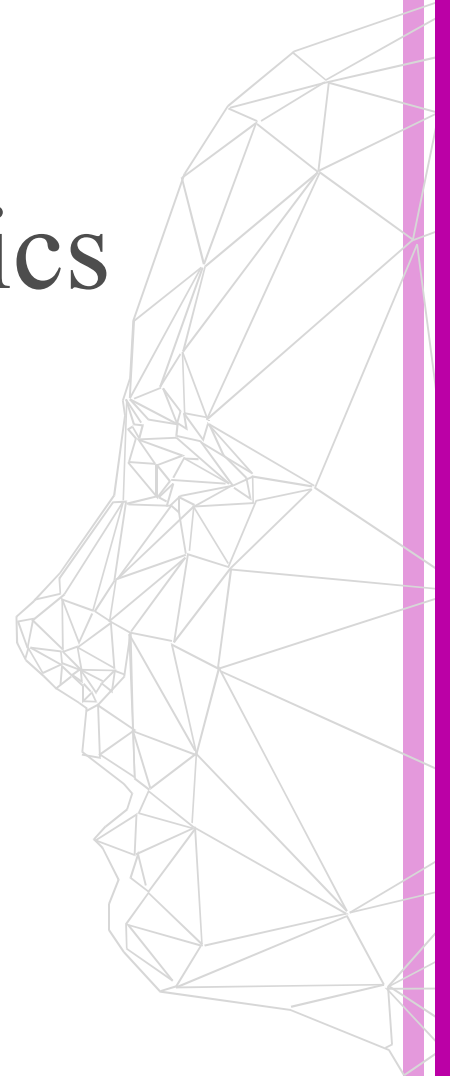
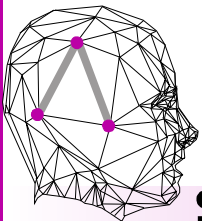


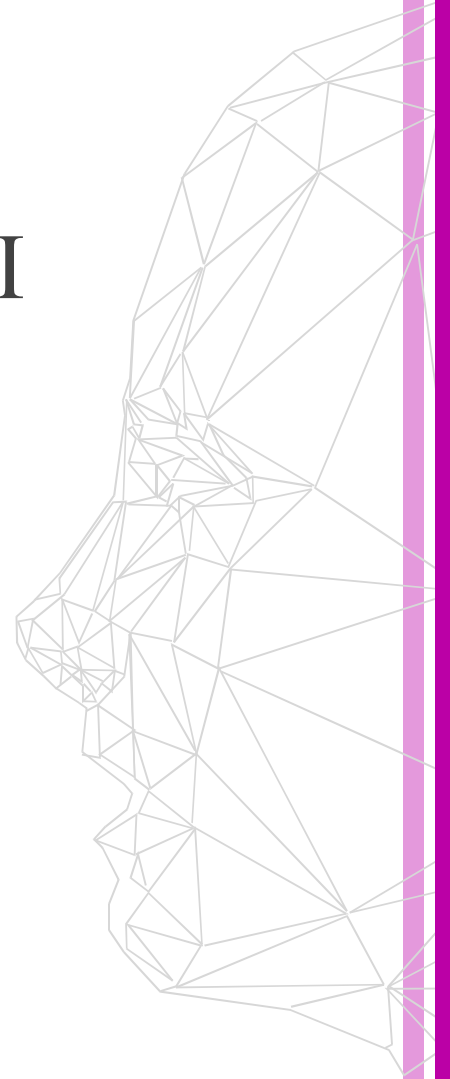
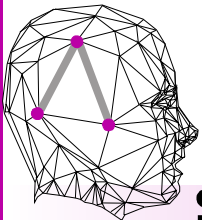
LiO: Tool for metaheuristics

