



**UNIVERSIDAD  
DE MÁLAGA**



**LENGUAJES Y  
CIENCIAS DE LA  
COMPUTACIÓN**  
UNIVERSIDAD DE MÁLAGA

TESIS DOCTORAL

---

# Optimización multi-objetivo en las ciencias de la vida

---

**E.T.S.I. Informática**  
R.D. 99/2011

Autor

**Esteban López Camacho**

Directores

**Dr. José F. Aldana Montes**

Departamento

Lenguajes y Ciencias de la Computación

Universidad de Málaga

**Dr. Antonio J. Nebro Urbaneja**

Departamento

Lenguajes y Ciencias de la Computación

Universidad de Málaga


October 2017





UNIVERSIDAD  
DE MÁLAGA

AUTOR: Esteban López Camacho

 <http://orcid.org/0000-0002-5147-3997>

EDITA: Publicaciones y Divulgación Científica. Universidad de Málaga



Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-SinObraDerivada 4.0 Internacional:

<http://creativecommons.org/licenses/by-nc-nd/4.0/legalcode>

Cualquier parte de esta obra se puede reproducir sin autorización pero con el reconocimiento y atribución de los autores.

No se puede hacer uso comercial de la obra y no se puede alterar, transformar o hacer obras derivadas.

Esta Tesis Doctoral está depositada en el Repositorio Institucional de la Universidad de Málaga (RIUMA): [riuma.uma.es](http://riuma.uma.es)



Los Drs. **José F. Aldana Montes**, Profesor Catedrático del Departamento de Lenguajes y Ciencias de la Computación de la Universidad de Málaga, y **Antonio J. Nebro Urbaneja**, Profesor Titular del Departamento de Lenguajes y Ciencias de la Computación de la Universidad de Málaga.

### Certifican

que **D. Esteban López Camacho**, Ingeniero en Informática por la Universidad de Málaga, España, ha realizado en el Departamento de Lenguajes y Ciencias de la Computación de la Universidad de Málaga, bajo sus direcciones, el trabajo de investigación correspondiente a su Tesis Doctoral titulada

### Optimización multi-objetivo en las ciencias de la vida

Revisado el presente trabajo, estimamos que puede ser presentado al tribunal que ha de juzgarlo, y autorizamos la presentación de esta Tesis Doctoral en la Universidad de Málaga.

En Málaga, octubre de 2017



Firmado:

**Dr. José F. Aldana Montes**  
Profesor Catedrático del Dpto. de Lenguajes y Ciencias  
de la Computación de la Universidad de Málaga y  
**Dr. Antonio J. Nebro Urbaneja**  
Profesor Titular del Dpto. de Lenguajes y Ciencias  
de la Computación de la Universidad de Málaga



UNIVERSIDAD  
DE MÁLAGA

# Acknowledgements

First, I would like to thank my supervisors Prof. José F. Aldana Montes and Prof. Antonio J. Nebro Urbaneja for their guidance and support throughout these years of developing the work that is part of this thesis. I would especially like to thank them for giving me the opportunity of doing a PhD thesis when I didn't think it was possible for me to do it.

I also wish to thank everyone who accepted to be part of my thesis committee, for agreeing so quickly, and making it all so easy for me. In addition I thank my external evaluators for all their insightful corrections that improved my manuscript.

My sincere thanks to Dr. Manuel López Ibañez, Dr. Julia Handl and Dr. Richard Allmendinger for the lovely stay I had in those three months at the University of Manchester, in particular Manuel who I worked with and who helped me in my research.

I would like to mention all the people I have been working with all these years in the Khaos research group, both doctors and students. Most of them I am glad to call friends, as we have shared lots of fun and hard work over the years.

I would also like to thank other members of the Grupo de Ingeniería del Software de la Universidad de Málaga (GISUM) who I know personally and have shared personal experiences with them. I want to thank Lisa Huckfield for all her corrections in all my papers, doing them faster than I thought humanly possible. In addition, I want to thank all the staff at the Ada Byron research centre for our little chats that we had after leaving late from work.

I don't want to forget to thank all my family, especially my parents, my brother, my two little sisters, and even my cat, thanks for always being there and making me smile in difficult times.

And last but not least, I would like to thank above all my girlfriend (soon to be Dr also) María Jesús García Godoy. The most hard working person I have ever known and the only one I know who can achieve whatever she puts her mind to. Without her support (both personal and professional) I would never have been able to complete this PhD thesis.



# Contents

<b>Resumen</b>	<b>1</b>
<b>1 Introduction</b>	<b>11</b>
1.1 Motivation	11
1.2 Objectives and phases	12
1.3 Thesis contributions	14
1.4 Thesis organization	15
<b>2 Metaheuristics</b>	<b>17</b>
2.1 Definition	17
2.2 Classification	22
2.2.1 Trajectory-based metaheuristics	22
2.2.2 Population-based metaheuristics	24
2.3 Multi-objective optimization metaheuristics	25
2.3.1 Basic concepts	26
2.3.2 Objectives in MOPs resolution	29
2.3.3 Design aspects	29
2.4 Statistical evaluation of results	32
2.4.1 Quality indicators	32
2.4.2 Statistical performance assessment	34
<b>3 The Molecular Docking Problem</b>	<b>37</b>
3.1 Molecular docking: Definition and biological significance	37
3.2 Problem formulation	38
3.2.1 Mono-objective optimization	39
3.2.2 Multi-objective optimization	40
3.3 Review of the State-of-the-Art	41
<b>4 Published Work</b>	<b>45</b>
4.1 List with Research Contributions	45
4.2 Summary of the articles that support the thesis	46
4.2.1 jMetalCpp: optimizing molecular docking problems with a C++ metaheuristic framework	46
4.2.2 Solving molecular flexible docking problems with metaheuristics: a comparative study	47
4.2.3 A new multi-objective approach for molecular docking based on RMSD and binding energy	47
4.2.4 A study of archiving strategies in multi-objective PSO for molecular docking	48



---

4.3	Summary of other publications related to this thesis . . . . .	49
4.4	Copies of the articles that support the thesis . . . . .	49
<b>5</b>	<b>Conclusions and Future Work</b>	<b>55</b>
5.1	Conclusions . . . . .	55
5.2	Future Work . . . . .	56
	<b>Bibliography</b>	<b>59</b>
	<b>List of Figures</b>	<b>67</b>



# Resumen

Las herramientas de acoplamiento molecular han llegado a ser bastante eficientes en el descubrimiento de fármacos y en el desarrollo de la investigación de la industria farmacéutica. Estas herramientas se utilizan para elucidar la interacción de una pequeña molécula (ligando) y una macromolécula (diana) a un nivel atómico para determinar cómo el ligando interactúa con el sitio de unión de la proteína diana y las implicaciones que estas interacciones tienen en un proceso bioquímico dado. El progreso experimentado en las técnicas de acoplamiento molecular ha estado a la par con los avances en los métodos espectroscópicos biomoleculares como la cristalografía de rayos X y la resonancia magnética nuclear (NMR), que han sido muy importantes en el dominio de la biología estructural. Estas técnicas han permitido determinar más de 100.000 estructuras tridimensionales de proteínas que pueden tener un papel importante en las rutas de bioseñalización. La base de datos de *Protein Data Bank* actualmente contiene 130.807 estructuras *PDB* de múltiples organismos, la mayoría de ellos habiendo sido obtenidos a través de cristalografía de rayos X (117.083), NMR (11.766) y cristalografía de electrón (1.545). En este contexto, en que existen miles de estructuras *PDB* almacenadas que pueden ser candidatas a ser analizadas como dianas terapéuticas, las técnicas de acoplamiento molecular juegan un papel importante en el diseño de nuevos fármacos analizando cómo estos interactúan con las dianas terapéuticas a nivel molecular.

En el desarrollo computacional de las herramientas de acoplamiento molecular los investigadores de este área se han centrado en mejorar los componentes que determinan la calidad del *software* de acoplamiento molecular: 1) la función objetivo y 2) los algoritmos de optimización. La función objetivo de energía se encarga de proporcionar una evaluación de las conformaciones entre el ligando y la proteína calculando la energía de unión, que se mide en kcal/mol. Según la literatura, existen varios tipos de funciones objetivo de energía pero la mayoría de ellos están basados en campos de fuerza que estiman la energía libre de unión de la conformación ligando-receptor, teniendo en cuenta términos como las conformaciones del ligando interno, las conformaciones proteína-ligando y los efectos solventes. En esta memoria, hemos usado AutoDock, ya que es una de las herramientas de acoplamiento molecular más citada y usada, y cuyos resultados son muy precisos en términos de energía y valor de RMSD (desviación de la media cuadrática). Además, se ha seleccionado la función de energía de AutoDock versión 4.2, ya que permite realizar una mayor cantidad de simulaciones realistas incluyendo flexibilidad en el ligando y en las cadenas laterales de los aminoácidos del receptor que están en el sitio de unión.

En esta tesis se han utilizado algoritmos de optimización para mejorar los resultados de acoplamiento molecular de AutoDock 4.2, el cual minimiza la energía libre de unión final que es la suma de todos los términos de energía de la función objetivo de energía. Dado que encontrar la solución óptima en el acoplamiento molecular es un problema de gran complejidad y la mayoría de las veces imposible, se suelen utilizar algoritmos no exactos como las metaheurísticas, para así obtener soluciones lo suficientemente buenas en un tiempo razonable.

Por todo lo anterior, como trabajo preliminar se puede analizar el rendimiento de un conjunto de metaheurísticas mono-objetivo de carácter general (en su diseño canónico) para determinar si es posible obtener mejores valores de la función objetivo que con aquellas técnicas proporcionadas



por AutoDock. Según la literatura consultada, existen pocos estudios que tengan en cuenta la flexibilidad en sus experimentos de acoplamiento molecular. Es por ello, que se aplicó flexibilidad tanto en los ligandos como en las cadenas laterales de las macromoléculas. De esta manera, es posible determinar el rendimiento de los algoritmos atendiendo si el espacio de búsqueda es diferente o no dependiendo del tamaño del ligando y su flexibilidad.

Dados los interesantes resultados obtenidos por Janson *et al.* (2008) en el que se minimizaron dos objetivos, la energía intermolecular ( $E_{inter}$ ) y la intramolecular ( $E_{intra}$ ), se puede ver que el problema puede ser formulado usando dos objetivos contrapuestos, dando lugar a un problema de optimización multi-objetivo. Después de revisar el resto de la literatura sobre los distintos enfoques multi-objetivo para resolver el acoplamiento, se observó que todos los estudios estaban basados en la función de energía de AutoDock 3.0 (una versión anterior a AutoDock 4.2), que no aplica flexibilidad a las cadenas laterales de los aminoácidos del receptor y, por lo tanto, solamente se hicieron simulaciones siendo rígidas la macromolécula y el ligando o con flexibilidad sólo en el ligando. Estos estudios también habían sido realizados sobre un conjunto pequeño de problemas, con lo que estudios con un mayor número de complejos flexibles podrían dar lugar a resultados muy interesantes.

Los estudios multi-objetivo anteriormente propuestos no han considerado anteriormente guiar la búsqueda usando uno de los objetivos cuando la estructura del ligando co-cristalizado es conocida, lo que podría completar la función de energía tradicional. Podrían planificarse nuevos enfoques utilizando este hecho como punto de partida. También se hipotetiza que este enfoque puede ser útil en aquellos estudios *in silico* que tengan que ver con la selección de nuevos compuestos anticancerígenos para dianas terapéuticas que sean resistentes a múltiples fármacos.

El objetivo principal de esta tesis es explorar un enfoque al problema del acoplamiento molecular que pueda dar lugar a un conjunto más amplio de soluciones dependiendo de los objetivos seleccionados. Con esto, se intenta promover el uso de estas nuevas técnicas en lugar de depender en los algoritmos más comúnmente usados. Como trabajo previo, se aplican nuevas técnicas mono-objetivo que puedan proporcionar resultados de mayor calidad que las técnicas usualmente aplicadas.

Las fases que se siguieron en el desarrollo de esta tesis fueron las siguientes:

1. Exploración del estado del arte actual sobre los estudios de acoplamiento molecular e investigación de las diferentes herramientas usadas y análisis del código de AutoDock 4.2, dado que es la más citada y popular entre la comunidad científica. Se observaron las técnicas de optimización que proporcionaba AutoDock y se estudió la posibilidad de añadir nuevos algoritmos que mejoraran los resultados obtenidos.
2. Para conseguir este objetivo, en lugar de intentar incorporar los nuevos algoritmos directamente en el código fuente de AutoDock, se utilizó un *framework* orientado a la resolución de problemas de optimización con metaheurísticas. Concretamente, se usó jMetal, que es una librería de código libre basada en Java. Ya que AutoDock está implementado en C++, se desarrolló una versión en C++ de jMetal. De esta manera, se consiguió integrar ambas herramientas (AutoDock 4.2 y jMetal) para optimizar la energía libre de unión entre compuesto químico y receptor.
3. Después de disponer de una amplia colección de metaheurísticas implementadas en jMetal-Cpp, se realizó un detallado estudio en el cual se aplicaron un conjunto de metaheurísticas para optimizar un único objetivo minimizando la energía libre de unión, el cual es el resultado de la suma de todos los términos de energía de la función objetivo de energía de AutoDock 4.2. Por lo tanto, cuatro metaheurísticas tales como dos variantes de algoritmo genético gGA (Algoritmo Genético generacional) y ssGA (Algoritmo Genético de estado estacionario), DE (Evolución Diferencial) y PSO (Optimización de Enjambres de Partículas)

fueron aplicadas para resolver el problema del acoplamiento molecular. Esta fase se dividió en dos subfases en las que usamos dos conjuntos de instancias diferentes, utilizando como receptores HIV-proteasas con cadenas laterales de aminoácidos flexibles y como ligandos inhibidores HIV-proteasas flexibles. El primer conjunto de instancias se usó para un estudio de configuración de parámetros de los algoritmos y el segundo para comparar la precisión de las conformaciones ligando-receptor obtenidas por AutoDock y AutoDock+jMetalCpp.

4. La siguiente fase implicó aplicar una formulación multi-objetivo para resolver problemas de acoplamiento molecular dados los resultados interesantes obtenidos por Janson *et al.* (2008) en que dos objetivos como la energía intermolecular ( $E_{inter}$ ) y la energía intramolecular ( $E_{intra}$ ) fueron minimizados. Por lo tanto, se comparó y analizó el rendimiento de un conjunto de metaheurísticas multi-objetivo mediante la resolución de complejos flexibles de acoplamiento molecular minimizando la  $E_{inter}$  y la  $E_{intra}$ . Estos algoritmos fueron: NSGA-II (Algoritmo Genético de Ordenación No dominada) y su versión de estado estacionario (ssNSGA-II), SMPSO (Optimización Multi-objetivo de Enjambres de Partículas con Modulación de Velocidad), GDE3 (Tercera versión de la Evolución Diferencial Generalizada), MOEA/D (Algoritmo Evolutivo Multi-Objetivo basado en la Decomposición) y SMS-EMOA (Optimización Multi-objetivo Evolutiva con Métrica S). Estos algoritmos han obtenido rendimientos satisfactorios en una amplia variedad de problemas de optimización, sin embargo, nunca se han usado con anterioridad para resolver problemas de acoplamiento molecular a excepción del algoritmo NSGA-II.
5. Después de probar enfoques multi-objetivo ya existentes, se probó uno nuevo. En concreto, el uso del RMSD como un objetivo para encontrar soluciones similares a la de la solución de referencia. Se replicó el estudio previo usando este conjunto diferente de objetivos.
6. Por último, se analizó de forma detallada el algoritmo que obtuvo mejores resultados en los estudios previos. En concreto, se realizó un estudio de variantes del SMPSO minimizando la  $E_{inter}$  y el RMSD. SMPSO aplica un mecanismo de limitación de la velocidad de las partículas para impedir el movimiento de éstas en las regiones de búsqueda ajenas a los rangos de los problemas. Este algoritmo usa un archivo externo para almacenar las soluciones no dominadas según a su distancia de *crowding*. También se usa este archivo en el mecanismo de selección del líder. Este estudio proporcionó algunas pistas sobre cómo nuevos algoritmos basados en SMPSO pueden ser adaptados para mejorar los resultados de acoplamiento molecular para aquellas simulaciones que involucren ligandos y receptores flexibles.

Resumiendo, esta tesis realiza las siguientes contribuciones:

- La implementación de un framework metaheurístico en C++ (jMetalCpp), versión del ampliamente usado framework en Java jMetal, para resolver problemas de optimización y para su posterior distribución pública entre la comunidad científica.
- La inclusión de técnicas metaheurísticas de jMetalCpp en la herramienta de acoplamiento molecular AutoDock, y su distribución pública para incrementar las posibilidades a los usuarios de ámbito biológico cuando resuelvan el problema del acoplamiento molecular.
- La demostración de que el uso de técnicas de optimización mono-objetivo diferentes aparte de aquéllas ampliamente usadas en las comunidades de acoplamiento molecular podría dar lugar a soluciones de mayor calidad. En nuestro caso de estudio, el algoritmo de evolución diferencial obtuvo mejores resultados que aquellos obtenidos por AutoDock.

- La propuesta de diferentes enfoques multi-objetivo para resolver el problema del acoplamiento molecular, tales como la decomposición de los términos de la energía de unión o el uso del RMSD como un objetivo.
- La demostración del SMPSO, una metaheurística de optimización multi-objetivo de enjambres de partículas, como una técnica remarcable para resolver problemas de acoplamiento molecular cuando se usa un enfoque multi-objetivo, obteniendo incluso mejores soluciones que las técnicas mono-objetivo.
- La presentación de dos nuevas variantes de SMPSO. La primera es SMPSOD, una aproximación sin archivo, que está inspirada en el MOEA/D. La segunda es SMPSOC, que usa la nueva similaridad del coseno para calcular el estimador de densidad.

El problema del acoplamiento molecular es una de las técnicas usadas en el proceso de diseño de fármacos basados en estructura. Este proceso consiste en estudios *in silico* para determinar compuestos químicos que puedan ser posibles candidatos para dianas terapéuticas. Son muchas las técnicas computacionales que se utilizan adicionalmente al acoplamiento molecular, algunas de éstas son dinámica molecular y screening virtual basado en estructuras. Aparte del proceso de diseño de fármacos basados en estructura, existe otro basado en el diseño de estructuras basado en ligandos que consiste en testear librerías de compuestos químicos activos para la detección de posibles dianas terapéuticas.

Como anteriormente se ha mencionado, el principal objetivo del problema de acoplamiento molecular es encontrar la conformación ligando-receptor cuya energía de unión sea mínima. Esta energía se computa utilizando la función de energía del *software* de acoplamiento molecular. La solución que representa la interacción ligando-receptor está codificada por un vector de números reales de tamaño  $n+7$  en el cual los tres primeros valores corresponden a los valores de los tres ejes ( $x, y, z$ ) en el espacio de coordenadas Cartesianas, los siguientes cuatro valores corresponden a la orientación ligando/macromolécula, y los  $n$  valores restantes son los ángulos dihedrales de torsión para el ligando y las cadenas laterales de los aminoácidos del receptor. En los experimentos realizados para esta tesis doctoral, se aplicó una metodología basada en el tamaño de malla implementada en AutoDock versión 4.2. La malla corresponde al espacio de búsqueda en el que se realiza los cálculos ligando-macromolécula en las simulaciones de acoplamiento molecular. Los parámetros utilizados fueron para ( $x, y, z$ ) 120 y 0,375Å de espacio de malla. Estos parámetros para la malla fueron suficientes para abarcar toda la superficie molecular de la macromolécula. Sin embargo, estos parámetros pueden ser modificados por el experto en acoplamiento molecular aumentando o disminuyendo tales parámetros en el espacio de malla.

Para el enfoque de optimización mono-objetivo, se minimizó el valor de la energía libre de unión, que se mide en kcal/mol. Cuanto más pequeño es este valor, más estable es el complejo ligando-receptor en términos energéticos. Atendiendo a la función de energía proporcionada por AutoDock, este valor es el resultado de la suma de la diferencia los estados de unión y no unión del ligando, receptor y del complejo ligando-receptor. Cada par de términos de evaluación incluyen evaluaciones de dispersión/repulsión, enlaces de van der Waals, puentes de hidrógeno, fuerzas de torsión e interacciones electrostáticas y de solvatación.

Para el enfoque de optimización multi-objetivo, en primer lugar, se optimizaron dos energías: la  $E_{inter}$  y  $E_{intra}$ . La primera energía representa la diferencia entre los estados de unión y desunión del ligando-receptor o el estado energético del complejo ligando-receptor. La segunda energía representa los estados de unión y desunión del ligando y el receptor, respectivamente. Esta energía involucra la deformidad desde el punto de vista de energía de los elementos de interacción durante las simulaciones de acoplamiento molecular. Esta estrategia de optimización multi-objetivo es muy útil en aquellos casos en los que el experto tiene que elegir una solución en el conjunto de

soluciones obtenidas en la que el ligando sea más estable en términos de energía o bien, otra solución en la que el complejo ligando-receptor es más estable energéticamente.

En una segunda estrategia, se optimizó la  $E_{inter}$  y el valor de RMSD calculado a partir del ligando co-cristalizado y el computado. Este valor mide la calidad de los resultados obtenidos en las simulaciones del acoplamiento molecular. RMSD básicamente es una medida de la distancia media entre las coordenadas atómicas ( $x, y, z$ ) de la estructura del ligando co-cristalizado y el ligando computado. Esta medida tiene en cuenta la simetría, la simetría parcial (por ejemplo, la simetría de una parte rotable de la molécula) y la simetría más próxima. La comunidad científica usa el límite de 2Å para distinguir entre resultados más o menos exactos. Esta medida es muy útil en aquellos casos en los que la estructura de ligando es conocida, es decir, la estructura cristalográfica del ligando con respecto al receptor está disponible en las bases de datos que almacenan estructuras cristalográficas (como la base de datos PDB). Es importante mencionar, que una estructura computada con un valor RMSD de 0Å no es la mejor solución que se podría obtener ya que el receptor puede tener otros sitios de unión no conocidos y estos podrían ser interesantes desde un punto de vista farmacológico.

## Trabajos publicados

El trabajo realizado en esta tesis ha dado lugar a varias publicaciones y divulgaciones científicas. Específicamente, cuatro artículos han sido publicados en revistas indexadas en el *Journal of Citation Report* (JCR) del *Institute of Scientific Information*. Además, otros cuatro artículos han sido publicados en congresos. Dos de ellos se publicaron en congresos internacionales y los otros dos en congresos nacionales. Para ver más detalle, véase el Capítulo 4.

