



Casa abierta al tiempo

**UNIVERSIDAD AUTÓNOMA METROPOLITANA**  
**Unidad Iztapalapa**

División de Ciencias Básicas e Ingeniería

Departamento de Química

Área de Fisicoquímica Teórica

---

# Búsqueda Estocástica de Isómeros Restringidos Espacialmente

---

**TESIS**

para el obtener el grado de  
**MAESTRO EN CIENCIAS (QUÍMICA)**

**Autor:**

Juan José García Miranda

**Asesor:**

Dr. Jorge Garza Olgún

El jurado estuvo formado por los siguientes sinodales:

**Presidente:** Dr. Marcelo Enrique Galván Espinosa

**Secretario:** Dr. Jorge Garza Olgún

**Vocal:** Dr. Rodolfo Gómez Balderas

El examen de grado se llevó a cabo el 14 de diciembre de 2018  
a las 10:00 horas en la Universidad Autónoma Metropolitana-Iztapalapa

*“A veces, la implementación elegante es sólo una función. No es un método. No es una clase. No es un marco. Sólo una función”*

---

*John Carmack*

Este proyecto fue realizado en el Área de Físicoquímica Teórica del Departamento de Química de la Universidad Autónoma Metropolitana-Iztapalapa.

Algunos de los resultados de este proyecto fueron difundidos en:

- 44<sup>a</sup> Congreso Internacional de Químicos Teóricos de Expresión Latina (QUITEL 2018). **Restricted Isomers Searching by the Simulated Annealing Method**. Santiago de Chile, Chile, 2018.
- XVII -Reunión Mexicana de Físicoquímica Teórica. **Búsqueda Estocástica de Isómeros Restringidos Espacialmente**. Monterrey, México, 2018.

# Agradecimientos

*“Nunca confíes en una computadora que  
no puedas arrojar por la ventana”*

---

*Steve Wozniak*

Agradezco con especial énfasis a mi asesor el Dr. Jorge Garza Olgún por haberme guiado en el transcurso de mis dos años de maestría, por todas sus enseñanzas y por asignarme un tema de investigación totalmente ajeno a mi formación previa con el cual me pude adentrarme a nuevas áreas de conocimiento logrando mejorar mis habilidades de programación.

A la Dra. Graciela Roman y a todos los profesores del área de computación y sistemas de la Universidad Autónoma Metropolitana-Iztapalapa que me permitieron asistir a sus clases las cuales fueron de gran ayuda para mi desarrollo académico.

A mi madre Ángela Miranda por haberme apoyado en estos dos años más de estudios.

A CONACyT por el apoyo económico a través de la beca 467303.

# Índice general

|  |           |
|--|-----------|
| <b>1. Introducción</b>   | <b>6</b>  |
| <b>2. Teoría</b>   | <b>8</b>  |
| 2.1. Complejidad algorítmica . . . . .   | 8         |
| 2.1.1. Análisis asintótico: la notación “ <i>Big O</i> ” . . . . .                 | 8         |
| 2.1.2. Las clases P, NP y NP-completa . . . . .                                    | 10        |
| 2.1.3. Métodos deterministas y métodos estocásticos . . . . .                      | 11        |
| 2.2. Predicción de estructuras químicas . . . . .                                  | 16        |
| 2.2.1. Optimización molecular . . . . .  | 17        |
| 2.2.2. Sistemas finitos . . . . .  | 18        |
| 2.2.3. Sistemas periódicos . . . . .   | 19        |
| 2.3. El lenguaje de programación . . . . .   | 22        |
| 2.3.1. Estructuras de Datos . . . . .  | 23        |
| <b>3. Metodología</b>  | <b>26</b> |
| 3.1. El algoritmo del Recocido Simulado . . . . .                                  | 27        |
| 3.2. Uso de estructuras . . . . .  | 29        |
| <b>4. El código RISSA</b>  | <b>30</b> |
| 4.1. Funciones objetivo, restricciones geométricas y archivos de entrada . . . . . | 30        |
| 4.1.1. Restricciones espaciales . . . . .  | 31        |
| 4.1.2. Funciones objetivo . . . . .  | 32        |
| 4.1.3. Archivos de entrada . . . . .   | 33        |
| 4.2. Detalles en la programación . . . . .   | 35        |
| 4.2.1. Movimientos aleatorios y restricciones espaciales . . . . .                 | 35        |
| 4.2.2. Archivos de salida . . . . .  | 38        |
| 4.2.3. Ejecución de RISSA . . . . .  | 38        |
| 4.3. Calibración de RISSA . . . . .  | 39        |
| <b>5. Resultados</b>   | <b>41</b> |
| 5.1. Primer sistema: $\text{NaCl}(\text{H}_2\text{O})_2$ . . . . .                 | 42        |
| 5.2. Segundo sistema: $(\text{NaCl})_2(\text{H}_2\text{O})_4$ . . . . .            | 43        |
| 5.3. Tercer sistema: $(\text{NaCl})_4(\text{H}_2\text{O})_8$ . . . . .             | 45        |
| <b>6. Perspectivas: periodicidad en 3D y 2D</b>                                    | <b>51</b> |
| <b>7. Conclusiones</b>   | <b>54</b> |

# Prefacio

En esta tesis, se presentan los resultados de la implementación y del uso de un método de optimización heurístico llamado “Recocido Simulado” para la búsqueda aleatoria de isómeros químicos sometidos a restricciones espaciales. La programación del algoritmo del recocido simulado fue realizada desde cero en lenguaje de programación C, haciendo uso de estructuras de datos para lograr el mejor desempeño del código. Muchos de los resultados obtenidos se centran en la obtención de estructuras químicas en fase gas, sin embargo nuestro código que lleva por nombre RISSA (Restricted Isomers Searchig by Simualted Annealing) puede ser aplicado para sistemas en fase sólida.

El segundo capítulo abarca la teoría detrás de la complejidad que existe en los problemas computacionales y el inminente uso de metodologías estocásticas para resolver los problemas denominados como intratables que frecuentemente aparecen en todas las ramas de la ciencia. Por este motivo se da una pequeña introducción a la notación “O grande” referida al peor escenario en tiempo de ejecución de los algoritmos, para de esta manera poder catalogarlos dentro de alguna de las clases P, NP y NP-completa. Por otra parte, se presentan las aproximaciones deterministas y estocásticas, así como algunos ejemplos de los algoritmos de carácter estocástico más usados para la optimización de funciones. Además, se hace un análisis detallado de los usos de los métodos estocásticos para la predicción de estructuras químicas tanto en fase gas como en fase sólida.

En el tercer capítulo relacionado con la metodología, hace una descripción minuciosa del algoritmo del Recocido Simulado, así como algunos detalles en el manejo de estructuras de datos en el lenguaje de programación C necesarios para una total comprensión del código fuente.

El capítulo cuatro presenta el código RISSA desarrollado para este trabajo de maestría. Todo lo relacionado con su uso: manejo de archivos de entrada y descripción de las funciones objetivo implementadas. Los detalles en la programación descritos en este capítulo abarcan la manera en que se generan los movimientos aleatorios, las condiciones necesarias para cumplir las restricciones espaciales implementadas, así como la información de la simulación que proporcionan los archivos de salida. Se presentan dos estudios hechos en la predicción de sistemas de dos sistemas ya estudiados y publicados con anterioridad (tetrámero de agua y cúmulos de  $\text{Na}_4$ ) como una forma de prueba y calibración de nuestro código.

El capítulo cinco muestra la potencia del código RISSA. Su uso en sistemas no estudiados previamente como lo son los agregados de cloruro de sodio dihidratado muestra la gran utilidad de las metodologías estocásticas en la predicción de estructuras químicas iniciales necesarias para explorar la Superficie de Energía Potencial. Las optimizaciones de las geometrías finales obtenidas de manera estocástica para los agregados  $(\text{NaCl})(\text{H}_2\text{O})_2$ ,  $(\text{NaCl})_2(\text{H}_2\text{O})_4$  y  $(\text{NaCl})_4(\text{H}_2\text{O})_8$  fueron realizadas bajo la Teoría de Funcionales de la Densidad. Lo cual muestra que en conjunto, un buen algoritmo de búsqueda local y una buena metodología de carácter determinista son la clave para poder predecir estructuras químicas teniendo como única información la composición del sistema.

# Capítulo 1

## Introducción

La disposición que toman los átomos en el espacio es la pieza de información más importante y fundamental cuando se estudian las propiedades fisicoquímicas de cristales, superficies, nanocristales, cúmulos atómicos y moleculares. Para ello, actualmente existen paquetes computacionales de estructura electrónica muy eficientes, capaces de llevar a cabo cálculos energéticos altamente precisos inclusive con una cantidad considerable de átomos. Llevar a cabo este tipo de cálculos es simple, basta con darle al programa una estructura inicial y subsecuentemente podremos calcular todo tipo de propiedades fisicoquímicas que se nos ocurran del sistema en cuestión, lamentablemente muchas de estas estructuras iniciales provienen de información experimental y en ciertas ocasiones mantener las condiciones de laboratorio ideales para obtener buenos productos de síntesis que sean altamente identificables por algún método de caracterización, resulta ser complicado. Por varios años ha sido el cuello de botella para la química teórica, debido a que en ausencia de información experimental es necesario recurrir a la intuición química la cual en ciertas ocasiones no es del todo acertada. En este sentido los retos a los que se ha enfrentado la química teórica han subido de nivel con el paso del tiempo, han quedado atrás aquellos años en que el objetivo principal era el de resolver la ecuación de Schrödinger para sistemas con grandes cantidades de átomos, para pasar a otro que siempre se ha mantenido latente y que hasta estos días se pensaba como una tarea totalmente imposible: “la predicción de estructuras químicas”, esto es, encontrar la estructura de mínima energía teniendo como única información la composición química del sistema, por supuesto sin hacer uso de la intuición humana. Esta predicción de estructuras ya sea en fase sólida o en fase gas está altamente relacionada con hacer una búsqueda de isómeros con restricciones espaciales y es un tema con gran auge dentro de la comunidad científica debido a su inminente aplicación al diseño de nuevos materiales. Encontrar el isómero de más baja energía conociendo solamente el tipo de átomos involucrados en el sistema, requiere de un tipo especial de algoritmos llamados “estocásticos” los cuales facilitarían esta tarea.

Dentro de la gran cantidad de isómeros que existen en la química, los isómeros conformacionales y los isómeros estructurales han sido de mayor interés en los estudios de la química teórica. En términos prácticos en un isómero conformacional se presentan cambios en las posiciones de los átomos manteniendo fija la conectividad entre ellos, por otro lado en los isómeros estructurales se presentan cambios las posiciones atómicas y en las conectividades. Esta sencilla descripción sitúa a los isómeros estructurales y conformacionales en dos espacios matemáticos diferentes [1]. Por un lado los isómeros conformacionales pertenecen a un espacio continuo, debido a que se puede generar una secuencia continua de transformaciones alrededor de un enlace para pasar de un conformero a otro. El espacio matemático de los isómeros estructurales es un espacio discreto ya que a cada estructura molecular que generemos le corresponde un único grafo molecular (un grafo es un conjunto

---

de vértices y aristas unidos entre sí). Para ambos tipos de isómeros, el interés está en explorar la Superficie de Energía Potencial (SEP) la cual representa los cambios de energía que se obtienen respecto a reacomodos en las posiciones atómicas para encontrar el Mínimo Global (MG) de dicha superficie.

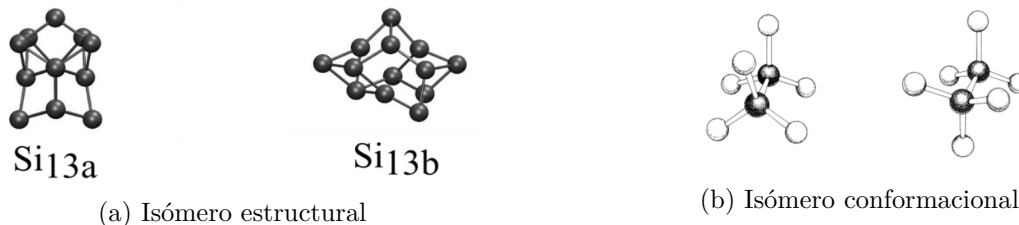


Figura 1.1: Tipos de isómeros generalmente estudiados por la química teórica

Llegar al MG en este contexto, significa que dentro de un conjunto de posibles combinaciones en las posiciones atómicas hemos encontrado la configuración en el espacio que minimiza la energía. En la figura (1.1a) se muestran dos isómeros de un cúmulo de 13 átomos de Silicio, para saber cuál es el isómero más estable podemos realizar un cálculo de estructura electrónica, al cual llamaremos “método determinista”, para determinar cuál posee la energía más baja, sin embargo existe la posibilidad de que otro isómero de Si<sub>13</sub> presente una energía aún más profunda. La manera tradicional de resolver este problema es proponer una serie de diferentes configuraciones hasta encontrar la geometría con la mínima energía. De esta manera el número de combinaciones posibles aumenta exponencialmente con el número de átomos involucrados en el sistema. Cordatos y colaboradores [2] aseguran que el problema de encontrar la estructura de más baja energía de un cúmulo de Cerio es equivalente a encontrar el MG de toda la superficie, así el número de mínimos locales y el tiempo de cómputo aumentan exponencialmente con el número de partículas. Este tipo de búsquedas caen dentro de los problemas computacionales de tipo NP-Complejos (Polinomial no determinista) para los cuales no se conocen algoritmos que los resuelvan en un tiempo de ejecución polinomial, es decir, no hay un polinomio asociado a la variable de entrada. Si existen más de un tipo de átomos el número de isómeros incrementa drásticamente. Afortunadamente no tenemos que explorar punto a punto la Superficie de Energía Potencial para encontrar el mínimo global, podemos hacerlo aleatoriamente utilizando métodos “estocásticos”, de esta manera logramos hacer una búsqueda exhaustiva ahorrando valioso tiempo de cómputo. El inconveniente de usar métodos estocásticos es que no proporcionan una única solución, a cambio de ello, generan un conjunto de buenas soluciones las cuales son candidatas para llegar a la solución óptima del problema, sin embargo el uso de métodos estocásticos ha tenido un gran auge en los últimos años debido al gran esfuerzo computacional que involucra explorar la SEP de los isómeros para encontrar el mínimo global, estas metodologías estocásticas han llevado a poder predecir nuevas estructuras cristalinas y nuevos cúmulos moleculares. La revolucionaria predicción de estructuras químicas encabezada por el científico ruso Artem R. Oganov con su algoritmo evolutivo USPEX[3] está tomando fuerza dentro de la comunidad científica incentivando a la creación de códigos de carácter estocástico para llevar a cabo la tarea de predecir todo tipo de estructuras químicas. Este proyecto de maestría está enfocado en hacer una búsqueda de isómeros con diferentes tipos de restricciones espaciales, ya que dependiendo del tipo de restricción geométrica estaremos simulando sistemas de diferente naturaleza química: sistemas extendidos o en fase gas.

# Capítulo 2

## Teoría

*“Algoritmos: un lenguaje común entre naturaleza, humanos y computadoras”*

---

*Avi Wigderson*

### 2.1. Complejidad algorítmica

Sin importar que estemos interesados en estudiar isómeros estructurales o isómeros conformacionales, requerimos de un algoritmo que nos ayude con la extensa tarea de explorar el espacio configuracional de los isómeros. Cuando elaboramos un algoritmo buscamos que sea eficiente, esto significa que para una entrada dada genere una salida en un número finito de pasos y como se mencionó en el capítulo (1) que se ejecute en tiempo polinomial. Encontrar la mejor solución a un problema computacional requiere de una optimización, ya sea para encontrar el mínimo óptimo o el máximo óptimo. Cuando el sistema es continuo las variables de las cuales depende el sistema cambian de forma continua a lo largo del tiempo, por ejemplo la posición y velocidad de un automóvil en la carretera, en cambio, cuando el sistema que estamos modelando es discreto, las variables del problema cambian en conjuntos separados en el tiempo, por ejemplo el número de clientes que recibe un banco a lo largo del día, ya que es imposible tener cliente y medio en algún instante del tiempo. Bajo estos conceptos existen dos tipos de optimizaciones:

1. Problemas de optimización de variables continuas.
2. Problemas de optimización de variables discretas.

Para el primer caso es claro que el dominio de la función debe ser continuo  $f(x_{opt})$  y las metodologías deterministas son las encargadas de abordar este tipo de optimizaciones. En el segundo caso el dominio de la función es discreto y las llamadas metodologías estocásticas o heurísticas son las que llevan a cabo este tipo de optimizaciones combinatorias.

#### 2.1.1. Análisis asintótico: la notación “*Big O*”

La notación “Big O” o en español “O grande” es usada en las ciencias computacionales para describir la complejidad de un algoritmo. Esta notación específicamente describe el peor escenario y es utilizada para describir el tiempo de ejecución requerido o el espacio usado (en memoria) por un algoritmo. Esto es de suma importancia cuando distinguimos entre los tipos de metodologías “deterministas” y “estocásticas”.



---

## Notación y terminología

Sean  $f$  y  $g$  funciones con dominio  $N = \{1, 2, 3, \dots\}$  entonces  $f(n) = O(g(n))$  si:

- $f(n)$  es como máximo del orden de  $g(n)$

## Complejidad $O(n)$

Describe un algoritmo cuya ejecución crece linealmente y en proporción directa con el conjunto de datos de entrada.

- Búsqueda secuencial: (secuencia de longitud  $n$ )
  - Mejor caso: Encontrar el elemento en la primera prueba. Una iteración en el ciclo  $O(1)$ .
  - Peor caso: No encontrar el elemento. Toma  $n$  iteraciones en el ciclo  $O(n)$ .

El peor de los casos está representado a continuación

---

```
int seq_funct(int x, int a[], int n) {
    for(int i=0; i<n; i++){
        if(x==a[i]) return a[i];
    }
    return 0;}

```

---

## Complejidad $O(n^2)$

Representa un algoritmo cuya ejecución es directamente proporcional al cuadrado del tamaño del conjunto de datos de entrada. Esta complejidad es común en algoritmos que involucran iteraciones anidadas sobre el conjunto de datos. Iteraciones anidadas muy profundas resultan en  $O(n^3)$ ,  $O(n^4)$  y así sucesivamente.

- Algoritmo burbuja para ordenamiento: (secuencia de orden  $n$ )
  - Mejor tiempo de ejecución:  $O(n^2)$ .
  - Peor caso de ejecución:  $O(n^2)$ .
  - Tanto el mejor tiempo de ejecución como el peor tienen complejidad  $O(n^2)$  ya que siempre es necesario recorrer todo el arreglo.

El algoritmo burbuja para ordenamiento se muestra en el código siguiente:

---

```
void bubble_sort(int a[], int n){
    int temp;
    for(int i=0; i<n-1; i++){
        for(int j=0; j<n-1-i; j++){
            if(a[j]>a[j+1]){
                temp=a[j];
                a[j] =a[j+1];
                a[j+1]=temp;
            }
        }
    }
}

```

---

Es debido a los dos ciclos anidados que el mejor tiempo de ejecución es  $O(n^2)$ .

---

## Complejidad $O(\log N)$

La búsqueda binaria es un clásico ejemplo de la complejidad  $O(\log N)$ , es una técnica usada para ordenamiento de datos. Funciona seleccionando el elemento medio del conjunto de datos el cual se compara con el valor objetivo. Si el valor coincide la búsqueda termina. Por el contrario si el valor es más grande, el algoritmo tomará la parte superior del conjunto de datos y repetirá el mismo procedimiento. De la misma manera si el valor objetivo es menor, se repetirá la búsqueda tomando la mitad inferior del conjunto de datos.

- Búsqueda binaria: (secuencia de longitud  $n$ )
  - Mejor escenario: valor objetivo ubicado en el medio del arreglo. Una iteración.
  - Peor escenario: Un posible tiempo  $\log N$  dentro del ciclo.

---

```
int binary_search(int x, int a[], int n){
    int low=0;
    int up=n-1;
    while(low<=up){
        int mid = (low+up)/2;
        if(a[mid]==x) {return mid;}
        else if{ x<a[mid]} {up = mid - 1;}
        else { low = mid + 1;}
    }
    return 0;
}
```

---

## Complejidad $O(c^n)$

El tiempo exponencial usualmente es para situaciones donde realmente no se conoce mucho sobre el problema, por lo cual es necesario probar cada posible combinación o permutación. El número de pasos que lleva completar la tarea es una constante  $c$  elevada a la potencia  $n$  que corresponde al número de entradas del problema. En el caso particular de la complejidad  $O(2^n)$  denota un algoritmo el cual duplica su crecimiento con cada adición en el número de entradas  $n$ . Un ejemplo de ello es la función recursiva para calcular los números Fibonacci.

