2.6</b> Gráfica de la función $f = x_1x_2$ .....	24
<b>Figura 2.7</b> Algoritmo GRASP.....	28
<b>Figura 2.8</b> Algoritmo miope aleatorio adaptativo.....	28
<b>Figura 2.9</b> Algoritmo de búsqueda local.....	28
<b>Figura 2.10</b> (a) Una instancia del TSP y un recorrido. (b) Otro recorrido el cual tiene 2- cambios del recorrido que se ve en (a).....	30
<b>Figura 2.11</b> Rejilla FCC.....	35
<b>Figura 2.12</b> Sitios de la rejilla FCC.....	35
<b>Figura 2.13</b> Sitios de la rejilla icosaedra $IC_2$ .....	38
<b>Figura 2.14</b> Sistema de coordenadas esféricas.....	42
<b>Figura 3.1</b> Método de optimización global.....	49
<b>Figura 3.2</b> Esquema de solución propuesto para el problema de LJ.....	50
<b>Figura 3.3</b> Algoritmo GRASP del enfoque de solución propuesto.....	52
<b>Figura 3.4</b> Algoritmo de la construcción miope aleatorizada del problema de LJ.....	56
<b>Figura 3.5</b> Algoritmo de la búsqueda local del problema de LJ.....	59
<b>Figura 3.6</b> Algoritmo del caso cortando con un plano.....	60
<b>Figura 3.7</b> Algoritmo del caso cortando con una esfera.....	61
<b>Figura 3.8</b> Algoritmo del caso cortando con una esfera centrada en el origen.....	62
<b>Figura 3.9</b> Algoritmo del caso de rotación de vecinos.....	63
<b>Figura 3.10</b> Secuencia de la función Cruzamiento.....	64
<b>Figura 3.11</b> Algoritmo de la función Agregar Átomo sin emplear reducción de la varianza.....	68
<b>Figura 3.12</b> Algoritmo de la función Agregar Átomo con reducción de la varianza.....	69
<b>Figura 4.1</b> Experimentos del plan de prueba.....	74
<b>Figura 4.2</b> Error de las funciones evaluadas.....	77
<b>Figura 4.3</b> Número de evaluaciones de las funciones.....	78
<b>Figura 4.4</b> Resultados de optimización de una solución inicial.....	80
<b>Figura 4.5</b> Evaluaciones de la función objetivo del experimento 3.....	82
<b>Figura 4.6</b> Número de iteraciones del experimento 3.....	84
<b>Figura 4.7</b> Complejidad del optimizador local Regiones de Confianza.....	85
<b>Figura 4.8</b> Complejidad del algoritmo propuesto.....	86
<b>Figura 4.9</b> Número de éxitos en cada grupo LJ con y sin maquillaje.....	90
<b>Figura 4.10</b> Porcentaje de error promedio en cada grupo LJ con y sin maquillaje.....	91
<b>Figura 4.11</b> Número de éxitos en cada grupo LJ optimizando todas y algunas configuraciones. .....	95

<b>Figura 4.12</b> Porcentaje de error promedio en cada grupo LJ optimizando todas y algunas configuraciones.....	95
-------------------------------------------------------------------------------------------------------------------	----

## LISTA DE TABLAS

<b>Tabla 2.1</b> Relación entre instancias TSP y LJ <sub>dd</sub> .....	13
<b>Tabla 2.2</b> Estructuras y complejidad instancias TSP.....	14
<b>Tabla 2.3</b> Estructuras y complejidad instancias LJ <sub>dd</sub> .....	14
<b>Tabla 2.4</b> Asignación de pesos a los lados. ....	15
<b>Tabla 2.5</b> Condiciones para los recorridos. ....	15
<b>Tabla 2.6</b> Coordenadas de los sitios de la rejilla $IC_2$ .....	37
<b>Tabla 2.7</b> Número de sitios en una rejilla icosaedral de 5 capas. ....	38
<b>Tabla 2.8</b> Distancia del punto $p(0.2763, -0.8506, 0.4472)$ a cada uno de los sitios de $IC_2$ . ....	40
<b>Tabla 2.9</b> Distancia del punto $p(1, 0, 1)$ a cada uno de los sitios de $IC_2$ .....	41
<b>Tabla 2.10</b> Enfoques de solución en los trabajos revisados.....	45
<b>Tabla 2.11</b> Trabajo relacionado con la solución del problema de LJ. ....	47
<b>Tabla 3.1</b> Salidas del experimento una vez aplicado el método de reducción. ....	69
<b>Tabla 3.2</b> Calculo de la varianza cuando se aplica el método de reducción.....	70
<b>Tabla 3.3</b> Salidas y cálculo de la varianza cuando no se aplica el método de reducción. ....	71
<b>Tabla 4.1</b> Funciones de prueba. ....	77
<b>Tabla 4.2</b> Evaluaciones de la función objetivo del experimento 3.....	82
<b>Tabla 4.3</b> Número de iteración del experimento 3. ....	83
<b>Tabla 4.4</b> Evaluaciones de la función objetivo con maquillaje. ....	88
<b>Tabla 4.5</b> Evaluaciones de la función objetivo sin maquillaje. ....	89
<b>Tabla 4.6</b> Número de éxitos, porcentaje de error promedio en cada grupo LJ con y sin maquillaje. ....	90
<b>Tabla 4.7</b> Evaluaciones de la función objetivo optimizando todas las configuraciones. ....	93
<b>Tabla 4.8</b> Evaluaciones de la función objetivo optimizando algunas configuraciones. ....	93
<b>Tabla 4.9</b> Número de éxitos, porcentaje de error promedio de todas y algunas configuraciones. ....	94
<b>Tabla 4.10</b> Reducción de la varianza en la Energía de LJ. ....	97
<b>Tabla 4.11</b> Reducción de la varianza en las evaluaciones de la función objetivo. ....	98
<b>Tabla 4.12</b> Reducción de la varianza en las iteraciones del GRASP.....	99

# Capítulo 1

## INTRODUCCIÓN

---

En este capítulo se presenta un panorama global de la tesis, se presentan los antecedentes del problema, se describe el problema de investigación y su justificación. Se muestran los objetivos que determinan el rumbo de esta investigación y la hipótesis. Como último punto se explica brevemente el contenido de cada capítulo de la tesis.

### 1.1 Antecedentes

Los grupos de átomos o moléculas son agregados de unos cuantos a miles de ellos, se han estudiado intensamente durante los últimos veinte años por su importancia en la física, química, bioquímica y ciencia de materiales [Zhao 2004]. Las características físicas y

químicas de los materiales varían de acuerdo a una serie de propiedades de los grupos de átomos que los integran. De hecho si se agrega un átomo más, puede resultar una estructura enteramente diferente. Desde un punto de vista tecnológico, las aplicaciones de los grupos sirven como bloques de construcción en nanociencia y nanotecnología; algunas aplicaciones específicas son el desarrollo de películas, sólidos y circuitos nanoelectrónicos.

Las propiedades físicas y químicas de los materiales dependen críticamente de la geometría de los grupos. Para explicar dichas propiedades se debe conocer la estructura geométrica de la configuración de los átomos con menor energía potencial. Uno de los problemas más simples de plantear, pero difíciles de resolver en Química computacional es la búsqueda de configuraciones moleculares óptimas bajo un modelo de energía potencial determinado. Es decir determinar las posiciones de  $n$  átomos en el espacio tridimensional de tal manera que su energía potencial sea mínima.

En general, la energía potencial  $V$  de un sistema de  $n$  átomos puede expresarse en función de las coordenadas de los puntos  $p_i$ , donde se encuentran ubicados los átomos individuales.

$$V = \sum_{i=1}^n v_1(r_i) + \sum_{i=1}^n \sum_{j>i}^n v_2(r_i, r_j) + \sum_{i=1}^n \sum_{j>i}^n \sum_{k>j>i}^n v_3(r_i, r_j, r_k) + \dots \quad (1.1)$$

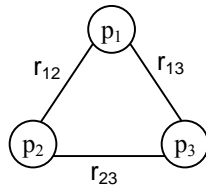
El primer término  $v_1$  representa las interacciones de los átomos del sistema con un cuerpo externo fijo,  $v_2$  las interacciones entre pares de átomos del sistema,  $v_3$  interacciones de tres átomos y así sucesivamente. En particular, el segundo término es muy relevante ya que describe por sí solo sistemas físicos tales como los gases nobles,  $v_2$  sólo depende de la distancia interatómica  $r_{ij} = |r_i - r_j|$ . Un modelo adecuado para  $v_2$ , debe corresponder con el siguiente comportamiento físico de los átomos: se atraen a grandes distancias y se repelen a cortas distancias.

En esta tesis se estudia el modelo del potencial de Lennard Jones (LJ). Este modelo establece que la energía potencial de una configuración determinada de un grupo de  $n$  átomos está dada por:

$$V = \sum_{i=1}^{n-1} \sum_{j=i+1}^n (r_{ij}^{-12} - 2r_{ij}^{-6}) \quad (1.2)$$

Donde  $r_{ij}$  es la distancia Euclidiana entre el punto  $p_i$  y  $p_j$ . A los grupos de átomos cuyas interacciones se ajustan mediante este modelo se les denomina grupos de LJ.

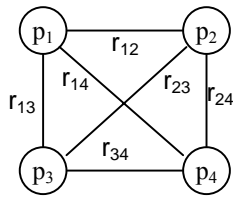
La Figura 1.1 muestra el desarrollo del modelo para el caso de un grupo de LJ con 3 átomos.



$$V = (r_{12}^{-12} - 2r_{12}^{-6}) + (r_{13}^{-12} - 2r_{13}^{-6}) + (r_{23}^{-12} - 2r_{23}^{-6})$$

**Figura 1.1** Modelo de la energía potencial de un grupo de LJ con 3 átomos.

Para un grupo de 4 átomos, la Figura 1.2 muestra el modelo correspondiente.



$$V = (r_{12}^{-12} - 2r_{12}^{-6}) + (r_{13}^{-12} - 2r_{13}^{-6}) + (r_{14}^{-12} - 2r_{14}^{-6}) + (r_{23}^{-12} - 2r_{23}^{-6}) + (r_{24}^{-12} - 2r_{24}^{-6}) + (r_{34}^{-12} - 2r_{34}^{-6})$$

**Figura 1.2** Modelo de la energía potencial de un grupo de LJ con 4 átomos.

Se observa que el número de términos del modelo es igual al número de parejas de átomos que se pueden formar. Por lo tanto el número de términos es igual al número de combinaciones de tamaño dos que se generan a partir de los  $n$  átomos:  $C(n, 2)$ . Por ejemplo, para un grupo de 3

átomos, el número de términos del modelo es  $C(3, 2) = 3$  y para un grupo de 4 átomos es  $C(4, 2) = 6$ . Por otra parte, como las variables del modelo son las coordenadas de todos los puntos donde se ubican los átomos  $(x_1, x_2, x_3, \dots, x_n, y_1, y_2, y_3, \dots, y_n, z_1, z_2, z_3, \dots, z_n)$ , se observa que el número de variables del modelo es  $3n$ .

## 1.2 Descripción del problema de investigación

Dado un grupo de Lennard-Jones de tamaño  $n$ , determinar las posiciones  $p_1, p_2, \dots, p_n \in R^3$ , de los átomos de tal manera que se minimice la energía potencial  $V = \sum_{i=1}^{n-1} \sum_{j=i+1}^n (r_{ij}^{-12} - 2r_{ij}^{-6})$  que se produce como resultado de las interacciones atómicas. Donde  $r_{ij}$  es la distancia Euclidiana  $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}$  entre los puntos  $p_i$  y  $p_j$ .

Las fuentes principales de dificultad de este problema son las siguientes:

- el número de términos de la función objetivo crece con el número de combinaciones posibles para  $n$ ,
- la función objetivo no es convexa,
- el número de variables crece con  $n$ ; y
- el número de mínimos locales crece exponencialmente [Wales 1999].

## 1.3 Justificación

La importancia práctica de descubrir configuraciones de energía mínima de grupos de átomos, constituye un elemento fundamental en la investigación de conformación molecular de los

materiales. A pesar de la simplicidad del modelo de LJ, las investigaciones en química y biología lo consideran un modelo razonablemente bueno de algunos grupos de gases raros. Otra característica relevante del modelo es que es un componente importante de otros modelos usados en la configuración de moléculas complejas y desdoblamiento de proteínas [Neumaier1997].

Desde otro punto de vista, el problema de la configuración atómica o molecular bajo el potencial de LJ, es un excelente sistema de prueba para la evaluación de nuevos métodos de optimización global [Locatelli2000]. La tarea de minimizar la energía de un grupo es notoriamente difícil porque el número de mínimos locales tiende a crecer exponencialmente cuando el número de átomos aumenta [Wales1999].

## **1.4 Objetivos**

### **1.4.1 Objetivo general**

Desarrollar un nuevo método de optimización global para la solución del problema de Lennard-Jones, con base en el metaheurístico GRASP y el método de búsqueda local Regiones de Confianza.

### **1.4.2 Objetivos específicos**

- Seleccionar el método de optimización local
- Seleccionar el método de reducción de la varianza
- Diseñar e implementar el método de optimización global
- Realizar el análisis experimental del desempeño de la solución propuesta
- Comparar los resultados obtenidos con los realizados en otras investigaciones



## **1.5 Hipótesis**

Es posible desarrollar un nuevo método de optimización global para la solución del problema de Lennard-Jones con base en el metaheurístico GRASP y el método Regiones de Confianza, el cual es capaz de obtener los mejores resultados reportados.

## **1.6 Organización de la tesis**

En el Capítulo 2 se presenta el marco teórico en el que se fundamenta el desarrollo de los diferentes elementos de este trabajo. Se muestra la complejidad del problema lo cual justifica el enfoque de solución, se describen los fundamentos teóricos de optimización matemática, fundamentos matemáticos, algoritmos metaheurísticos, optimización numérica, rejillas, rebanada esférica y la técnica de variables antitéticas. Además se describen y analizan los trabajos más relevantes relacionados con el análisis experimental de algoritmos y con la solución del problema de LJ.

En el Capítulo 3 se presenta el enfoque general que se propone para la solución del problema de LJ, se describen los módulos desarrollados en la solución del problema y se presenta la metodología que permite controlar la variación de los resultados de experimentos con algoritmos metaheurísticos.

En el Capítulo 4 se describen los experimentos realizados y el análisis de los resultados obtenidos para determinar la calidad y eficiencia del método propuesto.

En el Capítulo 5 se describen las aportaciones más importantes de esta investigación y se muestran las líneas de investigación que se identificaron en el proceso.

# Capítulo 2

## MARCO TEÓRICO

---

En este capítulo se describe el fundamento teórico en que se basa este trabajo. Este marco incluye la demostración de que el problema de Lennard-Jones (LJ) es *NP-duro*, información teórica de optimización matemática, fundamentos matemáticos, información acerca del metaheurístico que se empleó y conceptos relacionados con el método de solución. Se describen un método de búsqueda local y un método para la reducción de la varianza. Se incluye también una sección que contiene una descripción de los trabajos relacionados con el análisis experimental de algoritmos y con el problema de LJ.

## 2.1 Complejidad del problema de Lennard Jones (LJ)

En esta sección se demuestra que el problema de LJ es *NP-duro*. Para realizar la demostración se introducen algunas definiciones y notaciones de la teoría de la *NP*-completez que faciliten la descripción de los argumentos utilizados.

### 2.1.1 Conceptos preliminares

**Problema de decisión y algoritmo.** Un problema de decisión  $\pi = (D, Y)$  es una pareja formada por un conjunto de instancias  $D$ , las cuales se obtienen a partir de una instancia genérica que se especifica en términos de varios componentes: conjuntos, funciones, números, etc., y un subconjunto de instancias-sí  $Y \subseteq D$ . Una instancia  $i \in Y$ , si y sólo si, la respuesta a la cuestión del problema es sí para esa instancia. Un problema de decisión se asocia con un lenguaje formal usando algún mecanismo de codificación y un algoritmo con una máquina de Turing. Un algoritmo se dice que resuelve un problema de decisión si y sólo si, el lenguaje aceptado por la máquina de Turing es el subconjunto de todas las cadenas asociadas con las instancias-sí del problema. En las siguientes definiciones, si no se señala otra cosa, la palabra problema se usa como sinónimo de problema de decisión.

**Clase  $P$ .** Es el conjunto de todos los problemas de decisión que pueden ser resueltos en tiempo polinomial por un algoritmo determinista. A los problemas que pertenecen a esta clase se les denomina tratables.

**Problemas intratables.** Son todos los problemas de decisión para los que no existe un algoritmo determinista de tiempo polinomial que los resuelva, es decir, son todos los problemas que están en  $P^c$ .

**Clase  $NP$ .** La clase *NP* es el conjunto de todos los problemas de decisión que se pueden verificar en tiempo polinomial con un algoritmo no determinista.

**Relación entre  $P$  y  $NP$ .** Como toda máquina determinista es un caso particular de una máquina no determinista, se tiene entonces que  $P \subseteq NP$ .

**Transformación polinomial.** Se dice que un problema de decisión  $\pi_1 = (D_1, Y_1)$  se puede transformar polinomialmente en el problema de decisión  $\pi_2 = (D_2, Y_2)$ , si y sólo si, existe una función  $f : D_1 \rightarrow D_2$  que satisface las siguientes dos condiciones:

1.  $f$  es computable con un algoritmo determinista de tiempo polinomial.
2. Para toda instancia  $i \in D_1$ ,  $i \in Y_1$ , si y sólo si,  $f(i) \in Y_2$ .

En tal caso se dice que  $\pi_1 \alpha_p \pi_2$ .

**Problemas  $NP$ -completos y  $NP$ -duros.** Un problema  $\pi$  es  $NP$ -completo si y sólo si,  $\pi \in NP$  y  $\forall \phi \in NP, \phi \leq_p \pi$ . Al conjunto de todos los problemas  $NP$ -completos se les denota por  $NPC$ .

Un problema de optimización se dice que es  $NP$ -duro, si y sólo si, su versión de decisión es  $NP$ -completo. Para probar que  $\pi \in NPC$ , es suficiente y necesario probar que  $\pi \in NP$  y que existe  $\pi^* \in NPC$  tal que  $\pi^* \alpha_p \pi$ . Una propiedad importante de los problemas  $NP$ -completos es que si  $\pi \in NPC$ , entonces  $\pi \in P$ , si y sólo si,  $P = NP$ .

La demostración que se va a realizar consiste en probar primero que un subproblema de la versión de decisión del problema de LJ está en la clase  $NP$ . Una vez establecido lo anterior, se probará que es posible reducir polinomialmente el problema del agente viajero [Garey 1979, Papadimitriou 1998] a dicho subproblema. Esto es suficiente para probar que el subproblema es  $NP$ -completo [Garey 1979, Papadimitriou 1998]. Luego por restricción, la versión de decisión del problema de optimización es  $NP$ -completo y por tanto el problema es  $NP$ -duro [Garey 1979].

## 2.1.2 Problema de optimización de LJ

Dado un grupo de LJ de tamaño  $n$ , determinar las posiciones,  $p_1, p_2, \dots, p_n \in R^3$ , de los átomos de tal manera que se minimice la energía potencial  $V = \sum_{i=1}^{n-1} \sum_{j=i+1}^n (r_{ij}^{-12} - 2r_{ij}^{-6})$  que se produce como resultado de las interacciones atómicas entre pares de átomos. Donde  $r_{ij}$  es la distancia Euclidiana  $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}$  entre los puntos  $p_i$  y  $p_j$ .

### Problema de LJ discreto (LJd)

**Instancia:** Dado un conjunto finito de sitios en el espacio tridimensional  $s_1, s_2, \dots, s_m \in R^3$ , una distancia  $d(i, j) \in Z^+$  entre cada par de sitios, un entero  $n \in Z^+$  correspondiente al número de átomos y la función de energía potencial de LJ  $V = \sum_{i=1}^{n-1} \sum_{j=i+1}^n (r_{ij}^{-12} - 2r_{ij}^{-6})$  para las interacciones entre pares de átomos del sistema.

### Versión de optimización de LJd

Determinar los sitios adecuados donde posicionar los  $n$  átomos de tal forma que su energía potencial sea mínima.

## 2.1.3 Versión de decisión LJd (LJdd)

El problema de decisión LJ<sub>dd</sub> asociado al problema de LJ<sub>d</sub>, se formula de la manera siguiente:

**Instancia:** Dado un conjunto finito de sitios en el espacio tridimensional  $s_1, s_2, \dots, s_m \in R^3$ , una distancia  $d(i, j) \in Z^+$  entre cada par de sitios, un entero  $n \in Z^+$  correspondiente al número de

átomos y la función de energía potencial de LJ  $V = \sum_{i=1}^{n-1} \sum_{j=i+1}^n (r_{ij}^{-12} - 2r_{ij}^{-6})$  para las interacciones entre pares de átomos del sistema.

**Pregunta:** ¿Existe una forma de asignar los  $n$  átomos a  $m$  sitios de tal manera que se minimice su energía potencial?

### 2.1.4 El problema de LJ<sub>dd</sub> está en NP

Para probar que LJ<sub>dd</sub> está en NP, se construye una máquina no determinista que genera una solución candidata y una máquina determinista para validar su factibilidad, y establecer que las máquinas generan la solución candidata y la validan en tiempo polinomial respectivamente [Garey 1979, Papadimitriou 1998].

Una solución candidata de LJ<sub>dd</sub>, es cualquier asignación aleatoria de los  $n$  átomos a los  $m$  sitios. La Figura 2.1 muestra el algoritmo no determinista para la generación de la solución candidata.

1	//Sea $m$ el número de sitios donde posicionar los $n$ átomos // $m \geq n$ // Sea $X$ el arreglo de los sitios
2	<b>Para</b> ( $i=1, \dots, m$ ) <b>Hacer</b>
3	<b>Si</b> random[0,1] < 0.5
4	$X[i]=1$
5	<b>Sino</b>
6	$X[i]=0$
7	<b>Fin Para</b>

Figura 2.1 Generación no determinista de una solución candidata.

La asignación aleatoria de un átomo a cada uno de los sitios se puede realizar en a lo más  $m$  operaciones, por lo tanto la asignación candidata se puede generar en a lo más  $m$  pasos.

Una vez generada la asignación, la validación de que en la solución candidata se encuentren  $n$  átomos se puede realizar en a los más  $m$  pasos. Por lo tanto, la verificación para todos los sitios se puede realizar en a lo más  $m$  pasos. La Figura 2.2 muestra el algoritmo determinista para la validación de la solución candidata.

1	//Sea $m$ el número de sitios donde posicionar los $n$ átomos // $m \geq n$ // Sea $X$ el arreglo de los sitios // Sea solución una bandera (1 sol. factible; 0 sol. infactible)
2	contador $\leftarrow 0$
3	<b>Para</b> ( $i=1, \dots, m$ ) <b>Hacer</b>
4	<b>Si</b> $X[i]=1$
5	contador $\leftarrow$ contador + 1
6	<b>Fin Si</b>
7	<b>Si</b> contador = $n$
8	solución $\leftarrow 1$
9	<b>Fin Si</b>
10	<b>Fin Para</b>

**Figura 2.2** Validación determinista de una solución candidata.

Como la complejidad del algoritmo no determinista que genera la solución candidata es  $O(m)$  y la complejidad del algoritmo determinista que valida la solución es  $O(m)$ , se concluye que  $LJ_{dd}$  está en  $NP$  [Garey 1979, Papadimitriou 1998].

### 2.1.5 Problema de $LJ_{dd}$ es $NP$ -completo

Una vez probado que la versión de decisión del problema de  $LJ$  está en  $NP$ , se probará que es posible reducir polinomialmente el problema del agente viajero TSP (Traveling Salesman Problem, por sus siglas en inglés) al problema  $LJ_{dd}$  y así demostrar que  $LJ_{dd}$  pertenece a la clase  $NP$ -completo [Garey 1979, Papadimitriou 1998]. Para probar que  $TSP \alpha_p LJ_{dd}$ , se debe construir una correspondencia entre las instancias de ambos problemas, de tal modo que esta correspondencia y su inversa se puedan realizar en forma determinista en tiempo polinomial [Garey 1979, Papadimitriou 1998].

### Problema de decisión del TSP

El problema de decisión asociado al problema de TSP, se formula de la manera siguiente:

**Instancia:** Dado un conjunto finito de ciudades  $C = \{c_1, c_2, \dots, c_m\}$  y una distancia  $d(c_i, c_j) \in \mathbb{Z}^+$  entre cada par de ciudades  $c_i, c_j \in C$ .

**Pregunta:** ¿Existe una permutación  $[c_{\pi(1)}, c_{\pi(2)}, \dots, c_{\pi(m)}]$  de  $C$ , que minimice la longitud del recorrido  $\left[ \sum_{i=1}^{m-1} d(c_{\pi(i)}, c_{\pi(i+1)}) \right] + d(c_{\pi(m)}, c_{\pi(1)})$ ?

### Correspondencia entre instancias TSP y LJdd

Las instancias de TSP y LJ<sub>dd</sub> se pueden representar utilizando un grafo  $G = (V, E)$  donde,  $V$  es un conjunto finito de vértices,  $E$  es el conjunto de aristas y  $w(e)$  es el peso de los lados  $w(e) \forall e \in E$ . El grafo  $G$  es completo es decir existe un lado  $e = \langle u, v \rangle \in E \forall u, v \in V$ .

La Tabla 2.1 muestra la relación entre las instancias TSP y LJ<sub>dd</sub>.