A continuación se resumen los artículos que avalan esta tesis. Todos estos artículos están relacionados con la aplicación de optimizaciones tanto mono-objetivo como multi-objetivo para resolver el problema del acoplamiento molecular. En el primer artículo se describió la integración de AutoDock y jMetal y su aplicación en el acoplamiento molecular. En el segundo artículo publicado, se realiza un estudio comparando las técnicas mono-objetivo usando un conjunto de instancias flexibles. En el tercer estudio, se aplica un conjunto de metaheurísticos multi-objetivo para optimizar dos objetivos, guiando al algoritmo en su búsqueda de las mejores soluciones.

### **jMetalCpp: optimizing molecular docking problems with a C++ meta-heuristic framework**

En este artículo se presentó jMetalCpp, la versión C++ de jMetal, el framework de metaheurísticas originalmente programado en Java. También se presenta la combinación de este software con el ampliamente usado AutoDock. Como se ha mencionado anteriormente, ambos paquetes software fueron publicados en la web para ser libremente usados por la comunidad científica.

### **Solving molecular flexible docking problems with metaheuristics: a comparative study**

En este trabajo, se demostró que DE (jMetal) obtuvo los mejores resultados en 67 de las 75 instancias estudiadas, seguido por LGA (AutoDock que consiguió los mejores resultados en las ocho instancias restantes (1B6L, 1BDL, 1HEF, 1HIV, 1HPO, 1K6C, 1Z1H and 1ZIR). Estos resultados fueron proporcionados con confianza estadística ( $\alpha = 0.05$ ) ya que se aplicó una serie de tests estadísticos no paramétricos. En concreto, se calcularon los *ranking de Friedman* y los tests multicomparativos de Holm, y mostraron que el DE consiguió un mejor rendimiento estadísticamente que el resto de los algoritmos analizados. Este hecho es remarcable que los algoritmos de AutoDock

están específicamente diseñados para resolver problemas de acoplamiento molecular. También se observó que el DE mostraba un comportamiento de convergencia más lento, aunque tendiendo a soluciones más exitosas que sus competidores. Sin embargo, gGA demostró tener una rápida convergencia, y también consiguió soluciones de alta calidad, así que este algoritmo podría ser una buena opción cuando se buscara una alternativa que proporcionara soluciones lo suficientemente buenas en un tiempo de cómputo menor.

## A new multi-objective approach for molecular docking based on RMSD and binding energy

Este trabajo fue presentado en la 3ª *International Conference on Algorithms for Computational Biology* (AlCoB 2016), que se celebró en Trujillo (España) en junio de 2016. Dicho trabajo derivó de la idea de aplicar un enfoque de optimización multi-objetivo para resolver problemas de acoplamiento molecular. Al principio, la estrategia que se siguió fue la descomponer la energía final de unión (el objetivo a minimizar en el trabajo anterior) en varias componentes, concretamente las energías intra e intermolecular. Posteriormente, se decidió usar como objetivos la misma energía tomada como objetivo en el estudio mono-objetivo y el RMSD. Estos conceptos están explicados en más detalle en la Sección 3.2.2.

El  $I_{HV}$  es la suma del volumen contribuido de cada punto de un frente con respecto a un punto de referencia, así que cuanto más alto el grado de convergencia y diversidad de un frente, más alto será el valor del hipervolumen. Según estos resultados, SMPSO consiguió los mejores valores de  $I_{HV}$  en los 11 problemas, siendo MOEA/D la segunda técnica que obtuvo mejores resultados. Es importante destacar que muchos algoritmos obtuvieron un valor de  $I_{HV}$  igual a cero. Esto ocurre cuando todos los puntos de los frentes producidos están situados más allá de los límites del punto de referencia. Este hecho se da en la mayoría de los problemas en todos los algoritmos a excepción de SMPSO, lo que lleva a pensar que se está afrontando un problema de optimización de gran complejidad. SMPSO también consigue el mejor rendimiento según el indicador  $I_{\epsilon+}$  (en este caso, cuanto más bajo es el valor, mejor es). SMPSO alcanza los mejores valores para todas las instancias exceptuando el 1HTF donde consiguió el segundo mejor valor. MOEA/D (que fue el que obtuvo el mejor resultado para la instancia 1HTF) alcanzando los segundos mejores valores para 9 instancias. GDE3 consiguió el segundo mejor valor en la instancia restante (1HPX), mientras que NSGA-II obtuvo los peores resultados para todas las instancias.

Después de que se presentara este trabajo, se invitó a ser substancialmente extendido y enviado al número especial de la revista *IEEE/ACM Transactions on Computational Biology and Bioinformatics* (TCBB, Factor de impacto JCR 2014: 1.438, Cuartil Q1). Hasta el día de hoy, aún sigue en revisión.

## A study of archiving strategies in multi-objective PSO for molecular docking

Este artículo presentó la variante denominada SMPSOC, que se caracteriza por el uso de la similitud por coseno cuando se calcula el valor de densidad de cada punto en el frente de soluciones. La variante SMPSOD también fue presentada en este artículo por primera vez. Es un enfoque sin archivo, que está implementado como una versión agregativa de SMPSO inspirado por MOEA/D.

Según el indicador  $I_{HV}$ , SMPSO<sub>hv</sub> obtuvo los mejores resultados para las 11 instancias, mientras que SMPSOD tuvo los segundos mejores en 6 instancias, SMPSOC en tres y el SMPSO original en dos, respectivamente. De igual forma, SMPSO<sub>hv</sub> obtuvo de nuevo los mejores resultados en las 11 instancias según el indicador  $I_{\epsilon+}$ . Los segundos mejores valores fueron conseguidos por SMPSOD en 7 instancias, por el SMPSO original en tres y por SMPSOC en una instancia, respectivamente.



## Conclusiones y trabajos futuros

Al abordar problemas de acoplamiento molecular, las técnicas disponibles para resolverlos no han cambiado en los últimos años. Como estos problemas pueden ser formulados como problemas de optimización multiobjetivos, nuestra intención fue la de estudiar y proporcionar un conjunto de técnicas metaheurísticas modernas para resolverlas. Como la herramienta de acoplamiento molecular más utilizada (AutoDock) fue programada en C++, nos embarcamos en la tarea de crear una versión del *framework* metaheurístico jMetal en este lenguaje: jMetalCpp. De esta manera, hemos proporcionado a la comunidad de investigación una herramienta potente y de código abierto que se puede utilizar libremente.

La implementación del *framework* jMetalCpp proporciona ventajas a los investigadores, tanto en el descubrimiento de fármacos como en otros dominios de las ciencias de la vida, que están interesados en disponer de técnicas más modernas que les ayudarán a resolver diferentes problemas como el acoplamiento molecular. Ya hemos demostrado que existen diferentes técnicas aparte de las que se utilizan comúnmente para resolver problemas de acoplamiento molecular y que pueden conducir a resultados de mayor calidad. La inclusión de jMetalCpp en la ampliamente utilizada herramienta AutoDock proporciona a otros investigadores una colección de metaheurísticas y herramientas adicionales a las que ya están incluidas en AutoDock. También proporciona una estructura fácil para usuarios más avanzados con habilidades de programación en C++ para incorporar sus propias técnicas para resolver problemas de acoplamiento molecular. Esta herramienta está disponible *online* y ya ha sido descargada por investigadores de diferentes partes del mundo. El *framework* jMetalCpp independiente también está disponible para los investigadores que quieran utilizarla para resolver problemas de optimización de otros dominios. Se ha descargado cientos de veces de todo el mundo y hemos estado en contacto con personas que querían contribuir al código añadiendo sus propias herramientas y algoritmos, y utilizarlo en sus propios trabajos de investigación.

Usando AutoDock+jMetal, se realizó un estudio utilizando metaheurísticas mono-objetivo donde incluimos más algoritmos (aparte de los ya incluidos por AutoDock) para resolver un gran *benchmark* de complejos proteína-ligando. El estudio se llevó a cabo teniendo los mismos parámetros de configuración que comúnmente se utilizaron en las publicaciones de AutoDock. Probamos que otras metaheurísticas mono-objetivo podrían llevar a resultados de mayor calidad. En nuestro caso, el algoritmo de evolución diferencial demostró ser un mejor candidato a la hora de resolver problemas de acoplamiento molecular.

Cuando se abordan problemas de acoplamiento molecular, es común resolverlos adoptando un enfoque mono-objetivo. Sin embargo, cuando se utiliza un enfoque multi-objetivo, un conjunto de soluciones se devuelve al final de una ejecución en lugar de una única solución. Este conjunto de soluciones ofrece al usuario final varias posibilidades desde donde escoger dependiendo del peso que quiere dar a cada uno de los objetivos de optimización. Por lo tanto, hemos considerado dos enfoques multi-objetivos diferentes en nuestros estudios. La primera se basó en la descomposición de la energía de unión final (la función objetivo que es minimizada por los algoritmos mono-objetivo) en varios componentes. Se seleccionaron las energías intra e intermoleculares como objetivos de optimización. Esto resultó en un conjunto de soluciones en las que el usuario final podría elegir dependiendo de la importancia que le da a cada una de las energías.

La otra formulación multi-objetivo utilizó el mismo objetivo que la formulación mono-objetivo (la energía de unión) y el RMSD. El uso del RMSD como objetivo para guiar la búsqueda es útil en aquellos casos típicos en los que el sitio activo de una diana terapéutica dada muta y lo hace resistente a múltiples fármacos. Utilizando este enfoque, se devuelve un amplio conjunto de soluciones, que pueden seleccionarse de acuerdo con el peso de la RMSD y la energía de unión, en lugar de centrarse únicamente en los valores de energía. Se realizó un primer estudio utilizando cuatro algoritmos multi-objetivo: NSGA-II, SMPSO, GDE3 y MOEA/D. En este experimento, se

seleccionó un conjunto de 11 complejos de proteína-ligando heterogéneos con ligandos y receptores flexibles como instancias del problema. SMPSO proporcionó el mejor rendimiento general según los dos indicadores de calidad utilizados ( $I_{HV}$  y  $I_{\epsilon+}$ ) y para las instancias moleculares estudiadas, siendo MOEA/D el algoritmo con los segundos mejores valores. Así mismo, desde un punto de vista mono-objetivo, las soluciones obtenidas de SMPSO fueron mejores que las obtenidas por el algoritmo LGA de AutoDock. Esto es bastante notable ya que SMPSO es un algoritmo de optimización de propósito general, mientras que LGA está específicamente adaptado para hacer frente al problema de acoplamiento molecular. Finalmente, es interesante notar que SMPSO convergió a la región del frente que minimiza más el objetivo RMSD, mientras que MOEA/D colocó sus soluciones en la región opuesta de los frentes generados de soluciones no dominadas.

A partir de los resultados obtenidos en el último estudio, se llevó a cabo un nuevo experimento en el que se probarían varias variantes SMPSO con diferentes estrategias de archivo. Las variantes seleccionadas fueron: SMPSO<sub>hv</sub>, SMPSOD y SMPSOC. El SMPSO original y OMOPSO (el algoritmo del que SMPSO se inspiró) también se incluyeron en la comparación. El estudio multi-objetivo anterior se replicó utilizando estos seis algoritmos y las mismas configuraciones que antes. De acuerdo con nuestros dos indicadores habituales de calidad ( $I_{HV}$  y  $I_{\epsilon+}$ ), SMPSO<sub>hv</sub> demostró obtener los mejores valores, seguido de SMPSOD, SMPSOC y SMPSO. La primera variante obtuvo el mejor  $I_{HV}$  al realizar un método de selección de líder de aquellas soluciones no dominadas (del archivo externo) con las mayores contribuciones de hipervolumen, las cuales parecían ser responsables de los mejores valores de diversidad y convergencia en esta comparación. OMOPSO mostró resultados moderados, aunque alcanzando superar las soluciones atípicas para algunos casos. Cabe destacar que la variante SMPSOD fue capaz de cubrir el frente de referencia con soluciones no dominadas en los extremos de los dos objetivos (valores bajos de energía y bajos valores de RMSD, respectivamente).

La línea de estudio llevada a cabo en esta tesis nos ha llevado a planificar varios trabajos posibles. Por un lado, algunas de los trabajos futuros surgen de la idea de continuar el problema abordado (acoplamiento molecular) y todavía se centran en tratar de mejorar la calidad de los resultados obtenidos. Por otro lado, las nuevas líneas de investigación podrían partir de los conocimientos obtenidos en los experimentos anteriores y podrían considerarse como “ramas” de este trabajo.

El primer trabajo planeado está relacionado con nuestro primer estudio multi-objetivo, el cual obtuvo que al unir las soluciones generadas a partir de los algoritmos SMPSO y MOEA/D se cubría todo el frente de Pareto. Como trabajo futuro, esto nos llevó a pensar que una implementación híbrida de SMPSO y MOEA/D nos proporcionaría un conjunto más amplio de soluciones que cubriría el frente de referencia con soluciones no dominadas en los dos extremos de los objetivos. Los resultados obtenidos por SMPSOD en el segundo estudio multi-objetivo nos animaron a continuar este plan de trabajo.

En relación con el diseño del algoritmo híbrido, planeamos implementar e incluir en jMetalCpp algunos operadores específicamente diseñados para el problema de acoplamiento molecular. Hasta ahora, todas las técnicas metaheurísticas que hemos utilizado en nuestros estudios utilizan operadores de variación de propósito general, por lo que es natural llegar a la conclusión de que si las técnicas utilizadas para resolver el acoplamiento molecular están específicamente diseñadas para este problema concreto, podríamos obtener una mayor calidad de soluciones.

Otra contribución a la comunidad científica que queremos explorar es la creación de un servicio Web que proporcione las mismas herramientas que jMetalCpp integra en AutoDock. Este servicio Web permitiría ejecuciones de acoplamiento molecular utilizando todas las metaheurísticas de jMetalCpp en un complejo proteína-ligando (seleccionable de todos nuestros conjuntos anteriores o cargado por el usuario). Esta idea surgió ya que algunos usuarios con un perfil más biológico podrían tener problemas tratando de compilar y ejecutar nuestra herramienta AutoDock+jMetal.

Finalmente, como una idea más general, querríamos usar nuestro *framework* jMetalCpp independiente para resolver otros problemas en las ciencias de la vida, y no estar restringidos a



acoplamiento molecular. Nuestra herramienta es lo suficientemente abstracta para incluir más algoritmos y ser utilizada para resolver otros problemas de optimización de diferentes dominios. En concreto, la predicción de estructura terciaria de proteínas es un candidato muy adecuado donde aplicar el conjunto de técnicas de optimización de jMetalCpp.



# Chapter 1

## Introduction

Molecular docking tools have become a powerful tool for drug discovery and development in research-based pharmaceutical industry [1, 2, 3]. The molecular docking approach is used to elucidate the interaction of a small molecule (ligand) and a macromolecule (target) at the atomic level to characterize how the ligand interacts to the protein target's binding site and the implications that these interactions have in a given biochemical process. The progress in the molecular docking techniques have been hand-in-hand with advances in biomolecular spectroscopic methods such as X-ray crystallography and nuclear magnetic resonance (NMR), which are very important in the domain of structural biology [4]. These techniques have allowed to determine more than 100,000 tridimensional structures of proteins that can have an important role in biosignaling pathways. The Protein Data Bank database currently contains 130,807 PDB structures from multiple organisms [5], with most of them having been obtained through X-ray crystallography (117,083), NMR (nuclear magnetic resonance) (11,766) and electron crystallography (1,545). In this context in which there are thousands of PDB structures stored which can be candidates to be analyzed as therapeutic targets, the molecular docking techniques play an important role in the design of new drugs by analyzing how suitable drugs interact to therapeutic targets.

### 1.1 Motivation

In the computational development of the molecular docking tools, researchers in this area have focused on improving the components that determine the quality of the docking software: 1) the scoring function and 2) the optimization algorithm. The energy score function performs the evaluation of the conformations between the ligand and the protein by calculating the binding energy, which is measured in kcal/mol. According to the literature, there are several types of energy score functions but most of them are physics-based molecular mechanics force fields that estimate the final free binding energy of the ligand-receptor conformation considering terms as internal ligand conformations, protein-ligand conformations and solvent effects. In this dissertation, we have used AutoDock because it is one of the most popular and cited molecular docking tools whose docking results are very accurate in terms of energy and RMSD score (Root Mean Square Distance) [6]. Furthermore, we have selected the energy scoring function of AutoDock version 4.2 [2] as it allows to do more realistic simulations by including flexibility in the ligand and the side-chains of the receptor's aminoacids involved in the binding site.

In this thesis, we have focused on the algorithm optimization to improve the molecular docking results from AutoDock 4.2, which minimizes the final free binding energy that is the result of the sum of all energy terms from the AutoDock 4.2 energy scoring function. As finding the optimal



solution in molecular docking is a very complex problem and most of the time impossible, we use non exact algorithms like metaheuristics, so we can obtain good enough solutions in a feasible time. This leads us to have the following motivations for our studies:

- As preliminary work, analyzing the performance of a set of general-purpose single-objective metaheuristics (in their canonical design) to determine if they can lead to better scoring values compared to the techniques already provided by AutoDock 4.2.
- According to the reviewed literature, there are few studies that involve flexibility in the molecular docking experiments. So, the instances that were used to perform the experiments include a wide range of ligands' size. This allows to do analyses of the algorithms' performance taking into account if the search space is different or not depending on the ligand's size or/and its flexibility.

Given the interesting results obtained by Janson *et al.* [7] in which two objectives like the intermolecular ( $E_{inter}$ ) energy and the intramolecular ( $E_{intra}$ ) energy were minimized, it could be seen that the problem could be formulated using two contrary objectives, leading to a multi-objective optimization problem. Therefore, some additional motivations arise:

- After reviewing the literature corresponding to the application of the multi-objective approaches to solve the molecular docking, we concluded that all the studies are based on the AutoDock 3.0 energy function (an older version than AutoDock 4.2), which does not apply flexibility to the receptor's aminoacid side chains and therefore, rigid ligand-rigid macromolecule or flexible ligand-rigid macromolecule are the only docking simulations that can be carried out.
- We also noticed that these multi-objective approaches have been applied to a very reduced set of ligand-receptor problems. So, the studies performed in this thesis include a larger set of flexible complexes with also different sizes that can lead to interesting conclusions.
- The multi-objective approaches proposed in the literature do not consider guiding the search with a new objective when the co-crystallized ligand is known, which could complement the traditional energy function. New multi-objective approaches could be made taking this as a starting point.
- We also hypothesized that this approach could be useful in those studies *in silico* that involve to select new anticancer compounds for therapeutic targets that are multidrug resistant. Therefore, we have applied it to solve molecular docking problems that involve multi-drug resistant targets that can mutate in patients with lung cancer. These mutations make these targets resistant to drugs, which previously were used in the patients' standard treatment.

## 1.2 Objectives and phases

The purpose of this thesis is to explore a multi-objective approach to the molecular docking problem that could lead to a broader set of solutions depending on the selected objectives. We expect to promote the use of these new techniques instead of relying on the more commonly used algorithms. As a previous work, we also intent to apply new single-objective metaheuristic techniques that provides higher quality results than those obtained from the usual techniques applied when solving molecular docking problems.

The specific objectives of this work can be enumerated in the following points:

- Apply a set of metaheuristics for optimizing a single objective by minimizing the final free binding energy that is the result of the sum of all energy terms from the AutoDock 4.2 energy scoring function.
- Perform an algorithm's parameters analysis and an algorithm convergence behavior study, which will increase the value of that study as there are no previous studies in the reviewed literature.
- Use a set of multi-objective metaheuristics to perform a complete analysis to the molecular docking problem. The algorithms chosen should correspond to a varied set of modern multi-objective techniques in the state of the art, performing different learning procedures, and therefore inducing different behaviors in terms of convergence, diversity, and scalability.
- Define a new multi-objective strategy for molecular docking by minimizing the RMSD score in order to guide the search of results.

To carry out these objectives, the following phases have been followed:

1. We explored the current state-of-the-art of studies about molecular docking. We looked into the different tools used and we took the decision to review the code and functionality of AutoDock 4.2 as it was the most common used tool in molecular docking by the biological scientific community. We studied the optimization techniques that AutoDock provided and considered the possibility of adding new algorithms that would improve the obtained results.
2. To achieve this goal, instead of trying to incorporate the new algorithms into the source code of AutoDock, the approach has been to use a software framework oriented to solving optimization problems with metaheuristics. Specifically, the framework used is jMetal, which is a Java-based object oriented software library that incorporates a number of single-objective algorithms. As jMetal was implemented in Java and AutoDock in C++, we have developed a jMetal version in C++. So, we have integrated both tools (AutoDock 4.2 and jMetal) to optimize the resulting binding energy [8].
3. After having a broad collection of implemented metaheuristics thanks to jMetalCpp, a detailed study has been made where we applied a set of metaheuristics for optimizing a single objective by minimizing the final free binding energy that is the result of the sum of all energy terms from the AutoDock 4.2 energy scoring function. Therefore, four metaheuristics such as two variants of the GA (Genetic Algorithm) gGA (generational Genetic Algorithm) and ssGA (steady-state Genetic Algorithm), DE (Differential Evolution) [9] and PSO (Particle Swarm Optimization) [10] were used to solve the molecular docking problem. This phase has been divided in two substeps in which we used two different sets of instances, which involve as receptors HIV-proteases with flexible aminoacids' side chains and as ligands flexible HIV-proteases inhibitors [2]. The first set of instances was used to do a study about configurations for fine-tuning algorithms and the second to compare the accuracy of the ligand-receptor conformation obtained from AutoDock and AutoDock+jMetalCpp.
4. The next phase would be to apply a multi-objective formulation to solve the molecular docking problem given the interesting results obtained by Janson *et al.* (2008), in which two objectives like the intermolecular ( $E_{inter}$ ) energy and the intramolecular ( $E_{intra}$ ) energy were minimized [7]. Therefore, we compare and analyze the performance of a set of multi-objective metaheuristics when solving flexible molecular docking complexes by minimizing the  $E_{inter}$  and the  $E_{intra}$ . These algorithms are: Nondominated Sorting Genetic Algorithm II (NSGA-II) [11] and its steady-state version (ssNSGA-II) [12], Speed Modulation Multi-Objective

Particle Swarm Optimization (SMPSO) [13], Third Evolution Step of Generalized Differential Evolution (GDE3) [14], Multi-Objective Evolutionary Algorithm Based on Decomposition (MOEA/D) [15], and S Metric Evolutionary Multiobjective Optimization (SMS-EMOA) [16]. These algorithms have been shown to obtain successful performances on a wide variety of optimization problems [17, 11], however they have not been previously used to solve the molecular docking problem with exception of the NSGA-II algorithm [18].