---

```
int fibonacci(int n){
    if(n<=1) {return number;}
    else{return fibonacci(n-2) + fibonacci(number + 1);}
}
```

---

### 2.1.2. Las clases P, NP y NP-completa

Muchos de los problemas computacionales tratados en la ciencia utilizan algoritmos de tipo polinomial[4] P. Generalmente los algoritmos que se ejecutan en tiempo polinomial los identificamos como “tratables” o “fáciles” de resolver, son algoritmos los cuales tienen como peor escenario un tiempo  $O(n^k)$  para una constante  $k$  dada siendo  $n$  el tamaño de la entrada. Podemos tener un gran número de problemas que puedan resolverse con un gran número de algoritmos. La complejidad del problema tiene relación directa con la del mejor algoritmo que lo resuelva, como ejemplo, el problema de ordenar números puede llevarse a cabo con un distintos algoritmos como lo es el “Heapsort” y

el “Bubble Sort”. Por otro lado, decimos que son “intratables” o “difíciles” aquellos problemas que requieren algoritmos con un tiempo superpolinomial de ejecución, a este conjunto de problemas se les conoce como **NP** el cual significa “Polinomial No Determinista”, este tipo de problemas son fáciles de verificar pero difíciles de solucionar. Un ejemplo de problema de tipo NP es un rompecabezas, puesto que es fácil comprobar si está correctamente armado simplemente verificando que las piezas coincidan entre ellas, pero es muy distinto tratar de encontrar un algoritmo con el cual podamos armar el rompecabezas desde cero. Es fácil darse cuenta que todo P está en NP ya que si podemos resolver un problema de tipo P en tiempo razonable<sup>1</sup> también podemos verificarlo en tiempo razonable. La llamada clase **NP-completa**, contiene a los problemas mas duros (los más difíciles) de la clase NP que existen. Cuando demostramos que un problema pertenece a la clase NP-completa, realmente estamos mostrando qué tan difícil es resolverlo en lugar de qué tan fácil es de resolver. La clase NP-completa está dentro de la clase P, esto es importante debido a la constante y no resuelta pregunta ¿es  $P=NP$ ? Esta pregunta formulada en 1971 y aún sin respuesta es el gran reto para matemáticos e investigadores de las ciencias computacionales.

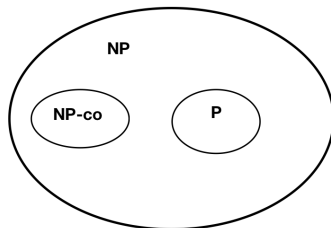


Figura 2.1: Relación simple entre las clases P, NP y NP-completa

En la figura (2.1) se muestra un diagrama sobre la relación que existe entre las tres diferentes clases de complejidad explicadas hasta el momento. Si llegásemos a la conclusión de que  $P \neq NP$  las cosas en la ciencia quedarían igual. Pero en el caso de que se demostrara que  $P=NP$  entonces las cosas cambiarían de una forma impresionante. De encontrar un algoritmo que resuelva un problema de complejidad NP-completa en tiempo polinomial, entonces cualquier problema de esta clase quedaría resuelto en tiempo polinómico y además la clase NP colapsaría a P. De lograr esto último muchos problemas de la ciencia quedarían resueltos, con ello, la predicción de estructuras químicas. Podríamos intentar predecir un gran número de isómeros sin el temor de llevarnos años en gasto computacional.

Para ser buen diseñador de algoritmos es necesario comprender todo lo que implica la clase NP-completa. Poder categorizar un problema como NP-completo provee una buena evidencia de su intratabilidad, con ello se podría invertir más tiempo en encontrar un buen algoritmo que aproxime una solución al problema en lugar de gastar tiempo en desarrollar un algoritmo que encuentre la solución exacta.

### 2.1.3. Métodos deterministas y métodos estocásticos

Dentro de la ciencia, modelar se refiere al proceso de generar una representación simple de un sistema real. Un modelo adecuado debe ser capaz de sustentar observaciones previas, integrar datos actuales del sistema y predecir con exactitud razonable la respuesta a los cambios hechos al sistema. Los modelos han evolucionado de una manera armoniosa con la ciencia debido a la gran ayuda que

<sup>1</sup>Con tiempo razonable nos referimos a una complejidad del tipo  $O(n^k)$

aportan para resolver algún paradigma científico. Existen tres tipos de modelos: modelos de analogía, modelos a escala y modelos matemáticos. Un modelo matemático describe un sistema por medio de un conjunto de variables de estado y un conjunto de ecuaciones que establecen relación entre ellas. Los modelos matemáticos pueden ser analíticos o numéricos. Los modelos analíticos con frecuencia requieren de simplificaciones para lograr que las ecuaciones tengan solución. En cambio, los modelos numéricos son más versátiles y hacen uso de computadoras para resolver las ecuaciones involucradas.

Los modelos matemáticos pueden ser determinísticos o estocásticos. Un método determinista es un modelo matemático donde las mismas entradas producirán invariable las mismas salidas. Sin embargo, los métodos deterministas suelen ser inestables, lo cual significa que pequeñas perturbaciones en las condiciones iniciales conducen a grandes variaciones en la solución final. Así, a pesar del hecho de que la solución es única, uno podría obtener soluciones que son significativamente diferentes por medio de cambios a las condiciones iniciales. Por el contrario, los parámetros de un modelo estocástico están descritos por medio de variables o distribuciones aleatorias. De esta manera un modelo estocástico produce una variedad de soluciones igualmente probables. En una manera simple una simulación estocástica es repetir muchas veces de manera artificial un problema (mediante un muestreo de variable aleatorias). Estas simulaciones sirven para entender el fenómeno de estudio y obtener una solución aproximada al problema.

## Optimización determinista

Optimizar una función  $f$  requiere de encontrar el punto donde el valor de la función  $f$  es mínimo o es máximo. Como es natural pensar, el esfuerzo computacional dependerá de la complejidad de evaluar la función. El extremo de una función (un punto máximo o mínimo) puede ser global (el verdadero valor máximo o mínimo de la función) o local (un valor máximo o mínimo en una región de la función). La figura (2.2) muestra una gráfica del cambio de la función  $E(x)$  con respecto a la variable  $x$  la cual contiene algunos mínimos y máximos. El determinismo en este caso radica en los puntos para iniciar la optimización, debido a que dependiendo del punto de partida es el extremo de la función al cual llegaremos.

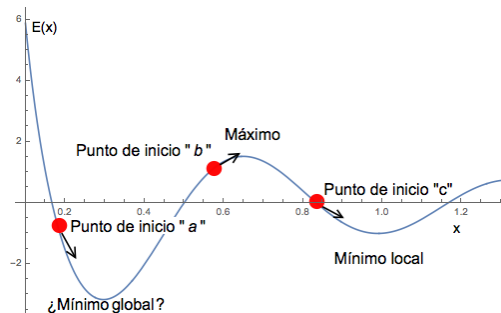


Figura 2.2: Los métodos deterministas dependen del punto de partida

Minimizar una función representa una extensa tarea y dependiendo de su naturaleza se debe elegir de manera correcta entre métodos que sólo necesitan de la evaluación de la función y métodos que también requieren de las derivadas de la función. Desde luego en el caso de funciones multivariadas se requiere del gradiente. Dentro de las aproximaciones deterministas existen dos grupos de algoritmos para la minimización de funciones[5] los cuales requieren de las primeras derivadas. El primero de ellos es conocido como “métodos de gradiente conjugado”, los cuales tienen un requisito en memoria y en tiempo de ejecución de  $O(n)$ . El segundo grupo se conoce como “métodos cuasi-

Newton” y su requisito en memoria es de  $O(n^2)$ , debido a que necesitan almacenar una matriz del orden de  $n \times n$ . Por supuesto este esfuerzo computacional aumenta cuando la función a la cual se quiere minimizar presenta un número considerable de mínimos locales. En la química, muchos de los isómeros presentan Superficies de Energía Potencial aún más complejas que las mostradas en la (2.3).

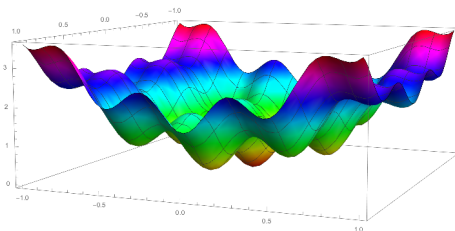


Figura 2.3: Ejemplo de gráfica multimodal

Para estas situaciones la mejor opción para tener una buena aproximación a la solución óptima es una optimización estocástica.

### Ejemplo de un algoritmo de optimización determinista

El algoritmo conocido como “Gradiente Descendente” pertenece al grupo de los métodos de gradiente conjugado y es de los más utilizados en “*Machine Learning*” y “*Deep Learning*”. Utilizando el método del gradiente descendente podemos encontrar un mínimo local de alguna función. La manera en la que funciona es simple, comienza con una prueba inicial de la solución y toma el gradiente de la función en ese punto. Pasamos la solución en la dirección negativa del gradiente y repetimos el proceso. El algoritmo eventualmente logrará tener una convergencia donde el gradiente tome el valor de cero (el cual corresponde a un mínimo local de la función). Su similar, el método del “Gradiente Ascendente”, busca el máximo local cercano a la solución actual siguiendo la dirección en donde el gradiente tome valores positivos. Ambos algoritmos son conocidos como algoritmos de primer orden porque sólo toman las primeras derivadas de la función.

Si estuviéramos interesados en encontrar el mínimo de alguna función  $f(\vec{r})$ . Dado un valor inicial  $\vec{r}_0$  para el vector  $\vec{r}$ , podemos generar cambios en el valor del vector en varias direcciones. Para resolver cuál es la mejor dirección que minimiza  $f$  tomamos el gradiente  $\nabla f$ . De manera intuitiva, el gradiente que proporcionará la pendiente de la curva en  $\vec{r}$  y la dirección apuntará al aumento de la función. Entonces, cambiamos el valor de  $\vec{r}$  en la dirección contraria para disminuir el valor de la función.

$$\vec{r}_{i+1} = \vec{r}_i - \lambda \nabla f(\vec{r}_i). \quad (2.1)$$

La constante  $\lambda$  (cuyo valor es mayor a cero) representa un número el cual restringe al algoritmo de hacer pequeños saltos. Eso mantiene estable al algoritmo y el valor óptimo de  $\lambda$  depende de la naturaleza de la función. Si damos buenas condiciones iniciales (una acertada elección de  $\lambda$ ) podemos garantizar el hecho de que  $f(\vec{r}_{k+1}) \leq f(\vec{r}_k)$ .

### Optimización estocástica

La limitante de los modelos deterministas es que los parámetros que gobiernan las ecuaciones deben ser conocidos y las soluciones deben ser únicas.. Por esta razón, a pesar de que la física o

---

la química del sistema sea simple y comprensible por medio de ecuaciones deterministas, es difícil confiar en los resultados, ya que los parámetros de entrada, geometrías, condiciones iniciales y de frontera no son del todo conocidas o en el peor de los casos son simples aproximaciones.

En una optimización, las simulaciones estocásticas se utilizan para mejorar la solución de un problema computacionalmente intratable. Los algoritmos de carácter estocástico proporcionan un conjunto de soluciones cercanas a la óptima global de problemas computacionales con una complejidad que en la mayoría de sus casos del tipo  $O(e^n)$ . Esto significa que en un tiempo polimomial podremos encontrar soluciones que probablemente están dentro de un pequeño factor de la solución optima. La característica más importante de las metodogías estocásticas es que utilizan una muestra de variables aleatorias para aproximar el valor de la función objetivo. La figura (2.4) muestra esquemáticamente la manera en que una superficie multimodal es explorada aleatoriamente.

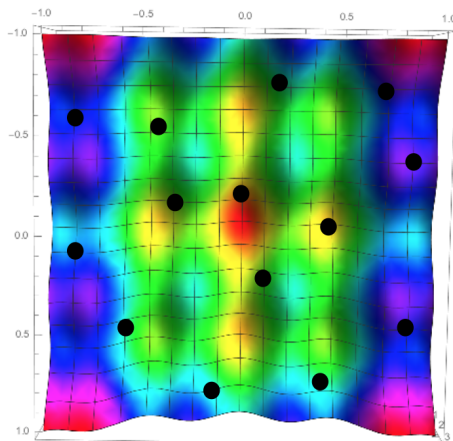


Figura 2.4: Superficie multimodal explorada aleatoriamente

En química, la complejidad por encontrar la estructura de más baja energía entre todas las posibles es una tarea costosa debido al número de ecuaciones involucradas para calcular la energía del sistema. Dicha tarea incrementa su dificultad al aumentar el tamaño del sistema. Por este motivo el uso de métodos estocásticos se está convirtiendo en una revolucionaria herramienta para acercarse a las mejores posibles soluciones.

## Principales algoritmos de búsqueda estocástica

En este trabajo estamos enfocados en el uso de metodogías estocásticas debido al gran impacto que han tenido estos algoritmos en la química. Tales aplicaciones han llevado al descubrimiento de muchas estructuras que la intuición química simplemente nos permitiría obtener. Existen diferentes algoritmos de carácter estocástico, cada uno de ellos sigue sus propias reglas y como es de esperarse los resultados entre ellos pueden variar drásticamente.

### 1. Algoritmos de Inteligencia de Enjambre

Estos algoritmos basan sus fundamentos en la manera en que la naturaleza se desenvuelve, esto es por ejemplo los comportamientos animales. Para ello los algoritmos basados en *Inteligencia de Enjambre* [6] buscan resolver el problema de manera colectiva, por ejemplo emulando el comportamiento de ciertas especies de hormigas que rápidamente encuentran caminos más

---

cortos desde su nido a un sitio con mejor alimento o colonias de abejas que mediante bailes se comunican entre ellas los mejores sitios que les permitan recolectar la mayor cantidad de néctar.

Los principales algoritmos basados en la inteligencia de enjambre son:

- Optimización por Enjambre de Partículas

La Optimización por Enjambre de Partículas (PSO, por sus siglas en inglés Particle Swarm Optimization). PSO es un método adaptativo que hace uso de partículas que se mueven a través del espacio de búsqueda utilizando los principios de Evaluación, Comparación e Imitación. Este método de optimización de basa en el uso de un conjunto de partículas las cuales hacen referencia a los estados de un problema de optimización, donde cada partícula se moverá en busca de una posición óptima o de una buena solución. La fuente de éxito del PSO es que las partículas mantienen comunicación entre ellas y de esta manera la partícula con la mejor posible posición atrae a las demás a la zona más prometidora.

- Optimización por Colonia de Abejas

La optimización por colonia artificial de abejas fue propuesta en 2005 por Karagoba[7] Este algoritmo está basado en el comportamiento de las colonias de abejas. En estas colonias, las abejas productoras de miel o abejas melíferas buscan encontrar las mejores fuentes de néctar. Para llevar a cabo la búsqueda de manera eficiente las abejas están especializadas para llevar a cabo tareas específicas. En el modelo del algoritmo de colonia artificial de abejas existen tres tipos de abejas: de trabajo, exploradoras y observadoras. Cada una de las abejas puede encontrar el néctar y dar un estimado de su calidad y lo más importante puede compartir su información con otras abejas con el fin de encontrar los sitios más ricos en néctar. En el contexto de una optimización, una solución de prueba  $\mathbf{X}$  es la fuente de néctar y la función evaluada en esa solución  $f(\mathbf{X})$  es la calidad del néctar. Lo cual significa que los valores más bajos en  $f(\mathbf{X})$  representan una mejor calidad en el néctar (o en la solución).

Los algoritmos anteriores han sido exitosamente aplicados en la química. Algunas de las aplicaciones se presentarán en la sección siguiente.

## 2. Algoritmos Genéticos y Evolutivos

Este conjunto de algoritmos basan sus principios en los procesos de la evolución natural. Existen diferentes tipos de algoritmos evolutivos. La diferencia entre cada uno de ellos radica en la forma de las representaciones o en las formas de codificación, en los esquemas de selección y operadores de búsqueda. Generalmente un algoritmo genético (AG) utiliza un lenguaje binario para la representación de las soluciones mientras que algunas estrategias evolutivas utilizan valores siempre los valores numéricos reales de la solución.

La principal característica que distingue a estos algoritmos de los otros algoritmos computacionales de búsqueda local, es que los AG se basan en una población inicial que representa un conjunto de soluciones del problema y que existe comunicación e intercambio de información

---

entre los individuos de la población lo cual se lleva a cabo con operadores genéticos: *mutación* y *cruzamiento*. Estos algoritmos son altamente eficientes en la optimización de funciones debido a que conforme crece la población el algoritmo comienza a aprender y a reconocer cuáles son las zonas más prometedoras de todo el espacio de búsqueda, esto es, las zonas donde se encuentran los valores mínimos (o máximos) de la función que se está optimizando. Debido a esto las estrategias evolutivas han sido adoptadas para su uso en el entrenamiento de algunos *Perceptrones* que representan la unidad más simple de una red neuronal artificial.

### 3. Algoritmo del Recocido Simulado

El recocido se refiere al templado de ciertas aleaciones metálicas, cristales o vidrios calentándolos por arriba de su punto de fusión, manteniendo su temperatura y luego enfriándolo de una manera lenta hasta que se solidifique en una estructura cristalina perfecta. Tal proceso fisicoquímico produce materiales de alta calidad. El algoritmo del recocido simulado es una técnica heurística de búsqueda aleatoria, introducida por Kirkpatrick y colaboradores [8] para encontrar soluciones aproximadas a problemas de optimización de carácter combinatorio. Se entrará en más detalle de este algoritmo en capítulos posteriores.

## 2.2. Predicción de estructuras químicas

En capítulos anteriores revisamos las diferentes aproximaciones que existen para abordar un problema de optimización. Las llamadas metodologías estocásticas y las metodologías deterministas. Todo en la dirección de obtener conocimiento sobre las dificultades que implica predecir estructuras químicas solamente con la información de la composición del sistema, esto es, el tipo de átomos o moléculas involucradas, pero ¿cuál es el significado de la palabra “predecir”? ¿y por qué es tan importante predecir estructuras químicas? La palabra *predecir* lleva por significado “acción de anunciar un hecho futuro”. En este sentido, es natural pensar que el descubrimiento de nuevos hechos sorprendentes genera una gran motivación en la investigación científica. Por ejemplo, poder predecir con una alta precisión el pronóstico del tiempo es uno de los objetivos principales de los meteorólogos. Pero la dificultad está en la aleatoriedad que presenta la naturaleza, su nulo determinismo nos limita a sólo poder hacer una simple predicción, a pesar que la mayoría de las veces es acertado.

En química las cosas no son muy diferentes. Las herramientas deterministas han sido empleadas exitosamente por muchos años, pero en la predicción del mejor ordenamiento que adoptan los átomos en el espacio, las cosas suelen ser más que complicadas. Debido a que la estructura más estable de un cúmulo atómico o molecular es frecuentemente la geometría que presenta la menor energía potencial, la localización de esta geometría es claramente de gran interés para los químicos experimentales y teóricos. Llevar a cabo este tipo de predicciones es de gran utilidad para el diseño de nuevos materiales y por varios años esta tarea se pensó imposible de realizar inclusive con un alto poder de cómputo, sin embargo, en años recientes ha sido probado que con una metodología confiable y un buen algoritmo de búsqueda heurístico esta tarea puede llevarse a cabo de una manera exitosa [9]. Aunado a la predicción de estructuras, la implementación de restricciones espaciales resulta de gran importancia para abordar diferentes problemas de interés químico, como imponer la periodicidad de un sólido para la búsqueda de isómeros en fase cristalina, confinar al sistema dentro de una esfera para generar interacciones que libremente no se presentarían y finalmente para generar una búsqueda de isómeros sobre la superficie de un catalizador, lo cual es esencial dentro del área de catálisis [10].