**Tabla 2.1** Relación entre instancias TSP y LJ<sub>dd</sub>.

<b>TSP</b>	<b>Grafo</b>	<b>LJ<sub>dd</sub></b>
Ciudades	$V$	Sitios
Distancia entre ciudades	$w(e) \forall e \in E$	Interacción entre sitios
Ciudades seleccionadas	$\leftrightarrow$	Configuración atómica
Recorrido mínimo	$\leftrightarrow$	Configuración de energía mínima

En las Tablas 2.2 y 2.3 se representan las estructuras de las instancias del TSP y LJ<sub>dd</sub> así como la complejidad del algoritmo determinista para su transformación.



**Tabla 2.2** Estructuras y complejidad instancias TSP.

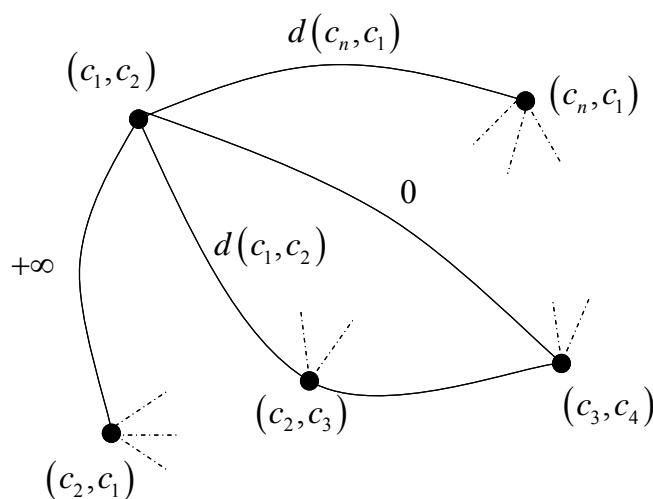
TSP	Estructura	Complejidad
Ciudades $C = \{c_1, c_2, \dots, c_m\}$	$C[1 \dots m]$	$O(m)$
Distancia entre ciudades $d(c_i, c_j) \in \mathbb{Z}^+$	$d[1 \dots m][1 \dots m]$	$O(m^2)$

**Tabla 2.3** Estructuras y complejidad instancias LJ<sub>dd</sub>.

LJ <sub>dd</sub>	Estructura	Complejidad
Sitios $S = \{s_1, s_2, \dots, s_m\}$	$S[1 \dots m]$	$O(m)$
Distancia entre sitios $d(i, j) \in \mathbb{Z}^+$	$d[1 \dots m][1 \dots m]$	$O(m^2)$

### Resolviendo instancias-si TSP y LJ<sub>d</sub>

Para resolver instancias-si de **TSP** y **LJ<sub>d</sub>** se utiliza el grafo no dirigido de la Figura 2.3, donde  $G = (V, E)$  con  $V = n(n-1)$  vértices, donde  $n$  es equivalente al número de átomos.



**Figura 2.3** Grafo no dirigido.

Cada vértice es etiquetado con  $(c_i, c_j)$  donde  $i, j = 1, \dots, n$  y  $i \neq j$ , esto indica que el vértice  $c_j$  es visitado inmediatamente después del vértice  $c_i$ . Los lados son pares sin ordenar de la forma  $e = \langle (c_i, c_j)(c_k, c_l) \rangle$  con los pesos asignados como se muestra en la Tabla 2.4.

**Tabla 2.4** Asignación de pesos a los lados.

Peso	Valor	Condición
$w(e)$	0	$i \neq k$ y $j \neq l$
	$+\infty$	$i = k$ y $j \neq l$
		$l = l$ y $i \neq k$
		$l = i$ y $k = j$
	$d(c_k, c_l)$	$l = i$ y $k \neq j$
	$d(c_i, c_l)$	$k = j$ y $l \neq i$

La Tabla 2.5 describe las condiciones que definen un recorrido válido e inválido.

**Tabla 2.5** Condiciones para los recorridos.

Condición	Definición
$i \neq k$ y $j \neq l$	Define un recorrido inconexo donde el recorrido visita $c_i$ seguido por $c_j$ y $c_k$ seguido por $c_l$ . Sin embargo no muestra que otras ciudades fueron visitadas entre $c_j$ y $c_k$ , por lo tanto no es posible describir la permutación y calcular la longitud del recorrido.
$i = k$ y $j \neq l$	Define un recorrido ilegal que sale de una ciudad para visitar dos ciudades diferentes.
$l = l$ y $i \neq k$	Define un recorrido ilegal donde dos ciudades diferentes visitan la misma ciudad siguiente.
$l = i$ y $k = j$	Define un recorrido cíclico ilegal entre solamente dos ciudades
$l = i$ y $k \neq j$	Son recorridos válidos.
$k = j$ y $l \neq i$	

Una instancia-si de TSP es equivalente a seleccionar  $V' \subset V$  con  $|V'| = n$ , donde

$$\frac{1}{2} \sum_{\substack{u, v \in V' \\ e = \langle u, v \rangle}} w(e) \text{ es mínimo.}$$

Una instancia-si de LJ<sub>dd</sub> es equivalente a seleccionar  $V' \subset V$  con  $|V'| = n$ , donde

$$\frac{1}{2} \sum_{\substack{u, v \in V' \\ e = \langle u, v \rangle}} w(e) \text{ es mínimo.}$$

Seleccionando  $n$  vértices  $\left\{ \left( c_{\pi(i)}, c_{\pi(i+1)} \right); i = 1, \dots, n, c_{\pi(1)} \right\}$ , se demuestra que la suma de los pesos mínimos es equivalente a encontrar la longitud del recorrido mínimo  $\left\langle c_{\pi(1)}, c_{\pi(2)}, \dots, c_{\pi(n)} \right\rangle$ . Cada uno de los vértices del recorrido mínimo es asignado por la correspondencia a un átomo, por lo que es equivalente a encontrar la energía mínima para un grupo de  $n$  átomos. Por lo tanto se concluye que el problema LJ<sub>dd</sub> es *NP-completo* [Garey 1979, Papadimitriou 1998]. Una vez probado que el problema LJ<sub>dd</sub> es *NP-completo*, se concluye por restricción que el problema LJ<sub>dd</sub> es *NP-duro* [Garey 1979].

## 2.2 Optimización matemática

En este apartado se describirán algunas definiciones relacionadas con la optimización matemática.

### 2.2.1 Problema de optimización

Un problema de optimización matemática o problema de optimización tiene la siguiente forma:

$$\begin{aligned} & \min f_0(x) \\ \text{Sujeta a } & f_i(x) \leq b_i, \quad i = 1, \dots, m \end{aligned} \tag{2.1}$$

Donde  $x = (x_1, \dots, x_n)$  son las variables de optimización del problema, la función  $f_0(x) = R^n \rightarrow R$  es la función objetivo, las funciones  $f_i(x) = R^n \rightarrow R$  son funciones de restricción o desigualdades y las constantes  $b_1, \dots, b_m$  son límites para las restricciones.

El vector  $x^*$  es llamado óptimo o solución del problema, si la función objetivo tiene el valor más pequeño entre todos los puntos que satisfacen las restricciones. Formalmente esto significa que para todo  $z$  tal que  $f_1(z) \leq b_1, \dots, f_m(z) \leq b_m$  se tiene  $f_0(z) \geq f_0(x^*)$ .

Un problema de optimización es entonces: encontrar los valores de las variables que minimizan o maximizan la función objetivo mientras satisface las restricciones.

### 2.2.2 Clasificación

La Ecuación 2.1 permite caracterizar diferentes tipos de problemas de optimización dependiendo de las características de la función objetivo y de las funciones restricción. Los problemas de optimización se dice que son: discretos si algunas o todas las variables tienen valores enteros, continuos si las variables toman valores reales; lineales si todas las funciones son lineales, no lineales en caso contrario; locales cuando es suficiente obtener un óptimo local, globales en caso de que el óptimo global sea requerido; restringidos si tienen funciones de restricción y no restringidos si no tienen restricciones.

En la Figura 2.4 se muestra una clasificación de la optimización, para más detalles revisar [Neos 1996].

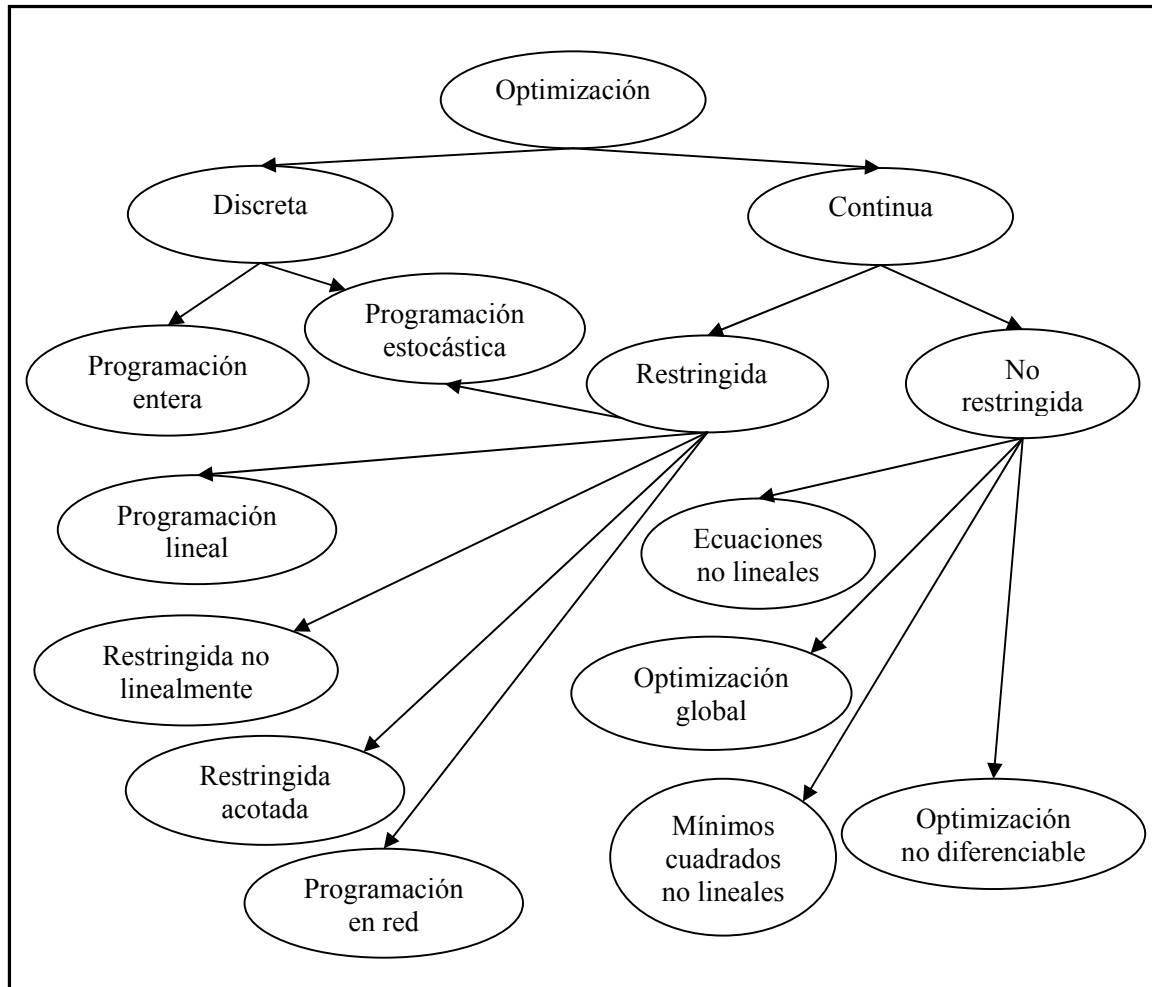


Figura 2.4 Clasificación de la optimización.

### 2.2.3 Optimización no lineal

La optimización no lineal o programación no lineal es un término usado para describir problemas de optimización donde la función objetivo o las funciones de restricción no son lineales y no se conocen que sean convexas. Hasta ahora no se han propuesto métodos efectivos para resolver el problema general de la programación no lineal [Boyd 2006]. Así, problemas con pocas variables pueden ser extremadamente complejos, mientras que problemas no lineales de muchas variables se consideran intratables. El problema de LJ es no

lineal y el número de variables crece con el tamaño del grupo que se esté considerando, por lo tanto es un problema que se considera intratable.

#### **2.2.4 Optimización local**

La optimización local, no se propone encontrar el óptimo global. En lugar de esto busca un punto que sea localmente óptimo, esto es, que sea óptimo con respecto a sus puntos vecinos. Los métodos de optimización local son rápidos y trabajan adecuadamente con problemas de gran escala, por lo tanto son comúnmente utilizados. Un requisito para su aplicación es que la función objetivo y las funciones de restricción sean derivables. Estos métodos son ampliamente usados en aplicaciones donde se quiere encontrar un buen resultado, que no necesariamente tiene que ser el mejor. Básicamente este tipo de métodos transitan desde una solución inicial dada hacia el óptimo local más cercano. Como consecuencia no tienen capacidad para escapar de los óptimos locales, por lo tanto la calidad de las soluciones que producen depende de la solución inicial que se utilice. Ésta es la limitación más importante de los métodos de búsqueda local. Además, la calidad de la solución depende también del valor de sus parámetros de control, teniendo que ajustarlos para un problema en particular o para un conjunto de problemas. Esto da como resultado que la optimización local sea más un arte que una ciencia [Boyd 2006].

En particular, los métodos de búsqueda local son insuficientes para la solución de problemas que presenten un gran número de mínimos locales. Como es el caso del problema de LJ.

#### **2.2.5 Optimización global**

Se podría definir una búsqueda global como una versión robusta de la optimización local, en la que en lugar de buscar un punto factible localmente óptimo se requiere el mejor punto en la región factible [Boyd 2006].

La optimización global es uno de los objetivos más importantes en el desarrollo de aplicaciones y al mismo tiempo es uno de los problemas más difíciles para la optimización y la computación en general. Dada su importancia práctica en diferentes áreas de aplicación y la escasez de teoría matemática (comparada con la teoría de optimización local), muchos métodos heurísticos han sido propuestos por expertos en diferentes áreas [Fraire 2005].

En muchas aplicaciones prácticas, encontrar el mejor punto globalmente no es esencial, por lo general sólo se requiere una mejora de un punto factible para obtener resultados satisfactorios. Sin embargo, existen algunos casos en los cuales es necesario encontrar un óptimo global:

- Problemas combinatorios (ejemplo, el diseño del brazo de un robot), donde los métodos de búsqueda local no retornan información útil dado que ellos generalmente se quedan atrapados en los mínimos locales de su función objetivo.
- Pruebas formales asistidas por computadora (ejemplo, la prueba de la conjetura de Kepler), donde las desigualdades deben ser establecidas con garantía matemática.
- Muchos problemas en química, donde el mínimo global es el único que corresponde a la realidad.

## 2.3 Fundamentos matemáticos

En esta sección se presentan las definiciones de: gradiente, matriz Hessiana y polinomio característico [Apostol 1969].

**Definición 2.1** Sea  $z = f(X)$  una función de  $X$  tal que  $\frac{\partial f(X)}{\partial x_k}$  existe para  $k = 1, 2, \dots, n$ . El

gradiente de  $f$ , denotado por  $\nabla f(X)$ , es el vector:

$$\nabla f(X) = \left( \frac{\partial f(X)}{\partial x_1}, \frac{\partial f(X)}{\partial x_2}, \dots, \frac{\partial f(X)}{\partial x_n} \right)$$

**Definición 2.2** La matriz Hessiana de  $f$  es una matriz cuadrada de orden  $n \times n$ , la cual se forma por las  $n^2$  derivadas parciales segundas de  $f$ .

$$H_f = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}$$

**Definición 2.3** Si  $A$  es una matriz  $n \times n$ , el determinante  $f(\lambda) = \det(\lambda I - A)$  se denomina polinomio característico de  $A$ .

**Proposición 2.1** Dada una función  $f$  diferenciable en un subconjunto abierto  $D$  de  $R^n$ , se dice que  $x^* \in D$  es un punto crítico de  $f$  cuando se verifica que  $\nabla f(x^*) = \bar{0}$ .

**Proposición 2.2** Sea  $D$  un subconjunto abierto de  $R^n$  y  $f$  una función definida de  $D$  en  $R$  que tiene derivadas parciales primeras y segundas continuas en  $D$ . Sea  $x^* \in D$  un punto crítico de  $f$ . Si por  $Hf(x^*)$  se denota la matriz Hessiana de la función  $f$  en el punto  $x^*$ , se verifica:

- Si  $Hf(x^*)$  es definida positiva, entonces  $x^*$  es un mínimo local estricto de  $f$ .
- Si  $Hf(x^*)$  es definida negativa, entonces  $x^*$  es un máximo local estricto de  $f$ .
- Si  $Hf(x^*)$  es indefinida, entonces  $x^*$  es un punto silla de  $f$ .



En las siguientes funciones se obtendrán los puntos críticos y con esto se ejemplificará el uso de las definiciones y proposiciones anteriores.

Obtener los puntos críticos de las siguientes funciones:

a)  $f(x_1, x_2) = 2x_1^2 + 2x_1x_2 + x_2^2 + 2x_1 - 3$

b)  $f(x_1, x_2) = x_1x_2$

c)  $V = \sum_{i=1}^{n-1} \sum_{j=i+1}^n (r_{ij}^{-12} - 2r_{ij}^{-6})$

a) Dada la función  $f(x_1, x_2) = 2x_1^2 + 2x_1x_2 + x_2^2 + 2x_1 - 3$  se obtiene su primera derivada parcial con respecto a  $x_1$

$$\frac{\partial f}{\partial x_1} = 4x_1 + 2x_2 + 2$$

Con respecto a  $x_2$

$$\frac{\partial f}{\partial x_2} = 2x_1 + 2x_2$$

Igualando a cero las primeras derivadas parciales ( $\nabla f(x^*) = \bar{0}$ ) se pueden obtener los puntos críticos, resultando en un único punto crítico  $x^* = (-1, 1)$ .

Su segundas derivadas parciales con respecto a  $x_1$

$$\frac{\partial^2 f}{\partial x_1^2} = 4$$

$$\frac{\partial^2 f}{\partial x_1 \partial x_2} = 2$$

y con respecto a  $x_2$

$$\frac{\partial^2 f}{\partial x_2^2} = 2$$

$$\frac{\partial^2 f}{\partial x_2 \partial x_1} = 2$$

Por definición 2.2, el Hessiano de  $f$  viene dado por:

$$Hf(x_1, x_2) = \begin{pmatrix} 4 & 2 \\ 2 & 2 \end{pmatrix}$$

Al obtener el polinomio característico

$$p(\lambda) = \begin{vmatrix} \lambda - 4 & 2 \\ 2 & \lambda - 2 \end{vmatrix}$$

$$p(\lambda) = (\lambda - 4)(\lambda - 2) - 4$$

$$p(\lambda) = \lambda^2 - 6\lambda + 4$$

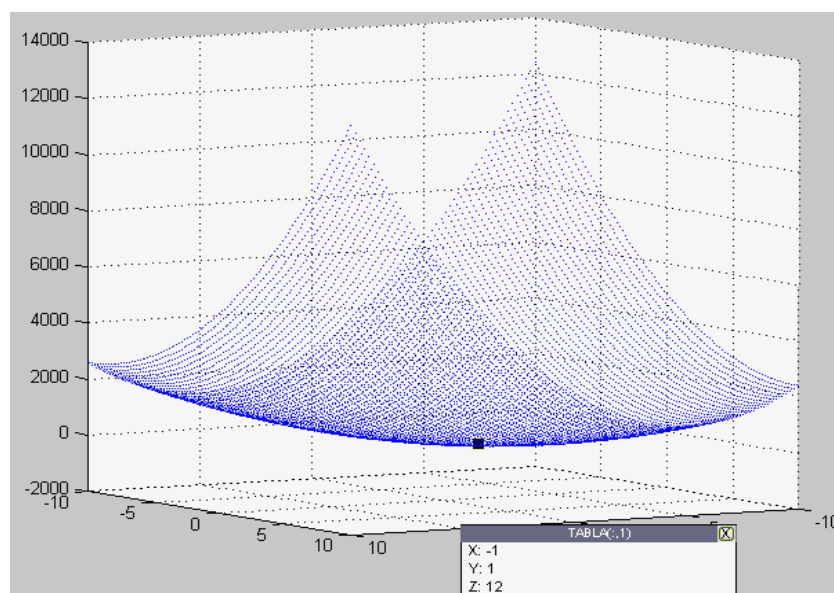
Igualando  $p(\lambda) = 0$ , para obtener los valores característicos

$$\lambda^2 - 6\lambda + 4 = 0$$

$$\lambda = \frac{6 \pm \sqrt{36 - 16}}{2}$$

$$\therefore \lambda_1 > 0, \lambda_2 > 0$$

Como  $\lambda_1 > 0, \lambda_2 > 0$ , la matriz Hessiana se define como una matriz positiva para todo  $(x_1, x_2) \in \mathbb{R}^2$ . Por lo tanto, el punto  $x^*$  es un mínimo local estricto de  $f$ , ver Figura 2.5.



**Figura 2.5** Gráfica de la función  $f(x_1, x_2) = 2x_1^2 + 2x_1x_2 + x_2^2 + 2x_1 - 3$ .

b) Dada la función  $f = x_1x_2$ . Se obtienen sus primeras derivadas parciales y aplicando la condición de que  $\nabla f(x^*) = \bar{0}$ , se determina el punto crítico  $x^* = (0, 0)$ .

La matriz Hessiana en dicho punto es la siguiente:

$$Hf(x_1, x_2) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

Puesto que la matriz es indefinida, el punto  $(0, 0)$  es un punto silla, ver Figura 2.6.

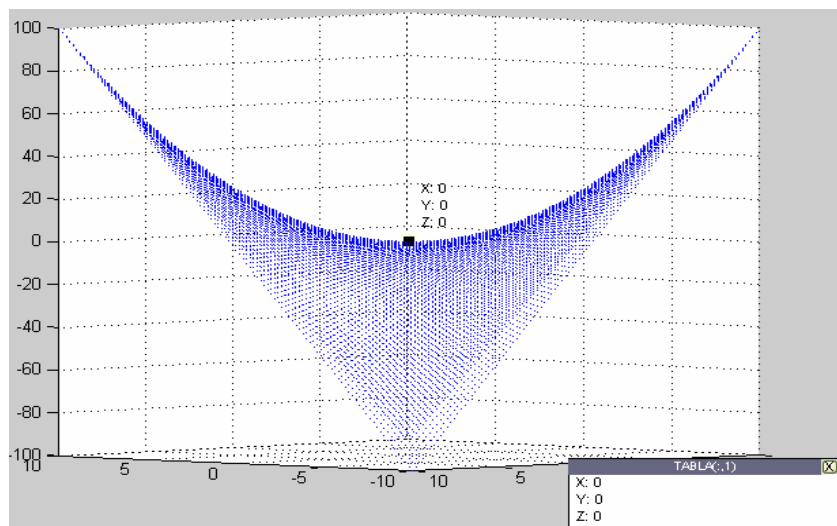


Figura 2.6 Gráfica de la función  $f = x_1x_2$ .

c) Dada la función de energía de LJ,  $V = \sum_{i=1}^{n-1} \sum_{j=i+1}^n (r_{ij}^{-12} - 2r_{ij}^{-6})$ , determinar su vector gradiente y su matriz Hessiana.

$$V = \sum_{i=1}^{n-1} \sum_{j=i+1}^n \left( \frac{1}{r_{ij}^{12}} - \frac{2}{r_{ij}^6} \right)$$

$$V = \sum_{i=1}^{n-1} \sum_{j=i+1}^n \left\{ \left[ (x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2 \right]^{-6} - 2 \left[ (x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2 \right]^{-3} \right\}$$

Se obtiene la derivada parcial de  $V$  con respecto a  $x_i$ :

$$\frac{\partial V}{\partial x_i} = \sum_{i=1}^{n-1} \sum_{j=i+1}^n \left\{ \begin{array}{l} -6 \left[ (x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2 \right]^{-7} \cdot 2 \cdot (x_i - x_j) \\ + 6 \left[ (x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2 \right]^{-4} \cdot 2 \cdot (x_i - x_j) \end{array} \right\}$$

$$\frac{\partial V}{\partial x_i} = \sum_{i=1}^{n-1} \sum_{j=i+1}^n \left\{ \frac{-12(x_i - x_j)}{\left[ (x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2 \right]^7} + \frac{12(x_i - x_j)}{\left[ (x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2 \right]^4} \right\}$$

De manera análoga se obtienen las derivadas parciales con respecto a  $y_i$  y  $z_i$ :

$$\frac{\partial V}{\partial y_i} = \sum_{i=1}^{n-1} \sum_{j=i+1}^n \left\{ \frac{-12(y_i - y_j)}{\left[ (x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2 \right]^7} + \frac{12(y_i - y_j)}{\left[ (x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2 \right]^4} \right\}$$

$$\frac{\partial V}{\partial z_i} = \sum_{i=1}^{n-1} \sum_{j=i+1}^n \left\{ \frac{-12(z_i - z_j)}{\left[ (x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2 \right]^7} + \frac{12(z_i - z_j)}{\left[ (x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2 \right]^4} \right\}$$

La matriz Hessiana de la función de LJ esta dada por:

$$H_V = \begin{bmatrix} \frac{\partial^2 V}{\partial x_i \partial x_i} & \frac{\partial^2 V}{\partial x_i \partial y_i} & \frac{\partial^2 V}{\partial x_i \partial z_i} \\ \frac{\partial^2 V}{\partial y_i \partial x_i} & \frac{\partial^2 V}{\partial y_i \partial y_i} & \frac{\partial^2 V}{\partial y_i \partial z_i} \\ \frac{\partial^2 V}{\partial z_i \partial x_i} & \frac{\partial^2 V}{\partial z_i \partial y_i} & \frac{\partial^2 V}{\partial z_i \partial z_i} \end{bmatrix}$$

Como consecuencia del número de variables implicadas en las primeras derivadas parciales, la determinación de los puntos críticos prácticamente no se puede realizar usando el método descrito.