5. After testing already existing multi-objective approaches, new ones can be tested. In particular the use of RMSD as an objective could be useful in guiding algorithms to find solutions similar to the reference solution. The previous study can be replicated using a different set of objectives.
6. Explore more deep-fully the algorithm which obtained the best results in previous studies. In particular, a study of variants of SMPSO by minimizing the  $E_{inter}$  and the RMSD score should be carried on. SMPSO performs a limitation mechanism of particle's velocity to avoid the movement of particles in search regions out of the problem ranges. This algorithm uses an external archive to store non-dominated solutions according to the crowding distance [19]. This archive is also used in the leader selection mechanism. The performance of these variants can be assessed by applying two main quality indicators intended to measure convergence and diversity of the computed Pareto front approximations. The study can provide some clues about how new algorithms based on SMPSO can be adapted to improve the molecular docking results for simulations that involve flexible ligand and receptors.

### 1.3 Thesis contributions

To summarize, the main contributions of this thesis are as follows:

- The implementation of a C++ metaheuristic framework (jMetalCpp), port of the widely used Java framework jMetal, to solve optimization problems and its later public distribution between the scientific community [8].
- The inclusion of the metaheuristic techniques from jMetalCpp into the molecular docking tool AutoDock, and its public distribution for increasing the possibilities of biological user when tackling the molecular docking problem [8].
- The demonstration that different single-objective optimization techniques apart from those widely used between the molecular docking communities could lead to a higher quality results. In our case of study, DE obtained better results than those obtained by AutoDock [20].
- The proposal of different multi-objective approaches to solve the molecular docking problem, such as the binding energy decomposition or the use of RMSD as an objective [21].
- The demonstration of SMPSO, a multi-objective particle swarm optimization metaheuristic, as a remarkable technique to solve molecular docking problems when taking a multi-objective approach and even achieving better solutions than the usual single-objective techniques [22].
- The introduction of two new variants of SMPSO. The first is SMPSOD, an archive-less approach, inspired by MOEA/D. The second is SMPSOC, which uses the new cosine similarity to calculate the density estimator [22].

## 1.4 Thesis organization

This thesis has been organized as follows. The current chapter contains an introduction to the work done, presenting the motivation to carry it out, the objectives that have been sought, the phases that have been followed to achieve those objectives and the main contributions of the thesis. Chapter 2 focuses on describing the principles about the optimization algorithms that have been used to tackle the molecular docking problem: definition of metaheuristic, their classification and a description of the multi-objective metaheuristics. Chapter 3 includes a full description about the molecular docking problem and its significance in the studios *in silico* to drug discovery, the formulation of the problem and the application of the mono-objective and multi-objective approaches. Chapter 4 contains all the published work that supports this thesis with a summary of each one of them. Finally, Chapter 5 includes the conclusions of this dissertation and the future research lines that can be opened by this study.





## Chapter 2

# Metaheuristics

In this chapter, we focus on establishing the principles about the optimization algorithms that are used to tackle the problems that are going to be used to solve the molecular docking problem. We start from a classic optimization approach to define the concept of metaheuristic and to understand its classification. Then, multi-objective optimization concepts are introduced, as we are dealing with molecular docking problems where the components of the energy function can be optimized at the same time. Finally, we end the chapter with the statistic procedure that has been followed to evaluate the different metaheuristics, where the main performance measures are introduced, as well as the quality indicators that have been used in single-objective and multi-objective optimization problems.

### 2.1 Definition

Optimization in the sense of finding the best solution, or at least a good enough solution, for a problem is a vital importance field in the real world and, particularly, in molecular docking. We are constantly solving optimization problems, as searching the shortest path to go from some place to another, organizing our activity schedule, etc. Generally, these problems are small enough, so it is possible to solve them by ourselves without additional help. However, as these problems get larger and more complex, computer assistance is inevitable to solve them.

We start giving a formal definition about the concept of optimization. Assuming, without loss of generality, the minimization case, we can define an *optimization problem* as follows:

**Definition 1** (Optimization problem). *An optimization problem is formalized as a pair  $(S, f)$ , where  $S \neq \emptyset$  represents the solution space (or search space) of the problem, while  $f$  is a function named objective function or fitness function, that is defined as:*

$$f : S \rightarrow \mathbb{R} . \quad (2.1)$$

*Therefore, solving an optimization problem consists in finding a solution,  $i^* \in S$ , that satisfies the following inequality:*

$$f(i^*) \leq f(i), \quad \forall i \in S . \quad (2.2)$$

Assuming the case of maximization or minimization does not restrict the generality of the results, as it is possible to establish an equality between maximization and minimization problems as follows [23, 24]:

$$\max\{f(i)|i \in S\} \equiv \min\{-f(i)|i \in S\} . \quad (2.3)$$



Depending on the domain to which  $S$  belongs, we can define *binary* ( $S \subseteq \mathbb{B}^*$ ), *integer* ( $S \subseteq \mathbb{N}^*$ ), *continuous* ( $S \subseteq \mathbb{R}^*$ ), or *heterogeneous* ( $S \subseteq (\mathbb{B} \cup \mathbb{N} \cup \mathbb{R})^*$ ) optimization problems.

Due to the great importance of the optimization problems, throughout the history of computing, several methods have been developed to try solving them. A very simple classification of these methods is shown in the Figure 2.1. Initially, the techniques can be classified as exact (or enumerative, exhaustive, etc.) and approximate. Exact techniques guarantee to find the optimal solution from any problem instance in a bounded time. The disadvantage of these methods is that the time and / or memory needed, although bounded, grow exponentially with the size of the problem, as most of them are NP-hard. This means in many cases that the use of these techniques is not feasible, since much time (possibly thousands of years) and / or an exorbitant amount of memory for the problem is required. For these reasons, approximate algorithms to solve these problems are receiving increasing attention from the international community for some decades. These methods sacrifice the guarantee of finding the optimum in exchange for finding a satisfactory solution in a reasonable time.

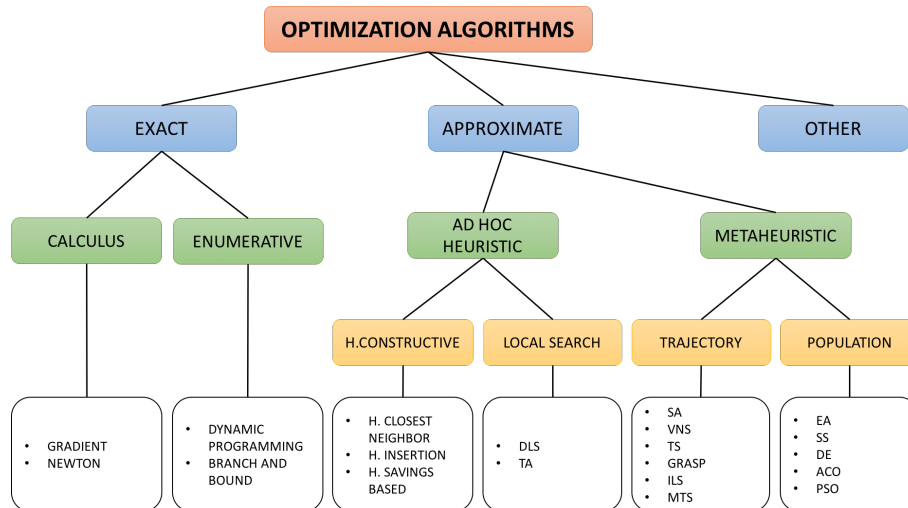


Figure 2.1: Optimization techniques classification.

Among approximate algorithms, two types can be found: *ad hoc* heuristics and metaheuristics (on which we focus on this chapter). *Ad hoc* heuristics, at the same time, are divided into *constructive heuristics* and *local search methods*.

Constructive heuristics are often the fastest methods. They build a solution from scratch by incorporating components to get a complete solution, which is the result of the algorithm. Although in many cases finding a constructive heuristic is relatively simple, the solutions offered are usually of very low quality. Finding methods of this kind that produce good solutions is very difficult, since they depend a lot on the problem, and one must have a very extensive knowledge of it for their approach. For example, when dealing with problems with many constraints, most partial solutions can only lead to non-feasible solutions.

Local search or gradient tracking methods start from a complete solution, and using the *neighborhood* concept, explore part of the search space until finding a *local optimum*. The neighborhood of a solution  $s$ , that we define as  $N(s)$ , is the set of solutions that can be built from  $s$  applying

a specific modification operator (generally named *movement*). A local optimum is a solution better or equal to any other solution from its neighborhood. These methods, starting from a initial solution, examine its neighborhood and they keep the best neighbor, continuing the process until finding a local optimum. In many cases, the complete exploration of the neighborhood is unfeasible and different strategies are approached, which lead to different variations of the generic scheme. According to the chosen movement operator, the neighborhood changes and the way of exploring changes as well, so the search process can be simplified or complicated.

In the 1980s a new class of approximate algorithms emerged, whose basic idea was to combine different heuristic methods at a higher level to achieve efficient and effective exploration of the search space. These techniques have been named *metaheuristics*. This term was introduced for the first time by Glover [25]. Before the term was completely accepted by the scientific community, these techniques were called *modern heuristics* [26]. This algorithm class includes techniques such as ant colonies, evolutionary algorithms, iterative local search, simulated annealing and tabu search. Metaheuristics reviews can be found in [27, 28]. From the different descriptions that can be found in the literature, several fundamental properties that characterize these types of methods can be highlighted:

- Metaheuristics are general strategies or templates that guide the search process.
- The goal is an efficient exploration of the search space to find (almost) optimal solutions.
- Metaheuristics are non-accurate algorithms and are generally non-deterministic.
- They can incorporate mechanisms to avoid unpromising regions of the search space.
- The basic scheme of any metaheuristic has a predefined structure.
- Metaheuristics can make use of knowledge from the problem to be solved by using specific heuristics that are controlled by the highest level strategy.

Summarizing these points, it can be agreed that a metaheuristic is a high level strategy that uses different methods to explore the search space. In other words, a metaheuristic is a general non-deterministic template that must be filled with problem-specific data (solution representation, operators to manipulate them, etc.) and allows problems with large spaces search to be tackled. In these type of techniques is really important the correct balance (generally dynamic) that exists between *diversification* and *intensification*. The term diversification refers to the evaluation of solutions that are placed in distant regions of the search space (according to a previously defined distance between solutions). This term is also known as search space *exploration*. The term intensification, however, refers to the evaluation of solutions in bounded and small regions from the search space centered on the neighborhood of concrete solutions (search space *exploitation*). The balance between these two contrary concepts is of great importance since, on the one hand promising regions from the global search space have to be quickly identified and, on the other hand, time shouldn't be wasted in already explored regions or in those that do not contain high quality solutions.

Within metaheuristics we can distinguish two types of search strategies. First, we have the "intelligent" extensions of the local search methods (metaheuristics based on trajectory in Figure 2.1). The goal of these strategies is to avoid in some way the local minimums and to move to other promising regions from the search space. This type of strategy is the one that is used by the tabu search, the iterated local search, the variable neighborhood search and the simulated annealing. These metaheuristics work with one or several neighborhood structures imposed by the local search. Another type of strategy is the one followed by the ant colonies or the evolutionary algorithms. These ones have a learning component, in the sense that, in an implicit or

explicit way, they try to learn the correlation between the problem variables in order to identify the search space regions with high quality solutions (population-based metaheuristics in the Figure 2.1). These methods perform, in this sense, biased sampling of the search space.

Formally, a metaheuristic is defined as a tuple of elements that, depending on how they are defined, leads to a particular technique or another. This formal definition has been developed in [29] and subsequently extended in [30].

**Definition 2** (Metaheuristic). *A metaheuristic  $\mathcal{M}$  is a tuple composed by the following eight components:*

$$\mathcal{M} = \langle \mathcal{T}, \Xi, \mu, \lambda, \Phi, \sigma, \mathcal{U}, \tau \rangle , \quad (2.4)$$

where:

- $\mathcal{T}$  is the set of elements that are handled by the metaheuristic. This set contains the search space and in most cases it coincides with it.
- $\Xi = \{(\xi_1, D_1), (\xi_2, D_2), \dots, (\xi_v, D_v)\}$  is a set of  $v$  pairs. Each pair is composed by a state variable of the metaheuristic and by the domain of this variable.
- $\mu$  is the number of solutions with which  $\mathcal{M}$  operates in one step.
- $\lambda$  is the number of new solutions that are generated in each iteration of  $\mathcal{M}$ .
- $\Phi : \mathcal{T}^\mu \times \prod_{i=1}^v D_i \times \mathcal{T}^\lambda \rightarrow [0, 1]$  represents the operator that creates new solutions from the existent ones. This function must fulfill for all  $x \in \mathcal{T}^\mu$  and for all  $t \in \prod_{i=1}^v D_i$ ,

$$\sum_{y \in \mathcal{T}^\lambda} \Phi(x, t, y) = 1 . \quad (2.5)$$

- $\sigma : \mathcal{T}^\mu \times \mathcal{T}^\lambda \times \prod_{i=1}^v D_i \times \mathcal{T}^\mu \rightarrow [0, 1]$  is a function that allows to select those solutions that will be handled in the following iteration of  $\mathcal{M}$ . This function must fulfill for all  $x \in \mathcal{T}^\mu$ ,  $z \in \mathcal{T}^\lambda$   $y, t \in \prod_{i=1}^v D_i$ ,

$$\sum_{y \in \mathcal{T}^\mu} \sigma(x, z, t, y) = 1 , \quad (2.6)$$

$$\forall y \in \mathcal{T}^\mu, \sigma(x, z, t, y) = 0 \vee \quad (2.7)$$

$$\vee \sigma(x, z, t, y) > 0 \wedge$$

$$(\forall i \in \{1, \dots, \mu\} \bullet (\exists j \in \{1, \dots, \mu\}, y_i = x_j) \vee (\exists j \in \{1, \dots, \lambda\}, y_i = z_j)) .$$

- $\mathcal{U} : \mathcal{T}^\mu \times \mathcal{T}^\lambda \times \prod_{i=1}^v D_i \times \prod_{i=1}^v D_i \rightarrow [0, 1]$  represents the update procedure of the state variable of the metaheuristic. This function must fulfill for all  $x \in \mathcal{T}^\mu$ ,  $z \in \mathcal{T}^\lambda$   $y, t \in \prod_{i=1}^v D_i$ ,

$$\sum_{u \in \prod_{i=1}^v D_i} \mathcal{U}(x, z, t, u) = 1 . \quad (2.8)$$

- $\tau : \mathcal{T}^\mu \times \prod_{i=1}^v D_i \rightarrow \{\text{false}, \text{true}\}$  is a function that decides the termination of the algorithm.

The above definition reflects the typical stochastic behavior of metaheuristic techniques. In particular, the  $\Phi$ ,  $\sigma$ , and  $\mathcal{U}$  functions must be interpreted as conditional probabilities. For example, the value of  $\Phi(x, t, y)$  is interpreted as the probability that the child vector  $y \in \mathcal{T}^\lambda$  is generated since at the moment the set of individuals with which the metaheuristic works is  $x \in \mathcal{T}^\mu$  and its internal state is defined by the state variables  $t \in \prod_{i=1}^v D_i$ . It can be seen that the constraints imposed on the functions  $\Phi$ ,  $\sigma$  y  $\mathcal{U}$  allow to consider them as functions that return these conditional probabilities.

**Definition 3** (Metaheuristic state). *Let  $\mathcal{M} = \langle \mathcal{T}, \Xi, \mu, \lambda, \Phi, \sigma, \mathcal{U}, \tau \rangle$  be a metaheuristic and  $\Theta = \{\theta_1, \theta_2, \dots, \theta_\mu\}$  the set of variables that will store the solution set with which the metaheuristic works. We will use the notation  $\text{first}(\Xi)$  to refer to the state variable set of the metaheuristic,  $\{\xi_1, \xi_2, \dots, \xi_v\}$ . A state  $s$  of the metaheuristic is a pair of functions  $s = (s_1, s_2)$  with*

$$s_1 : \Theta \rightarrow \mathcal{T}, \quad (2.9)$$

$$s_2 : \text{first}(\Xi) \rightarrow \bigcup_{i=1}^v D_i, \quad (2.10)$$

where  $s_2$  satisfies

$$s_2(\xi_i) \in D_i \quad \forall \xi_i \in \text{first}(\Xi). \quad (2.11)$$

We will denote with  $\mathcal{S}_{\mathcal{M}}$  the set of all states of a metaheuristic  $\mathcal{M}$ .

Finally, once defined the state of a metaheuristic, we can define its dynamic.

**Definition 4** (Metaheuristic dynamic). *Let  $\mathcal{M} = \langle \mathcal{T}, \Xi, \mu, \lambda, \Phi, \sigma, \mathcal{U}, \tau \rangle$  be a metaheuristic and  $\Theta = \{\theta_1, \theta_2, \dots, \theta_\mu\}$  the set of variables that will store the solution set with which the metaheuristic works. We will use the notation  $\bar{\Theta}$  for the tuple  $(\theta_1, \theta_2, \dots, \theta_\mu)$  and  $\bar{\Xi}$  for the tuple  $(\xi_1, \xi_2, \dots, \xi_v)$ . We will extend the state definition so that it can be applied to element tuples. Then, we define  $\bar{s} = (\bar{s}_1, \bar{s}_2)$  where*

$$\bar{s}_1 : \Theta^n \rightarrow \mathcal{T}^n, \quad (2.12)$$

$$\bar{s}_2 : \text{first}(\Xi)^n \rightarrow \left( \bigcup_{i=1}^v D_i \right)^n, \quad (2.13)$$

and besides that

$$\bar{s}_1(\theta_{i_1}, \theta_{i_2}, \dots, \theta_{i_n}) = (s_1(\theta_{i_1}), s_1(\theta_{i_2}), \dots, s_1(\theta_{i_n})) , \quad (2.14)$$

$$\bar{s}_2(\xi_{j_1}, \xi_{j_2}, \dots, \xi_{j_n}) = (s_2(\xi_{j_1}), s_2(\xi_{j_2}), \dots, s_2(\xi_{j_n})) , \quad (2.15)$$

for  $n \geq 2$ . We will say that  $r$  is a successor state of  $s$  if  $t \in \mathcal{T}^\lambda$  exists such that  $\Phi(\bar{s}_1(\bar{\Theta}), \bar{s}_2(\bar{\Xi}), t) > 0$  and besides that

$$\sigma(\bar{s}_1(\bar{\Theta}), t, \bar{s}_2(\bar{\Xi}), \bar{r}_1(\bar{\Theta})) > 0 \quad y \quad (2.16)$$

$$\mathcal{U}(\bar{s}_1(\bar{\Theta}), t, \bar{s}_2(\bar{\Xi}), \bar{r}_2(\bar{\Xi})) > 0 . \quad (2.17)$$

We will denote with  $\mathcal{F}_{\mathcal{M}}$  the binary relation “being a successor of” defined in the states set of a metaheuristic  $\mathcal{M}$ . That is,  $\mathcal{F}_{\mathcal{M}} \subseteq \mathcal{S}_{\mathcal{M}} \times \mathcal{S}_{\mathcal{M}}$ , and  $\mathcal{F}_{\mathcal{M}}(s, r)$  if  $r$  is a successor state of  $s$ .

**Definition 5** (Metaheuristic execution). *A metaheuristic  $\mathcal{M}$  execution is a finite or infinite sequence of states,  $s_0, s_1, \dots$  in which  $\mathcal{F}_{\mathcal{M}}(s_i, s_{i+1})$  for all  $i \geq 0$  and besides that:*

- if the sequence is infinite  $\tau(s_i(\bar{\Theta}), s_i(\bar{\Xi})) = \text{false}$  is satisfied for all  $i \geq 0$  and
- if the sequence is finite  $\tau(s_k(\bar{\Theta}), s_k(\bar{\Xi})) = \text{true}$  is satisfied for the last state  $s_k$  and, besides,  $\tau(s_i(\bar{\Theta}), s_i(\bar{\Xi})) = \text{false}$  for all  $i \geq 0$  such that  $i < k$ .

In the next sections we will have the opportunity to see how this general formulation can be adapted to the specific techniques (obviating those parameters not fixed by metaheuristics or that depend on other aspects such as the problem or the concrete implementation).

## 2.2 Classification

There are different ways of classifying and describing the metaheuristic techniques [27]. Depending on the selected characteristics, it is possible to obtain different taxonomies: based on nature or non based on nature, with or without memory, with one or several neighborhood structures, etc. One of the most popular classifications makes the following division: *trajectory-based* and *population-based* metaheuristics. The former manipulates a single element of the search space at each step, while the latter work on a set of them (population). This taxonomy is shown graphically in the Figure 2.2, which also includes the most representative techniques. These metaheuristics are described in the two following sections.

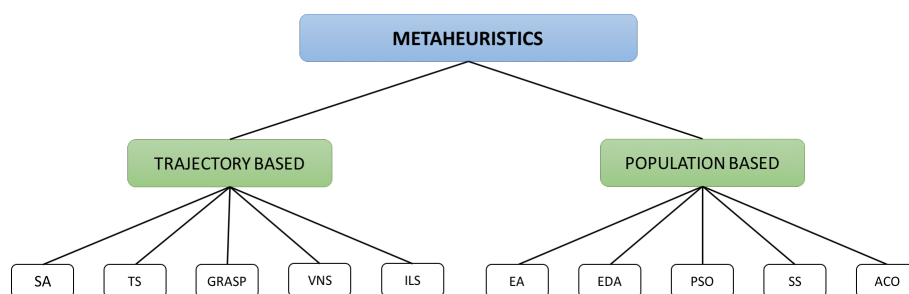


Figure 2.2: Metaheuristics classification.

### 2.2.1 Trajectory-based metaheuristics

In this section, we will briefly review some metaheuristics based on trajectory. The main characteristic of these methods is that they start from a solution and, through the neighborhood exploration, they update the current solution, forming a trajectory. According to the notation of Definition 2, this is formalized with  $\mu = 1$ . Most of these algorithms arise as extensions of simple local search methods to which some mechanism is added to escape local minimums. This implies the need for a more elaborated stopping condition than finding a local minimum. Usually the search is terminated when a predefined maximum number of iterations is reached, a solution with an acceptable quality is found, or a stagnation of the process is detected.