---

### 2.2.1. Optimización molecular

Hemos discutido sobre la gran tarea que requiere encontrar la estructura de mínima energía para una geometría molecular dada. Esto se debe a la complejidad y el esfuerzo computacional que se requiere para hacer los cálculos de la energía. La forma tradicional de abordar este problema es haciendo uso del método variacional en donde con una función de prueba buscamos hacer variaciones en sus parámetros hasta encontrar la energía más baja. La energía total del sistema está dada por la energía electrónica del sistema más la interacción que existe entre los núcleos.

$$E = E_{elec} + \sum_{A=1}^M \sum_{B>A}^M \frac{Z_A Z_B}{\vec{R}_{AB}} \quad (2.2)$$

La energía electrónica la obtenemos haciendo uso del método variacional

$$E_{elec}[\Psi] = \frac{\langle \Psi | \hat{H} | \Psi \rangle}{\langle \Psi | \Psi \rangle} \geq E_0 \quad (2.3)$$

Y al sustituir la ecuación (2.3) en la ecuación (2.2)

$$E = \langle \Psi | \hat{H} | \Psi \rangle + \sum_{A=1}^M \sum_{B>A}^M \frac{Z_A Z_B}{\vec{R}_{AB}}. \quad (2.4)$$

Donde

$$\hat{H} = -\frac{1}{2} \sum_{\alpha}^A \frac{1}{M_{\alpha}} \nabla_{\alpha}^2 - \frac{1}{2} \sum_i^N \nabla_i^2 + \sum_i^N \sum_{j>i}^N \frac{1}{r_{ij}} + \sum_i^N v(\vec{r}_i). \quad (2.5)$$

La ecuación (2.4) es de suma importancia ya que indica que de existir cambios en las posiciones de los núcleos de una molécula, habrá cambios en la energía. Estos cambios energéticos debido a los reordenamientos en las posiciones nucleares están contenidos en la Superficie de Energía Potencial. Debido a nuestro interés en encontrar los puntos críticos de dicha superficie necesitamos la condición en donde el gradiente de la energía tome el valor de cero. Esto es

$$\nabla_i E = 0 \quad (2.6)$$

Lo cual significa que estaremos evaluando las derivadas en las coordenadas  $(X_i, Y_i, Z_i)$  para cada núcleo

$$\nabla_i E = \frac{\partial E}{\partial X_i} + \frac{\partial E}{\partial Y_i} + \frac{\partial E}{\partial Z_i} \quad (2.7)$$

Como la expresión de la energía está dada por la ecuación (2.4) se requieren las derivadas de esta cantidad con respecto a la posición de los centros. En este sentido, encontrar la geometría óptima significa encontrar la estructura molecular con la energía más baja dentro de toda la Superficie de Energía Potencial. Sabiendo esto, surge la pregunta de dónde está esa gran complejidad en el cálculo de la energía que requerimos de usar aproximaciones estocásticas. La complejidad se encuentra al minimizar el número de parámetros con los que se escribe la función de onda. Donde el verdadero esfuerzo computacional se encuentra en el tercer término de la ecuación (2.5) debido a que está relacionado con las integrales de repulsión electrónica cuya evaluación requiere de un tiempo de ejecución  $O(n^4)$ .

## 2.2.2. Sistemas finitos

Imaginemos que estamos en la situación de querer encontrar la estructura más estable para un cierto compuesto conformado por un átomo de cobre Cu y dos de cloro Cl. En la sección (1) se mencionó que para llevar a cabo un cálculo de estructura electrónica se requiere de una geometría inicial, dicha geometría puede en ciertas ocasiones venir de evidencia experimental. La dificultad entonces radica en las situaciones en donde no contamos con información experimental previa, de ser este el caso podemos proponer geometrías iniciales y hacer los cálculos pertinentes para la optimización. Es claro que debido al determinismo del método si nuestra propuesta es la geometría **A** llegaremos de manera unívoca a la estructura **C**. Por otro lado si la propuesta inicial es la estructura **B** llegaremos a la geometría dada por **D** según la figura (2.5). Al haber finalizado nuestros cálculos basta con comparar las energías resultantes de las dos optimizaciones y quedarnos con el isómero que presente la energía más profunda. Conforme a la figura (2.5) le corresponde a la estructura **D**. Aún habiendo obtenido este resultado exitoso, nada garantiza que hayamos encontrado el mínimo global de la superficie.

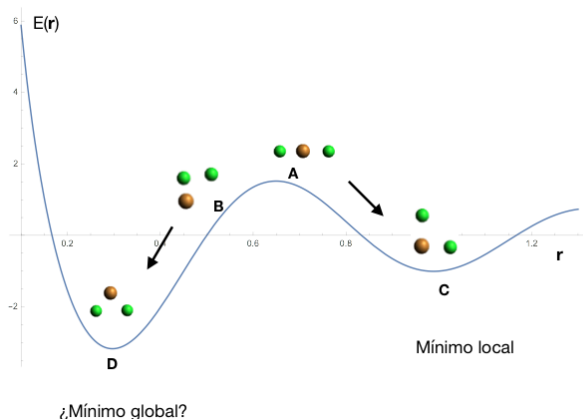


Figura 2.5: Esquematización en dos dimensiones de una Superficie de Energía Potencial

Quizás, proponiendo una geometría inicial diferente a **A** y **B** lleguemos a una estructura **G** que tenga un valor aún más bajo en energía que el encontrado previamente. Naturalmente, para cada propuesta que hagamos en la disposición de los átomos en el espacio el valor del segundo miembro de la ecuación (2.4) cambia, y con ello el valor de la energía total del sistema. Esto significa evaluar el término de las integrales de repulsión electrónica para cada propuesta en las posiciones iniciales de los átomos involucrados. Esta combinatoria requiere de un tiempo  $O(c^n)$  donde  $n$  es el número de átomos o moléculas del sistema que se traduce en un esfuerzo computacional inmenso.

Diferentes tipos de algoritmos estocásticos han sido utilizados para la predicción de estructuras químicas en fase gas. Zhang y colaboradores [11] hicieron uso del algoritmo basado en la colonia artificial de abejas para encontrar los mínimos globales (MG) de cuatro clusters de carácter no polar, mostrados en la figura (2.6)

En todos los los casos sus resultados concuerdan con los los trabajos teóricos y experimentales desarrollados anteriormene. Todos los ejemplos mencionados anteriormente se refieren sólo a sistemas en fase gas y es fácil pensar que el esfuerzo computacional incrementa conforme crece el tamaño del sistema de estudio. Sin embargo estudiar sistemas en fase gas resulta ser una tarea menos complicada que cuando se trata de estudios en materia condensada donde existen más parámetros a

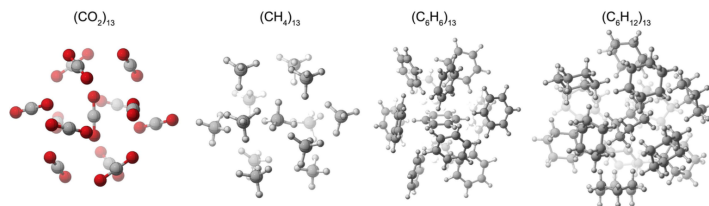


Figura 2.6: MG encontrados mediante el algoritmo ABC (Artificial Bee Colony).

considerar en la búsqueda del isómero más estable.

### 2.2.3. Sistemas periódicos

Cuando estamos tratando con sistemas periódicos la complejidad de los cálculos computacionales incrementa drásticamente, debido a que seis parámetros más para optimizar son incluidos, los llamados parámetros de celda. Afortunadamente los cálculos de estructura electrónica para sólidos pueden simplificarse teniendo como referencia la unidad mínima que representa a todo el sólido, esta es la celda unidad o celda unitaria. Esta celda repetida periódicamente en las tres dimensiones construye la representación de un sólido. Empecemos por describir qué es una malla, una estructura cristalina y cuál es la importancia de estudiarla teóricamente.

Un cristal ideal presenta una periodicidad perfecta. Tal periodicidad se puede describir en términos de redes o mallas. Matemáticamente una red es un conjunto infinito de puntos definidos por medio de sumas enteras de vectores que producen lo que conocemos como celda unidad.

$$\vec{R}[n_1, n_2, n_3] = n_1\vec{a} + n_2\vec{b} + n_3\vec{c} \quad (2.8)$$

Por otro lado, un cristal real puede ser considerado solamente como una sección finita de un cristal ideal. En un cristal los átomos están unidos de tal manera que forman un orden periódico en tres dimensiones. Este ordenamiento espacial que toman los átomos es lo que conocemos como estructura del cristal. Las posiciones que toman los átomos en la malla se les conoce como vectores de la base. Cuando tenemos tanto los vectores de la malla como los vectores de la base lo que producimos es una malla cristalina como se muestra en la figura (2.7). Lo cual cae dentro de la definición que está presente en la mayoría de los libros de texto.

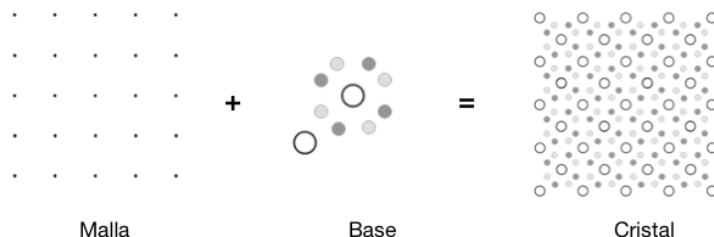


Figura 2.7: Esquematización en dos dimensiones de la representación de un cristal

Para representar la malla, usualmente se utilizan parámetros de celda (o malla) los cuales son las longitudes de las aristas que delimitan un paralelepipedo ( $a, b, c$ ) y los ángulos ( $\alpha, \beta, \gamma$ ) que

forman. Una representación de este paralelepípedo se muestra en la figura (4.3). De la definición anterior es claro que para poder predecir la estructura de un cristal requerimos de las posiciones de los átomos dentro de la malla y los parámetros de celda. Desafortunadamente este espacio de búsqueda no es sencillo. Para darnos cuenta de la gran complejidad que implica la predicción de cristales imaginemos una celda cúbica de volumen  $V$ , la cual contiene  $N$  átomos idénticos y para simplificar las cosas supongamos que los átomos pueden tomar sólo posiciones discretas sobre los nodos de una malla con un espaciado entre nodos de tamaño  $\delta$ . Hacer esta discretización hace el número de combinaciones  $C$  finito como lo muestra la ecuación (2.9).

$$C = \frac{1}{V/\delta^3} \frac{(V/\delta^3)!}{[(V/\delta^3) - N]!N!} \quad (2.9)$$

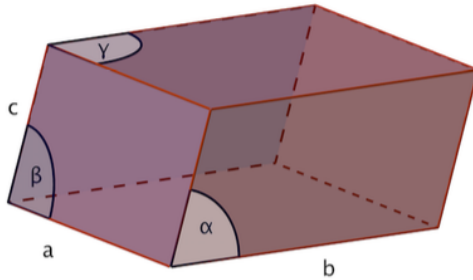


Figura 2.8: Paralelepípedo que representa una celda unitaria

Si elegimos  $\delta = 1,0\text{\AA}$  el valor promedio de longitud de un enlace químico. El valor que toma  $C$  podría darnos un estimado del número de mínimos locales. Si existen más de un tipo de átomos el número de isómeros incrementa drásticamente. Suponiendo un volumen para la celda de aproximadamente  $10 \text{\AA}^3$  y tomando en cuenta la aproximación de Stirling ( $n! \approx (n/e)^n \sqrt{2\pi n}$ ) el número de posibles estructuras para un elemento A con 10 átomos en la celda unidad es de  $10^{11}$ , para un sistema con 20 átomos en la celda unidad el número de estructuras es del orden de  $10^{25}$  y  $10^{39}$  para 30 átomos en la celda unidad. Este estimado en la cantidad de mínimos globales que puede presentar un cristal demuestra la gran complejidad del problema y que evidentemente no se puede abordar con metodologías de carácter determinístico.

En ocasiones puede resultar difícil predecir estructuras cristalinas desde cero inclusive para los métodos estocásticos. Por este motivo es conveniente utilizar la simetría como herramienta. La definición de cristal dada anteriormente no cubre la característica más importante de los cristales: la simetría. En general un cristal tiene más simetría que sólo la simetría traslacional (excepto por los sistemas triclinicos que sólo poseen este tipo de simetría). Las simetrías adicionales pueden ser rotaciones, inversiones, reflexiones dadas por planos entre muchas otras.

$$\text{Cristal} = \text{Malla} + (\text{Motivo}^2 + \text{Simetría})$$

En la figura (2.9) se muestra cómo se genera la estructura de un cristal para el sistema  $\text{Al}_{12}\text{W}$ , donde es claro ver que sólo necesitamos las posiciones de un sólo átomo de Al y un sólo átomo de W para generar todo el sólido. Mientras que en la figura (2.10) ilustra la manera de generar la misma estructura de  $\text{Al}_{12}\text{W}$  con la primera definición de cristal.

<sup>2</sup>Con motivo nos referimos a las posiciones de los átomos que toman en la malla

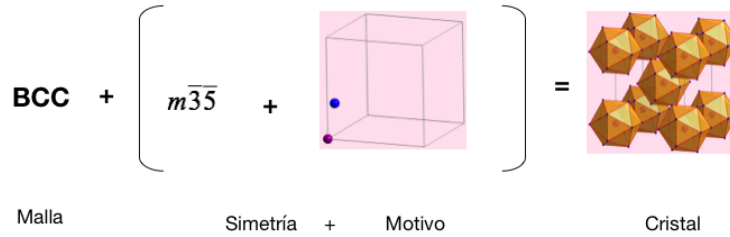


Figura 2.9: Cristal formado a partir de operaciones de simetría

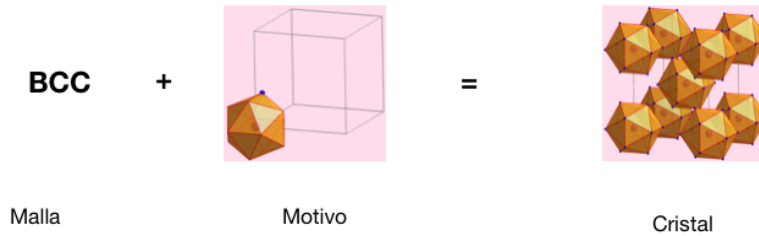


Figura 2.10: Cristal formado a partir de la malla y de las posiciones atómicas

Es notorio que en la cantidad de átomos necesarios para generar el cristal de  $\text{Al}_{12}\text{W}$  en la figura anterior es mayor que en la figura (2.9). Por lo cual hacer uso de los grupos espaciales, que son los que contienen la información de la simetría es de suma importancia en el estudio de sistemas periódicos.

En la parte de los métodos estocásticos el científico ruso Artem R. Oganov con su código de carácter evolutivo USPEX (Universal Structure Predictor: Evolutionary Xtallography) hace uso de la simetría para hacer una exploración más precisa de la Súperficie de Energía Potencial en sólidos. La figura (2.11) muestra la fase iónica de boro elemental a altas presiones encontrada por Oganov y colaboradores haciendo uso de su código USPEX[12].

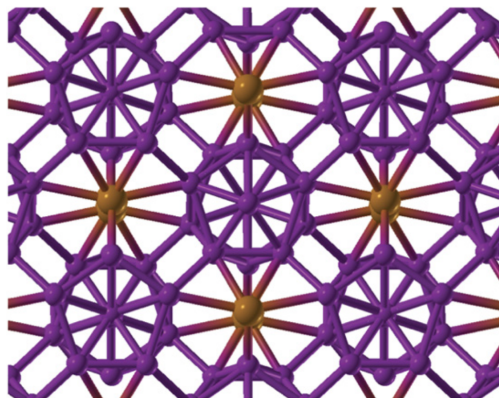


Figura 2.11: Fase iónica de boro elemental a altas presiones

Esta es la primera vez que se han visto enlaces de tipo iónico en la fase sólida de un sólo elemento. La química a altas presiones es una química totalmente diferente. En otro de sus trabajos Artem

---

R. Oganov [13] muestra que sometiendo a una presión aproximada de 200 GPa el sodio metálico comienza a tomar una fase óptica transparente (2.12). No está demás mencionar que el gran éxito del código USPEX se debe a que explota con gran eficiencia la simetría que presentan los grupos espaciales en estos sistemas periódicos. Por otra parte Schön [14] haciendo uso del algoritmo del recocido simulado ha logrado obtener resultados acordes a los experimentales para compuestos de nitruros de boro.

Sin importar el método estocástico que se utilice, se debe tener presente que las soluciones obtenidas (estructuras cristalinas o cúmulos moleculares) son simplemente candidatos óptimos para alcanzar el mínimo global. Siempre es necesario completar la búsqueda global con un algoritmo de búsqueda local. Este refinamiento en el conjunto de soluciones permite hacer una metodología más confiable.

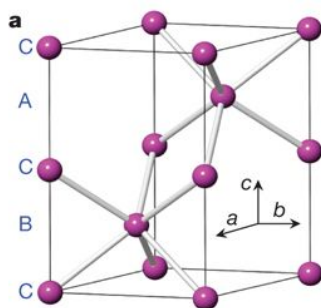


Figura 2.12: Celda unitaria para la estructura cristalina del Na-hP4.

La constante lucha por implementar y encontrar el mejor algoritmo de carácter heurístico permite que la comunidad científica se involucre en el desarrollo de los mismos, permitiendo que la química teórica tome nuevos rumbos.

### 2.3. El lenguaje de programación

Debido a que nuestro objetivo es la implementación de un algoritmo heurístico escoger un lenguaje de programación que sea versátil es de suma importancia para el buen funcionamiento en las distintas plataformas: MacOS, GNU/Linux.

El código desarrollado para este proyecto de maestría fue programado en lenguaje ANSI C para una plena compatibilidad entre diferentes sistemas, aunque no quiere decir que no existan otros lenguajes de programación aptos para desarrollar este tipo de algoritmos heurísticos. Los principales códigos heurísticos hechos con el propósito de predecir estructuras químicas están implementados en diferentes lenguajes de programación

- USPEX: Desarrollado en MATLAB con la finalidad de predecir estructuras en fase cristalina.
- XtalOPT: Desarrollado en C++ con la finalidad de predecir estructuras en fase cristalina.
- ASCEC: Desarrollado en Fortran con la finalidad de predecir estructuras en fase gas.

Cada lenguaje aporta una cierta dificultad en cuanto a la implementación y al manejo de los datos requeridos por el programa. En este sentido puede haber ciertas ventajas en cuanto a la elección del lenguaje de programación. En el desarrollo de algoritmos heurísticos es esencial el manejo

---

de operaciones que involucran productos entre matrices, además debido al interés en estudiar sistemas químicos se necesita el uso de arreglos de gran tamaño para guardar N cantidad de átomos o fragmentos moleculares. El paquete MATLAB (Matrix Laboratory) es de gran ayuda para las operaciones matriciales implícitas en los algoritmos heurísticos aunque es necesario comprar una licencia para su uso. Por el lado “Open Source” tenemos la opción de implementar todo por nuestra propia cuenta haciendo uso de lenguajes de programación como: Java, C, C++ o Fortran. El uso de los lenguajes orientados a objetos no ha sido explotado por completo. El lenguaje C++ aunque no es del todo orientado a objetos es muy utilizado en el desarrollo de códigos de carácter estocástico.

Una vez seleccionado el lenguaje de programación es necesario saber a qué tipo de problema nos estamos enfrentando y con qué tipo de datos estaremos lidiando durante el desarrollo del código. Para ello existen un conjunto de elementos que son útiles para organizar y almacenar datos llamados “Estructuras de Datos”.

### 2.3.1. Estructuras de Datos

Los datos o la información de un programa deben de organizarse de una manera eficiente y que aporte la mayor simplicidad, para ello se hace uso de las estructuras de datos. Las estructuras de datos son el arreglo lógico o matemático de datos en memoria, las cuales no sólo consideran el almacenamiento físico, consideran las relaciones entre los datos y las operaciones que pueden realizarse sobre ellos. La elección apropiada de estructuras de datos en conjunto con buenos algoritmos es un paso fundamental en el diseño de programas eficientes.

Las estructuras de datos se dividen en dos categorías: primitivas y no primitivas. La primera categoría corresponde a los tipos de dato propios del lenguaje de programación por ejemplo: tipo entero, tipo real y tipo carácter. La segunda corresponde a los tipos de datos creados por el usuario haciendo uso de los datos de tipo primitivo, también se les conoce como “datos de tipo usuario”. Dentro de los tipos de datos definidos por el usuario tenemos las estructuras de datos lineales y no lineales, las primeras como su nombre lo indica hacen una alocaión de memoria lineal, es decir en sectores contiguos de memoria (listas ligadas, arreglos pilas) por el otro lado las estructuras de datos no lineales las alocaiones de memoria se hacen de manera no consecutiva (árboles binarios y grafos). Existen diversos tipos de estructuras de datos pero los utilizados para el desarrollo de nuestro código hacemos uso de los “structs” un tipo de estructura de datos de tipo lineal.

#### Arreglos vs estructuras en C

Una estructura es similar a un registro, debido a que almacena información relacionada perteneciente a una entidad. Una estructura es básicamente un tipo de dato definido por el usuario que puede guardar información de varios tipos de datos.

Al declarar un arreglo podemos hacerlo para un número N de elementos pero sólo de un tipo, en cambio haciendo uso de los “structs” podremos declarar un arreglo de tipo N con la posibilidad de declarar diferentes tipos de datos: enteros, reales, caracteres o inclusive de tipo estructura, a esto último se le conoce como estructuras de datos anidadas.