## 2.4 Algoritmos metaheurísticos

Por su complejidad computacional el problema de LJ pertenece a una clase especial de problemas denominados *NP-duros* [Greenwood 2002]. Para este tipo de problemas, actualmente no se conocen algoritmos exactos eficientes para resolver instancias grandes y se tiene la sospecha de que no es posible construirlos. En consecuencia la solución de instancias grandes de problemas *NP-duros*, sólo se puede realizar simplificando el problema o utilizando un método de solución aproximada [Garey 1979, Papadimitriou 1998]. Los métodos heurísticos no deterministas de propósito general, denominados metaheurísticos [Glover 1986], son considerados actualmente herramientas muy prometedoras para la solución aproximada de instancias grandes de este tipo de problemas [Barr 2001, Michalewicz 1999].

Los metaheurísticos ofrecen soluciones “cercanas” a la solución óptima del problema con una inversión razonable de recursos [Glover 1986, Feo 1995, Michalewicz 1999, Barr 2001, Yagiura 2001, Resende 2003]. Actualmente existe una diversidad de metaheurísticos disponibles, tales como: Algoritmos Genéticos [Holland 1975, Goldberg 1989, Romero 1999b, Pérez 2004], Recocido Simulado [Kirkpatrick 1983, Cerny 1985, Sanvicente 2004], Búsqueda Tabú [Glover 1986, Glover 1997], Aceptación por Umbral [Dueck 1990, Pérez 2000, Pérez 2002], Colonia de Hormigas [Dorigo 1992, Dorigo 2003], GRASP [Feo 1995, González-Velarde 1996, Resende 2003] y Búsqueda Dispersa [Glover 1999].

Básicamente los metaheurísticos definen estrategias de navegación a través del espacio de soluciones que les permiten escapar de los óptimos locales y aproximarse al óptimo global. Esta propiedad los convierte en una alternativa prometedora para la construcción de métodos de optimización global, en los que colaboren con un método de búsqueda local. Este enfoque

es el que más comúnmente se está utilizando en la construcción de métodos de solución del problema de LJ.

### **2.4.1 GRASP**

Un Procedimiento de Búsqueda Miope, Aleatorizado y Adaptativo, GRASP por sus siglas en inglés, es un metaheurístico para encontrar soluciones aproximadas a problemas de optimización combinatoria. Se basa en la premisa de que soluciones iniciales diversas y de buena calidad juegan un papel importante en el desarrollo de métodos de búsqueda global [Resende 2003].

El método consiste en un proceso iterativo donde se llevan a cabo dos fases: la construcción de una solución miope aleatorizada y un proceso de búsqueda local. La Figura 2.7 muestra el algoritmo de un GRASP básico. En la primera fase una solución miope aleatorizada inicial es construida elemento por elemento. GRASP utiliza una lista restringida de candidatos (LRC) que constituye el factor aleatorio del método. Para construir la lista, se seleccionan aquellos elementos cuyo valor se encuentre en un rango dado. De la LRC se elige un elemento para formar parte de la solución. La Figura 2.8 muestra un algoritmo para el procedimiento Miope Aleatorio Adaptativo. Una vez que ha terminado la fase de construcción, se realiza un proceso de búsqueda sobre la vecindad de la solución obtenida. La Figura 2.9 muestra un procedimiento de Búsqueda Local. Mientras que la solución en la vecindad no sea la óptima se repite la búsqueda hasta que se encuentre una mejor solución. Se actualiza la solución general y se procede a realizar una nueva iteración hasta llegar al número máximo de iteraciones establecido. El mejor valor encontrado hasta este momento será el resultado final del metaheurístico.

```

Procedimiento GRASP(MaxIteraciones, Semilla)
  Leer entrada
  Para (k=1, ..., MaxIteraciones) Hacer
    Solución←MiopeAleatorioAdaptativo(Semilla)
    Solución←BúsquedaLocal(Solución)
    ActualizarSolución(Solución, MejorSolución)
  Fin Para
  Regresar MejorSolución
Fin GRASP
    
```

Figura 2.7 Algoritmo GRASP.

```

Procedimiento MiopeAleatorioAdaptativo(Semilla)
  Solución←∅
  Evaluar el incremento del costo de los elementos candidatos
  Mientras que la Solución no este completa Hacer
    Construir la lista restringida de candidatos (LRC)
    Seleccionar aleatoriamente un elemento S de LRC
    Solución← Solución ∪ {S}
    Reevaluar el incremento del costo S
  Fin Mientras
  Regresar Solución
Fin MiopeAleatorioAdaptativo
    
```

Figura 2.8 Algoritmo miope aleatorio adaptativo.

```

Procedimiento BúsquedaLocal(Solución)
  Mientras Solución no sea óptima local Hacer
    Encontrar  $S' \in V(\text{Solución})$  con  $f'(S') < f(\text{Solución})$ 
    Solución←  $S'$ 
  Fin Mientras
  Regresar Solución
Fin BúsquedaLocal
    
```

Figura 2.9 Algoritmo de búsqueda local.

## 2.4.2 Estructura de vecindades

Dada una instancia de un problema de optimización combinatoria  $(S, c)$ , donde  $S$  es el conjunto de las soluciones factibles y  $c$  es el costo de la función objetivo  $c: S \rightarrow R$ . Una

estructura de vecindades  $V : S \rightarrow 2^S$ , asocia a cada solución  $s \in S$  un subconjunto  $V(s)$  de  $S$  [Papadimitriou 1998].

- $V(s)$  es llamada una vecindad de  $s$  y contiene todas las soluciones vecinas o “cercanas” en algún sentido a la solución  $s$ .
- Si  $r \in V(s)$  entonces se dice que  $r$  es un vecino de  $s$ .

Si  $S = R^n$ , el conjunto de soluciones con una distancia Euclidiana fija proporciona una vecindad. En muchos problemas combinatorios, la elección de  $V$  puede depender críticamente de la estructura de  $S$ .

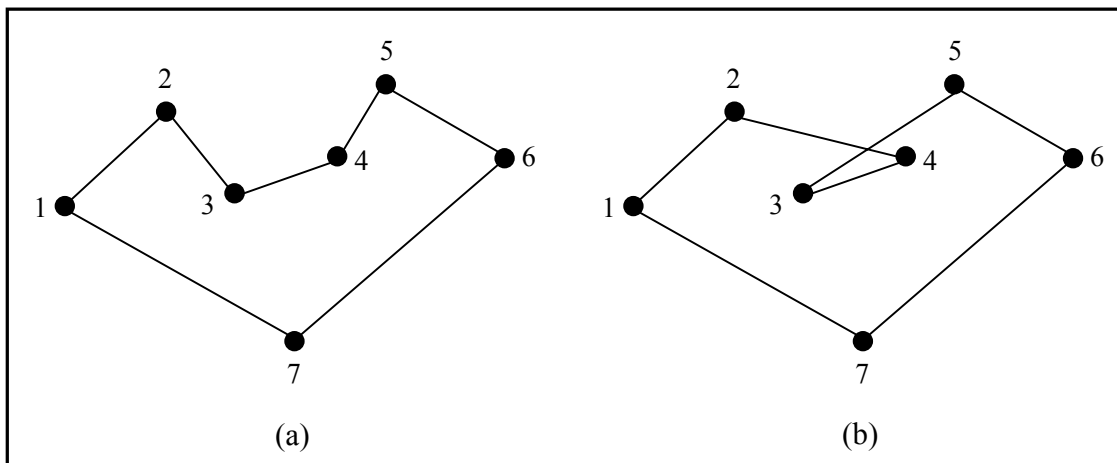
Por ejemplo, en el TSP se puede definir una vecindad llamada 2-cambios por:

$$V_2(s) = \left\{ r : r \in S \text{ y } r \text{ se puede obtener de } s \text{ de la siguiente manera:} \right. \\ \left. \begin{array}{l} \text{remover dos lados del recorrido y reemplazarlos} \\ \text{por dos diferentes.} \end{array} \right\}$$

La Figura 2.10 muestra un ejemplo de un recorrido  $s$  y otro recorrido  $r \in V_2(s)$  para una instancia del TSP con siete ciudades y una matriz que contiene las distancias Euclidianas entre las ciudades.

La vecindad se puede generalizar como  $V_k$ , llamándola  $k$ -cambios, donde como máximo  $k$  lados son reemplazados.





**Figura 2.10** (a) Una instancia del TSP y un recorrido. (b) Otro recorrido, el cual tiene 2-cambios del recorrido que se ve en (a).

### 2.4.3 Lista restringida de candidatos

Las construcciones miope aleatorizadas construyen una solución incorporando un elemento a la vez. En cada paso del proceso de construcción, se tiene una solución parcial. Un elemento que pueda seleccionarse como parte de una solución parcial se llama elemento candidato. Generalmente se hace uso de un función miope para determinar qué elemento candidato seleccionar e incluirse en la solución. Una función miope mide la contribución local de cada elemento a la solución parcial. Para introducir aleatoriedad a esta función se utiliza una lista restringida de candidatos (LRC). Tal lista esta formada por los elementos candidatos con los mejores valores de la función miope. Por ejemplo, sean  $c^*$  y  $c_*$  los valores máximo y mínimo de la función miope sobre los candidatos considerados, y sea  $\alpha$  un número real tal que  $0 \leq \alpha \leq 1$ . La LRC consiste en todos los elementos candidatos  $e$  cuyo valor de función miope  $c(e)$  es tal que  $c(e) \leq c_* + \alpha(c^* - c_*)$  [Resende 2003].

## 2.5 Optimización numérica

El método del gradiente es un método numérico de optimización local que se aplica a funciones que son dos veces diferenciables. La idea es generar puntos sucesivos en la dirección del incremento más rápido (maximización) de la función, iniciando en un punto dado. El nombre de la técnica se debe a que el gradiente tiene la propiedad de que siempre apunta en la dirección de mayor incremento de la función.

### Método del Gradiente

Los pasos del método del gradiente [Mathews 2005] consisten en:

Dado un punto inicial  $P_k$ .

1. Evaluar el vector gradiente  $\nabla f(P_k)$ .
2. Calcular la dirección de búsqueda  $S_k = -\nabla f(P_k) / \|\nabla f(P_k)\|$
3. Ejecutar un proceso de minimización  $\Phi(\gamma) = f(P_k + \gamma S_k)$  en el intervalo de  $[0, b]$ , donde  $b$  es un valor grande. Esto producirá un valor  $\gamma = h_{\min}$  donde ocurre un mínimo local para  $\Phi(\gamma)$ . La relación  $\Phi(h_{\min}) = f(P_k + h_{\min} S_k)$  es un mínimo para  $f(X)$  a lo largo de la línea de búsqueda  $X = P_k + h_{\min} S_k$
4. Construir el siguiente punto  $P_{k+1} = P_k + h_{\min} S_k$
5. Ejecutar la prueba de terminación para minimización, esto es, ¿Son los valores de las funciones  $f(P_k)$  y  $f(P_{k+1})$  suficientemente cercanas y la distancia  $\|P_{k+1} - P_k\|$  bastante pequeña?

Repetir el proceso.

Ejemplo: Dado  $f(x, y) = \frac{x-y}{x^2 + y^2 + 2}$  y el punto inicial  $P_0 = (-3, -2)$

Encontrar el gradiente de la función  $\nabla f(x, y)$ .

$$\frac{\partial(x, y)}{\partial x} = \frac{-x^2 + 2xy + y^2 + 2}{(x^2 + y^2 + 2)^2} \quad y \quad \frac{\partial(x, y)}{\partial y} = \frac{-x^2 - 2xy + y^2 - 2}{(x^2 + y^2 + 2)^2}$$

Calcular  $P_1$ .

Paso 1.

Sustituyendo el punto inicial, se tiene

$$\nabla f(-3, -2) = \left( \frac{\partial(-3, -2)}{\partial x}, \frac{\partial(-3, -2)}{\partial y} \right) = \left( \frac{9}{225}, -\frac{19}{225} \right)$$

Paso 2.

Calcular la dirección de búsqueda,

$$S_0 = \frac{1}{\|-\nabla f(P_0)\|} (-\nabla f(P_0))$$

$$S_0 = \frac{1}{\|-\nabla f(-3, -2)\|} (-\nabla f(-3, -2))$$

$$S_0 = (-0.4280863, 0.9037378)$$

Paso 3.

$$f(P_0 + \gamma S_0) = f((-3, -2) + \gamma(-0.4280863, 0.9037378))$$

$$f(P_0 + \gamma S_0) = f((-3 - 0.4280863\gamma, -2 + 0.9037378\gamma))$$

$$f(P_0 + \gamma S_0) = \frac{(-3 - 0.4280863\gamma) - (-2 + 0.9037378\gamma)}{(-3 - 0.4280863\gamma)^2 + (-2 + 0.9037378\gamma)^2 + 2}$$

Por interpolación cuadrática la función  $f(P_0 + \gamma S_0)$  tiene un mínimo en 4.8186760.

$$\gamma = h_{\min_0} = 4.8186760$$

Paso 4.

Construir el siguiente punto.

$$P_1 = P_0 + h_{\min_0} S_0$$

$$P_1 = (-3, -2) + 4.8186760(-0.4280863, 0.9037378)$$

$$P_1 = (-5.0628094, 2.3548199)$$

Calcular  $P_2$ .

Paso 2.

Calcular la dirección de búsqueda,

$$S_1 = \frac{1}{\|-\nabla f(P_1)\|} (-\nabla f(P_1))$$

$$S_1 = \frac{1}{\|-\nabla f(-5.0628094, 2.3548199)\|} (-\nabla f(-5.0628094, 2.3548199))$$

$$S_1 = (0.9991231, -0.0418690)$$

Paso 3.

$$f(P_1 + \gamma S_1) = f((-5.0628094, 2.3548199) + \gamma(0.9991231, -0.0418690))$$

$$f(P_1 + \gamma S_1) = f(-5.0628094 + 0.9991231\gamma, 2.3545199 - 0.0418690\gamma)$$

$$f(P_1 + \gamma S_1) = \frac{(-5.0628094 + 0.9991231\gamma) - (2.3545199 - 0.0418690\gamma)}{(-5.0628094 + 0.9991231\gamma)^2 + (2.3545199 - 0.0418690\gamma)^2 + 2}$$

Por interpolación cuadrática la función  $f(P_1 + \gamma S_1)$  tiene un mínimo en 2.7708281.

$$\gamma = h_{\min_1} = 2.7708281$$

Paso 4.

Construir el siguiente punto.

$$P_2 = P_1 + h_{\min_1} S_1$$

$$P_2 = (-5.0628094, 2.3548199) + 2.7708281(0.9991231, -0.0418690)$$

$$P_2 = (-2.2944111, 2.2388080)$$

Y así sucesivamente, la función  $f(x, y) = \frac{x-y}{x^2 + y^2 + 2}$  tiene un punto mínimo local en

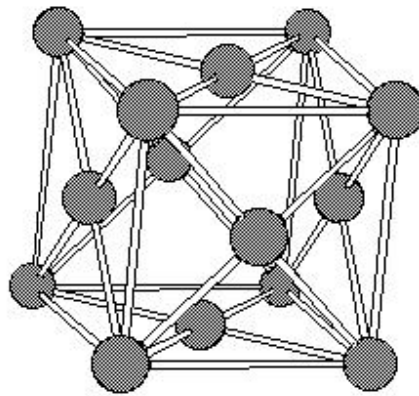
$$P = (-1, 1).$$

## 2.6 Rejillas

Para discretizar en alguna medida el problema de LJ, se utilizan estructuras reticulares de puntos del espacio denominadas retículas o rejillas. Las rejillas son utilizadas para definir los sitios en los que se pueden colocar inicialmente los átomos de un grupo de LJ. A esta configuración inicial de los átomos se le aplica un método de búsqueda local para tratar de transformarla en la configuración local óptima más cercana. Generalmente una rejilla tiene una cierta estructura geométrica básica la que determina que puntos del espacio forman los sitios de la rejilla. Algunas de las estructuras utilizadas son por ejemplo: “cuadrangular”, “triangular”, “diamante”, “cúbica”, etc. [Meyer 2001]. Cada uno de los átomos de un grupo se ubica en un sitio de una rejilla, pero no todos los sitios son ocupados por un átomo del grupo.

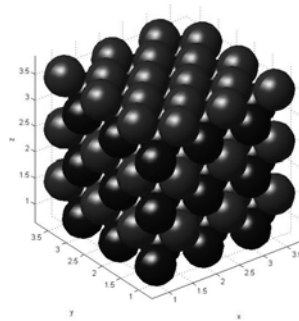
### 2.6.1 Rejilla FCC

Los sitios de una rejilla FCC (Face Centered Cubic) son los vértices y los centros de las caras de estructuras cúbicas básicas [Watkins 2006]. Los sitios están alineados en diagonal, como se puede apreciar en la Figura 2.11.



**Figura 2.11** Rejilla FCC.

Una característica de la rejilla FCC es que esta formada de 4 capas. Para una rejilla FCC de tamaño 3x3x3, se generan 4 capas, cada una con 27 sitios. En la Figura 2.12 se muestra como los sitios de la rejilla se apilan de manera similar a la manera en que se apila un conjunto de esferas.



**Figura 2.12** Sitios de la rejilla FCC.

## 2.6.2 Rejilla icosaedral

Una rejilla icosaedral  $IC_d$  consiste de  $d$  de capas icosaedrales concéntricas [Xue 1992]. La rejilla icosaedral  $IC_1$  consiste de un sitio localizado en el origen. Para  $d = 2, 3, \dots, k$ , la rejilla icosaedral  $IC_d$  incluye todos los sitios que se describen a continuación:

- a) Los sitios de la capa anterior  $IC_{d-1}$ .
- b) Los 12 sitios correspondientes a los vértices del icosaedro centrado en el origen, ubicado a una distancia  $\tau = \sigma(d-1)$ , donde  $\sigma = 1$ . Se fijan los 2 primeros sitios en las posiciones  $p_1 = (0, 0, \tau)$  y  $p_2 = (0, 0, -\tau)$ . El resto de los sitios se determina de la siguiente manera:

$$p_{k+3} = \begin{cases} x_{k+3} = \tau \cos((-1)^k \theta) \cos(k\pi/5) \\ y_{k+3} = \tau \cos((-1)^k \theta) \sin(k\pi/5) \\ z_{k+3} = \tau \sin((-1)^k \theta) \end{cases} \quad (2.2)$$

$$k = 0, \dots, 9$$

$$\theta = \pi/2 - \tan^{-1} 2$$

- c) Para cada par de vértices  $p_a, p_b$  que definen una arista del icosaedro referido anteriormente, las coordenadas de los  $d-2$  sitios ubicados sobre dicha arista están dados por:

$$p_a + i(p_b - p_a)/(d-1), \quad i = 1, \dots, d-2 \quad (2.3)$$

- d) Para cada terna de vértices  $p_a, p_b, p_c$  que definen una cara del icosaedro considerado en b), las coordenadas de los  $(d-2)(d-3)/2$  sitios ubicados sobre dicha cara están dados por:

$$\begin{aligned} & (ip_a + jp_b + (d-1-i-j)p_c)/(d-1), \\ & j = 1, \dots, d-i-2, \quad i = 1, \dots, d-3 \end{aligned} \tag{2.4}$$

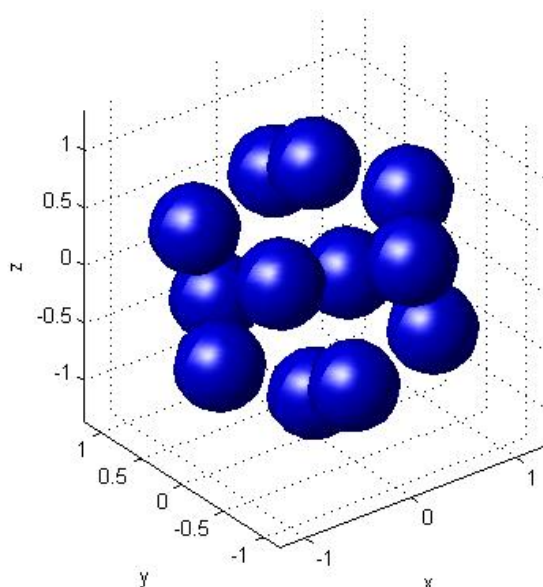
El número de sitios de  $IC_d$  es  $|IC_d| = (10d^3 - 15d^2 + 11d - 3)/3$ .

La Tabla 2.6 muestra las coordenadas y la Figura 2.13 la ubicación de los 12 sitios de la rejilla  $IC_2$ .

**Tabla 2.6** Coordenadas de los sitios de la rejilla  $IC_2$ .

<b>x</b>	<b>y</b>	<b>z</b>
0	0	1.0000
0	0	-1.0000
0.8944	0	0.4472
0.7236	0.5257	-0.4472
0.2764	0.8507	0.4472
-0.2764	0.8507	-0.4472
-0.7236	0.5257	0.4472
-0.8944	0.0000	-0.4472
-0.7236	-0.5257	0.4472
-0.2764	-0.8507	-0.4472
0.2764	-0.8507	0.4472
0.7236	-0.5257	-0.4472





**Figura 2.13** Sitios de la rejilla icosaedral  $IC_2$ .

Los sitios de cada capa se localizan sobre un icosaedro formado por 12 vértices, 30 aristas y 20 caras. La Tabla 2.7 muestra el número de sitios en: los vértices, las aristas y las caras de cada una de las capas de la rejilla  $IC_5$ .

**Tabla 2.7** Número de sitios en una rejilla icosaedral de 5 capas.

No. Capa $d$	1	2	3	4	5	Número de sitios en una rejilla icosaedral de 5 capas = 309.
No. de sitios en los vértices	1	12	12	12	12	
No. sitios en las aristas = $30 * (d - 2)$	0	0	30	60	90	
No. sitios en las caras = $20 * \frac{(d - 2)(d - 3)}{2}$	0	0	0	20	60	
Total de sitios por capa	1	12	42	92	162	

## 2.7 Rebanada esférica

Dado un punto  $p \in R^3$  y un entero  $m > 0$ , una rebanada esférica  $S(p, m)$  de una rejilla IC (o FCC) es el conjunto de los  $m$  sitios de la rejilla más cercanos a  $p$  [Romero 1999b].

Los sitios de una rebanada esférica son utilizados para colocar todos los átomos de un grupo de LJ. Esta técnica es la base de los métodos de solución que utilizan rejillas, ya que permite generar de una manera muy eficiente configuraciones de baja energía.

Por otra parte, la definición de rebanada esférica no especifica de qué manera se debe elegir el punto  $p$ . Para su elección se consideran las siguientes dos posibilidades: elegir un punto del espacio que corresponda a un sitio de la rejilla o elegirlo sin esta restricción. En el segundo caso el rango para la elección del punto se limita a un cubo que contenga a la rejilla. Para referir uno u otro método se indicará que el punto se elige en los sitios o en la rejilla.

A continuación se presentarán dos ejemplos de rebanadas esféricas en rejilla icosaedral, en el primer ejemplo el punto elegido es un sitio de la rejilla y en el segundo ejemplo el punto es un punto de la rejilla.

Dada la rejilla  $IC_2$  y el punto  $p(0.2763, -0.8506, 0.4472)$  que en este caso es el sitio 12  $S_{12}$  de la rejilla; determinar los puntos de la rejilla que forman la rebanada esférica  $S(p, 5)$ .

Los sitios de la rejilla  $IC_2$  se pueden apreciar en la Tabla 2.8,

**Tabla 2.8** Distancia del punto  $p(0.2763, -0.8506, 0.4472)$  a cada uno de los sitios de  $IC_2$ .

Sitio	x	y	z	Distancia	Orden en función de la distancia
1	0.000000	0.000000	0.000000	1.00000000279243	12
2	0.000000	0.000000	1.000000	1.05146222261423	1
3	0.000000	0.000000	-1.000000	1.70130162099049	2
4	0.894427	0.000000	0.447214	1.05146222630629	10
5	0.723607	0.525731	-0.447214	1.70130162223742	4
6	0.276393	0.850651	0.447214	1.70130162000000	11
7	-0.276393	0.850651	-0.447214	2.00000000558486	13
8	-0.723607	0.525731	0.447214	1.70130161632289	8
9	-0.894427	0.000000	-0.447214	1.70130162002311	6
10	-0.723607	-0.525731	0.447214	1.05146222540236	9
11	-0.276393	-0.850651	-0.447214	1.05146222952838	3
12	0.276393	-0.850651	0.447214	0.00000000000000	5
13	0.723607	-0.525731	-0.447214	1.05146223497227	7

Como se puede observar en la columna seis de la tabla anterior los cinco sitios más cercanos al punto  $p(0.2763, -0.8506, 0.4472)$  son los sitios  $S(p, 5) = \{S_{12}, S_1, S_2, S_{10}, S_4\}$ .

Dada la rejilla  $IC_2$  y el punto de la rejilla  $p(1, 0, 1)$ ; determinar los puntos de la rejilla que forman la rebanada esférica  $S(p, 5)$ .

La Tabla 2.9 muestra los sitios de la rejilla  $IC_2$  y la distancia del punto  $p(1, 0, 1)$  a cada uno de los sitios.

De acuerdo a la columna seis de la Tabla 2.9 la rebanada esférica de tamaño 5 está conformada por los sitios  $S(p, 5) = \{S_4, S_2, S_6, S_{12}, S_1\}$ .

