



UNIVERSIDAD  
DE MÁLAGA

**UVIC** UNIVERSITAT DE VIC  
UNIVERSITAT CENTRAL DE CATALUNYA

Master in Omics Data Analysis



## TUTORIAL

*Multi-objective optimization with  
metaheuristics.  
Applications to multiple sequence alignment  
and other bioinformatics problems*

**Antonio J. Nebro**  
PhD in Computer Science

[ajnebro@uma.es](mailto:ajnebro@uma.es)



## Table of contents

- **About this tutorial**
- Metaheuristics as optimization techniques
- Dealing with problems with more than one objective function
- Case study: multiple sequence alignment
- Further developments and related projects

## About this tutorial

- This tutorial is aimed at offering a practical view of solving **multi-objective optimization problems** with **metaheuristics**.
- After presenting the basic concepts, which will be illustrated with problems from the Bioinformatics domain (e.g., molecular docking, phylogenetic inference) we will learn how to use the **jMetal** software framework (<https://github.com/jMetal/jMetal>) to apply evolutionary algorithms to find the solutions of academic problems.
- Then, we will focus on the particular issue of using the **M2Align** algorithm (<https://github.com/KhaosResearch/M2Align>) to address the **multiple sequence alignment problem**

## About this tutorial

- In this tutorial we will learn:
  - What are metaheuristic techniques and why they are useful.
  - Issues related to optimizing problems having more than one objective function (multi-objective optimization problems).
  - Use the jMetal framework to solve both single and multi-objective optimization problems.
  - Deal with multi-objective formulations of the multiple sequence alignment problema.

## About this tutorial

- Software requirements
  - Java JDK 8 64 bits (<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>)
  - Maven (<https://maven.apache.org/download.cgi>)
  - IntelliJ Idea Community Edition (<https://www.jetbrains.com/idea/download>)
  - Git (<https://git-scm.com/>) and account in GitHub
  - jMetal (<https://github.com/jMetal/jMetal>) and M2Align (<https://github.com/KhaosResearch/M2Align>)
  - Gnuplot, R, ... to plot data
  - Optionally: R and Latex
- The stuff of this tutorial is available in GitHub:
  - <https://github.com/ajnebro/omicsmaster2018tutorial>

## Short bio

- Antonio J. Nebro
  - PhD in Computer Science Engineering (1999)
  - Associate professor (University of Málaga – Spain)
  - Khaos research group (<http://khaos.uma.es/en>)
  - <http://www.lcc.uma.es/~antonio/>
- Research interests
  - Multi-objective optimization
  - Parallel metaheuristics
  - Applications to real-world problems (bioinformatics, civil engineering, Big Data)
  - Software tools (jMetal, jMetalSP, jMetalPy)



# Publications

Antonio J. Nebro - Citas de Google Scholar



**Antonio J. Nebro** 

[SEGUIR](#)

[University of Málaga](#)  
Dirección de correo verificada de lcc.uma.es  
Computer Science

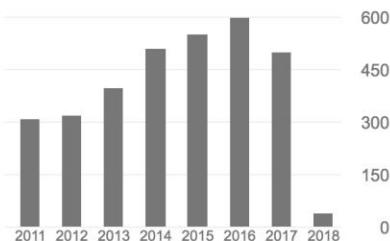
 [CITAS](#) 

TÍTULO	CITADO POR	AÑO
<a href="#">jMetal: A Java framework for multi-objective optimization</a>	672	2011
JJ Durillo, AJ Nebro Advances in Engineering Software 42 (10), 760-771		
<a href="#">The jMetal framework for multi-objective optimization: Design and architecture</a>	354	2010
JJ Durillo, AJ Nebro, E Alba Evolutionary Computation (CEC), 2010 IEEE Congress on, 1-8		
<a href="#">Smpso: A new pso-based metaheuristic for multi-objective optimization</a>	283	2009
AJ Nebro, JJ Durillo, J García-Nieto, CAC Coello, F Luna, E Alba Computational intelligence in miulti-criteria decision-making, 2009. mcdm'09 ...		
<a href="#">Mocell: A cellular genetic algorithm for multiobjective optimization</a>	270	2009
AJ Nebro, JJ Durillo, F Luna, B Dorronsoro, E Alba International Journal of Intelligent Systems 24 (7), 726-746		
<a href="#">AbYSS: Adapting scatter search to multiobjective optimization</a>	251	2008
AJ Nebro, F Luna, E Alba, B Dorronsoro, JJ Durillo, A Beham IEEE Transactions on Evolutionary Computation 12 (4), 439-457		
<a href="#">Multi-objective particle swarm optimizers: An experimental comparison</a>	130	2009
JJ Durillo, J García-Nieto, AJ Nebro, CAC Coello, F Luna, E Alba International Conference on Evolutionary Multi-Criterion Optimization, 495-509		
<a href="#">Heterogeneous computing and parallel genetic algorithms</a>	117	2002
E Alba, AJ Nebro, JM Troya Journal of Parallel and Distributed Computing 62 (9), 1362-1385		

Citado por

Citas	Total	Desde 2013
Citas	4023	2606
Índice h	32	25
Índice i10	56	45

[VER TODO](#)



A bar chart showing the number of citations per year from 2011 to 2018. The y-axis represents the number of citations, ranging from 0 to 600 in increments of 150. The x-axis lists the years from 2011 to 2018. The bars show a general upward trend with some fluctuations.

AÑO	CITAS
2011	320
2012	330
2013	450
2014	580
2015	580
2016	600
2017	550
2018	10

Coautores [EDITAR](#)

No existen coautores.

Antonio J. Nebro – Multi-objective Optimization with Metaheuristics

7 - 127

## PhD thesis supervision

- Metaheurísticas de Optimización Multiobjetivo Aplicadas a la Inferencia Filogenética y al Alineamiento Múltiple de Secuencias
  - Cristian Zambrano Vega. 2017

C. Zambrano Vega, A.J. Nebro, J. García Nieto, J.F. Aldana Montes. **M2Align: parallel multiple sequence alignment with a multi-objective metaheuristic.** Bioinformatics. Vol 33, No 19, pp: 3011-3017. October 2017.

C. Zambrano-Vega, A.J. Nebro and J.F. Aldana-Montes. **MO-Phylogenetics: a phylogenetic inference software tool with multi-objective evolutionary metaheuristics.** Methods in Ecology and Evolution. Volume 7, Issue 7 July 2016. Pages 800–805.

- Multi-objective optimization in life sciences
  - Esteban López Camacho. 2017

Esteban López-Camacho, María Jesús García Godoy, José García-Nieto, Antonio J.Nebro and José F. Aldana-Montes, **Solving molecular flexible docking problems with metaheuristics: A comparative study.** Applied Soft Computing. Volume 28, Pp. 379-393. March 2015.

Esteban López-Camacho, María Jesús García Godoy, Antonio J. Nebro and José F. Aldana-Montes, **MetalCpp: optimizing molecular docking problems with a C++ metaheuristic framework.** Bioinformatics. Volume 30, Issue 3, Pp. 437-438. 2014

# Khaos research group



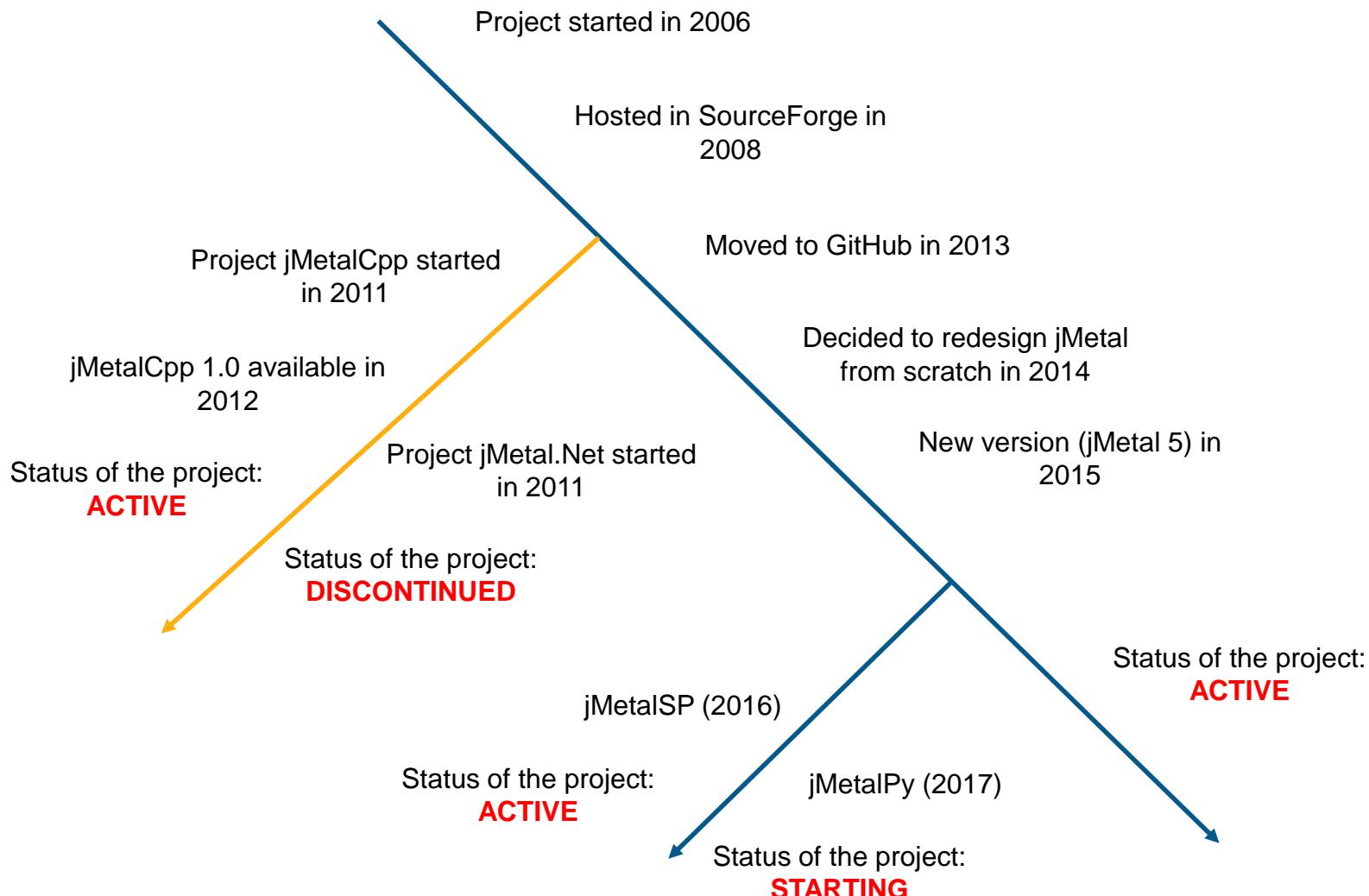
## Khaos research group

- **Research Lines:**
  - Data Management and Integration (Relational Data Bases, NoSQL, Linked Data, Open Data)
  - Data Analysis (Data Mining, Test Mining, Information Recovery, Optimization Algorithms, Big Data Analytics)
  - Semantic Annotation. Integrated Data Analysis
  - Semantics (Scalable Reasoning, Semantic Relations Discovery, Semantic Assignment to Mobile Objects, Semantic Driven Contents Recommendation, Web Semantics for E-Sciences, Smart Data for Big Data interpretation)
  - **Metaheuristics. Multi-objective Optimization. Big Data Optimization**
- **Applications (multi-disciplinary domains):**
  - **Life Sciences and Biomedicine**
  - Cultural Patrimony and Tourism
  - E-commerce
  - Smart Cities

# jMetal: A framework for multi-objective optimization

- jMetal is
  - A Java-based framework for multi-objective optimization with metaheuristics
  - Developed by computer science engineers to make research on multi-objective optimization
  - Open source project (MIT license)
- Web sites
  - Web page: <http://jmetal.github.io/jMetal/>
  - GitHub: <https://github.com/jMetal/jMetal>
  - Documentation: <https://github.com/jMetal/jMetalDocumentation>

# jMetal: A framework for multi-objective optimization



## jMetal: A Maven project hosted in GitHub

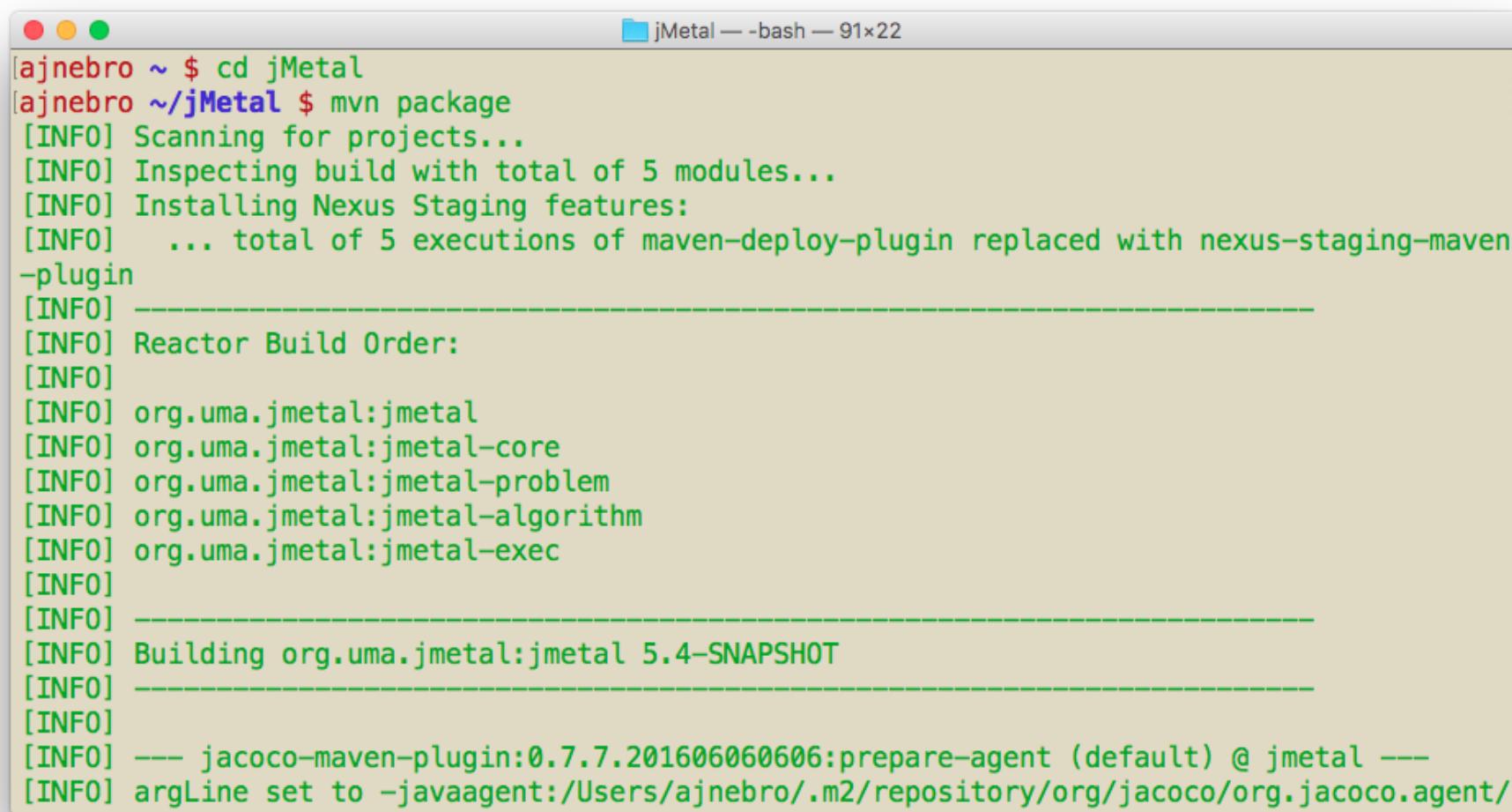
- Getting the full source code of the project



```
[ajnebro ~ $ git clone https://github.com/jMetal/jMetal.git
Cloning into 'jMetal'...
remote: Counting objects: 53691, done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 53691 (delta 0), reused 11 (delta 0), pack-reused 53677
Receiving objects: 100% (53691/53691), 109.34 MiB | 5.82 MiB/s, done.
Resolving deltas: 100% (27036/27036), done.
[ajnebro ~ $ ls jMetal/
LICENSE.txt          jmetal-exec
README.md            jmetal-problem
jmetal-algorithm    pom.xml
jmetal-core          sonar-project.properties
ajnebro ~ $ ]
```

## jMetal: A Maven project hosted in GitHub

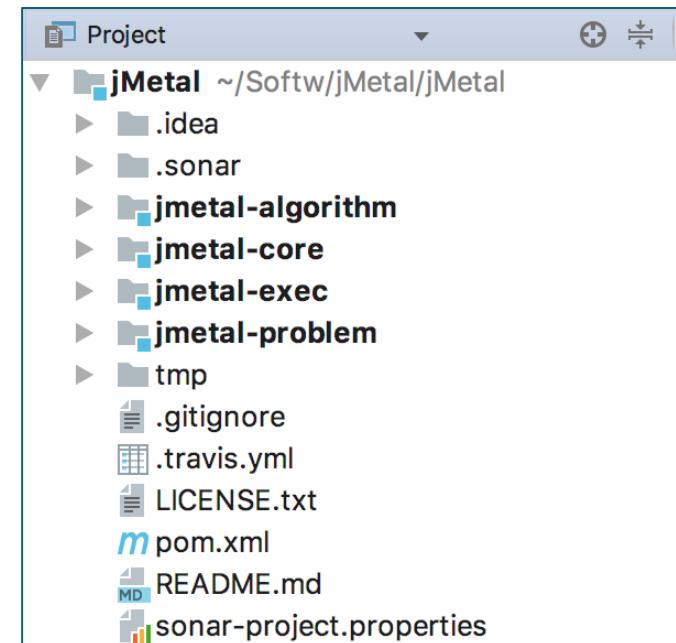
- The project can be built from the command line using Maven



```
[ajnebro ~ $ cd jMetal
[ajnebro ~/jMetal $ mvn package
[INFO] Scanning for projects...
[INFO] Inspecting build with total of 5 modules...
[INFO] Installing Nexus Staging features:
[INFO]   ... total of 5 executions of maven-deploy-plugin replaced with nexus-staging-maven
-p plugin
[INFO] -----
[INFO] Reactor Build Order:
[INFO]
[INFO] org.uma.jmetal:jmetal
[INFO] org.uma.jmetal:jmetal-core
[INFO] org.uma.jmetal:jmetal-problem
[INFO] org.uma.jmetal:jmetal-algorithm
[INFO] org.uma.jmetal:jmetal-exec
[INFO]
[INFO] -----
[INFO] Building org.uma.jmetal:jmetal 5.4-SNAPSHOT
[INFO] -----
[INFO]
[INFO] --- jacoco-maven-plugin:0.7.7.201606060606:prepare-agent (default) @ jmetal ---
[INFO] argLine set to -javaagent:/Users/ajnebro/.m2/repository/org/jacoco/org.jacoco.agent/
```

## jMetal: A Maven project hosted in GitHub

- jMetal 5 is composed of four Maven sub-packages
- In the current version (jMetal 5.4):
  - Multi-Objective algorithms (20+)
  - Benchmark problems (70+)
  - Solution representations (6)
  - Quality indicators (10)



## Table of contents

- About this tutorial
- **Metaheuristics as optimization techniques**
- Dealing with problems with more than one objective function
- Case study: multiple sequence alignment
- Further developments and related projects

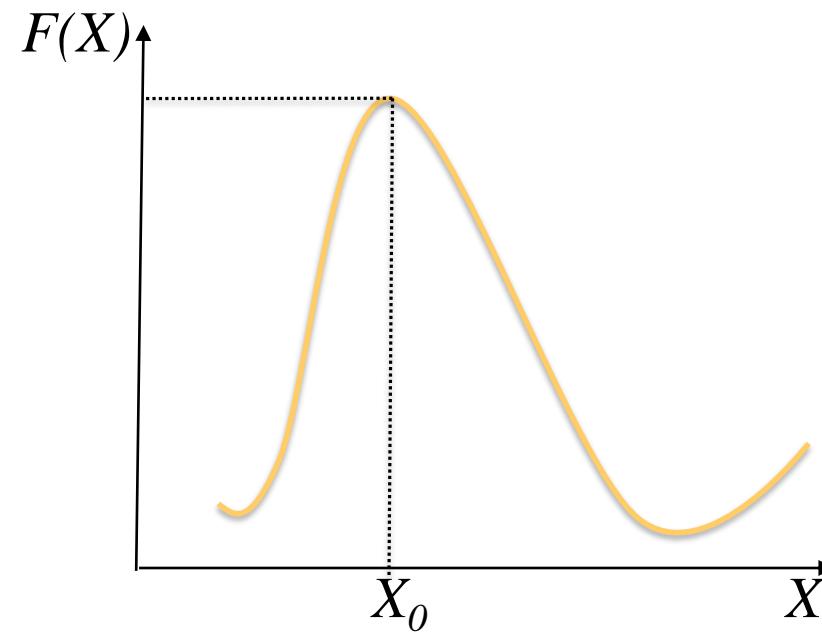
# What is Optimization?

- Optimization problem
  - Problem characterized by having one or more functions or objectives that have to be maximized or minimized
- Optimization
  - The process of finding the optimum of a problem



# What is Optimization?

- More precisely
  - Given a problem characterized by a vector of decision variables  $X$  and a function  $F(X)$ 
    - Optimization is the task of finding the value of  $X$  (solution  $X_0$ ) yielding the maximum/minimum value of  $F(X)$



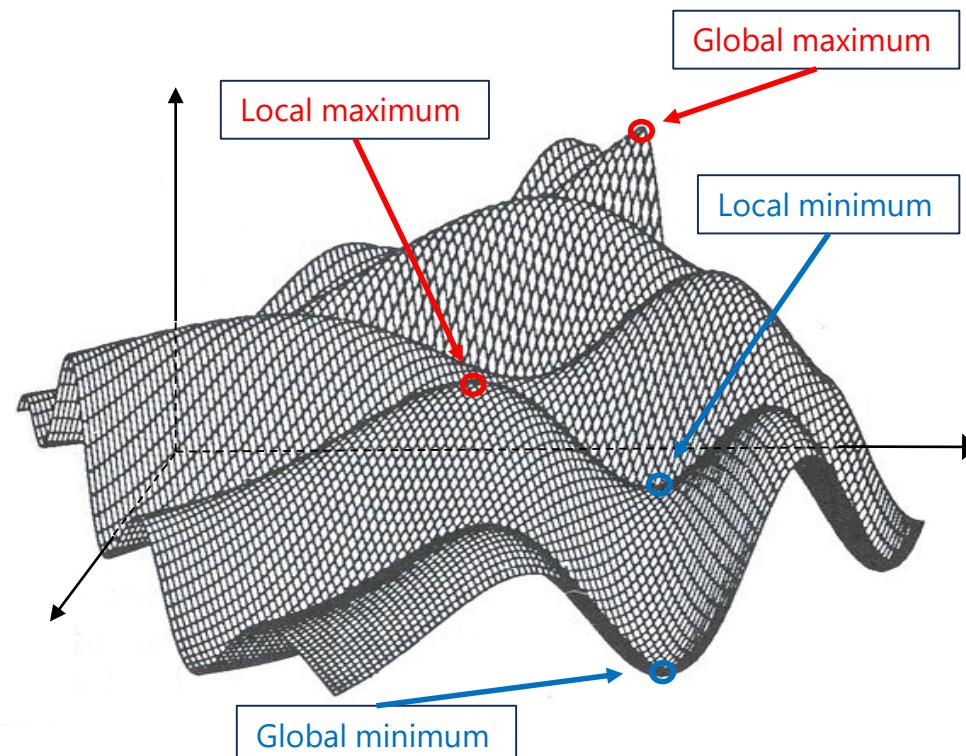
# What is Optimization?

- An **optimization problem** is a pair  $(S, f)$  where:
  - $S$  is a solution set (solution space), and
  - $f : S \rightarrow \mathbb{R}$  is an objective function.

- In the case of **minimization**, the objective is to find a solution  $i^* \in S$ , that satisfies:

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

- **$\text{Max } f(x) = \text{Min } -f(x)$**



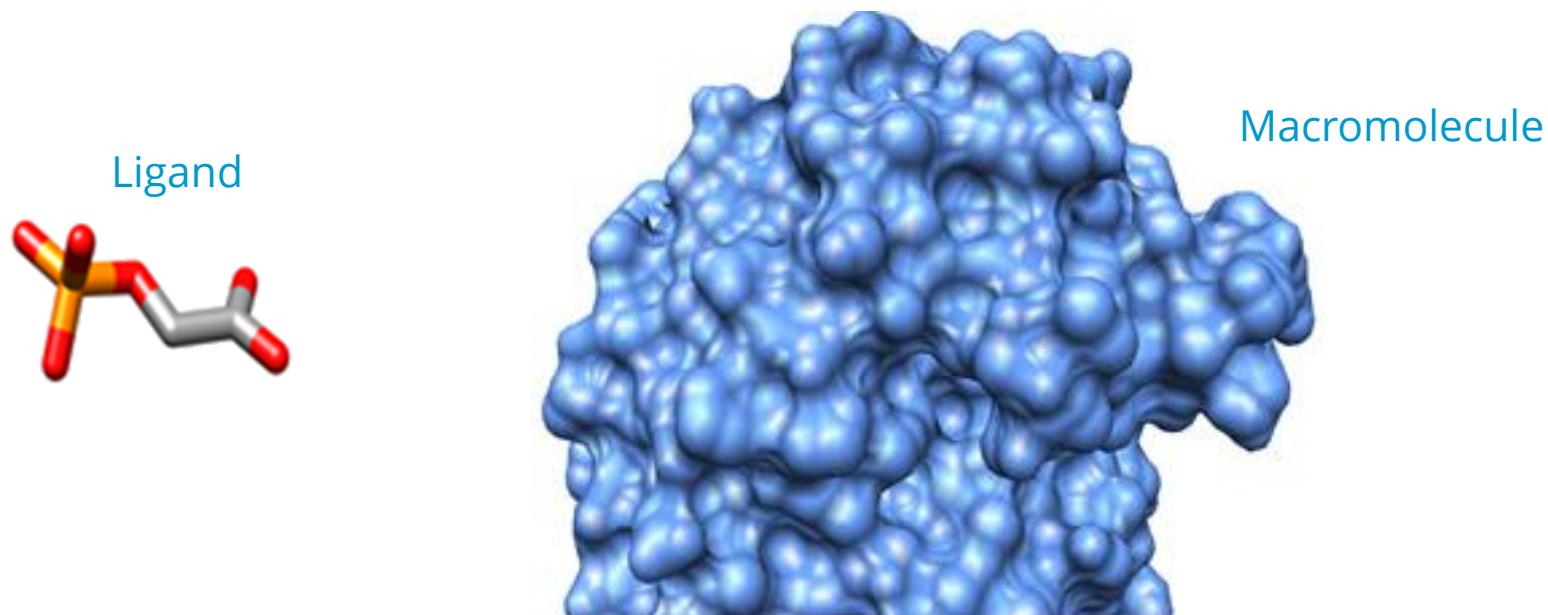
## What is Optimization?

- Finding the optimum of an optimization problem can be very difficult:
  - NP hard complexity
  - Non-linear functions
  - Multiple objectives
  - Epistasis
  - Deceptiveness
  - Noisy functions
  - Etc.

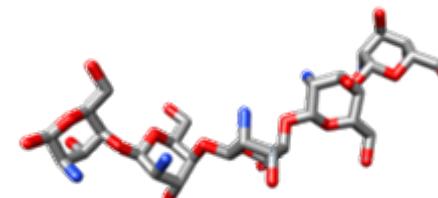
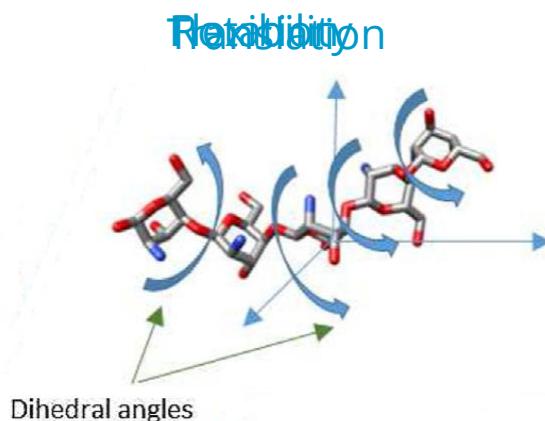
Thomas Weise, Michael Zapf, Raymond Chiong, and Antonio J. Nebro. **Why Is Optimization Difficult?**. R. Chiong (Ed.): Nature-Inspired Algorithms for Optimisation, SCI 193, pp. 1–50.