#### 2.2.1.1 Simulated Annealing (SA)

*Simulated annealing* (SA) is one of the oldest techniques between metaheuristics and is possibly the first algorithm with an explicit strategy to escape from the local minimum. The algorithm origins are found in an statistic mechanism named *metropolis* [31]. The SA idea is to simulate the cooling process of metal and crystal. SA was initially introduced in [32]. In order to avoid being trapped in

a local minimum, the algorithm allows to choose, with a determined probability, a solution whose value of *fitness* is worse than that of the current solution. In each iteration, a solution  $s'$  of the neighborhood  $N(s)$  is chosen from the current solution  $s$ . If  $s'$  is better than  $s$  (that is, it has a better value in the *fitness* function),  $s$  is substituted for  $s'$  as the current solution. If the solution  $s'$  is worse, then it is accepted with a certain probability that depends on the current temperature  $T$  and the difference of *fitness* between both solutions,  $F(s') - f(s)$  (case of minimization).

### 2.2.1.2 Tabu search (TS)

*Tabu search* (TS) is one of the metaheuristics that have been applied more successfully when solving combinatorial optimization problems. These method fundamentals were introduced in [25], and are based on the ideas formulated in [33]. A good summary of this technique and its components can be found in [33].

The basic idea of the tabu search is the explicit use of a search record (a short-term memory), both to escape local minima and to implement its exploration strategy, and to avoid searching several times in the same region. This short-term memory is implemented as a tabu list, where the more recently visited solutions are kept to exclude them from the next movements. In each iteration the best solution is chosen among the allowed ones and is added to the tabu list.

From the point of view of implementation, maintaining a list of complete solutions is often impractical due to its inefficiency. Therefore, generally, movements that had led the algorithm to generate that solution or the main components that define the solution are usually stored. In any case, the elements of this list allow filtering the neighborhood, generating a reduced set of eligible solutions called  $N_a(s)$ . The store of movements instead of complete solutions is much more efficient, but introduces a loss of information. To avoid this problem, an aspiration criterion is defined which allows to include a solution in  $N_a(s)$  even if it is prohibited due to the tabu list. The most widely used aspiration criterion is to allow solutions whose *fitness* is better than the best solution found so far.

### 2.2.1.3 GRASP

The *Greedy Randomized Adaptive Search Procedure* (GRASP) [34] is a simple metaheuristic that combines constructive heuristics with local search. GRASP is an iterative procedure, composed of two phases: first the construction of a solution and then an improvement process. The improved solution is the result of the search process. The solution-building mechanism is a random constructive heuristic. It adds step by step different components  $c$  to the partial solution  $s^p$ , which is initially empty. The components added in each step are randomly selected from a restricted list of candidates (*RCL*). This list is a subset of  $N(s^p)$ , the set of allowed components for the partial solution  $s^p$ . To generate this list, the components of the solution in  $N(s^p)$  are ordered according to some function dependent on the problem ( $\eta$ ).

The *RCL* list is composed by the  $\alpha$  best components of that set. In the extreme case of  $\alpha = 1$ , we always add the best found component deterministically, so that the construction method is equivalent to a voracious algorithm. At the other end, with  $\alpha = |N(s^p)|$ , the component to be added is chosen in a totally random way from all available ones. Therefore,  $\alpha$  is a key parameter that influences how the search space is to be sampled. The second phase of the algorithm consists in applying a local search method to improve the generated solution. This enhancement mechanism may be a simple enhancement technique or other more complex algorithms such as SA or TS.

### 2.2.1.4 Variable Neighborhood Search (VNS)

The *Variable Neighborhood Search* (VNS) is a metaheuristic proposed in [35] which applies explicitly one strategy to change between different neighborhoods during the search. This algorithm is very



general and with many degrees of freedom when designing particular variations and instantiations.

The first step is to define a set of neighborhoods. This choice can be made in many ways: from being randomly chosen to using complex equations deduced from the problem. Each iteration consists of three phases: the candidate's choice, a phase of improvement and, finally, the movement. In the first phase, a neighbor  $s'$  of  $s$  is chosen randomly using the  $k$ -th neighborhood. This solution  $s'$  is used as the starting point of the local search of the second phase. When the improvement process ends, the new  $s''$  solution is compared to the original  $s$ . If it is better,  $s''$  becomes the current solution and the neighborhood counter is initialized ( $k \leftarrow 1$ ). If it is not better, the process is repeated but using the following neighborhood ( $k \leftarrow k+1$ ). The local search is the intensification step of the method and the neighborhood change can be considered as the diversification step.

### 2.2.1.5 Iterated Local Search (ILS)

The *Iterated Local Search* (ILS) [36, 37] is a metaheuristic based in a simple but very effective concept. In each iteration, the current solution is disturbed, and then, a local search method is applied to this solution to improve it. The local minimum obtained by the improvement method can be accepted as the current new solution if it passes an acceptance test. The importance of the disturbance process is obvious: if it is too small, the algorithm may not be able to escape the local minimum; however, if it is too large, the disturbance can make the algorithm behaves as a local search method with a random restart. Therefore, the perturbation method must generate a new solution that serves as a start to the local search, but that should not be very far from the current one so that it is not considered to be a random solution. The acceptance criterion acts as a counterbalance, since it filters the acceptance of new solutions depending on the search history and the characteristics of the new local minimum.

## 2.2.2 Population-based metaheuristics

The population-based methods are characterized by work with a solution set, usually called population, in each iteration (that is, generally  $\mu > 1$  and/or  $\lambda > 1$ ), as opposed to methods based in trajectory, that only manipulate a solution of the search space by iteration.

### 2.2.2.1 Evolutionary Algorithms (EA)

*Evolutionary algorithms* (EAs) are inspired by the natural evolution theory. This family of techniques follows an iterative and stochastic process that operates on a solution population, named in this context *individuals*. Initially, the population is typically generated randomly (perhaps with the help of a construction heuristic).

The general scheme of an evolutionary algorithm comprises three main phases: selection, reproduction and replacement. The entire process is repeated until some termination criterion is met (usually after a given number of iterations). In the selection phase, the most suitable individuals of the present population are generally chosen to be subsequently recombined in the reproduction phase. Individuals resulting from recombination are altered by a mutation operator. Finally, from the current population and/or the best individuals generated (according to their value of *fitness*) the new population is formed, giving way to the next generation of the algorithm.

### 2.2.2.2 Estimation of Distribution Algorithms (EDA)

*Estimation of Distribution Algorithms* (EDAs) [38] show a similar behavior to the evolutionary algorithms presented in the previous section. In fact, many authors consider the EDAs as another type of EA. The EDAs operate on a population of tentative solutions such as evolutionary algorithms but, unlike the latter, which use recombination and mutation operators to improve the



solutions, EDAs infer the probability distribution of the selected set and, using it, they generate new solutions that will be part of the population.

Probabilistic graphical models are tools commonly used in the context of the EDAs to efficiently represent the probability distribution. Some authors [39, 40, 41] have proposed the Bayesian networks to represent the probability distribution in discrete domains, whereas the gaussian networks are usually used in the continuous domains [42].

### 2.2.2.3 Scatter Search (SS)

*Scatter Search* (SS) [43] is a metaheuristic whose principals were introduced in [33] and which nowadays is receiving a great deal of attention from the scientific community [44]. The algorithm is based on maintaining a relatively small set of tentative solutions (called reference set or *RefSet*) that is characterized by containing quality and diverse solutions (distant in the search space). For the complete definition of SS, five components must be defined: creation of the initial population, generation of the reference set, generation of subsets of solutions, method of combining solutions and improvement method.

### 2.2.2.4 Ant Colony Optimization (ACO)

The *Ant Colony Optimization* (ACO) [45, 46] algorithms are inspired by the behavior of real ants when looking for food. This behavior is described as follows: initially, ants explore the area near their nest randomly. As soon as an ant finds food, it takes it to the nest. While performing this path, the ant is depositing a chemical called pheromone. This substance will help the rest of the ants find the food. Indirect communication between ants through the pheromone trail enables them to find the shortest path between the nest and the food. This behavior is the one that tries to simulate this method to solve optimization problems. The technique is based on two main steps: construction of a solution based on the behavior of an ant and update of the artificial pheromone traces. The algorithm does not set any *a priori* planning or synchronization between phases, and can even be performed simultaneously.

### 2.2.2.5 Particle Swarm Optimization (PSO)

*Particle Swarm Optimization* (PSO) [47] algorithms are inspired by the social behavior of the flight of flocks of birds or the movement of fish banks. The PSO algorithm maintains a set of solutions, also called *particles*, that are randomly initialized in the search space. Each particle has a position and velocity that changes as the search progresses. The particle movement is influenced by its velocity and by the positions where the particle itself and others in its neighborhood have found good solutions. In the context of PSO, the *neighborhood of a particle* neighborhood of a particle is defined as a set of particles in the cluster. It should not be confused with the neighborhood concept of a solution previously used in this chapter. The neighborhood of a particle can be *global*, in which all cluster particles are considered neighbors, or *local*, where only the nearest particles are considered to be neighbors.

## 2.3 Multi-objective optimization metaheuristics

Most real-world optimization problems are multiobjective in nature, which means that you have to minimize or maximize several functions at the same time as they are normally in conflict with each other (multi-objective problems or MOPs, *Multi-objective Optimization Problems*). Due to the lack of adequate methodological solutions, multi-objective problems have been solved in the past as single-objective problems. However, there are fundamental differences in the operating

principles of algorithms for single- and multi-objective optimization. Thus, the techniques used to solve MOPs are not usually restricted to finding a single solution, but a set of compromise solutions between the multiple conflicting objectives, since there is usually no solution that simultaneously optimizes all objectives. Two stages can therefore be distinguished when addressing this type of problem: on the one hand, the optimization of several objective functions involved and, on the other hand, the decision-making process on which compromise solution is most appropriate [48]. Given how they handle these two stages, techniques for solving MOPs can be classified in [49]:

- *A priori*: when decisions are taken before searching solutions.
- *Progressives*: when the search for solutions and decision-making are integrated.
- *A posteriori*: when searching is done before making decisions.

Each of them has certain advantages and disadvantages that make them more suitable for certain concrete scenarios [48, 50]. However, in the first two classes, the search is heavily influenced by an expert (*decision maker*) that determines the importance of one objective over another and that can arbitrarily limit the search space, preventing an optimal resolution of the problem. In the *a posteriori* techniques, on the contrary, an exploration is made as wide as possible to generate as many compromise solutions as possible. It is, then, when the decision-making process by the expert takes place. Precisely, because of this approach, these *a posteriori* techniques are being used in the field of metaheuristics and, particularly, in the field of evolutionary computing [48, 50]. More specifically, the most advanced algorithms apply *a posteriori* techniques based on the concept of *Pareto Optimality* [51] and this is the approach followed in this thesis. Thus, we have structured this section into three sections. The first one presents formally the basic concepts related to this Pareto optimality. The following section presents the goals that should be pursued by any algorithm that uses these techniques when approaching a MOP. Finally, the third section discusses some aspects of design that must be adopted in the algorithms that solve problems following the previous approach.

### 2.3.1 Basic concepts

In this section, we present some basic concepts of multi-objective optimization to familiarize the reader with this field. We will begin by giving some notions of what we mean by multi-objective optimization problem or MOP. Informally, an MOP can be defined as the problem of finding a vector of decision variables that satisfies a set of constraints and that optimizes a set of objective functions. These functions form a mathematical description of performance criteria that are usually in conflict with each other. Therefore, the term “optimization” refers to the search for a solution such that it contains acceptable values for all objective functions [52].

Mathematically, the MOP formulation extends the classical definition of single-objective optimization (Definition 1) to consider the existence of several objective functions. Therefore, there is not a single solution to the problem, but a solution set. This set of solutions is found by using the Pareto Optimality Theory [53]. Formally [54]:

**Definition 6 (MOP).** *Finding a vector  $\vec{x}^* = [x_1^*, x_2^*, \dots, x_n^*]$  that satisfies the  $m$  inequality constraints  $g_i(\vec{x}) \geq 0, i = 1, 2, \dots, m$ , the  $p$  equality constraints  $h_i(\vec{x}) = 0, i = 1, 2, \dots, p$ , and that minimizes the vector function  $\vec{f}(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), \dots, f_k(\vec{x})]^T$ , where  $\vec{x} = [x_1, x_2, \dots, x_n]^T$  is the decision variables vector.*

The set of all values satisfying the constraints defines the *feasible solutions region*  $\Omega$  and any point in  $\vec{x} \in \Omega$  is a *feasible solution*.

Having several objective functions, the notion of “optimum” changes, since the goal for any MOP is to find good compromises (*trade-offs*) between these functions. The most used “optimum” notion is the one proposed by Francis Ysidro Edgeworth [55], later generalized by Vilfredo Pareto [51]. Although some authors call it the Edgeworth-Pareto optimum, the Pareto optimum term is commonly accepted. Its formal definition is given as follows:

**Definition 7** (Pareto optimality). *A point  $\vec{x}^* \in \Omega$  is a Pareto optimum if for each  $\vec{x} \in \Omega$  and  $I = \{1, 2, \dots, k\}$ , or  $\forall_{i \in I} (f_i(\vec{x}) = f_i(\vec{x}^*))$  or there is at least one  $i \in I \mid f_i(\vec{x}) > f_i(\vec{x}^*)$ .*

This definition says that  $\vec{x}^*$  is a Pareto optimum if there is not any feasible vector  $\vec{x}$  that improves any criterion without simultaneously causing a worsening in at least one other criterion (assuming minimization). The Pareto optimality concept is integral to both the theory and the resolution of MOPs. There are a few additional definitions that are also basic in multi-objective optimization [54]:

**Definition 8** (Pareto dominance). *A vector  $\vec{u} = (u_1, \dots, u_k)$  is said to dominate another vector  $\vec{v} = (v_1, \dots, v_k)$  (represented by  $\vec{u} \prec \vec{v}$ ) if and only if  $\vec{u}$  is partially less than  $\vec{v}$ , that is,  $\forall i \in \{1, \dots, k\}, u_i \leq v_i \wedge \exists i \in \{1, \dots, k\} : u_i < v_i$ .*

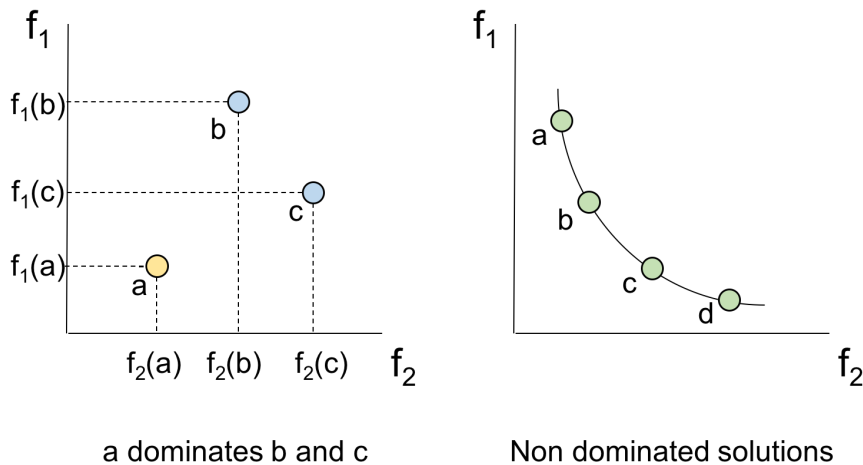


Figure 2.3: Pareto dominance example.

We are going to illustrate this concept graphically. Figure 2.3 includes two sets of solutions for a multi-objective problem with two functions  $f_1$  and  $f_2$ , which have to be minimized. Both objectives being equally important, it is not trivial to distinguish which solution is better than another. We can use the above definition for this. Thus, if we look at the left side of the figure, we can say that  $a$  is better than  $b$  since  $f_1(a) < f_1(b)$  and  $f_2(a) < f_2(b)$ . That is, it is better in both objectives and, therefore, it is said that  $a$  dominates  $b$  ( $a \prec b$ ). The same happens if we compare  $a$  and  $c$ , in both objectives  $f_1(a) < f_1(c)$  and  $f_2(a) < f_2(c)$ , so  $a \prec c$ . Let us now compare the solutions  $b$  and  $c$  between them. It can be seen that  $c$  is better than  $b$  in  $f_1$  ( $f_1(c) < f_1(b)$ ), but  $b$  is better than  $c$  for  $f_2$  ( $f_2(b) < f_2(c)$ ). According to the Definition 8, we can not say that  $b$  dominates  $c$  nor that  $c$  dominates  $b$ . That is, we cannot conclude that one solution is better than the other, in which case both solutions are said to be non-dominated. In the right-hand part of the Figure 2.3, 4 solutions of this type are shown, where none is better than the others.

Solving a MOP consists, therefore, of finding the set of solutions that dominate any other solutions from the solution space, which means that they are the best solutions for the problem and, therefore, make up its optimal solution. Formally:

**Definition 9** (Pareto optimal set). *For a given MOP  $\vec{f}(\vec{x})$ , the Pareto optimal set is defined as  $\mathcal{P}^* = \{\vec{x} \in \Omega | \neg \exists \vec{x}' \in \Omega, \vec{f}(\vec{x}') \preceq \vec{f}(\vec{x})\}$ .*

It should not be forgotten that Pareto-optimal solutions (which are in  $\mathcal{P}^*$ ), are in the variables space (genotype). Their vector components are in the objective space (phenotype) and they can not be improved simultaneously. These solutions are also often called *not lower*, *admissible* or *efficient*. The Pareto front is then defined as:

**Definition 10** (Pareto front). *For a given MOP  $\vec{f}(\vec{x})$  and its Pareto optimal set  $\mathcal{P}^*$ , the Pareto front is defined as  $\mathcal{PF}^* = \{\vec{f}(\vec{x}), \vec{x} \in \mathcal{P}^*\}$ .*

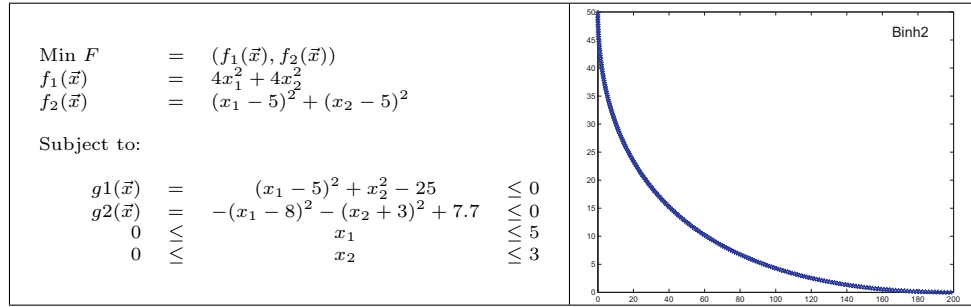


Figure 2.4: Formulation and Pareto front for the Binh2 problem.

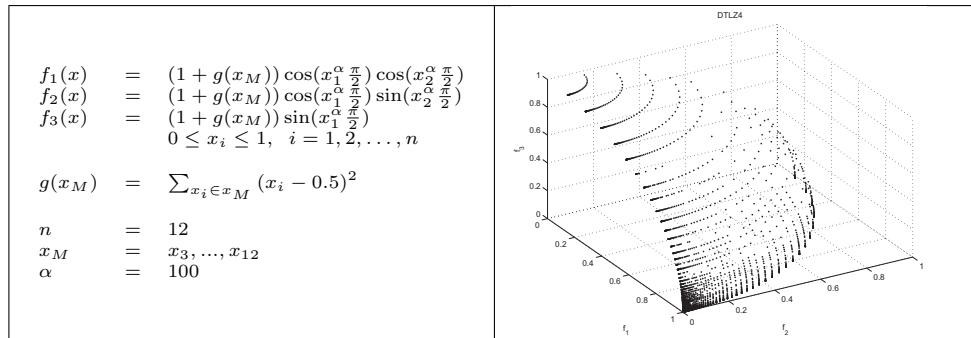


Figure 2.5: Formulation and Pareto front for the DTLZ4 problem.

That is, the Pareto front is composed of the values in the objective space of the Pareto optimal set. In general, it is not easy to find an analytical expression of the line or surface that contains these points. In fact, in most cases it is impossible. As an example, Figures 2.4 and 2.5 show the formulation and its corresponding Pareto front of problems Binh2 and DTLZ4 [48]. In the first case, it is a bi-objective problem,  $f_1$  and  $f_2$ , with two decision variables  $x_1$  and  $x_2$ , which has two constraints defined as  $g_1$  and  $g_2$ . The DTLZ4 problem, however, has three objectives and no constraint (the  $g()$  function here is only a notation used for its formulation).

### 2.3.2 Objectives in MOPs resolution

When addressing the resolution of a multi-objective optimization problem, the main goal of any optimization algorithm that uses the concepts and techniques described in the previous section is to find its Pareto front (or, what is the same, its Pareto optimal set). However, the presence of multiple Pareto-optimal solutions makes it difficult to choose one solution over another without additional information on the problem, since all these solutions are equally important. Given a MOP, therefore, we are ideally looking for a number of non-dominated solutions that pursues two goals:

1. To find a set of solutions as close as possible to the optimal Pareto front.
2. To find a set of solutions as uniformly diverse as possible.

While the first goal, converging towards the optimal solution, is mandatory in all tasks of single- or multi-objective optimization, the second one is completely specific for multi-objective optimization. Besides converging towards the optimum front, the solutions must be uniformly distributed along the whole front. Only with a diverse set of solutions can we ensure, on the one hand, a good set of compromise solutions between the different objectives for the subsequent decision-making by the expert and, on the other hand, that a good exploration of the search space has been made. Figure 2.6 shows two examples of fronts each failing in one of the previous goals. In part (a) we can see an approximation to the front in which the non-dominated solutions are perfectly distributed. However, it is a MOP designed in a way that contains multiple misleading fronts and, in fact, the solutions obtained are not Pareto-optimal, although their diversity is excellent. On the contrary, in part (b) of the same figure, we have a solution set that have converged towards the Pareto optimal front, but nevertheless some regions are left uncovered. Although neither case is desirable, the first situation is clearly worse: none of the obtained solutions is Pareto-optimal.

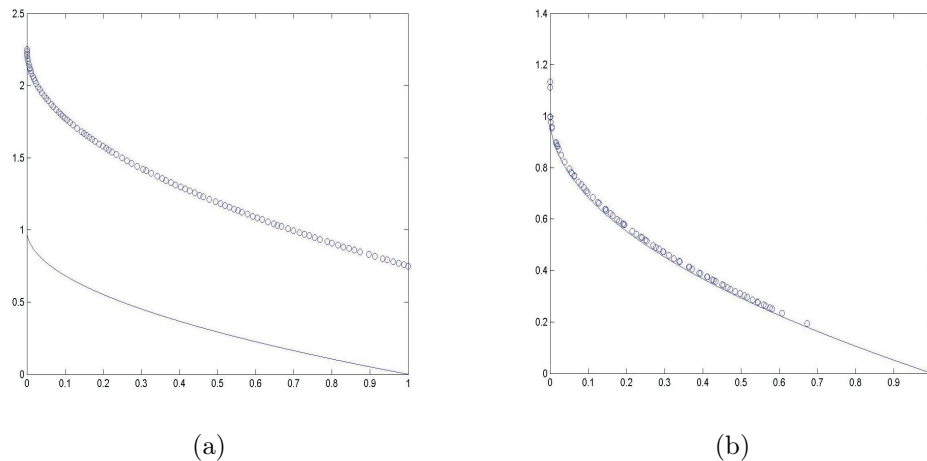


Figure 2.6: Pareto fronts examples with bad convergence (a) and bad diversity (b).