---

Declaración de estructuras:

---

```
struct Empleado
{
    int numEmpleado;
    double salarioMensual;
    char name[180];
};
```

---

El código mostrado anteriormente es un clásico ejemplo de la declaración y la utilidad de una estructura en el lenguaje C, donde se puede notar claramente la sencilla manera de agrupar los datos para los datos propios de un empleado: nombre, número de empleado y salario mensual. Vale la pena mencionar que al definir un estructura no estamos consiguiendo una alocaión de memoria simplemente es una declaración de un tipo de dato. El siguiente código muestra cómo reservar la memoria para una estructura

---

```
int main ()
{
    struct Empleado emp;

    return 0;
}
```

---

La verdadera utilidad de las estructuras radica en hacer arreglos de éstas mismas y anidarlas entre ellas para poder manejar una gran cantidad de datos altamente correlacionados.

---

```
struct Empleado
{
    int numEmpleado;
    double salarioMensual;
    char name[180];
};
struct Departamento
{
    char nombreDep[180];
    struct Empleado emp[120];
};
int main ()
{
    struct Departamento dep[20];
}
```

---

El código anterior muestra la manera en que se anidan dos arreglos de estructuras con diferentes tipos de datos. Para ello tomamos un ejemplo de una tienda departamental que tiene 20 departamentos y con 120 empleados en cada uno de ellos los cuales tienen los atributos de número de empleado, salario mensual y nombre. Con esto podemos generar una manera muy eficiente para organizar los distintos tipos de datos.

Para acceder a los elementos de una estructura es necesario hacer uso del operador punto (.). Por ejemplo si después de almacenar los datos queremos acceder a la información de un empleado en un departamento en específico debemos hacer lo siguiente:



---

```

void imprime resultado ()
{
    int i=0;
    printf(‘Departamento: %s Empleado: %s Id: %d\n’,dep[i].nombreDep,
        dep[i].emp[i].nombre, dep[i].emp[i].numEmpleado);
}

```

OUTPUT:

Departamento: Electronica, Empleado: Juan Garcia, Id: 2018

---

En el ejemplo anterior es evidente que el uso de las estructuras en el lenguaje C es de gran utilidad para almacenar diferentes tipos de datos. Las estructuras las podemos utilizar para guardar datos de moléculas donde son necesarios diferentes tipos de datos como el nombre de los átomos que es de tipo caracter, su número atómico del tipo entero, y sus coordenadas cartesianas que son del tipo doble precisión.

---

```

struct Atom
{
    double x,y,z;
    int atomicN;
    char symbol[3];
};
struct Mol
{
    struct Atom *atm;
};
int main ()
{
    struct Mol *m;
    m = (struct Mol *)malloc(sizeof(struct Mol)*N);
    for(int i=0; i<N; i++)
    {
        m[i].atm = (struct Atom *)malloc(sizeof(struct Atom)*tm[i].natom);
        for(int j=0; j<m[i].natom; j++)
        {
            scanf(‘\%s %lf %lf %lf ’,m[i].atm.symbol, m[i].atm[j].x, m[i].atm[j].y,
                m[i].atm[j].y);
        }
    }
}

```

---

En el ejemplo anterior hemos definido una estructura que almacena la información de N moléculas con diferente cantidad de átomos cada una. Así como los datos de los respectivos símbolos químicos y los números atómicos para los diferentes átomos contenidos en las moléculas.

## Capítulo 3

# Metodología

*“La heurística es un algoritmo en un traje de payaso. Es menos predecible, es más divertido y viene sin una devolución de dinero de 30 días garantizada”*

---

*Steve McConnell*

Dentro del gran compendio de técnicas heurísticas bio-inspiradas que existen para este proyecto de maestría fue elegido el algoritmo del recocido simulado. Un algoritmo de naturaleza estocástica que trata de emular el templado que sufren los sólidos al someterlos a baños térmicos. Esta técnica ha sido utilizada para la optimizaciones de funciones, teniendo un gran éxito principalmente en las áreas de matemáticas y computación, sin embargo su uso en la química no es ajeno ni tampoco reciente, puesto que encontrar el mejor acomodo de los átomos en el espacio para un ente molecular es un problema que ha persistido por muchos años. El desarrollo de este capítulo se enfoca en dar a detalle la implementación del algoritmo del recocido simulado para la búsqueda de isómeros restringidos espacialmente el cual culmina en la creación del código RISSA presentado en capítulo siguiente. Una ligera variación en la naturaleza del algoritmo del recocido simulado permite que el código RISSA explore de una manera exhaustiva la superficie de energía potencial para sistemas moleculares de gran tamaño. Es preciso explicar en qué consiste el recocido de un sólido antes de comenzar con la minuciosa descripción del algoritmo y de cómo éste converge a un conjunto de soluciones aceptables en un tiempo polinomial.

El recocido de un sólido o templado consiste en aumentar la temperatura del metal por arriba de su punto de fusión para después disminuir la temperatura de una manera lenta y conseguir una estructura cristalina perfecta. Al último paso se le conoce como enfriamiento, el cual es una pieza clave para el buen funcionamiento del método, debido a que los decrementos bruscos en la temperatura generan estructuras quebradizas e inestables. En su artículo, Kirkpatrick y colaboradores demuestran que existe una profunda y muy útil conexión entre la termodinámica estadística y las optimizaciones combinatorias. La analogía con el recocido de los sólidos permite establecer una relación con los conceptos termodinámicos y los conceptos propios de las optimizaciones combinatorias.

- Energía  $\rightarrow$  Función de costo o función objetivo  $f(i)$ .
- Cambio de estado  $\rightarrow$  Solución vecina  $f(i + 1)$ .
- Estado de baja energía  $\rightarrow$  Solución óptima  $f(i_{opt})$ .
- Temperatura  $\rightarrow$  Parámetro de control  $C_T$ .

---

### 3.1. El algoritmo del Recocido Simulado

El algoritmo del recocido simulado es una técnica heurística de búsqueda propiamente de carácter local, puesto que hace una secuencia de búsquedas aleatorias en las vecindades de un punto elegido aleatoriamente en un espacio  $S$  de soluciones, como se muestra en la figura (3.1). Sin embargo, puede considerarse a este algoritmo como una variante de la búsqueda local, puesto que permite movimientos ascendentes que evitan quedar atrapado prematuramente en un mínimo local.

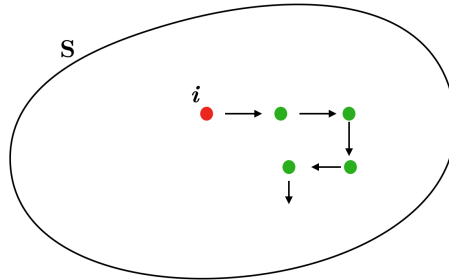


Figura 3.1: Malla de búsqueda local en el algoritmo del recocido simulado.

En 1953 Metropolis y Teller [15] presentaron un algoritmo para simular la evolución que lleva al equilibrio de un sólido sometido a un baño térmico. El algoritmo presentado por Metropolis y colaboradores se basa en el método de Monte Carlo [16] en el cual se generan una secuencia de estados en el sólido. El algoritmo consiste en comenzar con un estado inicial del sólido denotado como  $i$ , con una energía asociada  $E_i$ , entonces un estado subsecuente  $j$  se genera debido a una pequeña distorsión, la cual puede ser por ejemplo, un desplazamiento en la partícula. La energía de este nuevo estado estará denotada por  $E_j$ . Si la diferencia de energía  $E_j - E_i$  es menor o igual a 0, el nuevo estado  $j$  será aceptado como nuevo punto de partida. En caso contrario si la diferencia de energía es mayor a 0, el estado  $j$  será aceptado con una cierta probabilidad  $P$  dada por

$$P(X_i) = \exp\left(\frac{-\Delta E}{T}\right) \quad (3.1)$$

Donde identificamos a  $T$  como el parámetro de control inicial del baño térmico al cual sometemos al sólido. La regla de aceptación mencionada anteriormente se conoce como criterio de Metropolis y el algoritmo que conduce a este criterio de aceptación es el conocido algoritmo de Metropolis. Es importante mencionar que en la ecuación (3.1)  $T$  no representa la temperatura termodinámica del sistema, simplemente es un parámetro con el cual podremos controlar los descensos para acercarnos a las soluciones óptimas. De igual forma el  $\Delta E$  sólo representa las diferencias entre valores evaluados de la función objetivo que está siendo optimizada por el algoritmo. Sabiendo esto, la ecuación (3.1) puede escribirse de una manera general

$$P(X_i) = \exp\left(\frac{-\Delta f}{T}\right) \quad (3.2)$$

Como se mencionó anteriormente una de las claves para alcanzar una estructura o un estado de menor energía es decrementar la temperatura lentamente, así el sólido alcanzará el equilibrio térmico en cada paso. Dentro del algoritmo de Metropolis, esto se logra generando un gran número de transiciones aleatorias a un valor de temperatura.

---

El motivo principal por el cual el recocido simulado fue seleccionado como el algoritmo de búsqueda local para este proyecto es debido a que tiene la capacidad de escapar de mínimos locales. Esta idea debe usarse cuidadosamente ya que el concepto de “escapar” de mínimos locales recae en el criterio de aceptación y rechazo del algoritmo de Metropolis y en los valores del parámetro de control  $C_T$ . Esto se debe a que en cada decremento en el parámetro de control el algoritmo acepta soluciones donde el valor de la función objetivo aumenta (ecuación 3.2); pero a medida que la temperatura desciende sólo algunas desviaciones serán aceptadas y finalmente cuando el parámetro de control se aproxime a cero, no se aceptarán más desviaciones y sólo serán aceptadas aquellas configuraciones donde la función objetivo disminuye su valor. De ahora en adelante podemos ver al algoritmo del recocido simulado como una repetición de muchos algoritmos de Metropolis, uno por cada decremento de temperatura.

La parte medular del algoritmo del recocido simulado consta de dos ciclos

- Un ciclo externo **Do...while**
- Un ciclo interno **For...**

El ciclo interno mantiene fijo el parámetro de control mientras genera las  $N_k$  soluciones que el algoritmo de Metropolis acepta o rechaza. El ciclo externo disminuye el parámetro de control y termina cuando la condición de paro se cumple ( $C_T \approx 0$ ). Una de las claves para que el algoritmo del recocido simulado nos lleve a un conjunto de buenas soluciones es la elección de una buena secuencia de enfriamiento. Existen diferentes secuencias de enfriamiento, las cuales dependen de parámetros que las hacen más eficientes o menos eficientes.

Algunos de los métodos de enfriamiento más utilizados son

- Enfriamiento logarítmico

$$T_k = \frac{\alpha T_0}{\ln(1+k)} \quad (3.3)$$

En este caso  $\alpha$  es el parámetro de velocidad de enfriamiento. Este método de enfriamiento ha sido probado para garantizar una convergencia a un buen conjunto de soluciones utilizando el valor de  $\alpha = 1$ .

- Enfriamiento geométrico

$$T_k = \alpha^k T_0 \quad (3.4)$$

En este método, el valor óptimo del parámetro  $\alpha$  es cuando es pequeño, pero cercano a 1. Los valores usuales de  $\alpha$  para este tipo de enfriamiento están entre 0,88 y 0,99. El poner en nuestro algoritmo valores muy pequeños de  $\alpha$  puede resultar en un enfriamiento muy rápido.

- Enfriamiento exponencial

$$T_k = T_0 e^{-\alpha k/N} \quad (3.5)$$

Este tipo de enfriamiento es el más rápido durante las primeras iteraciones. La velocidad de decaimiento puede manipularse con el valor del parámetro  $\alpha$  y  $N$  representa la dimensionalidad del sistema[?].

### Pseudo-código para el algoritmo del recocido simulado

1. Escoger un estado inicial  $\vec{r}_0$  y Temperatura  $C_{T0}$
2. Propagar el sistema  $\vec{r}_{rand} \leftarrow \vec{r}_0 + rand()$
3. Evaluar energía  $E(\vec{r}_{rand})$
4. Calcular el cambio  $\Delta E = E_{fin}(\vec{r}_{rand}) - E_{init}(\vec{r}_0)$ 
  - Si  $\Delta E < 0$  entonces  $E_{init} \leftarrow E_{fin}$
  - De lo contrario genera:  $p \leftarrow rand() \% 1$ 
    - Si  $p < exp(\Delta E/T)$  entonces  $E_{init} \leftarrow E_{fin}$
    - De lo contrario regresa a (2)
5. Disminuye la temperatura  $T_{n+1} = T_n(1 - (T_0/N_{steps}))$
6. Repite hasta que  $T = 0$

## 3.2. Uso de estructuras

Como se mencionó en la sección (2.3.1) el uso de estructuras de datos es esencial para añadir buena funcionalidad a los algoritmos, como parte de la metodología se utilizaron estructuras, un tipo de datos creados por el usuario que permiten agrupar distintos tipos de datos en un arreglo. Para nuestro caso, al estar trabajando con moléculas es fácil hacer uso de las estructuras propias del lenguaje C para guardar sus números atómicos, símbolos químicos y la información de las posiciones atómicas. En la figura (3.2) se muestra un ejemplo de la implementación de estructuras para dos

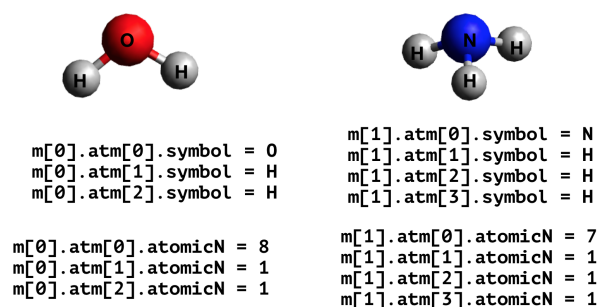


Figura 3.2: Ejemplo del uso de estructuras para dos moléculas

moléculas: una de agua con el identificador 0 y otra de amoníaco con el identificador 1 dentro del arreglo de estructuras  $m$  de tamaño  $m[2]$ . Para acceder a los miembros de las estructuras es necesario hacer uso del operador punto (.) como se mencionó anteriormente y como se muestra en la figura (3.2). Comprender el uso de este tipo de datos creados por el usuario es esencial para poder manipular, de ser necesario el código fuente del programa desarrollado para este proyecto el cual se presentará en la sección siguiente.

## Capítulo 4

# El código RISSA

*“Los malos programadores se preocupan por el código. Los buenos programadores se preocupan por las estructuras de datos y las relaciones entre ellas.”*

---

*Linus Torvalds*

El código desarrollado para este proyecto lleva por nombre RISSA (por sus siglas en inglés: Restricted Isomers Searching by Simulated Annealing). Está programado desde cero en lenguaje C y es compatible con los sistemas GNU/Linux y MacOS. Nuestro código tiene la capacidad de explorar la Superficie de Energía Potencial de átomos o moléculas de gran tamaño sometidas a tres restricciones espaciales: a) dentro de una esfera b) condiciones periódicas en tres dimensiones y c) condiciones periódicas en dos dimensiones. Las funciones objetivo que tiene implementadas el código son: 1) Lennard-Jones 2) potencial de interacción relacionado con las funciones condensadas de Fukui y 3) la energía semiempírica proveniente del software MOPAC.

Una de las características extra que incluye RISSA es la posibilidad de mantener fijo uno de los fragmentos en el centro de la geometría elegida por el usuario, así como también mantener fijas las coordenadas cartesianas provenientes de la información experimental de un fragmento si así se desea, como un ejemplo esto último tiene la finalidad de observar los mejores sitios en los que se posicionarán fragmentos en un MOF (Metal Organic Framework). En donde es claro que sólo es necesario hacer los movimientos aleatorios de los fragmentos manteniendo fijo el MOF dentro de la celda.

### 4.1. Funciones objetivo, restricciones geométricas y archivos de entrada

La elaboración de los archivos de entrada para usar el código RISSA dependen del tipo de función objetivo y del tipo de restricción espacial. Puesto que para cada combinación se requieren de algunos parámetros adicionales como por ejemplo los parámetros  $\sigma$  y  $\epsilon$  para el potencial de Lennard-Jones, los parámetros de celda si es que se desea hacer una simulación en fase sólida o las instrucciones necesarias para hacer un cálculo con MOPAC. Por ello es necesario dar una detallada descripción en cuanto a la creación de los archivos de entrada.

---

### 4.1.1. Restricciones espaciales

#### Confinamiento esférico

Como se mencionó antes, la manera en que confinemos al sistema de estudio nos permitirá simular diferentes ambientes químicos y estudiar agregados atómicos y moleculares, cristales y superficies. En el caso de un confinamiento esférico podremos forzar a que existan interacciones que el sistema libre no presentaría.

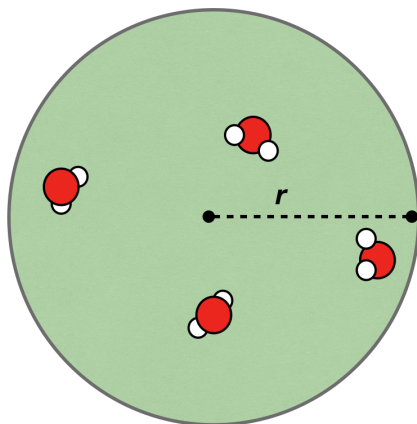


Figura 4.1: Restricción esférica en un sistema modelo de cuatro moléculas de agua

En la figura (4.1) se muestra un esquema de la restricción esférica, donde es claro que sólo es necesario que el usuario ingrese el radio en el cual quiere confinar al sistema de estudio. Este tipo de restricción espacial es muy útil para modelar sistemas en fase gas, así como también poder estudiar las interacciones entre fragmentos grandes de proteínas en donde nos interesa encontrar aquellas configuraciones en el espacio que minimizan la energía.

#### Restricción periódica en 3D

Este tipo de restricción presenta una ligera dificultad comparado con el confinamiento esférico, puesto que entran en juego otros tres parámetros, los parámetros de la celda. Dichos parámetros son requeridos por el programa y el usuario tiene que proporcionarlos en el archivo de entrada.

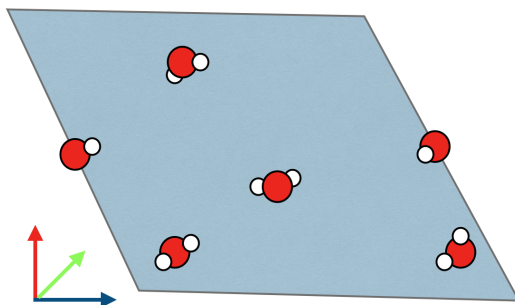


Figura 4.2: Restricción periódica en 3D para un sistema modelo de cinco moléculas de agua

Tal y como lo muestra la figura (4.3) el código RISSA puede trabajar con celdas oblicuas y con celdas ortogonales. Hacer una búsqueda de isómeros sobre condiciones periódicas en 3D tiene implicaciones que van más allá de sólo mover átomos o moléculas dentro de una celda que se repite,

como lo es el uso de la simetría de grupo espacial para acortar el espacio de búsqueda. Sin embargo para sistemas periódicos de gran tamaño en donde es importante predecir las posiciones que tomarán los diferentes fragmentos involucrados es suficiente con hacer movimientos aleatorios.

## Restricción periódica en 2D

Los estudios teóricos orientados a la catálisis son una de las aplicaciones más relevantes de la química computacional pero también de las más complicadas. Trabajar con superficies catalíticas para encontrar los mejores sitios de adsorción es un reto computacional. En este sentido debemos trabajar por ensayo y error para proponer los mejores sitios de adsorción sobre la superficie, la intuición química juega un papel fundamental para evitar propuestas que sólo nos lleven a máximos en la superficie de energía potencial, no obstante, usar la intuición humana conlleva más errores que aciertos en algunas ocasiones.

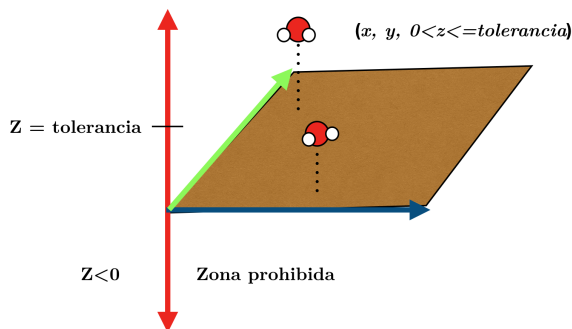


Figura 4.3: Restricción periódica en 2D

RISSA puede explorar la SEP aleatoriamente y encontrar los sitios más prometedores del adsorbato sobre la superficie que minimizan la energía. Para ello sólo son necesarios los vectores que definen la superficie y el programa se encargará de hacer aplicar las condiciones periódicas correspondientes.