**Tabla 2.9** Distancia del punto  $p(1, 0, 1)$  a cada uno de los sitios de  $IC_2$ .

Sitio	x	y	z	Distancia	Orden en función de la distancia
1	0.000000	0.000000	0.000000	1.41421356237310	4
2	0.000000	0.000000	1.000000	1.00000000000000	2
3	0.000000	0.000000	-1.000000	2.23606797749979	6
4	0.894427	0.000000	0.447214	0.56277741802266	12
5	0.723607	0.525731	-0.447214	1.56435724981637	1
6	0.276393	0.850651	0.447214	1.24610850473980	5
7	-0.276393	0.850651	-0.447214	2.10884176874057	13
8	-0.723607	0.525731	0.447214	1.88488365822749	8
9	-0.894427	0.000000	-0.447214	2.38396341881252	10
10	-0.723607	-0.525731	0.447214	1.88488365822749	7
11	-0.276393	-0.850651	-0.447214	2.10884176874057	11
12	0.276393	-0.850651	0.447214	1.24610850473980	3
13	0.723607	-0.525731	-0.447214	1.56435724981637	9

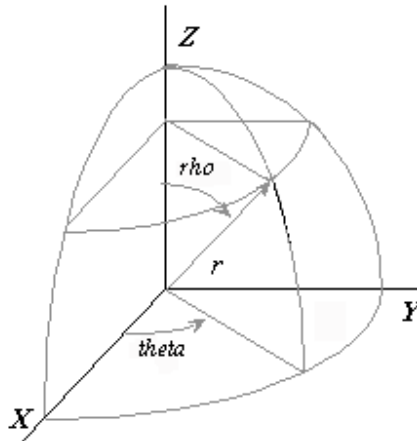
## 2.8 Coordenadas esféricas

El sistema de coordenadas esféricas se utiliza para determinar la posición espacial de un punto mediante una distancia y dos ángulos. En consecuencia, un punto  $P$  queda representado por un conjunto de tres magnitudes: el radio  $r$ , el ángulo polar o colatitud  $\rho$  (rho) y el azimut  $\theta$  (theta) [Purcell 1987].

La convención utilizada en este trabajo es:

- $r$  (radio): es la distancia entre el punto  $P$  y el origen.
- $\rho$  (colatitud o ángulo polar) de  $0^\circ$  a  $180^\circ$  es el ángulo entre el eje  $z$  y la línea que une el origen y el punto  $P$ , y
- $\theta$  (acimut o longitud) de  $0^\circ$  a  $360^\circ$  es el ángulo entre el eje  $x$  positivo y la línea que une el origen con la proyección del punto  $P$  en el plano  $xy$ .

La Figura 2.14 muestra los ejes  $x, y, z$ , los dos ángulos y el radio.



**Figura 2.14** Sistema de coordenadas esféricas.

Las relaciones entre  $r, \rho, \theta$  y  $x, y, z$  son las siguientes [Hsu 1973]:

$$x = r * \text{sen}(\rho) * \cos(\theta) \quad (2.5)$$

$$y = r * \text{sen}(\rho) * \text{sen}(\theta) \quad (2.6)$$

$$z = r * \cos(\rho) \quad (2.7)$$

## 2.9 Técnica de variables antitéticas

Cuando se trabaja con algoritmos aleatorios se deben aplicar técnicas de reducción de la varianza para asegurar la calidad de los resultados experimentales y la solidez de las conclusiones [Barr 2001, Johnson 2002, McGeoch 2002, Moret 2003]. En esta sección se describe la técnica de reducción de la varianza de variables antitéticas [Ross 1999].

### 2.9.1 Técnicas de reducción de la varianza

Uno de los objetivos de un estudio de simulación es determinar un parámetro  $\theta$ , asociado con algún modelo aleatorio. Para estimar  $\theta$ , el modelo se simula para obtener el dato de salida  $x$  tal que  $\theta = E[x]$ . Se realizan varias ejecuciones de la simulación, en las que la  $i$ -ésima ejecución produce la salida  $x_i$ . El estudio concluye después de  $n$  ejecuciones y la estimación de  $\theta$  está dada por  $\bar{x} = \sum_i x_i / n$  y su varianza por  $E[(\bar{x} - \theta)^2] = \text{Var}(\bar{x}) = \text{Var}(x) / n$  [Ross 1999]. Si es posible obtener otra estimación de  $\theta$  con menor varianza que  $\bar{x}$ , se tendría un estimador cuya determinación requeriría un menor número de experimentos [McGeoch 1992].

### 2.9.2 Técnica de variables antitéticas

Sean  $x$  y  $y$  variables aleatorias con la misma distribución, tales que  $\theta = E[x] = E[y]$ . Entonces  $E[(x + y) / 2] = \theta$ , por lo que  $z = (x + y) / 2$  es un estimador de  $\theta$ . Además como:  $\text{Var}(z) = \text{Var}((x + y) / 2) = \frac{1}{4} [\text{Var}(x) + \text{Var}(y) + 2\text{Cov}(x, y)]$ , se observa que la varianza del estimador  $z$  se reduce si  $\text{Cov}(x, y) < 0$  [Ross 1999].

Para lograr que  $x$  e  $y$  estén correlacionadas negativamente, lo que se puede hacer es utilizar un generador uniforme de números aleatorios  $U$ , para asignar números complementarios a ambas variables ( $x = U$  e  $y = 1 - U$ ). Detalles adicionales de este desarrollo pueden ser revisados en [Ross 1999].

## 2.10 Trabajos relacionados

En esta sección se describirán los trabajos relacionados de análisis experimental de algoritmos y de la solución del problema de LJ.

### 2.10.1 Análisis experimental de algoritmos

Para realizar el estudio experimental del problema de LJ, se consideró conveniente revisar el estado del arte del análisis experimental de algoritmos para determinar los requisitos fundamentales de un estudio de este tipo. En esta sección se presenta un resumen de los trabajos más relevantes que se revisaron.

Barr y Johnson señalan que la efectividad de una metodología se puede probar mediante análisis teórico o con pruebas experimentales [Barr 2001, Johnson 2002]. Señalan también que el análisis teórico resulta muchas veces limitado para analizar el desempeño de los algoritmos en aplicaciones reales. El enfoque del peor caso tiende a producir cotas demasiado pesimistas, y no provee información del desempeño de los algoritmos sobre instancias específicas [McGeoch 1992]. El enfoque del caso promedio requiere la determinación de un modelo adecuado de la distribución de los datos asociados a las instancias utilizadas, lo que muchas veces resulta difícil de realizar [McGeoch 1992].

Johnson señala que en los últimos años la comunidad de los analistas teóricos ha mostrado un creciente interés en el análisis experimental de algoritmos. Dicho interés se debe al reconocimiento de que el análisis teórico es insuficiente para analizar el rendimiento de los algoritmos del mundo real [Johnson 2002].

Johnson, Barr, McGeoch, Moret y Fraire proponen una serie de principios, para el diseño de experimentos computacionales que permiten producir conocimiento científico relevante [Johnson 2002, Barr 2001, McGeoch 2002, Moret 2003, Fraire 2005]. Señalan que el análisis experimental de algoritmos debe incluir experimentos relevantes, vinculados con la literatura, que se puedan reproducir y comparar con los realizados en otras investigaciones, y que utilicen implementaciones razonablemente eficientes y casos de prueba estándar.

## 2.10.2 Solución del problema de LJ

En este apartado se muestra el estado del arte del problema de LJ. La mayoría de los trabajos revisados utilizan estructuras geométricas (rejillas), en las cuales los átomos se colocan para establecer una configuración inicial. Sólo un trabajo no utiliza rejillas, aquí los átomos se colocan en forma aleatoria en el espacio tridimensional. Dentro de la revisión también se detectaron dos enfoques de solución al problema de LJ, ver la Tabla 2.10.

**Tabla 2.10** Enfoques de solución en los trabajos revisados.

Enfoque	Descripción
1	Navegar colocando los átomos en las estructuras geométricas, creando configuraciones iniciales diversas, evaluar cada configuración y obtener la mejor para optimizarla.
2	Navegar colocando los átomos en las estructuras geométricas, creando configuraciones iniciales diversas, optimizar cada configuración y obtener la mejor.

El enfoque general empleado en estos trabajos para la solución del problema de LJ es la utilización de un método de búsqueda local y un metaheurístico de búsqueda global. El método de búsqueda local lleva a la configuración inicial hacia el óptimo local más cercano y el metaheurístico permite escapar del gran número de mínimos locales que existen en la función de LJ [Wales1999].

Locatelli propone un algoritmo multiarranque para la búsqueda global [Locatelli 2000]. No utiliza estructuras geométricas, lo que hace es generar aleatoriamente sitios dentro de un espacio tridimensional y colocar los átomos en tales sitios. El enfoque de solución es navegar en los sitios, para posteriormente crear configuraciones iniciales, evaluar cada configuración y obtener la mejor para optimizarla mediante L-BFGS. En promedio realiza 10,000 evaluaciones de la función objetivo. No tiene un control estadístico de los resultados.



Romero propone un algoritmo genético [Romero 1999a], los átomos los coloca en estructuras geométricas icosaedrales y decaedrales. El enfoque de solución que emplea es obtener configuraciones iniciales, cada configuración obtenida se optimiza con un método de optimización local, el cual no lo define. Romero reporta el número de evaluaciones realizadas a la función objetivo y la generación donde encuentra la mejor solución para los grupos de átomos resueltos, pero no presenta como se validan estos resultados.

Romero propone nuevamente un algoritmo genético en [Romero 1999b]. Utiliza una rejilla icosaedral para colocar los átomos. Trabaja con el mismo enfoque de solución que en [Romero 1999a]. No reporta el número de evaluaciones de la función objetivo y tampoco la generación donde encuentra el mejor valor, solo presenta el número de vecinos más cercanos en cada grupo, la distancia del vecino más cercano y la distancia del vecino más lejano.

Shao en [Shao 2004a, Shao 2004b, Shao 2005] utiliza un enfoque equivalente a Locatelli, navega colocando los átomos, crea configuraciones iniciales, evalúa cada configuración hasta que encuentra la mejor configuración y la optimiza, para la optimización utiliza el método L-BFGS. A diferencia de Locatelli, Shao utiliza estructuras geométricas para posicionar los átomos para cada configuración creada. En [Shao 2004a] propone la construcción del icosaedron, decaedro de Ino y el octaedron completo para formar las rejillas, utiliza algoritmos genéticos para la búsqueda global. Lo que reporta es el tipo de estructura geométrica de los grupos en donde encuentra la mejor solución. En [Shao 2004b, Shao 2005] utiliza rejillas icosaedrales y decaedrales para formar las rejillas y un procedimiento de búsqueda voraz para la búsqueda global, sólo reporta que el número de intentos para obtener la mejor configuración inicial es de 100,000 o más.

En la Tabla 2.11 describen las características más relevantes de los trabajos relacionados con el problema de LJ.

**Tabla 2.11** Trabajos relacionado con la solución del problema de LJ.

Trabajo	Estructuras geométricas	Enfoque de solución (Tabla 2.10)	Búsqueda global	Búsqueda local	Control estadístico
Locatelli 2000		1	Multiarranque	L-BFGS	
Romero 1999a	✓	2	AG	No se define	
Romero 1999b	✓	2	AG	No se define	
Shao 2004a	✓	1	AG	L-BFGS	
Shao 2004b	✓	1	Voraz	L-BFGS	
Shao 2005	✓	1	Voraz	L-BFGS	
Esta tesis	✓	2	GRASP	Regiones de Confianza	✓

En esta tesis se utilizará la estructura geométrica icosaedral para la colocación de los átomos, dado que las estructuras han demostrado ser una buena estrategia [Romero 1999a, Romero 1999b, Shao 2004a, Shao 2004b, Shao 2005]. El enfoque que se empleará es navegar creando configuraciones iniciales diversas y optimizar cada configuración creada [Romero 1999a, Romero 1999b]. Para la búsqueda global se propone utilizar por primera ocasión un metaheurístico GRASP y para la optimización de la búsqueda local se usará el método Regiones de Confianza [Moré 1983], el cual no ha sido utilizado para la solución del problema. Además en este trabajo se incorpora un método de control de la varianza para validar los resultados experimentales que se desarrollan.

# Capítulo 3

## MÉTODO DE SOLUCIÓN

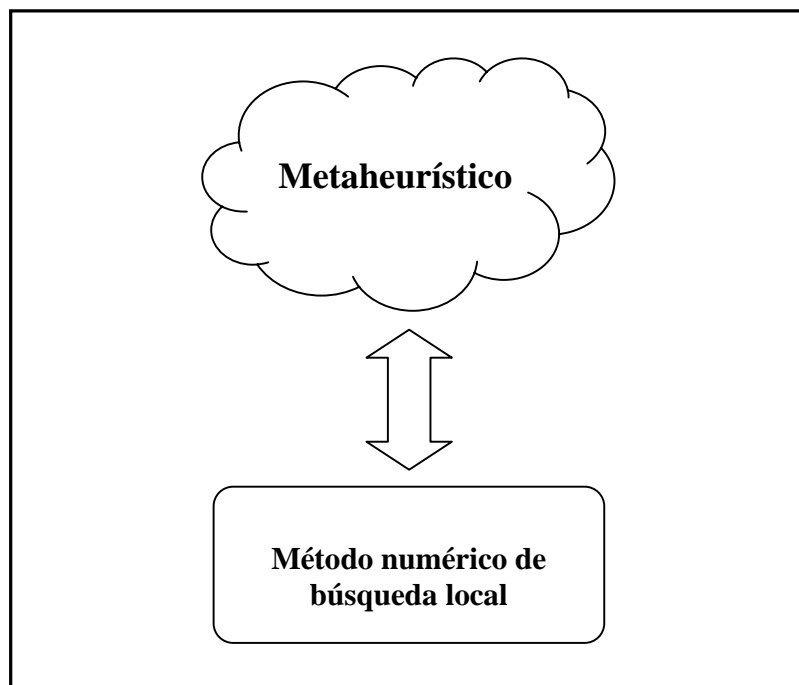
---

En este capítulo se describe el enfoque que se propone para la solución del problema de Lennard-Jones (LJ). El método propuesto es un método de optimización global, el cual consiste de un metaheurístico de búsqueda global y un método numérico de búsqueda local. También se presenta un método para reducir la varianza de los resultados en experimentos con algoritmos metaheurísticos.

### 3.1 Enfoque de solución

El enfoque propuesto es un método de optimización global que integra dos elementos: un metaheurístico de búsqueda global y un método numérico de búsqueda local (ver Figura 3.1).

La función principal del metaheurístico es evitar que el proceso de búsqueda global, se quede atrapado en los óptimos locales. La función del buscador local es refinar las configuraciones que se generan con base en rebanadas esféricas de la rejilla icosaedral. El metaheurístico que se implementó es un GRASP [Resende 2003] y para la búsqueda local se utilizó el método de Regiones de Confianza [Moré 1983]. Para la generación de configuraciones iniciales y la estructura de vecindades (configuraciones vecinas) se realiza con base en las técnicas reportadas en [Romero 1999a].



**Figura 3.1** Método de optimización global.

La Figura 3.2 muestra más a detalle el esquema de solución propuesto para el problema de LJ. Básicamente lo que se muestra en la figura es el metaheurístico GRASP, el método numérico de búsqueda local no se representa aquí; dado que interviene internamente en los procesos de construcción de una solución miope aleatorizada y de búsqueda local. Más adelante cuando se traten dichos procesos, también se hablará del método numérico de búsqueda local.

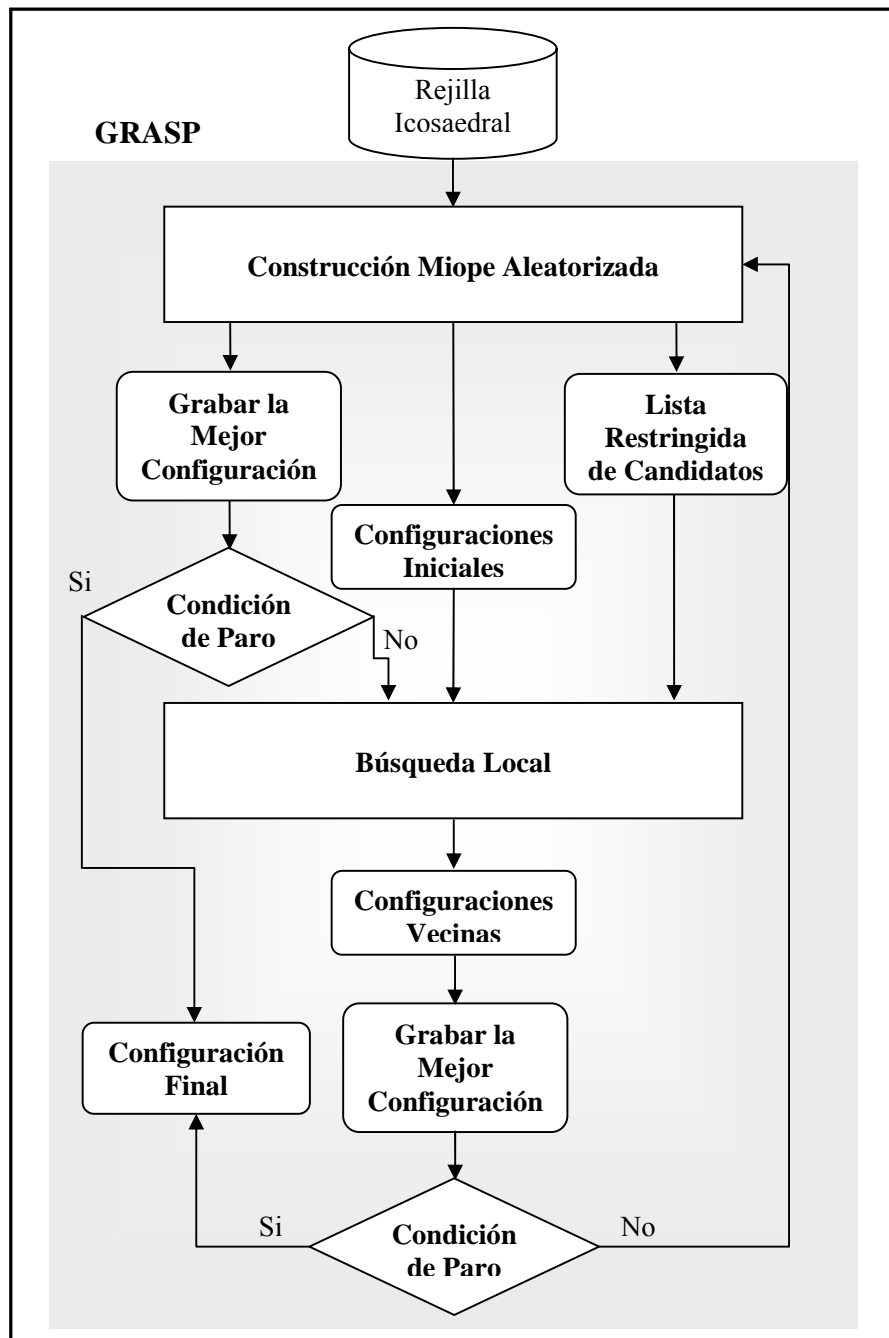


Figura 3.2 Esquema de solución propuesto para el problema de L.J.

Como primer paso en la solución propuesta se toman las coordenadas de una rejilla icosaedral, las cuales se utilizan en la construcción miopie aleatorizada. En la construcción miopie

aleatorizada se forman configuraciones iniciales, se obtiene la lista restringida de candidatos y al salir se guarda la mejor configuración hasta el momento. Si la configuración guardada cumple la condición de paro, esa es la configuración final que se va a obtener; sino la cumple se continúa con la búsqueda local. De acuerdo a la lista restringida de candidatos que se tiene, en la búsqueda local se forman configuraciones vecinas. Al final se obtiene la mejor configuración de todas las configuraciones vecinas y si esta configuración es mejor a la que se encuentra guardada, la reemplaza. Después si se cumple la condición de paro, la configuración almacenada es la configuración final, sino se cumple la condición se continúa el ciclo de iteraciones en el proceso de construcción miope aleatorizada. En las siguientes secciones de este capítulo se detallan los procesos implicados en esta solución.

### **3.2 Procedimiento de búsqueda miope, aleatorizado y adaptativo**

El algoritmo GRASP que se implementó, básicamente consta de dos procesos: la construcción miope aleatorizada y la búsqueda local. La Figura 3.3 muestra el algoritmo GRASP incorporado en el método propuesto. La meta del algoritmo es encontrar las mejores configuraciones para un grupo de átomos dado. La energía de LJ de cada configuración determina si la configuración es buena o no. Como entrada el algoritmo lee las coordenadas de la rejilla icosaedral. En la línea 3 inicialmente *MenorEnergía* se hace igual a un número grande, en este caso se le dio el valor de mil. Esta variable *MenorEnergía* se va actualizar cada vez que se encuentre una energía mejor a la que se tiene almacenada, esto se puede apreciar en las líneas 7 y 17 al finalizar los procesos de construcción miope aleatorizada y el de búsqueda local. El número de iteraciones lo tiene *NumIteraciones*, en este caso se manejaron cuatrocientas iteraciones; pero cabe señalar que no se llevan a cabo las cuatrocientas iteraciones. A partir de que se encontró una energía menor y si en las próximas diez iteraciones no se ha encontrado una mejor, se guarda esta configuración menor y se da por terminado el GRASP, en las líneas 13 y 23 se muestra esto.

Algoritmo GRASP	
<u>Objetivo:</u> Encontrar las mejores configuraciones para el grupo de átomos dado.	
<u>Entrada:</u> Rejilla icosaedra	
<u>Funciones:</u>	
<i>ConstrucciónMiopeAleatorizada(Rejilla)</i> . Obtiene las configuraciones iniciales, la mejor configuración y la lista restringida de candidatos (LRC).	
<i>BusquedaLocal(Configuraciones Iniciales, LRC)</i> . Obtiene la mejor configuración de las configuraciones vecinas.	
<u>Salida:</u> Mejores configuraciones	
1	Leer la rejilla icosaedra
2	Pedir el átomo inicial y el átomo final
3	<i>MenorEnergía</i> =1000
4	<b>Para</b> ( $n=\text{ÁtomoInicial}, \dots, \text{ÁtomoFinal}$ ) <b>Hacer</b>
5	<b>Para</b> ( $iter=1, \dots, \text{NumIteraciones}$ ) <b>Hacer</b>
6	<i>Energía, Configuración, Código</i> ← <i>ConstrucciónMiopeAleatorizada(Rejilla)</i>
7	<b>Si</b> <i>Energía</i> < <i>MenorEnergía</i>
8	<i>MenorEnergía</i> ← <i>Energía</i>
9	<b>Sino</b>
10	$cont=cont+1$
11	<b>Fin Si</b>
12	<b>Si</b> $cont=10$
13	Grabar( <i>Energía, Configuración, Código, Iter, TotalEvaluaciones</i> )
14	Salir
15	<b>Fin Si</b>
16	<i>Energía, Configuración, Código</i> ← <i>BusquedaLocal</i>
17	<b>Si</b> <i>Energía</i> < <i>MenorEnergía</i>
18	<i>MenorEnergía</i> ← <i>Energía</i>
19	<b>Sino</b>
20	$cont=cont+1$
21	<b>Fin Si</b>
22	<b>Si</b> $cont=10$
23	Grabar( <i>Energía, Configuración, Código, Iter, TotalEvaluaciones</i> )
24	Salir
25	<b>Fin Si</b>
26	<b>Fin Para</b>
27	<b>Fin Para</b>

Figura 3.3 Algoritmo GRASP del enfoque de solución propuesto.

### 3.2.1 Construcción miope aleatorizada

La construcción miope aleatorizada consiste en generar nueve configuraciones iniciales diferentes [Romero 1999b] con base en las coordenadas de la rejilla icosaedra. Una vez obtenidas las configuraciones se consigue la menor configuración y la lista restringida de candidatos.

Dada la rejilla icosaedra  $IC_{10}$ , entonces una rebana esférica de  $n$  átomos centrada en el origen  $O$  se denota como:

$$Q_n = S(O, n) \quad (3.1)$$

Las primeras siete configuraciones iniciales serán denotadas por:

$$\hat{Q}_n, \hat{Q}_{n-i}, \dots, \hat{Q}_{n+i}, \dots \quad \text{donde } i = 1 \dots 3 \quad (3.2)$$

**Primera configuración.** Se genera la rebana esférica  $Q_n = S(O, n)$  y se optimiza localmente para generar la primera configuración  $\hat{Q}_n$ .

**Segunda configuración.** Se genera la rebana esférica  $Q_{n-1} = S(O, n-1)$ , se le agrega un átomo aleatorio que no se haya utilizado de la capa más externa y se optimiza localmente para generar la segunda configuración  $\hat{Q}_{n-1}$ .

**Tercera configuración.** Se genera la rebana esférica  $Q_{n-2} = S(O, n-2)$ , se adicionan aleatoriamente dos átomos que no se han utilizado de la capa más externa y se optimiza localmente para generar la tercera configuración  $\hat{Q}_{n-2}$ .



**Cuarta configuración.** Se genera la rebanada esférica  $Q_{n-3} = S(O, n-3)$ , se añaden aleatoriamente tres átomos que no se han utilizado de la capa más externa y se optimiza localmente para generar la cuarta configuración  $\hat{Q}_{n-3}$ .

**Quinta configuración.** Se genera la rebanada esférica  $Q_{n+1} = S(O, n+1)$ , se calcula el número de vecinos para cada uno de los átomos que forman la configuración y el número máximo de vecinos. Se elimina aleatoriamente un átomo que tenga menos vecinos que el número máximo de vecinos y la configuración resultante se optimiza localmente para generar la quinta configuración  $\hat{Q}_{n+1}$ .

Un átomo  $i$  es el vecino más cercano del átomo  $j$  siempre que  $r^* - \varepsilon \leq r_{ij} \leq r^* + \varepsilon$ , donde  $\varepsilon = 0.2$ ,  $r^* = \sqrt[6]{2}$  y  $r_{ij}$  es la distancia Euclidiana entre el átomo  $i$  y el átomo  $j$ .