### 2.3.3 Design aspects

Adopting techniques based on Pareto optimality within metaheuristic algorithms involves, on the one hand, working with non-dominated solutions that make it necessary to incorporate specific

mechanisms to handle them, and on the other hand, to find not a single solution but a set of Pareto-optimal solutions that, besides, must have enough diversity to cover the whole front. Although there are many aspects to consider depending on each specific algorithm, the following ones can be considered common to all of them: fitness function of the solutions, diversity maintenance, and handling constraints. Each one of them is discussed below.

### 2.3.3.1 Fitness function

In the running cycle of all metaheuristics, there is always some phase in which the solution that are being handled have to be sorted according to their fitness function in order to select any of them. We refer, for example, to selection and replacement operators in evolutionary algorithms or the updating method of the reference set in scattered search. In the case of single-objective optimization, the fitness of a solution is a unique value and the ordering of solutions is trivial according to this value. However, in our approach to solving MOPs, the fitness is a vector of values (a value for each objective) so ordering is not so straightforward.

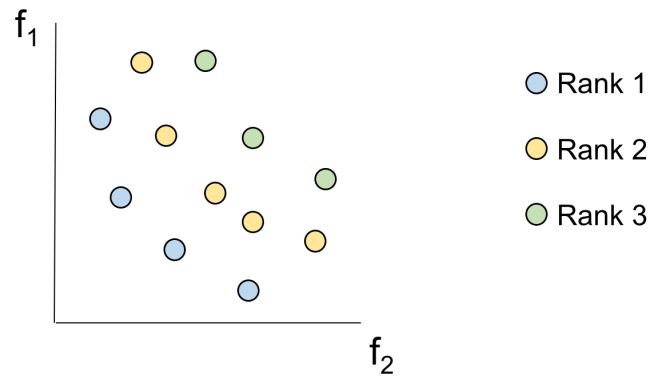


Figure 2.7: (*Ranking*) ordering example for solutions in a MOP with two objectives.

The dominance relation (Equation 8) is the key in this type of Pareto optimality-based techniques, since it will allow us to establish a solution ordering. In fact, this relation is a strict partial order relation, since it is not reflexive, neither symmetric, nor antisymmetric, but transitive. Thus, different methods have been proposed in the literature [48, 50] that basically transform the fitness vector into a unique value using this relation. This strategy was originally proposed by Goldberg in [24] to guide the population of a GA to the Pareto front of a MOP. The basic idea is to find the solutions of the population that are not dominated by any other. These solutions are assigned the highest order (the best in the ordering established by the dominance relationship). Next, the remaining non-dominated solutions are considered if all previous ones are deleted, to which the next range is assigned. The process continues until all solutions are assigned a range. Figure 2.7 shows an example of the operation of this sorting method ( $f_1$  and  $f_2$  are functions to be minimized). This dominance-based ordering is the most basic. Another more advanced, such as *strength* of SPEA2 [56] also takes into account the number of solutions dominated by each solution.

### 2.3.3.2 Diversity

Although the fitness function based on dominance already directs the search towards the Pareto front giving a greater aptitude to the non-dominated solutions, this approximation alone is not

sufficient when addressing a MOP. If we remember the Section 2.3.2, in addition to converging to the optimal front, the solutions have to be distributed as best as possible on this front in order to offer the expert the widest range of solutions to the multi-objective problem.

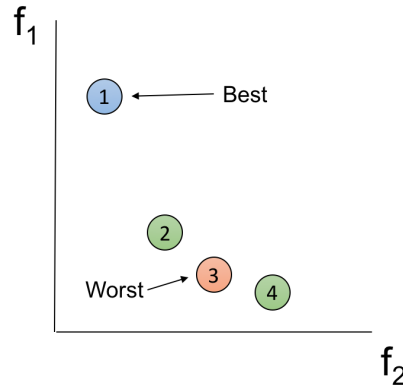


Figure 2.8: Density estimator example for non-dominated solutions in a MOP with two objectives.

Although there are different approaches in the literature [48], the most used in the state of the art algorithms are based on complementing the dominance-based fitness function (previous section) with an estimator that measures the density, in the objective space, of solutions around a given solution. Thus, given two solutions with the same fitness (*ranking, strength*), the density estimator discriminates between the best and worst solutions taking into account the diversity of them. Let us consider, for example, the set of non-dominated solution in Figure 2.8. According to their density, solution 1 can be considered as the best since it is the one that is placed in the less “populated” area. Solution 3, on the contrary, would be the worst because it is found in a front area where solutions already exist nearby. Some of the density estimators proposed by the best-known multi-objective algorithms are: *niching* in MOGA [57] and NSGA [58], the adaptive grid of PAES [59], *crowding* in NSGA-II [11] and the distance to the  $k$ -th neighbor of SPEA2 [56].

### 2.3.3.3 Constraint handling

The definition of multi-objective problem (Equation 6) included in Section 2.3.1 explicitly includes constraints since, mainly, it is the typical situation when considering real-world problems, as those considered in this thesis. Restrictions can be considered as hard or weak. A constraint is hard when it must be satisfied for a given solution to be acceptable. On the contrary, a weak constraint is one that can be relaxed in some way to accept a solution.

The most used approach in multi-objective metaheuristics of the state-of-the-art to deal with constraints are based on a schema in which feasible solutions are superior to those not feasible [60, 50]. That is, given two solutions that are to be compared, three cases can occur:

1. If both solutions are feasible, the dominance-based fitness function explained in Section 2.3.3.1 is used. In the case in which both solutions are non dominated (equal fitness), a density estimator (Section 2.3.3.2) is used to discriminate between them.
2. If one solution is feasible and the other is not, the feasible is considered as best.
3. If both solutions are not feasible, then the one that least violates the constraints is selected.



It remains to be determined how the amount of constraint violation of a given solution is quantified. In order to do this, the most commonly used strategy is to transform all constraints so that they are of type *greater-or-equal-than zero*:  $g_i(\vec{x}) \geq 0$ , according to the definition of MOP (Equation 6) [50]. It can be considered as a type of normalization, so that the value  $g_i(\vec{x})$  is used to measure how much the constraint is violated. The major drawback to this strategy is given by the equality constraints  $h_i(\vec{x}) = 0$ . If it is a weak constraint, it can be relaxed directly to  $h_i(\vec{x}) \geq 0$ . However, if  $h_i(\vec{x}) = 0$  is a hard constraint, the transformation is not direct (especially when it is a nonlinear constraint). According to a result obtained in [61], it is possible to convert these hard equality constraints into weak constraints with loss of precision, allowing all constraints of the same type to be already considered. There are many other strategies to deal with constraints in multiobjective optimization [48, 50] but we have only detailed what will be used in this thesis.

## 2.4 Statistical evaluation of results

As has been commented on several times throughout this document, metaheuristics are non-deterministic techniques. This implies that different runs of the same algorithm on a given problem do not have to find the same solution. This characteristic property of metaheuristics is an important problem for researchers when evaluating their results and, therefore, when comparing their algorithm with other algorithms.

There are some papers that address the theoretical analysis for a large number of heuristics and problems [62, 63], but given the difficulty of this type of theoretical analysis, the behavior of algorithms is traditionally analyzed by empirical comparisons. For this, it is necessary to define indicators that allow these comparisons. We can find, in general, two different types of indicators. On the one hand, we have those who measure the quality of the solutions obtained. Given that throughout the development of this thesis we have addressed problems of optimization both single- and multi-objective, it is necessary to consider different quality indicators for each type since, although the result in the first case is a single solution (the global optimum), in the second case we have a set of solutions, the optimal Pareto set (Equation 9). On the other hand, we have the indicators that measure the performance of the algorithms and that refer to the execution times or the amount of computational resources used. We have centered our discussion in the following section in the quality indicators, as the problem executions tackled in this thesis ends at reasonable times, both are closely linked and are often used together for the evaluation of metaheuristics, since the goal of this type of algorithm is to find high quality solutions at a reasonable time.

Once the indicators are defined, a minimum of independent runs of the algorithm must be performed to obtain statistically consistent results. A value of 30 is considered a minimum acceptable according to the values often chosen in the literature. The mere inclusion of means and standard deviations may be insufficient, since erroneous conclusions can be obtained. A global statistical testing may be necessary to assess whether differences are significant and not the product of random variations [64, 65]. This topic is discussed in more detail in the 2.4.2 section.

### 2.4.1 Quality indicators

These indicators are the most important when evaluating a metaheuristic. They are different depending on whether or not the optimal solution of the problem in question is known (a common problem for classical literature, but unusual in real world problems). As already mentioned above, it is necessary to distinguish between indicators for single-objective and multi-objective problems.



### 2.4.1.1 Single-objective optimization indicators

For instances of problems where the optimal solution is known, it is easy to define an indicator to control the quality of the metaheuristic: the number of times it is reached (*hit rate*). This measure is generally defined as the percentage of times the optimal solution is reached with respect to the total number of executions performed. Unfortunately, knowing the optimal solution is not a common case for realistic problems or, even if known, its computation can be so computationally heavy that it is important to find a good approximation in a shorter time. In fact, it is common for experiments with metaheuristics to be limited to performing at most a predefined computational effort (visit a maximum number of points in the search space or a maximum execution time).

In these cases, when the optimum is not known, statistical measures of the corresponding indicator are usually used. The most popular are the average and median of the best fitness value found in each independent run. In general, it is necessary to provide other statistical data such as variance or standard deviation, in addition to the corresponding statistical analysis, to give statistical confidence to the results.

In problems where the optimum is known both metrics can be used, both the number of successes and the average / median of the final (or effort) fitness. What's more, using both gets more information: for example, a low number of hits but a high precision indicates that rarely meets the optimum but is a robust method.

### 2.4.1.2 Multi-objective optimization indicators

Although the procedure for measuring the quality of solutions in single-objective problems is clear, within the multi-objective field this is a very active research topic [64, 66], since the result of these algorithms is a set of non dominated solutions and not a single solution. We must therefore define quality indicators for the Pareto front approaches. There are usually two aspects to consider when measuring the quality of a front: convergence and diversity. The first one refers to the distance between the approach and the optimum Pareto front, while the second measures the uniformity of the solution distribution on the front. As for the single-objective case, there are indicators based on whether or not the optimal front is known. The quality indicators used in this thesis are Hypervolume ( $I_{HV}$ ) [67] and Unary Additive Epsilon Indicator ( $I_{\epsilon+}$ ) [66], being the two of them Pareto-compliant [64]. Further details of these indicators are shown below (the interested reader can see [48, 50] for other quality indicators defined in the literature):

- **Hypervolume –  $I_{HV}$ .** The metric *hypervolume* [67] is a combined metric of convergence and diversity that calculates the volume, in the objective space, covered by the members of a set  $Q$  of non-dominated solutions for the discontinuous line in Figure 2.9,  $Q = \{A, B, C\}$  for problems in which all targets must be minimized. Mathematically, for each solution  $i \in Q$ , a hypercube  $v_i$  is constructed using a reference point  $W$  (which may be composed of the worst solution for each objective, for example) and the solution  $i$  as the corners of the hypercube diagonal. The reference point can be obtained simply by constructing a vector of the worst values for the functions. Thus,  $HV$  is calculated as the volume of the union of all hypercubes:

$$HV = volume \left( \bigcup_{i=1}^{|Q|} v_i \right) . \quad (2.18)$$

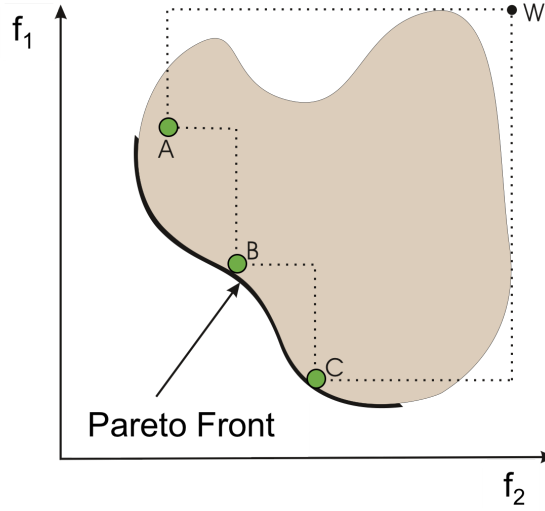


Figure 2.9: Hypervolume covered by the non-dominated solutions.

- **Epsilon** –  $I_{\epsilon+}$ . Given an computed front for a problem,  $A$ , this indicator is a measure of the smallest distance one would need to translate every solution in  $A$  so that it dominates the optimal Pareto front of this problem [66]. More formally, given  $z^{\vec{1}} = (z_1^1, \dots, z_n^1)$  and  $z^{\vec{2}} = (z_1^2, \dots, z_n^2)$ , where  $n$  is the number of objectives:

$$I_{\epsilon+}^1(A) = \inf_{\epsilon \in \mathbb{R}} \left\{ \forall z^{\vec{2}} \in \mathcal{PF}^* \exists z^{\vec{1}} \in A : z^{\vec{1}} \prec_{\epsilon} z^{\vec{2}} \right\} \quad (2.19)$$

where,  $z^{\vec{1}} \prec_{\epsilon} z^{\vec{2}}$  if and only if  $\forall 1 \leq i \leq n : z_i^1 < \epsilon + z_i^2$ . In this case, solution fronts with lower values of  $I_{\epsilon+}$  are desirable.

### 2.4.2 Statistical performance assessment

As previously explained, metaheuristics are stochastic based algorithms with different random components in their operations. Opposite to deterministic procedures, for which, just a single execution is required, when working with metaheuristics, performing a series of independent runs for each algorithm's configuration is a mandatory task in order to obtain a distribution of results. In this case, it is possible to compute a global indicator (median, mean, standard deviation, etc.) from the resulted distribution to measure the performance of the studied algorithm. Nevertheless, using one single global indicator to directly compare metaheuristics should lead empirical analyses to biased conclusions. Therefore, the correct practice is to compare the distributions of results by means of statistical tests, which are indispensable tools to validate and to provide confidence to our empirical analysis.

The standard procedure, recommended by the scientific community [68, 65], for the statistical comparison of metaheuristics lies in the use of *parametric* or *non-parametric* tests. Parametric tests show a high precision to detect differences in comparisons, although they are restricted to distributions fulfilling three specific conditions: *independency*, distributions are obtained from independent executions; *normality*, they follow a Gaussian distribution; and *heteroskedasticity*, indicating the existence of a violation of the hypothesis of equality of variances. Non-parametric tests also show a successful performance, although they are less restrictive, since they can be

applied regardless of the three previous conditions. Among all these tests, we can find procedures to perform rankings, pair-wise comparisons, and multiple post hoc comparisons.

In this thesis, we have adopted the non-parametric procedure to validate our results and to compare our proposals with other techniques in the current state of the art. Our null hypothesis (equality of distributions) has been set with a confidence level of 95%, meaning that statistical differences can be found in distributions when resulted tests are with a  $p - value < 0.05$ . First, *Friedman's* test is first performed in order to check whether statistical differences exist or not. If so, a *Wilcoxon's* (signed rank variant) or *Holm's* tests are performed depending on the number of distributions to compare: 2 or more than 2, respectively. The KEEL (Knowledge Extraction based on Evolutionary Algorithm) [69] implementation of these tests were used in all the studies that are presented in this thesis.



## Chapter 3

# The Molecular Docking Problem

This section has been divided in three subsections. In the Section 3.1, we have defined the molecular docking problem and described the importance of the application of molecular docking approaches in the context of drug discovery. Section 3.2 included the molecular docking problem formulation for the mono-objective and multi-objective optimizations. Finally, in Section 3.3, we have concluded with a complete review of those studies in which optimization techniques are applied to solve the molecular docking problem.

### 3.1 Molecular docking: Definition and biological significance

The research based-pharmaceutical industry has increasingly included computational approaches to know the intricate aspect of intermolecular recognition. These approaches have evolved hand-by-hand with biomolecular spectroscopic methods such as the X-ray crystallography and NMR that have an important impact in molecular and structural biology discovery [70]. These experimental techniques have allowed to discover the resolution of more than 100,000 tridimensional structures as the PDB specifies. However, to analyze how the molecules with a known resolution interact, it is necessary to integrate studios *in silico* and experimental techniques.

In the context of structure-based drug design (SBDD) methods, there are three computational techniques which are widely used such as molecular docking, structure-based virtual screening (DBVS) and also molecular dynamics (MD) in order to determine binding energies between a ligand and a given therapeutic target, molecular interactions between atoms and also the changes of molecules' conformations during an interaction. A different approach to the SBDD is the ligand-based structure design (LSBD) which consists of the use of libraries of active ligands and computational approaches (as molecular docking) to detect possible therapeutic targets.

The molecular docking is one of the approaches used in SBDD which tries to predict the conformation of small molecules to a binding site of a given macromolecule that can be a therapeutic target. In the process of SBDD, studies *in silico* like molecular docking are performed to identify candidate ligands to a target. The PDB database currently contains 130,599 biological macromolecules structures which most of them are involved in metabolic and biosignaling and therefore, they can be possible therapeutic targets. Once that the ligand-receptor (the ligand-receptor complex) has been identified in terms of energetic affinity and molecular interactions, the ligand can be modified to increase its efficiency to bind to the therapeutic target. The results are analyzed using molecular docking to know how these molecular modifications alter the binding efficiency of the ligand.

The application of the molecular docking to the SBDD is possible given the accuracy of



the ligand-protein predictions performed by molecular docking softwares. The main objective of the molecular docking is to determine the minimal binding energy of the predicted ligand-macromolecule complex. The more negative the obtained binding energy score is, the more stable the ligand-receptor interaction is and thus, the ligand is likely more efficient inhibiting the therapeutic target.

In the computational development of molecular docking software, researchers in this field have traditionally focused on two of the components which determine the quality of the results obtained from the molecular docking software: the energy scoring function and the optimization algorithm. The energy scoring function evaluates the conformation with a given binding energy score. In the literature, there are molecular docking software tools that use different energy scoring functions such as AutoDock [2], AutoDock Vina [3], GOLD [1] etc. In fact, there have been some studies based on replacing and comparing energy functions in terms of accuracy of ligand affinity predictions and speeding as is reported in [71]. However, in this dissertation, we have focused on the optimization algorithms by doing an extensive study based on the application of mono- and multi-objective algorithms to solve the molecular docking problem. In the following section, we have introduced the formulation of the problem, how the solutions have been encoded for the mono- and multi-objective optimization approaches and a full description of the objectives that were optimized.

## 3.2 Problem formulation

The main objective of the molecular docking problem is to find an energetically stable complex between a ligand, which can be a small compound (e.g. metabolite, inhibitors etc.), a peptide or a peptidomimetic inhibitor and a macromolecule. There are some computational tools to predict ligand-receptor complexes. In this thesis, we have selected AutoDock 4.2 which is one of the most popular and cited molecular docking software in the research community [6, 72]. AutoDock 4.2 is a C++ software package that provides an energy scoring function and several algorithms such as a Simulated Annealing (SA) and two Genetic Algorithms (GAs), one of which, referred to as the Lamarckian Genetic Algorithm (LGA), which incorporates a local search [73]. AutoDock 4.2 energy scoring function is a semi-empirical force field which allows to apply flexibility in ligand and side-chains of protein's aminoacids. The method to apply flexibility in the macromolecule is the same as used in the conformational space of the flexible ligand. The application of flexibility to the ligand and receptor makes the docking simulations more realistic and gives more complexity to the problem increasing the freedom degrees. The limit of freedom degrees that can be applied to AutoDock 4.2 function is 32 [2].

For the mono-objective and multi-objective optimization, the solution of the ligand-receptor complex is encoded in the same way. As illustrated in Fig. 3.1, each problem solution for AutoDock 4.2 and jMetal is encoded by a real-value vector of  $n + 7$  variables, in which the first three values correspond to the ligand translation involving the three axis values ( $x, y, z$ ) in Cartesian coordinate space, the next four values correspond to the ligand and/or macromolecule orientation, and the remaining  $n$  values are the ligand and macromolecule torsion dihedral angles. For the mono- and multi-objective approaches, we have used a grid-based methodology provided by AutoDock in which the macromolecule's interaction site is embedded in a 3D rectangular grid. For each point of the grid, the electrostatic interaction energy and the van der Waals terms for each ligand atom type are pre-computed and stored, taking into account all the protein atoms. In this way, the protein contribution at any given point is obtained by tri-linear interpolation in each grid cell. This interpolation leads to a range of translation variables ( $x, y, z$ ) of 120 grid spacing points dimension [74]. The values of these variables are delimited between the range of the coordinates of the grid space that has been chosen for each problem. All ranges are selected randomly, so if

the center of the grid is for example the (10, 10, 10) point, a solution with values of ten in its  $x$ ,  $y$  and  $z$  will be in such a position. In the case of orientation (quaternion) and torsion variables, they are measured in radians and encoded in the range of  $[-\pi, \pi]$ .

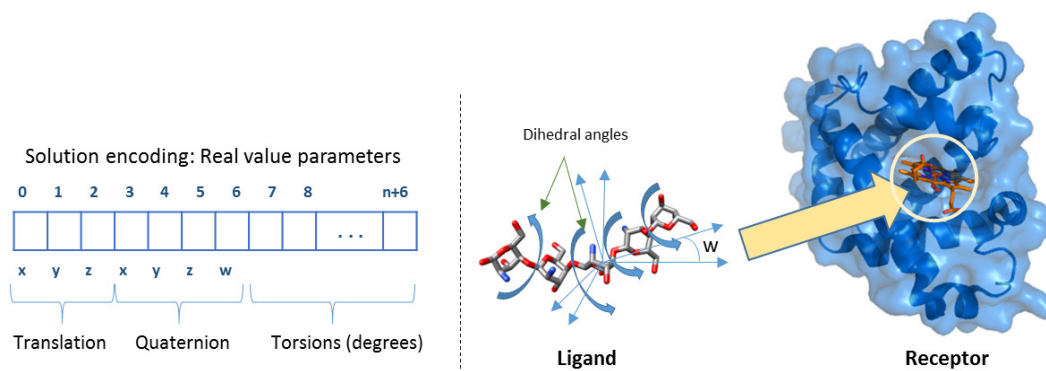


Figure 3.1: Solution encoding in AutoDock 4.2 and jMetal. The first three values (translation) are the coordinates of the center of rotation of the ligand. The next four values (quaternion) are the unit vector describing the direction of rigid body rotation ( $x$ ,  $y$  and  $z$ ) and the rotation of the angle degrees ( $w$ ) that are applied. The rest of the values hold the torsion angles in degrees, being  $n$  the number of torsions of the ligand.