## Example of optimization problem: molecular docking

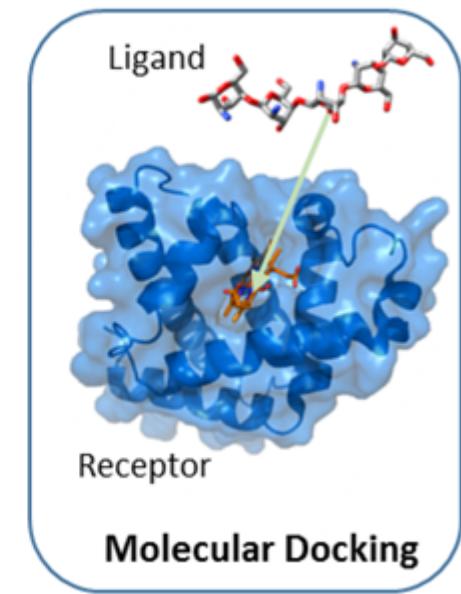
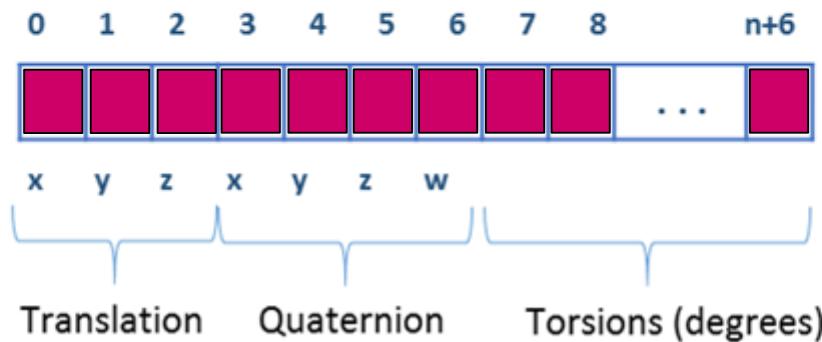
- Predicting the position of a small molecule (**ligand**) to a receptor (**macromolecule**) with the goal of finding a **minimum binding energy** conformation



## Example of optimization problem: molecular docking



**Solution encoding:**  
Real-coded variables

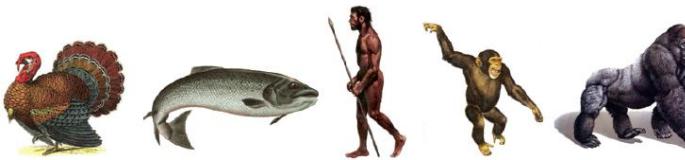


# Example of optimization problem: phylogenetic inference

## Secuencias Biológicas

5 species 42 nucleotides (DNA-based analysis)

```
AAGCTNGGCATTTCAGGGTGAGCCCGGGCAATACAGGGTAT
AAGCCTTGGCAGTGAGGGTGGCCGGGACGGTAT
ACCGGTGGCCGTACGGTACAGGTGGCCGGTACGGTAA
AAACCTTGGCGTACGGCTAACCGAGGGCCGGACACTCAT
AAACCTTGGCGGTACGCTAACCATIGCCGGTACGCTAA
```



Búsqueda y reconstrucción de un árbol genealógico que describa, lo más exactamente posible, las relaciones ancestrales entre un grupo de especies

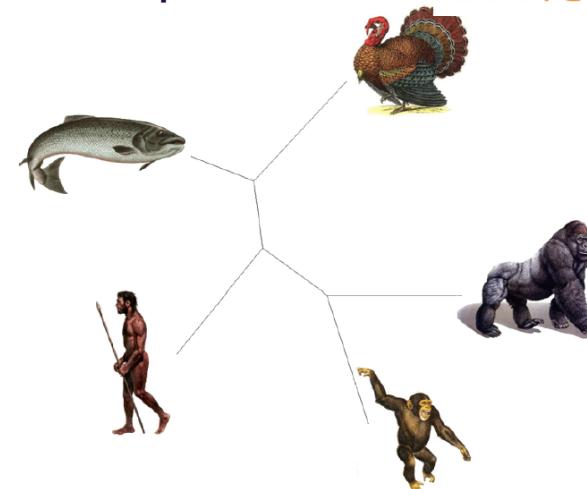
## Optimización

### Basados en Distancias

- Least Squares (LS)
- Minimum Evolution (ME)
- Neighbor Joining (NJ)

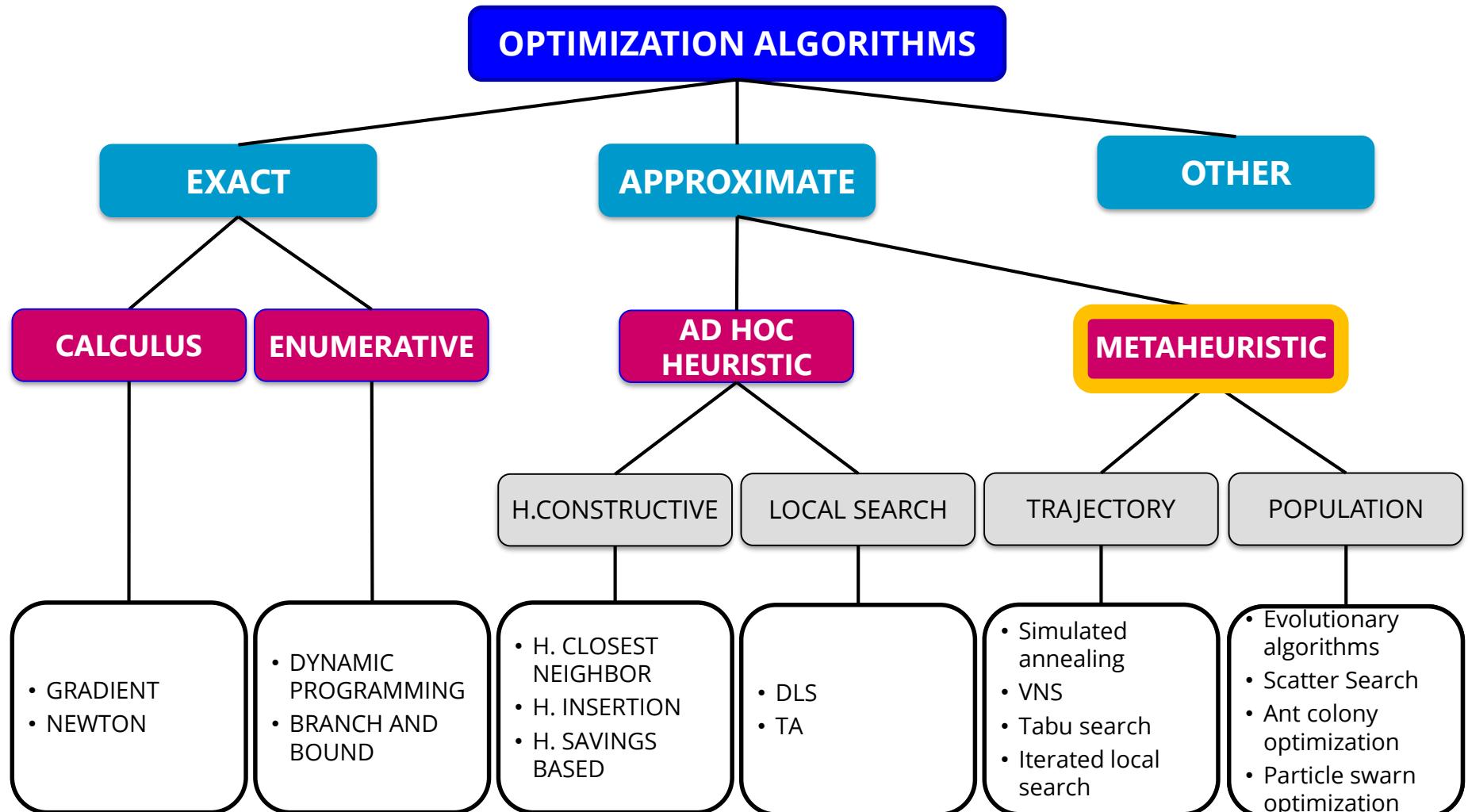
### Basados en criterios de optimización

- Máxima Parsimonia (Fitch, 1971)
- Máxima Verosimilitud (Felsenstein, 1984)



Árbol Filogenético

# Optimization techniques taxonomy



## Optimization problems in Bioinformatics

- Examples of (multi-objective) optimization problems in Bioinformatics
  - Molecular docking
  - Phylogenetic inference
  - Molecular structure prediction
  - Gene regulatory networks
  - **Multiple sequence alignment**
  - Protein structure prediction

J. Handl, DB Kell, J Knowles, **Multiobjective optimization in bioinformatics and computational biology**,  
IEEE/ACM Transactions on Computational Biology and Bioinformatics (2), 279-292. 2007

## Definition of metaheuristic

- Components (Greek terms):
  - Meta: “higher level”
  - Heuristic: “to find”, generic method to solve a problem
- Metaheuristic:
  - *High-level strategy that combines a set of underlying simpler operation techniques (usually heuristics) aimed at obtaining a more powerful procedure*
  - They are based on stochastic components
  - They have become very popular optimization techniques
    - They provide (quasi) optimal solutions in a reasonable amount of time
    - Optimality is not guaranteed
    - Can work without knowledge of the problem to be optimized

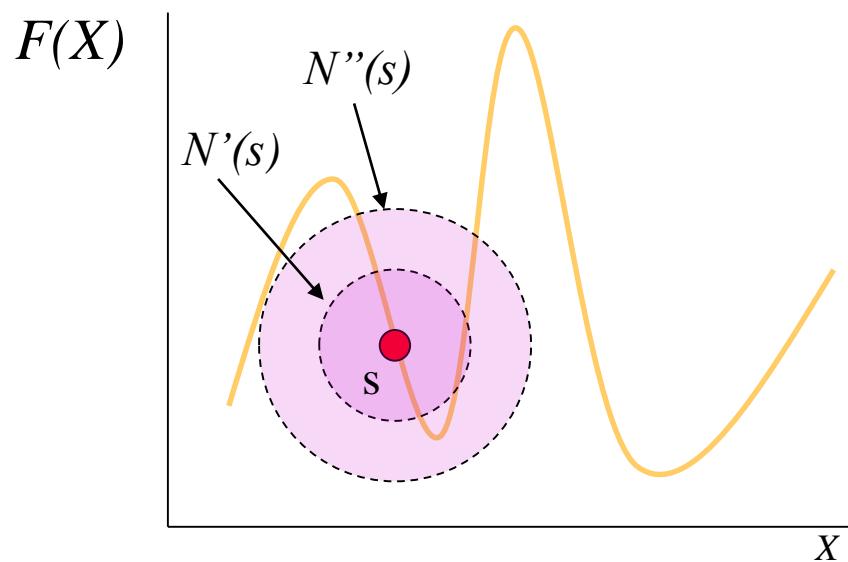
## Definition of Metaheuristic

- Example:
  - Heuristic: **Basic local search**
    - If a new solution is better than the current one, then it will become the current one
    - Drawback: it can easily be trapped in local minima
  - Metaheuristic: **Simulated Annealing (SA)**
    - Basic local search, but a worse solution have a chance to be the current one attending to a decaying probability

Let us see these ideas in detail

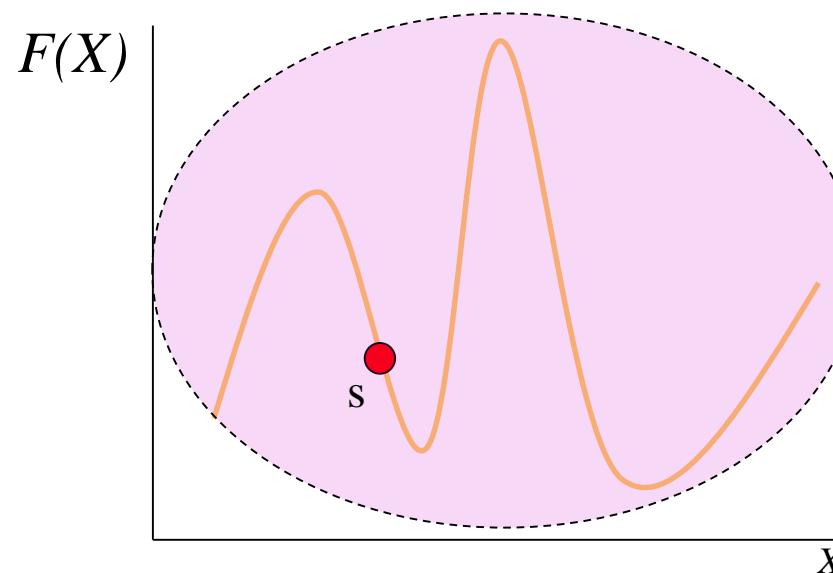
## Basic concepts: neighborhood

- Neighborhood
  - *Given a solution  $s$ , the neighborhood of  $s$ ,  $N(s)$ , is the set of solutions of the search space that can be reached using some kind of transformation on  $s$*



## Basic concepts: neighborhood

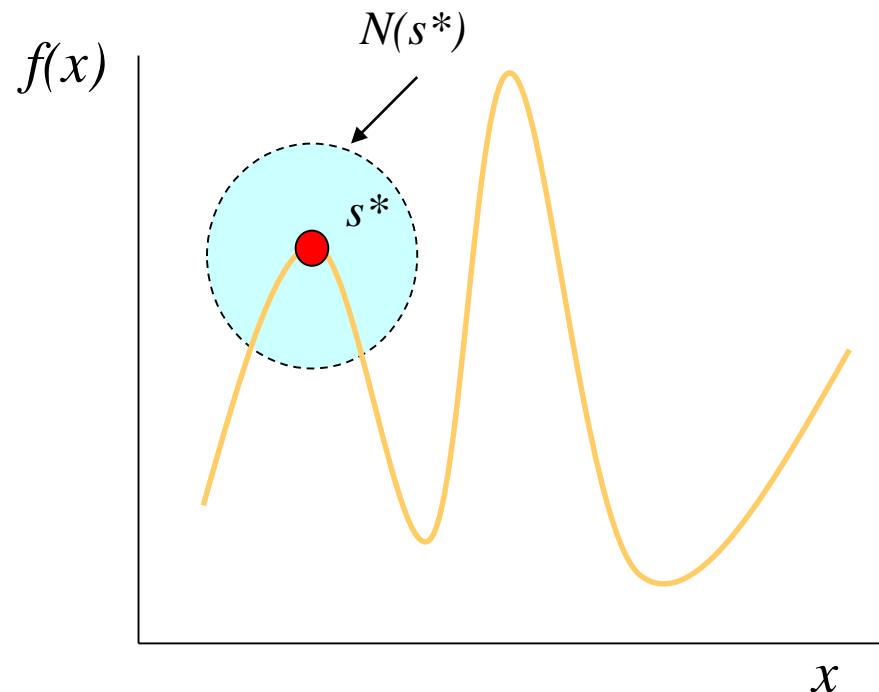
- Why the need of defining neighborhoods?
  - If we search in the whole space, we are applying a **random search**
  - We are interested in an **intelligent search**



## Basic concepts: local optimum

- Local maximum (assuming maximization)
  - A solution  $s^*$  is a local maximum with respect to a neighborhood  $N$  if:

"  $s \in N(s^*), f(s) \geq f(s^*)$



- Search techniques must avoid to get trapped in local optima

## Example of Heuristic: Basic Local Search

- Basic idea:
  - Iterative process where, given a current solution  $s$ , a new solution from  $N(s)$  is accepted if it is better than  $s$
- Features
  - Very simple strategy
  - The algorithm stops after:
    - Finding a local/global optimum
    - Performing a fixed number of iterations

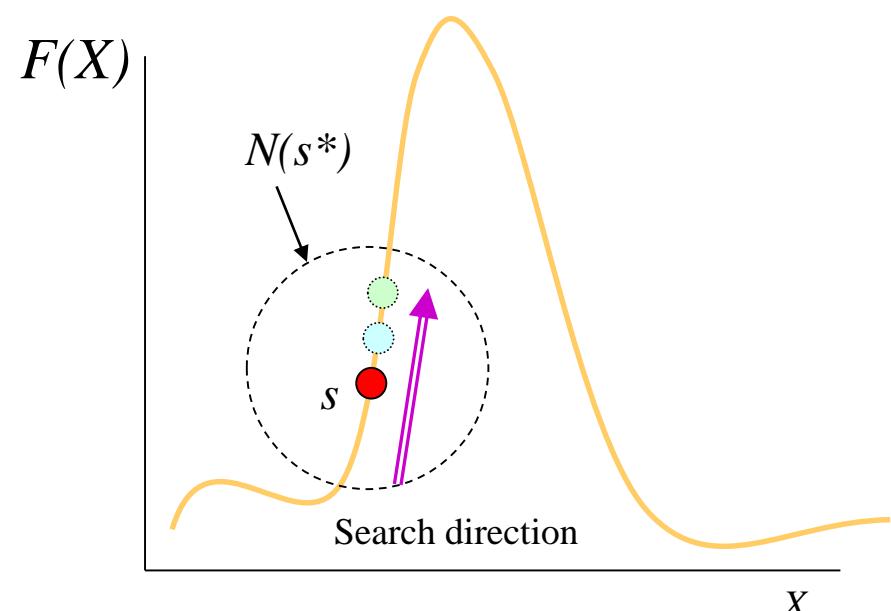
---

```

1:  $s \leftarrow \text{GenerateInitialSolution}()$ 
2: repeat
3:    $s \leftarrow \text{Improve}(N(s))$ 
4: until no improvement is possible

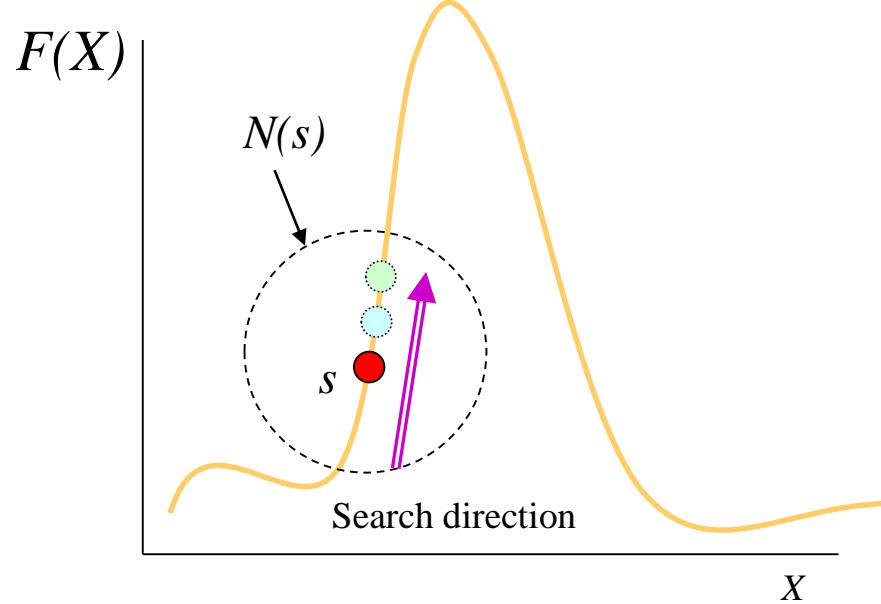
```

---

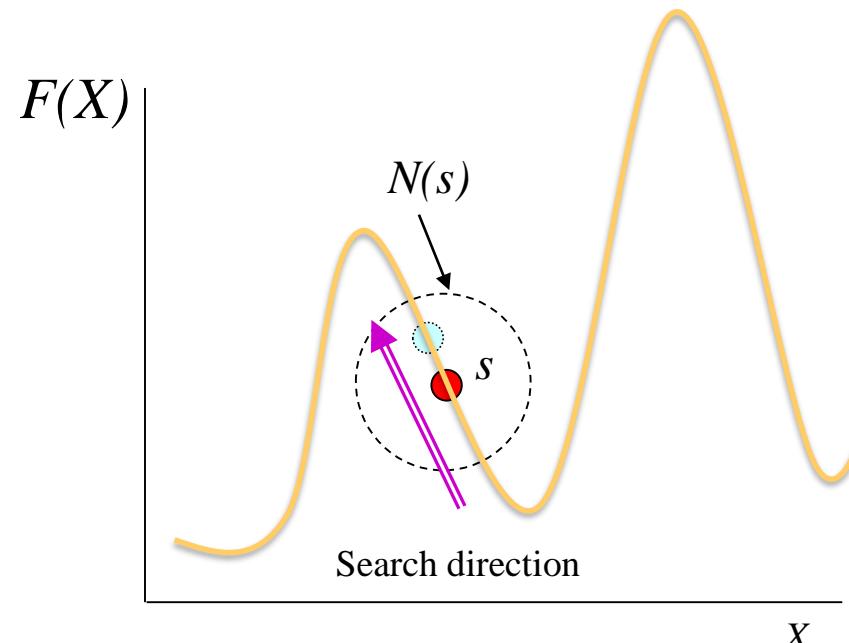


# How effective is Local Search?

- Unimodal function



- Multimodal function



- Basic local search is prone to be trapped in local optima

# How effective is Local Search?

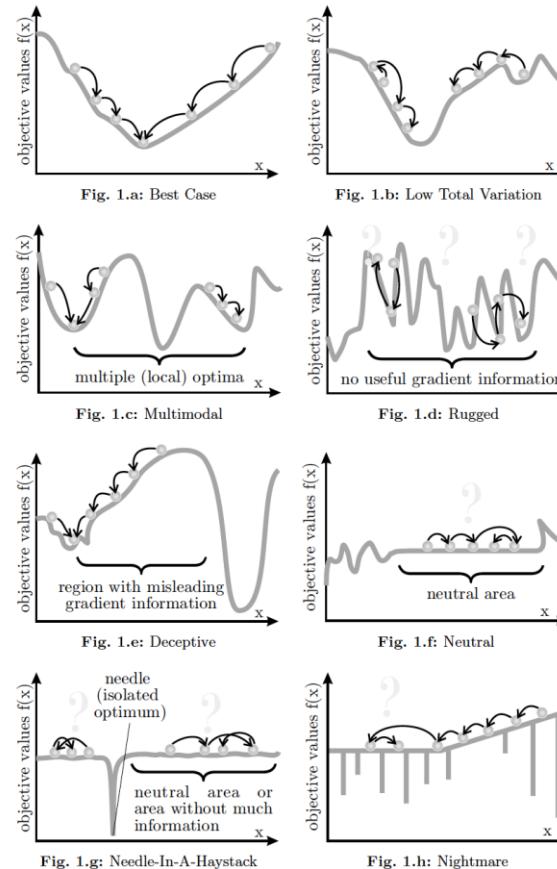


Fig. 1 Different possible properties of fitness landscapes (minimization)

Thomas Weise, Michael Zapf, Raymond Chiong, and Antonio J. Nebro. **Why Is Optimization Difficult?**. R. Chiong (Ed.): Nature-Inspired Algorithms for Optimisation, SCI 193, pp. 1–50.

# How effective is Local Search?

- Running a local search in jMetal to solve the OneMax problem

## OneMax: Maximizing the number of 1's in a binary string

## Example of Metaheuristic: Simulated Annealing

- Basic idea
  - To choose solutions which are **worse** than the current one in order to try to escape from local optima

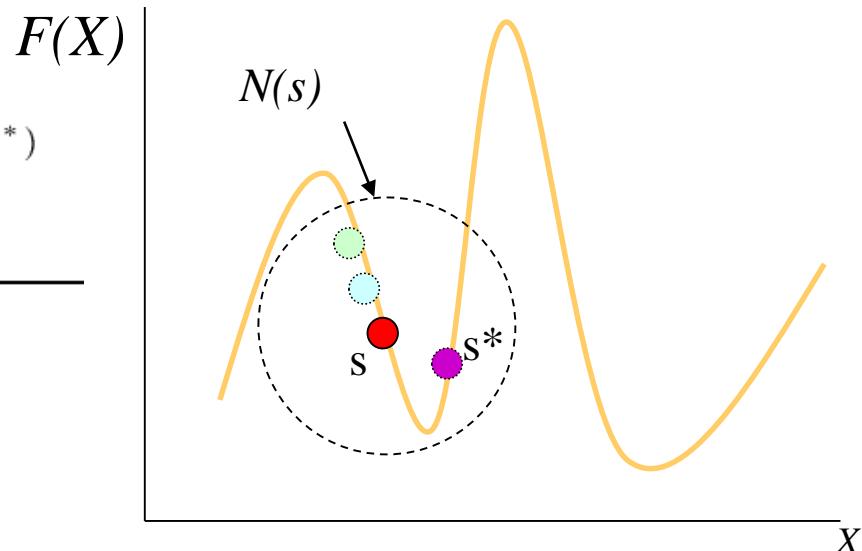
---

```

1:  $s \leftarrow \text{GenerateInitialSolution}()$ 
2:  $T \leftarrow T_0$ 
3: while termination conditions not met do
4:    $s^* \leftarrow \text{PickAtRandom}(N(s))$ 
5:   if  $f(s^*) < f(s)$  then
6:      $s \leftarrow s^*$ 
7:   else
8:     Accept  $s^*$  as new solution with probability  $p(T, s, s^*)$ 
9:   end if
10:   $\text{Update}(T)$ 
11: end while

```

---

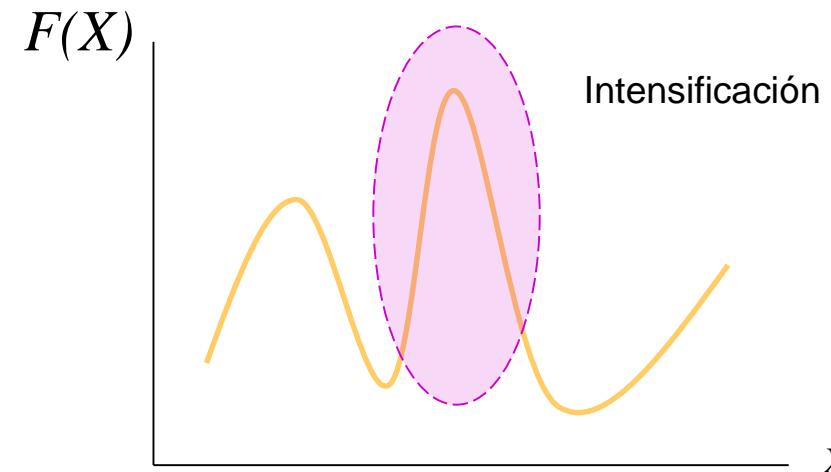
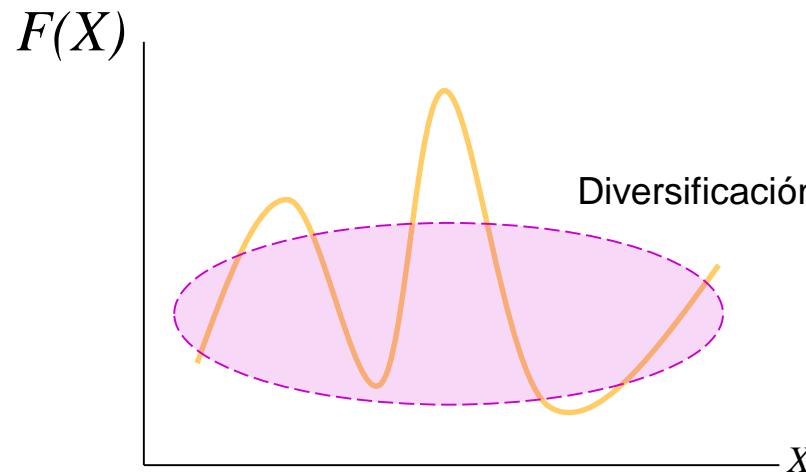


## Example of Metaheuristic: Simulated Annealing

- Simulated annealing (SA)
  - Is considered the first metaheuristic
  - The technique is inspired from annealing in metallurgy
- Features
  - The probability of accepting worse solutions decreases during the search (regulated by the temperature  $T$ )
- Key factors (parameters to set)
  - The initial temperature  $T$
  - The length of the Markov chain (to decrement  $T$ )
  - The acceptance probability (depends on  $T$ )

## Basic concepts: Diversification vs Intensification

- A metaheuristic must achieve a balance between **diversification** and **intensification**
  - Diversification: exploration of the search space
  - Intensification: exploitation of promising regions of the search space



## Classification of Metaheuristics

- Criteria: Nature inspired vs. non-nature inspired
  - Nature inspired
    - Evolutionary Algorithm (EA)
    - Ant Colony Optimization (ACO)
    - Particle Swarm Optimization (PSO)
    - Simulated Annealing (SA)
    - Artificial Immune Systems
  - Non-nature inspired
    - Scatter Search (SS)
    - Tabu Search (TS)
    - Variable Neighborhood Search (VNS)

## Classification of Metaheuristics

- Criteria: Trajectory-based vs. population-based
  - Trajectory-based
    - Iterated Local Search (ILS)
    - Guided Local Search (GLS)
    - Variable Neighborhood Search (VNS)
    - Others: SA, TS
  - Population-based
    - Ant Colony Optimization (ACO),
    - Evolutionary Algorithm (EA),
    - Particle Swarm Optimization (PSO),
    - Scatter Search