**Sexta configuración.** Se genera la rebanada esférica  $Q_{n+2} = S(O, n+2)$ , se determina el número de vecinos para cada uno de los átomos que forman la configuración y el número máximo de vecinos. Después se quitan de la configuración aleatoriamente dos átomos que tengan menos vecinos que el número máximo de vecinos y se optimiza localmente para generar la sexta configuración  $\hat{Q}_{n+2}$ .

**Séptima configuración.** Se genera la rebana esférica  $Q_{n+3} = S(O, n+3)$ , se calcula el número de vecinos para cada uno de los integrantes que forman la configuración y el número máximo de vecinos. Posteriormente se suprimen aleatoriamente tres átomos que tengan menos vecinos que el número máximo de vecinos y por último se optimiza la configuración para generar la séptima configuración  $\hat{Q}_{n+3}$ .

Dada la rejilla icosaedroal  $IC_{10}$ , entonces una rebana esférica de  $n$  átomos centrada en el punto  $p$  se denota como:

$$Q_n = S(p, n) \quad (3.3)$$

Las dos últimas configuraciones iniciales están dadas por

$$\hat{Q}_i \quad \text{donde } i = 8, 9 \quad (3.4)$$

**Octava y novena configuración.** Estas dos configuraciones se obtienen por medio del mismo procedimiento, el cual consiste en: generar un número aleatorio  $x$  entre cero y uno, si  $x < 1/3$  entonces el punto  $p$  se elige aleatoriamente en un rayo, de lo contrario  $p$  es elegido en forma aleatoria en una esfera centrada en el origen  $O$ ; si  $1/3 \leq x < 2/3$  entonces el radio de la esfera es uno, sino el radio es igual a ocho. Después que se tiene el punto  $p$  se genera la rebanada esférica  $Q_i = S(p, n)$ , finalmente se optimiza localmente para generar configuración  $\hat{Q}_i$   $i=8,9$ .

Un rayo es un segmento de línea recta que conecta el origen  $O$  a uno de los doce vértices del icosaedro más externo. Para determinar el punto  $p$  en el rayo, primero se elige un vértice aleatorio y la longitud del rayo, la cual puede ser entre uno y ocho; después se multiplican las coordenadas del vértice por la longitud y se obtiene el punto  $p$ .

Para elegir el punto  $p$  en una esfera centrada en el origen, una vez que se tiene el radio se obtienen las coordenadas esféricas. El valor de rho  $\rho$  es elegido en forma aleatoria entre  $0^\circ$  y  $180^\circ$  y theta  $\theta$  aleatoriamente entre  $0^\circ$  y  $360^\circ$ .

La Figura 3.4 contiene la descripción del algoritmo de construcción miope aleatorizada.

Algoritmo CONSTRUCCIÓN MIOPE ALEATORIZADA	
<u>Objetivo:</u> Generar nueve configuraciones iniciales y la lista restringida de candidatos.	
<u>Entrada:</u> Rejilla icosaedral.	
<u>Funciones:</u>	
<i>CalcularEnergiaLJ(Configuración, n)</i> . Obtiene una configuración optimizada de $n$ átomos y su energía.	
<i>AgregarAtomo(Configuración(<math>n-i</math>), <math>i</math>)</i> . Adiciona $i$ átomos a la configuración.	
<i>NumeroVecinos(Configuración)</i> . Obtiene el número de vecinos de la configuración.	
<i>EliminarAtomo(Configuración(<math>n+i</math>), <math>i</math>)</i> . Suprime $i$ átomos a la configuración.	
<u>Salida:</u> Configuraciones iniciales, <i>LRC</i> .	
1	Tomar rebanada esférica de tamaño $n$ para <i>Config 1</i>
2	<i>Energía, ConfOpt, Eval</i> ← <i>CalcularEnergiaLJ(Config 1, n)</i>
3	<b>Para</b> ( $i=1, \dots, 3$ ) <b>Hacer</b>
4	Tomar rebanada esférica de tamaño $n-i$ para <i>Config <math>i+1</math></i>
5	<i>ConfAux</i> ← <i>AgregarAtomo(Config <math>i+1</math>, <math>i</math>)</i>
6	<i>Energía, ConfOpt, Eval</i> ← <i>CalcularEnergiaLJ(ConfAux, n)</i>
7	<b>Fin Para</b>
8	<b>Para</b> ( $i=1, \dots, 3$ ) <b>Hacer</b>
9	Tomar rebanada esférica de tamaño $n+i$ para <i>Config <math>i+4</math></i>
10	<i>ConfigVecinos</i> ← <i>NumeroVecinos(Config <math>i+4</math>)</i>
11	<i>ConfAux</i> ← <i>EliminarAtomo(ConfigVecinos, <math>i</math>)</i>
12	<i>Energía, ConfOpt, Eval</i> ← <i>CalcularEnergiaLJ(ConfAux, n)</i>
13	<b>Fin Para</b>
14	Tomar rebanada esférica de tamaño $n$ para <i>Config 8</i>
15	<i>Energía, ConfOpt, Eval</i> ← <i>CalcularEnergiaLJ(Config 8, n)</i>
16	Tomar rebanada esférica de tamaño $n$ para <i>Config 9</i>
17	<i>Energía, ConfOpt, Eval</i> ← <i>CalcularEnergiaLJ(Config 9, n)</i>
18	Construir la <i>LRC</i> de entre las 9 configuraciones

Figura 3.4 Algoritmo de la construcción miope aleatorizada del problema de LJ.

### 3.2.1.1 Método numérico de búsqueda local

El método numérico que se utilizó es el método de Regiones de Confianza, el cual se ejecuta cada vez que se optimiza una configuración de átomos.

El enfoque de Regiones de Confianza para optimización, considera el problema de minimización no restringida  $\min f(x)$ ; donde la función recibe como argumento un vector y regresa valores escalares. Suponer que se tiene un punto  $x$  en un espacio  $n$  y se desea mejorar, por ejemplo: moviendo el punto a un menor valor de la función. La idea básica es aproximar  $f$  con una función simple  $q$ , la cual refleja el comportamiento de la función  $f$  en una vecindad  $V$  alrededor del punto  $x$ . Esta vecindad es la Región de Confianza. Un paso de prueba  $s$  se calcula para minimizar sobre  $V$ , este es el subproblema de la Región de Confianza:

$$\min_s \{q(s) \quad s \in V\} \quad (3.5)$$

El punto  $x$  se actualiza por  $x+s$  si  $f(x+s) < f(x)$ , de otra forma el punto  $x$  permanece igual y  $V$ , la Región de Confianza se reduce y el cálculo del paso de prueba se repite [Matlab 2004].

### 3.2.1.2 Lista restringida de candidatos

Una lista restringida de candidatos (LRC) contiene un conjunto de elementos candidatos con los mejores valores de la construcción miope aleatorizada. Las energías de las nueve configuraciones iniciales se toman como base para formar la LRC. Se va agregando a la LRC todo aquel elemento que su energía de LJ cumpla con la siguiente condición:

$$E \leq E_{\min} + \alpha(E_{\max} - E_{\min}) \quad (3.6)$$

Donde  $E_{\min}$  es la mínima energía de entre las nueve configuraciones iniciales,  $E_{\max}$  es la máxima energía de entre las nueve configuraciones y  $\alpha$  es un valor aleatorio entre 0.5 y 0.8.

### 3.2.2 Búsqueda local

En el proceso de búsqueda local se forman configuraciones vecinas a partir de las configuraciones iniciales y la LRC. Lo primero que se hace es elegir de la LRC el vecino 1, se van formando parejas del vecino 1 con cada uno de los integrantes de la LRC a excepción de

que el vecino 1 no puede tener como pareja a él mismo. A cada una de las parejas se les aplica el operador de cruzamiento [Romero 1999a], el cual define una estructura de vecindades.

### **Operador de cruzamiento**

A partir de una pareja de configuraciones iniciales y un valor aleatorio  $x$ , se pueden generar por cruzamiento dos configuraciones vecinas en alguna de las siguientes formas:

**Caso 1 Cortando con un plano.** Si  $0.0 \leq x < 0.3$  entonces uno de los tres planos  $xy, xz, yz$  se elige al azar. La pareja es cortada con el plano elegido y de las cuatro mitades se forman dos configuraciones vecinas.

**Caso 2 Cortando con una esfera.** Si  $0.3 \leq x < 0.6$  entonces el radio y el centro de la esfera se seleccionan aleatoriamente. La pareja es cortada con la esfera y dos configuraciones son ensambladas de las cuatro piezas.

**Caso 3 Cortando con una esfera centrada en el origen.** Si  $0.6 \leq x < 0.9$  entonces el radio de la esfera se elige en forma aleatoria y el centro es el origen. La pareja se corta con la esfera y de las cuatro porciones se forman dos configuraciones vecinas.

**Caso 4 Por rotación.** Si  $0.9 \leq x < 1.0$  entonces surgen dos configuraciones por rotación de cada uno de los integrantes de la pareja. El eje y el ángulo de rotación se eligen al azar.

La Figura 3.5 muestra el algoritmo del proceso de búsqueda local. De la línea 4 a la línea 16 comprende los cuatro casos para obtener configuraciones vecinas, sus algoritmos respectivos se van a explicar más adelante. En la línea 17 se optimiza la configuración vecina 1. Después la configuración optimizada se manda como parámetro a la función maquillaje (línea 18), en donde se afina la configuración. Lo mismo se hace para la configuración vecina 2 (líneas 19 y 20). De estas dos configuraciones se elige la menor y se guarda. Al finalizar el proceso de búsqueda en las parejas, se obtiene la energía menor de entre todas las configuraciones de todas las parejas y se regresa como la mejor configuración.

Algoritmo BÚSQUEDA LOCAL	
<u>Objetivo</u> : Obtener configuraciones vecinas.	
<u>Entrada</u> : Configuraciones iniciales y LRC.	
<u>Funciones</u> :	
<i>CalcularEnergiaLJ(Configuración, n)</i> . Obtiene una configuración optimizada de $n$ átomos y su energía.	
<i>Maquillaje(ConfiguraciónVecina)</i> . Refina la configuración de un vecino para obtener una mejor configuración.	
<u>Salida</u> : Mejor configuración.	
1	Elegir el <i>Vecino 1</i> de la <i>LRC</i>
2	Formar parejas del <i>Vecino 1</i> con los elementos de la <i>LRC</i>
3	<b>Para</b> ( $i=1, \dots, NumParejas$ ) <b>Hacer</b>
4	$x \leftarrow \text{random}[0, 1]$
5	<b>Si</b> $x \geq 0.0$ y $x < 0.3$
6	CortandoPlano()
7	<b>Fin Si</b>
8	<b>Si</b> $x \geq 0.3$ y $x < 0.6$
9	CortandoEsfera()
10	<b>Fin Si</b>
11	<b>Si</b> $x \geq 0.6$ y $x < 0.9$
12	CortandoEsferaCentradaOrigen()
13	<b>Fin Si</b>
14	<b>Si</b> $x \geq 0.9$ y $x < 1.0$
15	RotaciónVecinos()
16	<b>Fin Si</b>
17	Energía, ConfOpt, Eval $\leftarrow$ CalcularEnergiaLJ(ConfiguraciónVecina1, $n$ )
18	ConfiguraciónVecina1 $\leftarrow$ Maquillaje(ConfOpt)
19	Energía, ConfOpt, Eval $\leftarrow$ CalcularEnergiaLJ(ConfiguraciónVecina2, $n$ )
20	ConfiguraciónVecina2 $\leftarrow$ Maquillaje(ConfOpt)
21	Obtener la energía menor de entre las dos configuraciones vecinas
22	Guardar la energía menor
23	<b>Fin Para</b>
24	Obtener la energía menor de entre todas las configuraciones de todas las parejas
25	Regresar Energía, ConfiguraciónVecina

Figura 3.5 Algoritmo de la búsqueda local del problema de LJ.

El caso cortando con un plano se muestra en la Figura 3.6. Se elige al azar uno de los tres planos (línea 1), el plano 1 equivale al plano  $xy$ , el plano 2 al plano  $xz$  y el plano 3 al plano

yz. En el plano 1, los elementos de la configuración del vecino 1 y del vecino 2 que tengan en su coordenada z valores positivos y negativos se guardan en sus respectivas variables (línea 3 a la línea 6) de manera similar es para los otros dos planos, sólo que cambia la coordenada. Después se manda a llamar a la función cruzamiento, la cual mezcla los valores de las variables para obtener la configuración vecina 1 y configuración vecina 2.

Algoritmo CORTANDO CON UN PLANO	
<u>Objetivo:</u> Formar dos configuraciones vecinas.	
<u>Entrada:</u> Pareja de configuraciones.	
<u>Funciones:</u>	
<i>Cruzamiento</i> ( <i>Vecino1Pos</i> , <i>Vecino1Neg</i> , <i>Vecino2Pos</i> , <i>Vecino2Neg</i> ). Mezcla las cuatro secciones que recibe como parámetros para formar una configuración de tamaño <i>n</i> .	
<u>Salida:</u> Configuración vecina 1 y configuración vecina 2	
1	plano←random[1, 3]
2	<b>Si</b> plano = 1
3	<i>Vecino1Pos</i> ←Valores positivos en z del vecino 1
4	<i>Vecino1Neg</i> ←Valores negativos en z del vecino 1
5	<i>Vecino2Pos</i> ←Valores positivos en z del vecino 2
6	<i>Vecino2Neg</i> ←Valores negativos en z del vecino 2
7	<b>Fin Si</b>
8	<b>Si</b> plano = 2
9	<i>Vecino1Pos</i> ←Valores positivos en y del vecino 1
10	<i>Vecino1Neg</i> ←Valores negativos en y del vecino 1
11	<i>Vecino2Pos</i> ←Valores positivos en y del vecino 2
12	<i>Vecino2Neg</i> ←Valores negativos en y del vecino 2
13	<b>Fin Si</b>
14	<b>Si</b> plano = 3
15	<i>Vecino1Pos</i> ←Valores positivos en x del vecino 1
16	<i>Vecino1Neg</i> ←Valores negativos en x del vecino 1
17	<i>Vecino2Pos</i> ←Valores positivos en x del vecino 2
18	<i>Vecino2Neg</i> ←Valores negativos en x del vecino 2
19	<b>Fin Si</b>
20	<i>ConfVecina1</i> ← <i>Cruzamiento</i> ( <i>Vecino1Pos</i> , <i>Vecino2Neg</i> , <i>Vecino2Pos</i> , <i>Vecino1Neg</i> )
21	<i>ConfVecina2</i> ← <i>Cruzamiento</i> ( <i>Vecino2Pos</i> , <i>Vecino1Neg</i> , <i>Vecino1Pos</i> , <i>Vecino2Neg</i> )

Figura 3.6 Algoritmo del caso cortando con un plano.

La Figura 3.7 muestra el algoritmo del caso cortando con una esfera. El radio y el centro de la esfera se eligen aleatoriamente (línea 1 y 2), se calcula la distancia del centro de la esfera al primer vecino de la pareja de configuraciones (línea 3). Se separan los valores de la distancia que es menor o igual al radio y de la distancia que es mayor al radio (línea 4 y 5). Estos mismos cálculos en cuanto a la distancia se realizan para la configuración del vecino 2 (líneas 6, 7 y 8). Después de obtener las cuatro secciones de las configuraciones, se manda a llamar a la función de cruzamiento, la cual las ensambla para obtener la configuración vecina 1 y 2.

Algoritmo CORTANDO CON UNA ESFERA	
<u>Objetivo:</u> Formar dos configuraciones vecinas.	
<u>Entrada:</u> Pareja de configuraciones.	
<u>Funciones:</u>	
<i>Cruzamiento</i> ( <i>Vecino1Pos</i> , <i>Vecino1Neg</i> , <i>Vecino2Pos</i> , <i>Vecino2Neg</i> ). Mezcla las cuatro secciones que recibe como parámetros para formar una configuración de tamaño <i>n</i> .	
<u>Salida:</u> Configuración vecina 1 y configuración vecina 2	
1	radio←random[1, 10]
2	Elegir centro aleatorio de la esfera
3	distancia1←Distancia del centro al vecino 1
4	<i>Vecino11</i> ←Valores donde distancia1 es ≤ radio
5	<i>Vecino12</i> ←Valores donde distancia1 es > radio
6	distancia2←Distancia del centro al vecino 2
7	<i>Vecino21</i> ←Valores donde distancia2 es ≤ radio
8	<i>Vecino22</i> ←Valores donde distancia2 es > radio
9	<i>ConfVecina1</i> ←Cruzamiento( <i>Vecino11</i> , <i>Vecino22</i> , <i>Vecino21</i> , <i>Vecino12</i> )
10	<i>ConfVecina2</i> ←Cruzamiento( <i>Vecino21</i> , <i>Vecino12</i> , <i>Vecino11</i> , <i>Vecino22</i> )

**Figura 3.7** Algoritmo del caso cortando con una esfera.

El caso cortando con una esfera centrada en el origen (Figura 3.8) es de manera análoga al caso anterior cortando con una esfera, ahora el centro de la esfera es el origen. En la línea 6 y 10 una vez que se obtienen los valores donde las distancias respectivas al vecino 1 y 2 son menor que el radio, los valores se rotan por medio de la función RotarMatriz. Como último paso en las líneas 12 y 13 las porciones se mezclan en la función Cruzamiento.



Algoritmo CORTANDO CON UNA ESFERA CENTRADA EN EL ORIGEN
<p><u>Objetivo</u>: Formar dos configuraciones vecinas.</p> <p><u>Entrada</u>: Pareja de configuraciones.</p> <p><u>Funciones</u>:</p> <p><i>Cruzamiento</i>(<i>Vecino1Pos</i>, <i>Vecino1Neg</i>, <i>Vecino2Pos</i>, <i>Vecino2Neg</i>). Mezcla las cuatro secciones que recibe como parámetros para formar una configuración de tamaño <math>n</math>.</p> <p><i>RotarMatriz</i>(<i>Configuración</i>). Dado un eje y ángulo de rotación aleatorios, rota una configuración.</p> <p><u>Salida</u>: Configuración vecina 1 y configuración vecina 2</p> <pre> 1  radio←random[1, 10] 2  p←0 3  Elegir centro aleatorio de la esfera 4  distancia1←Distancia de p al vecino 1 5  Vecino11←Valores donde distancia1 es ≤ radio 6  Vecino11R←RotarMatriz(Vecino11) 7  Vecino12←Valores donde distancia1 es &gt; radio 8  distancia2←Distancia del p al vecino 2 9  Vecino21←Valores donde distancia2 es ≤ radio 10 Vecino21R←RotarMatriz(Vecino21) 11 Vecino22←Valores donde distancia2 es &gt; radio 12 ConfVecina1←Cruzamiento(Vecino11R, Vecino22, Vecino21R, Vecino12) 13 ConfVecina2←Cruzamiento(Vecino21R, Vecino12, Vecino11R, Vecino22) </pre>

**Figura 3.8** Algoritmo del caso cortando con una esfera centrada en el origen.

La rotación de vecinos, que es el caso 4 del operador de cruzamiento tiene como objetivo formar dos configuraciones vecinas. Recibe como entrada la pareja de configuraciones, después dado un eje y ángulo de rotación aleatorios las configuraciones son rotadas, dando como resultado las configuraciones vecinas 1 y 2. Ver la Figura 3.9.

<p>Algoritmo ROTACIÓN DE VECINOS</p> <p><u>Objetivo</u>: Formar dos configuraciones vecinas.</p> <p><u>Entrada</u>: Pareja de configuraciones.</p> <p><u>Funciones</u>:</p> <p><i>RotarMatriz(Configuración)</i>. Dado un eje y ángulo de rotación aleatorios, rota una configuración.</p> <p><u>Salida</u>: Configuración vecina 1 y configuración vecina 2</p> <p>1 <i>ConfVecina1</i> ← RotarMatriz(<i>Vecino1</i>)</p> <p>2 <i>ConfVecina2</i> ← RotarMatriz(<i>Vecino2</i>)</p>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Figura 3.9** Algoritmo del caso de rotación de vecinos.

### 3.2.2.1 Método numérico de búsqueda local

El método numérico de búsqueda local que se empleó para optimizar las configuraciones es el método de Regiones de Confianza, el cual fue descrito en la sección 3.2.1.1.

### 3.2.2.2 Cruzamiento

La función de cruzamiento recibe como parámetros cuatro porciones de una configuración de átomos (ver Figura 3.6). Para estandarizar la función, los valores que recibe en el primer parámetro se toman como valores positivos del vecino 1, los valores del segundo parámetro como valores negativos del vecino 1, los del tercer parámetro como valores positivos del vecino 2 y los valores del cuarto parámetro como valores negativos del vecino 2. La Figura 3.10 muestra la secuencia que se utilizó para mezclar los valores de los cuatro parámetros y de esta manera formar una nueva configuración de átomos.

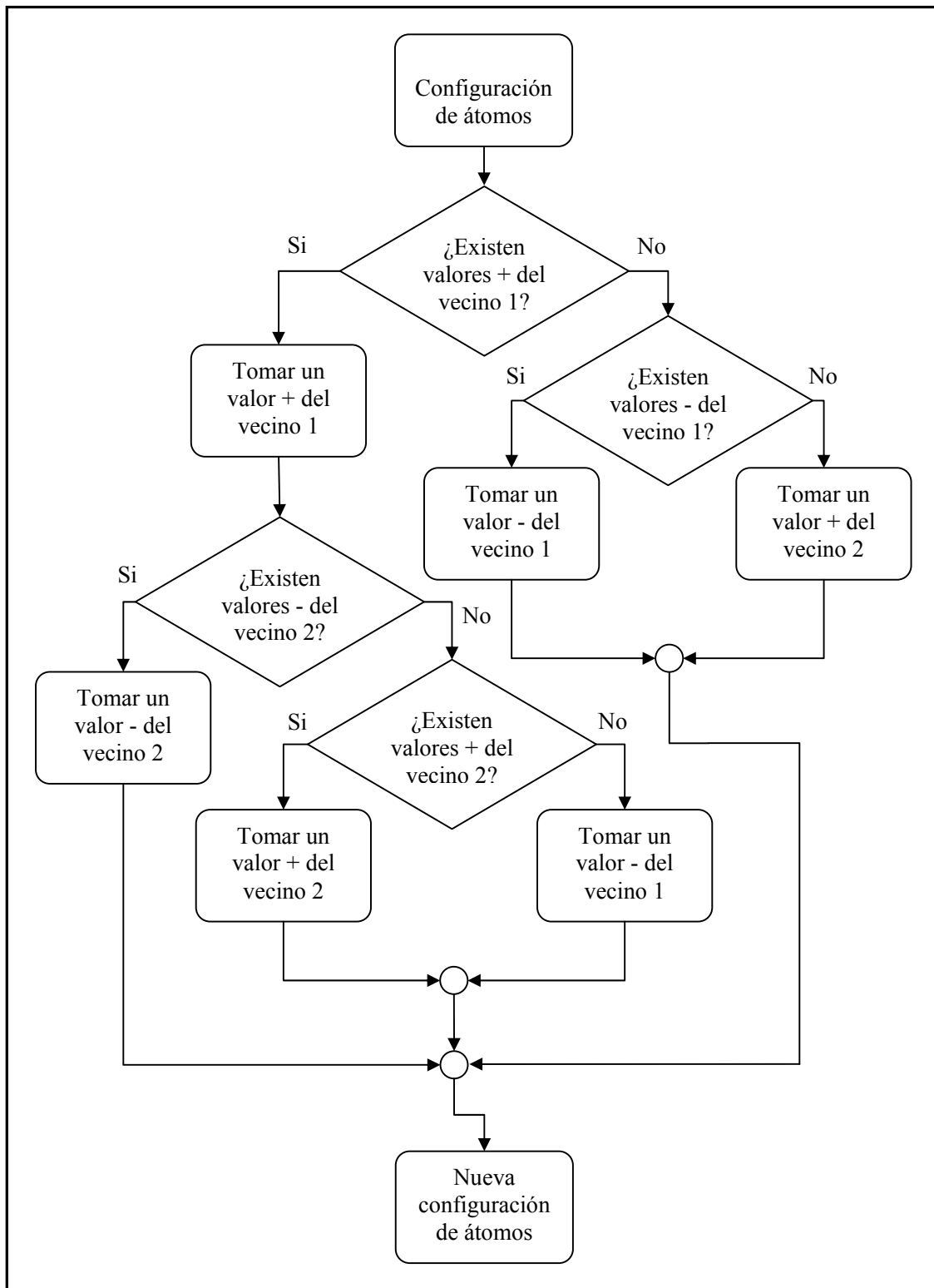


Figura 3.10 Secuencia de la función Cruzamiento.

### 3.2.2.3 Rotación

La forma de rotar las configuraciones, es pensar que alguno de los tres ejes está fijo y que la configuración de átomos se mueve en el espacio en sentido contrario a las manecillas del reloj [Oviedo 2005].

La rotación tridimensional alrededor del eje  $x$  es:

$$R_x = \begin{vmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \text{sen} \theta \\ 0 & -\text{sen} \theta & \cos \theta \end{vmatrix}$$

Para hacer una rotación alrededor del eje  $y$ , del tal forma que el eje  $z$  se convierte en el  $x$  se realiza la siguiente matriz en coordenadas homogéneas:

$$R_y = \begin{vmatrix} \cos \theta & 0 & -\text{sen} \theta \\ 0 & 1 & 0 \\ \text{sen} \theta & 0 & \cos \theta \end{vmatrix}$$

La rotación alrededor del eje  $z$  se realiza con la matriz de transformación:

$$R_z = \begin{vmatrix} \cos \theta & \text{sen} \theta & 0 \\ -\text{sen} \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

### 3.2.2.4 Maquillaje

El maquillaje es un paso que se realiza para cada uno de los vecinos en la búsqueda local, consiste en las operaciones de relajación y eliminación de bultos.