### 3.2.1 Mono-objective optimization

In the mono-objective optimization approach, the objective to optimize is the final free binding energy ( $\Delta G$ ), which is measured in kcal/mol. The more negative  $\Delta G$  is, the more stable the computed ligand-receptor complex.  $\Delta G$  is computed by the energy scoring function provided by AutoDock 4.2, which is used to measure the quality of the ligand-receptor binding solutions [2].  $\Delta G$  is calculated according to the following equations (each term of the equations is described below):

$$\Delta G = (Q_{bound}^{R-L} - Q_{unbound}^{R-L} + \Delta S_{conf}) + (Q_{bound}^{L-L} - Q_{unbound}^{L-L}) + (Q_{bound}^{R-R} - Q_{unbound}^{R-R}) \quad (3.1)$$

$$Q = W_{vdw} \sum_{i,j} \left( \frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^6} \right) + W_{hbond} \sum_{i,j} E(t) \left( \frac{C_{ij}}{r_{ij}^{12}} - \frac{D_{ij}}{r_{ij}^{10}} \right) + W_{elec} \sum_{i,j} \frac{q_i q_j}{\epsilon(r_{ij}) r_{ij}} + W_{sol} \sum_{i,j} (S_i V_j + S_j V_i) e^{(-r_{ij}^2 / 2\sigma^2)} \quad (3.2)$$

$$\Delta S_{conf} = W_{conf} \cdot N_{tors} \quad (3.3)$$

As formulated in Eq. 3.1, the free binding energy function is calculated from the differences between ligand/s (L) and receptor/s (R) in bounded and unbounded states. That is, the free energy of binding is based on the evaluation of the transition of the ligand and protein intramolecular energetics from an unbounded to a bounded state, and the intermolecular energetics of ligand–protein complex. Therefore, the force field involves six pair-wise evaluations (V) plus a term of conformational entropy (Ntors), which is directly proportional to the number of rotatable bonds of the ligand molecule ( see Eq. 3.3). Each pair of energetic evaluation terms includes the evaluations of dispersion/repulsion (vdw), hydrogen bonds (hbond), electrostatics (elec), and desolvation (sol). Weights  $W_{vdw}$ ,  $W_{hbond}$ ,  $W_{conf}$ ,  $W_{elec}$ , and  $W_{sol}$  of Eqs. 3.2 and 3.3, are constants for van der Waals, hydrogen bonds, torsional forces, electrostatic interactions, and desolvation, respectively. In Eq. 3.2,  $r_{ij}$  represents the interatomic distance,  $A_{ij}$  and  $B_{ij}$  in the first term are Lennard–Jones parameters taken from Amber force field [75]. Similarly,  $C_{ij}$  and  $D_{ij}$  in the second term are Lennard–Jones parameters for maximum well depth of potential energies between two atoms, and  $E(t)$  represents the angle-dependent directionality. The third term in Eq. 3.2 uses a Coulomb approach for electrostatics. Finally, the fourth term is calculated from the volume ( $V$ ) of the atoms that are surrounding a given atom weighted by  $S$ , and an exponential term which involves atom distances.

Ligand–protein docking is a highly complex optimization problem, with unknown optimum and usually characterized by multimodal landscape energy functions [76]. In addition, the computational cost of each energy evaluation increases with the number of atoms in complex ligand–protein (with thousands of them), hence involving millions of energy evaluations, since a minimum quality of molecular binding is mandatory in molecular docking modeling. Therefore, the use of meta-heuristic approaches is highly recommendable for molecular docking, since they are able to explore a great number of combinations with a fast convergence to successful solutions [77].

### 3.2.2 Multi-objective optimization

A multi-objective optimization problem is characterized by two spaces: the decision space and the objective space. The former refers to all the possible feasible solutions, and the latter includes their corresponding objective values.

**Decision Space:** As mentioned in Section 3.2, the AutoDock 4.2 solution for the multi-objective approach is encoded in the same way as the mono-objective approach. This means that all the returned solutions are encoded in a real-value vector of  $7+n$  variables in which the first three values correspond to the ligand translation, the next four values correspond to the ligand and/or macromolecule orientation, and the remaining  $n$  values are the ligand and macromolecule torsion dihedral angles. These solutions correspond to the decision space that characterized the multi-objective optimization. It is worth noting that we have also applied the grid-based methodology to define the ligand–receptor binding site.

**Objective Space:** We have applied two bi-objective formulations. In the first formulation, we have optimized the intermolecular energy ( $E_{inter}$ ) and the intramolecular energy ( $E_{intra}$ ). The values of these terms are given from the AutoDock energy function [2] (see Eq. 3.4), being opposite between them [7], and therefore giving rise to a multi-objective approach of this problem as follows:

- **Objective 1:** the  $E_{inter}$  energy (see Eq. 3.5) is estimated by the difference of the bound and unbound states of the ligand–macromolecule complex. The  $E_{inter}$  energy describes the binding affinity of the conformation.
- **Objective 2:** The  $E_{intra}$  energy (see Eq. 3.6) of the ligand and receptor is estimated by the difference between the bound and unbound states of the ligand and receptor. The  $E_{intra}$  characterizes the stability of the ligand in terms of energy.



$$\Delta G = E_{inter} + E_{intra} + \Delta S_{conf} \quad (3.4)$$

$$E_{inter} = (Q_{bound}^{R-L} - Q_{unbound}^{R-L}) \quad (3.5)$$

$$E_{intra} = (Q_{bound}^{L-L} - Q_{unbound}^{L-L}) + (Q_{bound}^{R-R} - Q_{unbound}^{R-R}) \quad (3.6)$$

For Eq. 3.5 and Eq. 3.6, each pair of energetic evaluations terms are described in Eq. 3.5 in subsection 3.2.1. The  $\Delta S_{conf}$  term is described in Eq. 3.2.

In the second multi-objective formulation, we have optimized the intermolecular energy ( $E_{inter}$ ) and the RMSD score. These two measures are contrary and consequently a bi-objective optimization approach is reasonable with these measures. These objectives are calculated by the following equations:

- **Objective 1:** the  $E_{inter}$  energy (see Eq. 3.5) of the ligand and receptor is estimated by the difference between the bound and unbound states of the ligand and receptor. The  $E_{inter}$  energy describes the binding affinity of the conformation.
- **Objective 2:** The RMSD is a measure of distance between the co-crystallized ligand in the receptor and the predicted position of the docking ligand (see Eq. 3.7). The RMSD score is a measure to compare the accuracy of the results obtained from the computational docking approaches. The RMSD takes into account symmetry, partial symmetry (e.g. symmetry within a rotatable branch) and near-symmetry in a simple heuristic way [3]. The lower the RMSD score, the better the docking solution is. The RMSD cutoff of 2Å is widely considered as a criterion to consider the computed ligand–protein conformation as a good prediction among the research community. This measure is very useful in those cases in which the ligand pose to the macromolecule is known. It is worth mentioning that, from a pharmacological point of view, a ligand conformation with an RMSD score of 0Å (the co-crystallized and computed ligands completely overlap) is not the best solution as the macromolecule could involve other ligand binding sites, which have not been discovered yet.

$$RMSD_{ab} = \max(RMSD'_{ab}, RMSD'_{ba}), \text{ with } RMSD'_{ab} = \sqrt{\frac{1}{N} \sum_i \min_j r_2^{ij}} \quad (3.7)$$

The sum is over all  $N$  heavy atoms in structure  $a$ , the minimum is over all atoms in structure  $b$  with the same element type as atom  $i$  in structure  $a$ .

### 3.3 Review of the State-of-the-Art

Over the last two decades, different metaheuristics have been applied as search methods to solve the docking problem [78]. One example is the docking software AutoDock, which incorporates three metaheuristic techniques. AutoDock is considered to be the most cited and one of the most used software packages in molecular modeling studies to discovery new compounds [6] as is reported in [72].

AutoDock was released in 1990, and it included a rapid search method using Monte Carlo simulated annealing [79]. However, this method proved to be inadequate for ligands with more than eight rotatable bonds [73]. Eight years later, in an attempt to improve the software, AutoDock 3.0 was released, adding the Genetic Algorithm (GA) and the Lamarckian Genetic Algorithm (LGA), which incorporates a local search, and an empirical binding free energy force that enables the prediction of the free binding energies. Docking analyses have demonstrated that the LGA is the most efficient search method of the three AutoDock algorithms in terms of the lowest energy found in a number of energy function evaluations [73]. AutoDock 4 was presented in 2009 [2]. It allows conformational models of side chains of proteins, provides torsional degrees of freedom and tries to solve the problem of flexibility in the receptor, a challenge in docking approaches as we have mentioned in section 3.2. More recently, a new release has appeared, AutoDock 4.2, which incorporates several enhancements over AutoDock 4. The latest version includes a default unbounded state, different to the extended unbounded state of AutoDock 4, an improvement over the time required to run a high-quality docking with flexible and rigid components, involving an attempt to ensure compatibility between the different releases of AutoDock software.

In 2010, as an improvement on the previous releases, the AutoDock authors implemented a new program for molecular docking called AutoDock Vina [3]. A study performed by Chang *et al.* [80] compares the two softwares in drug virtual screening being AutoDock and AutoDock Vina very accurate for virtual screening in cases in which ligands had fewer than eight rotatable bonds. The results shown that AutoDock Vina was faster than other molecular docking tools. This can be explained by the improvements implemented in AutoDock Vina such as multithreading to speed up the execution in multicore processors, and the use of an iterated local search algorithm (ILS) as the search engine. The stopping condition of the ILS is adaptively determined, thus making it difficult to compare it fairly with other techniques that use a fixed number of function evaluations.

A number of approaches can be found in the current literature that have proposed metaheuristic techniques designed around AutoDock versions. Atilgan *et al.* [81] developed a new program named AutoDockX which incorporates a sustainable GA, namely Age-Layered Population Structure (ALPS), including the age attribute for individuals. Chen *et al.* [82] presented an algorithm called SODOCK, which is an adaptation of PSO including Solis and Wets local search and uses an older version of the AutoDock energy function (version 3.05). Two other PSO related proposals are the varCPSO-ls algorithm, an extension of the CPSO algorithm with a local optimizer which is embedded inside the AutoDock 3 source code and uses its energy function [83], and the FIPSDock algorithm, which adopts the AutoDock 4.2 energy function [84]. DE has also been applied in this context. A first attempt is DockDE [85], a variant of DE which uses an older version of the AutoDock energy function. ODE is also an extension of DE enhanced by a local search algorithm and a pseudo-elitism operator, and using the AutoDock scoring function [86]. A more recent version is SADock [87], that incorporates a Hooke Jeeves local search.

Among studies on molecular docking with metaheuristics not based on AutoDock, there are several approaches that are also worth mentioning. PARADockS is a framework implemented to predict the ligand-protein interaction adapting PSO to several objective functions [88]. An Ant Colony Optimization (ACO) approach is also presented in [89] using a systematic molecular simulator. A variant of DE called MolDock was parallelized on both GPU and CPU using a fitness function designed by the authors [90]. However, although the technique is adapted to a flexible docking receptor, it has not been evaluated using flexible targets. Other optimization methods such as multi-scale optimization models and information entropy-based searching techniques with narrowing space were applied in a new docking algorithm [91]. Herberlé *et al.* [92] review EAs applied to Mycobacterium tuberculosis docking targets.

In terms of analyzing the influence of algorithm operators and parameters, a study developed by Thomsen [93] compared the performance of the LGA and DockEA algorithms by selecting different EA operators, populations and usage of local search. However, the parameter setting study

proposed, included very few docking problems and was only applied to the DockEA algorithm. Another interesting study performed by Tavares *et al.* [94] investigated the effects of Gaussian and Cauchy mutation operators through a locality analysis (small genotype variations imply small variations in phenotype); the results showed that Gaussian-based operators had a stronger locality than Cauchy-based operators. They also demonstrated that the results of runs using the Gaussian-based operator were better than those returned by the Cauchy-based operator.

There are only a few articles which can be found in the literature concerning the multi-objective optimization applied to the molecular docking problem. A first attempt was carried out in 2006 by Oduguwa *et al.* [95], in which three evolutionary multi-objective optimization algorithms (NSGA-II, PAES, and SPEA) were evaluated on three molecular complexes. Grosdidier *et al.* [96] proposed a new hybrid evolutionary algorithm called EADock, which was interfaced with the CHARMM package for energy calculations. In 2008, Janson *et al.* [7] designed a parallel multi-objective optimization algorithm using AutoDock energy function version 3.05, called ClustMPSO, that used K-Means to guide the migration strategy when dealing with six molecular complexes. In this study, the two objective to optimize were the  $E_{inter}$  and  $E_{intra}$ . Also, in 2008, Boisson *et al.* [97] implemented a parallel evolutionary bi-objective model using ParadisEO platform and GOLD for the docking of six instances. Sandoval-Perez *et al.* [18] used the implementation of NSGA-II provided by the jMetal framework [98] to optimize bound and non-bound energy terms as objectives applied to four docking instances.

These publications mentioned above, although proposed different approaches, they performed only limited comparisons with other current multi/single-objective techniques. Furthermore, a low number of molecular instances were used in these studies and were not flexible. In the area of ligand design, there have been several studies that apply the multi-objective approach. Sanchez-Faddeev *et al.* [99] proposed a bi-objective optimization approach using the SMS-EMOA to solve the problem of finding a peptide ligand. The results obtained show the possibility to design a peptide ligand of the  $\Gamma 1$  isoform of the 14-3-3 protein with predicted selectivity over the  $\epsilon 1$  isoform. Van der Horst *et al.* [100] used the multi-objective evolutionary algorithm (MOEA) for *de novo* ligand design applied to the new adenosine receptor antagonists. The selection of the candidate A1 adenosine receptor antagonists was based on multiple criteria and several objectives such as the high predicted affinity and the selectivity of the ligands for the receptors and properties like the ADMET score.



## Chapter 4

# Published Work

We have published several research studies based on the application to multi-objective metaheuristics to solve the molecular docking problem. Specifically, four articles have been published in journals indexed in the Journal of Citation Report (JCR) from the Institute of Scientific Information. In addition to this, four articles have been published in congresses. Two of them have been published in international congresses and the rest in national congresses.

### 4.1 List with Research Contributions

These four JCR articles apply metaheuristics to solve the molecular docking problem. These contributions can be organized as follows:

#### Articles published in journal indexed in JCR:

- E. López-Camacho, M. J. García-Godoy, A. J. Nebro, and J. F. Aldana-Montes. “jMetalCpp: optimizing molecular docking problems with a C++ metaheuristic framework”. *Bioinformatics* 30.3 (Feb. 2014), pp. 437–438. DOI: 10.1093/bioinformatics/btt679  
Impact Factor: 4,981. Q1 (3/57) in the category of mathematical and computational biology.
- E. López-Camacho, M. J. García-Godoy, J. García-Nieto, A. J. Nebro, and J. F. Aldana-Montes. “Solving molecular flexible docking problems with metaheuristics: A comparative study”. *Applied Soft Computing* 28 (2015), pp. 379–393. ISSN: 1568-4946. DOI: 10.1016/j.asoc.2014.10.049  
Impact Factor: 2,857. Q1 (21/130) in the category of Computer Science and Artificial Intelligence.
- M. J. García-Godoy, E. López-Camacho, J. García Nieto, A. J. Nebro, and J. F. Aldana-Montes. “Solving Molecular Docking Problems with Multi-Objective Metaheuristics”. *Molecules* 20.6 (2015), pp. 10154–10183. DOI: 10.3390/molecules200610154  
Impact Factor: 2,465. Q2 (25/59) in the category of Chemistry, Organic.
- M. J. García-Godoy, E. López-Camacho, J. García-Nieto, A. J. Nebro, and J. F. Aldana-Montes. “Molecular Docking Optimization in the Context of Multi-Drug Resistant and Sensitive EGFR Mutants”. *Molecules* 21.11 (2016), p. 1575. ISSN: 1420-3049. DOI: 10.3390/molecules21111575  
Impact Factor: 2,465. Q2 (25/59) in the category of Chemistry, Organic.



**Articles published in international congresses:**

- E. López-Camacho, M. J. García-Godoy, J. García-Nieto, A. J. Nebro, and J. F. Aldana-Montes. “A New Multi-objective Approach for Molecular Docking Based on RMSD and Binding Energy”. *Algorithms for Computational Biology: Third International Conference, AlCoB 2016, Trujillo, Spain, June 21-22, 2016, Proceedings*. Ed. by M. Botón-Fernández, C. Martín-Vide, S. Santander-Jiménez, and M. A. Vega-Rodríguez. Cham: Springer International Publishing, 2016, pp. 65–77. ISBN: 978-3-319-38827-4. DOI: 10.1007/978-3-319-38827-4\_6
- J. García-Nieto, E. López-Camacho, M. J. García Godoy, A. J. Nebro, J. J. Durillo, and J. F. Aldana-Montes. “A Study of Archiving Strategies in Multi-objective PSO for Molecular Docking”. *Swarm Intelligence: 10th International Conference, ANTS 2016, Brussels, Belgium, September 7-9, 2016, Proceedings*. Ed. by M. Dorigo, M. Birattari, X. Li, M. López-Ibáñez, K. Ohkura, C. Pinciroli, and T. Stützle. Cham: Springer International Publishing, 2016, pp. 40–52. ISBN: 978-3-319-44427-7. DOI: 10.1007/978-3-319-44427-7\_4

**Articles published in national congresses:**

- E. López-Camacho, M. J. García-Godoy, J. García-Nieto, A. J. Nebro, and J. F. Aldana-Montes. “Docking Inter/Intra-Molecular mediante metaheurísticas multi-objetivo”. *X Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados, MAEB 2015, Mérida, Spain, February 4-6. 2015*
- E. López-Camacho, M. J. García-Godoy, J. García-Nieto, A. J. Nebro, and J. F. Aldana-Montes. “Estudio de Estrategias de Archivo en PSO Multi-Objetivo para el Docking Molecular”. *XI Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados, MAEB 2016 (CAEPIA'16), Salamanca, Spain, September 13-15. 2016*, pp. 113–122

## 4.2 Summary of the articles that support the thesis

This section summarizes the articles that support this thesis. All these papers are related to the application of the single-objective and multi-objective optimizations to solve the problem of the molecular docking. In the first article, we have described the integration of AutoDock and jMetal and its application in the domain of molecular docking. In the second published article, we have performed a study comparing the mono-objective techniques using a set of flexible instances. In a third and fourth study, we have applied a set of multi-objective metaheuristics that optimize two objectives, guiding the algorithm to search the best molecular docking solutions.

### 4.2.1 jMetalCpp: optimizing molecular docking problems with a C++ metaheuristic framework

**Reference:** [8] E. López-Camacho, M. J. García-Godoy, A. J. Nebro, and J. F. Aldana-Montes. “jMetalCpp: optimizing molecular docking problems with a C++ metaheuristic framework”. *Bioinformatics* 30.3 (Feb. 2014), pp. 437–438. DOI: 10.1093/bioinformatics/btt679

Page in this thesis: 50

In [8] we introduced jMetalCpp, the C++ version of the metaheuristic framework jMetal (originally written in Java). We also presented the combination of this software with the widely used AutoDock, which is the most used tool to solve molecular docking problems. The inclusion of jMetalCpp inside the AutoDock provided the latter several additional metaheuristic techniques to

solve molecular docking problems. Both new softwares (the standalone jMetalCpp and the “fusion” of jMetalCpp with AutoDock) were published online<sup>1,2</sup> to be freely used by the scientific community.

#### 4.2.2 Solving molecular flexible docking problems with metaheuristics: a comparative study

**Reference:** [20] E. López-Camacho, M. J. García-Godoy, J. García-Nieto, A. J. Nebro, and J. F. Aldana-Montes. “Solving molecular flexible docking problems with metaheuristics: A comparative study”. *Applied Soft Computing* 28 (2015), pp. 379–393. ISSN: 1568-4946. DOI: 10.1016/j.asoc.2014.10.049

Page in this thesis: 51

In our first single-objective study [20], we tested the performance of new metaheuristic techniques apart from those included in the AutoDock Tools for solving molecular docking problems. This study approached the problem as a single-objective optimization problem, as AutoDock does. AutoDock provides two different techniques to solve the problem: a Lamarckian Genetic Algorithm (LGA), which includes local search, and a common Genetic Algorithm (GA). We added four single-objective metaheuristic: generational Genetic Algorithm (gGA), steady-state Genetic Algorithm (ssGA), Differential Evolution (DE) and Particle Swarm Optimization (PSO). A study with 75 protein-ligand complexes taken from PDB was carried on using the same fitness function and configuration parameters than AutoDock to have a comparison as fair as possible. The objective was the binding energies in kcal/mol associated with the receptor-ligand complex, as explained in Section 3.2.1. Therefore, the lower the binding energy the better the result.

This study had two different steps. In the first one, we tuned the parameter configuration of the four single-objective metaheuristic techniques. In order to do so, we selected 11 protein-ligand complexes taken from the PDB database. After we obtained a set of configuration parameters for the 4 single-objective metaheuristics, a thorough comparison was made between these four and the algorithms provided by AutoDock. This time, a set of 75 instances also from PDB were used.

It was demonstrated that DE (jMetal) obtained the best results in 67 of the 75 instances, followed by LGA (AutoDock) that achieved the best results in the remaining eight instances (1B6L, 1BDL, 1HEF, 1HIV, 1HPO, 1K6C, 1Z1H and 1ZIR). These results were provided with statistical confidence ( $\alpha = 0.05$ ) as a series of non-parametric statistical tests were applied. In particular, Friedman’s ranking and Holm’s post-hoc multicompare tests were calculated and showed that DE achieved a statistically better performance than the rest of the other analyzed algorithms. This fact is remarkable as the AutoDock algorithms are specifically designed to solve molecular docking problems. It was also noted that DE showed a slower convergence behavior, though with more successful solutions than its competitors. However, gGA demonstrated a fast convergence, and also achieved high-quality solutions, so this algorithm could be a good choice when looking for fast, but good enough solutions.