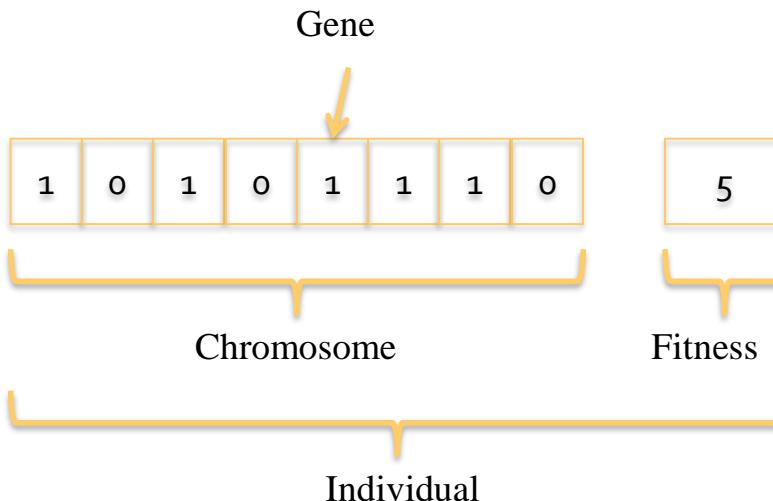
## Evolutionary Algorithms

- Evolutionary algorithms (EAs)
  - Are the most widely known and used subfamily of metaheuristics
- Inspiration
  - Emulating nature's capability of living beings to evolve and to adapt to their environment
- Key ideas
  - Solutions are known as **individuals**
  - There exist operators for the selection and reproduction of individuals
  - The best individuals have more chance to survive

## Example: OneMax problem

- Maximizing the number of 1's in a binary string
- Concepts

$$\begin{aligned} \max F(\vec{x}) &= \sum_{i=1}^N x_i \\ \vec{x} &= (x_1, x_2, \dots, x_N) \\ x_i &\in \{0,1\} \end{aligned}$$



Population

<table border="1"> <tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td></tr> </table>	1	0	1	0	1	1	1	0	5
1	0	1	0	1	1	1	0		
<table border="1"> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td></tr> </table>	0	0	1	1	0	1	0	0	3
0	0	1	1	0	1	0	0		
<table border="1"> <tr><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> </table>	1	1	1	0	1	1	1	1	7
1	1	1	0	1	1	1	1		
<table border="1"> <tr><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr> </table>	0	0	1	0	0	1	0	0	2
0	0	1	0	0	1	0	0		

# Evolutionary Algorithms

- Solution encodings

- Binary:

1	0	1	0	1	1	1	0
---	---	---	---	---	---	---	---

- Real:

1.5	2.6	-1.1	3.2
-----	-----	------	-----

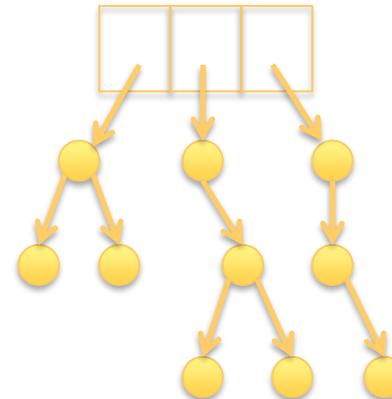
- Integer:

3	1	5	8	3	6	3	2
---	---	---	---	---	---	---	---

- Permutation:

4	6	1	3	2	5	8	7
---	---	---	---	---	---	---	---

- Tree



## Evolutionary Algorithms

- Generic pseudo-code:

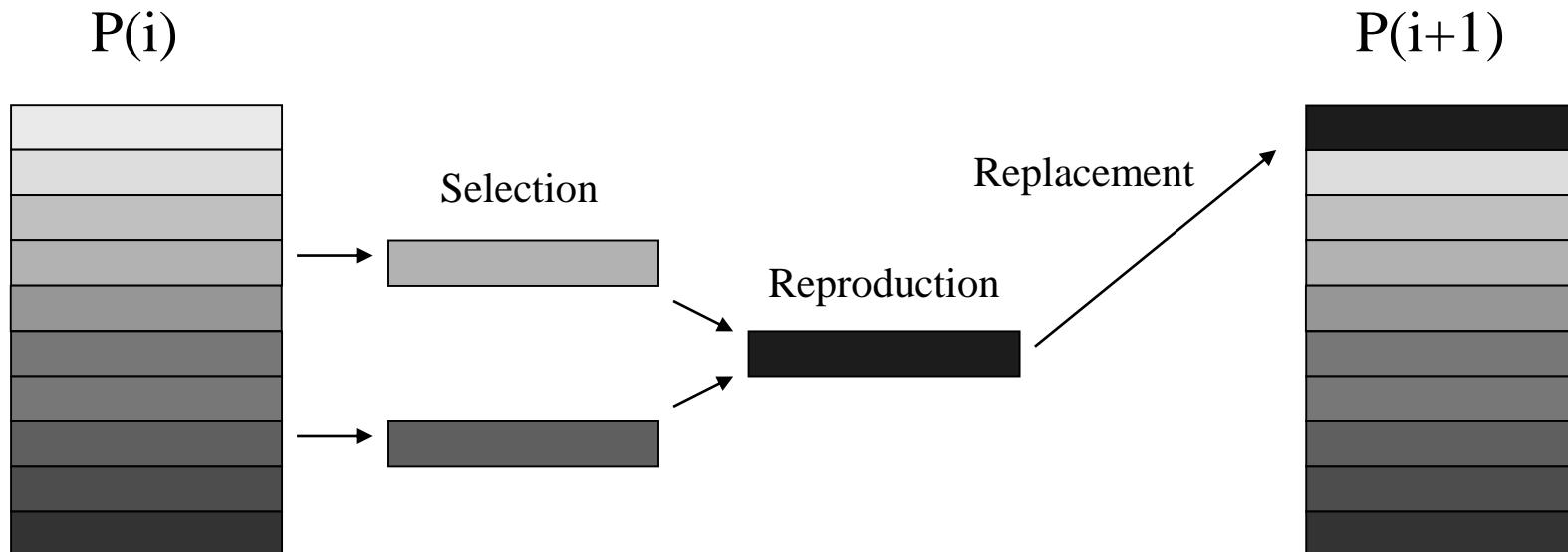
---

```
1:  $P(0) \leftarrow \text{GenerateInitialPopulation}()$ 
2:  $t \leftarrow 0$ 
3: while termination conditions not met do
4:   Evaluate( $P(t)$ )
5:    $P'(t) \leftarrow \text{Select}(P(t))$ 
6:    $P''(t) \leftarrow \text{ApplyReproductionOperators}(P'(t))$ 
7:    $P(t + 1) \leftarrow \text{Replace}(P(t), P''(t))$ 
8:    $t \leftarrow t + 1$ 
9: end while
```

---

## Evolutionary Algorithms

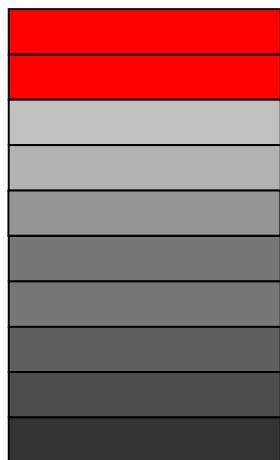
- Generic functioning scheme (Steady-State)



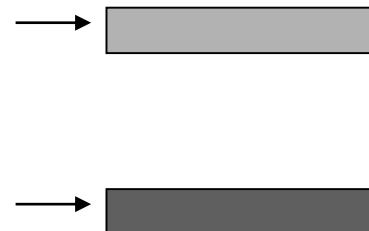
# Evolutionary Algorithms

- Generic functioning scheme (Generational)

$P(i)$



Selection



Reproduction



$P(i+1)$

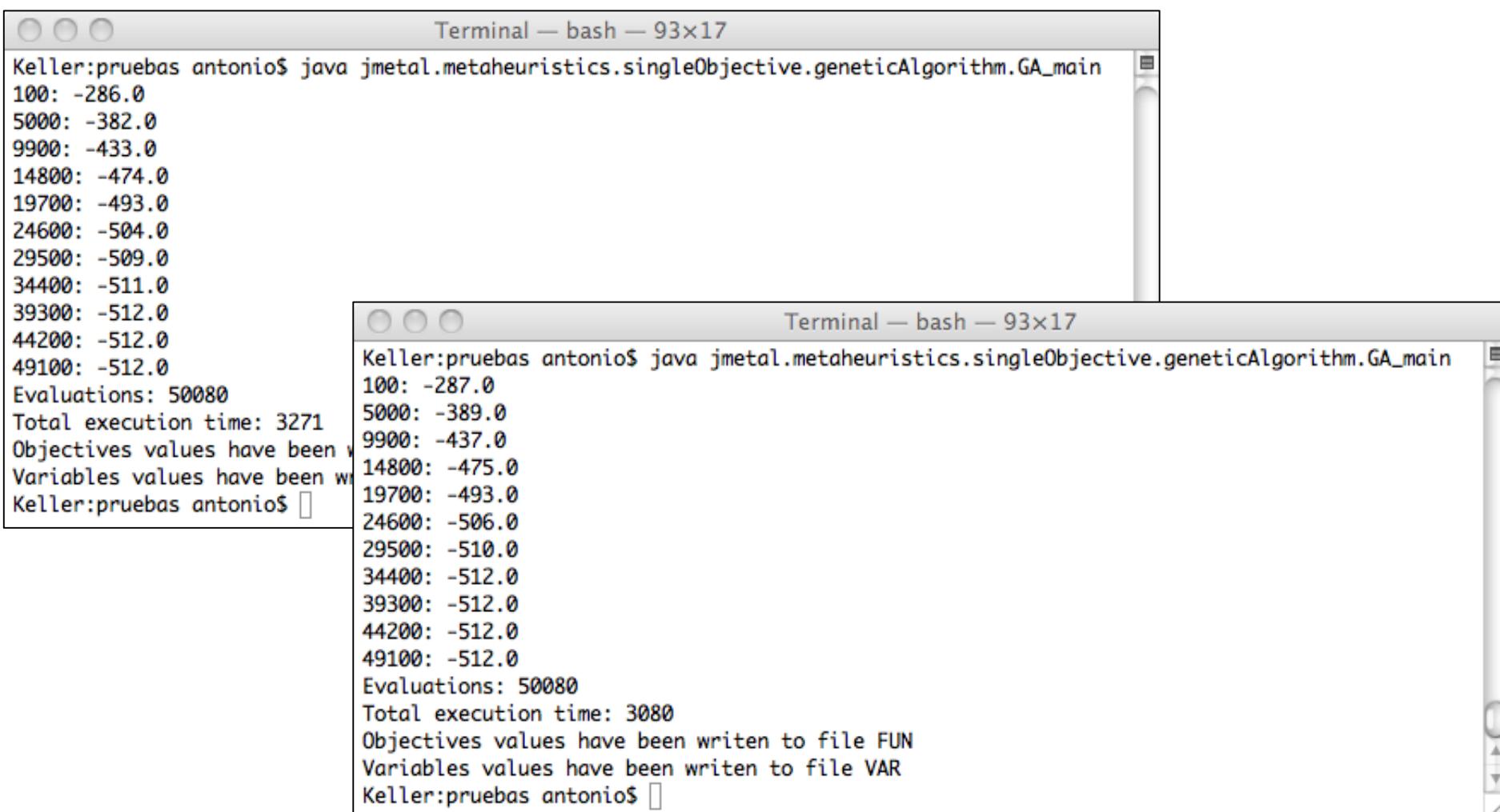


Replacement  
 $P(i) + \text{Aux}$

Aux. population

*Elitism: the  $N$  best solutions of  $P(i)$  are included in  $P(i+1)$*

## Example: OneMax problem



The image shows two terminal windows side-by-side, both titled "Terminal — bash — 93x17".

**Left Terminal Window:**

```
Keller:pruebas antonio$ java jmetal.metaheuristics.singleObjective.geneticAlgorithm.GA_main
100: -286.0
5000: -382.0
9900: -433.0
14800: -474.0
19700: -493.0
24600: -504.0
29500: -509.0
34400: -511.0
39300: -512.0
44200: -512.0
49100: -512.0
Evaluations: 50080
Total execution time: 3271
Objectives values have been written to file FUN
Variables values have been written to file VAR
Keller:pruebas antonio$
```

**Right Terminal Window:**

```
Keller:pruebas antonio$ java jmetal.metaheuristics.singleObjective.geneticAlgorithm.GA_main
100: -287.0
5000: -389.0
9900: -437.0
14800: -475.0
19700: -493.0
24600: -506.0
29500: -510.0
34400: -512.0
39300: -512.0
44200: -512.0
49100: -512.0
Evaluations: 50080
Total execution time: 3080
Objectives values have been written to file FUN
Variables values have been written to file VAR
Keller:pruebas antonio$
```

Maximizing  $F(X)$  = Minimizing  $-F(X)$

## Variants of EAs

- Genetic Algorithm (GA)
  - Binary representation (although other representations are also used)
  - Selection, crossover, and mutation operators
- Evolution Strategy (ES)
  - Floating point representation and no crossover
- Genetic Programming (GP)
  - Similar to GAs, but it is used to evolve computer programs
  - Tree representation
- Estimation of Distribution Algorithm (EDA)
  - Probabilistic models to estimate a distribution over the search space are used to evolve the population

## Genetic Algorithms (GAs)

- EAs using crossover (recombination) and mutation operators

---

```
1:  $P(0) \leftarrow \text{GenerateInitialPopulation}()$ 
2:  $t \leftarrow 0$ 
3: while termination conditions not met do
4:   Evaluate( $P(t)$ )
5:    $P_{parents} \leftarrow \text{SelectParents}(P(t))$ 
6:    $P_{offspring} \leftarrow \text{Crossover}(P_{parents})$ 
7:    $P_{offspring} \leftarrow \text{Mutation}(P_{offspring})$ 
8:    $P(t+1) \leftarrow \text{Replace}(P(t), P_{offspring})$ 
9:    $t \leftarrow t + 1$ 
10: end while
```

---

## GAs: Selection Operators

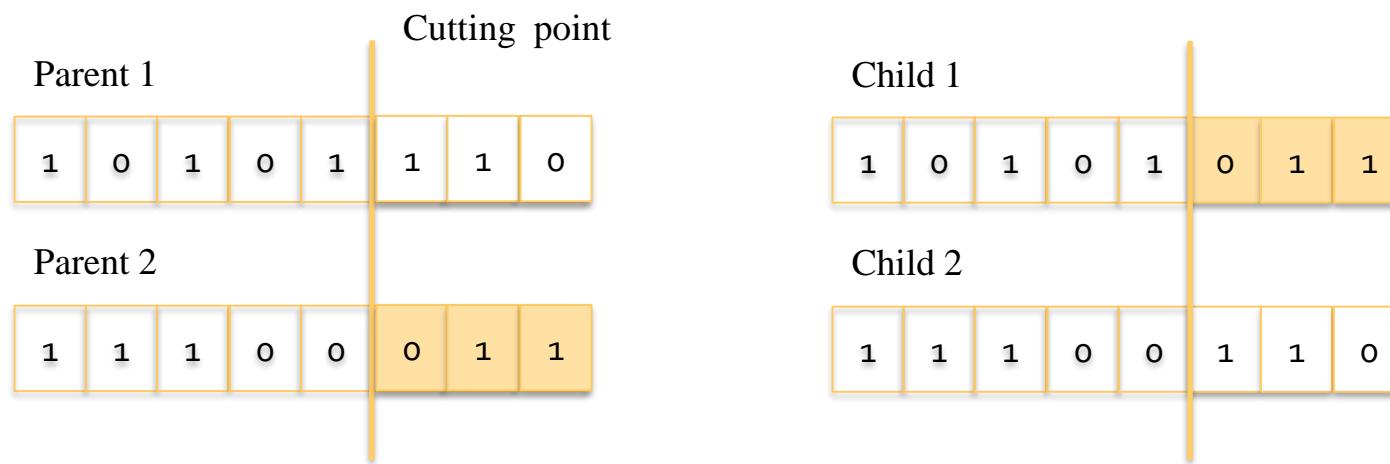
- The objective of a selection operator is to give a higher survival probability to those individuals having better fitness
- Some operators:
  - **Roulette wheel**: individuals are selected according to a probability distribution which considers their fitness values
  - **Tournament**: A set of  $m$  individuals are randomly chosen, with the same probability, and the best among them is selected
    - E.g. Binary tournament ( $m = 2$ )
  - **Random**: An individual is randomly selected

## GAs: Crossover Operators

- One or more new individuals (children) are created from a set of parents (usually from two parents) with a given probability
  - The objective is that the children combine good features of their parents (building blocks)
- Many operators depending on the representation
  - **Binary**: one point (SPX), two points (DPX), uniform, ...
  - **Real**: simulated binary (SBX), BLX- $\alpha$ , linear, ...
  - **Permutations**: PMX, OX, EX, ...

## GAs: Crossover Operators

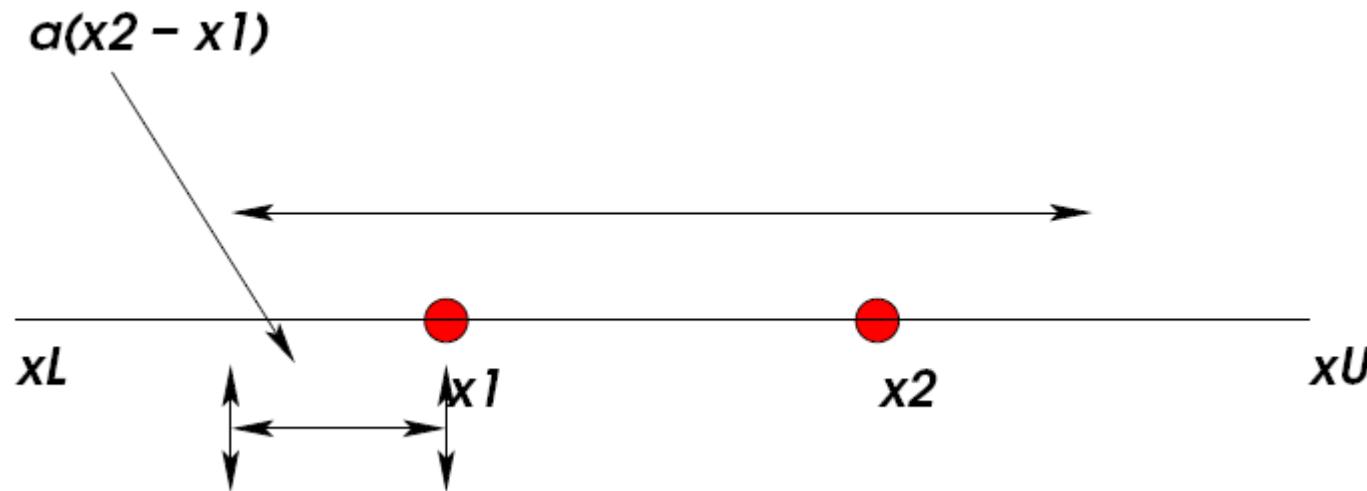
- Example: Single point crossover



## GAs: Crossover Operators

- Example: BLX- $\alpha$ 
  - Depending on the value  $\alpha$ , a value is randomly chosen from the interval:

$$[x_i - \alpha(x_j - x_i), x_j + \alpha(x_j - x_i)]$$

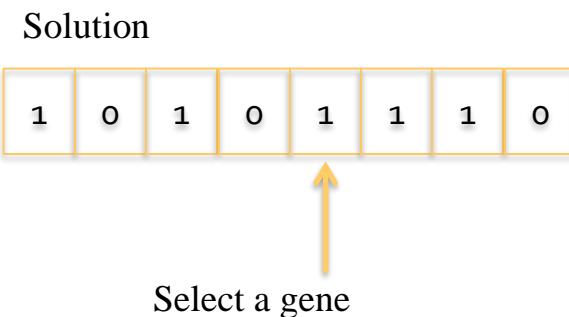


## GAs: Mutation Operators

- Mutation: A new individual is created after applying some kind of perturbation to an existing one with a given probability
  - Idea: to allow the algorithm to avoid local minima by preventing the population of solutions from becoming too similar to each other
- Many operators exist depending on the representation
  - **Binary**: bit-flip, random, ...
  - **Real**: uniform, non-uniform, polynomial, ...
  - **Permutation**: swap, exchange, ...

## GAs: Mutation Operators

- Example: bit-flip mutation

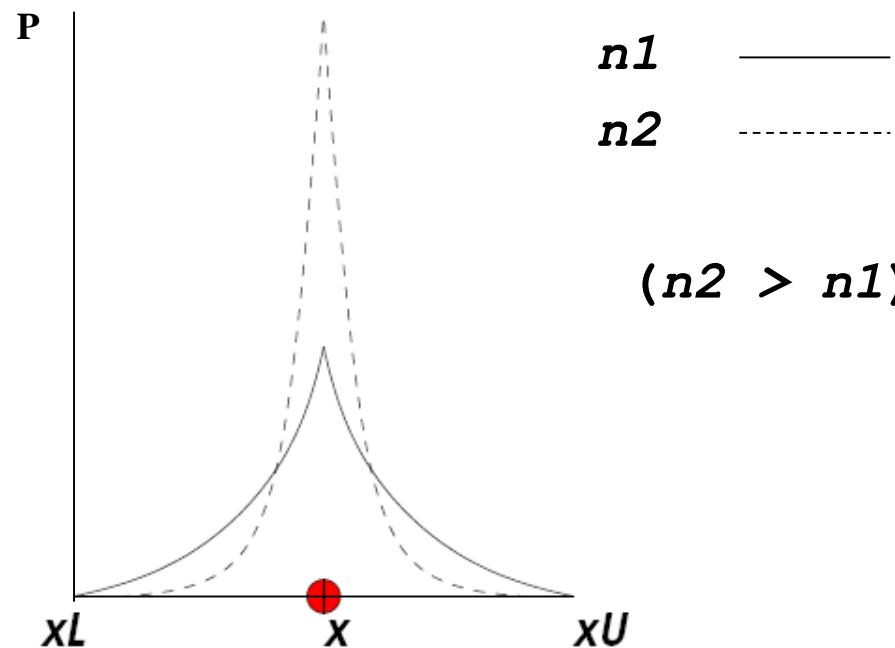


Mutated solution

1	0	1	0	0	1	1	0
---	---	---	---	---	---	---	---

## GAs: Mutation Operators

- Example: polynomial mutation
  - A new individual is obtained around  $x$  with a given probability  $P$ , which depends on the distribution index  $n$

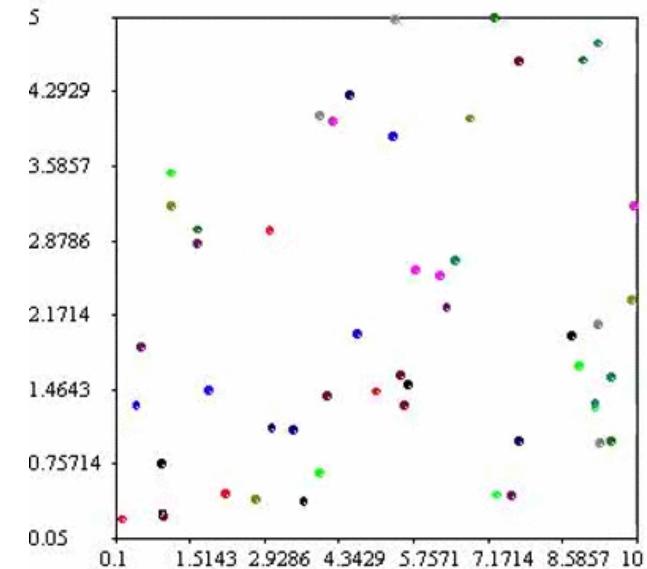


## GAs: Parameter settings

- Parameters to set in a GA:
  - Population size
  - Stopping condition (e.g, maximum number of evaluations)
  - Crossover operator
  - Mutation operator
  - Selection operator

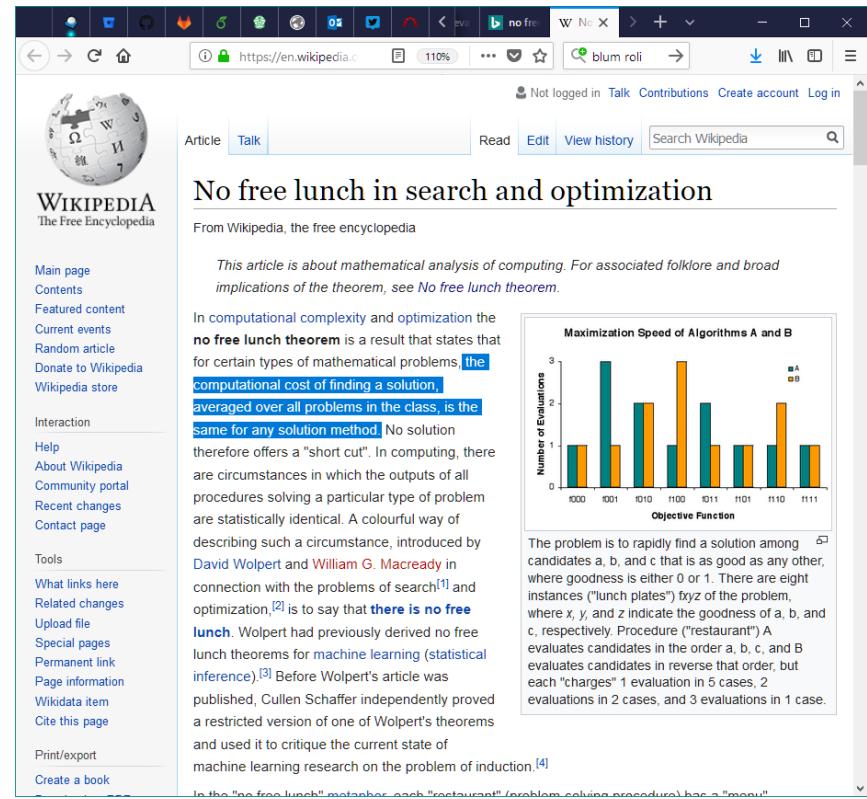
# Particle Swarm Optimization (PSO)

- Features:
  - Solutions are called **particles**.
  - Variables are the particles **positions** in a n-dimensional space.
  - The **movement** and **speed** of the particles depend of the other particles' positions



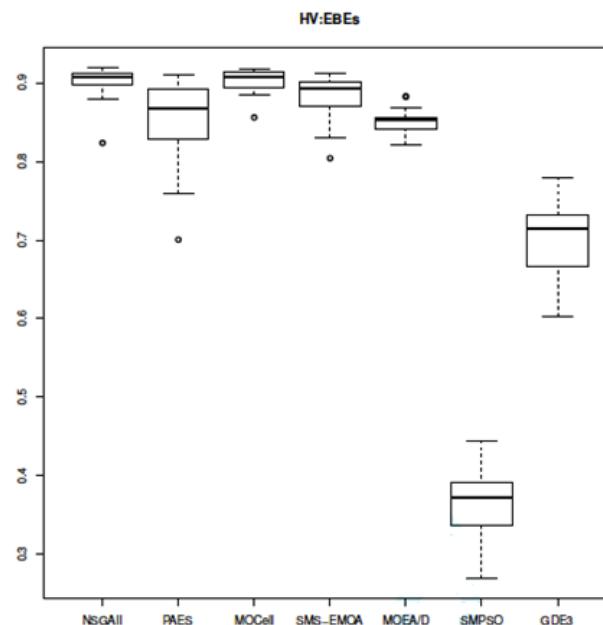
# About choosing the proper metaheuristic

- Given a problem, finding the best algorithm to solve it is not trivial
  - No Free Lunch theorem:
    - there is no one algorithm (heuristic/metaheuristic) that works best for every problem



# About choosing the proper metaheuristic

- Experience helps, but every problem is different:



Hypervolume (HV)			Epsilon ( $I_{\epsilon+}$ )		
Algorithm	$FriRank$	$Holm_{Ap}$	Algorithm	$FriRank$	$Holm_{Ap}$
*SMPSO	<b>1.02</b>	-	*SMPSO	<b>1.09</b>	-
MOEA/D	2.68	2.24e-03	MOEA/D	2.00	9.87e-02
GDE3	3.09	1.45e-04	GDE3	3.09	2.79e-04
NSGA-II	3.22	5.21e-05	NSGA-II	3.81	7.25e-07

Problem: Molecular Docking Optimization

Domain: Bioinformatics

Quality indicators: Hypervolume, Epsilon

## About choosing the proper metaheuristic

- What to do when you have to solve a new problem?
- Questions:
  - Is the problem continuous or combinatorial?
    - If combinatorial DE and PSO are discarded
  - Is a constrained problem?
- Some hints:
  - Start with a simple algorithm (or one you are familiarized with)
    - Random search -> sanity check
    - ES (1+1) Evolution Strategy -> requires only mutation
    - Genetic algorithmr
  - Once you have some reference solutions
    - Pilot tests with promising algorithms to adjusts their settings
    - Perform an empirical study

## About choosing the proper metaheuristic

- Empirical studies
  - Idea: compare a number of algorithms over the problem/s to optimize
- Methodology
  - Select the metaheuristics to compare
  - Perform a number of independent runs per configuration (> 30)
  - Apply quality indicators for convergence and diversity (in multi-objective problems)
  - Report statistical data:
    - Mean/median + standard deviation/interquartile range
    - Statistical tests about the significance of the results among the algorithms: t-test, Wilcoxon, Kolmogorov Smirnoff, etc.
  - Analysis of the results

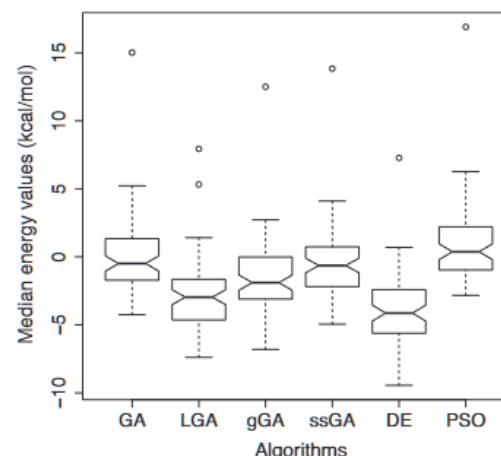
# About choosing the proper metaheuristic

- Empirical studies: an example

**Table 7**

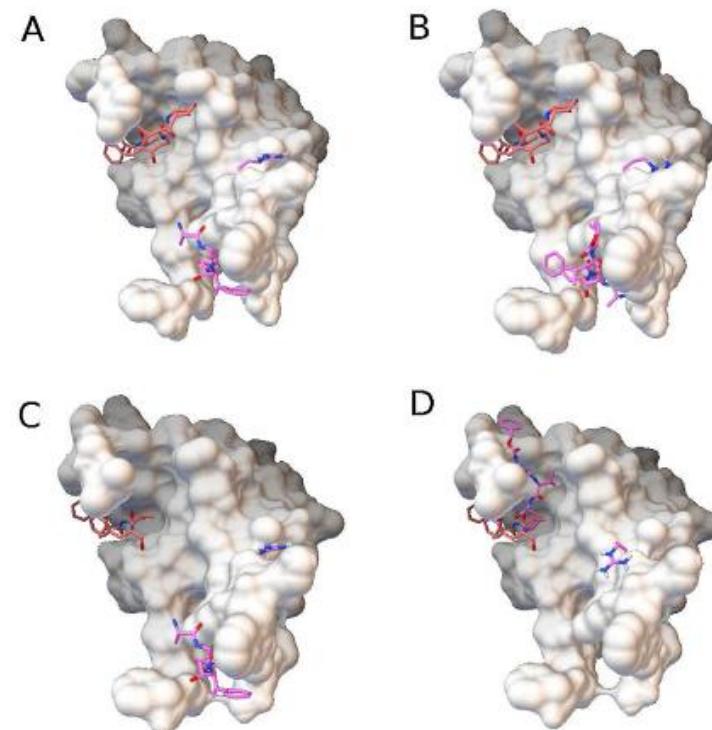
Average Friedman's rankings with Holm's Adjusted  $p$ -values ( $\alpha=0.05$ ) of compared algorithms (gGA, ssGA, DE, and PSO from jMetal; GA and LGA from AutoDock) for the test set of 75 docking instances. Symbol \* indicates the Control algorithm.