Relajación. En la relajación la configuración  $C_k$  es relajada a su mínimo local más cercano

$C_k^*$ .

Eliminación de bultos. En esta operación el primer paso es obtener el número de vecinos  $N$  (ver sección 3.2.1) de la configuración a la que se le va aplicar el maquillaje. Hay que separar los átomos fijos de los átomos bultos. Un átomo  $i$  es un bulto de  $C_k^*$  si  $N(i) \leq 6$ , es decir si el número de vecinos es menor igual que seis; de lo contrario es un átomo fijo. El número de bultos en  $C_k^*$  es  $w$ . Si  $w > 0$  entonces se ejecutan los siguientes pasos:

- Remover los  $w$  bultos de  $C_k^*$ . De los  $n$  átomos de la configuración, se quitan  $w$  átomos bultos ( $n - w$ ); la nueva configuración es  $\bar{C}_k$ .
- Iniciar con  $\bar{C}_k$ . Para  $i = 1, \dots, w$  agregar un átomo de la capa más externa de la presente configuración, previamente el átomo es relajado. Cuando los  $w$  átomos han sido agregados a  $\bar{C}_k$ , la configuración completa se relaja para conseguir la configuración  $\hat{C}_k$ .
- Si  $E(\hat{C}_k) < E(C_k^*)$  entonces  $\hat{C}_k$  reemplaza  $C_k^*$ .

### 3.3 Método para reducir la varianza de los resultados en experimentos con algoritmos metaheurísticos

Este es un método de propósito general que se aplica a cualquier tipo de algoritmo metaheurístico, sin importar cual sea el problema que se resuelva. En realidad el enfoque es aplicable sin modificaciones mayores a cualquier tipo de algoritmo aleatorizado, sin embargo en este trabajo sólo se muestran los resultados de experimentos realizados con un algoritmo metaheurístico. En las siguientes líneas se describen los pasos del método.

**Paso 1. Identificar la *decisión base*.** Analizar la secuencia de decisiones aleatorias que se produce en cada ejecución del algoritmo. Determinar las decisiones que no necesariamente se producen en cada corrida del algoritmo y aquellas que se producen consistentemente en cada

corrida. Analizar las que se producen en cada corrida del algoritmo y elegir la primera que se produce para definirla cómo la *decisión base*.

**Paso 2. Generar el aleatorio asociado a la *decisión base*.** Sean  $a$  la variable global en la que se almacena el número aleatorio vinculado a la *decisión base* y *corrida* la variable global que contiene el número de la corrida del algoritmo en el experimento. Ahora en lugar de simplemente generar el número aleatorio, se debe realizar el siguiente proceso: if (*corrida* es impar)  $a = \text{random}()$  else  $a = 1 - a$ . Una vez realizado el proceso anterior,  $a$  es utilizada para realizar la *decisión base*. Con esta modificación, los números aleatorios asociado a la *decisión base* de cada una de las corridas impares y pares estarán correlacionados negativamente.

**Paso 3. Determinar las salidas de cada corrida.** Ejecutar el experimento y determinar los valores de cada una de las salidas  $t_i$  de cada corrida  $i$  del algoritmo ( $i = 1, 2, \dots, \text{CORRIDAS}$ ).

**Paso 4. Calcular la media y la varianza.** Al término del experimento se determinan los valores de las variables aleatorias  $x, y$  y  $z$ , considerando que:

$$\begin{aligned}x_j(a) &= t_{(2j-1)} \\y_j(1-a) &= t_{2j} \\z_j &= (x_j(a) + y_j(1-a)) / 2 \\j &= 1, 2, \dots, \text{CORRIDAS} / 2\end{aligned}$$

Finalmente se calcula la media y la varianza de  $z$ . Definidas de esta manera  $t_i, x_j, y_j$  y  $z_j$ , son estimadores del valor esperado de la variable aleatoria  $t$ . Por otra parte como  $z = (x + y) / 2$  se tiene que  $\text{Var}(z) = \text{Var}((x + y) / 2) = \frac{1}{4}[\text{Var}(x) + \text{Var}(y) + 2\text{Cov}(x, y)]$ , y dado que  $x$  e  $y$  fueron generadas a partir de entradas correlacionadas negativamente  $\text{Cov}(x, y) < 0$ . Esta condición produce una disminución de la varianza del estimador  $z_j$  con respecto a la del

estimador  $z'_j$  que se obtiene de manera similar a partir de  $t_i$  cuando este se genera con entradas no correlacionadas negativamente.

A continuación se muestra un ejemplo del método de reducción de la varianza aplicado al metaheurístico desarrollado en esta tesis.

**Paso 1. Identificar la *decisión base*.** La *decisión base* que se identificó en el GRASP se encuentra en la función Agregar Átomo, ya que aquí siempre en todas las *corridas* del algoritmo se genera un número aleatorio y es la primera vez que se genera. En la Figura 3.11 se muestra el algoritmo de la función Agregar Átomo sin aplicar reducción de la varianza.

Algoritmo AGREGAR ÁTOMO	
<u>Objetivo:</u> Añadir $i$ átomos de la capa externa a la configuración.	
<u>Entrada:</u> Configuración y número de átomos a adicionar.	
<u>Salida:</u> Configuración.	
1	<b>Para</b> ( $i=1, \dots, NumÁtomos$ ) <b>Hacer</b>
2	$átomo \leftarrow \text{round}(1+(812-1)*\text{random}())$
3	$Configuración \leftarrow Configuración + átomo$
4	<b>Fin Para</b>

**Figura 3.11** Algoritmo de la función Agregar Átomo sin emplear reducción de la varianza.

**Paso 2. Generar el aleatorio asociado a la *decisión base*.** En la Figura 3.12 se presenta el algoritmo de la función Agregar Átomo utilizando la reducción de la varianza. Si la *corrida* es impar, en la línea 4 se genera el número aleatorio vinculado a la *decisión base* de la línea 5. Si la *corrida* es par se obtiene el complemento del aleatorio anterior (línea 7) y se asocia a la *decisión base* (línea 8).

Algoritmo AGREGAR ÁTOMO	
<u>Objetivo:</u> Añadir $i$ átomos de la capa externa a la configuración.	
<u>Entrada:</u> Configuración y número de átomos a adicionar.	
<u>Salida:</u> Configuración.	
1	<b>Para</b> ( $i=1, \dots, NumÁtomos$ ) <b>Hacer</b>
2	<b>Si</b> iteración = 1
3	<b>Si</b> corrida es impar
4	$a \leftarrow \text{random}()$
5	átomo $\leftarrow \text{round}(1+(812-1)* a)$
6	<b>Sino</b>
7	$a \leftarrow 1- a$
8	átomo $\leftarrow \text{round}(1+(812-1)* a)$
9	<b>Fin Si</b>
10	<b>Sino</b>
11	átomo $\leftarrow \text{round}(1+(812-1)*\text{random}())$
12	<b>Fin Si</b>
13	Configuración $\leftarrow$ Configuración + átomo
14	<b>Fin Para</b>

Figura 3.12 Algoritmo de la función Agregar Átomo con reducción de la varianza.

**Paso 3. Determinar las salidas de cada corrida.** En el experimento se ejecutaron 30 corridas del metaheurístico GRASP. Para cada una de las corridas se guardaba la energía de LJ obtenida, el número de evaluaciones de la función objetivo y el número de iteración donde se encontró la energía. La Tabla 3.1 muestra el número de evaluaciones para cada corrida.

Tabla 3.1 Salidas del experimento una vez aplicado el método de reducción.

Corrida $i$	Evaluaciones $t_i$
1	1,381
2	279
3	3,167
4	242
5	2,212
6	1,625
7	243
8	349
9	383
10	3,123



Corrida i	Evaluaciones t <sub>i</sub>
11	277
12	1,041
13	884
14	1,322
15	238
16	328
17	2,045
18	1,100
19	1,135
20	2,739
21	2,060
22	269
23	2,734
24	2,662
25	3,778
26	1,469
27	2,183
28	1,253
29	233
30	4,739

**Paso 4. Calcular la media y la varianza.** Una vez que se terminó el experimento se calculó la media y la varianza, como se muestra en la Tabla 3.2. La primera columna representa los valores de las salidas impares, la segunda columna los valores de las salidas pares y en la tercera se presenta la media de las salidas pares e impares. Al final de la tercera columna aparece la varianza para este experimento.

**Tabla 3.2** Calculo de la varianza cuando se aplica el método de reducción.

j	$x_j = t_{2j-1}$	$y_j = t_{2j}$	$z_j = (x_j + y_j)/2$
1	1,381	279	830
2	3,167	242	1,705
3	2,212	1,625	1,919
4	243	349	296
5	383	3,123	1,753
6	277	1,041	659
7	884	1,322	1,103

<b>j</b>	$x_j = t_{2j-1}$	$y_j = t_{2j}$	$z_j = (x_j + y_j)/2$
8	238	328	283
9	2,045	1,100	1,573
10	1,135	2,739	1,937
11	2,060	269	1,165
12	2,734	2,662	2,698
13	3,778	1,469	2,624
14	2,183	1,253	1,718
15	233	4,739	2,486
<b>Varianza =</b>			<b>610,527</b>

Para hacer una comparación entre aplicar reducción de la varianza y no emplearla, la Tabla 3.3 muestra las salidas de este mismo experimento, pero sin utilizar reducción de la varianza.

**Tabla 3.3** Salidas y cálculo de la varianza cuando no se aplica el método de reducción.

<b>Corrida i</b>	<b>Evaluaciones <math>t_i</math></b>
1	1,258
2	1,971
3	2,315
4	956
5	5,751
6	239
7	328
8	1,222
9	5,275
10	3,322
11	2,819
12	2,479
13	6,435
14	1,254
15	3,952
16	617
17	230
18	1,451
19	1,995
20	2,728
21	275
22	3,892
23	890

<b>Corrida i</b>	<b>Evaluaciones t<sub>i</sub></b>
24	974
25	353
26	777
27	1,084
28	1,446
29	1,157
30	2,073
<b>Varianza =</b>	<b>2,775,921</b>

De acuerdo a las Tablas 3.3 y 3.2, la varianza se logró reducir de 2,775,921 a 610,527; es decir se redujo un 78.01%.

# Capítulo 4

## **RESULTADOS EXPERIMENTALES**

---

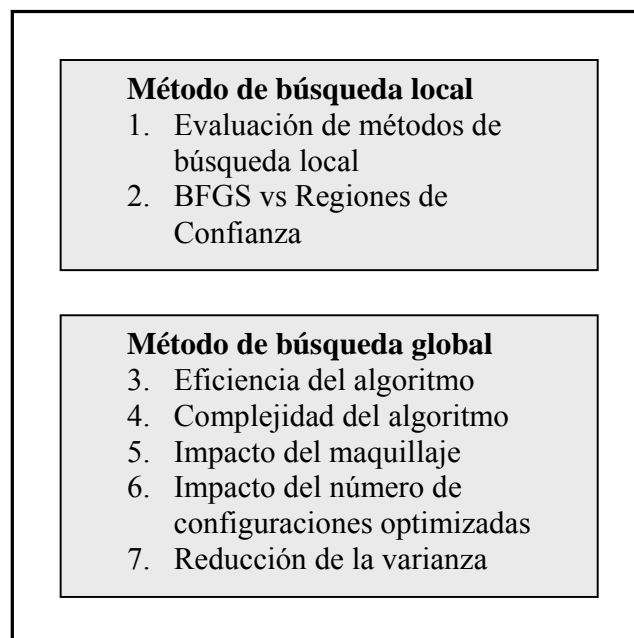
En este apartado se describe la experimentación realizada para probar la eficiencia del método de solución propuesto. Se presenta el plan de pruebas utilizado y se describe la plataforma experimental de software y hardware utilizada en los experimentos. Posteriormente se describen los experimentos realizados, se muestran los resultados obtenidos y se efectúa el análisis de los mismos.

### **4.1 Plan de pruebas**

En esta sección se describe el plan de pruebas que se llevó a cabo para validar la eficiencia del método propuesto. Hay que recordar que el enfoque de solución consiste en diseñar e

implementar un método de búsqueda global que invoca un optimizador local. Los dos primeros experimentos se realizaron para elegir el optimizador local y los cuatro siguientes para evaluar el desempeño del método de búsqueda global (ver Figura 4.1).

La hipótesis fundamental de esta propuesta es que es posible desarrollar un nuevo método de optimización global para la solución del problema de Lennard-Jones con base en el metaheurístico GRASP y el método Regiones de Confianza, el cual es capaz de obtener los mejores resultados reportados. Para mostrar que es posible construir el método, se analizó experimentalmente el desempeño del mismo. Las pruebas realizadas se diseñaron considerando las recomendaciones especificadas para este tipo de experimentos [Barr 2001, McGeoch 2002, Johnson 2002, Moret 2002]. Algunos de los principios aplicados son los siguientes: los experimentos deben estar vinculados con la literatura, se deben poder reproducir y comparar con los realizados en otras investigaciones, deben utilizar implementaciones razonablemente eficientes y casos de prueba estándar e incorporar técnicas de reducción de la varianza.



**Figura 4.1** Experimentos del plan de prueba.

## 4.2 Plataforma experimental

Todos los experimentos fueron realizados en una computadora Dell Opti Plex con procesador Xeon a 3.06 GHZ, memoria RAM de 4 GB y disco duro de 60 GB. El sistema operativo es Microsoft Windows Server 2003 y el software donde se programó el método fue Matlab 7.0.0.19920 (R14).

## 4.3 Experimento 1: Evaluación de métodos de búsqueda local

### 4.3.1 Objetivo

Evaluar el desempeño de los métodos de búsqueda local que se encuentran disponibles en Matlab.

### 4.3.2 Procedimiento

Matlab dispone de métodos de optimización local de gran escala y de mediana escala. Sólo cuenta con un método de gran escala el cual es una implementación del método de optimización numérica de Regiones de Confianza con precondicionamiento [Moré 1983]. El siguiente segmento de código se utilizó para resolver los casos de prueba con este método:

```
options= optimset('LargeScale','on','GradObj','on');  
x0=[1,1];  
[x,fval,exitflag,output] = fminunc(fun,x0,options);
```

Por default la función *fminunc* elige el algoritmo de gran escala si el usuario proporciona el gradiente en *fun* y la opción *GradObj* en *on* usando la función *optimset*. En cada iteración se realiza la solución aproximada de un sistema lineal grande usando el método de gradientes conjugados precondicionados (PCG).

Para la optimización de mediana escala se evaluaron los siguientes métodos de optimización numérica: Quasi-Newton (*bfgs*), búsqueda directa (*dff*) y gradiente descendente (*steepdesc*).

El siguiente segmento de código se utilizó para evaluar estas opciones:

```
options = optimset('LargeScale','off','GradObj','on','HessUpdate','bfgs')
x0=[1,1];
[x,fval,exitflag,output] = fminunc(fun,x0,options);
```

La función *optimset* con la opción *LargeScale* en *off* y *HessUpdate* en *bfgs* utiliza el método Quasi-Newton BFGS con un procedimiento de búsqueda cuadrático mixto y cúbico lineal [Broyden 1970, Fletcher 1970, Goldfarb 1970, Shanno 1970]. Este método usa la fórmula BFGS para actualizar la aproximación de la matriz Hessiana.

Si se elige la opción *dff* en *HessUpdate* (y la opción *LargeScale* en *off*), se aproxima la inversa de la matriz Hessiana en el proceso de aproximación. Para seleccionar el método de gradiente descendente se tiene que establecer *HessUpdate* en *steepdesc* (y la opción *LargeScale* en *off*).

La Tabla 4.1 muestra las funciones de prueba que se utilizaron, las cuales fueron tomadas del conjunto de funciones de prueba estándar denominado More, Garbow and Hillstrom [Shores 2006].

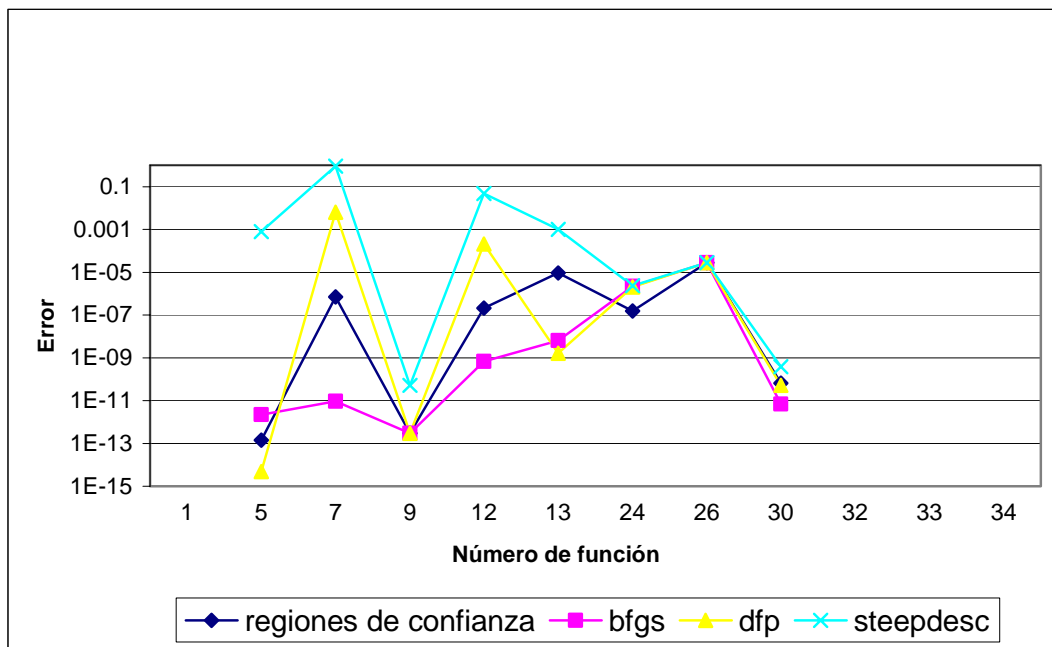
Para cada una de las funciones del caso de prueba se obtuvo la solución óptima con los métodos mencionados, se determinó el valor de la función y el número de veces que se evaluó la función objetivo.

**Tabla 4.1** Funciones de prueba.

Número de función	Nombre de la función	Archivo
1	Rosenbrock	rosen.m
5	Beale	beale.m
7	Helical valley	helix.m
9	Gaussian	gauss.m
12	Box three-dimensional	box.m
13	Powell singular	sing.m
24	Penalty II	pen2.m
26	Trigonometric	trig.m
30	Broyden tridiagonal	trid.m
32	Linear function - full rank	lin.m
33	Linear function - rank 1	lin1.m
34	Linear function - rank 1 with zero columns and rows	lin0.m

### 4.3.3 Resultados

La Figura 4.2 muestra el error que se obtuvo al evaluar las funciones de la Tabla 4.1 con los métodos de búsqueda local. Dado que el error es muy pequeño, la Figura 4.2 se presenta en escala logarítmica para una mejor apreciación.



**Figura 4.2** Error de las funciones evaluadas.



La Figura 4.3 muestra el número de evaluaciones de la función objetivo realizadas con cada uno de los métodos.

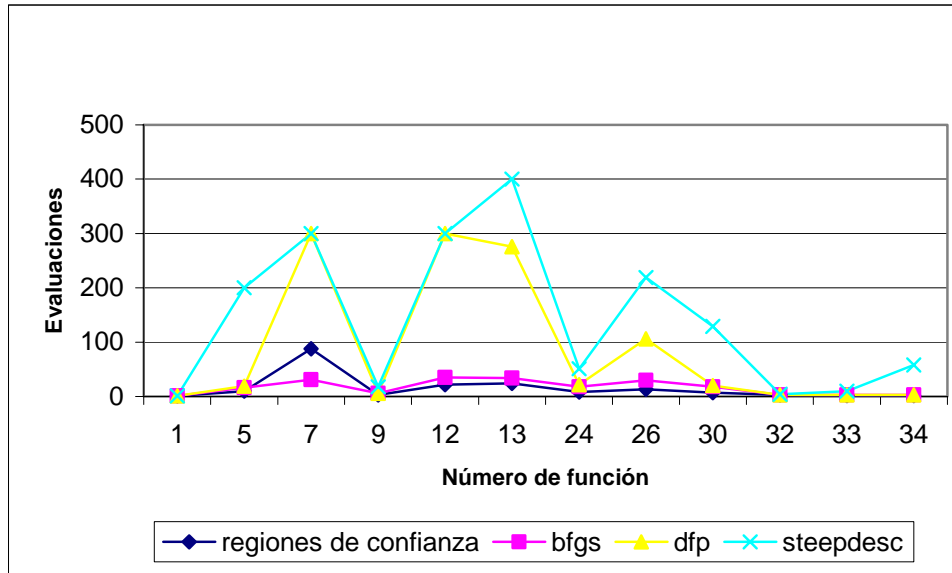


Figura 4.3 Número de evaluaciones de las funciones.

#### 4.3.4 Análisis de resultados

Como se puede apreciar en la Figura 4.2, los métodos de Regiones de Confianza y BFGS son los que menos error tienen, son los que más se acercan al resultado real. En la Figura 4.3 se puede ver que el método BFGS es el que encuentra el valor óptimo de la función objetivo con un número menor de evaluaciones. De acuerdo con estos resultados el método BFGS es el que mejor rendimiento muestra en la solución del caso de prueba y el segundo mejor es el método de Regiones de Confianza. Esto aparentemente es una contradicción, sin embargo existe una explicación del resultado ya que las funciones utilizadas en el experimento son relativamente poco complejas comparadas con la función objetivo de LJ. A diferencia de esta función, las utilizadas en el experimento tienen muy pocas variables y sólo unos cuantos óptimos locales. Para discriminar entre estos métodos se requiere realizar un experimento con base en la función de LJ.

## **4.4 Experimento 2: Discriminación entre los métodos de búsqueda local BFGS y Regiones de Confianza**

### **4.4.1 Objetivo**

Comparar el desempeño de los métodos BFGS y Regiones de Confianza en la solución del problema de LJ.

### **4.4.2 Procedimiento**

En este experimento la única función que se probó es la de LJ para un grupo de 15 átomos. Para realizar el experimento se genera una solución inicial extrayendo una rebanada esférica de la rejilla IC y se evalúa la función objetivo. A dicha configuración inicial se le aplica el método BFGS y se determina el valor de la función objetivo en la configuración optimizada. Finalmente a esta última configuración se le aplica el método de Regiones de Confianza y se calcula de nueva cuenta el valor de la función objetivo en esta nueva configuración. En ambos casos se determina el número de evaluaciones de la función objetivo realizadas por cada uno de los métodos. Este proceso se itera doscientas veces. Si el método de gran escala es mejor que el de mediana escala debe mejorar algunas de las configuraciones generadas por este último. En caso contrario se debe invertir el orden de aplicación de los métodos y probar que el otro es mejor. En el proceso se acumulan todas las evaluaciones realizadas por ambos métodos para comparar su rendimiento global.

### **4.4.3 Resultados**

La Figura 4.4 muestra una parte representativa de los resultados del experimento.

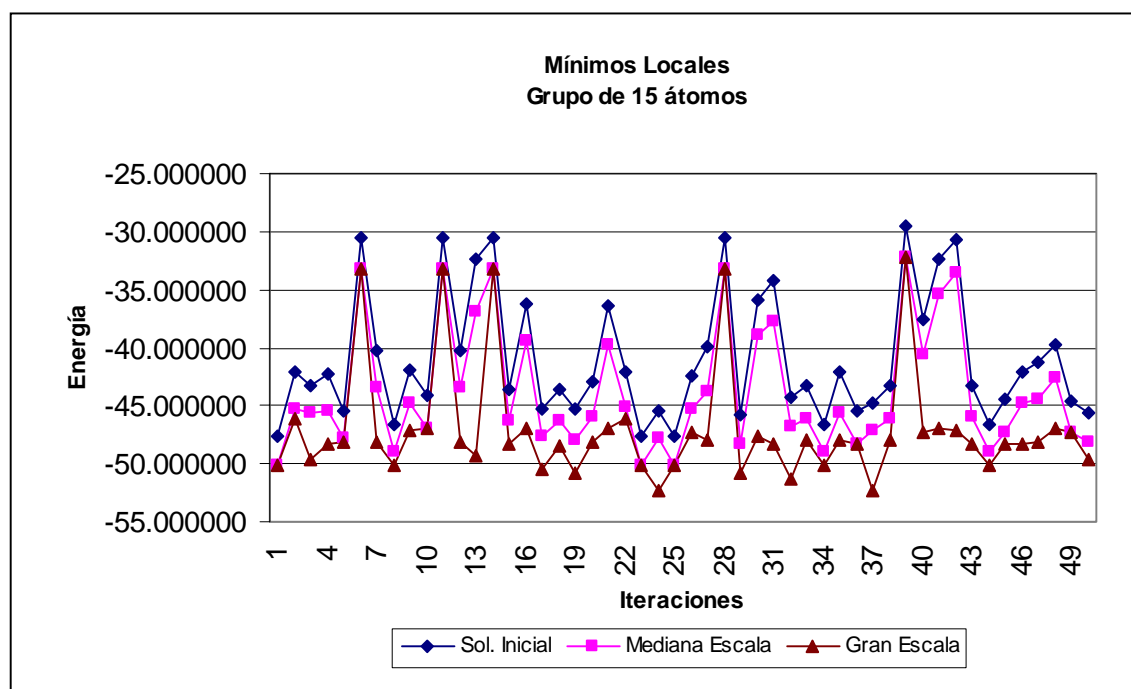


Figura 4.4 Resultados de optimización de una solución inicial.