### 4.1.2. Funciones objetivo

Las funciones objetivo implementadas en RISSA no trabajan bajo las mismas restricciones espaciales. Por el momento la única función objetivo que trabaja para las tres restricciones geométricas es la energía semiempírica que el software MOPAC proporciona.

La primera función objetivo implementada debido a su simplicidad es el potencial de Lennard-Jones, donde la energía del sistema viene dada por la suma de todas las interacciones a pares del potencial

$$V(r_{ij}) = 4\epsilon\left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^6\right]. \quad (4.1)$$

Para esta función objetivo basta con tener un archivo con las coordenadas cartesianas de los átomos en el sistema y sus respectivos parámetros  $\sigma$  y  $\epsilon$ . Esta función objetivo está implementada para trabajar bajo la restricción esférica y bajo condiciones periódicas en 3D.



La segunda función objetivo implementada en RISSA surge a partir de una colaboración con colegas de la Universidad Andrés Bello en Chile, en donde ellos utilizan el criterio conocido como Maximun Matching [17] de las funciones de Fukui para predecir cómo grandes agregados moleculares pueden ser formados.

$$\Delta W = - \sum_{\delta \in D}^{\#basines} \sum_{\gamma \in A}^{\#basines} \frac{f_{\delta}^{+} f_{\gamma}^{-}}{|R_{\delta} - R_{\gamma}|}, \quad (4.2)$$

donde  $f_{\delta}^{+}$  y  $f_{\gamma}^{-}$  representan a las funciones condensadas de Fukui del fragmento  $D$  y  $A$  respectivamente. Para este caso basta tener las coordenadas cartesianas de cada molécula y cada función condensada de Fukui en un archivo de entrada. En la figura (4.4) se muestran los basines que están presentes en la suma de la ecuación (4.2).

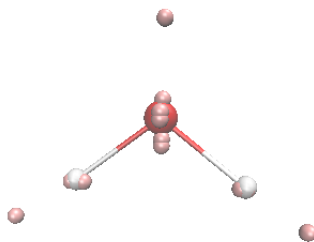


Figura 4.4: Los puntos rosas sobre la molécula de agua representan los basines

La tercera y última función objetivo, hasta el momento, es la energía proveniente de un método semiempírico, el paquete computacional MOPAC [18] proporciona el valor de la energía correspondiente con las estructuras que el código RISSA le dará como punto de partida en cada iteración. La elección de un método semiempírico para evaluar la energía de cada conformación recae en que estos métodos son computacionalmente más rápidos debido a que no evalúan las integrales bielectrónicas presentes en el hamiltoniano, a cambio de ello se utilizan parámetros experimentales que aumentan la velocidad en el cálculo. Debido a que sólo estamos interesados en la energía de cada movimiento aleatorio se realiza un cálculo *single point* (sin optimización de geometría) con MOPAC.

### 4.1.3. Archivos de entrada

Una vez descritas las restricciones espaciales y las funciones objetivo desarrolladas en RISSA es momento de explicar con cierto detalle la elaboración de los archivos de entrada que nos permitirán correr nuestras simulaciones. Pensando en el usuario final RISSA ha sido programado de una manera que la interacción humano-programa sea de lo más simple posible. De la misma manera los resultados de la simulación están organizados en un archivo que permite su fácil lectura y manipulación.

### Simulando un sistema en fase gas

Tomemos un sistema compuesto por seis moléculas de agua y un átomo de cobre el cual queremos simular usando una restricción esférica y teniendo como función objetivo la energía semiempírica. El archivo de entrada ejemplo se encuentra en la figura (4.5) Debido a que queremos ver cuáles son los acomodos en el espacio que tendrán las 6 moléculas de agua alrededor del átomo de cobre, pondremos la línea `FIXED FRAGMENT:TRUE`. Esto por default permite fijar el primer fragmento del archivo de

entrada, en este caso, el átomo de cobre. La segunda línea: `FIXED FRAGMENT:FALSE` permite usar información experimental de las posiciones atómicas dadas por el usuario y mantenerlas fijas. La primera línea son instrucciones para ejecutar el programa MOPAC, contiene información del método para calcular la energía.

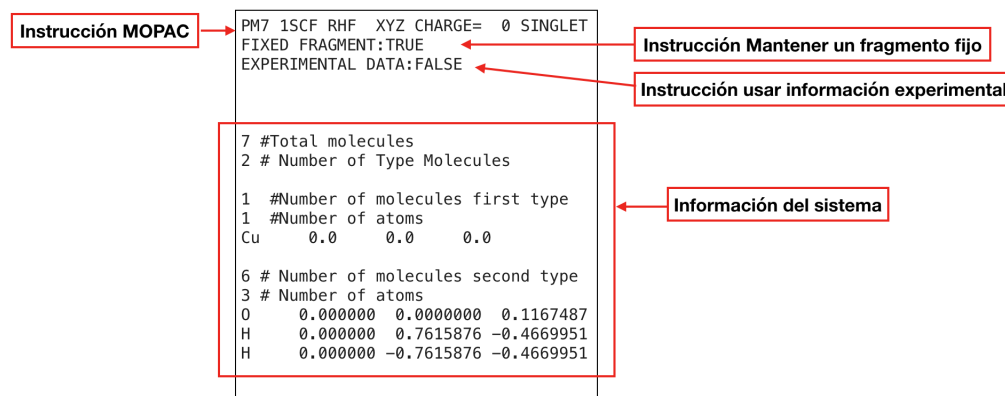


Figura 4.5: Archivo de entrada muestra para un sistema en fase gas

Por último tenemos la información de los fragmentos del sistema. El número 7 corresponde al total de moléculas o fragmentos moleculares presentes en la simulación. El número 2 hace referencia al total de fragmentos diferentes en el sistema:  $\text{H}_2\text{O}$  y Cu. Los números 1 y 6 arriba de la cuerda: `# Number of atoms`, representan la cantidad de moléculas para cada tipo de fragmento, esto es, 1 átomo de cobre y 6 moléculas de  $\text{H}_2\text{O}$ .

### Simulando un sistema en fase sólida

Supongamos que queremos encontrar el mejor acomodo en el espacio que tendrían dos átomos de oro con uno de paladio en una celda cúbica que se repite periódicamente. Para ello debemos elaborar un archivo de entrada como lo muestra la figura (4.6).

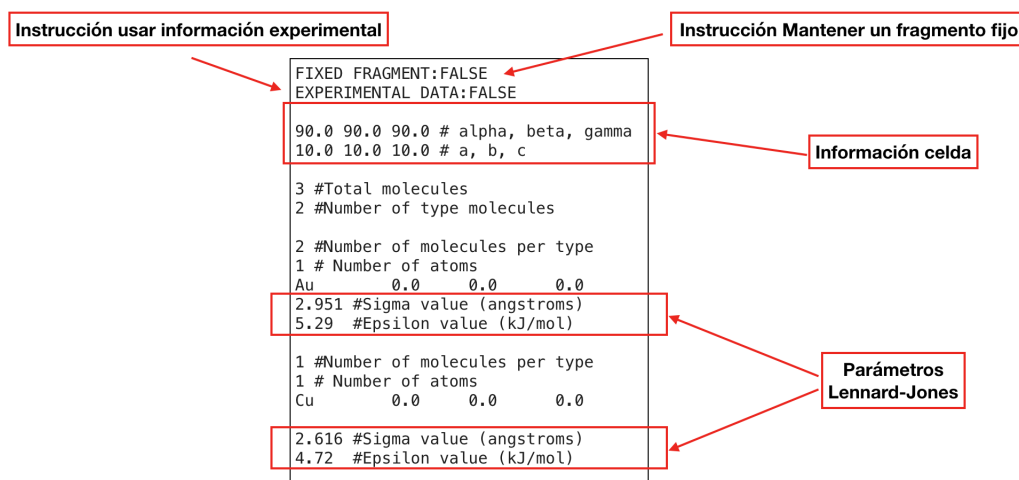


Figura 4.6: Archivo de entrada muestra para un sistema en fase sólida

La sintaxis del archivo de entrada tiene algunas variaciones en cuanto al caso esférico, puesto que ahora debemos proporcionar los parámetros de celda; en los cuales primero proporcionaremos

---

la información de los ángulos  $\alpha$ ,  $\beta$ ,  $\gamma$  y después los valores para  $a$ ,  $b$  y  $c$  respetando el orden anterior. Después de ello en la sección de la información del sistema la sintaxis tiene el mismo significado para declarar el número de moléculas/átomos y la cantidad que hay de cada uno. La única diferencia es que, debido a que estamos usando como función objetivo el potencial de interacción de Lennard-Jones necesitamos de los parámetros  $\sigma$  y  $\epsilon$  para cada átomo los cuales se ponen debajo de la información de las coordenadas cartesianas.

## 4.2. Detalles en la programación

El código RISSA fue programado en módulos para que la comprensión del código sea lo más sencilla posible. Modular el código ayuda a que el usuario agregue sus propias funciones de una manera rápida y fácil. Con un total de 5,113 líneas de código, en RISSA cada módulo está en un archivo de extensión (.c) diferente, teniendo un total de 46, puesto que cada uno de ellos desempeña una función diferente en el código. La descripción de las 46 funciones es una tarea extensa que se sale de los propósitos de un trabajo de tesis. Por ello sólo se describirán las partes que son medulares en el recocido simulado, así como una ligera descripción de algunas funciones del código que son necesarias en la comprensión de la metodología.

### 4.2.1. Movimientos aleatorios y restricciones espaciales

Dentro de las funciones más importantes del algoritmo del recocido simulado está la generación de movimientos aleatorios para explorar el conjunto de posibles soluciones de la función objetivo. En el capítulo (3) se mencionó que una de las características del recocido simulado es que maneja un tipo de búsqueda conocida como “local”, lo cual significa que busca aleatoriamente en las vecindades de alguna solución inicial aleatoria y a partir de ahí obtiene los mejores candidatos para la solución óptima. Esta aproximación es la adecuada para optimizar funciones de una o más variables y ha sido demostrada su eficiencia al ser aplicada a un gran número de problemas de optimización y recientemente para resolver el problema relacionado con el “Sudoku”[19]. Sin embargo cada problema computacional tiene características propias, lo cual implica que no siempre se pueda usar la misma lógica para cualquier problema. En nuestro caso, como se mencionó en capítulos anteriores, estamos interesados en explorar la Superficie de Energía Potencial asociada a un sistema químico y encontrar el conjunto de configuraciones subóptimas que minimizan la energía. Para ello es fácil pensar, que sólo basta con generar los movimientos aleatorios para perturbar al sistema y evaluar la función objetivo para cada estado. Sin embargo, al estar trabajando con átomos, moléculas o fragmentos moleculares muy grandes, desplazar aleatoriamente al sistema no es suficiente. Es necesario hacer uso de otras estrategias que permitan una mejor exploración de nuestra función objetivo.

En nuestro caso, hacemos tanto rotaciones como traslaciones aleatorias. Para la rotación aleatoria se le aplica la matriz de rotación ( $\vec{R}_{random}$ ) con giros aleatorios en torno a los ángulos de Euler al vector  $\vec{r}_i$  que contiene las coordenadas de cada átomo, para el traslado aleatorio de las moléculas se mueve aleatoriamente el centro de masa de las moléculas y luego se le aplica el vector de traslación a cada coordenada atómica

$$\vec{t}_{random} = \vec{r}'_{cm} - \vec{r}_{cm} \quad (4.3)$$

Donde  $\vec{r}'_{cm}$  representa el movimiento aleatorio del centro de masa. Finalmente la rotación y la traslación aleatorias para cada molécula queda representada en la siguiente ecuación

$$\vec{r}'_i = \vec{R}_{random}\vec{r}_i + \vec{t}_{random} \quad (4.4)$$

---

Para garantizar que se cumplan las restricciones espaciales y que tampoco exista traslape entre los átomos de las moléculas en el sistema se realiza lo siguiente:

1. Restricciones espaciales:

■ Caso esférico:

Se calcula la distancia de cada átomo al centro de la esfera  $r_i$  y se comprueba que  $r_i$  no sea mayor que  $r_{usuario}$  que es el valor del radio esférico ingresado por el usuario, de suceder esto se prueba con otro movimiento aleatorio.

■ Restricción periódico en 3D:

Se utilizan condiciones periódicas, lo cual significa que siempre se mantiene el número de átomos constante dentro de la celda, es decir si un átomo sale de la celda en realidad está entrando por el lado contrario. El código RISSA aplica las condiciones periódicas sobre las coordenadas relativas de los átomos presentes en el sistema, esto permite trabajar tanto con celdas no ortogonales ( $\alpha \neq \beta \neq \gamma \neq 90^\circ$ ) como ortogonales ( $\alpha = \beta = \gamma = 90^\circ$ ).

■ Restricción periódica en 2D:

Trabaja bajo el mismo concepto que las restricciones periódicas en 3D, con la diferencia de que la periodicidad sólo va a estar dada por los vectores que definen la superficie de estudio, esto es, sólo en 2D.

2. Traslapes atómicos:

Para cada restricción espacial, se calculan las distancias entre los átomos de cada una de las moléculas  $r_{ij}$  y se comprueba que esta distancia no sea menor que la suma de los radios covalentes de los átomos en cuestión.

RISSA trabaja con el método conocido como la aproximación del Big-Bang propuesto por Leary[20] para cúmulos Lennard-Jones. Este método fue aplicado por Jackson en cúmulos de  $Si_n$  ( $n=20-27$ ) y por Fuentealba para cúmulos de metales alcalinos[21]. La efectividad de este método radica en que siempre se parte del evento menos probable o con costo energético muy alto, que es tener a todas las moléculas del sistema sobrepuestas en un punto del espacio y a partir de ahí generar los movimientos aleatorios. Es evidente que por mínima que sea la perturbación que se realiza al sistema inicial, siempre se obtendrá un estado energético menor, tal y como lo muestra la figura (4.7). En nuestra aproximación, los puntos B son las soluciones que estaremos aceptando al evaluar el algoritmo de metropolis según la temperatura en la que se encuentre el ciclo externo del recocido simulado. Un problema de este procedimiento es que a medida de que el parámetro de control disminuye o se aproxima a cero, las probabilidades de aceptar malas soluciones son menores. Así que si siempre partimos de un conglomerado de moléculas a temperaturas muy bajas difícilmente podremos encontrar soluciones que mejoren la previa encontrada. El problema anterior lo solucionamos haciendo uso del concepto de un “baño frío”. Conceptualmente podemos entenderlo como un fenómeno relacionado con la termodinámica, debido a que entre más baja la temperatura de un sistema los movimientos moleculares son cada vez menores, o en el caso de nuestra simulación la generación de movimientos aleatorios se dan en un pequeño intervalo.

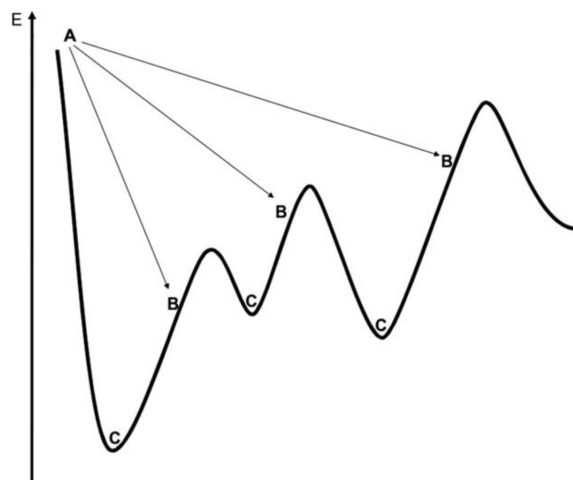


Figura 4.7: Aproximación del Big-Bang. Al inicio de cada movimiento aleatorio las moléculas parten de un mismo punto.

La generación de números aleatorios en ambos intervalos de temperatura están dados por números de precisión doble y los intervalos que toman dependen de la geometría en la que se esté trabajando.

1. Temperaturas altas ( $T_k > 0.001$ )

a) Restricción esférica:

Los números aleatorios se normalizan al radio de la esfera ingresada por el usuario. Esto es, se generan números aleatorios en el intervalo  $[-r_{user}, r_{user}]$

b) Restricción periódica en 3D:

Se generan números aleatorios en las coordenadas  $x, y, z$  normalizadas con los valores correspondientes de los parámetros  $a, b, c$  respectivamente. Esto es, para la coordenada  $x$ , tendremos valores que van de  $[0, a]$  y lo mismo para las coordenadas restantes.

c) Restricción en periódica en 2D:

Los manera en que se toman los valores para las coordenadas  $x$  y  $y$  es el mismo que en el caso en 2D, pero la coordenada en  $z$  está restringida por un valor de tolerancia.

2. Temperaturas bajas ( $T_k < 0.001$  hasta la condición de paro  $T_k \leq 0.00001$ )

a) Restricción esférica:

Los movimientos a temperaturas bajas están dados por un intervalo de  $[-1.0, 1.0]$  sin importar el radio de la esfera dado por el usuario.

b) Restricciones en 2D y 3D:

Para ambos casos los movimientos aleatorios se hacen en un intervalo de  $[-0.25, 0.25]$  en el espacio de coordenadas relativas. Recordemos que cualquier celda en coordenadas relativas siempre tiene valores de  $a = 1,0, b = 1,0, c = 1,0$

### 4.2.2. Archivos de salida

El archivo de salida que imprime RISSA (.rissa) contiene las coordenadas cartesianas de todos los fragmentos, moléculas o átomos presentes en el sistema, tiene un nombre que se identifica fácilmente con el método elegido al inicio y es compatible con cualquier tipo de visualizador. Para el caso de sistemas periódicos, el código RISSA tiene la opción de mostrar los resultados en formato (.xtl) el cual contiene los parámetros de celda y las correspondientes coordenadas relativas. Estos archivos pueden visualizarse con el software Vesta. Cuando los sistemas son de gran tamaño, por ejemplo un estudio de contactos moleculares en MOFs (Metal Organic Frameworks) o sobre superficies. Para ambos casos se mantienen fijos tales fragmentos, así el archivo de salida sólo contiene la información de las coordenadas cartesianas de los fragmentos no fijos.

Debido al gran número de estructuras aceptadas a temperaturas altas, RISSA tiene un valor de tolerancia para comenzar a guardar la información de las estructuras en el archivo de salida de 10.0 unidades en el parámetro de control, de esta manera sólo se guardarán en disco las estructuras que estén por debajo de este valor. A la par, un archivo llamado "Path\_SA.out" contiene el historial de energías obtenidas en cada uno de los decrementos de temperatura, con el cual se puede ir monitoreando el avance del cálculo.

### 4.2.3. Ejecución de RISSA

Una vez descritas las partes más importantes del programa, es útil mostrar un ejemplo de como correr el programa. Teniendo el binario para el sistema operativo deseado se guarda en un directorio en el cual se desea trabajar, paso siguiente en el mismo directorio y desde una terminal se escribe lo siguiente: ./RISSA. Y se desplegará un menú como lo muestra la figura (4.8).

```

  R I S S A   G O D S
  R I S S A   G O D S

*****
Restricted Isomers Searching by Simulated Annealing Code
*****

Version: 3.0

Authors:
Juan Jose Garcia & Jorge Garza

Departamento de Química
Universidad Autónoma Metropolitana-Iztapalapa
*****

Available Geometries:
Type: Cell or cell
Type: Sphere or sphere
Type: Surface or surface
Type the selected geometry:
Cell
Your choice is: Cell
What do you want to work with?
Lennard-Jones potential -> 1
Fukui potential -> 2
Semiempirical method -> 3
3
Enter a high temperature
500.0
How many random moves?
100
MOPAC initial data from file >> cell_data.xyz
```

Restricción espacial

Tipo de función objetivo

Valor del parámetro de control inicial

Número de iteraciones

Archivo de entrada

Figura 4.8: Menú de ejecución para el código RISSA. Los datos mostrados son para un cálculo en sistemas periódicos usando MOPAC

---

En la figura anterior las casillas en rojo muestran los datos necesarios para llevar a cabo una simulación en fase sólida utilizando como función objetivo la energía semiempírica de MOPAC. Una vez terminada la simulación se generarán los archivos descritos en la sección (4.2.2).

### 4.3. Calibración de RISSA

Como parte de conocer el buen funcionamiento del código desarrollado, probamos estudiar sistemas que ya habían sido reportados en la literatura para constatar que nuestro código nos estuviera llevando por el camino correcto para la obtención de buenas estructuras iniciales las cuales nos permitirán alcanzar el mínimo global.