Algorithm	Friedman's rank	Holm's adjusted $p$ -value
*DE	1.10	—
LGA	2.02	2.45E - 03
gGA	3.05	4.25E - 11
ssGA	4.07	1.31E - 25
GA	4.78	1.19E - 37
PSO	5.94	7.20E - 62



**Fig. 3.** Boxplots of median binding energy (kcal/mol) distributions of the six compared algorithms, and for the 75 molecular docking instances. It is clearly observable that DE obtains the best distribution of results, for the worked instances.

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. Volume 28, Pp. 379-393. March 2015



**Fig. 5.** Best energy conformations for LGA (left molecules) and DE (right molecules) algorithms in the 1HEG (molecules at the top) and 3TLH (molecules at the bottom) instances.

## Table of contents

- About this tutorial
- Metaheuristics as optimization techniques
- **Dealing with problems with more than one objective function**
- Case study: multiple sequence alignment
- Further developments and related projects

## Multi-objective optimization

- Many real-world optimization problems require to optimize **more than one** objective or function at the same time
  - These objectives are usually in conflict among them
  - Improving one means worsening the others



- Multi-objective (or multi-criteria) optimization
  - Discipline focused in solving multiobjective optimization problems (MOPs)

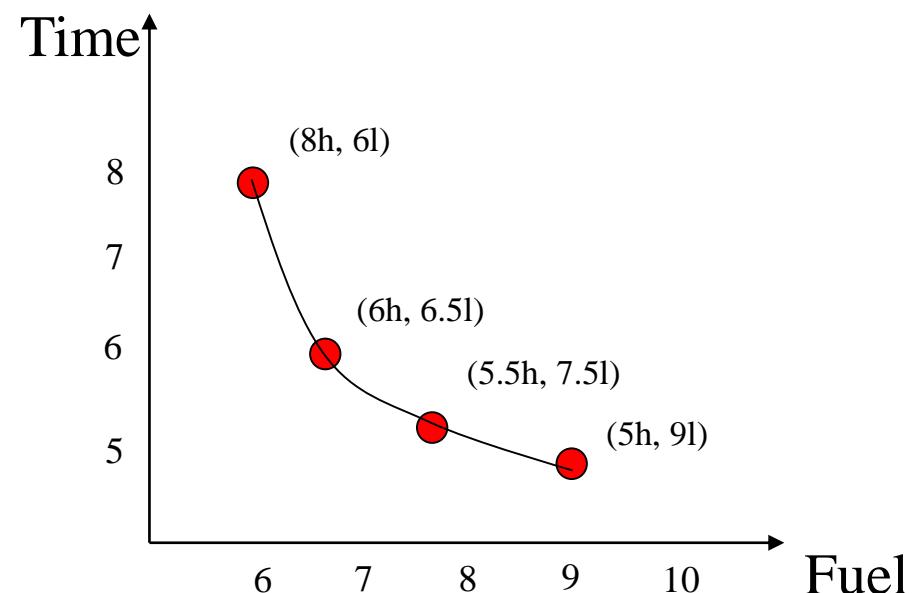
## An example

- Example: travelling by car from Málaga to Madrid (535 km)
  - Objective 1:
    - Minimizing time
  - Objective 2:
    - Minimizing fuel
  - Constraints:
    - Max. speed: 120 km/h
    - Min. speed: 60 km/h
  - Decision variable:
    - mean car speed



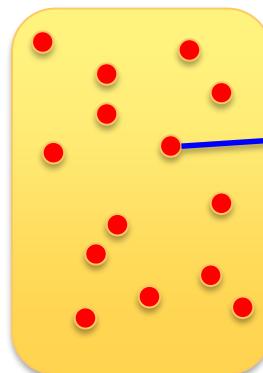
## An example

- Example: travelling by car from Málaga to Madrid (535 km)
  - Extreme solutions
    - Time: 5 hours, fuel: 9.0 litres
    - Time: 8 hours, fuel: 6.0 litres
  - Other solutions
    - Time: 5.5 hours, fuel: 7.5 litres
    - Time: 6 hours, fuel: 6.5 litres

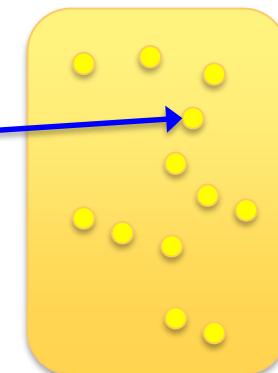


## Consequences

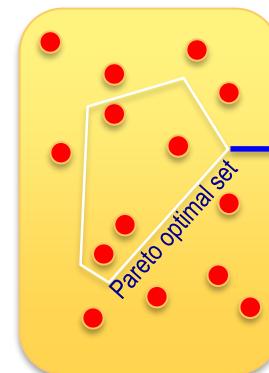
- In single-objective optimization (SO)
  - The optimum is a solution
- In multi-objective optimization (MO)
  - The optimum (**Pareto optimal set**) is a set of (**non-dominated**) solutions



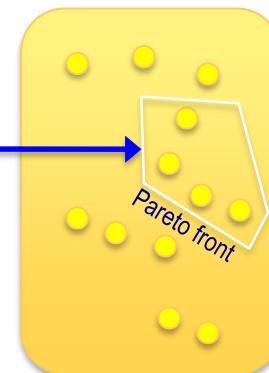
$X$   
(Solution space)



$F(X)$   
(Objective space)



$X$   
(Solution space)



$F(X), G(X), \dots$   
Objective space

## Dominance and non-dominance

- In single-objective optimization
  - We look for a single solution
  - The concept of “A better than B” is trivial
- In multi-objective optimization
  - We are not restricted to find a unique optimal solution
  - the concept of “A better than B” is not trivial

A	2	3	4	5
---	---	---	---	---

B	4	6	5	7
---	---	---	---	---

**A is better than B**

A	3	7	4	8
---	---	---	---	---

B	2	1	2	5
---	---	---	---	---

**B is better than A**

A	1	9	4	5
---	---	---	---	---

B	3	6	5	7
---	---	---	---	---

**None is better**

**A and B are NON-DOMINATED**

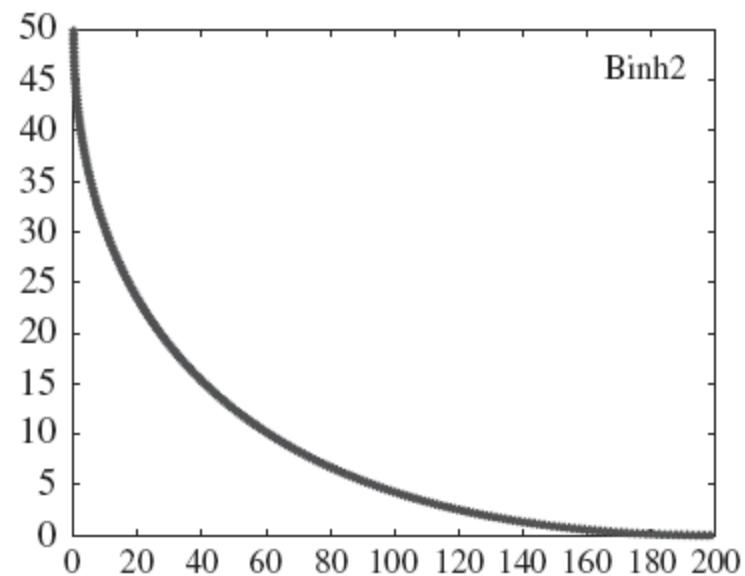
## Pareto optimal set, Pareto optimal front

- The solution of a MOP is a set composed of those non-dominated solutions that cannot be dominated by any other solution outside that set: **Pareto Optimal Set**
- The correspondence of the Pareto optimal set in the objective space is known as **Pareto Optimal Front** (or just **Pareto Front**)

$$\begin{aligned}
 \text{Min } F &= (f_1(\vec{x}), f_2(\vec{x})) \\
 f_1(\vec{x}) &= 4x_1^2 + 4x_2 \\
 f_2(\vec{x}) &= (x_1 - 5)^2 + (x_2 - 5)^2
 \end{aligned}$$

Subject to:

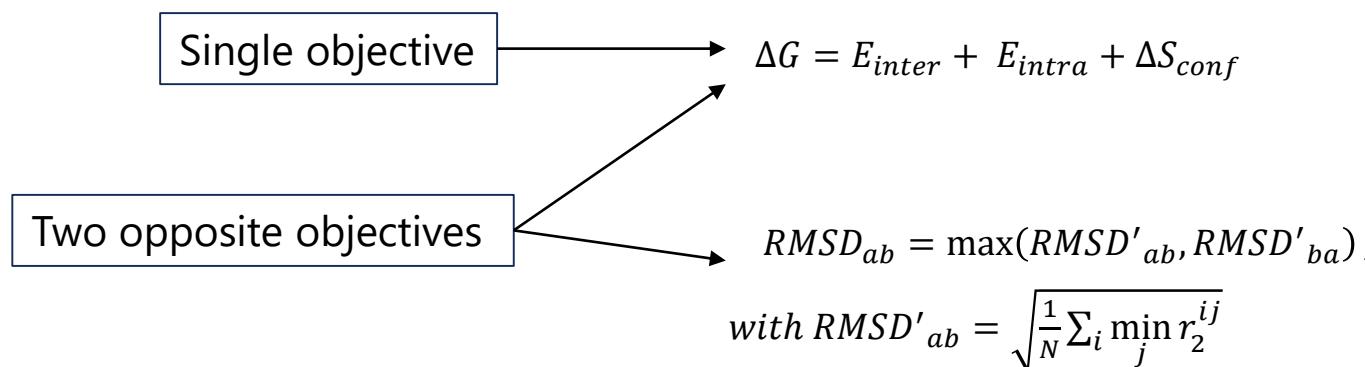
$$\begin{aligned}
 g_1(\vec{x}) &= (x_1 - 5)^2 + x_2^2 - 25 \leq 0 \\
 g_2(\vec{x}) &= -(x_1 - 8)^2 - (x_2 + 3)^2 + 7.7 \leq 0 \\
 0 &\leq x_1 \leq 5 \\
 0 &\geq x_2 \leq 3
 \end{aligned}$$



## Examples in the domain of Bioinformatics

- Molecular docking

Binding energy formulation:



Root Mean Square Deviation (RMSD) score: measure of similarity between the reference ligand position in the receptor and the computed position of the docked ligand.

**Purpose of using RMSD as an objective:**

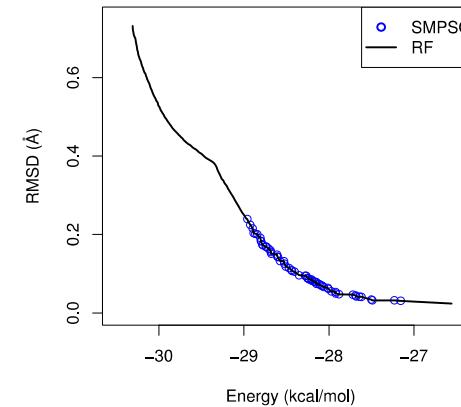
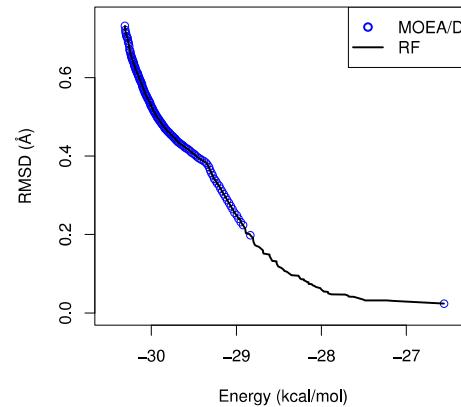
- Guide the algorithm to find solutions whose ligand conformations are close to the ligand position that has been found in studios *in vivo*.
- Useful when the active site of a target mutates and makes it multi-drug resistant.

E. López-Camacho, M.J. García-Godoy, J. García-Nieto, A. J. Nebro, J. F. Aldana-Montes, **A new multi-objective approach for molecular docking based on RMSD and binding energy**. Third International Conference on Algorithms for Computational Biology (AlCoB 2016) Trujillo, Spain

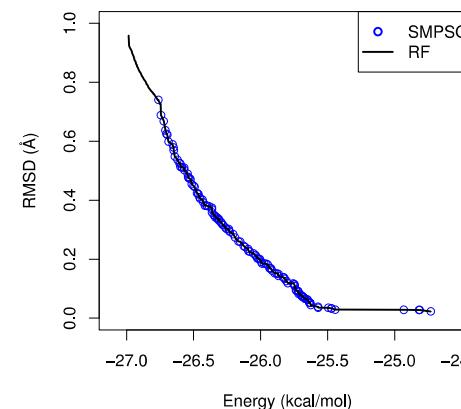
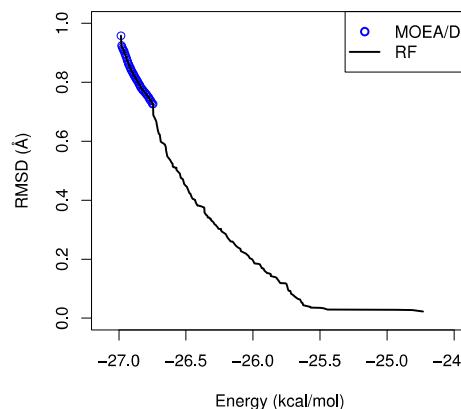
# Examples in the domain of Bioinformatics

- Molecular docking

1D4K



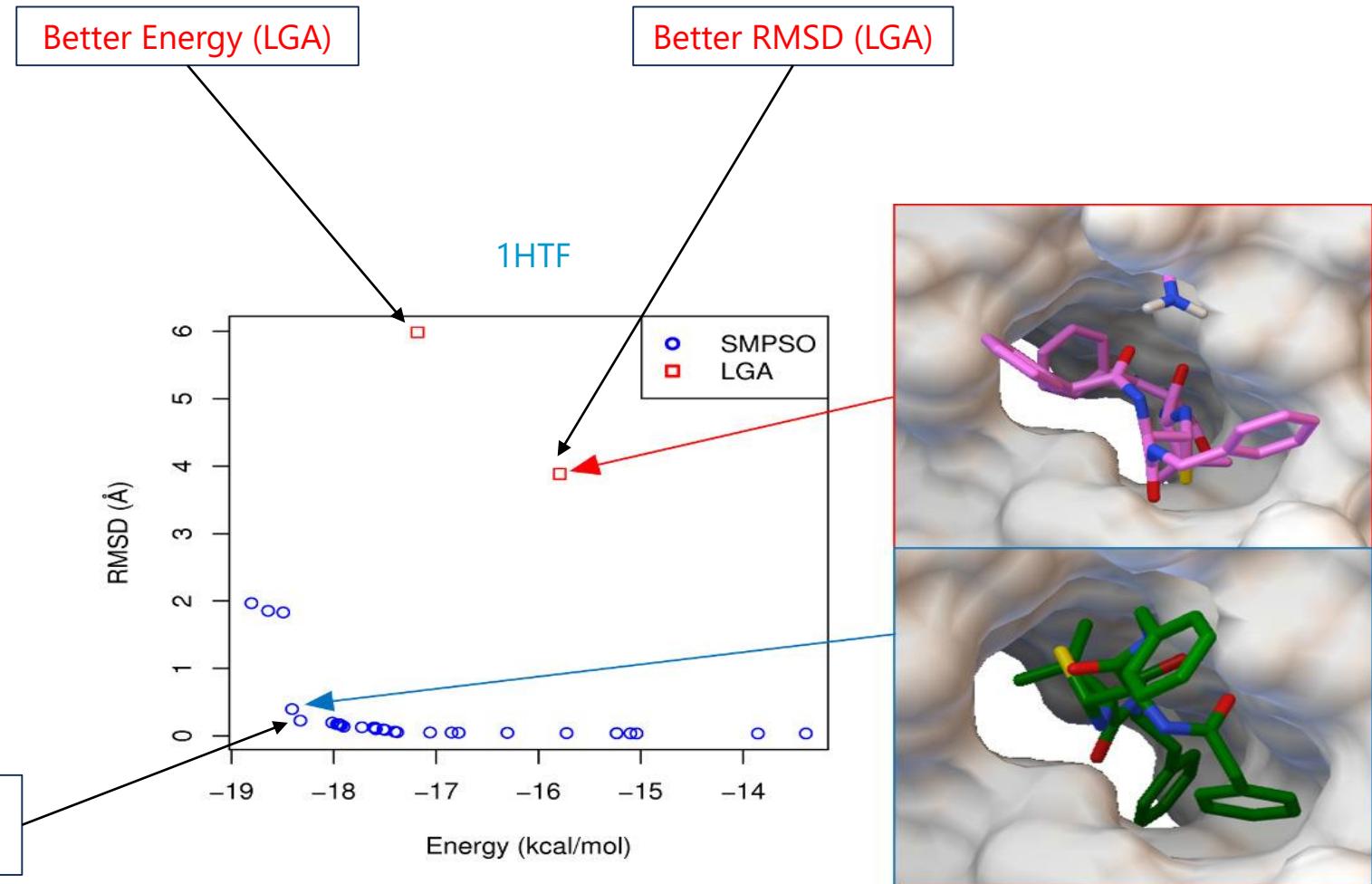
1HIV



E. López-Camacho, M.J. García-Godoy, J. García-Nieto, A. J. Nebro, J. F. Aldana-Montes, **A new multi-objective approach for molecular docking based on RMSD and binding energy**. Third International Conference on Algorithms for Computational Biology (AlCoB 2016) Trujillo, Spain

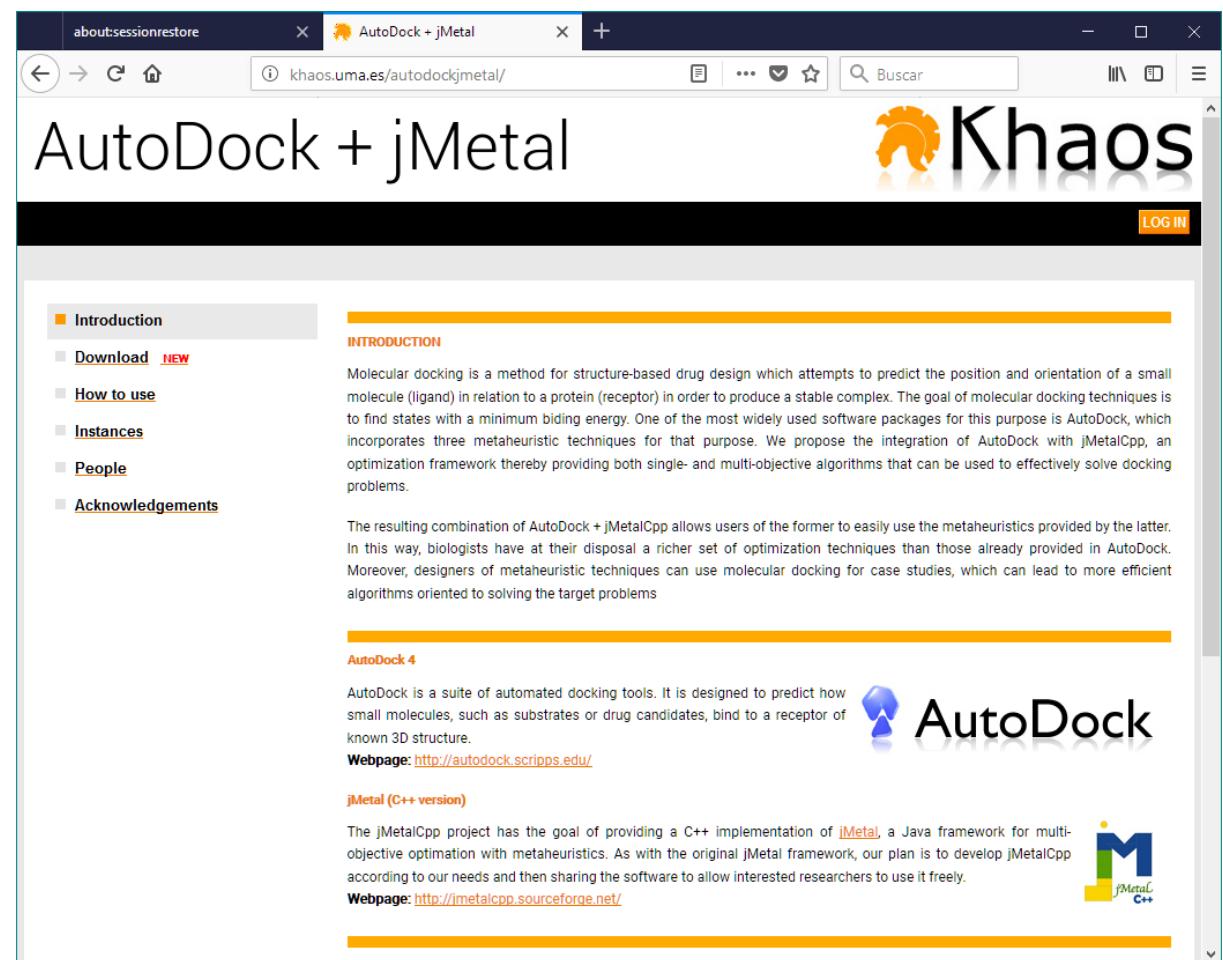
# Examples in the domain of Bioinformatics

- Molecular docking



# Examples in the domain of Bioinformatics

- Molecular docking



The screenshot shows a web browser window with the URL [khaos.uma.es/autodockjmetal/](http://khaos.uma.es/autodockjmetal/). The page title is "AutoDock + jMetal". The left sidebar has a yellow header "Introduction" and links to "Download NEW", "How to use", "Instances", "People", and "Acknowledgements". The main content area has a yellow header "INTRODUCTION" and text about molecular docking. It mentions AutoDock 4 and jMetal C++ version. Logos for AutoDock and jMetal C++ are present.

**INTRODUCTION**

Molecular docking is a method for structure-based drug design which attempts to predict the position and orientation of a small molecule (ligand) in relation to a protein (receptor) in order to produce a stable complex. The goal of molecular docking techniques is to find states with a minimum binding energy. One of the most widely used software packages for this purpose is AutoDock, which incorporates three metaheuristic techniques for that purpose. 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.

**AutoDock 4**

AutoDock is a suite of automated docking tools. It is designed to predict how small molecules, such as substrates or drug candidates, bind to a receptor of known 3D structure.

**Webpage:** <http://autodock.scripps.edu/>

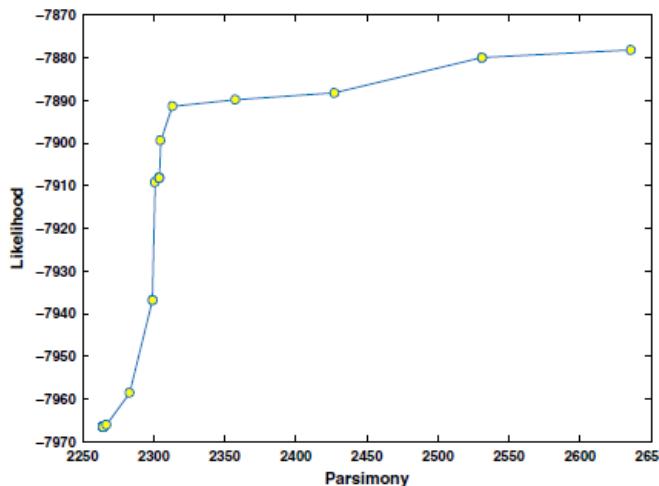
**jMetal (C++ version)**

The jMetalCpp project has the goal of providing a C++ implementation of [jMetal](#), a Java framework for multi-objective optimization with metaheuristics. As with the original jMetal framework, our plan is to develop jMetalCpp according to our needs and then sharing the software to allow interested researchers to use it freely.

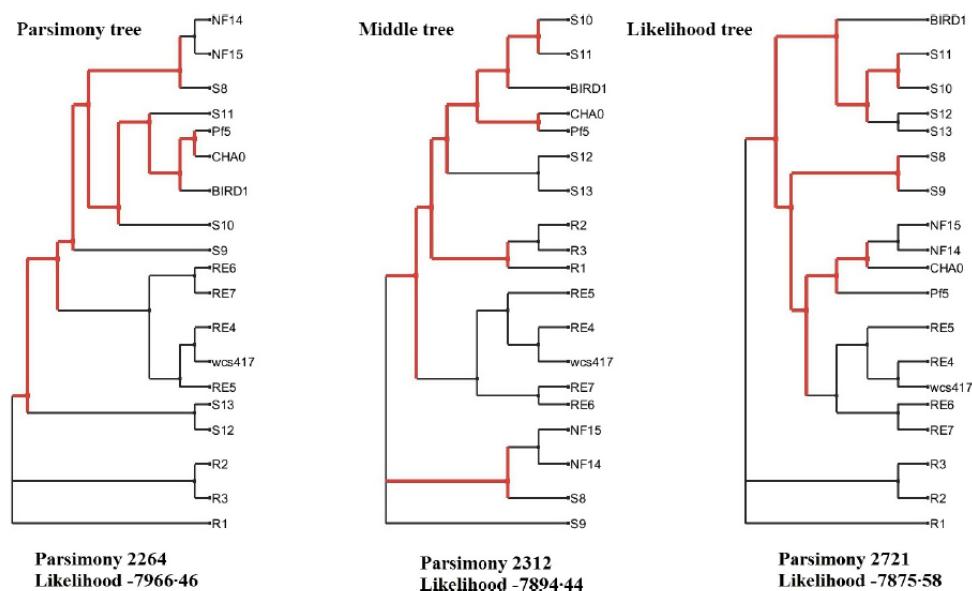
**Webpage:** <http://jmetalcpp.sourceforge.net/>

# Examples of the domain of Bioinformatics

- Phylogenetic inference
  - Two objectives: Likelihood and parsimony



**Fig. 3.** Pareto front approximation performed by MO-Phylogenetics over a real nucleotide data set (*16S\_rRNA*) using the stepwise addition method to generate starting phylogenetic trees and the numeric method Newton-Raphson to optimize all branch lengths for these starting topologies. The metaheuristic is SMS-EMOA.