#### 4.2.3 A new multi-objective approach for molecular docking based on RMSD and binding energy

**Reference:** [21] E. López-Camacho, M. J. García-Godoy, J. García-Nieto, A. J. Nebro, and J. F. Aldana-Montes. “A New Multi-objective Approach for Molecular Docking Based on RMSD and Binding Energy”. *Algorithms for Computational Biology: Third International Conference, AlCoB 2016, Trujillo, Spain, June 21-22, 2016, Proceedings*. Ed. by M. Botón-Fernández, C. Martín-Vide,

<sup>1</sup><http://jmetalcpp.sourceforge.net/>

<sup>2</sup><http://khaos.uma.es/autodockjmetal/>



S. Santander-Jiménez, and M. A. Vega-Rodríguez. Cham: Springer International Publishing, 2016, pp. 65–77. ISBN: 978-3-319-38827-4. DOI: 10.1007/978-3-319-38827-4\_6

Page in this thesis: 52

This work was presented in the 3rd International Conference on Algorithms for Computational Biology (AlCoB 2016), celebrated in Trujillo, Spain in June of 2016. It was derived from the idea of taking a multi-objective optimization approach to solve molecular docking problems. In the beginning, the strategy we had followed was to decompose the final binding energy (the minimization objective of the previous work) into several components, particularly the intra- and inter-molecular energy [101]. After that, it was decided to use as objectives the same energy taken in the single-objective study and the RMSD. These concepts were explained in more detail in Section 3.2.2.

However, in this paper [21], we selected four representative multi-objective optimization algorithms such as NSGA-II, GDE3, SMPSO and MOEA/D. A benchmark composed of 11 complexes having receptor and ligand flexibility was selected. The selection of these complexes was motivated as they are docking problems containing a wide range of ligand sizes (from small to large inhibitors). Two quality indicators were calculated to measure the performance of each algorithm: Hypervolume ( $I_{HV}$ ) and Unary Additive Epsilon Indicator ( $I_{\epsilon+}$ ). The first indicator takes into account both convergence and diversity, whereas the second one ( $I_{\epsilon+}$ ) gives a measure of the convergence degree of the obtained Pareto front approximations. It is worth noting that, as we are dealing with real-world optimization problems, the true Pareto fronts needed to calculate these metrics are not known, so they had to be obtained using all the approximated fronts from all the executions of all the multi-objective algorithms for each problem.

The  $I_{HV}$  is the sum of the contributed volume of each point of a front in respect to a reference point, so the higher the convergence and diversity degrees of a front, the higher its  $I_{HV}$  value. According to these results, SMPSO achieved the best  $I_{HV}$  values in all the eleven problems, being MOEA/D the second best performing technique. It is important to note that many algorithms had a  $I_{HV}$  value equal to zero, this happens when all the points of the produced fronts are beyond the limits of the reference point. This happened in most of the problems in all the algorithms excepting SMPSO, what leaded us to think that we are facing a hard optimization problem. Also, SMPSO achieved the best performance results according the  $I_{\epsilon+}$  indicator (in this case, the lower the value, the better). SMPSO achieved the best values for all 11 instances except for 1HTF, in which it got the second best value. MOEA/D, which was the algorithm that got the best value for 1HTF, achieved the second best values for 9 instances. GDE3 got the second best value in one instance (1HPX) while NSGA-II got the worst results for all the instances.

After this work was presented, it was invited to be substantially extended and be submitted to a special issue of the journal IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB, 2014 JCR impact factor: 1.438, quartile Q1). To this day, it is still under-review.

#### 4.2.4 A study of archiving strategies in multi-objective PSO for molecular docking

**Reference:** [22] J. García-Nieto, E. López-Camacho, M. J. García Godoy, A. J. Nebro, J. J. Durillo, and J. F. Aldana-Montes. “A Study of Archiving Strategies in Multi-objective PSO for Molecular Docking”. *Swarm Intelligence: 10th International Conference, ANTS 2016, Brussels, Belgium, September 7-9, 2016, Proceedings*. Ed. by M. Dorigo, M. Birattari, X. Li, M. López-Ibáñez, K. Ohkura, C. Pinciroli, and T. Stützle. Cham: Springer International Publishing, 2016, pp. 40–52. ISBN: 978-3-319-44427-7. DOI: 10.1007/978-3-319-44427-7\_4

Page in this thesis: 53

This work [22] was presented in the 10th International Conference on Swarm Intelligence (ANTS 2016), celebrated in Brussels, Belgium in September of 2016. It is the natural continuation from the



previous one, where we obtained that SMPPO obtained best overall results when applying a multi-objective approach to solve molecular docking problems. The previous experiment was replicated using several SMPPO variants based on different archiving strategies. The selected variants are:  $SMPPO_{hv}$ , SMPPOD and SMPPOC. The original SMPPO and OMOPPO (the algorithm which SMPPO was inspired from) were also included in the comparison.

This paper introduced the variant named SMPPOC. It is characterized by the use of a cosine similarity when calculating the density value of each point in the solution front. The variant SMPPOD was also presented in this paper for the first time. It is an archive-less approach, implemented as an aggregative version of SMPPO inspired by MOEA/D.

According to the  $I_{HV}$  indicator,  $SMPPO_{hv}$  obtained the best results for all the 11 instances, whereas SMPPOD got the second best value in 6 instances, SMPPOC in three and the original SMPPO in two, respectively. In the same manner,  $SMPPO_{hv}$  obtained again the best values for the 11 instances according to the  $I_{\epsilon+}$  indicator. The second best values were achieved by SMPPOD in 7 instances, the original SMPPO in three and SMPPOC in one instance, respectively.

### 4.3 Summary of other publications related to this thesis

This section briefly comments the other four articles that do not support this thesis but are related to its topic. Two of them were published in the *Molecules* journal and the other two were published in national congresses.

In [101], we presented our first multi-objective approach. The final binding energy (the minimization objective of the mono-objective study) was decomposed into the intra- and inter-molecular energy. These two components were used as two contrary objectives. We selected six multi-objective optimization algorithms such as NSGA-II, ssNSGA-II, GDE3, SMPPO, MOEA/D and SMS-EMOA. A heterogeneous set of 11 protein-ligand complexes with flexible ligands and receptors was selected in order to carry out the experiments. A use case of drug discovery that involves the aeroplysinin-1 compound and the human Epidermal Growth Factor (EGFR) was also provided. The results demonstrated that according to the use cases presented, it can be more interesting to select a specific docking solution with a balanced tradeoff between the  $E_{inter}$  and  $E_{intra}$  values.

In [102], we presented our latest multiobjective approach. This time, we selected the final binding energy and the RMSD as optimization objectives and NSGA-II, GDE3, SMPPO and MOEA/D as multi-objective optimization algorithms. In this study, we performed an analysis on binding sites in the EGFR kinase domain and molecular interactions. The use cases were based on instances with wild-type EGFR, EGFR with mutations L858R and G719S and EGFR double mutants (T790M/L858R and T790M/G719S). This proposed approach can be used for *in silico* studies to test other analog kinase inhibitors or similar compounds for drug discovery in those cancers in which therapeutic targets are changed by somatic mutations.

The two latter articles were published in the X and XI ‘Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB)’ in 2015 and 2016. The first article [103] presented our first multi-objective approach (using the intra- and inter-molecular energy as contrary objectives). The following year, we presented our multi-objective PSO study in this same congress [104].

### 4.4 Copies of the articles that support the thesis

This section includes copies of the four articles summarized in Section 4.2

E. López-Camacho, M. J. García-Godoy, A. J. Nebro, and J. F. Aldana-Montes. “jMetalCpp: optimizing molecular docking problems with a C++ metaheuristic framework”. *Bioinformatics* 30.3 (Feb. 2014), pp. 437–438. DOI: [10.1093/bioinformatics/btt679](https://doi.org/10.1093/bioinformatics/btt679)

**Motivation:** Molecular docking is a method for structure-based drug design and structural molecular biology, which attempts to predict the position and orientation of a small molecule (ligand) in relation to a protein (receptor) to produce a stable complex with a minimum binding energy. One of the most widely used software packages for this purpose is AutoDock, which incorporates three metaheuristic techniques. We propose the integration of AutoDock with jMetalCpp, an optimization framework, thereby providing both single- and multi-objective algorithms that can be used to effectively solve docking problems.

**Results:** The resulting combination of AutoDock + jMetalCpp allows users of the former to easily use the metaheuristics provided by the latter. In this way, biologists have at their disposal a richer set of optimization techniques than those already provided in AutoDock. Moreover, designers of metaheuristic techniques can use molecular docking for case studies, which can lead to more efficient algorithms oriented to solving the target problems.

**Availability and implementation:** jMetalCpp software adapted to AutoDock is freely available as a C++ source code at <http://khaos.uma.es/AutodockjMetal/>.

E. López-Camacho, M. J. García-Godoy, J. García-Nieto, A. J. Nebro, and J. F. Aldana-Montes. “Solving molecular flexible docking problems with metaheuristics: A comparative study”. *Applied Soft Computing* 28 (2015), pp. 379–393. ISSN: 1568-4946. DOI: 10.1016/j.asoc.2014.10.049

The main objective of the molecular docking problem is to find a conformation between a small molecule (ligand) and a receptor molecule with minimum binding energy. The quality of the docking score depends on two factors: the scoring function and the search method being used to find the lowest binding energy solution. In this context, AutoDock 4.2 is a popular C++ software package in the bioinformatics community providing both elements, including two genetic algorithms, one of them endowed with a local search strategy. This paper principally focuses on the search techniques for solving the docking problem. In using the AutoDock 4.2 scoring function, the approach in this study is twofold. On the one hand, a number of four metaheuristic techniques are analyzed within an extensive set of docking problems, looking for the best technique according to the quality of the binding energy solutions. These techniques are thoroughly evaluated and also compared with popular well-known docking algorithms in AutoDock 4.2. The metaheuristics selected are: generational and a steady-state Genetic Algorithm, Differential Evolution, and Particle Swarm Optimization. On the other hand, a C++ version of the jMetal optimization framework has been integrated inside AutoDock 4.2, so that all the algorithms included in jMetal are readily available to solve docking problems. The experiments reveal that Differential Evolution obtains the best overall results, even outperforming other existing algorithms specifically designed for molecular docking.

E. López-Camacho, M. J. García-Godoy, J. García-Nieto, A. J. Nebro, and J. F. Aldana-Montes. “A New Multi-objective Approach for Molecular Docking Based on RMSD and Binding Energy”. *Algorithms for Computational Biology: Third International Conference, AlCoB 2016, Trujillo, Spain, June 21-22, 2016, Proceedings*. Ed. by M. Botón-Fernández, C. Martín-Vide, S. Santander-Jiménez, and M. A. Vega-Rodríguez. Cham: Springer International Publishing, 2016, pp. 65–77. ISBN: 978-3-319-38827-4. DOI: 10.1007/978-3-319-38827-4\_6

Ligand-protein docking is an optimization problem based on predicting the position of a ligand with the lowest binding energy in the active site of the receptor. Molecular docking problems are traditionally tackled with single-objective, as well as with multi-objective approaches, to minimize the binding energy. In this paper, we propose a novel multi-objective formulation that considers: the Root Mean Square Deviation (RMSD) difference in the coordinates of ligands and the binding (intermolecular) energy, as two objectives to evaluate the quality of the ligand-protein interactions. To determine the kind of Pareto front approximations that can be obtained, we have selected a set of representative multi-objective algorithms such as NSGA-II, SMPSO, GDE3, and MOEA/D. Their performances have been assessed by applying two main quality indicators intended to measure convergence and diversity of the fronts. In addition, a comparison with LGA, a reference single-objective evolutionary algorithm for molecular docking (AutoDock) is carried out. In general, SMPSO shows the best overall results in terms of energy and RMSD (value lower than 2Å for successful docking results). This new multi-objective approach shows an improvement over the ligand-protein docking predictions that could be promising in in silico docking studies to select new anticancer compounds for therapeutic targets that are multidrug resistant.

J. García-Nieto, E. López-Camacho, M. J. García Godoy, A. J. Nebro, J. J. Durillo, and J. F. Aldana-Montes. “A Study of Archiving Strategies in Multi-objective PSO for Molecular Docking”. *Swarm Intelligence: 10th International Conference, ANTS 2016, Brussels, Belgium, September 7-9, 2016, Proceedings*. Ed. by M. Dorigo, M. Birattari, X. Li, M. López-Ibáñez, K. Ohkura, C. Pinciroli, and T. Stützle. Cham: Springer International Publishing, 2016, pp. 40–52. ISBN: 978-3-319-44427-7. DOI: 10.1007/978-3-319-44427-7\_4

Molecular docking is a complex optimization problem aimed at predicting the position of a ligand molecule in the active site of a receptor with the lowest binding energy. This problem can be formulated as a bi-objective optimization problem by minimizing the binding energy and the Root Mean Square Deviation (RMSD) difference in the coordinates of ligands. In this context, the SMPSO multi-objective swarm-intelligence algorithm has shown a remarkable performance. SMPSO is characterized by having an external archive used to store the non-dominated solutions and also as the basis of the leader selection strategy. In this paper, we analyze several SMPSO variants based on different archiving strategies in the scope of a benchmark of molecular docking instances. Our study reveals that the SMPSO<sub>hv</sub>, which uses an hypervolume contribution based archive, shows the overall best performance.



## Chapter 5

# Conclusions and Future Work

This chapter exposes the final ideas of this dissertation. Section 5.1 contains the conclusions obtained in all the past experiments. Then, in Section 5.2 we explained the future lines of work that we plan to explore from the latter works.

### 5.1 Conclusions

When tackling molecular docking problems, the available techniques to solve them have not changed over the last years. As these problems can be formulated as multi-objective optimization problems, our intention was to study and provide a set of modern metaheuristic techniques to solve them. As the most used molecular docking tool (AutoDock) was coded in C++, we embarked on the task of the creation of a port of the metaheuristic framework jMetal in this language: jMetalCpp. This way, we have provided the research community with a powerful and open-source tool that can be freely used.

The implementation of the jMetalCpp framework provides advantages to researchers both in drug discovery and other life sciences domains who are interested in having more modern techniques that will help them to solve different problems like molecular docking. We have already demonstrated that different techniques exist apart than the ones that are commonly used to solve molecular docking problems and that they can lead to higher quality results. The inclusion of jMetalCpp into the widely used tool AutoDock provides other researchers with a collection of metaheuristics and tools additional from those that are already included in AutoDock. It also provides an easy structure for more advanced users with C++ coding skills to incorporate their own techniques to solve molecular docking problems. This tool is publicly online and has been already downloaded by researchers from different parts of the world. The standalone jMetalCpp framework is also available for researchers to be used for solving optimization problems of other domains. It has been downloaded hundreds of times from all the world<sup>1</sup> and we have been in contact with people who wanted to contribute to the code adding their own tools and algorithms, and use it in their own research work.

Using AutoDock+jMetal, a study was done using single-objective metaheuristics where we included more algorithms (apart from those already included by AutoDock) to solve a large benchmark of protein-ligand complexes. The study was carried on taking the same configuration parameters that commonly were used in the AutoDock publications. We proved that other single-objective metaheuristics could lead to higher quality results. In our case, the differential evolution algorithm proved to be a better candidate when solving molecular docking problems.

---

<sup>1</sup>1,917 downloads from SourceForge at the present day



When tackling molecular docking problems using a multi-objective approach, a set of solutions is returned at the end of one execution instead of a single solution. This set of solutions provides the end user with several possibilities from where to choose depending of the weight she/he wants to give to each of the optimization objectives. So, we have considered two different multi-objective approaches in our studies. The first one was based on decomposing the final binding energy (the objective function that is minimized by the single-objective algorithms) into several components. We selected the intra- and inter-molecular energies as optimization objectives. This resulted in a set of solutions where the end user could select from depending on the importance that he gives to each one of the energies.

The other multi-objective formulation used the same objective as the single-objective formulation (the binding energy) and the RMSD. The use of RMSD as objective to guide the search is useful in those typical cases in which the active site of a given therapeutic target mutates and makes it multi-drug resistant. Using this approach, a broad set of solutions are returned, which can be selected according to the weight of the RMSD and binding energy, instead of only focusing on energy values. A first study was made using four multi-objective algorithms: NSGA-II, SMPSO, GDE3 and MOEA/D. In this experiment, an heterogeneous set of 11 protein-ligand complexes with flexible ligands and receptors were selected as problem instances. SMPSO provided the best overall performance according to the two quality indicators used ( $I_{HV}$  and  $I_{\epsilon+}$ ) and for the studied molecular instances, being MOEA/D the algorithm with second best values. Also, from a single-objective point of view, the solutions obtained from SMPSO were better than those obtained from the LGA algorithm from AutoDock. This was remarkable as SMPSO is a general purpose optimization algorithm, whereas LGA is specifically adapted to deal with the molecular docking problem. Finally, it is interesting to note that SMPSO converged to the region biased towards the RMSD objective, whereas MOEA/D placed its solutions in the opposite region of the generated fronts of non-dominated solutions.

From the results obtained in the last study, a new one was carried on where several SMPSO variants with different archiving strategies would be tested. The selected variants were: SMPSO<sub>hv</sub>, SMPSOD and SMPSOC. The original SMPSO and OMOPSO (the algorithm which SMPSO was inspired from) were also included in the comparison. The previous multi-objective study was replicated using these six algorithms and the same configurations than before. According to our two usual quality indicators ( $I_{HV}$  and  $I_{\epsilon+}$ ), SMPSO<sub>hv</sub> was revealed to obtain the best values, followed by SMPSOD, SMPSOC and SMPSO. The former variant obtained the best  $I_{HV}$  as it included a leader selection method of those non-dominated solutions (from the external archive) having the largest hypervolume contributions, which seemed to be responsible of the best diversity and convergence values in this comparison. OMOPSO showed moderated results, although reaching outperforming outlier solutions for some instances. It is worth noting that SMPSOD variant was able to cover the reference front with non-dominated solutions in the two objectives extremes (low energy and low RMSD values respectively).

We can summarize the research work of this thesis in the fact that the use of modern multi-objective algorithms can provide the biologists with accurate solutions for the molecular docking problems. The use of these more modern variants of SMPSO instead of the common used techniques for solving the molecular docking problem have been demonstrated to achieve better results.

## 5.2 Future Work

The line of study carried on in this dissertation has lead us to plan several possible works. On one hand, some of future works emerge from the idea of continuing the tackled problem (molecular docking) and still focus on trying to improve the quality of the obtained results. On the other hand, new research lines could be started from the knowledge obtained in the previous experiments



and could be considered as “branches” of this work.

The first planned work is related to our first multi-objective study, which obtained that joining the solutions generated from SMPSO and MOEA/D algorithms covered the full Pareto front. As a future work, this led us to think that a hybrid implementation of SMPSO and MOEA/D would provide us with a broader set of solutions that would cover the reference front with non-dominated solutions in the two objective ends. The results obtained by SMPSOD in the second multi-objective study encouraged us in continuing this plan of work.

Related to the hybrid algorithm design, we plan to implement and include into jMetalCpp some operators that are specifically designed to the molecular docking problem. Until now, all the metaheuristic techniques that we have used in our studies use general purpose variation operators (crossover and mutation), so it is natural to get the conclusion that if the techniques used to solve the molecular docking are specifically designed to this concrete problem we could obtain higher quality solutions.

Other contribution to the scientific community that we want to explore is the creation of a Web service that provides the same tools that jMetalCpp integrates to the AutoDock tools. This Web service would allow molecular docking executions using all the jMetalCpp metaheuristics on one protein-ligand complex (selectable from all our previous sets or uploaded by the user). This idea emerged as some users with a more biological background could have problems trying to compile and execute our AutoDock+jMetal tool.

We also plan to work in the automatic design of algorithms in order to develop ad-hoc metaheuristics that could lead to better solutions according to the optimization objectives. Some preliminary work has already been tackled on the automatic design of algorithms but for general purpose problems. It is in our interest to apply these advances for solving the molecular docking problem.

Finally, as a more general idea, we would want to use our standalone jMetalCpp framework to solve other problems in the life sciences, and not be restricted to only molecular docking. Our tool is abstract enough to include more algorithms and to be used to solve other optimization problems from different domains. In particular, the tertiary protein structure prediction is a very promising candidate to apply the set of jMetalCpp optimization techniques.



# Bibliography

- [1] M. L. Verdonk, J. C. Cole, M. J. Hartshorn, C. W. Murray, and R. D. Taylor. “Improved protein-ligand docking using GOLD.” *Proteins* 52.4 (2003), pp. 609–623. DOI: 10.1002/prot.10465.
- [2] G. M. Morris, R. Huey, W. Lindstrom, M. F. Sanner, R. K. Belew, D. S. Goodsell, and A. J. Olson. “AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility.” *J. Comput. Chem.* 30.16 (2009), pp. 2785–2791.
- [3] O. Trott and A. J. Olson. “AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading”. *J. Comput. Chem.* 31.2 (2010), pp. 455–461. DOI: 10.1002/jcc.21334.
- [4] X.-Y. Meng, H.-X. Zhang, M. Mezei, and M. Cui. “Molecular Docking: A Powerful Approach for Structure-Based Drug Discovery”. *Curr Comput Aided Drug Des* 7 (2 2011), pp. 146–57. DOI: 10.2174/157340911795677602.
- [5] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. “The Protein Data Bank”. *Nucleic Acids Research* 28.1 (2000), pp. 235–242. DOI: 10.1093/nar/28.1.235. eprint: <http://nar.oxfordjournals.org/content/28/1/235.full.pdf+html>.
- [6] S. F. Sousa, P. A. Fernandes, and M. J. Ramos. “Protein ligand docking: Current status and future challenges”. *Proteins* 65.1 (2006), pp. 15–26.
- [7] S. Janson, D. Merkle, and M. Middendorf. “Molecular docking with multi-objective Particle Swarm Optimization”. *Appl. Soft Comput.* 8.1 (2008), pp. 666–675.
- [8] E. López-Camacho, M. J. García-Godoy, A. J. Nebro, and J. F. Aldana-Montes. “jMetalCpp: optimizing molecular docking problems with a C++ metaheuristic framework”. *Bioinformatics* 30.3 (Feb. 2014), pp. 437–438. DOI: 10.1093/bioinformatics/btt679.
- [9] R. Storn and K. Price. “Differential Evolution: A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces”. *Journal of Global Optimization* 11.4 (1997), pp. 341–359.
- [10] J. Kennedy and R. Eberhart. “Particle swarm optimization”. *IEEE IJCNN*. Vol. 4. 1995, 1942–1948 vol.4.
- [11] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. “A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II”. *IEEE Transactions on Evolutionary Computation* 6.2 (2002), pp. 182–197.
- [12] J. J. Durillo, A. J. Nebro, F. Luna, and E. Alba. “On the Effect of the Steady-State Selection Scheme in Multi-Objective Genetic Algorithms”. *Evolutionary Multi-Criterion Optimization*. Vol. 5467. LNCS. Springer Berlin Heidelberg, 2009, pp. 183–197. ISBN: 978-3-642-01019-4. DOI: 10.1007/978-3-642-01020-0\_18.