#### Cúmulos de sodio

Se realizó una búsqueda de isómeros de cúmulos de sodio  $\text{Na}_4$ . Los parámetros requeridos para la simulación fueron obtenidos de los reportados por Slowicka [22]. Confinando al sistema en una esfera de radio  $r = 3,0\text{\AA}$  y con un valor de  $c_T = 500,0$  para el parámetro de control, después de 200 movimientos, se obtuvieron 106 estructuras de las cuales sólo las primeras cuatro de más baja energía fueron optimizadas con un cálculo de estructura electrónica, usando el método Hartree-Fock con la base 6-311G\* basados en los cálculos de Solov [23]. En la figura (4.9) se muestran las cuatro estructuras de más baja energía, todas convergieron al mismo mínimo global después de hacer la optimización con el método de estructura electrónica.

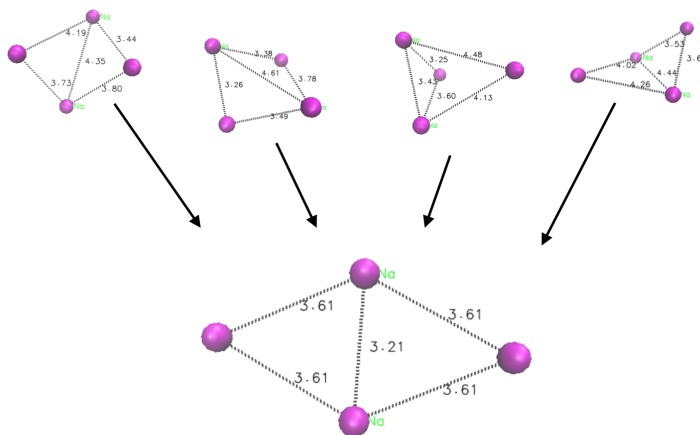


Figura 4.9: Todas las estructuras iniciales llegaron al mismo mínimo global

#### Tetrámero de agua

Se realizó una búsqueda de isómeros dónde la energía a optimizar en el recocido simulado fue obtenida por un método semiempírico PM7 (incluido en el código MOPAC). Utilizando una esfera con radio de confinamiento  $r=4.0\text{\AA}$ , un valor para el parámetro de control  $c_T=500$  y 200 movimientos se obtuvieron un total de 20 estructuras que pasaron los criterios mencionados en la sección anterior. Estos candidatos fueron optimizados a un nivel de teoría B3LYP/6-311++G\*\* [24] utilizando el paquete computacional NWChem [25]. Los isómeros fueron numerados del más bajo al más alto

según la energía del método *ab-initio* y se presentan en la tabla (4.1) con su respectiva energía relativa reportada en kcal/mol.

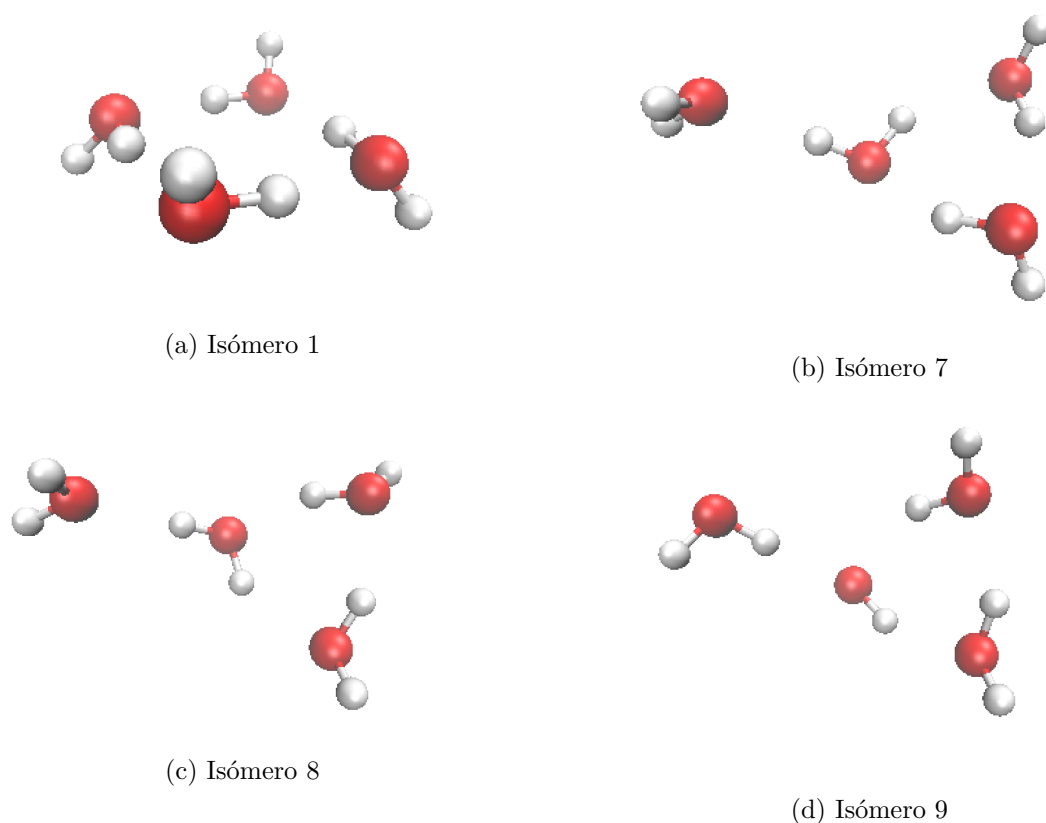


Figura 4.10: Isómeros que llegaron a la energías bajas después de la optimización B3LYP/6-311++G\*\*

Tabla 4.1: Energías relativas para cada isómero del tetrámero de agua figura (4.10)

| Isómero | $\Delta E(kcal/mol)$ |
|---------|----------------------|
| I       | 0.0                  |
| II      | 0.2                  |
| III     | 0.3                  |
| IV      | 0.9                  |

Más allá de tratar de reproducir estructuras reportadas en la literatura, el objetivo principal hasta este momento fue desarrollar exitosamente un programa de carácter estocástico que permita explorar de una manera extensa la Superficie de Energía Potencial en isómeros de cualquier tamaño y obtener los mejores candidatos para alcanzar el mínimo global. El siguiente paso es utilizar el código RISSA para el estudio de isómeros que no han sido abordados hasta ahora.



## Capítulo 5

# Resultados

*“Dale a alguien un programa y lo frustrarás por un día. Enséñale cómo programar y lo frustrarás por toda su vida”*

---

*David Leinweber*

La implementación de un código de carácter estocástico para la búsqueda de isómeros bajo restricciones espaciales tiene como objetivo el poder predecir las mejores estructuras químicas iniciales con las cuales aplicando una buena metodología determinista podremos alcanzar o encontrar el mínimo global de ese sistema. Para ello, teniendo como motivación los resultados obtenidos en la sección anterior (4.3) se estudiaron una serie de sistemas químicos que no han sido reportados en la literatura: cúmulos de cloruro de sodio dihidratado.

La interacción entre NaCl y moléculas de agua ha sido por muchos años un tema muy importante para entender la disociación de esta sal. En muchos de los trabajos actuales [26, 27, 28] la proporción que presentan las moléculas de agua con respecto a las moléculas de NaCl es muy grande (de 3 a 10 moléculas de agua por cada molécula de NaCl). Con ello se pretende conocer la cantidad de moléculas de agua necesarias para poder separar el par iónico Na-Cl de una sola molécula de cloruro de sodio y solvatar sus correspondientes iones. La discusión sobre cuántas moléculas de agua son necesarias para separar al par iónico es un debate al cual Hou[29] responde que hay una delgada línea entre tener al par iónico separado-solvatado y tenerlo unido-solvatado. Sin embargo, al mantener la proporción de 2 moléculas de H<sub>2</sub>O por 1 molécula de NaCl la disociación no ocurre y el contacto Na-Cl está presente. En fase sólida el contacto Na-Cl está comprobado experimentalmente con la existencia de la celda de NaCl dihidratado[30] debido a que a temperaturas bajas dichos cristales crecen en la superficie de cristales de cloruro de sodio. Estudiar estos sistemas es de alto interés para las ciencias atmosféricas, debido a que los aerosoles de sales marinas tienen la habilidad de afectar directa o indirecta la cantidad de radiación que recibe la tierra. El estudio de los cúmulos de cloruro de sodio dihidratado es importante para el proceso de nucleación que se lleva a cabo en las nubes donde las temperaturas bajas están entre 200 y 235 K.

A pesar de la suma importancia de este tipo de sistemas y sus repercusiones atmosféricas existen pocos estudios teóricos al respecto, de hecho para cúmulos de (NaCl)<sub>n</sub>(H<sub>2</sub>O)<sub>2n</sub> con valores de  $n > 1$  no existen trabajos reportados en la literatura. Evidentemente para este tipo de agregados la Teoría de Funcionales de la Densidad es la metodología más apropiada debido a que usar métodos de función de onda conllevan a un costo computacional elevado. El objetivo de este último capítulo consiste en estudiar cúmulos (NaCl)<sub>n</sub>(H<sub>2</sub>O)<sub>2n</sub> con  $n = 1, 2, 4$  usando métodos de la química cuántica

para proporcionar los mejores candidatos al mínimo global y predecir la estructura más estable en fase gas.

## Metodología basada en búsquedas estocásticas y cálculos de estructura electrónica

Como primer paso en el estudio de los cúmulos de  $\text{NaCl}(\text{H}_2\text{O})_2$ , se realizó una búsqueda estocástica con el código RISSA descrito en la sección (4) utilizando varios confinamientos esféricos: 3.0, 3.5 y 4.0 Å, tomando como función objetivo la energía semiempírica del modelo paramétrico número siete (PM7) del software MOPAC[18]. De este paso se obtuvieron 500 estructuras por radio de confinamiento y todas aquellas que estuvieran en un intervalo de 10.0 kcal/mol con respecto a la de más baja energía fueron llevadas a una pre-optimización con el método PBE0/6-31+G, de este último paso sólo se tomaron las estructuras dentro de un intervalo de 10.0 kcal/mol con diferencias geométricas considerables.

Como paso siguiente se hizo una optimización y análisis de frecuencias de las geometrías provenientes del método PBE0/6-31+G con los ocho funcionales de intercambio y correlación: PBE[31], PBE0[32], B3LYP[33], PBE-D3[34], PBE0-D3[34], LC- $\omega$ PBE[35] y M06-2x[36] con las funciones de base 6-311++G(d,p)[37] y aug-cc-pVTZ[38] para el sistema  $\text{NaCl}(\text{H}_2\text{O})_2$  así como una optimización de geometría con los métodos MP2[39] y CCSD(T)[40]. Para el resto de los cúmulos  $(\text{NaCl})_2(\text{H}_2\text{O})_4$  y  $(\text{NaCl})_4(\text{H}_2\text{O})_8$  sólo se optimizó con el funcional M06-2X y la base 6-311++G(d,p).

### 5.1. Primer sistema: $\text{NaCl}(\text{H}_2\text{O})_2$

Para este primer agregado  $\text{NaCl}(\text{H}_2\text{O})_2$  existen varios estudios los cuales son principalmente bajo metodologías de función de onda. Tal es el caso de Hou y colaboradores[29] donde estudiaron  $\text{NaCl}(\text{H}_2\text{O})_m$ . Para  $m = 2$ , esto es  $\text{NaCl}(\text{H}_2\text{O})_2$ , reportaron las geometrías y las energías relativas usando el método CCSD(T)/aug-cc-pVTZ sobre las geometrías de un método LC- $\omega$ PBE/6-311++G(d,p).

En nuestro caso los isómeros obtenidos para  $\text{NaCl}(\text{H}_2\text{O})_2$  son de suma importancia debido a que nuestro objetivo es estudiar sistemas con proporciones mayores, fue necesario desarrollar una metodología basada en la TFD que proporcione la mejor descripción de las interacciones presentes en los cúmulos de cloruro de sodio dihidratado, por ello se probó el desempeño de ocho funcionales de intercambio y correlación: PBE, PBE0, B3LYP, PBE-D3, PBE0-D3, LC- $\omega$ PBE y M06-2x con respecto a dos cálculos de función de onda MP2 y CCSD(T), como se mencionó anteriormente. Las geometrías obtenidas con el funcional M06-2X fueron reoptimizadas con los métodos MP2 y CCSD(T) usando las funciones de base 6-311++G(d,p) y aug-cc-pVTZ. La tabla (5.1) muestra los resultados obtenidos.

Tabla 5.1: Energías relativas en eV. Los Isómeros I y II se muestran en la figura 5.1.

|    | CCSD(T)       | MP2   | PBE-D3 | PBE   | PBE0-D3 | PBE0  | B3LYP-D3 | B3LYP | M06-2X | LC- $\omega$ PBE |
|----|---------------|-------|--------|-------|---------|-------|----------|-------|--------|------------------|
|    | aug-cc-pVTZ   |       |        |       |         |       |          |       |        |                  |
| I  | 0.0           | 0.0   | 0.0    | 0.072 | 0.0     | 0.0   | 0.0      | 0.0   | 0.0    | 0.0              |
| II | 0.091         | 0.097 | 0.011  | 0.0   | 0.034   | 0.112 | 0.022    | 0.051 | 0.118  | 0.065            |
|    | 6-311++G(d,p) |       |        |       |         |       |          |       |        |                  |
| I  | 0.0           | 0.0   | 0.094  | 0.053 | 0.057   | 0.0   | 0.0      | 0.0   | 0.0    | 0.0              |
| II | 0.115         | 0.109 | 0.0    | 0.0   | 0.0     | 0.047 | 0.008    | 0.037 | 0.100  | 0.048            |

Para el cúmulo  $\text{NaCl}(\text{H}_2\text{O})_2$  obtuvimos dos únicos isómeros que se presentan en la figura (5.1)

los cuales son congruentes con los reportados por Hou y colaboradores[29]. Para este sistema, según la tabla (5.1) es claro que el funcional de intercambio y correlación PBE exhibe los peores resultados para ambas funciones de base. Los mejores resultados se obtuvieron con los funcionales PBE y M06-2X para el caso de la base grande. A pesar de ello, el resultado de decrementar el tamaño de la función de base repercute de manera importante en el funcional PBE0, dejando a M06-2X como la única buena aproximación al método de cúmulos acoplados.

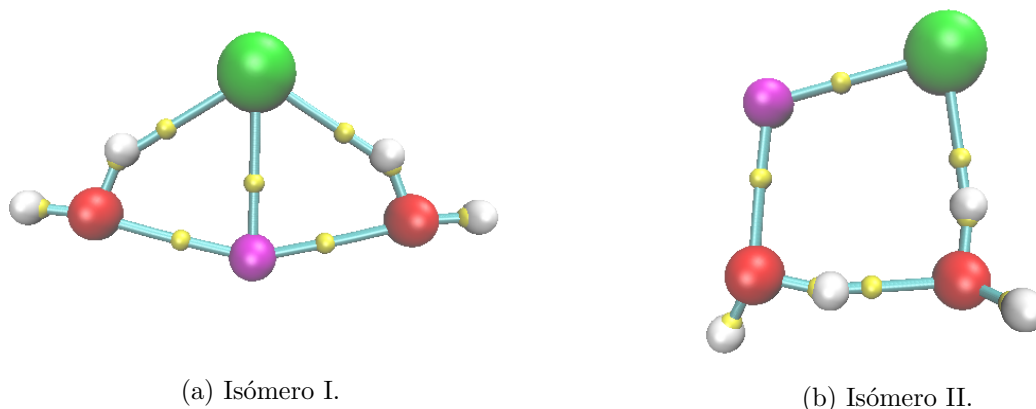


Figura 5.1: Estructuras obtenidas con la metodología CCSD(T)/aug-cc-pVTZ. La caracterización con la teoría de átomos en moléculas se hizo con M06-2X/6-311++G(d,p)

Para corroborar la existencia del contacto Na-Cl se realizó un análisis con la teoría de Átomos en Moléculas. Este análisis muestra la ruta de enlace “Bond Path” (en cilindros azules) y el punto crítico de enlace “Bond Critical Point” (en esferas amarillas) entre el átomo de cloro y el átomo de sodio (figura 5.1) lo cual prueba que el par iónico no se disocia en presencia de dos moléculas de H<sub>2</sub>O.

Debido al costo computacional es claro que para estudiar los cúmulos siguientes no es viable utilizar bases como la aug-cc-pVTZ. Por ello fue que se hizo el análisis completo de todos los funcionales y de ambos métodos de función de onda con la base 6-311++G(d,p). Por lo tanto a partir de los resultados para el cúmulo de menor tamaño, el método M06-2X/6-311++G(d,p) fue el elegido para estudiar los cúmulos de mayor tamaño.

Una conclusión parcial que se puede obtener del estudio de este primer sistema es la confiabilidad del código RISSA, ya que a pesar de haber utilizado la energía semiempírica como función objetivo se pudo llegar a las dos estructuras de más baja energía reportadas para el sistema NaCl(H<sub>2</sub>O)<sub>2</sub>. Esto proporciona una gran motivación para seguir con el estudio de los cúmulos de mayor tamaño (NaCl)<sub>2</sub>(H<sub>2</sub>O)<sub>4</sub> y (NaCl)<sub>4</sub>(H<sub>2</sub>O)<sub>8</sub>.

## 5.2. Segundo sistema: (NaCl)<sub>2</sub>(H<sub>2</sub>O)<sub>4</sub>

De acuerdo a los resultados obtenidos en la sección (5.1) el funcional M06-2X es el indicado para estudiar las interacciones presentes en los sistemas de NaCl dihidratados. Como se mencionó previamente, para evitar el uso de la intuición química se utilizó el código estocástico RISSA para obtener las mejores estructuras iniciales de más baja energía. En este paso se obtuvieron 150 candidatos los cuales se sometieron a una optimización inicial con el funcional PBE0 y la base 6-31+G\*. Utilizando un criterio energético, todos los isómeros dentro de un intervalo de 10.0 kcal/mol (obteniendo un total de 120 isómeros) fueron optimizados con la metodología M06-2X/6-311++G(d,p) previamente

desarrollada. De los 120 isómeros anteriores sólo 56 estuvieron dentro del criterio de 10.0 kcal/mol. Estas 56 geometrías fueron estudiadas con el modelo de átomos en moléculas para caracterizar las interacciones presentes en los agregados de  $(\text{NaCl})_2(\text{H}_2\text{O})_4$ . En la figura (5.3) se muestran algunos de los isómeros de más baja energía obtenidos para el cúmulo  $(\text{NaCl})_2(\text{H}_2\text{O})_4$ . Claramente podemos observar las interacciones entre el agua y las moléculas de cloruro de sodio persisten y donde al menos en el isómero de más baja energía, existe un arreglo perfecto entre las moléculas de agua. En este isómero es evidente que sólo dos moléculas de agua están donando hidrógenos para formar los puentes de hidrógeno, ya que las otras dos están orientando las posiciones de sus hidrógenos hacia los átomos de cloro.

La figura (5.3) presenta una curiosa situación, en la cual se encontró que el isómero XXVII tiene una gran similitud estructural con el isómero I a pesar de que la diferencia entre ellos es de 5.5 kcal/mol tal y como lo muestra la tabla (5.2), donde D representa la suma de la energía electrónica y la corrección de la energía de punto cero,  $D=E+ZPE$ . Es clara que la única diferencia geométrica radica en las orientaciones de las moléculas de agua, puesto que en el caso del isómero XXVII sólo una molécula de agua está proporcionando sus dos hidrógenos para formar los puentes de hidrógeno.

Tabla 5.2: Energías relativas para los primeros seis isómeros de más baja energía y el isómero XXVII

| Isómero | $\Delta D(kcal/mol)$ |
|---------|----------------------|
| I       | 0.0                  |
| II      | 2.0                  |
| III     | 2.2                  |
| IV      | 2.2                  |
| V       | 2.9                  |
| VI      | 3.1                  |
| XXVII   | 5.5                  |

Este hecho demuestra que la superficie de energía potencial es lo suficientemente rugosa para pasar de un isómero a otro por medio de una pequeña perturbación geométrica.

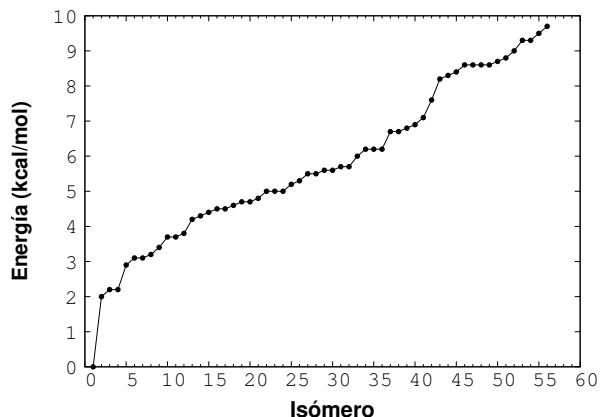


Figura 5.2: Energía relativa incluyendo la energía de punto cero para los 56 isómeros encontrados para el sistema  $(\text{NaCl})_2(\text{H}_2\text{O})_4$

La brecha entre el isómero de más baja energía y el segundo está alrededor de 2.0 kcal/mol.