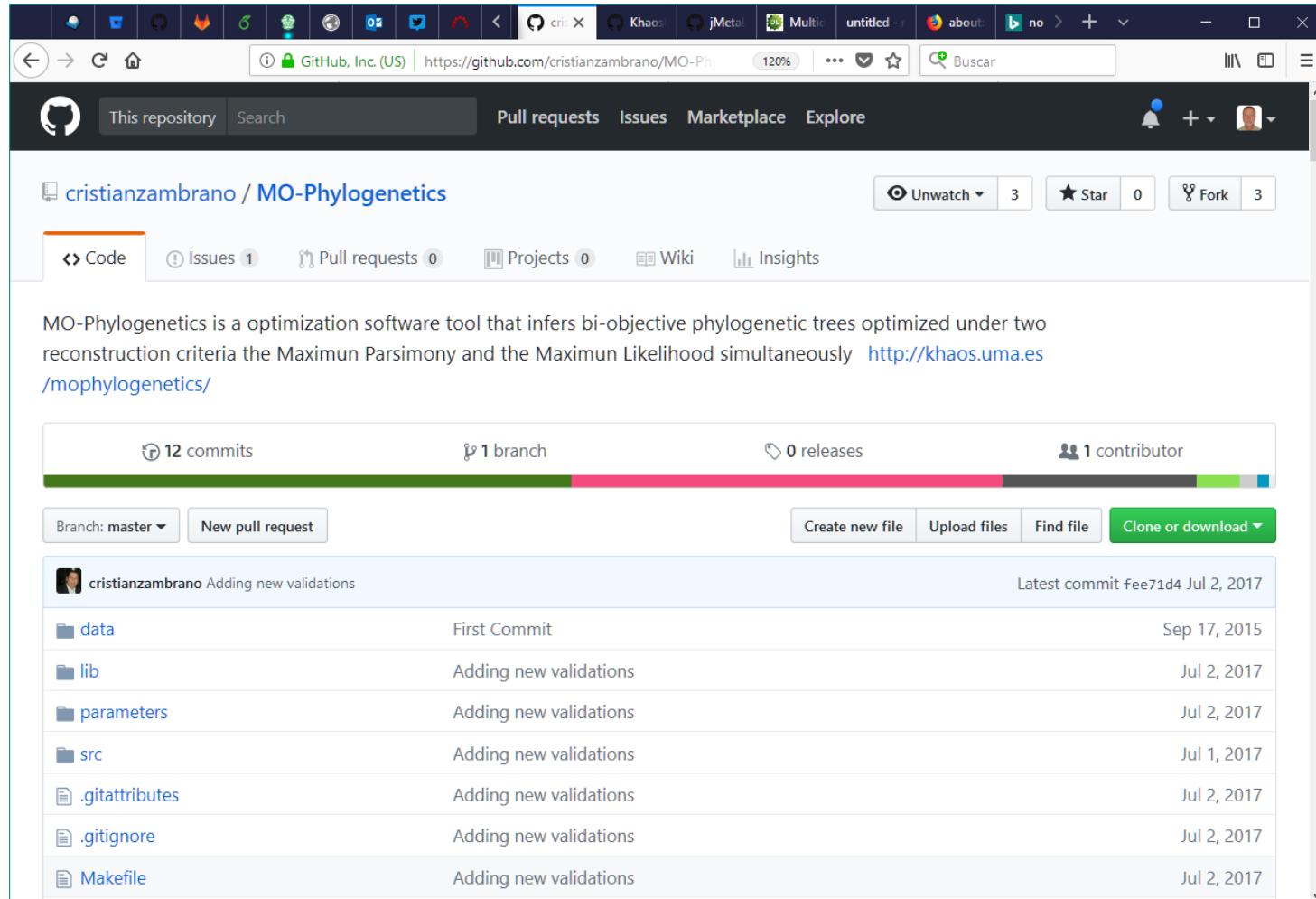


**Fig. 4.** Comparison of the extreme (best parsimony - left, best likelihood - right) and middle (centre) fronts of the *16S\_rRNA* data set obtained by MO-Phylogenetics in Example 2. The differences among the trees are highlighted in red colour. (The trees have been obtained with the tool TreeJuxtaposer)

C. Zambrano-Vega, A.J. Nebro and J.F. Aldana-Montes. **MO-Phylogenetics: a phylogenetic inference software tool with multi-objective evolutionary metaheuristics**. Methods in Ecology and Evolution. Volume 7, Issue 7 July 2016. Pages 800–805.

# Examples of the domain of Bioinformatics

- Phylogenetic inference

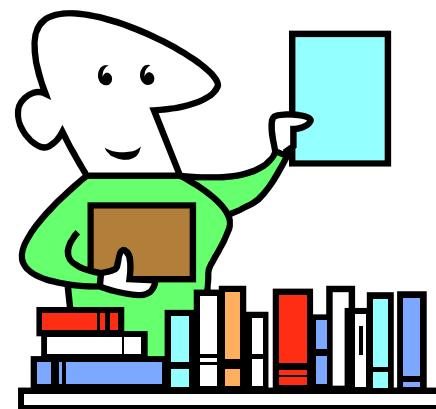


MO-Phylogenetics is a optimization software tool that infers bi-objective phylogenetic trees optimized under two reconstruction criteria the Maximun Parsimony and the Maximun Likelihood simultaneously <http://khaos.uma.es/mophylogenetics/>

Branch: master	New pull request	Create new file	Upload files	Find file	Clone or download
cristianzambrano Adding new validations					
					Latest commit fee71d4 Jul 2, 2017
	First Commit				Sep 17, 2015
	Adding new validations				Jul 2, 2017
	Adding new validations				Jul 2, 2017
	Adding new validations				Jul 1, 2017
	Adding new validations				Jul 2, 2017
	Adding new validations				Jul 2, 2017
	Adding new validations				Jul 2, 2017

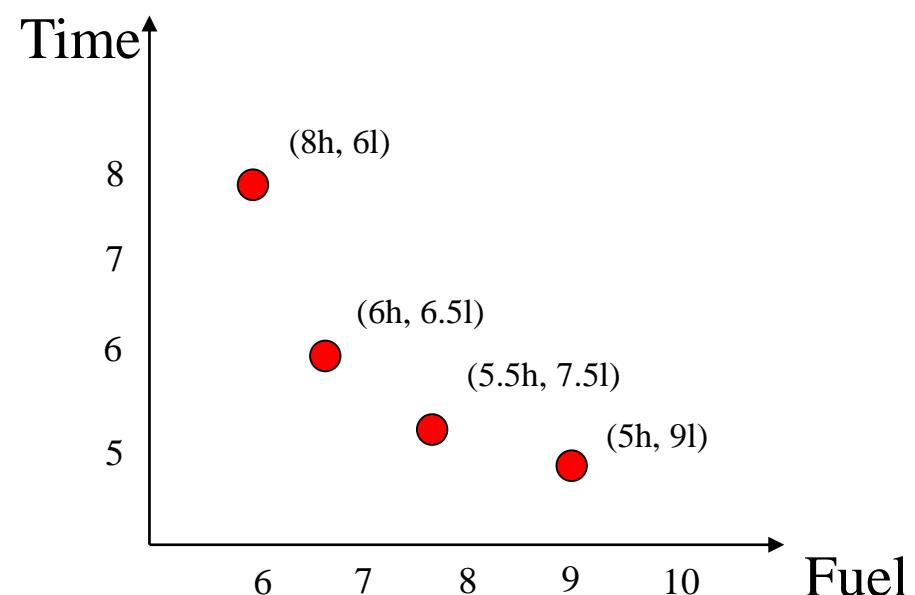
## Multi-Objective Optimization and Decision Making

- Finding the Pareto front of a problem is not the last step in multi-objective optimization
- In practice, an expert in the domain (the **decision maker**) has to choose the best trade-off solution



## Multi-Objective Optimization and Decision Making

- In the example of traveling from Málaga to Madrid
  - If time is important
    - Choose (5h, 9l)
  - If consumption is important:
    - Choose (8h, 6l)
  - Compromise solution:
    - (6h, 6.5l)

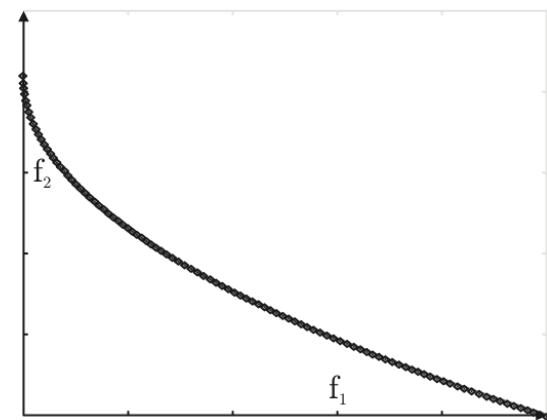
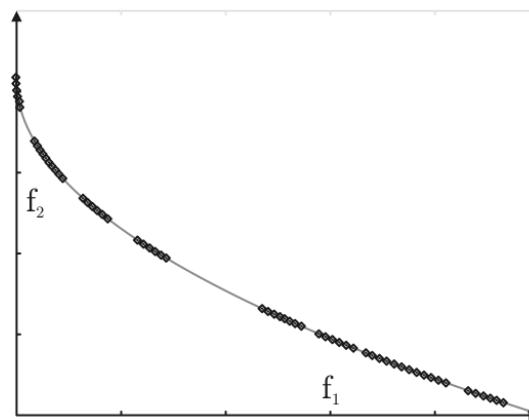
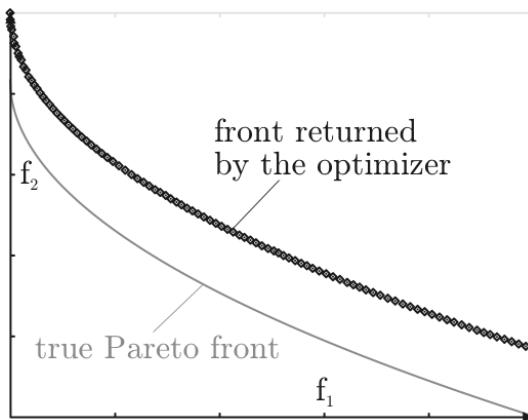


## Goals of Multi-Objective Optimization

- The ideal goal is to obtain the Pareto front
- Unfortunately, this is unpractical in real-world problems as commented before:
  - NP-hard complexity, non-linearity, epistasis, ...
  - Frequently, exact techniques are not useful
- Alternative: Use non-exact algorithms
  - E.g. *Metaheuristics*
  - These techniques provide an *approximation* to the Pareto front

## Goals of Multi-Objective Optimization

- Main goal when using non-exact techniques:
  - Obtaining an approximation to the Pareto front with two properties
    - Convergence
    - Diversity



## Other important goals in practice

- Fast convergence to the Pareto front
  - The decision maker may need the results quickly
- This can be achieved
  - Designing more efficient algorithms
    - Requiring less function evaluations to converge
  - Using parallelism
    - To speed-up the execution time

## The issue of fitness assignment

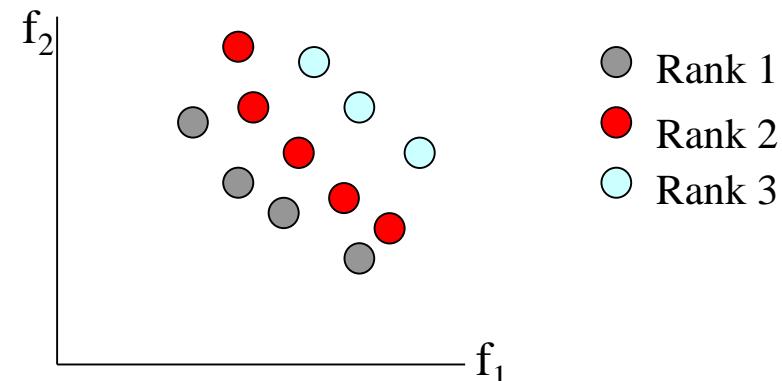
- Single-objective optimization
  - The fitness of a solution is one value
  - The concept “better than” is easily defined according to the fitness value
- Multi-objective optimization
  - The fitness is a vector of values
  - The concept “better than” cannot be defined as in single-optimization
- Quality indicators are used to assess algorithm performance
  - Convergence: generational distance, epsilon, etc.
  - Diversity: spread, ...
  - Both: hypervolume, inverted generational distance, ...

## The issue of fitness assignment

- Multi-objective optimization
  - The strategy applied in many multi-objective metaheuristics is assign to the solutions a unique value to be used as “*fitness*”
  - This way it is possible to define a relationship “better than”
  - Usually, the fitness is the combination of measures about convergence and diversity
- Examples:
  - Ranking -> convergence
  - Crowding distance -> diversity

## Ranking

- The population is classified (ranked):
  - Rank 1: non-dominated individuals
  - Rank 2: non-dominated individuals after removing the individuals with Rank 1
  - An so on
- Individual ranking is used to decide about better individuals



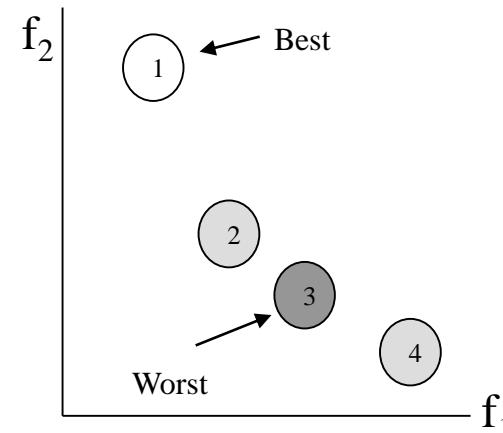
- Fitness assignment using only information about dominance is not enough
- Intuitively, it can help in driving the search to the true Pareto front (**convergence**)
  - But, also intuitively, something more is necessary to obtain a well distributed Pareto front approximation

Alternative

- To complement the fitness value with a **density indicator**

## Density Estimator

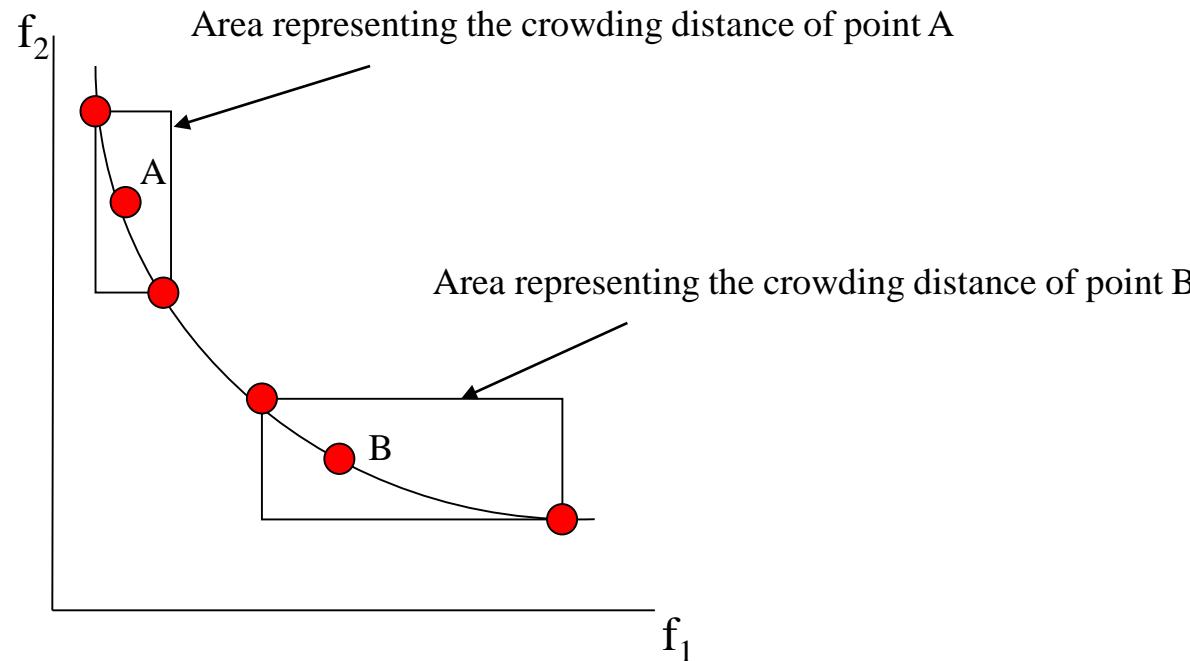
- A density estimator is a measure that gives information about density in the objective space around a given solution
- Given two individuals with the same fitness, the density estimator indicates the best and/or the worst individuals



- If a solution has to be chosen to be replaced
  - The candidate one is solution 3

## Density Estimator: Crowding Distance

- Proposed in NSGA-II

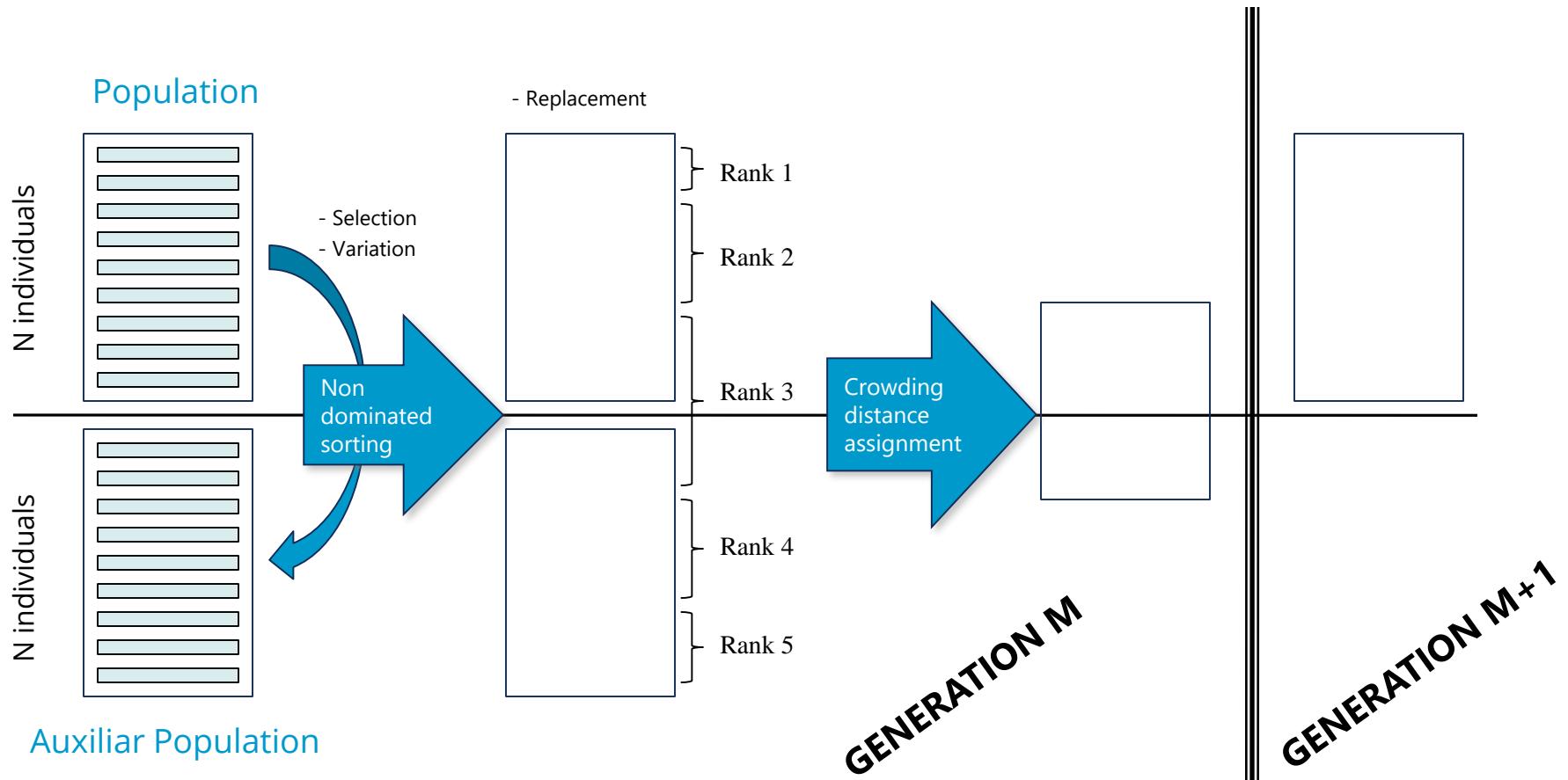


Point B is in a less crowded region than point A

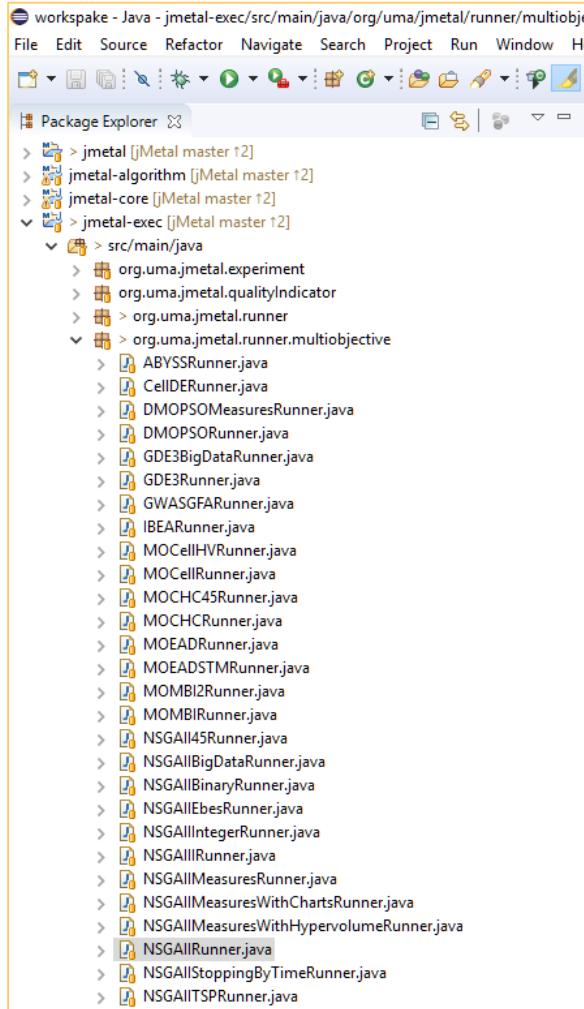
## Example of multi-objective metaheuristic: NSGA-II

- *Non-dominated Sorting Genetic Algorithm II*
- Proposed by K. Deb (2002)
- By far, the most popular metaheuristic for multi-objective optimization
  - It has become the *de facto* standard
- Features
  - Ranking using non-dominated sorting
  - Crowding distance as density estimator

## Example of multi-objective metaheuristic: NSGA-II



# Example 1: configuring and running NSGA-II



```

public class NSGAIIRunner extends AbstractAlgorithmRunner {
    * @param args Command line arguments.
    public static void main(String[] args) throws JMetalException, FileNotFoundException {
        Problem<DoubleSolution> problem;
        Algorithm<List<DoubleSolution>> algorithm;
        CrossoverOperator<DoubleSolution> crossover;
        MutationOperator<DoubleSolution> mutation;
        SelectionOperator<List<DoubleSolution>, DoubleSolution> selection;
        String referenceParetoFront = "";

        String problemName ;
        if (args.length == 1) {
            problemName = args[0];
        } else if (args.length == 2) {
            problemName = args[0] ;
            referenceParetoFront = args[1] ;
        } else {
            problemName = "org.uma.jmetal.problem.multiobjective.zdt.ZDT1";
            referenceParetoFront = "jmetal-problem/src/test/resources/pareto_fronts/ZDT1.pf" ;
        }

        problem = ProblemUtils.<DoubleSolution> LoadProblem(problemName);

        double crossoverProbability = 0.9 ;
        double crossoverDistributionIndex = 20.0 ;
        crossover = new SBXCrossover(crossoverProbability, crossoverDistributionIndex) ;

        double mutationProbability = 1.0 / problem.getNumberOfVariables() ;
        double mutationDistributionIndex = 20.0 ;
        mutation = new PolynomialMutation(mutationProbability, mutationDistributionIndex) ;

        selection = new BinaryTournamentSelection<DoubleSolution>(
            new RankingAndCrowdingDistanceComparator<DoubleSolution>());

        algorithm = new NSGAIIBuilder<DoubleSolution>(problem, crossover, mutation)
            .setSelectionOperator(selection)
            .setMaxEvaluations(25000)
            .setPopulationSize(100)
            .build() ;

        AlgorithmRunner algorithmRunner = new AlgorithmRunner.Executor(algorithm)
            .execute() ;

        List<DoubleSolution> population = algorithm.getResult() ;
        long computingTime = algorithmRunner.getComputingTime() ;

        JMetalLogger.logger.info("Total execution time: " + computingTime + "ms");

        printFinalSolutionSet(population);
        if (!referenceParetoFront.equals("")) {
            printQualityIndicators(population, referenceParetoFront) ;
        }
    }
}

```

# Example 1: configuring and running NSGA-II

RStudio Console output:

```

terminated> NSGAIIRunner [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (5 sept. 2017 10:41:35)
2017-09-05 10:41:35.682 INFORMACIÓN: Loggers configured with null [org.uma.jmetal.util.JMetalLogger configureLoggers]
2017-09-05 10:41:35.709 INFORMACIÓN: Total execution time: 410ms [org.uma.jmetal.runner.multiobjective.NSGAIIRunner main]
2017-09-05 10:41:35.746 INFORMACIÓN: Random seed: 1504600895258 [org.uma.jmetal.runner.AbstractAlgorithmRunner printFinal]
2017-09-05 10:41:35.747 INFORMACIÓN: Objectives values have been written to file FUN.tsv [org.uma.jmetal.runner.AbstractAlgorithmRunner printFinal]
2017-09-05 10:41:35.747 INFORMACIÓN: Variables values have been written to file VAR.tsv [org.uma.jmetal.runner.AbstractAlgorithmRunner printFinal]
2017-09-05 10:41:35.818 INFORMACIÓN:
Hypervolume (N) : 0.6597068785327224
Hypervolume      : 0.6597068785327224
Epsilon (N)       : 0.012834091463729713
Epsilon          : 0.012834091463729713
GD (N)           : 2.0135698109623943E-4
GD               : 2.0135698109623943E-4
IGD (N)          : 1.8094689755722708E-4
IGD              : 1.8094689755722708E-4
IGD+ (N)         : 0.003580506492809246
IGD+             : 0.003580506492809246
Spread (N)        : 0.33127271727483215
Spread            : 0.33127271727483215
Error ratio       : 1.0
[org.uma.jmetal.runner.AbstractAlgorithmRunner printQualityIndicators]

```

Windows PowerShell output:

```

PS C:\Users\ajnebro\Softw\jMetal\jMetal> dir
Directorio: C:\Users\ajnebro\Softw\jMetal\jMetal

Mode                LastWriteTime     Length Name
----                -----        --  --
d----- 05/09/2017 10:13                 .git
d----- 05/09/2017 10:13                 .idea
d----- 02/06/2017 11:36                 .settings
d----- 05/09/2017  9:10   jmetal-algorithm
d----- 25/07/2017 16:58   jmetal-core
d----- 05/09/2017 11:19   jmetal-exec
d----- 25/07/2017 16:58   jmetal-problem
d----- 19/06/2017 13:30                 target
d----- 25/07/2017 16:58   2152 .gitignore
a----  02/06/2017 11:36      386 .project
a----  05/09/2017 10:51   4231 .RData
a----  05/09/2017 10:51      102 .Rhistory
a----  25/07/2017 16:58      119 .travis.yml
a----  05/09/2017 10:41   4086 FUN.tsv
a----  12/05/2017 11:51      574 jmetal.iml
a----  05/09/2017 10:41   2742 jMetal.log
a----  25/07/2017 16:58      1101 LICENSE.txt
a----  25/07/2017 16:58   10703 pom.xml
a----  25/07/2017 16:58      1697 README.md
a----  25/07/2017 16:58      1596 sonar-project.properties
-a---  05/09/2017 10:41   64146 VAR.tsv

```

RStudio Environment pane:

```

Environment History
Import Dataset List
Global Environment
Data
a 100 obs. of 2 variables