- [13] A. J. Nebro, J. J. Durillo, J. Garcia-Nieto, C. A. Coello Coello, F. Luna, and E. Alba. “SMPSO: A new PSO-based metaheuristic for multi-objective optimization”. *IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making*. Mar. 2009, pp. 66–73. DOI: 10.1109/MCDM.2009.4938830.
- [14] S. Kukkonen and J. Lampinen. “GDE3: The third Evolution Step of Generalized Differential Evolution”. *IEEE Congress on Evolutionary Computation (CEC’2005)*. Vol. 1. 2005, pp. 443–450. DOI: 10.1109/CEC.2005.1554717.
- [15] Q. Zhang and H. Li. “MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition”. *IEEE T. Evolut. Comput.* 11.6 (2007), pp. 712–731. ISSN: 1089-778X. DOI: 10.1109/TEVC.2007.892759.
- [16] N. Beume, B. Naujoks, and M. Emmerich. “SMS-EMOA: Multiobjective selection based on dominated hypervolume”. *European Journal of Operational Research* 181.3 (2007), pp. 1653–1669. ISSN: 0377-2217. DOI: 10.1016/j.ejor.2006.08.008.
- [17] C. A. Coello, G. B. Lamont, and D. A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, 2007.
- [18] A. Sandoval-Perez, D. Becerra, D. Vanegas, D. Restrepo-Montoya, and F. Niño. “A Multi-objective Optimization Energy Approach to Predict the Ligand Conformation in a Docking Process”. *EuroGP*. 2013, pp. 181–192.
- [19] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. “A Fast Elitist Non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II”. *Proceedings of the Parallel Problem Solving from Nature VI Conference*. 2000, pp. 849–858.
- [20] E. López-Camacho, M. J. García-Godoy, J. García-Nieto, A. J. Nebro, and J. F. Aldana-Montes. “Solving molecular flexible docking problems with metaheuristics: A comparative study”. *Applied Soft Computing* 28 (2015), pp. 379–393. ISSN: 1568-4946. DOI: 10.1016/j.asoc.2014.10.049.
- [21] E. López-Camacho, M. J. García-Godoy, J. García-Nieto, A. J. Nebro, and J. F. Aldana-Montes. “A New Multi-objective Approach for Molecular Docking Based on RMSD and Binding Energy”. *Algorithms for Computational Biology: Third International Conference, AlCoB 2016, Trujillo, Spain, June 21-22, 2016, Proceedings*. Ed. by M. Botón-Fernández, C. Martín-Vide, S. Santander-Jiménez, and M. A. Vega-Rodríguez. Cham: Springer International Publishing, 2016, pp. 65–77. ISBN: 978-3-319-38827-4. DOI: 10.1007/978-3-319-38827-4\_6.
- [22] J. García-Nieto, E. López-Camacho, M. J. García Godoy, A. J. Nebro, J. J. Durillo, and J. F. Aldana-Montes. “A Study of Archiving Strategies in Multi-objective PSO for Molecular Docking”. *Swarm Intelligence: 10th International Conference, ANTS 2016, Brussels, Belgium, September 7-9, 2016, Proceedings*. Ed. by M. Dorigo, M. Birattari, X. Li, M. López-Ibáñez, K. Ohkura, C. Pinciroli, and T. Stützle. Cham: Springer International Publishing, 2016, pp. 40–52. ISBN: 978-3-319-44427-7. DOI: 10.1007/978-3-319-44427-7\_4.
- [23] T. Bäck. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford University Press, 1996.
- [24] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.
- [25] F. Glover. “Future Paths for Integer Programming and Links to Artificial Intelligence”. *Computers & Operations Research* 13 (1986), pp. 533–549.
- [26] C. Reeves. *Modern Heuristic Techniques for Combinatorial Problems*. Oxford, UK: Blackwell Scientific Publishing, 1993.



- [27] C. Blum and A. Roli. “Metaheuristics in combinatorial optimization: Overview and conceptual comparison”. *ACM Computing Surveys* 35.3 (2003), pp. 268–308.
- [28] F. W. Glover and G. A. Kochenberger. *Handbook of Metaheuristics*. Kluwer, 2003.
- [29] G. Luque. “Resolución de Problemas Combinatorios con Aplicación Real en Sistemas Distribuidos”. PhD thesis. University of Málaga, 2006.
- [30] J. F. Chicano. “Metaheurísticas e Ingeniería del Software”. PhD thesis. University of Málaga, 2007.
- [31] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. “Equation of state calculations by fast computing machines”. *Journal of Chemical Physics* 21 (1953), pp. 1087–1092.
- [32] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. “Optimization by Simulated Annealing”. *Science* 220 (1983), pp. 671–680.
- [33] F. Glover. “Heuristics for Integer Programming Using Surrogate Constraints”. *Decision Sciences* 8 (1977), pp. 156–166.
- [34] T. Feo and M. Resende. “Greedy randomized adaptive search procedures”. *Journal of Global Optimization* 6 (1999), pp. 109–133.
- [35] N. Mladenovic and P. Hansen. “Variable Neighborhood Search”. *Com. Oper. Res* 24 (1997), pp. 1097–1100.
- [36] H. R. Lourenço, O. Martin, and T. Stützle. “Handbook of Metaheuristics”. Kluwer Academic Publishers, 2002. Chap. Iterated local search, pp. 321–353.
- [37] T. Stützle. *Local Search Algorithms for Combinatorial Problems Analysis, Algorithms and New Applications*. Tech. rep. DISKI Dissertationen zur Künstlichen Intelligenz. Sankt Augustin, Germany, 1999.
- [38] H. Mühlenbein. “The Equation for Response to Selection and its Use for Prediction”. *Evolutionary Computation* 5 (1998), pp. 303–346.
- [39] P. Larrañaga, R. Etxeberria, J. A. Lozano, and J. M. Peña. *Optimization by learning and simulation of Bayesian and Gaussian networks*. Tech. rep. KZZA-IK-4-99. Department of Computer Science and Artificial Intelligence, University of the Basque Country, 1999.
- [40] M. Pelikan, D. E. Goldberg, and E. Cantú-Paz. “BOA: The Bayesian optimization algorithm”. *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99*. Ed. by W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, and R. E. Smith. Vol. 1. Morgan Kaufmann Publishers, San Francisco, CA, 1999, pp. 525–532.
- [41] M. Soto, A. Ochoa, S. Acid, and L. M. de Campos. “Introducing the Polytree Aproximation of Distribution Algorithm”. *Second Symposium on Artificial Intelligence. Adaptive Systems. CIMAF 99*. 1999, pp. 360–367.
- [42] J. Whittaker. *Graphical models in applied multivariate statistics*. John Wiley & Sons, Inc., 1990.
- [43] F. Glover. “A template for Scatter Search and Path Relinking”. *Artificial Evolution*. Ed. by J.-K. H. et al. LNCS 1363. Springer, 1998, pp. 13–54.
- [44] M. Laguna and R. Martí. *Scatter Search. Methodology and Implementations in C*. Kluwer, 2003.
- [45] M. Dorigo. “Optimization, Learning and Natural Algorithms”. PhD thesis. Dipartimento di Elettronica, Politecnico di Milano, 1992.

- [46] M. Dorigo and T. Stützle. “Handbook of Metaheuristics”. Vol. 57. International Series In Operations Research and Management Science. Kluwer Academic Publisher, 2003. Chap. The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances, pp. 251–285.
- [47] J. Kennedy. “Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance”. *Proceedings of IEEE Congress on Evolutionary Computation (CEC 1999)*. 1999, pp. 1931–1938.
- [48] C. A. Coello, G. B. Lamont, and D. A. V. Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Second. Genetic and Evolutionary Computation Series. Springer, 2007.
- [49] J. L. Cohon and D. H. Marks. “A Review and Evaluation of Multiobjective Programming Techniques”. *Water Resources Research* 11.2 (1975), pp. 208–220.
- [50] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, NY, USA: John Wiley & Sons, 2001.
- [51] V. Pareto. *Cours D’Economie Politique*. Vol. I and II. Lausanne: F. Rouge, 1896.
- [52] A. Osyczka. “Multicriteria optimization for engineering design”. *Design Optimization*. Ed. by J. S. Gero. Academic Press, 1895, pp. 193–227.
- [53] M. Ehrgott. *Multicriteria Optimization*. Second. Springer, 2005.
- [54] D. A. Van Veldhuizen. “Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations”. PhD thesis. Wright-Patterson, AFB, OH: Dept. Elec. Comput. Eng., Graduate School of Eng., Air Force Inst. Technol., 1999.
- [55] F. Y. Edgeworth. *Mathematical Psychics*. London: P. Keagan, 1881.
- [56] E. Zitzler, M. Laumanns, and L. Thiele. *SPEA2: Improving the Strength Pareto Evolutionary Algorithm*. Tech. rep. 103. Computer Engineering and Networks Laboratory (TIK); Swiss Federal Institute of Technology (ETH); Zurich; Switzerland, 2001.
- [57] C. M. Fonseca and P. J. Fleming. “Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization”. *Proc. of the Fifth Int. Conference on Genetic Algorithms*. 1993, pp. 416–423.
- [58] N. Srinivas and K. Deb. “Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms”. *Evolutionary Computation* 2.3 (1994), pp. 221–248.
- [59] J. Knowles and D. Corne. “The Pareto Archived Evolution Strategy: A New Baseline Algorithm for Multiobjective Optimization”. *Proceedings of the 1999 Congress on Evolutionary Computation*. Piscataway, NJ: IEEE Press, 1999, pp. 9–105.
- [60] K. Deb. “An Efficient Constraint Handling Mechanism Method for Genetic Algorithms”. *Computer Methods in Applied Mechanics and Engineering* 186.2/4 (2000), pp. 311–338.
- [61] K. Deb. *Optimization for Engineering Design*. New Delhi: Prentice-Hall, 1995.
- [62] R. L. Graham. “Bounds on multiprocessor timing anomalies”. *SIAM Journal of Applied Mathematics* 17 (1969), pp. 416–429.
- [63] R. M. Karp. “Probabilistic analysis of partitioning algorithms for the traveling salesman problem in the plane”. *Mathematics of Operations Research* 2 (1977), pp. 209–224.
- [64] J. D. Knowles, L. Thiele, and E. Zitzler. *A tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers*. Tech. rep. TIK-Report 214. Computer Engineering and Networks Laboratory, ETHC Zurich, 2006.

- [65] S. García, D. Molina, M. Lozano, and F. Herrera. “A study on the use of non-parametric tests for analyzing the evolutionary algorithms’ behaviour: a case study on the CEC’2005 Special Session on Real Parameter Optimization”. *Journal of Heuristics* 15.6 (2008), p. 617. ISSN: 1572-9397. DOI: 10.1007/s10732-008-9080-4.
- [66] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca. “Performance assessment of multiobjective optimizers: an analysis and review”. *IEEE Transactions on Evolutionary Computation* 7.2 (2003), pp. 114–132.
- [67] E. Zitzler and L. Thiele. “Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach”. *IEEE Transactions on Evolutionary Computation* 3.4 (1999), pp. 257–271.
- [68] D. J. Sheskin. *Handbook of Parametric and Nonparametric Statistical Procedures: Fourth Edition*. 4th. Chapman and Hall/CRC, 2007. ISBN: 1584888148.
- [69] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera. “Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework.” *Journal of Multiple-Valued Logic & Soft Computing* 17 (2011).
- [70] L. G. Ferreira, R. N. dos Santos, G. Oliva, and A. D. Andricopulo. “Molecular Docking and Structure-Based Drug Design Strategies”. *Molecules* 20.7 (2015), pp. 13384–13421. DOI: 10.3390/molecules200713384.
- [71] M. W. Chang, C. Ayeni, S. Breuer, and B. E. Torbett. “Virtual Screening for HIV Protease Inhibitors: A Comparison of AutoDock 4 and Vina”. *PLoS ONE* 5.8 (Aug. 2010), e11955+. DOI: 10.1371/journal.pone.0011955.
- [72] S. Cosconati, S. Forli, A. L. Perryman, R. Harris, D. S. Goodsell, and A. J. Olson. “Virtual screening with AutoDock: theory and practice”. *Expert Opin. Drug. Discov.* 5.6 (2010), pp. 597–607.
- [73] G. M. Morris, D. S. Goodsell, R. S. Halliday, R. Huey, W. E. Hart, R. K. Belew, and A. J. Olson. “Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function”. *J. Comput. Chem.* 19 (1998), pp. 1639–1662.
- [74] S. Dallakyan, M. E. Pique, and R. Huey. *Autodock version 4.2*. <http://autodock.scripps.edu/>.
- [75] S. J. Weiner, P. A. Kollman, D. A. Case, U. C. Singh, C. Ghio, G. Alagona, S. Profeta, and P. Weiner. “A new force field for molecular mechanical simulation of nucleic acids and proteins”. *J. Am. Chem. Soc.* 106.3 (1984), pp. 765–784.
- [76] G. M. Verkhivker, P. A. Rejto, D. Bouzida, S. Arthurs, A. B. Colson, S. T. Freer, D. K. Gehlhaar, V. Larson, B. A. Luty, T. Marrone, and P. W. Rose. “Towards understanding the mechanisms of molecular recognition by computer simulations of ligand protein interactions”. *Journal of Molecular Recognition* 12.6 (1999), pp. 371–389. ISSN: 1099-1352. DOI: 10.1002/(SICI)1099-1352(199911/12)12:6<371::AID-JMR479>3.0.CO;2-0.
- [77] C. Blum, J. Puchinger, G. R. Raidl, and A. Roli. “Hybrid metaheuristics in combinatorial optimization: A survey”. *Applied Soft Computing* 11.6 (2011), pp. 4135–4151. ISSN: 1568-4946. DOI: 10.1016/j.asoc.2011.02.032.
- [78] E.-W. Lammeijer, T. Bäck, J. N. Kok, and A. P. Ijzerman. “Evolutionary Algorithms in Drug Design”. *Nat. Comp.* 4.3 (2005), pp. 177–243.
- [79] D. S. Goodsell and A. J. Olson. “Automated docking of substrates to proteins by simulated annealing”. *Proteins*. 8 (1990), pp. 195–202.





- [80] C.-E. A. Chang, J. Trylska, V. Tozzini, and J. Andrew McCammon. “Binding Pathways of Ligands to HIV-1 Protease: Coarse-grained and Atomistic Simulations”. *Chemical Biology and Drug Design* 69.1 (2007), pp. 5–13.
- [81] E. Atilgan and J. Hu. “Efficient protein-ligand docking using sustainable evolutionary algorithm”. *GECCO*. 2010, pp. 211–212.
- [82] H.-M. Chen, B.-F. Liu, H.-L. Huang, S.-F. Hwang, and S.-Y. Ho. “SODOCK: Swarm optimization for highly flexible protein-ligand docking”. *J. Comput. Chem.* 28.2 (2007), pp. 612–623.
- [83] V. Namasivayam and R. Günther. “Research Article: pso@autodock: A Fast Flexible Molecular Docking Program Based on Swarm Intelligence”. *Chem. Biol. Drug. Des.* 70.6 (2007), pp. 475–484.
- [84] Y. Liu, L. Zhao, W. Li, D. Zhao, M. Song, and Y. Yang. “FIPSDock: A new molecular docking technique driven by fully informed swarm optimization algorithm”. *J. Comput. Chem.* 34 (2012), pp. 67–75.
- [85] R. Thomsen. “Flexible ligand docking using differential evolution”. *Evolutionary Computation, 2003. CEC '03. The 2003 Congress on*. Vol. 4. 2003, pp. 2354–2361.
- [86] M. Koochi-Moghadam and A. T. Rahmani. “Molecular docking with opposition-based differential evolution”. *Proceedings of the 27th Annual ACM Symposium on Applied Computing. SAC '12*. ACM, 2012, pp. 1387–1392.
- [87] H. W. Chung, S. J. Cho, K.-R. Lee, and K.-H. Lee. “Self-adaptive differential evolution algorithm incorporating local search for protein-ligand docking”. *Journal of Physics: Conference Series* 410.1 (2013), p. 012030.
- [88] R. Meier, M. Pippel, F. Brandt, W. Sippl, and C. Baldauf. “ParaDockS: A Framework for Molecular Docking with Population-Based Metaheuristics”. *J. Chem. Inf. Model.* 50.5 (2010), pp. 879–889.
- [89] O. Korb, T. Stutzle, and T. E. Exner. “Application of ant colony optimization to structure-based drug design”. in *Ant Colony Optimization and Swarm Intelligence, 5th International Workshop, ANTS 2006, ser. TECHNICAL REPORT SERIES: TR/IRIDIA/2006-023 11 LNCS*, M. Dorigo et al., Eds. Springer Verlag, 2006, pp. 247–258.
- [90] M. Simonsen, C. N. Pedersen, M. H. Christensen, and R. Thomsen. “GPU-accelerated high-accuracy molecular docking using guided differential evolution: real world applications”. *Proceedings of the 13th annual conference on Genetic and evolutionary computation. GECCO '11*. Dublin, Ireland: ACM, 2011, pp. 1803–1810. ISBN: 978-1-4503-0557-0. DOI: 10.1145/2001576.2001818.
- [91] L. Kang and X. Wang. “Multi-scale optimization model and algorithm for computer-aided molecular docking”. *Natural Computation (ICNC), 2012 Eighth International Conference on*. 2012, pp. 1208–1211.
- [92] G. Heberlé and W. de Azevedo Jr. “Bio-Inspired Algorithms Applied to Molecular Docking Simulations”. *Current Medicinal Chemistry* 18.9 (2011).
- [93] R. Thomsen. “Flexible ligand docking using evolutionary algorithms: investigating the effects of variation operators and local search hybrids”. *Biosystems* 72.2 (2003), pp. 57–73.
- [94] J. Tavares, A.-A. Tantar, N. Melab, and E.-G. Talbi. “The Influence of Mutation on Protein-Ligand Docking Optimization: A Locality Analysis”. *Proceedings of the 10th international conference on Parallel Problem Solving from Nature: PPSN X*. Springer-Verlag, 2008, pp. 589–598.





- [95] A. Oduguwa, A. Tiwari, S. Fiorentino, and R. Roy. “Multi-objective optimisation of the protein-ligand docking problem in drug discovery”. *Proceedings of the 8th annual conference on Genetic and evolutionary computation*. 2006, pp. 1793–1800.
- [96] A. Grosdidier, V. Zoete, and O. Michielin. “EADock: Docking of small molecules into protein active sites with a multiobjective evolutionary optimization”. *Proteins* 67.4 (2007), pp. 1010–1025.
- [97] J.-C. Boisson, L. Jourdan, E. Talbi, and D. Horvath. “Parallel multi-objective algorithms for the molecular docking problem”. *IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*. Sept. 2008, pp. 187–194. DOI: 10.1109/CIBCB.2008.4675777.
- [98] J. J. Durillo and A. J. Nebro. “jMetal: A Java framework for multi-objective optimization”. *Advances in Engineering Software* 42.10 (2011), pp. 760–771.
- [99] H. Sanchez-Faddeev, M. T. M. Emmerich, F. J. Verbeek, A. H. Henry, S. Grimshaw, H. P. Spaink, H. W. van Vlijmen, and A. Bender. “Using Multiobjective Optimization and Energy Minimization to Design an Isoform-selective Ligand of the 14-3-3 Protein”. *Proceedings of the 5th International Conference on Leveraging Applications of Formal Methods, Verification and Validation: Applications and Case Studies - Volume Part II*. Heraklion, Crete, Greece: Springer-Verlag, 2012, pp. 12–24. DOI: 10.1007/978-3-642-34032-1\_3.
- [100] E. van der Horst, P. Marqués-Gallego, T. Mulder-Krieger, J. van Veldhoven, J. W. Kruiselsbrink, A. Aleman, M. T. M. Emmerich, J. Brussee, A. Bender, and A. P. IJzerman. “Multi-Objective Evolutionary Design of Adenosine Receptor Ligands”. *Journal of Chemical Information and Modeling* 52.7 (2012), pp. 1713–1721. DOI: 10.1021/ci2005115.
- [101] M. J. García-Godoy, E. López-Camacho, J. García Nieto, A. J. Nebro, and J. F. Aldana-Montes. “Solving Molecular Docking Problems with Multi-Objective Metaheuristics”. *Molecules* 20.6 (2015), pp. 10154–10183. DOI: 10.3390/molecules200610154.
- [102] M. J. García-Godoy, E. López-Camacho, J. García-Nieto, A. J. Nebro, and J. F. Aldana-Montes. “Molecular Docking Optimization in the Context of Multi-Drug Resistant and Sensitive EGFR Mutants”. *Molecules* 21.11 (2016), p. 1575. ISSN: 1420-3049. DOI: 10.3390/molecules21111575.
- [103] E. López-Camacho, M. J. García-Godoy, J. García-Nieto, A. J. Nebro, and J. F. Aldana-Montes. “Docking Inter/Intra-Molecular mediante metaheurísticas multi-objetivo”. *X Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados, MAEB 2015, Mérida, Spain, February 4-6*. 2015.
- [104] E. López-Camacho, M. J. García-Godoy, J. García-Nieto, A. J. Nebro, and J. F. Aldana-Montes. “Estudio de Estrategias de Archivo en PSO Multi-Objetivo para el Docking Molecular”. *XI Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados, MAEB 2016 (CAEPIA'16), Salamanca, Spain, September 13-15*. 2016, pp. 113–122.



# List of Figures

2.1	Optimization techniques classification. . . . .	18
2.2	Metaheuristics classification. . . . .	22
2.3	Pareto dominance example. . . . .	27
2.4	Formulation and Pareto front for the Bihn2 problem. . . . .	28
2.5	Formulation and Pareto front for the DTLZ4 problem. . . . .	28
2.6	Pareto fronts examples with bad convergence (a) and bad diversity (b). . . . .	29
2.7	( <i>Ranking</i> ) ordering example for solutions in a MOP with two objectives. . . . .	30
2.8	Density estimator example for non-dominated solutions in a MOP with two objectives. . . . .	31
2.9	Hypervolume covered by the non-dominated solutions. . . . .	34
3.1	Solution encoding in AutoDock 4.2 and jMetal. . . . .	39