#### 4.4.4 Análisis de resultados

En la Figura 4.4 se observa que el método de gran escala o Regiones de Confianza, en muchos casos logra mejorar la solución generada por el método BFGS (mediana escala). Por otra parte, en las doscientas iteraciones, el número total de evaluaciones de la función objetivo realizadas por el método BFGS fue de 901,278; mientras que el método de Regiones de Confianza sólo realizó 3,549 evaluaciones de la función objetivo. Estos resultados muestran claramente la superioridad del método de Regiones de Confianza.

## **4.5 Experimento 3: Evaluación de la eficiencia del algoritmo desarrollado**

### **4.5.1 Objetivo**

Evaluar qué tan eficiente es el método propuesto en la búsqueda de la mejor energía reportada para grupos de LJ en el rango 13-25 átomos.

### **4.5.2 Procedimiento**

En el experimento se realizó una búsqueda de las mejores energías reportadas para grupos de LJ en el rango de 13-25. El algoritmo se ejecutó 30 veces por cada grupo, en cada ejecución se realizaron 400 iteraciones GRASP. Aunque no se realizaron todas las iteraciones, ya que en el momento de obtener una configuración se comprobaba si se obtenía la mejor energía reportada y se rompía el ciclo de iteraciones. En este experimento se optimizaron localmente todas las configuraciones de la Construcción miope aleatorizada (descritas en la sección 3.2.1) y todas las configuraciones vecinas de la Búsqueda local (sección 3.2.2).

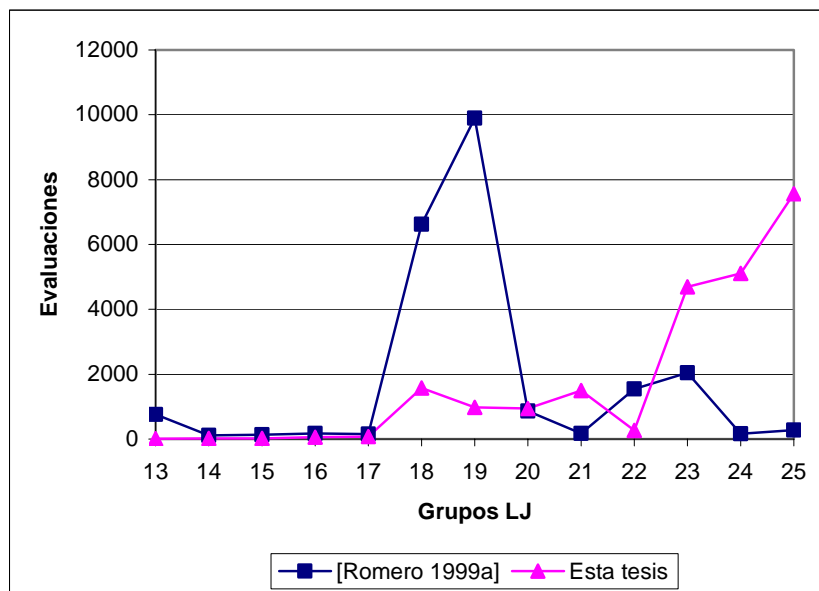
### **4.5.3 Resultados**

La Tabla 4.2 muestra el número de evaluaciones de la función objetivo realizadas para obtener la mejor energía reportada. En la primer columna  $n$  representa el grupo de LJ, en la segunda columna aparece el número de veces que se encontró la energía reportada, en la tercera columna se presenta las evaluaciones de la función objetivo reportadas en [Romero 1999a]; en la cuarta, quinta, sexta y séptima columna se visualiza la mínima, máxima, promedio y la desviación estándar de las evaluaciones respectivamente.

**Tabla 4.2** Evaluaciones de la función objetivo del experimento 3.

n	Éxitos	Evaluaciones de la función objetivo				
		[Romero 1999a]	Mínima	Máxima	Promedio	Desviación Estándar
13	30	754	7	7	7	0
14	30	112	17	17	17	0
15	30	132	17	17	17	0
16	30	177	55	5,316	1,019	1,795
17	30	153	73	21,878	5,408	5,670
18	30	6,620	1,575	171,683	54,535	50,956
19	30	9,901	980	79,838	29,572	23,757
20	30	866	940	79,407	17,294	16,764
21	30	175	1,493	355,797	75,068	77,971
22	30	1,542	270	212,761	58,901	60,572
23	30	2,041	4,697	290,509	103,552	80,199
24	30	163	5,108	180,047	65,753	56,824
25	30	283	7,566	341,639	62,864	68,406
	<b>Totales:</b>	<b>22,919</b>	<b>22,798</b>			

La Figura 4.5 muestra la comparación del número de evaluaciones de la función objetivo que se reporta en [Romero 1999a] contra el valor que se obtiene en esta tesis. En los totales de la columna tres y cuatro de la Tabla 4.2 se aprecia una disminución en las evaluaciones.



**Figura 4.5** Evaluaciones de la función objetivo del experimento 3.

La Tabla 4.3 muestra el número de iteración GRASP donde se encontró la mejor energía reportada. La columna uno muestra el grupo LJ, en la columna dos se muestra el número de generación que reporta [Romero 1999a], en las siguientes cuatro columnas se presenta la mínima, máxima, promedio y desviación estándar del número de iteraciones de esta tesis.

**Tabla 4.3** Número de iteración del experimento 3.

n	Iteración				
	[Romero 1999a]	Mínima	Máxima	Promedio	Desviación Estándar
13	0	0	0	0	0
14	0	0	0	0	0
15	0	0	0	0	0
16	0	0	2	0.3	0.7
17	0	0	10	2.4	2.71
18	2	0	87	26.13	25.29
19	3	0	37	12.73	10.93
20	0	0	38	7.77	8.07
21	0	0	160	33.7	35.36
22	0	0	93	25.07	26.6
23	0	1	121	44.53	34.69
24	0	1	74	27.07	23.88
25	0	3	137	24.27	27.43

La comparación del número de iteraciones entre las que reportada Romero y esta tesis se puede apreciar en la Figura 4.6.

#### 4.5.4 Análisis de resultados

Como se puede apreciar en la columna dos de la Tabla 4.2, el algoritmo desarrollado presenta 100% de eficiencia en la búsqueda de la mejor energía reportada para los grupos LJ de 13-25 átomos. En 8 de los grupos se logra una reducción del número de evaluaciones de la función objetivo con respecto a lo reportado en [Romero 1999a]. Además en este indicador se logra una reducción global de 0.53%.

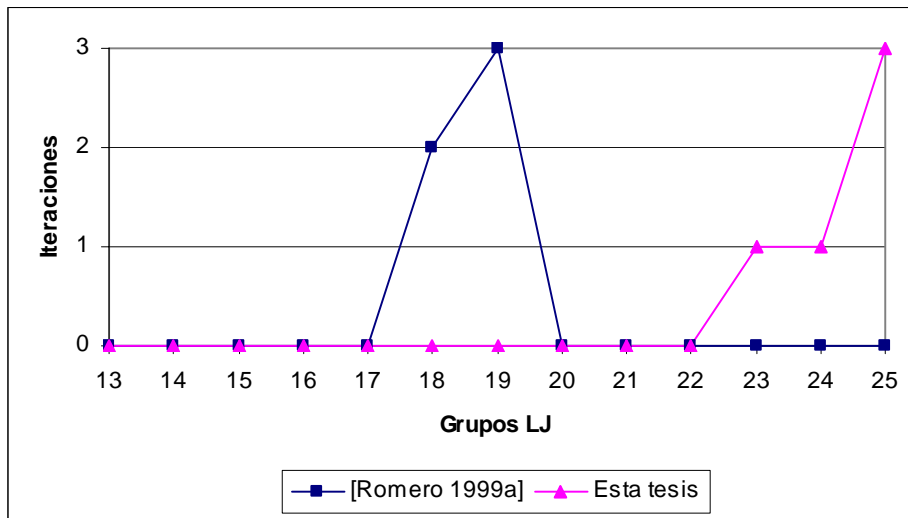


Figura 4.6 Número de iteraciones del experimento 3.

## 4.6 Experimento 4: Complejidad del algoritmo

### 4.6.1 Objetivo

Determinar experimentalmente la complejidad del algoritmo desarrollado en esta tesis.

### 4.6.2 Procedimiento

Barr y Johnson señalan que la efectividad de una metodología se puede probar mediante análisis teórico o con pruebas experimentales [Barr 2001, Johnson 2002]. Señalan también que el análisis teórico resulta muchas veces limitado para analizar el desempeño de los algoritmos en aplicaciones reales. El enfoque del peor caso tiende a producir cotas demasiados pesimistas, y no provee información del desempeño de los algoritmos sobre instancias específicas [McGeoch 1992].

La complejidad teórica del peor de los casos del algoritmo propuesto es  $O(n^5)$ . Esto se debe a que el optimizador local es invocado en el ciclo de búsqueda del algoritmo el cual es de complejidad  $O(n)$ . Por su parte el optimizador local es de complejidad  $O(n^4)$ , dado que integra el ciclo de aproximaciones sucesivas de complejidad  $O(n)$ , con el ciclo de interpolación cúbica de complejidad  $O(n^3)$ , la Figura 4.7 muestra los detalles de este análisis. Sin embargo la cota  $O(n^5)$  es demasiado pesimista, por lo que se procedió a determinar experimentalmente una cota más realista.

Se hizo una búsqueda de las mejores energías reportadas para grupos de LJ en el rango de 13-30. El algoritmo se ejecutó 30 veces por cada grupo, de las 30 ejecuciones se eligió el máximo número de evaluaciones para cada grupo LJ. En este experimento se optimizaron todas las configuraciones de la Construcción miope aleatorizada (descritas en la sección 3.2.1) y todas las configuraciones vecinas de la Búsqueda local (sección 3.2.2).

	Algoritmo REGIONES DE CONFIANZA	Complejidad	
1	// Especificar un punto inicial $x^0$		
2	// Inicializar variables y constantes		
3	<b>Para</b> $k=1$ hasta ..... <i>&lt;condición de paro&gt;</i> <b>Hacer</b>		$O(n)$
4	// Evaluar el vector gradiente $\nabla f(x^k)$		
5	// Calcular la dirección de búsqueda $S_k = -\nabla f(x^k) / \ \nabla f(x^k)\  \quad O(n)$		
6	// Proceso de minimización $q_k(S_k) = f(x^k) + \nabla f(x^k)^T S_k + a S_k^T \nabla^2 f(x^k) S_k + b S_k^T \nabla^3 f(x^k) S_k$ Sujeto a: $\ S_k\  \leq \Delta_k$	$O(n^3)$	
7	// Construir el siguiente punto $x^{k+1} = x^k + q_k S_k$		
8	// Actualizar $\Delta_k$		
9	// ¿Son los valores de las funciones $f(x^k)$ y $f(x^{k+1})$ suficientemente cercanas y la distancia $\ x^{k+1} - x^k\ $ bastante pequeña?		
10	<b>Fin Para</b> $k$		

Figura 4.7 Complejidad del optimizador local Regiones de Confianza.



### 4.6.3 Resultados

Para determinar experimentalmente la complejidad del peor de los casos, se proponen cotas de la forma  $n^k$ , considerando que  $O(n^k) = O(a_k n^k + a_{k-1} n^{k-1} + \dots + a_1 n + a_0)$ . El valor de  $k$  fue ajustado manualmente hasta lograr una cota superior adecuada para la gráfica del número máximo de evaluaciones de la función objetivo. La Figura 4.8 muestra la forma en que se realizó este proceso, la línea punteada corresponde a la gráfica del número de evaluaciones de la función objetivo para cada uno de los grupos y las gráficas continuas a las cotas correspondientes a diferentes valores de  $k$ .

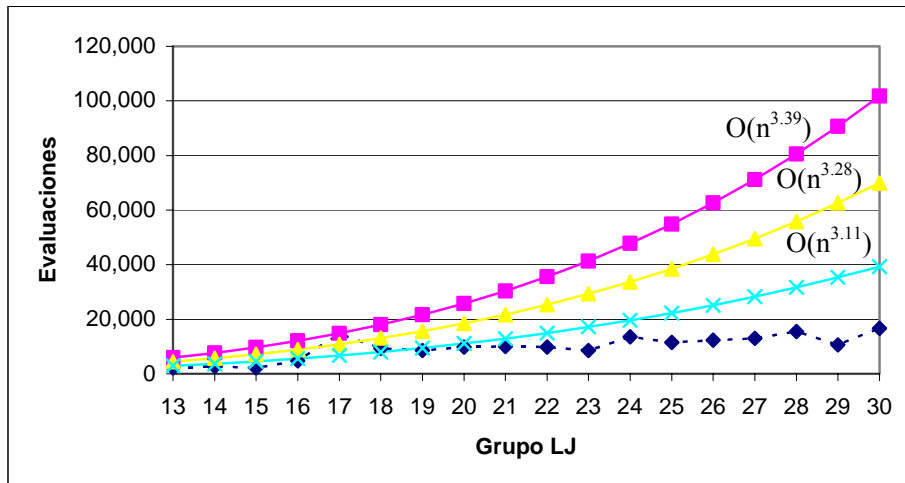


Figura 4.8 Complejidad del algoritmo propuesto.

### 4.6.4 Análisis de resultados

Como se puede observar en la Figura 4.8 la complejidad del algoritmo se ubica entre  $O(n^{3.11})$  y  $O(n^{3.39})$ . Estas cotas constituyen una aproximación más realista que la cota teórica de  $O(n^5)$ .

## **4.7 Experimento 5: Impacto de la función de maquillaje**

### **4.7.1 Objetivo**

Evaluar el desempeño del algoritmo propuesto con la opción maquillaje y sin maquillaje.

### **4.7.2 Procedimiento**

En este experimento, el algoritmo se ejecutó en dos ocasiones para grupos de LJ en el rango de 13-30. En la primera ocasión se activo la opción de maquillaje y en la segunda ocasión se omitió. Tanto como para la primera como para la segunda ocasión se realizó lo siguiente: el algoritmo se ejecutó 30 veces por cada grupo, en cada ejecución se realizaron 400 iteraciones GRASP; cabe mencionar que no se realizaron todas las iteraciones, ya que si la energía no era actualizada en 10 iteraciones consecutivas se daba por terminado el GRASP. En este experimento se optimizaron todas las configuraciones de la Construcción miope aleatorizada (descritas en la sección 3.2.1) y todas las configuraciones vecinas de la Búsqueda local (sección 3.2.2).

### **4.7.3 Resultados**

El número de evaluaciones de la función objetivo empleando la opción de maquillaje se muestra en la Tabla 4.4. En la primera columna  $n$  representa el grupo LJ, en la segunda columna se muestra el número de evaluaciones que reporta [Romero 1999a]; la tercera, cuarta, quinta y sexta columna representan la mínima, máxima, promedio y desviación estándar del número de evaluaciones que se encontraron en esta tesis.

**Tabla 4.4** Evaluaciones de la función objetivo con maquillaje.

n	Evaluaciones de la función objetivo				
	[Romero 1999a]	Mínima	Máxima	Promedio	Desviación Estándar
13	754	167	2,826	627	647
14	112	191	1,940	532	477
15	132	169	4,253	1,108	1,036
16	177	230	6,435	1,984	1,666
17	153	275	12,646	2,927	2,639
18	6,620	526	12,495	4,295	2,839
19	9,901	258	9,455	3,550	2,725
20	866	234	11,843	3,240	2,825
21	175	250	7,123	2,337	2,242
22	1,542	245	10,364	3,166	2,245
23	2,041	513	13,196	4,804	3,206
24	163	844	11,821	4,755	3,220
25	283	392	9,724	3,557	2,586
26	2,213	1,306	17,893	4,440	3,432
27	3,599	227	10,867	4,836	2,811
28	411	257	13,492	4,151	3,332
29	3,949	387	18,544	5,084	4,582
30	11,198	220	16,050	4,747	4,169
<b>Totales:</b>	<b>44,289</b>	<b>6,691</b>			

La Tabla 4.5 reporta el número de evaluaciones de la función objetivo sin utilizar la opción maquillaje, esta tabla es similar a la Tabla 4.4; por ello ya no se explica.

El número de veces o éxitos que se encontró la energía de LJ en 30 ejecuciones y el porcentaje de error promedio se presentan en la Tabla 4.6. La segunda y tercera columna muestran estos datos usando la opción maquillaje, la cuarta y la quinta los presentan sin usar el maquillaje.

**Tabla 4.5** Evaluaciones de la función objetivo sin maquillaje.

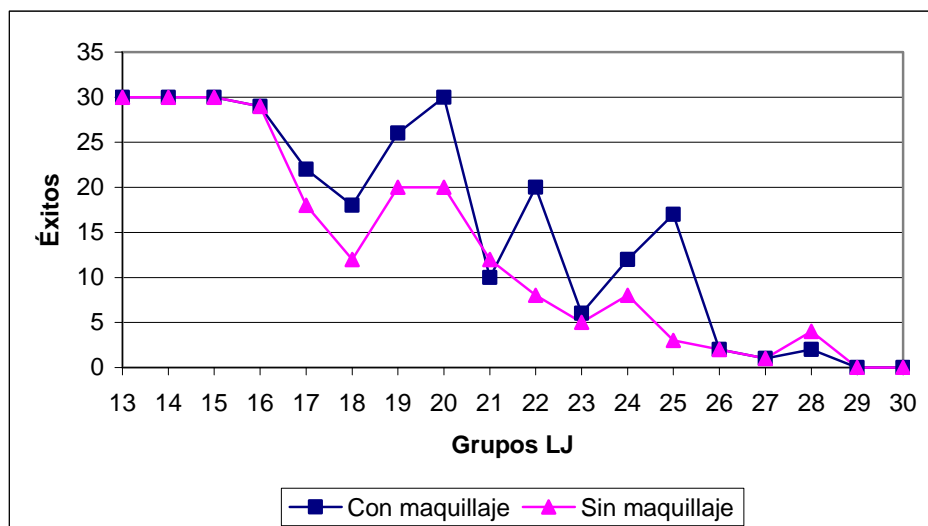
n	Evaluaciones de la función objetivo				
	[Romero 1999a]	Mínima	Máxima	Promedio	Desviación Estándar
13	754	183	1,651	462	407
14	112	206	1,379	529	344
15	132	219	3,220	790	729
16	177	209	9,443	2,047	2,047
17	153	252	11,088	2,574	2,328
18	6,620	227	6,879	3,025	2,106
19	9,901	230	11,751	2,115	2,318
20	866	272	12,188	2,916	2,632
21	175	207	6,092	2,745	1,716
22	1,542	244	11,489	2,571	2,594
23	2,041	271	10,015	3,700	2,377
24	163	241	10,827	4,323	3,157
25	283	243	14,034	4,104	3,533
26	2,213	234	8,928	2,833	2,618
27	3,599	200	10,119	4,166	3,186
28	411	167	12,852	3,280	2,943
29	3,949	208	13,921	5,025	3,896
30	11,198	216	11,451	3,153	3,054
<b>Totales:</b>	<b>44,289</b>	<b>4,029</b>			

La Figura 4.9 muestra gráficamente el número de éxitos obtenidos en cada grupo LJ de las 30 ejecuciones utilizando el maquillaje y sin emplearlo.

El porcentaje de error promedio en los grupos LJ del 13 al 30 con y sin maquillaje se puede apreciar en forma gráfica en la Figura 4.10.

**Tabla 4.6** Número de éxitos, porcentaje de error promedio en cada grupo LJ con y sin maquillaje.

n	Con Maquillaje		Sin Maquillaje	
	Éxitos	% Error Promedio	Éxitos	% Error promedio
13	30	0.00%	30	0.00%
14	30	0.00%	30	0.00%
15	30	0.00%	30	0.00%
16	29	0.05%	29	0.06%
17	22	0.01%	18	0.09%
18	18	0.26%	12	0.59%
19	26	0.30%	20	0.86%
20	30	0.00%	20	0.70%
21	10	0.06%	12	0.24%
22	20	0.16%	8	0.91%
23	6	1.34%	5	1.53%
24	12	0.52%	8	0.90%
25	17	0.32%	3	1.24%
26	2	1.41%	2	1.88%
27	1	0.98%	1	1.42%
28	2	0.54%	4	1.04%
29	0	1.33%	0	1.64%
30	0	1.02%	0	1.22%
	<b>Totales:</b>	<b>8.30%</b>		<b>14.32%</b>



**Figura 4.9** Número de éxitos en cada grupo LJ con y sin maquillaje.

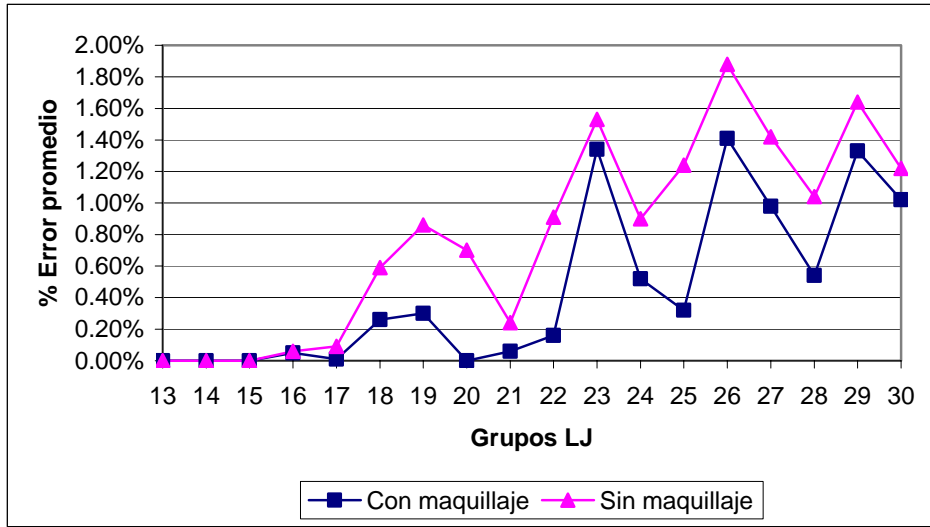


Figura 4.10 Porcentaje de error promedio en cada grupo LJ con y sin maquillaje.

#### 4.7.4 Análisis de resultados

De acuerdo a las Tablas 4.4 y 4.5 el total de evaluaciones de la función objetivo usando el maquillaje en este trabajo es de 6,691 y sin usar el maquillaje son 4,029 evaluaciones. En ambos casos comparando respecto al reportado en [Romero 1999a] hay un porcentaje de reducción de 84.89% y de 90.90% respectivamente. Aunque el número de evaluaciones incrementa cuando se usa el maquillaje, está comprobado que es mejor utilizarlo. La Tabla 4.6 muestra que el porcentaje de error promedio usando maquillaje es de 8.30% y sin utilizarlo es de 14.32%; es decir, empleando el maquillaje hay más probabilidad de encontrar la energía de LJ mejor reportada. También en la Tabla 4.6 se observa que en la mayoría de los grupos LJ con maquillaje, el número de éxitos es mayor que sin maquillaje, con excepción de los grupos 21 y 28.

## **4.8 Experimento 6: Impacto del número de configuraciones optimizadas**

### **4.8.1 Objetivo**

Evaluar el desempeño del algoritmo optimizando todas o algunas configuraciones.

### **4.8.2 Procedimiento**

En el experimento, el algoritmo se ejecutó en dos ocasiones para grupos de LJ en el rango de 13-27. La primera ocasión se ejecutó el algoritmo con todas las configuraciones y en la segunda ocasión sólo se optimizaron algunas configuraciones. Con optimizar todas las configuraciones se refiere a que se optimizaron las nueve configuraciones de la Construcción miope aleatorizada (descritas en la sección 3.2.1) y las configuraciones vecinas de la Búsqueda Local (sección 3.2.2). Optimizar algunas configuraciones significa que la configuración ocho y nueve de la Construcción miope aleatorizada no se optimizan y las configuraciones vecinas de la Búsqueda Local tampoco. Tanto como para la primera como para la segunda ocasión el algoritmo se ejecutó 30 veces por cada grupo, en cada ejecución se realizaron 400 iteraciones GRASP; aunque que no se realizaron todas las iteraciones, ya que si la energía no era actualizada en 10 iteraciones consecutivas se daba por terminado el GRASP.

### **4.8.3 Resultados**

La Tabla 4.7 muestra el número de evaluaciones de la función objetivo si se optimizan todas las configuraciones. En la columna dos aparecen las evaluaciones reportadas por [Romero 1999a], en las siguientes columnas se muestra la mínima, máxima, promedio y desviación estándar de las evaluaciones de este trabajo. En la Tabla 4.8 de manera similar que en la Tabla 4.7 se muestran las evaluaciones de la función optimizando algunas configuraciones.

**Tabla 4.7** Evaluaciones de la función objetivo optimizando todas las configuraciones.

n	Evaluaciones de la función objetivo				
	[Romero 1999a]	Mínima	Máxima	Promedio	Desviación Estándar
13	754	170	2,181	516	514
14	112	207	2,697	529	594
15	132	196	2,138	604	546
16	177	233	4,739	1,516	1,216
17	153	214	14,594	3,182	3,217
18	6,620	272	9,343	3,696	2,741
19	9,901	338	8,521	2,974	2,399
20	866	240	9,819	3,049	2,818
21	175	234	10,089	3,216	2,547
22	1,542	312	9,800	3,930	3,150
23	2,041	448	8,553	2,853	2,026
24	163	232	13,591	3,748	3,723
25	283	1,015	11,489	4,960	3,203
26	2,213	247	12,265	4,524	3,147
27	3,599	966	12,981	4,334	2,871
<b>Totales:</b>	<b>28,731</b>	<b>5,324</b>			

**Tabla 4.8** Evaluaciones de la función objetivo optimizando algunas configuraciones.

n	Evaluaciones de la función objetivo				
	[Romero 1999a]	Mínima	Máxima	Promedio	Desviación Estándar
13	754	155	1,145	352	264
14	112	147	777	325	160
15	132	181	2,359	630	560
16	177	184	2,521	966	840
17	153	257	3,890	1,688	1,046
18	6,620	273	8,497	1,787	1,853
19	9,901	268	7,210	2,557	1,832
20	866	220	6,691	2,292	1,700
21	175	263	6,533	2,454	1,917
22	1,542	209	5,630	1,966	1,365
23	2,041	268	12,291	3,393	2,672
24	163	253	12,879	3,508	3,169
25	283	227	8,563	2,965	2,402
26	2,213	319	11,733	3,503	2,746
27	3,599	285	9,580	3,300	2,405
<b>Totales:</b>	<b>28,731</b>	<b>3,509</b>			



El número de éxitos obtenidos en los grupos LJ del 13 al 27 y el porcentaje de error promedio se presentan en la Tabla 4.9, esta información es con la optimización de todas las configuraciones y con solo optimizar algunas configuraciones.