```

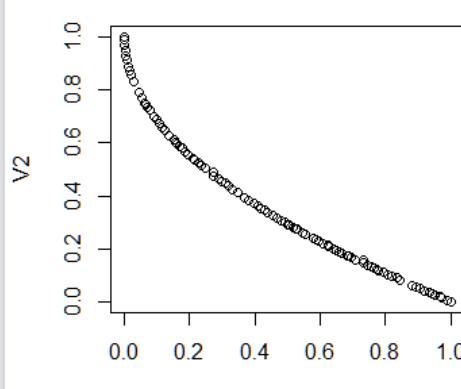
RStudio Data pane:

```

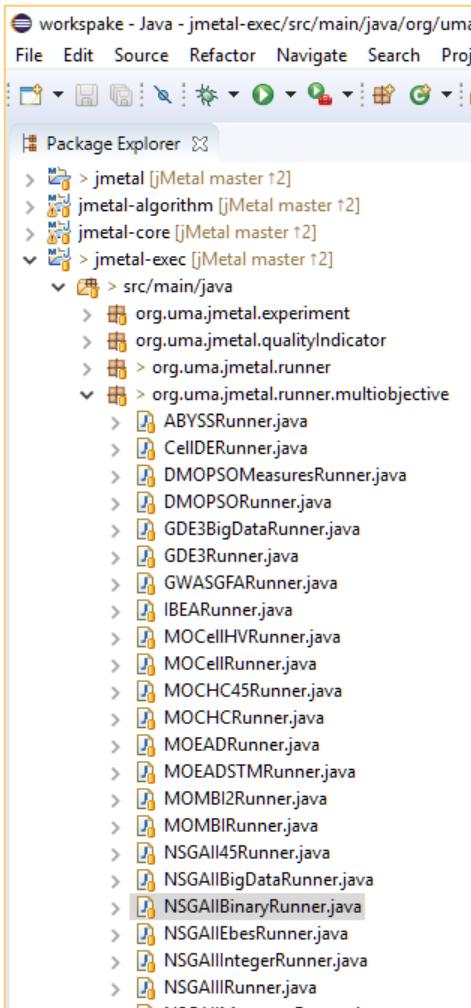
a

```

RStudio Plot pane:



## Example 2: solving binary problems with NSGA-II



```

public class NSGAIIBinaryRunner extends AbstractAlgorithmRunner {
    * @param args Command line arguments.
    public static void main(String[] args) throws Exception {
        BinaryProblem problem;
        Algorithm<List<BinarySolution>> algorithm;
        CrossoverOperator<BinarySolution> crossover;
        MutationOperator<BinarySolution> mutation;
        SelectionOperator<List<BinarySolution>, BinarySolution> selection;

        String problemName ;
        String referenceParetoFront = "" ;
        if (args.length == 1) {
            problemName = args[0];
        } else if (args.length == 2) {
            problemName = args[0] ;
            referenceParetoFront = args[1] ;
        } else {
            problemName = "org.uma.jmetal.problem.multiobjective.zdt.ZDT5";
            referenceParetoFront = "" ;
        }

        problem = (BinaryProblem) ProblemUtils.<BinarySolution> LoadProblem(problemName);

        double crossoverProbability = 0.9 ;
        crossover = new SinglePointCrossover(crossoverProbability) ;

        double mutationProbability = 1.0 / problem.getNumberOfBits(0) ;
        mutation = new BitFlipMutation(mutationProbability) ;

        selection = new BinaryTournamentSelection<BinarySolution>() ;

        algorithm = new NSGAIIBuilder<BinarySolution>(problem, crossover, mutation)
            .setSelectionOperator(selection)
            .setMaxEvaluations(25000)
            .setPopulationSize(100)
            .build() ;

        AlgorithmRunner algorithmRunner = new AlgorithmRunner.Executor(algorithm)
            .execute() ;

        List<BinarySolution> population = algorithm.getResult() ;
        long computingTime = algorithmRunner.getComputingTime() ;

        JMetalLogger.Logger.info("Total execution time: " + computingTime + "ms");

        printFinalSolutionSet(population);
        if (!referenceParetoFront.equals("")) {
            printQualityIndicators(population, referenceParetoFront) ;
        }
    }
}

```

## Example 2: solving binary problems with NSGA-II

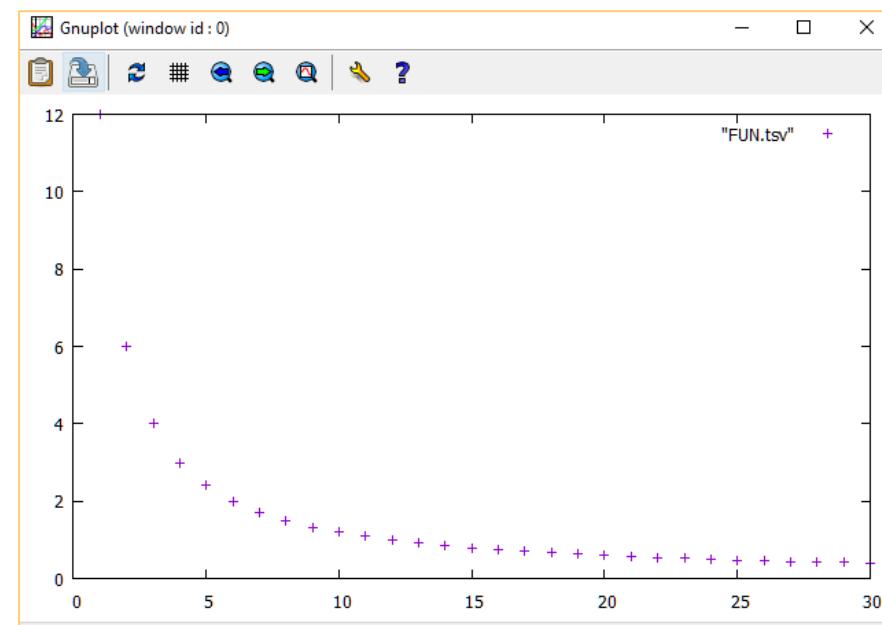
Problems Javadoc Declaration Search Console

```
<terminated> NSGAIIBinaryRunner [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (5 sept. 2017 11:19:16)
2017-09-05 11:19:17.671 INFORMACIÓN: Loggers configured with null [org.uma.jmetal.util.JMetalLogger configureLoggers]
2017-09-05 11:19:17.700 INFORMACIÓN: Total execution time: 631ms [org.uma.jmetal.runner.multiobjective.NSGAIIBinaryRunner main]
2017-09-05 11:19:17.716 INFORMACIÓN: Random seed: 1504603157026 [org.uma.jmetal.runner.AbstractAlgorithmRunner printFinalSolutionSet]
2017-09-05 11:19:17.717 INFORMACIÓN: Objectives values have been written to file FUN.tsv [org.uma.jmetal.runner.AbstractAlgorithmRunner printFinalSolutionSet]
2017-09-05 11:19:17.717 INFORMACIÓN: Variables values have been written to file VAR.tsv [org.uma.jmetal.runner.AbstractAlgorithmRunner printFinalSolutionSet]
```

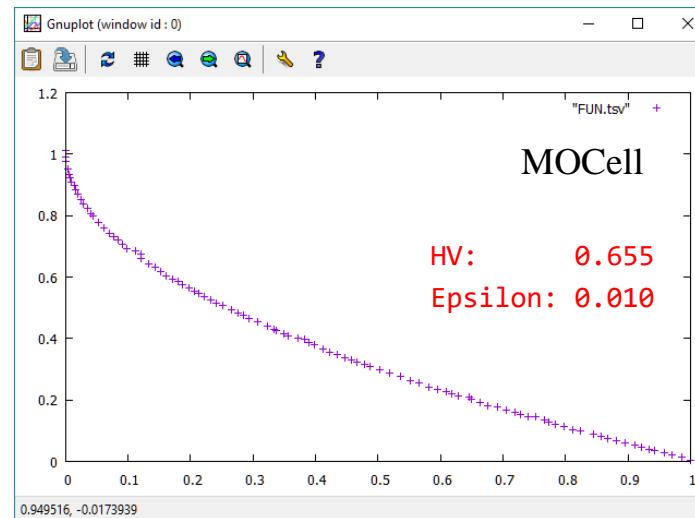
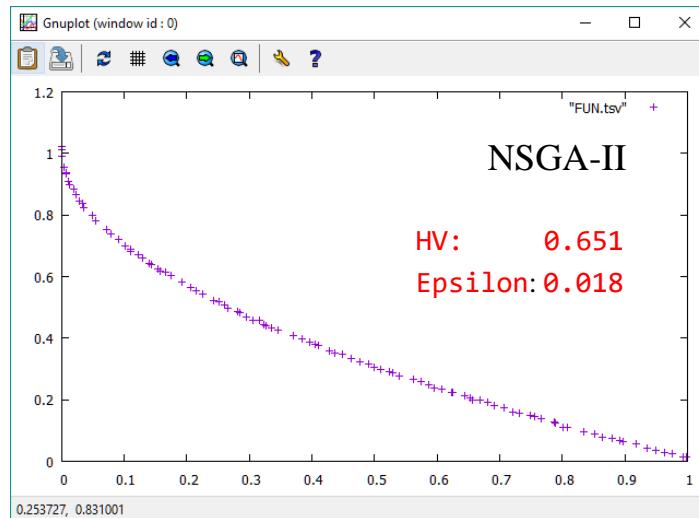
gnuplot 5.0 patchlevel 7 - console version

```
Terminal type set to 'wxt'
gnuplot> pwd
C:\Users\ajnebro\Documents
gnuplot> cd ..
^
      invalid expression

gnuplot> cd "../Softw/jMetal/jMetal"
gnuplot> plot "FUN.tsv"
gnuplot>
```



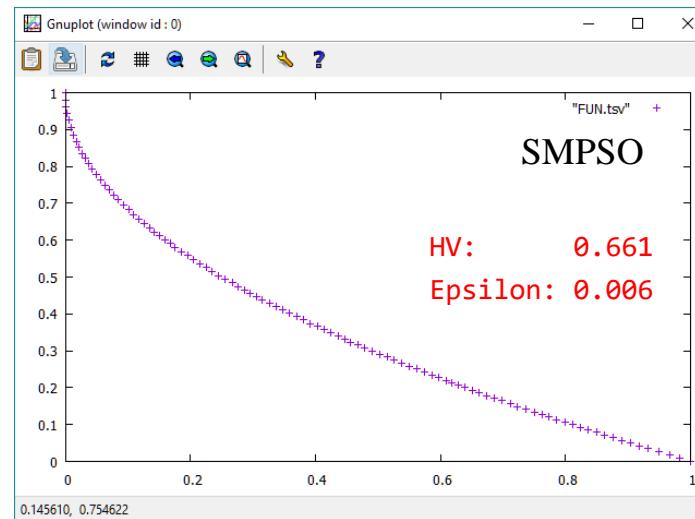
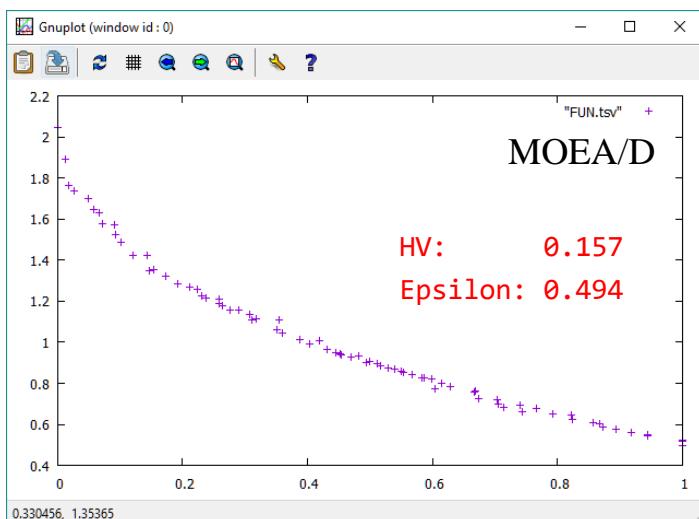
## Example 3: solving the ZDT4 problem



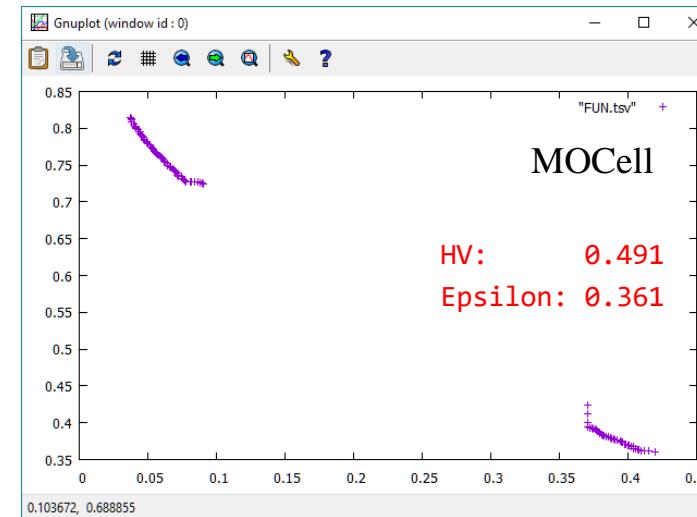
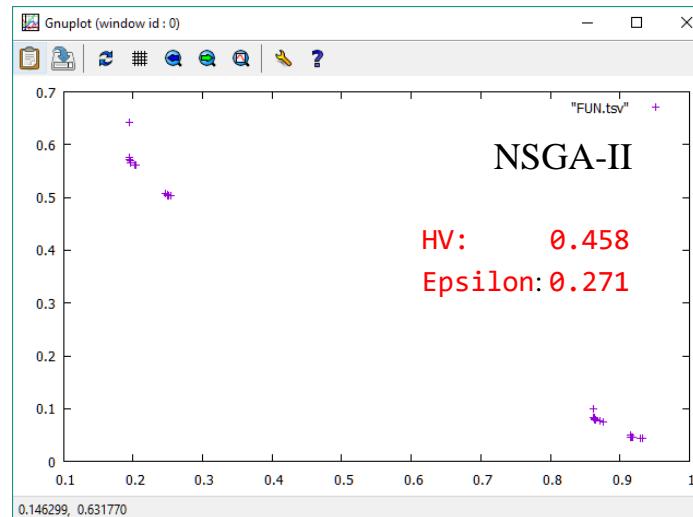
Common settings

Pop. size: 100

Evaluations: 25000



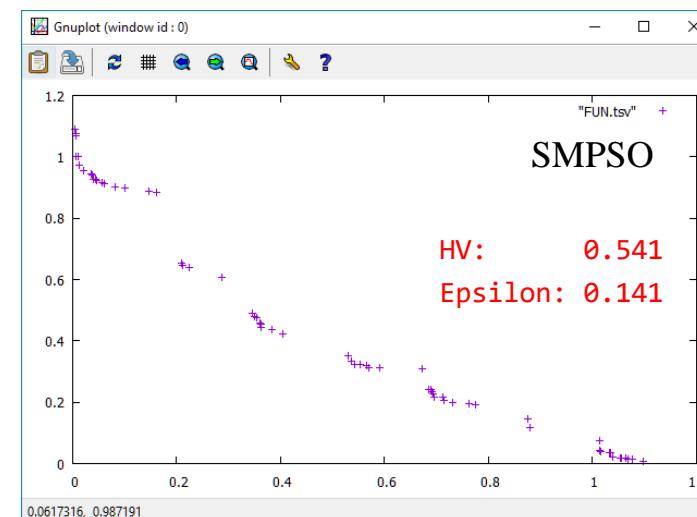
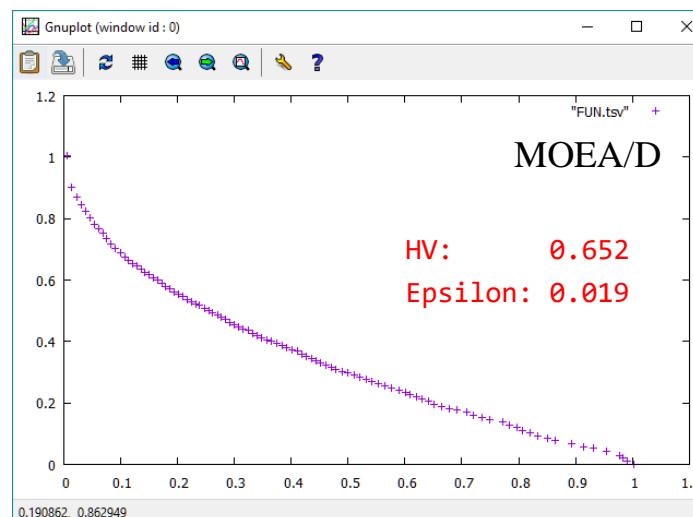
## Example 4: solving the LZ09-F2 problem



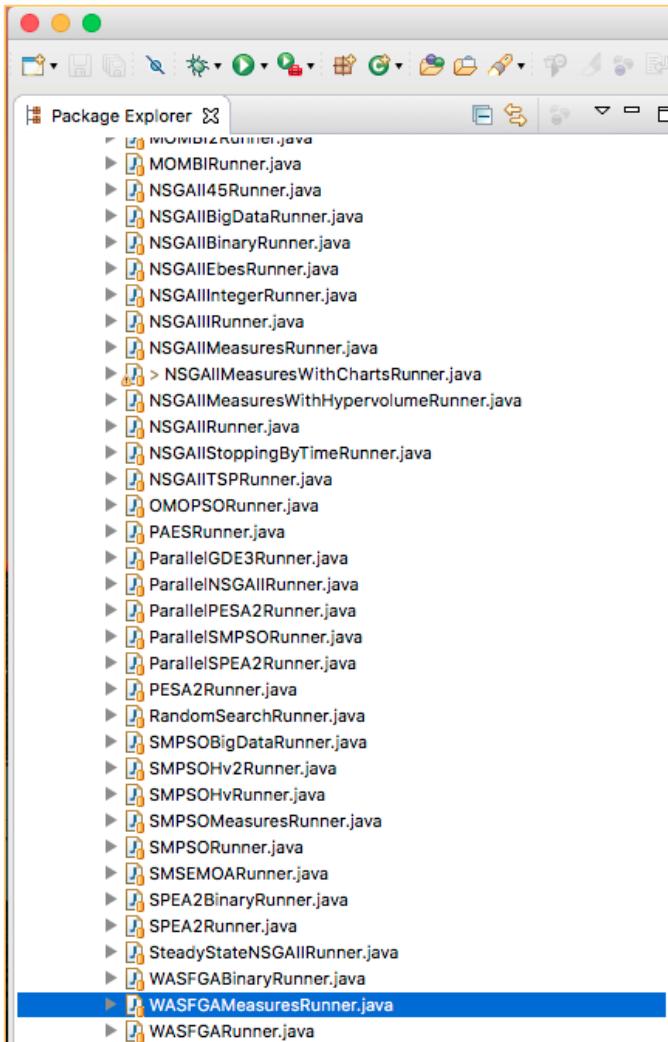
Common settings

Pop. size: 100

Evaluations: 150000



## Example 4: indicating preferences (WASF-GA)



```

public class WASFGAMeasuresRunner extends AbstractAlgorithmRunner {
    * @param args Command line arguments. */
    public static void main(String[] args) throws JMetalException, IOException {
        Problem<DoubleSolution> problem;
        Algorithm<List<DoubleSolution>> algorithm;
        CrossoverOperator<DoubleSolution> crossover;
        MutationOperator<DoubleSolution> mutation;
        SelectionOperator<List<DoubleSolution>, DoubleSolution> selection;
        String referenceParetoFront = "";
        List<Double> referencePoint = null;

        String problemName ;
        if (args.length == 1) {
            problemName = args[0];
        } else if (args.length == 2) {
            problemName = args[0] ;
            referenceParetoFront = args[1] ;
        } else {
            problemName = "org.uma.jmetal.problem.multiobjective.zdt.ZDT1";
            referenceParetoFront = "jmetal-problem/src/test/resources/pareto_fronts/ZDT1.pf" ;
        }

        problem = ProblemUtils.<DoubleSolution> loadProblem(problemName);

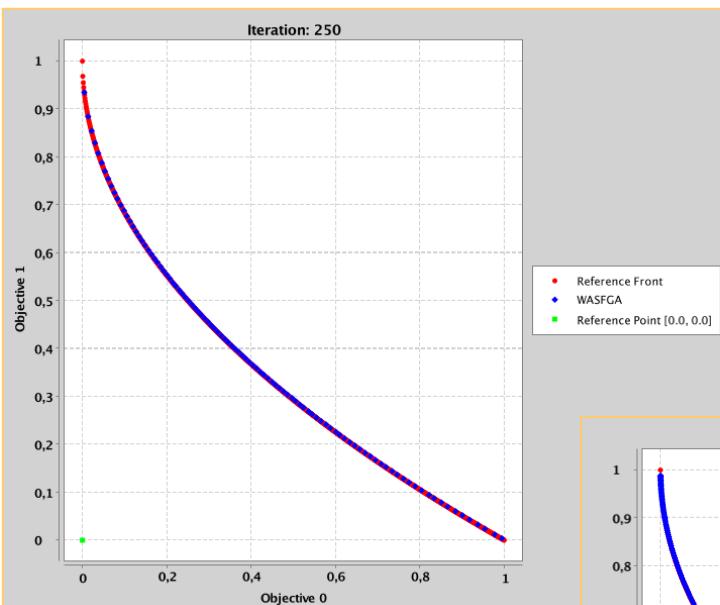
        referencePoint = new ArrayList<>();
        referencePoint.add(0.5);
        referencePoint.add(0.5);

        double crossoverProbability = 0.9 ;
        double crossoverDistributionIndex = 20.0 ;
        crossover = new SBXCrossover(crossoverProbability, crossoverDistributionIndex) ;

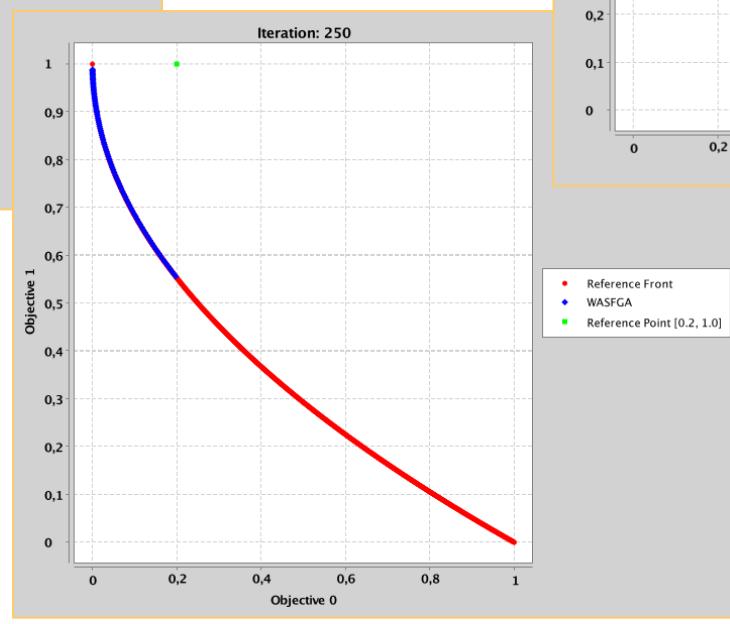
        double mutationProbability = 1.0 / problem.getNumberOfVariables() ;
        double mutationDistributionIndex = 20.0 ;
        mutation = new PolynomialMutation(mutationProbability, mutationDistributionIndex) ;
    }
}

```

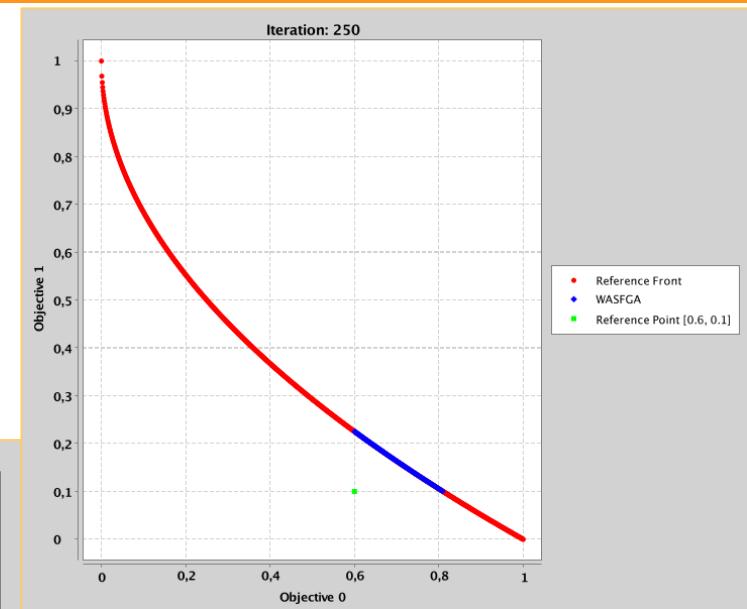
## Example 4: indicating preferences (WASF-GA)



Reference point: [0.0, 0.0]

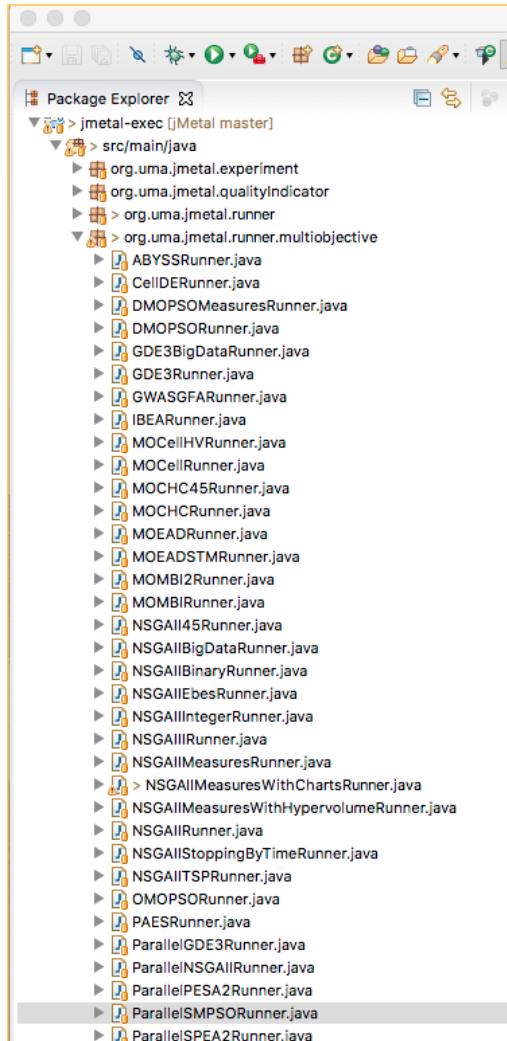


Reference point: [0.2, 1.0]



Reference point: [0.6, 0.1]

## Example 5: parallel execution



```

public class ParallelSMPSORunner extends AbstractAlgorithmRunner {
    * @param args Command line arguments. The first (optional) argument specifies...
    public static void main(String[] args) throws Exception {
        DoubleProblem problem;
        Algorithm<List<DoubleSolution>> algorithm;
        MutationOperator<DoubleSolution> mutation;
        SolutionListEvaluator<DoubleSolution> evaluator ;

        String referenceParetoFront = "" ;

        String problemName ;
        if (args.length == 1) {
            problemName = args[0];
        } else if (args.length == 2) {
            problemName = args[0] ;
            referenceParetoFront = args[1] ;
        } else {
            problemName = "org.uma.jmetal.problem.multiobjective.zdt.ZDT1";
            referenceParetoFront = "jmetal-problem/src/test/resources/pareto_fronts/ZDT1.pf" ;
        }

        problem = (DoubleProblem) ProblemUtils.<DoubleSolution> loadProblem(problemName);

        BoundedArchive<DoubleSolution> archive = new CrowdingDistanceArchive<DoubleSolution>(100) ;

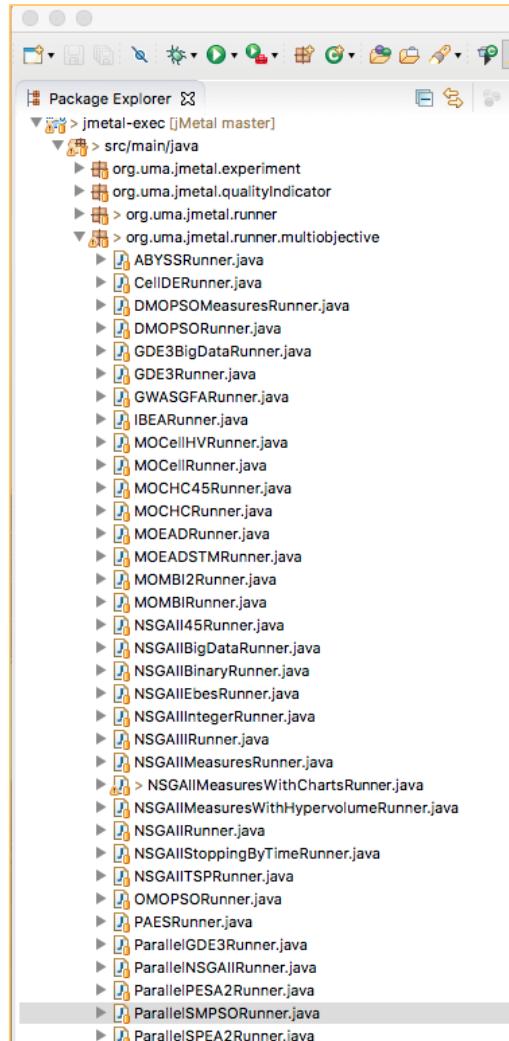
        double mutationProbability = 1.0 / problem.getNumberOfVariables() ;
        double mutationDistributionIndex = 20.0 ;
        mutation = new PolynomialMutation(mutationProbability, mutationDistributionIndex) ;

        evaluator = new MultithreadedSolutionListEvaluator<DoubleSolution>(0, problem) ;

        algorithm = new SMPSOBuilder(problem, archive)
            .setMutation(mutation)
            .setMaxIterations(250)
            .setSwarmSize(100)
            .setSolutionListEvaluator(evaluator)
            .build();
    }
}