Debido a la exactitud ( $\approx 2.0$  kcal/mol) que presentan los cálculos basados en TFD podemos decir que el isómero con más probabilidad de ser encontrado en fase gas es el I.

### 5.3. Tercer sistema: $(\text{NaCl})_4(\text{H}_2\text{O})_8$

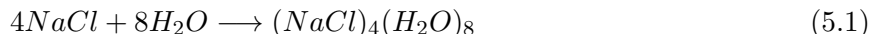
El sistema de mayor tamaño  $(\text{NaCl})_4(\text{H}_2\text{O})_8$  fue estudiado bajo la misma metodología descrita para el sistema  $(\text{NaCl})_2(\text{H}_2\text{O})_4$ . Para este agregado de cloruro de sodio dihidratado se obtuvieron un total de 38 isómeros. De los cuales el isómero de más baja energía presenta una estructura muy peculiar, puede observarse que las cuatro moléculas de agua forman un cubo de proporciones no simétricas y las ocho moléculas de agua están rodeando dicho cubo como lo muestra la figura (5.4).

En un estudio previo, el cúmulo  $\text{Na}_4\text{Cl}_4$  fue caracterizado experimentalmente por medio de espectroscopia de rayos-x y la forma de cubo distorcionado fue determinada como la estructura más probable[41], apoyando nuestro resultado. En pocas palabras, el agregado molecular  $(\text{NaCl})_4(\text{H}_2\text{O})_8$  usando los términos propuestos por Hou[29], es del tipo par iónico unido y solvatado debido a que el par Na-Cl no fue separado por el efecto del solvente. En la figura 5.6 los puntos críticos de enlace presentes entre los átomos de Na y Cl corroboran nuestra afirmación.

El gráfico de la energía en la figura (5.5) muestra que la gran brecha energética que existe entre el isómero más bajo y el siguiente energía ronda las 3.0 kcal/mol, por lo cual podemos concluir lo mismo de antes, el isómero de más baja energía mostrado en la figura (5.4) es la estructura más probable de encontrar a la temperatura de 0K en fase gas.

A pesar de que nuestros resultados predicen la estructura más estable, esto es a temperatura de 0K, en consecuencia, es deseable conocer el efecto de la temperatura en el ordenamiento de los isómeros. Por esta razón se calculó la termoquímica para todos los isómeros a varias temperaturas con la finalidad de observar los cambios en la energía libre de cada isómero debido a las contribuciones entálpicas y entrópicas.

Para constatar que la reacción tenga un caracter exotérmico, adicionalmente se calculó el cambio en la energía libre para la reacción



La figura (5.7) muestra que el isómero I constatado como el más bajo en energía y por lo tanto más probable a 0K, mantiene este comportamiento para las temperaturas de 0K, 100 K y 300K mostradas en la gráfica. Además las brechas energéticas entre los isómeros I y II a las tres temperaturas se mantienen lo suficientemente grandes como para asegurar la existencia del isómero I como el más probable a la temperatura correspondiente.

Los resultados de la energía libre de formación para la reacción de la ecuación (5.1) se listan en la tabla (5.3). La tabla muestra que la reacción asociada a la formación del agregado de  $(\text{NaCl})_4(\text{H}_2\text{O})_8$  es lo suficientemente exotérmica para que el compuesto se forme a condiciones normales de presión y temperatura. Con esta información obtenida podemos argumentar que gran parte de los aerosoles de sales marinas tienen presente el cúmulo de  $(\text{NaCl})_4(\text{H}_2\text{O})_8$  en la fase gas, teniendo como más probable a la estructura de la figura (5.4).

Como fue mencionado en el inicio de este capítulo, los aerosoles de sales marinas además de contribuir al equilibrio de la radiación terrestre, son una pieza importante en la nucleación de partículas hielo en el modo de congelamiento por inmersión a temperaturas por debajo de los 220 K. Los aerosoles de NaCl pueden existir ya sea como gotitas de solución acuosa o en una de sus dos formas cristalinas: NaCl anhidro y NaCl dihidratado[42]. El cloruro de sodio dihidratado es la fase sólida termodinámicamente estable a temperaturas debajo de los 273.3 K y fue experimentalmente

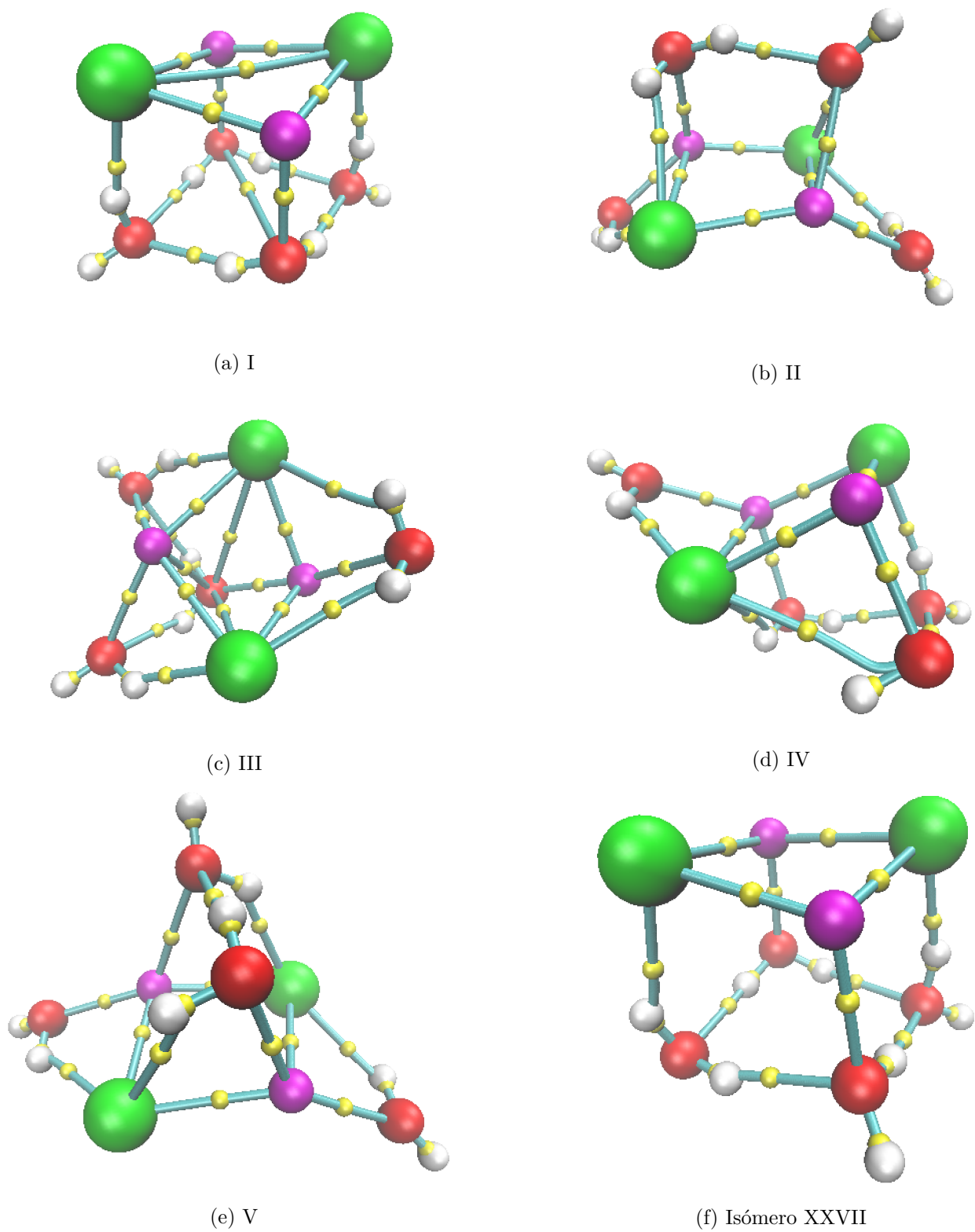
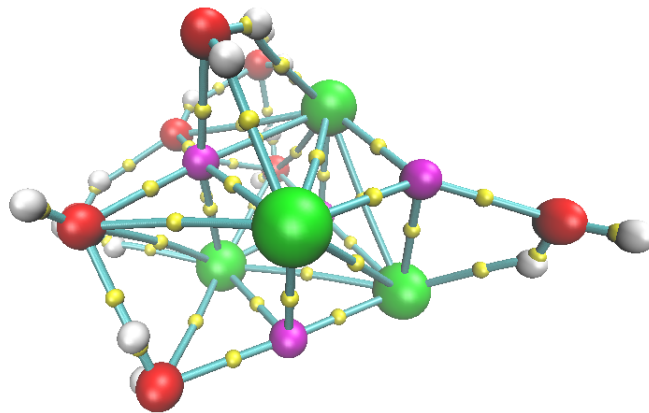


Figura 5.3: Representación gráfica para los isómeros I-V y XXVII. Análisis de átomos en moléculas con la metodología M06-2X/6-311++G(d,p).



(I)

Figura 5.4: Isómero de más baja energía encontrado para el agregado  $(\text{NaCl})_4(\text{H}_2\text{O})_8$ . Análisis de átomos en moléculas con la metodología M06-2X/6-311++G(d,p).

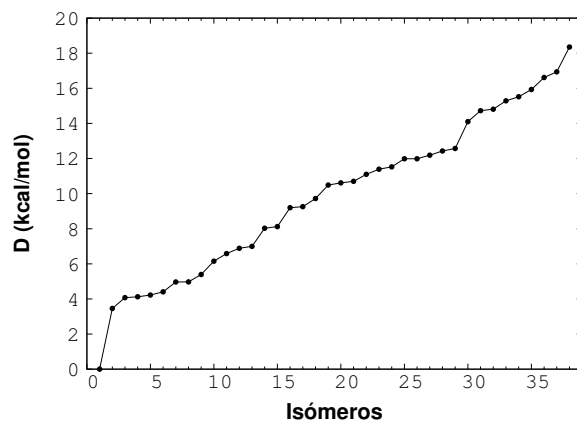


Figura 5.5: Energía relativa D para los 36 isómeros encontrados para el sistema  $(\text{NaCl})_4(\text{H}_2\text{O})_8$

Tabla 5.3: Energía libre para la formación del agregado  $(\text{NaCl})_4(\text{H}_2\text{O})_8$  a 298.15 K

| Isómero | $\Delta G_{\text{formación}}$ (kcal/mol) |
|---------|--|
| I       | -258.64                                  |
| II      | -255.86                                  |
| III     | -255.53                                  |
| IV      | -255.27                                  |
| V       | -255.17                                  |

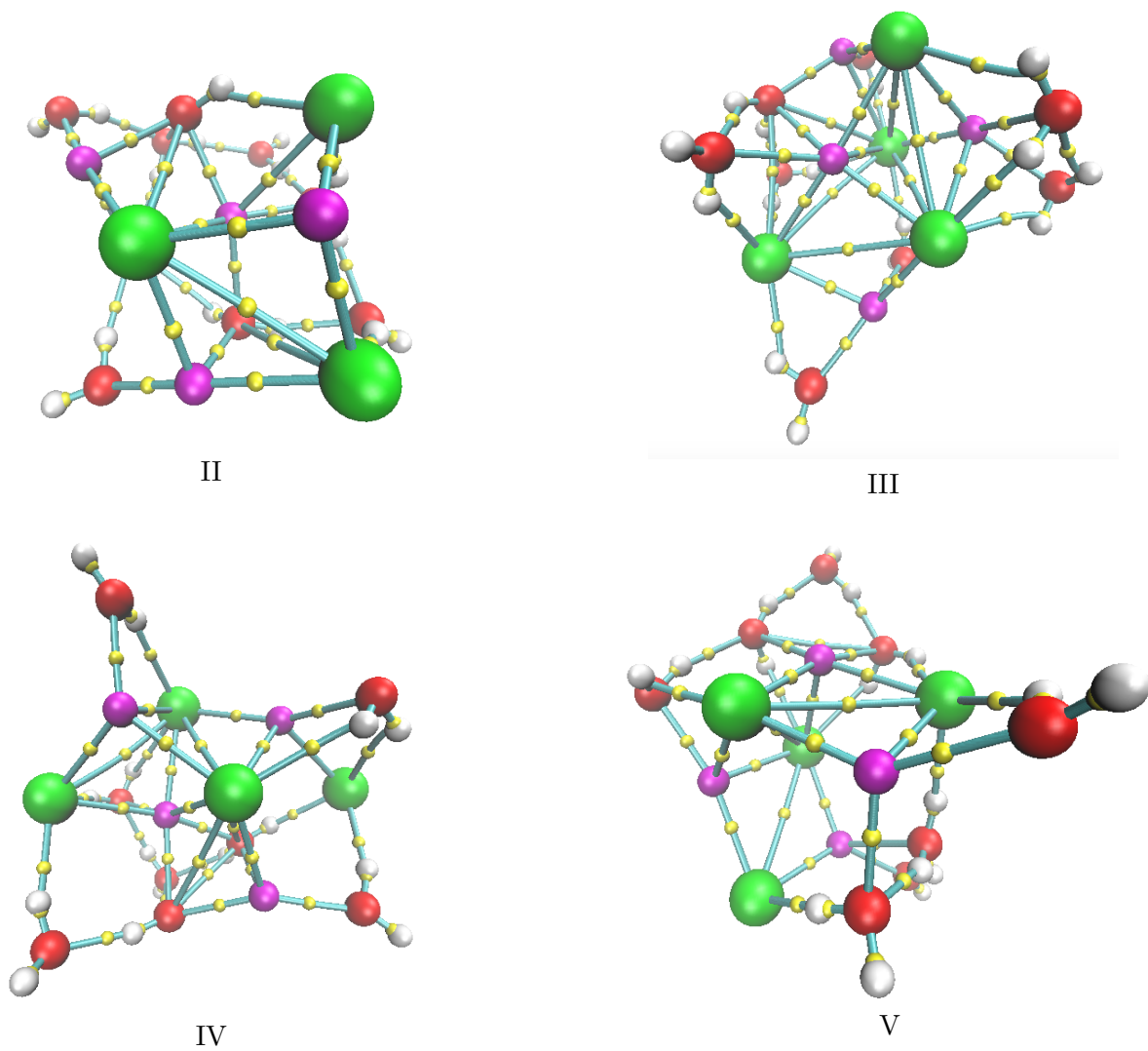


Figura 5.6: Isómeros II-V para el agregado  $(\text{NaCl})_4(\text{H}_2\text{O})_8$ . Análisis de átomos en moléculas con la metodología M06-2X/6-311++G(d,p).



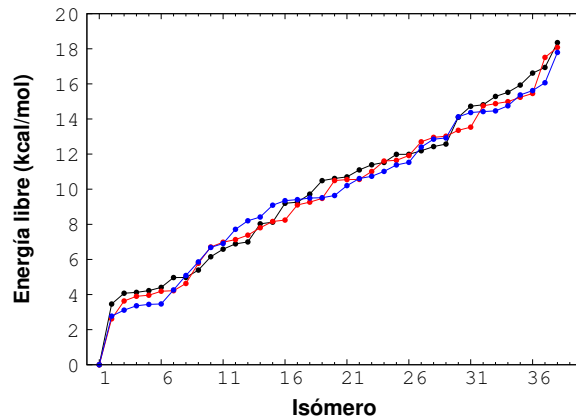


Figura 5.7: Energía libre relativa a diferentes temperaturas para los 36 isómeros encontrados para el sistema  $(\text{NaCl})_4(\text{H}_2\text{O})_8$ . Líneas y puntos en negro, rojo y azul para las temperaturas de 0 K, 100 K y 300 K respectivamente

comprobado que se forma por eflorescencia en las gotitas de NaCl en solución acuosa[43].

La información más relevante de lo mencionado anteriormente, es la caracterización de la estructura cristalina para el cloruro de sodio dihidratado. Arno y colaboradores[30] lograron el crecimiento de cristales de  $\text{NaCl}(\text{H}_2\text{O})_2$  sobre los cristales de NaCl. La celda unitaria para este sistema se presenta en la figura (5.8).

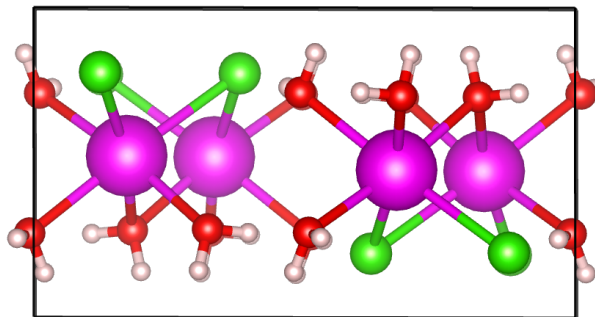


Figura 5.8: Celda unitaria para  $\text{NaCl}(\text{H}_2\text{O})_2$  obtenida experimentalmente. La celda contiene dos unidades formula, lo cual deja la proporción  $(\text{NaCl})_4(\text{H}_2\text{O})_8$ .

Evidentemente la proporción que presenta la celda unitaria del cloruro de sodio dihidratado es la misma que presenta el cúmulo de mayor tamaño estudiado anteriormente  $(\text{NaCl})_4(\text{H}_2\text{O})_8$ . Estudiar los agregados moleculares de  $(\text{NaCl})_n(\text{H}_2\text{O})_{2n}$  con  $n = 1, 2, 4$  tiene como finalidad encontrar posibles similitudes entre las interacciones presentes en las moléculas de NaCl y  $\text{H}_2\text{O}$  tanto en el cristal como en el agregado de mayor tamaño en fase gas, el cual presenta la misma proporción de

moléculas de NaCl y agua que el cristal.

Motivados por esto y utilizando los resultados de los análisis con la teoría de átomos en moléculas realizada previamente para cada agregado, hicimos un promedio de la longitud de enlace Na-Cl presente en cada uno de los isómeros de  $(\text{NaCl})_n(\text{H}_2\text{O})_{2n}$  con  $n = 1, 2, 4$  y adicionalmente para la molécula de cloruro de sodio aislada.

Tabla 5.4: Distancias promedio de enlace Na-Cl ( $\text{\AA}$ ) encontradas con el análisis de átomos en moléculas.

| Isómero | NaCl  | $\text{NaCl}(\text{H}_2\text{O})_2$ | $(\text{NaCl})_2(\text{H}_2\text{O})_4$ | $(\text{NaCl})_4(\text{H}_2\text{O})_8$ | $\text{NaCl}(\text{H}_2\text{O})_2$ (Cristal exp.) |
|---------|-------|-------------------------------------|---|---|--|
| I       | 2.367 | 2.546                               | 2.593                                   | 2.700                                   | 2.790  |
| II      |       | 2.445                               | 2.674                                   | 2.695                                   |  |
| III     |       |                                     | 2.659                                   | 2.734                                   |  |
| IV      |       |                                     | 2.636                                   | 2.670                                   |  |
| V       |       |                                     | 2.656                                   | 2.693                                   |  |

Los resultados de la tabla (5.4), muestran que a medida de que se incrementa el tamaño del cúmulo las interacciones presentes se van asemejando a las que presenta el cristal. Para la molécula aislada de NaCl su valor de  $2,367\text{\AA}$  se ve afectado por la presencia de dos moléculas de agua, aumentando su distancia de enlace a  $2.546\text{\AA}$ . Es evidente que el efecto que producen las aguas es incrementar la distancia del enlace NaCl, sin embargo mantener la proporción 1:2 evita la total disociación del par iónico.

## Capítulo 6

# Perspectivas: periodicidad en 3D y 2D

*“La simplicidad es un requisito previo para la confiabilidad.”*

---

*E. W. Dijkstra*

A pesar de que las restricciones en 3D y 2D ya están por completo implementadas en el código RISSA, existen dificultades en cuanto al uso de software de terceros para calcular la energía al trabajar con cristales y superficies. Estos detalles surgen cuando el usuario desea utilizar información previa (experimental) para analizar su sistema. En estos casos, el usuario debe adaptarse a la forma en que el software MOPAC lee los datos de entrada para llevar a cabo los cálculos en cristales y superficies debido a que la energía semiempírica es la única implementada para hacer cálculos sobre este tipo de sistemas.