**Tabla 4.9** Número de éxitos, porcentaje de error promedio de todas y algunas configuraciones.

n	Optimizando todas las configuraciones		Optimizando algunas configuraciones	
	Éxitos	% Error Promedio	Éxitos	% Error promedio
13	30	0.00%	30	0.00%
14	30	0.00%	30	0.00%
15	30	0.00%	29	0.06%
16	30	0.00%	28	0.16%
17	27	0.00%	22	0.01%
18	16	0.26%	9	0.58%
19	18	1.00%	19	0.89%
20	26	0.23%	22	0.39%
21	14	0.06%	13	0.11%
22	15	0.40%	13	0.42%
23	10	1.15%	10	1.18%
24	19	0.32%	16	0.46%
25	14	0.33%	12	0.78%
26	3	1.28%	6	1.47%
27	3	0.74%	1	1.03%
	<b>Totales:</b>	<b>5.77%</b>		<b>7.54%</b>

La Figura 4.11 muestra de manera gráfica el número de éxitos alcanzados en los dos casos que se presentan aquí.

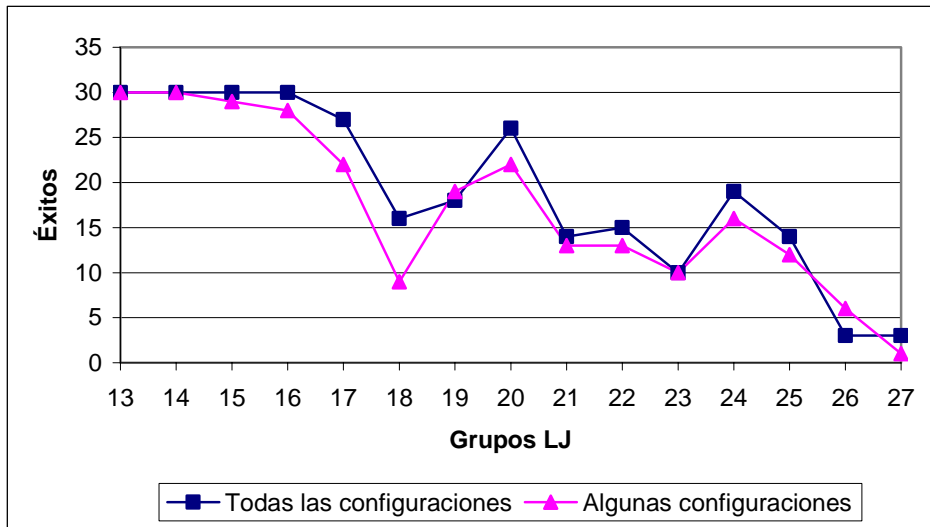


Figura 4.11 Número de éxitos en cada grupo LJ optimizando todas y algunas configuraciones.

El porcentaje de error promedio para cada grupo LJ se ve gráficamente en la Figura 4.12, para el caso de optimizar todas las configuraciones y para el caso de optimizar algunas configuraciones.

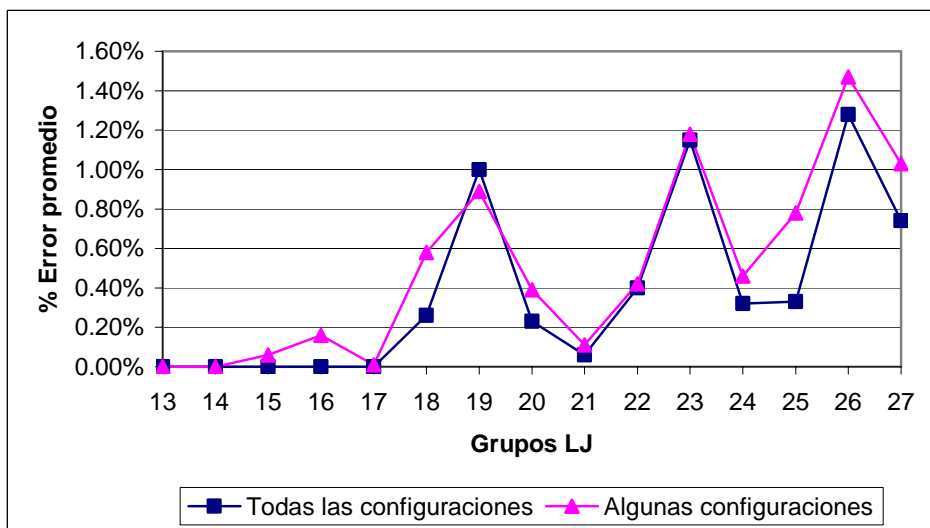


Figura 4.12 Porcentaje de error promedio en cada grupo LJ optimizando todas y algunas configuraciones.

#### **4.8.4 Análisis de resultados**

Las Tabla 4.7 muestra un total de evaluaciones de 5,324 en el rango de grupos del 13 al 27 y la Tabla 4.8 tiene un total de 3,509 evaluaciones. En las dos tablas al comparar las evaluaciones con respecto a las reportadas por [Romero 1999a] se obtiene una reducción de un 81.47% cuando se optimizan todas las configuraciones y un 87.79% si se optimizan algunas configuraciones. En cuanto al porcentaje de error promedio optimizando todas las configuraciones es de 5.77%, en cambio si se optimizan algunas configuraciones el porcentaje de error promedio es 7.54%; existe más probabilidad de encontrar la energía mejor reportada si se optimizan todas las configuraciones. Si se observa la Tabla 4.9, en la mayoría de los grupos LJ en el rango del 13 al 27 el número de éxitos es mayor al optimizar todas las configuraciones que cuando se optimizan algunas.

### **4.9 Experimento 7: Evaluación del método de reducción de la varianza**

#### **4.9.1 Objetivo**

Obtener evidencia experimental del nivel de reducción de la varianza que se puede lograr en experimentos con algoritmos metaheurísticos, al aplicar el método de reducción de la varianza propuesto en esta tesis.

#### **4.9.2 Procedimiento**

El algoritmo se ejecutó dos veces para grupos de LJ en el rango de 13-30. En la primera vez al algoritmo no se le aplicó reducción de la varianza y en la segunda vez se ejecutó empleando el método de reducción de la varianza propuesto en esta tesis (descrito en la sección 3.3). En ambos casos el algoritmo se ejecutó 30 ocasiones por cada grupo, en cada ejecución se

realizaron 400 iteraciones GRASP; aunque no se realizaron todas las iteraciones, ya que si la energía no era actualizada en 10 iteraciones consecutivas se daba por terminado el GRASP. En este experimento se optimizaron todas las configuraciones de la Construcción miope aleatorizada (sección 3.2.1) y todas las configuraciones vecinas de la Búsqueda Local (sección 3.2.2).

### 4.9.3 Resultados

La Tabla 4.10 muestra la reducción de la varianza en la energía de LJ. En la primera columna  $n$  es el grupo LJ, en la segunda columna se presenta la varianza de la energía sin aplicar el método de reducción (SR), la tercera columna es la varianza de la energía aplicando el método de reducción de la varianza (CR) y en la cuarta columna se muestra el porcentaje de reducción de la varianza para cada grupo LJ (%R). El porcentaje de reducción promedio se encuentra al final de la columna cuatro.

**Tabla 4.10** Reducción de la varianza en la Energía de LJ.

<b>n</b>	<b>SR</b>	<b>CR</b>	<b>%R</b>
13	0.000000	0.000000	0%
14	0.000000	0.000000	0%
15	0.000000	0.000000	0%
16	0.027545	0.000000	100%
17	0.000033	0.000005	85%
18	0.065718	0.028750	56%
19	0.324515	0.434445	-34%
20	0.000000	0.102788	-100%
21	0.024619	0.012260	50%
22	0.054814	0.121035	-121%
23	0.417390	0.250894	40%
24	0.320148	0.108912	66%
25	0.209553	0.086917	59%
26	0.258199	0.103297	60%
27	0.391133	0.180852	54%
28	0.305555	0.257397	16%
29	0.380853	0.146381	62%
30	0.458097	0.206154	55%
<b>Promedio:</b>			<b>25%</b>

La reducción de la varianza en el número de evaluaciones de la función objetivo se puede observar en la Tabla 4.11. La varianza de las evaluaciones sin aplicar el método de reducción se muestra en la columna dos, la tercera columna presenta la varianza de las evaluaciones empleando el método de reducción. En la cuarta columna se ve el porcentaje de reducción de la varianza.

**Tabla 4.11** Reducción de la varianza en las evaluaciones de la función objetivo.

<b>n</b>	<b>SR</b>	<b>CR</b>	<b>%R</b>
13	418,578	117,499	72%
14	227,869	155,417	32%
15	1,072,877	175,907	84%
16	2,775,921	610,527	78%
17	6,963,894	3,732,348	46%
18	8,061,432	2,510,826	69%
19	7,427,723	3,935,313	47%
20	7,981,558	5,350,087	33%
21	5,025,155	3,412,578	32%
22	5,040,562	6,033,789	-20%
23	10,279,068	1,652,359	84%
24	10,366,554	7,525,603	27%
25	6,685,933	3,348,489	50%
26	11,780,718	4,034,876	66%
27	7,902,129	3,996,836	49%
28	11,100,997	12,045,711	-9%
29	20,998,737	6,208,258	70%
30	17,377,176	7,040,370	59%
<b>Promedio:</b>			<b>48%</b>

En la Tabla 4.12 se muestra la reducción de la varianza en el número de iteraciones GRASP. La columna dos presenta la varianza en las iteraciones sin usar el método de reducción, la varianza de las iteraciones utilizando el método de reducción se observa en la columna tres. Finalmente en la columna cuatro se incluye el porcentaje de reducción de la varianza para cada grupo LJ.

#### 4.9.4 Análisis de resultados

En la Tabla 4.10 se observa un porcentaje de reducción promedio de un 25% en la energía de LJ para el rango de 13 a 30 átomos. El porcentaje de reducción promedio en las evaluaciones de la función objetivo es de un 48% (Tabla 4.11), así mismo en las iteraciones del GRASP se ve una reducción promedio de la varianza de un 48% (Tabla 4.12). Con esto se prueba que el método de reducción de la varianza propuesto en la sección 3.3 de esta tesis es realmente efectivo para reducir la varianza. Además que es importante incluir en toda experimentación con algoritmos metaheurísticos un método de reducción de la varianza, ya que mejora los resultados obtenidos.

**Tabla 4.12** Reducción de la varianza en las iteraciones del GRASP.

<b>n</b>	<b>SR</b>	<b>CR</b>	<b>%R</b>
13	0.4	0.1	80%
14	0.3	0.1	58%
15	0.9	0.2	76%
16	2.7	0.5	81%
17	5.7	2.9	49%
18	9.0	2.0	78%
19	5.3	3.8	28%
20	5.4	4.4	18%
21	4.0	2.5	37%
22	3.9	5.2	-36%
23	8.1	1.3	84%
24	8.0	5.5	32%
25	5.3	3.4	37%
26	9.5	3.6	62%
27	6.1	2.1	65%
28	7.4	8.8	-19%
29	18.5	4.5	76%
30	12.2	4.8	61%
<b>Promedio:</b>			<b>48%</b>

# Capítulo 5

## CONCLUSIONES Y TRABAJOS FUTUROS

---

En este capítulo se describen las aportaciones de este trabajo, así como las líneas de investigación identificadas en el desarrollo del mismo.

### 5.1 Conclusiones

En esta sección se describen las aportaciones más importantes de este trabajo.

En el contexto de la solución del problema de LJ se realizaron las siguientes contribuciones:

- Se desarrolló un nuevo método de solución al problema de Lennard-Jones que

incorpora un enfoque basado en rejillas, el metaheurístico GRASP y el optimizador local Regiones de Confianza. El método logra un 100% de eficiencia en la búsqueda de la mejor energía reportada para los grupos LJ de 13-25 átomos. En 8 de los grupos se logra una reducción del número de evaluaciones de la función objetivo con respecto a lo reportado en [Romero 1999a]. Además en este indicador se logra una reducción global de 0.53%, ver sección 4.5.

- Para validar el método propuesto se desarrolló un prototipo en Matlab, el cual puede ser configurado de manera sencilla para experimentar con otros modelos de energía potencial interatómica (Morse) y con diferentes rejillas básicas (FCC, HCP).
- Se realizó la evaluación experimental de los componentes estructurales más importantes del método: maquillaje de las configuraciones, estrategia de optimizaciones locales en todas las configuraciones. Una descripción detallada de esta evaluación experimental se puede encontrar en las secciones 4.7 y 4.8. Un aspecto relevante de este estudio es que a diferencia de los reportes de investigación analizados, por primera ocasión se presenta evidencia experimental soportada estadísticamente del rendimiento del método propuesto. Como resultado del análisis se observa que el uso del maquillaje en las configuraciones genera una reducción promedio del 42.04% en la calidad de la solución, por otra parte el uso de la estrategia de optimización en todas las configuraciones genera una reducción promedio del 23.47%.

En el contexto del análisis experimental de algoritmos heurísticos, se realizó la siguiente contribución:

- Se propone un nuevo método de reducción de la varianza que generaliza lo propuesto en [Fraire 2005] y que consistentemente logra una reducción de la varianza de los indicadores de rendimiento evaluados de un 25% a un 48%, los detalles se pueden ver en la sección 4.9.



## 5.2 Trabajos futuros

Otro resultado relevante de esta investigación es que permite identificar algunos problemas abiertos que pueden servir de base a futuras investigaciones. Algunos de esos problemas son los siguientes:

- **Rejillas multigeométricas**  
Como se observa en este trabajo se utiliza un solo tipo de rejilla. Sin embargo a pesar de que la rejilla tiene asociada una sola estructura geométrica icosaedra (IC) como base, el algoritmo tiene la capacidad de identificar configuraciones con estructuras diferentes de las básicas como octaedros, dodecaedros, cúbica etc. La pregunta en este sentido es si, ¿es posible la construcción de una rejilla que incorpore la estructura básica de dos o más rejillas dadas?
- **Nuevos métodos de solución**  
Uno de los resultados en este trabajo muestra que el maquillaje de las configuraciones tiene un impacto muy importante en la calidad de la solución, la pregunta en este sentido es ¿es posible diseñar nuevos métodos de solución basados en el maquillaje de las configuraciones y otras estrategias metaheurísticas?
- **Caracterización de las instancias**  
Los resultados de los niveles de reducción de la varianza alcanzaron muestra que existen algunas instancias en las que el método no logra una reducción, este fenómeno se observa también en los resultados reportados en [Fraire 2005], la pregunta en este sentido es ¿que características tienen este tipo de instancias y cual es la razón de que el mecanismo de reducción no opere sobre ellas?
- **Mejora del rendimiento de la solución**  
El prototipo que se implementó como resultado de este trabajo fue realizado utilizando

Matlab lo que limita su rendimiento potencial, se propone que este prototipo sea emigrado a lenguajes que mejoren su rendimiento como C y Fortran.

## REFERENCIAS BIBLIOGRÁFICAS

---

[Apostol 1969] Apostol T. M.: Calculus. Ed. Wiley International Edition, 2a ed. vol. II, USA, (1969)

[Barr 2001] Barr R.S., Golden B.L., Kelly J., Steward W.R., Resende M.: Guidelines for Designing and Reporting on Computational Experiments with Heuristic Methods. Proceedings of International Conference on Metaheuristics for Optimization. Kluwer Publishing, Norwell, MA (2001) 1-17

[Boyd 2006] Boyd S., Vandenberghe L.: Convex Optimization. Cambridge University Press. (2006) 1-17

[Broyden 1970] Broyden C.G.: The Convergence of a Class of Double-Rank Minimization Algorithms. Journal Inst. Math. Applic., Vol. 6. (1970) 76-90

[Cerny 1985] Cerny V.: A Thermodynamical Approach to the Travelling Salesman Problem: An Efficient Simulation Algorithm. J. Optim. Theory Appl. (1985) 41-51

[Dorigo 1992] Dorigo M.: Optimization, Learning and Natural Algorithms. Ph.D. thesis, DEI, Politecnico di Milano, Italy (1992) 140

[Dorigo 2003] Dorigo M., Stutzle, T.: Ant Colony Optimization. Mit Press, Boston, MA. (2003)

[Doye 1997] Doye J., Lennard-Jones Clusters, <http://physchem.ox.ac.uk/~doye/jon/structures/LJ/tables.150.html>

[Doye 2000] Doye J. P. K.: Physical Perspectives on the Global Optimization of Atomic Clusters. (2000).

[Dueck 1990] Dueck G., Scheuer T.: Threshold Accepting: A General Purpose Optimization Algorithm Appearing Superior to Simulated Annealing. J. Comput. Phy. Vol 90. (1990) 161-175

[Feo 1995] Feo T., Resende M.: Greedy Randomized Adaptive Search Procedures. Journal Global Optimization, No. 6. (1995) 109-133

[Fletcher 1970] Fletcher R.: A New Approach to Variable Metric Algorithms. Computer Journal, Vol. 13. (1970) 317-322

[Fraire 2005] Fraire H. J.: Una metodología para el diseño de la fragmentación y ubicación en grandes bases de datos distribuidas. Tesis doctoral. CENIDET, Cuernavaca, México (2005)

[Garey 1979] Garey M.R., Johnson D.S.: Computer and intractability: a Guide to the Theory of NP-Completeness. WH Freeman, New York (1979)

[Glover 1986] Glover F.: Future Paths for Integer Programming and Links to Artificial Intelligence. Comput. Oper. Res. Vol 13. (1986) 533-549

[Glover 1997] Glover F., Laguna M.: Tabu Search. Kluwer Academic Publishers (1997)

[Glover 1999] Glover F.: Scatter Search and Path Relinking. In New Ideas in Optimization, D. Corne, M. Dorigo and F. Glover, Eds. Advanced Topics in Computer Science Series. McGraw-Hill. (1999)

[Goldberg 1989] Goldberg D.: Genetic Algorithms in Search, Optimization and Machine Learning: Addison-Wesley (1989)

[Goldfarb 1970] Goldfarb D.: A Family of Variable Metric Updates Derived by Variational Means. *Mathematics of Computing*, Vol. 24. (1970) 23-26

[González-Velarde 1996] González-Velarde J.L.: GRASP. In A. Díaz, ed. *Heuristic Optimization and Neural Networks in Operations Management and Engineering*. Editorial Paraninfo, Madrid, (1996) 143-161

[Greenwood 2002] Greenwood G. W.: Revisiting the Complexity of Finding Globally Minimum Energy Configurations in Atomic Clusters. *Zeitschrift für Physikalische Chemie*, Vol. 211. (2002) 105-114

[Holland 1975] Holland J.: *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Harbor, MI. (1975)

[Hsu 1973] Hsu H. P.: *Análisis Vectorial*. Fondo Educativo Interamericano (1973)

[Johnson 2002] Johnson D.S., McGeoch L.A.: Experimental Analysis of Heuristics for the STSP. In: Gutin G., Punnen A. (eds.): *The Traveling Salesman Problem and its Variations*. Kluwer Academic Publishers, Dordrecht (2002) 369-443

[Kirkpatrick 1983] Kirkpatrick S., Gelatt C., Vecchi M.: Optimization by Simulated Annealing. *Science*, Vol 220. No. 4598. (1983) 671-680

[Locatelli 2000] Locatelli M., Schoen F.: *Fast Global Optimization of Difficult Lennard-Jones Clusters*. Kluwer Academic Publishers, Netherlands (2000) 1-20

[Mathews 2005] Mathews J., Fink K.: *Numerical Methods Using Matlab*. Prentice-Hall, 4 ed. (2005)

[Matlab 2004] Matlab Help, Versión 7.0.0.19920 (R14).

[McGeoch 1992] McGeoch C.C.: Analyzing Algorithms by Simulation: Variance Reduction Techniques and Simulation Speedups. *ACM Computer Survey*, Vol. 24, No. 2. (1992) 195-212

[McGeoch 2002] McGeoch C.C.: Experimental Analysis of Algorithms. In: Pardalos P.M., Romeijn H.E. *Handbook of Global Optimization*, Vol. 2. (2002) 489-513

[Meyer 2001] Meyer P., Lattice Geometries, <http://www.hermetic.ch/compsci/lattgeom.htm>

[Michalewicz 1999] Michalewicz, Z., Fogel D.B.: *How to Solve It: Modern Heuristics*. Springer Verlag (1999)

[Moré 1983] Moré J.J., Sorensen D.C.: Computing a Trust Region Step. *SIAM Journal on Scientific and Statistical Computing*, Vol. 3 (1983) 553-572

[Moret 2003] Moret B. M. E.: Toward a Discipline of Experimental Algorithmics. In: Goldwasser, M.H., Johnson, D.S., McGeoch, C. (eds.): *Data Structures, Near Neighbor Searches, and Methodology: Fifth and Sixth DIMACS Implementation Challenges*, Series DIMACS, Vol. 5 (2003) 197-214

[Neos 1996] NEOS Guide, NEOS Guide Optimization Tree, <http://www-fp.mcs.anl.gov/otc/Guide/OptWeb/index.html>

[Neumaier 1997] Neumaier A.: Molecular Modeling of Proteins and Mathematical Prediction of Protein Structure. *SIAM Review*, Vol. 39(3). (1997) 407-460

[Oviedo 2005] Universidad de Oviedo, Dibujo Asistido por Ordenador, <http://www.mieres.uniovi.es/egi/dao/apuntes/trans3d.html>

[Papadimitriou 1998] Papadimitriou C. H., Steiglitz K.: *Combinatorial Optimization: Algorithms and Complexity*. Dover Publications (1998)

[Pérez 2000] Pérez J., Pazos R.A., Frausto J., Romero D., Cruz L.: *Vertical Fragmentation and Allocation in Distributed Databases with Site Capacity Restrictions Using the Threshold Accepting Algorithm*. *Lectures Notes in Computer Science*, Vol. 1793. Springer-Verlag, Berlin Heidelberg New York (2000) 75-81

[Pérez 2002] Pérez J., Pazos R.A., Vélez L., Rodríguez G.: *Automatic Generation of Control Parameters for the Threshold Accepting Algorithm*. *Lectures Notes in Computer Science*, Vol. 2313. Springer Verlag, Berlin Heidelberg New York (2002) 119-127

[Pérez 2004] Pérez J., Pazos R.A., Frausto J., Rodríguez G., Cruz L., Mora G., Fraire H.: *Self-Tuning Mechanism for Genetic Algorithms Parameters, an Application to Data-Object Allocation in the Web*. *Lectures Notes in Computer Science*, Vol. 3046. Springer Verlag, Berlin Heidelberg New York (2004) 77-86

[Purcell 1987] Purcell E. J., Varberg D.: *Cálculo con Geometría Analítica*. Prentice-Hall, 4 ed. (1987)

[Resende 2003] Resende M. G. C., González Velarde J. L.: *GRASP: Procedimientos de Búsqueda Miopes Aleatorizados y Adaptativos*. *Inteligencia Artificial*, No. 19. (2003) 61-76

[Romero 1999a] Romero, D., Barrón, C., Gómez, S., Saavedra, A.: *A genetic algorithm for Lennard-Jones atomic clusters*. *Appl. Math. Lett.* 12 (1998) 85-90

[Romero 1999b] Romero D., Barrón C., Gómez S.: *The Optimal Geometry of Lennard-Jones Clusters*:148-309. *Computer Physics Communications* 123. (1999) 87-96

[Ross 1999] Ross, S.M.: *Simulación*. Segunda edición. Prentice Hall (1999)

[Sanvicente 2004] Sanvicente H., Frausto-Solís J.: A Method to Establish the Cooling Scheme in Algorithms Like Simulated Annealing. Lecture Notes in Computer Science. Vol. 3045. Springer Verlag Berlin Heidelberg New York. (2004) 755-763

[Shanno 1970] Shanno D.F.: Conditioning of Quasi-Newton Methods for Function Minimization. Mathematics of Computing, Vol. 24. (1970) 647-656

[Shores 2006] Shores, T.S., Public Directory Page, <http://www.math.unl.edu/~tshores1/Public/Math433S05/uncprobs/>

[Wales 1999] Wales D. J., Scheraga H. A.: Global Optimization of Clusters, Crystals and Biomolecules. Science, Vol. 285. (1999) 1368-1372

[Watkins 2006] Watkins S. F., Crystal Lattice Structures: Face Centered Cubic, <http://wb.chem.lsu.edu/htdocs/people/sfwatkins/MERLOT/lattice/04fcc.html>

[Xue 1992] Xue G.L., Maier R.S., Rosen J.B.: A discrete-continuous algorithm for molecular energy minimization. IEEE Computer Society Press. (1992) 778-786

[Yagiura 2001] Yagiura M., Ibaraki T.: On Metaheuristic Algorithms for Combinatorial Optimization Problems: The Transactions of Institute of Electronics, Information and Communication Engineers. J83-D-1(1), (2001) 3-25

[Zhao 2004] Zhao J., Xie R.: Genetic Algorithms for the Geometry Optimization of Atomic and Molecular Clusters. Journal of Computational and Theoretical Nanoscience, Vol. 1. American Scientific Publishers (2004) 117-131