```

## Example 5: parallel execution



```

public class ParallelSMPSORunner extends AbstractAlgorithmRunner {
    * @param args Command line arguments. The first (optional) argument specifies..
    public static void main(String[] args) throws Exception {
        DoubleProblem problem;
        Algorithm<List<DoubleSolution>> algorithm;
        MutationOperator<DoubleSolution> mutation;
        SolutionListEvaluator<DoubleSolution> evaluator ;

        String referenceParetoFront = "" ;

        String problemName ;
        if (args.length == 1) {
            problemName = args[0];
        } else if (args.length == 2) {
            problemName = args[0] ;
            referenceParetoFront = args[1] ;
        } else {
            problemName = "org.uma.jmetal.problem.multiobjective.zdt.ZDT1";
            referenceParetoFront = "jmetal-problem/src/test/resources/pareto_fronts/ZDT1.pf" ;
        }

        problem = (DoubleProblem) ProblemUtils.<DoubleSolution> loadProblem(problemName);

        BoundedArchive<DoubleSolution> archive = new CrowdingDistanceArchive<DoubleSolution>(100) ;

        double mutationProbability = 1.0 / problem.getNumberOfVariables() ;
        double mutationDistributionIndex = 20.0 ;
        mutation = new PolynomialMutation(mutationProbability, mutationDistributionIndex) ;

        evaluator = new MultithreadedSolutionListEvaluator<DoubleSolution>(0, problem) ;

        algorithm = new SMPSOBuilder(problem, archive)
            .setMutation(mutation)
            .setMaxIterations(250)
            .setSwarmSize(100)
            .setSolutionListEvaluator(evaluator)
        .build();
    }
}

```

## Table of contents

- About this tutorial
- Metaheuristics as optimization techniques
- Dealing with problems with more than one objective function
- **Case study: multiple sequence alignment**
- Further developments and related projects

## Case study: multiple sequence alignment

- The problem:
  - To find an optimum alignment of three or more biological sequences (DNA, RNA, proteins) to identify common regions
  - How to align: Inserting gaps inside the sequences

Unaligned sequences

$s_1$ : SKPKPIVAANWSLSELI  
 $s_2$ : PKPIVAG  
 $s_3$ : APPKFFVGGNWKMNGKRKSLG  
 $s_4$ : APSRKFFVGGNW

Aligned sequences

$s'_1$ : SK-PKPIVAANWSLSELI----  
 $s'_2$ : ---PKPIVAG-----  
 $s'_3$ : AP-PKFFVGGNWKMNGKRKSLG  
 $s'_4$ : APSRKFFVGGNW-----

## Case study: multiple sequence alignment

- Computational complexity:
  - NP-complexity
    - The computational complexity augments exponentially with the number of sequences and their length
    - The pairwise sequence alignment can be solved using exact techniques
  - Approach to deal with multiple alignment sequence (MSA):
    - Metaheuristics again

# MSA as an multi-objective optimization problem

- There are many methods to evaluate the quality of an alignment:

### **Only nucleotide or aminoAcid information**

- 1. Sum-Of-Pairs
  - 2. Sum-of-pairs with weights (wSOP)
  - 3. Percentage of Non-gaps
  - 4. Percentage of Aligned Columns
  - 5. COFFEE (Consistency based Objective Function For alignment Evaluation) **Based on homologies or Protein Structure**
  - 6. Strike (Single sTRucture Induced Evaluation)
  - 7. Entropy
  - 8. MetAl Metric

*Global Objective Score =  $\sum_{i=1}^n \sum_{j=i}^m$*

$$Sum-of-Pairs = \sum_{i=1}^{n-1} \sum_{j=i+1}^n ScoringMatrix(l_i, l_j)$$

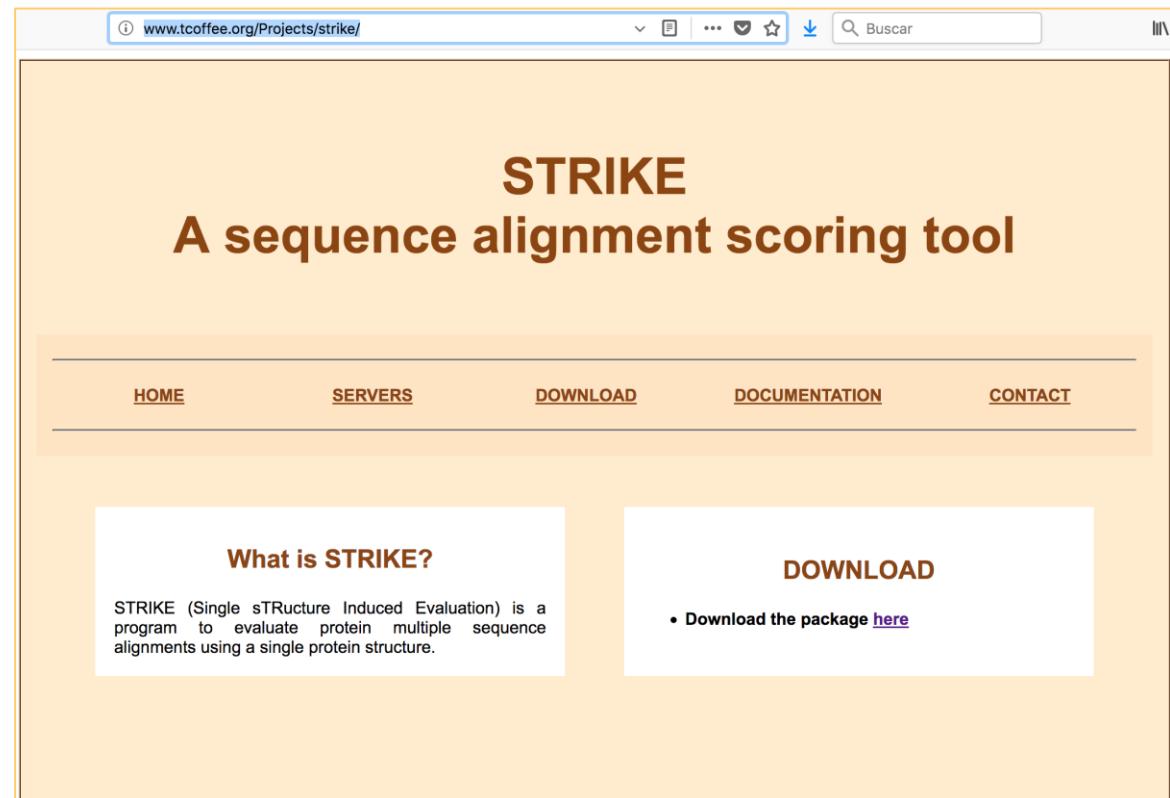
$$NonGaps(S) = 100 \sum_{i=1}^k \sum_{j=1}^L \frac{isNotGap(s_{ij})}{k * L}$$

$$Global\ Objective\ Score = \frac{\sum_{i=1}^{N-1} \sum_{j=i+1}^N W_{ij} * Score(A_{ij})}{\sum_{i=1}^{N-1} \sum_{j=i+1}^N W_{ij} * Len},$$

**MSA can be formulated as a multi-objective optimization problem**

# MSA as an multi-objective optimization problem

- STRIKE
  - Web site: <http://www.tcoffee.org/Projects/strike/>



## MSA as an multi-objective optimization problem

- Question: Which objectives should be optimized?
  - Do we have structural information of the problem?
    - If so, consider STRIKE
  - Minimizing the percentage of non-gaps and maximizing the percentage of aligned columns should be intuitively chosen always, but there is an important matter:
    - Both functions have a search space with many plateaus:

s1: G-----ERSLAA--TLV-  
 s2: NAILAH-ER-----LSI  
 s3: NGYLFI-E---Q---L-N  
 s4: GLVSDVFEARH--MQRL--

s1: --G----ERSLAA--TLV-  
 s2: NAI-LAHER-----LSI  
 s3: -NGYLFIE---Q---LN-  
 s4: GLVSDVFE-ARH-MQRL--

%Aligned columns: 10.526

%Non-gaps: 63.1579

- So at least a third objective is required

# Related work

BIOINFORMATICS

ORIGINAL PAPER

Vol. 29 no. 17 2013, pages 2112–2121  
doi:10.1093/bioinformatics/btt360

Sequence analysis

Advance Access publication June 21, 2013

## Optimizing multiple sequence alignments using a genetic algorithm based on three objectives: structural information, non-gaps percentage and totally conserved columns

Francisco M. Ortúñoz<sup>1,\*</sup>, Olga Valenzuela<sup>2</sup>, Fernando Rojas<sup>1</sup>, Hector Pomares<sup>1</sup>, Javier P. Florido<sup>3</sup>, Jose M. Urquiza<sup>4</sup> and Ignacio Rojas<sup>1</sup>

<sup>1</sup>Department of Computer Architecture and Computer Technology, CITIC-UGR, <sup>2</sup>Department of Applied Mathematics, University of Granada, 18071 Granada, Spain, <sup>3</sup>Bioinformatics Department, Genomics and Bioinformatics Platform of Andalusia (GBPA), 41092 Seville, Spain and <sup>4</sup>Chromatin and Disease Group, Bellvitge Biomedical Research Institute (IDIBELL), L'Hospitalet, Barcelona 08907, Spain

Associate Editor: Alfonso Valencia

### ABSTRACT

**Motivation:** Multiple sequence alignments (MSAs) are widely used approaches in bioinformatics to carry out other tasks such as structure predictions, biological function analyses or phylogenetic modeling. However, current tools usually provide partially optimal alignments, as each one is focused on specific biological features. Thus, the same set of sequences can produce different alignments, above all when sequences are less similar. Consequently, researchers and biologists do not agree about which is the most suitable way to evaluate MSAs. Recent evaluations tend to use more complex scores including further biological features. Among them, 3D structures are increasingly being used to evaluate alignments. Because structures are more conserved in proteins than sequences, scores with structural

Received on October 10, 2012; revised on May 24, 2013; accepted on June 18, 2013

### 1 INTRODUCTION

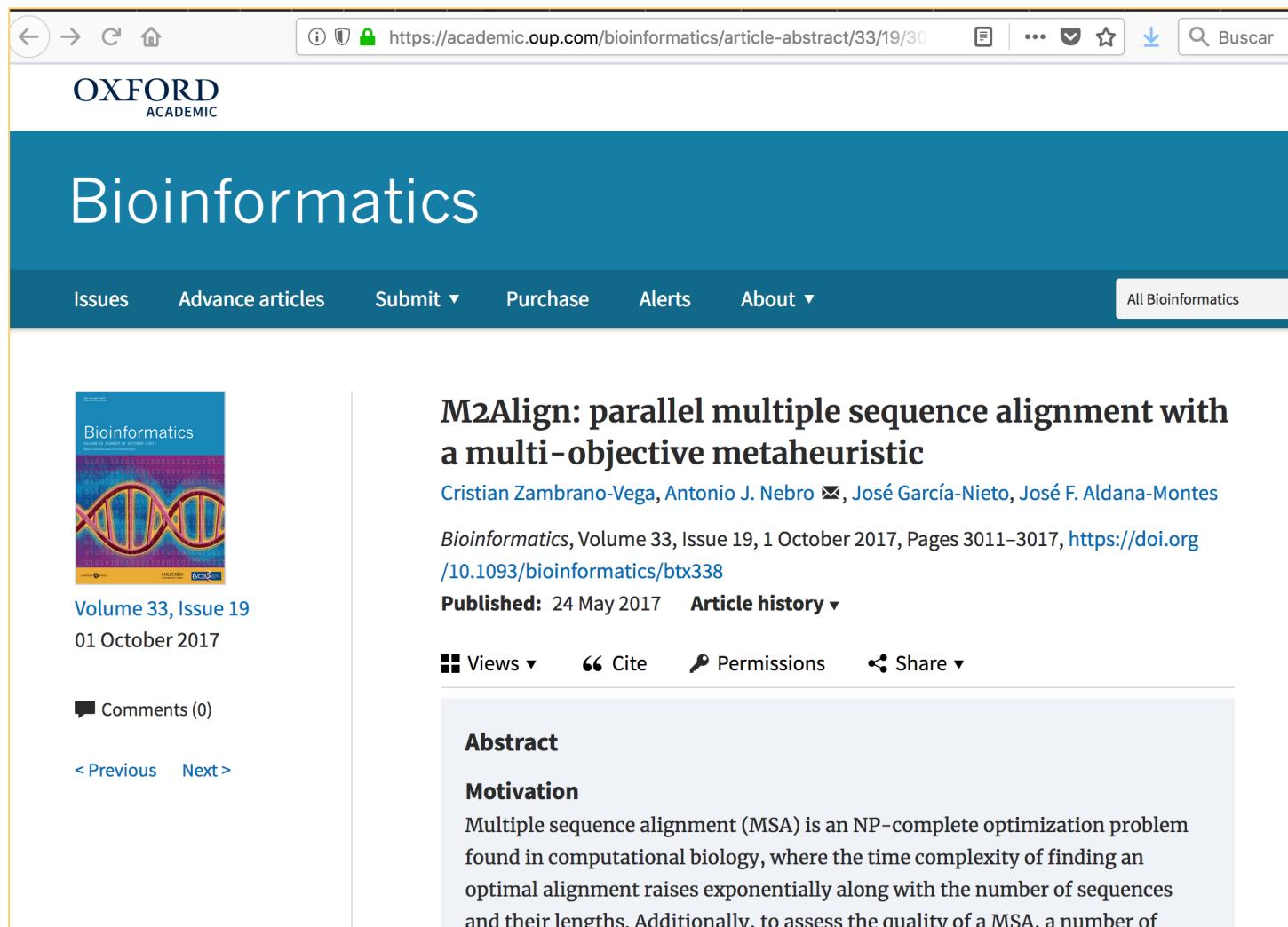
Multiple sequence alignments (MSAs) are widely used strategies in current molecular biology. These approaches are often used for homology transfer (Doolittle, 1981; Fitch, 1966), where poorly characterized sequences are compared with well-studied homologs from typical model organisms. MSA strategies have traditionally been applied to researches in phylogenetic analyses, structural modeling, functional predictions or sequence database searching (Bacon and Anderson, 1986). MSA tools have also

Algorithm: MO-SAStrE

Three objectives:

- % gaps
- % aligned columns
- STRIKE

# M2Align



The screenshot shows a web browser displaying an article from the journal *Bioinformatics*. The URL in the address bar is <https://academic.oup.com/bioinformatics/article-abstract/33/19/30>. The page header includes the Oxford Academic logo and the journal title "Bioinformatics". The main content area features the article title "M2Align: parallel multiple sequence alignment with a multi-objective metaheuristic" by Cristian Zambrano-Vega, Antonio J. Nebro, José García-Nieto, and José F. Aldana-Montes. It also shows the publication details: "Bioinformatics, Volume 33, Issue 19, 1 October 2017, Pages 3011–3017, <https://doi.org/10.1093/bioinformatics/btx338>". Below the title, there are links for "Views", "Cite", "Permissions", and "Share". The abstract and motivation sections are partially visible at the bottom.

**Bioinformatics**

Issues   Advance articles   Submit ▾   Purchase   Alerts   About ▾   All Bioinformatics

**M2Align: parallel multiple sequence alignment with a multi-objective metaheuristic**

Cristian Zambrano-Vega, Antonio J. Nebro, José García-Nieto, José F. Aldana-Montes

*Bioinformatics*, Volume 33, Issue 19, 1 October 2017, Pages 3011–3017, <https://doi.org/10.1093/bioinformatics/btx338>

Published: 24 May 2017   Article history ▾

Views ▾   Cite   Permissions   Share ▾

**Abstract**

**Motivation**

Multiple sequence alignment (MSA) is an NP-complete optimization problem found in computational biology, where the time complexity of finding an optimal alignment raises exponentially along with the number of sequences and their lengths. Additionally, to assess the quality of a MSA, a number of

# M2Align

As the original MO-SAStrE, our algorithm has the following characteristics:

- It is based on NSGA-II.
- The evolutionary variation operators are single point crossover and closed gap shifting.
- The objectives to optimize are non-gaps percentage, totally conserved columns and STRIKE.
- The initial population of NSGA-II is filled with pre-alignments obtained with representative MSA tools: ClustalW, MUSCLE, Kalign, Mafft, RetAlign, TCOFFEE, ProbCons and FSA.

Unlike the original MO-SAStrE, our solution incorporates the following features:

- M2Align is written in Java (MO-SAStrE is implemented in Matlab).
- The algorithm in M2Align can be executed in parallel on multi-core systems.
- The solution encoding based on gap information.
- If PDB structures are not available, M2Align provides SOP and WSP scores as alternatives to STRIKE.
- M2Align is an Open Source Project hosted in GitHub

<https://github.com/KhaosResearch/M2Align>

## M2Align

- To promote a faster convergence
  - An approach is to generate the initial population with pre-computed alignments

Table 1. Methods used to generate the initial population of the algorithms. These eight tools are applied to build initial multiple sequence alignments for the BALiBASE datasets.

Tool	Version	Type
ClustalW (Thompson <i>et al.</i> , 1994)	2.1	Progressive
MUSCLE (Edgar, 2004)	3.8.31	Progressive
Kalign (Lassmann and Sonnhammer, 2005)	2.04	Progressive
Mafft (Katoh <i>et al.</i> , 2002)	7.245	Progressive
RetAlign (Novák <i>et al.</i> , 2010)	1.0	Progressive
TCOFFEE (Notredame <i>et al.</i> , 2000)	11.00	Consistency-based
ProbCons (Do <i>et al.</i> , 2005)	1.12	Consistency-based
FSA (Roberts <i>et al.</i> , 2009)	1.15.9	Consistency-based

## M2Align

- Solution encoding
  - The used representation only stores the positions (begin, end) of groups of gaps

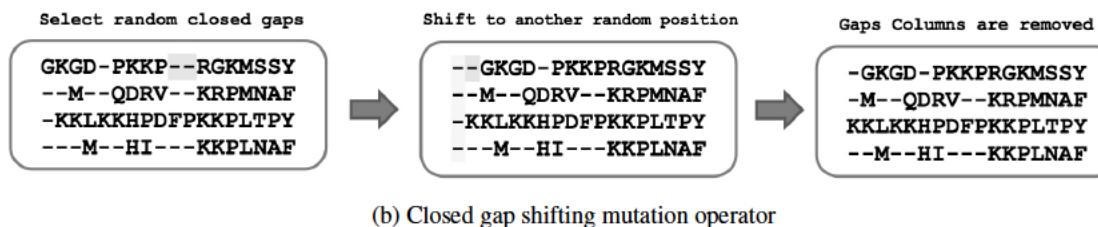
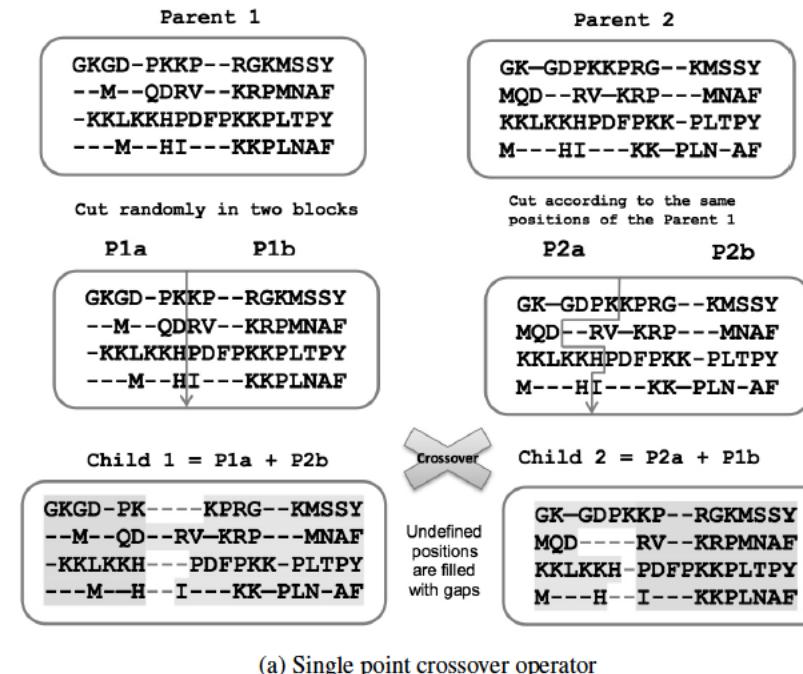
s1: GKGD-PKKP--RGKMSSY  
s2: --M--QDRV--KRPVMNAF  
s3: -KKLKKHPDFPKKPLTPY  
s4: ---M--HI---KKPLNAF



s'1: (5,5), (10,11)  
s'2: (1,2), (4,5), (10,11)  
s'3: (1,1)  
s'4: (1,3), (5,6), (9,11)

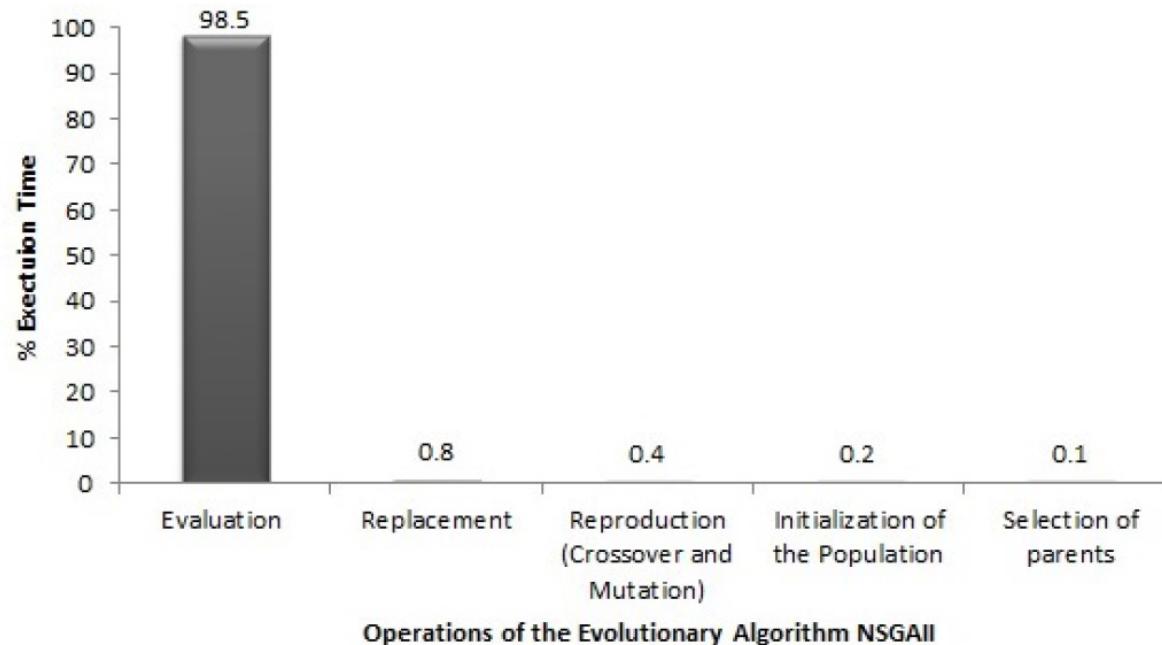
# M2Align

- Variation operators



## M2Align

- Parallel justification
  - Profiling NSGA-II



## M2Align

- Parallel scheme
  - Evaluating the solutions in parallel
  - The behavior of the algorithm does not change
- Performance assessment
  - BALiBASE datasets
    - Six subsets having different biological features
    - Summing up to 218 problems
  - Multicore system composed of 20 cores

# M2Align

- Parallel performance

Table 2. Parallel performance evaluation of M2Align (time units are hours) over 218 instances of the BALiBASE v3.0.  $T_1$ : Sequential runtime,  $Sp$ : Speed-up (T<sub>1</sub> divided number of cores), ( $Ef$ ): Efficiency ( $Sp$  divided by the number of cores).

Family	$T_1$	4 cores			10 cores			20 cores		
		$T_4$	$Sp$	$Ef$	$T_{10}$	$Sp$	$Ef$	$T_{20}$	$Sp$	$Ef$
RV11	9.05	2.54	3.56	89%	1.27	7.15	72%	0.99	9.12	46%
RV12	4.58	1.34	3.43	86%	0.73	6.29	63%	0.63	7.22	36%
RV20	7.78	2.30	3.38	84%	1.20	6.49	65%	0.94	8.25	41%
RV30	6.55	1.88	3.48	87%	0.99	6.61	66%	0.79	8.28	41%
RV40	9.14	2.70	3.39	85%	1.50	6.11	61%	1.25	7.33	37%
RV50	5.19	1.46	3.54	89%	0.74	7.03	70%	0.59	8.77	44%

Table 3. Execution Time in minutes) of the sequential version of M2Align against the original version of MO-SAStrE when solving nine BALiBASE datasets.

Instance	Sequential Runtime (Mins)	
	MOSAStrE	Sequential M2Align
<b>BB11001</b>	24.02	0.44
<b>BB11009</b>	37.50	4.09
<b>BB11011</b>	80.96	3.49
<b>BB11013</b>	31.41	0.41
<b>BB12002</b>	85.14	0.76
<b>BB12004</b>	172.18	2.73
<b>BB12010</b>	183.04	3.92
<b>BB12015</b>	142.65	1.80
<b>BB30009</b>	294.24	0.55

# M2Align

- Numerical performance

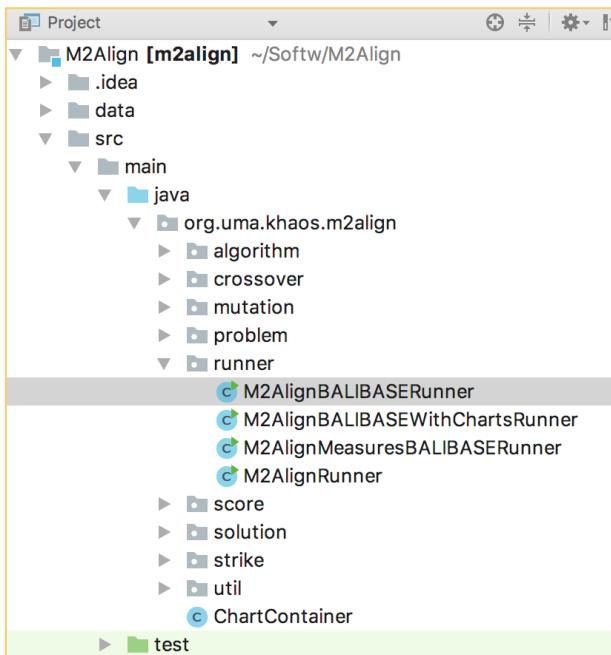
Table 4. Average scores for the 218 BALiBASE problems optimized by M2Align and 9 classical MSA techniques (the values of MO-SAStrE have been taken from (Ortuño et al., 2013)).