### Restricción en 3D: fragmento fijo e información experimental

Como se mencionó en capítulos anteriores RISSA tiene la propiedad de mantener fijo uno de los fragmentos dentro de la geometría esférica, lo cual es de gran interés si se quiere estudiar algún fenómeno de solvatación o de coordinación. El uso de esta propiedad es de gran utilidad en los conocidos “Metal-Organic Frameworks” (MOFs) ya que son de gran interés los sitios donde otros fragmentos moleculares serán adsorbidos. Para ello, se mantienen fijas las coordenadas del MOF y las demás moléculas se mueven aleatoriamente dentro de la celda aplicando las condiciones periódicas en 3D. Es importante que todos los átomos estén dentro de la celda para que el cálculo del Campo Auto-Consistente (SCF) que lleva a cabo MOPAC sea exitoso, de lo contrario existirán traslapes atómicos cuando la celda sea replicada en las direcciones  $x$ ,  $y$ ,  $z$ .

La búsqueda estocástica para un sistema como el mostrado en la figura (6.1) consiste en colocar los fragmentos móviles al centro de la celda y a partir de ese punto comenzar los movimientos estocásticos, como se mencionó anteriormente sólo las estructuras que minimizen la energía serán mandadas a un archivo excluyendo las coordenadas del MOF. Este tipo de sistemas debido a su tamaño demandan horas en tiempos de ejecución, debido a que RISSA debe comprobar las distancias de los átomos de cada uno de los fragmentos móviles a cada uno de los átomos involucrados en el MOF. La figura (6.2) muestra la estructura de más baja energía dentro del conjunto de buenas soluciones de un archivo de salida del código RISSA. Evidentemente, al ser una estructura dentro de un conjunto de varios posibles candidatos al mínimo global (alrededor de 300) es necesario mejorar la estructura permitiendo el relajamiento con algún método *ab-initio*.

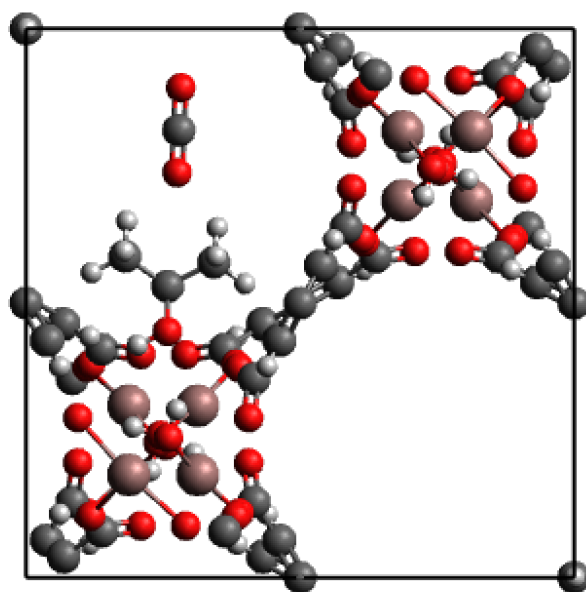


Figura 6.1: Propuesta para las posiciones de los fragmentos  $\text{CO}_2$  e Isopropanol dentro de la celda haciendo uso de la intuición química.

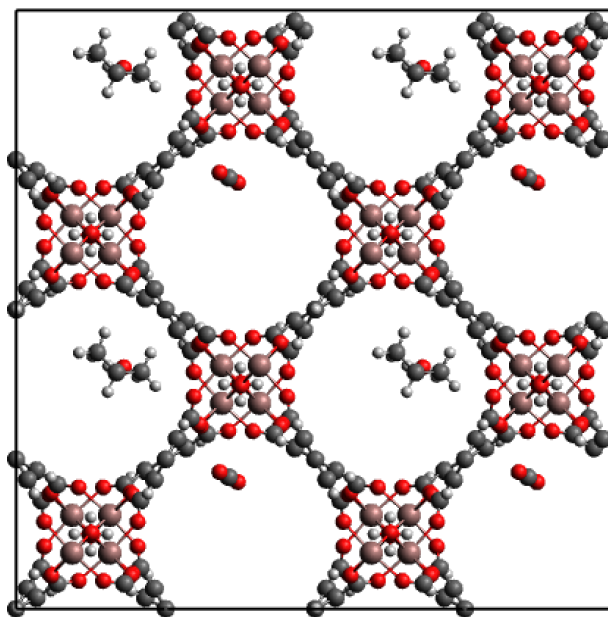


Figura 6.2: Estructura de más baja energía obtenida con el código RISSA usando como función objetivo la energía semiempírica de MOPAC.

---

## Restricción periódica en 2D

Dentro de RISSA la restricción periódica en 2D tiene predeterminado fijar un fragmento y usar las coordenadas iniciales de la superficie sin modificarlas. Para realizar este tipo de cálculos, en donde nos interesa encontrar los mejores sitios de donde se situará un adsorbato es importante que todos los átomos de la superficie estén dentro de los vectores de traslación que definen la definen de lo contrario MOPAC no realizará el cálculo SCF.

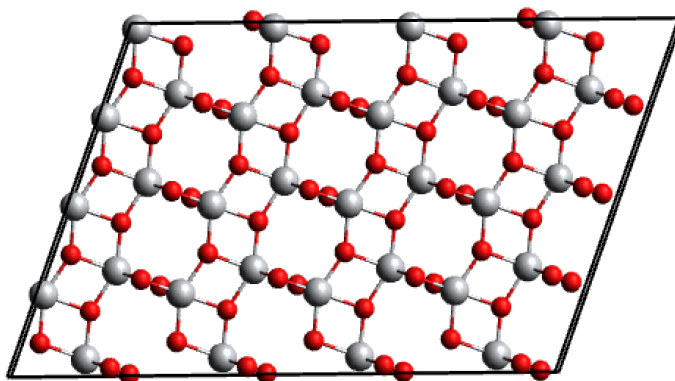


Figura 6.3: Ejemplo de una superficie catalítica con todos los átomos dentro de la celda construida a partir de los vectores de traslación.

En la figura (6.3) se muestra un ejemplo de una superficie con titanio y oxígeno, tanto MOPAC como RISSA requieren que todos los átomos de la superficie estén dentro de la celda que se construye a partir de los vectores de traslación que ingresa el usuario. Este requisito se debe principalmente para aplicar las condiciones periódicas en 2D de manera satisfactoria. A pesar de la gran utilidad que proporciona el código RISSA para la búsqueda de isómeros en 2D, los resultados dependen en gran medida de los cálculos energéticos proporcionados por MOPAC y de los datos que ingresa el usuario al inicio para definir su superficie catalítica.

Debido a las problemáticas en cuanto a construir una superficie correcta y a la dependencia de los cálculos de la energía proporcionados por MOPAC, queda como perspectiva generar una interface que verifique los datos de entrada para una ejecución exitosa.

## Capítulo 7

# Conclusiones

La predicción de estructuras químicas por medio de algoritmos estocásticos tanto para sistemas finitos como para sistemas periódicos es una área de investigación muy activa actualmente y es relevante para el diseño de nuevos materiales. Cuando se tratan sistemas de gran tamaño, hacer una propuesta de las posibles estructuras de baja energía pasa a ser una tarea altamente compleja y la intuición química no es suficiente. Entonces cuando la intuición humana falla, los algoritmos de búsqueda heurística tienen un gran éxito. Nuestro código RISSA (Restricted Isomers Searching by Simulated Annealing) es una excelente herramienta para generar múltiples estructuras de mínima energía, las cuales sirven como punto de partida para llegar al mínimo global sin la necesidad de usar la intuición química. Prueba de ello son todos los isómeros encontrados y reportados para los agregados moleculares de cloruro de sodio dihidratado, los cuales muestran geometrías que difícilmente podríamos imaginar que existen. Si bien, actualmente existen muchos códigos de búsqueda global que pueden llevar a cabo esta tarea, la mayoría de ellos trabajan bajo restricciones muy específicas. El código desarrollado por nosotros apunta a ser un herramienta versátil en cuanto a la exploración de isómeros con restricciones espaciales: cúmulos moleculares o atómicos confinados por esferas, estructuras cristalinas las cuales implícitamente poseen restricciones periódicas en 3D y superficies que presentan una periodicidad en 2D.

En la parte de aplicación, haciendo uso del código RISSA en conjunto con métodos de la química cuántica, se logró la obtención de los mínimos globales reportados para los cúmulos de  $\text{Na}_4$  y para el tetrámero de agua. Estos resultados proporcionaron gran confiabilidad en el código, motivando al estudio de sistemas de mayor tamaño y que no se habían estudiado anteriormente como es el caso de los cúmulos de  $\text{NaCl}$  dihidratado. Estos sistemas al ser de gran importancia para procesos de nucleación además de afectar directa e indirectamente la radiación atmosférica valen la pena ser estudiados para conocer el tipo de interacciones presentes en ellos. Los resultados obtenidos para los cúmulos de  $(\text{NaCl})_n(\text{H}_2\text{O})_{2n}$  con  $n = 1, 2, 4$  exhiben una tendencia a presentar el mismo tipo de interacciones que muestra el cristal de cloruro de sodio dihidratado, el cual ha sido caracterizado experimentalmente. Así mismo la gran estabilidad energética que presenta el isómero de más baja energía para el sistema  $(\text{NaCl})_4(\text{H}_2\text{O})_8$  apunta a ser el más probable para ser encontrado a las temperaturas de 100 K y 300 K. Con esta información y con los valores altamente exotérmicos (arriba de 100 kcal/mol) en la energía libre de formación a 298.15 K podemos asegurar que parte de la brisa marina contiene un gran número de cúmulos de  $(\text{NaCl})_4(\text{H}_2\text{O})_8$ .

# Bibliografía

- [1] JL Faulon. Stochastic generator of chemical structure .2. Using simulated annealing to search the space of constitutional isomers. *J. Chem. Inf. Comput. Sci.*, 36(4):731–740, 1996.
- [2] H Cordatos, D Ford, and RJ Gorte. Simulated annealing study of the structure and reducibility in ceria clusters. *J. Phys. Chem.*, 100:18128–18132, 1996.
- [3] Colin W. Glass, Artem R. Oganov, and Nikolaus Hansen. USPEX - Evolutionary crystal structure prediction. *Comput. Phys. Commun.*, 175(11-12):713–720, 2006.
- [4] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition, 2009.
- [5] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 3 edition, 2007.
- [6] Gerardo De los Cobos Silva, John Goddard, Miguel Gutiérrez, and Martínez Alma. *Búsqueda y exploración estocástica*. Colección CBI. Universidad Autónoma Metropolitana, Unidad Iztapalapa, 2010.
- [7] Dervis Karaboga. An idea based on honey bee swarm for numerical optimization. Technical report, Erciyes University, 2005.
- [8] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.
- [9] Artem R. Oganov, Andriy O. Lyakhov, and Mario Valle. How Evolutionary Crystal Structure Prediction Works-and Why. *Accounts Chem. Res.*, 44(3):227–237, 2011.
- [10] Zhai H. and Alexandrova A. N. Fluxionality of Catalytic Clusters: When It Matters and How to Address It. *ACS Catal.*, 7:1905–1911, 2017.
- [11] Jun Zhang and Michael Dolg. Global optimization of clusters of rigid molecules using the artificial bee colony algorithm. *Phys. Chem. Chem. Phys.*, 18:3003–3010, 2016.
- [12] Artem R. Oganov, Jiuhua Chen, Carlo Gatti, Yanzhang Ma, Yanming Ma, Colin W. Glass, Zhenxian Liu, Tony Yu, Oleksandr O. Kurakevych, and Vladimir L. Solozhenko. Ionic high-pressure form of elemental boron. *Nature*, 457(7231):863–867, 2009.
- [13] Yanming Ma, Mikhail Eremets, Artem R. Oganov, Yu Xie, Ivan Trojan, Sergey Medvedev, Andriy O. Lyakhov, Mario Valle, and Vitali Prakapenka. Transparent dense sodium. *Nature*, 458(7235):182–U3, 2009.

- 
- [14] K. Doll, J. C. Schoen, and M. Jansen. Structure prediction based on ab initio simulated annealing for boron nitride. *Phys. Rev. B*, 78(14), 2008.
- [15] N Metropolis, AW Rosenbluth, MN Rosenbluth, AH Teller, and E Teller. Equations Of State Calculations By Fast Computing Machines. *J. Chem. Phys*, 21(6):1087–1092, 1953.
- [16] N Metropolis and S Ulam. The Monte Carlo Method. *J. Am. Stat. Assoc.*, 44(247):335–341, 1949.
- [17] Osorio E, Ferraro M. B., Ona O. B., Cardenas C., Fuentealba P., and Tiznado W. Assembling Small Silicon Clusters Using Criteria of Maximum Matching of the Fukui Functions. *J. Chem. Theory Comput.*, 7:3995–4001, 2011.
- [18] Stewart Computational Chemistry, Colorado Springs, CO, USA. Mopac. <https://OpenMOPAC.net>.
- [19] Gerardo de los Cobos Silva, John Close G., and Miguel Gutierrez Andrade. Solución al acertijo del sudoku con recocido simulado. *ContactoS*, 84:31–39, 2012.
- [20] Robert H. Leary. Global Optima of Lennard-Jones Clusters. *Journal of Global Optimization*, 11(1):35–53, 1997.
- [21] Jesus Centeno and Patricio Fuentealba. Big bang methodology applied to atomic clusters. *International Journal of Quantum Chemistry*, 111:1419–1435, 2011.
- [22] A. Slowicka and Z. A. Walenta. Creating thin layers at the contact surface of two nonmixing liquids. *Bull. Pol. Acad. Sci.-Tech. Sci.*, 55:173–178, 2007.
- [23] Solov'yov IA., Solov'yov AV., and Greiner W. Structure and properties of small sodium clusters. *Phys. Rev. A*, 65, 2002.
- [24] Perez J. F., Hadad C. Z., and Restrepo A. Structural studies of the water tetramer. *Int. J. Quantum Chem.*, 108:1653–1659, 2008.
- [25] M. Valiev, E. J. Bylaska, N. Govind, K. Kowalski, T. P. Straatsma, H. J. J. Van Dam, D. Wang, J. Nieplocha, E. Apra, T. L. Windus, and Wa. de Jong. NWChem: A comprehensive and scalable open-source solution for large scale molecular simulations. *Comput. Phys. Commun.*, 181:1477–1489, 2010.
- [26] S. S. M. C. Godinho, P. C. do Couto, and B. J. C Cabral. Charge separation and charge transfer to solvent in nacl-water clusters. *Chem. Phys. Lett.*, 399:200–205, 2004.
- [27] Manik K. Ghosh, Suyong Re, Michael Feig, Yuji Sugita, and Cheol Ho Choi. Interionic hydration structures of nacl in aqueous solution: A combined study of quantum mechanical cluster calculations and qm/efp-md simulations. *J. Phys. Chem. B*, 117:289–295, 2013.
- [28] T Koop, A Kapilashrami, L T Molina, and M J Molina. Phase transitions of sea-salt/water mixtures at low temperatures: Implications for ozone chemistry in the polar marine boundary layer. *J. Geophys. Res.-Atmos.*, 105:26393–26402, 2000.
- [29] Gao-Lei Hou, Cheng-Wen Liu, Ren-Zhong Li, Hong-Guang Xu, Yi Qin Gao, and Wei-Jun Zheng. Emergence of Solvent-Separated Na<sup>+</sup>-Cl<sup>-</sup> Ion Pair in Salt Water: Photoelectron Spectroscopy and Theoretical Calculations. *J. Phys. Chem. Lett.*, 8(1):13–20, 2017.



- 
- [30] Arno A. C. Bode, Paul G. M. Pulles, Martin Lutz, Ward J. M. Poulisse, Shanfeng Jiang, Jan A. M. Meijer, Willem J. P. van Enkevort, and Elias Vlieg. Sodium Chloride Dihydrate Crystals: Morphology, Nucleation, Growth, and Inhibition. *Cryst. Growth Des.*, 15(7):3166–3174, 2015.
- [31] J. P. Perdew, K. Burke, and M. Ernzerhof. Generalized gradient approximation made simple. *Phys. Rev. Lett.*, 77:3865–3868, 1996.
- [32] Carlo Adamo and Vincenzo Barone. Toward reliable density functional methods without adjustable parameters: The PBE0 model. *J. Chem. Phys.*, 110:6158–6170, 1999.
- [33] Axel D. Becke. Density functional thermochemistry. iii. the role of exact exchange. *The Journal of Chemical Physics*, 98(7):5648–5652, 1993.
- [34] Stefan Grimme, Jens Antony, Stephan Ehrlich, and Helge Krieg. A consistent and accurate ab initio parametrization of density functional dispersion correction (dft-d) for the 94 elements h-pu. *The Journal of Chemical Physics*, 132(15):154104, 2010.
- [35] Oleg A. Vydrov and Gustavo E. Scuseria. Assessment of a long-range corrected hybrid functional. *The Journal of Chemical Physics*, 125(23):234109, 2006.
- [36] Yan Zhao and Donald G. Truhlar. The m06 suite of density functionals for main group thermochemistry, thermochemical kinetics, noncovalent interactions, excited states, and transition elements: two new functionals and systematic testing of four m06-class functionals and 12 other functionals. *Theor. Chem. Account.*, 120:215–241, 2008.
- [37] M. P. Andersson and P. Uvdal. New scale factors for harmonic vibrational frequencies using the b3lyp density functional method with the triple- $\xi$  basis set 6-311+g(d,p). *J. Phys. Chem A*, 109(12):2937–2941, 2005.
- [38] David E. Woon and Thom H. Dunning. Gaussian basis sets for use in correlated molecular calculations. iii. the atoms aluminum through argon. *J. Chem. Phys.*, 98(2):1358–1371, 1993.
- [39] Martin Head-Gordon, John A. Pople, and Michael J. Frisch. Mp2 energy evaluation by direct methods. *Chemical Physics Letters*, 153(6):503–506, 1988.
- [40] John A. Pople, Martin Head-Gordon, and Krishnan Raghavachari. Quadratic configuration interaction. a general technique for determining electron correlation energies. *The Journal of Chemical Physics*, 87(10):5968–5975, 1987.
- [41] G. É. Yalovega, A. V. Soldatov, K. Novak, M. Riedler, O. Löfken, A. Kolmakov, and T. Möller. Local geometry and electronic structure of free nacl clusters. *Physics of the Solid State*, 42:1942–1945, 2000.
- [42] Andreas Peckhaus, Alexei Kiselev, Robert Wagner, Denis Duft, and Thomas Leisner. Temperature-dependent formation of nacl dihydrate in levitated nacl and sea salt aerosol particles. *The Journal of Chemical Physics*, 145(24):244503, 2016.
- [43] Robert Wagner, Julia Kaufmann, Ottmar Möhler, Harald Saathoff, Martin Schnaiter, Romy Ullrich, and Thomas Leisner. Heterogeneous ice nucleation ability of nacl and sea salt aerosol particles at cirrus temperatures. *Journal of Geophysical Research: Atmospheres*, 123(5):2841–2860, 2018.



Casa abierta al tiempo

UNIVERSIDAD AUTÓNOMA METROPOLITANA

# ACTA DE EXAMEN DE GRADO

No. 00090

Matrícula: 2171801435

BÚSQUEDA DE ISÓMEROS  
RESTRINGIDOS ESPACIALMENTE.

En la Ciudad de México, se presentaron a las 10:00 horas del día 14 del mes de diciembre del año 2018 en la Unidad Iztapalapa de la Universidad Autónoma Metropolitana, los suscritos miembros del jurado:

DR. MARCELO ENRIQUE GALVAN ESPINOSA  
DR. RODOLFO GOMEZ BALDERAS  
DR. JORGE GARZA OLGUIN

Bajo la Presidencia del primero y con carácter de Secretario el último, se reunieron para proceder al Examen de Grado cuya denominación aparece al margen, para la obtención del grado de:


MAESTRO EN CIENCIAS (QUIMICA)

DE: JUAN JOSE GARCIA MIRANDA

y de acuerdo con el artículo 78 fracción III del Reglamento de Estudios Superiores de la Universidad Autónoma Metropolitana, los miembros del jurado resolvieron:


APROBAR

Acto continuo, el presidente del jurado comunicó al interesado el resultado de la evaluación y, en caso aprobatorio, le fue tomada la protesta.



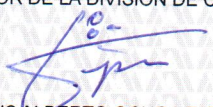

JUAN JOSÉ GARCIA MIRANDA  
ALUMNO

REVISÓ



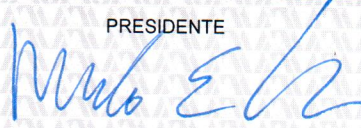
DR. JOSE ANTONIO DE LOS REYES HEREDIA  
SECRETARIO GENERAL

DIRECTOR DE LA DIVISIÓN DE CBI



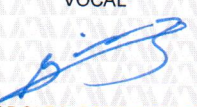
DR. JESUS ALBERTO OCHOA TAPIA

PRESIDENTE




DR. MARCELO ENRIQUE GALVAN ESPINOSA

VOCAL



DR. RODOLFO GOMEZ BALDERAS

SECRETARIO



DR. JORGE GARZA OLGUIN