Method	M2Align Objectives			BALiBASE Scores	
	STRIKE	TCC (%)	Non-Gaps (%)	SP	TC
ClustalW	1.54	1.70	55.39	0.67	0.29
Muscle	1.76	1.90	52.48	0.72	0.36
Kalign	1.75	1.85	48.04	0.73	0.36
RetAlign	1.72	2.10	49.38	0.71	0.33
Tcoffee	1.75	1.87	45.35	0.77	0.41
ProbCons	1.74	1.85	44.21	0.77	0.42
3D-Coffee	1.80	1.64	42.27	0.72	0.39
Mafft	1.80	1.97	49.70	0.77	0.43
FSA	1.37	1.40	31.36	0.68	0.32
BALiBASE	1.79	1.94	52.16	1.00	1.00
MO-SAStrE	2.37	2.44	<b>58.51</b>	0.79	-
<b>M2Align</b>	<b>2.44</b>	<b>2.45</b>	58.13	<b>0.81</b>	<b>0.46</b>

## M2Align

- Discussion
  - Important speed-ups up with up to 20 cores
    - But the algorithm doesn't scale with the number of cores
  - Solutions may have different lengths
    - Their evaluation time is not same
    - It is difficult to have all the cores busy the 100% of the time
  - The number of gaps in the MSA has a strong influence in the performance of M2Align

# Example 1: configuring and running M2Align (BALiBASE)



```

public static void main(String[] args) throws Exception {
    BALiBASEMSAProblem problem;
    Algorithm<List<MSASolution>> algorithm;
    CrossoverOperator<MSASolution> crossover;
    MutationOperator<MSASolution> mutation;
    SelectionOperator<List<MSASolution>, MSASolution> selection;

    if (args.length != 5) {
        throw new JMetalException("Wrong number of arguments");
    }
    String instance = args[0];
    String dataDirectory = args[1];
    Integer maxEvaluations = Integer.parseInt(args[2]);
    Integer populationSize = Integer.parseInt(args[3]);
    Integer numberOfCores = Integer.parseInt(args[4]);

    crossover = new SPXMSACrossover(0.8);
    mutation = new ShiftClosedGapsMSAMutation(0.2);
    selection = new BinaryTournamentSelection<(new RankingAndCrowdingDistanceComparator<>()>);

    List<Score> scoreList = new ArrayList<()>;
    StrikeScore objStrike = new StrikeScore();
    scoreList.add(new StrikeScore());
    scoreList.add(new PercentageOfAlignedColumnsScore());
    scoreList.add(new PercentageOfNonGapsScore());

    problem = new BALiBASEMSAProblem(instance, dataDirectory, scoreList);
    objStrike.initializeParameters(problem.PDBPath, problem.getListOfSequenceNames());

    SolutionListEvaluator<MSASolution> evaluator;

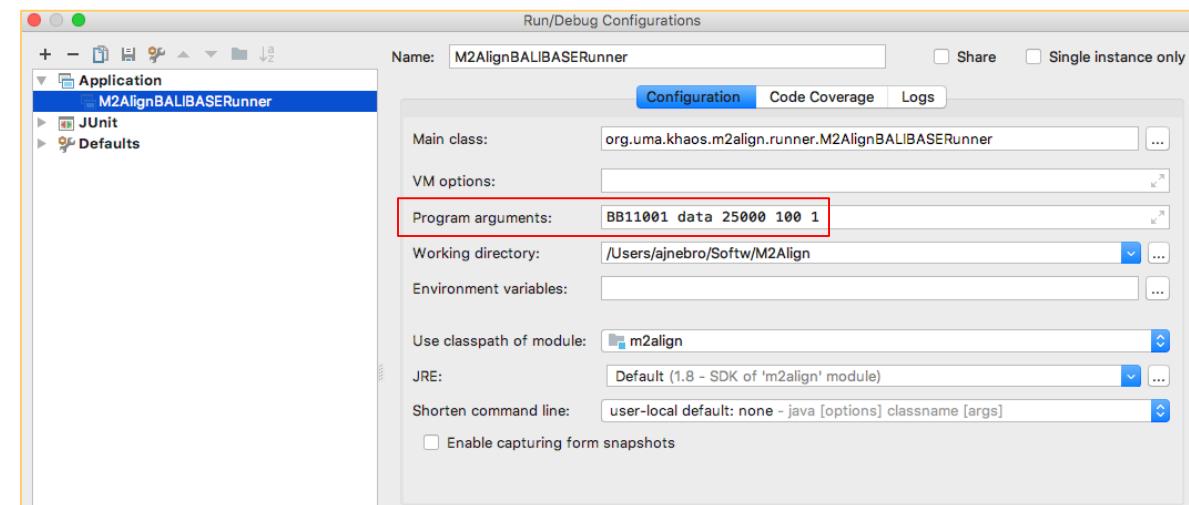
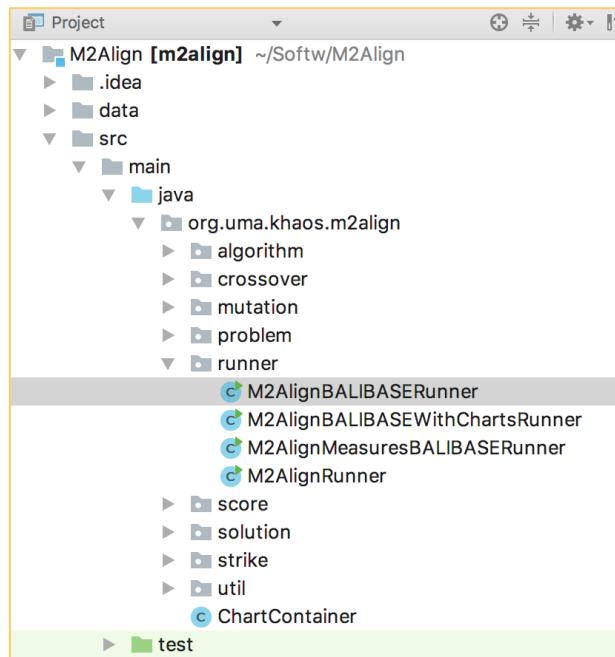
    if (numberOfCores == 1) {
        evaluator = new SequentialSolutionListEvaluator<()>;
    } else {
        evaluator = new MultithreadedSolutionListEvaluator<(numberOfCores, problem)>;
    }

    algorithm = new M2AlignBuilder(problem, crossover, mutation)
        .setSelectionOperator(selection)
        .setMaxEvaluations(maxEvaluations)
        .setPopulationSize(populationSize)
        .setSolutionListEvaluator(evaluator)
        .build();

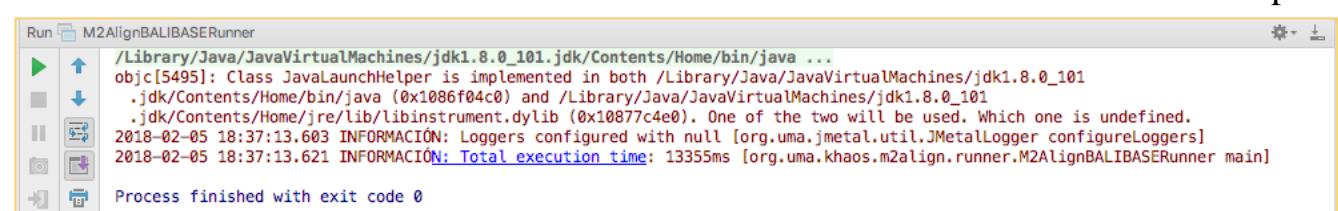
    AlgorithmRunner algorithmRunner = new AlgorithmRunner.Executor(algorithm)
        .execute();
}

```

# Example 1: configuring and running M2Align (BALiBASE)



Program arguments



Output

# Example 1: configuring and running M2Align (BALiBASE)

Terminal

```
+ MacBook-Pro-de-Antonio:M2Align ajnebro$ ls
x FUN.BB11001.tsv      VAR           data          m2align.iml      release.properties
LICENSE.txt        VAR.BB11001.tsv   jMetal.log    pom.xml       src
README.md          chart_Front.png jMetal.log.1  pom.xml.releaseBackup target
MacBook-Pro-de-Antonio:M2Align ajnebro$
```

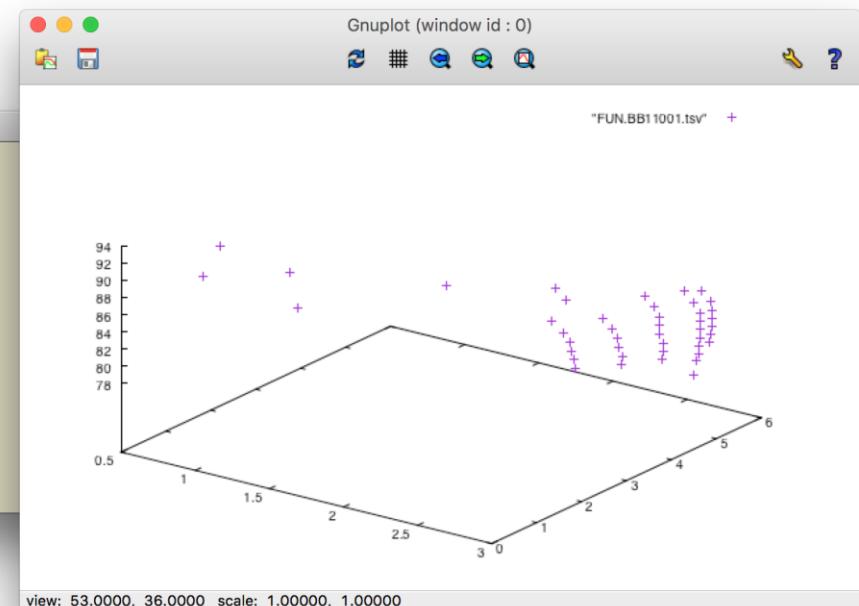
Program results

```
ajnebro — gnuplot-run.sh — gnuplot — 80x13
Version 5.0 patchlevel 3      last modified 2016-02-21

Copyright (C) 1986-1993, 1998, 2004, 2007-2016
Thomas Williams, Colin Kelley and many others

gnuplot home:      http://www.gnuplot.info
faq, bugs, etc:    type "help FAQ"
immediate help:    type "help"  (plot window: hit 'h')

Terminal type set to 'wxt'
[gnuplot> cd "Softw/M2Align/
[gnuplot> splot "FUN.BB11001.tsv"
gnuplot>
```



## Example 2: configuring and running M2Align (BALiBASE)

- Parallel performance (instance: BB11001)

Cores	Time
1	12.634 secs
2	5.996 secs
4	4.815 secs
8	4.783 secs

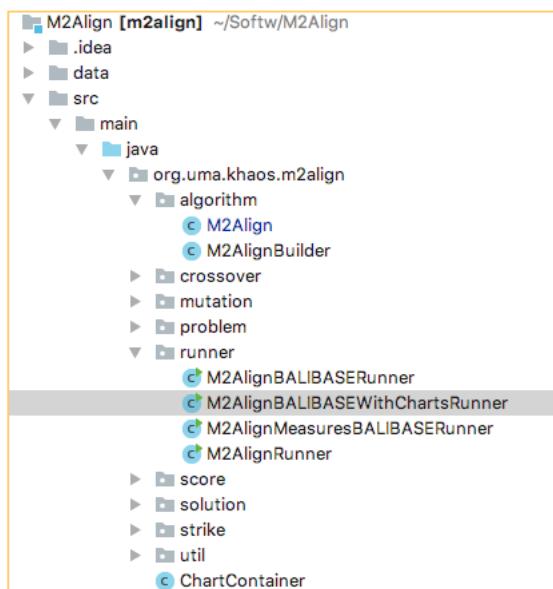
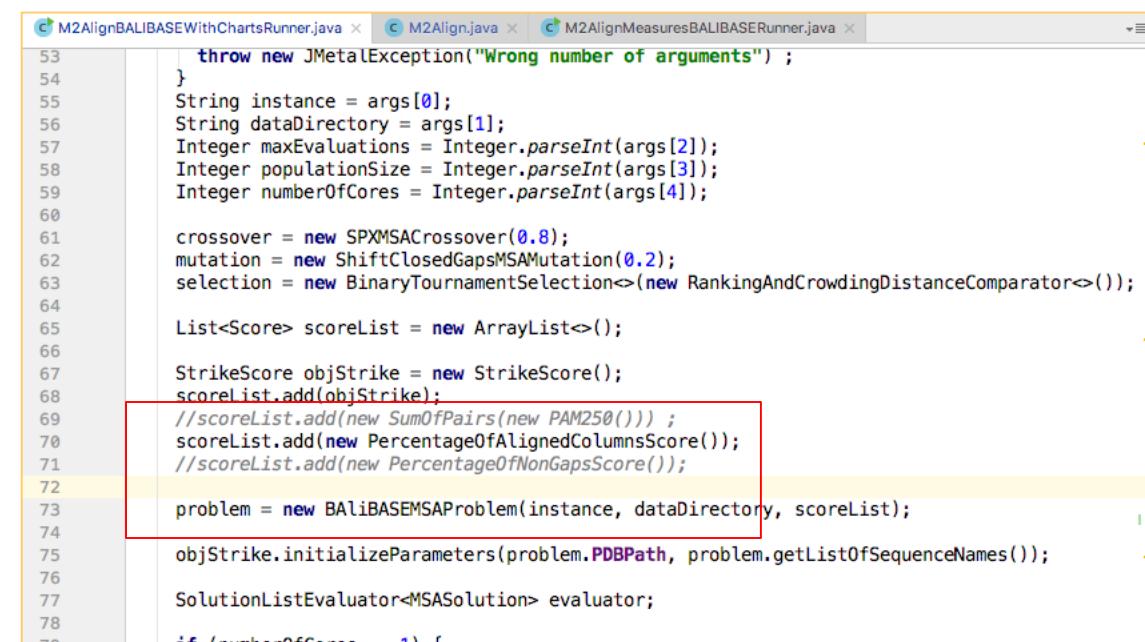


Cores	Time
1	11.171 secs
2	4.619 secs
4	3.713 secs
8	3.460 secs

Elemento	Valor
Nombre del SO	Microsoft Windows 10 Pro
Versión	10.0.16299 compilación 16299
Descripción adicional del SO	No disponible
Fabricante del SO	Microsoft Corporation
Nombre del sistema	AJNEBRO
Fabricante del sistema	LENOVO
Modelo del sistema	30ATS0Y300
Tipo de sistema	PC basado en x64
SKU del sistema	LENOVO_MT_30AT_BU_LENOVO_FM_ThinkStation P310
Procesador	Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz, 3408 Mhz, 4 procesadores princip...

## Example 3: configuring and running M2Align (BALiBASE)

- Version with charts

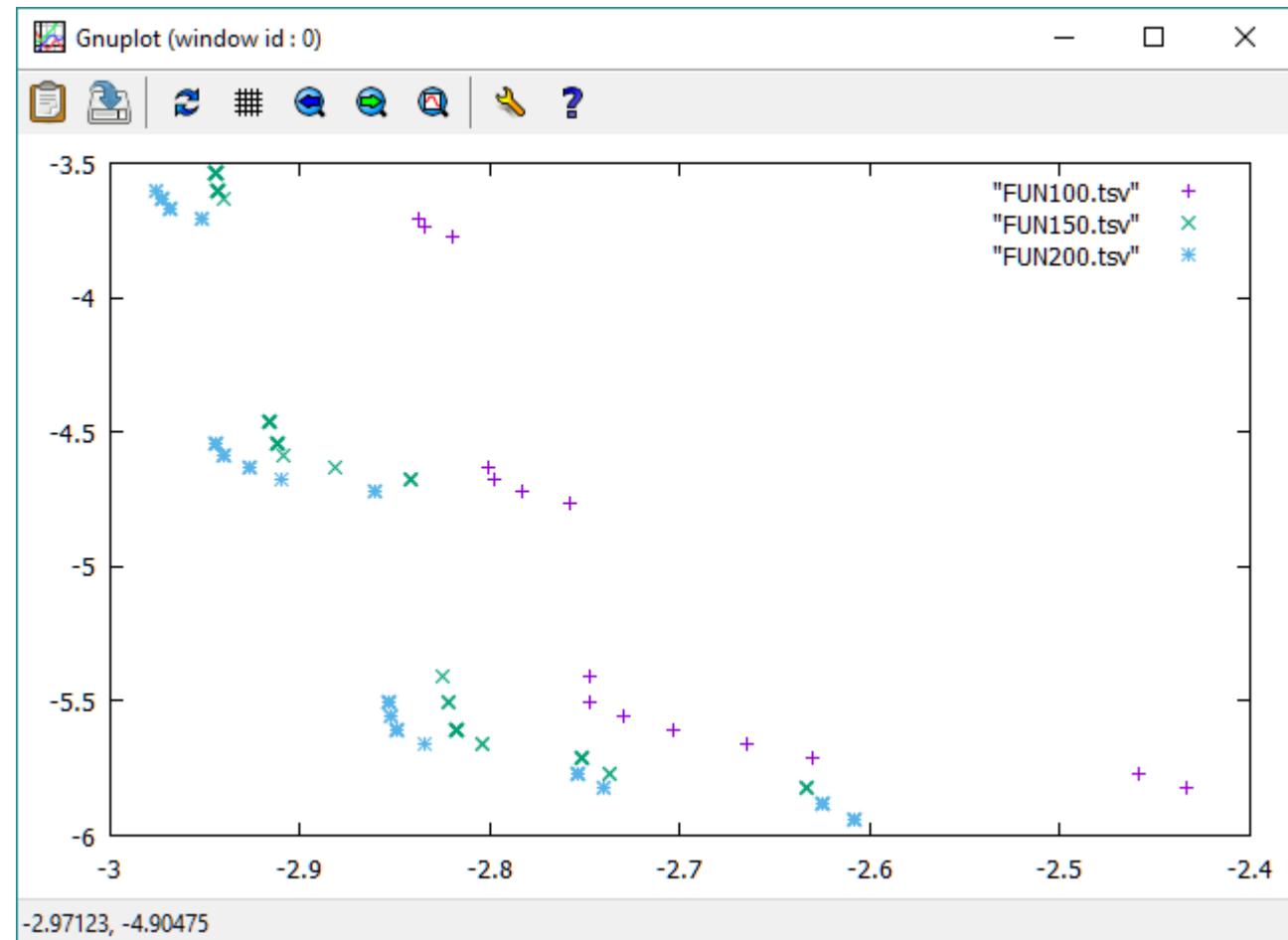
```

53     throw new JMetalException("Wrong number of arguments");
54 }
55 String instance = args[0];
56 String dataDirectory = args[1];
57 Integer maxEvaluations = Integer.parseInt(args[2]);
58 Integer populationSize = Integer.parseInt(args[3]);
59 Integer numberOfCores = Integer.parseInt(args[4]);
60
61 crossover = new SPXMSACrossover(0.8);
62 mutation = new ShiftClosedGapsMSAMutation(0.2);
63 selection = new BinaryTournamentSelection<>(new RankingAndCrowdingDistanceComparitor<>());
64
65 List<Score> scoreList = new ArrayList<>();
66
67 StrikeScore objStrike = new StrikeScore();
68 scoreList.add(objStrike);
69 //scoreList.add(new SumOfPairs(new PAM250()));
70 scoreList.add(new PercentageOfAlignedColumnsScore());
71 //scoreList.add(new PercentageOfNonGapsScore());
72
73 problem = new BALiBASEMSAProblem(instance, dataDirectory, scoreList);
74
75 objStrike.initializeParameters(problem.PDBPath, problem.getListOfSequenceNames());
76
77 SolutionListEvaluator<MSASolution> evaluator;
78
79 if (numberOfCores == 1) {

```

## Example 4: configuring and running M2Align (BALiBASE)

- Executing M2AlignMeasuresBALIBASERunner



## Table of contents

- About this tutorial
- Metaheuristics as optimization techniques
- Dealing with problems with more than one objective function
- Case study: multiple sequence alignment
- **Further developments and related projects**

## Further developments related to MSA

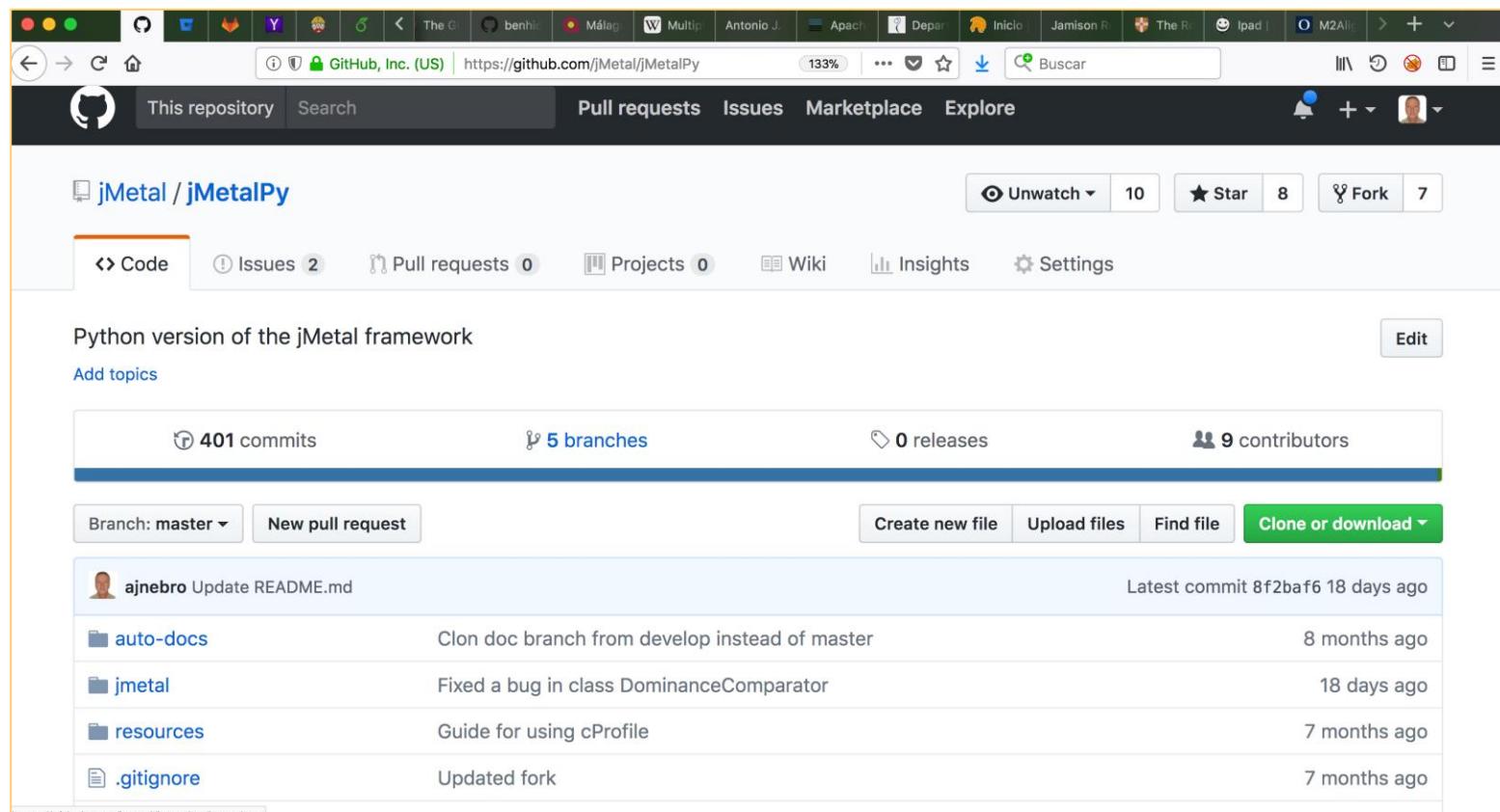
- To improve the search capabilities of the algorithm
  - Design crossover and mutation operators with problem knowledge
  - Design of efficient local search strategies
- To reduce the computing time
  - Design of efficient local search strategies
  - Use an different parallel programming model that scales better

## Further developments related to MSA

- We have not considered the MSA from the point of view of a biologist:
  - We have tried to find the best Pareto front approximations from a quantitative point of view (in terms of convergence and diversity)
  - But we have not taken into account qualitative issues
- Some ideas:
  - Introduce user preferences
  - Use interactive methods to guide the search to the interest region of the front

## Related projects

- We started a Python version of jMetal in 2017: jMetalPy



## Related projects

- pyMSA: *Scoring Multiple Sequence Alignments with Python*



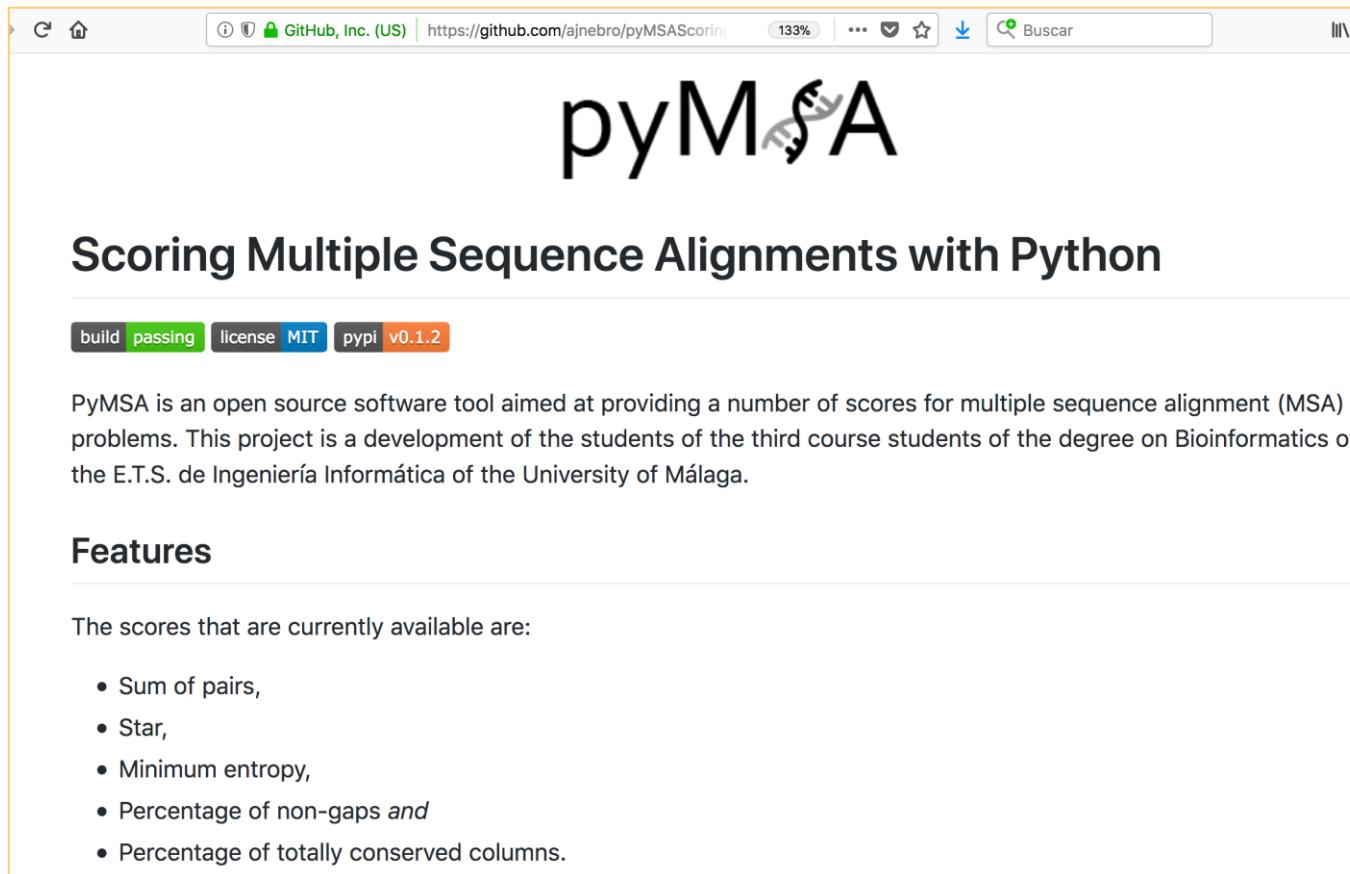
The screenshot shows the PyPI (Python Package Index) page for the pyMSA package. The URL is https://pypi.python.org/pypi/pyMSA/0.1.1. The page title is "pyMSA 0.1.1" and the subtitle is "Scoring Multiple Sequence Alignments with Python". A green "Downloads ↓" button is visible. On the left, there's a sidebar with links like "PACKAGE INDEX", "Browse packages", "RSS (latest 40 updates)", and "RSS (newest 40 packages)". Below that is a "ABOUT" section with links to "NEWS", "DOCUMENTATION", "DOWNLOAD", "COMMUNITY", "FOUNDATION", and "CORE DEVELOPMENT". The main content area shows a table of available files:

File	Type	Py Version
<a href="#">pyMSA-0.1.1-py3-none-any.whl (md5)</a>	Python Wheel	py3
<a href="#">pyMSA-0.1.1.tar.gz (md5)</a>	Source	

Below the table, there are sections for "Author", "Home Page", "License", and "Categories".

## Related projects

- pyMSA: *Scoring Multiple Sequence Alignments with Python*



The screenshot shows a browser window displaying the GitHub page for the PyMSA project. The title of the page is "pyMSA" with a DNA helix icon. Below the title, the subtitle "Scoring Multiple Sequence Alignments with Python" is displayed. At the top of the page, there are status indicators for build (passing), license (MIT), and pypi (v0.1.2). The main text describes PyMSA as an open source software tool for multiple sequence alignment scoring, developed by students of the University of Málaga. A section titled "Features" lists several scoring methods: Sum of pairs, Star, Minimum entropy, Percentage of non-gaps, and Percentage of totally conserved columns.

PyMSA is an open source software tool aimed at providing a number of scores for multiple sequence alignment (MSA) problems. This project is a development of the students of the third course students of the degree on Bioinformatics of the E.T.S. de Ingeniería Informática of the University of Málaga.

### Features

The scores that are currently available are:

- Sum of pairs,
- Star,
- Minimum entropy,
- Percentage of non-gaps *and*
- Percentage of totally conserved columns.

Thanks for your attention ☺

- Comments and suggestions are welcome