Luis de la Ossa
Juan Luis Mateo

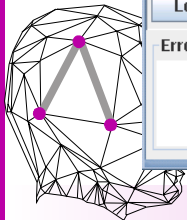
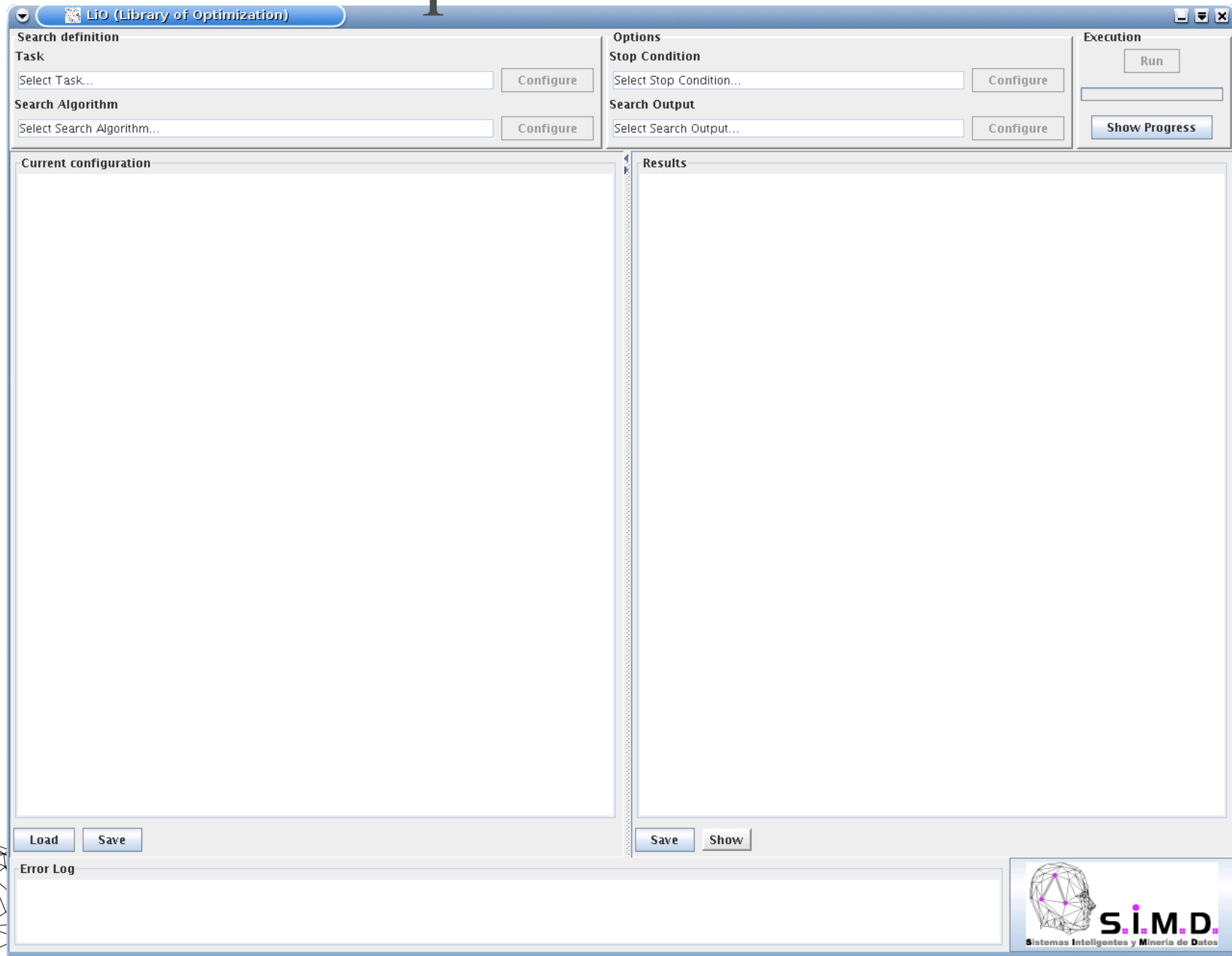


Index

- LiO graphical user interface: LiOGUI
- Configuration files
- Command line interface



Graphical interface

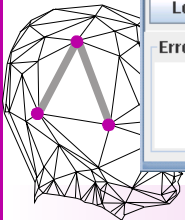


Task/Problem selection

The screenshot displays the LIO (Library of Optimization) application window. The interface is divided into several sections:

- Search definition:** Contains a "Task" section with a "Select Task..." dropdown menu and a tree view. The tree view shows a hierarchy: "Tasks" > "problems" > "bitchain" > "Ackley", "CheckerBoard", "ColMlle", "DecomposableProblem", "EqualProducts", "FC2", and "FC3".
- Options:** Includes "Stop Condition" and "Search Output" sections, each with a text input field and a "Configure" button.
- Execution:** Features a "Run" button and a "Show Progress" button.
- Results:** A large empty area for displaying search results.
- Bottom Panel:** Contains "Load" and "Save" buttons on the left, and "Save" and "Show" buttons on the right. Below these is an "Error Log" section.

The title bar of the window reads "LIO (Library of Optimization)".



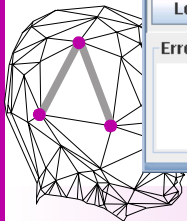
Algorithm selection

The screenshot displays the LIO (Library of Optimization) application window. The interface is divided into several sections:

- Search definition:** Contains a "Task" field with the value "problems.bitchain.OneMax" and a "Configure" button.
- Search Algorithm:** Features a "Select Search Algorithm..." dropdown menu and a "Configure" button. Below this is a tree view of search algorithms:
 - Search Algorithms
 - lio
 - search
 - genetic
 - CHC
 - StdGeneticAlgorithm
 - local
 - probabilistic
 - Custom...

- Options:** Includes "Stop Condition" and "Search Output" sections, each with a dropdown menu and a "Configure" button.
- Execution:** Contains a "Run" button and a "Show Progress" button.
- Results:** A large empty area for displaying search results.
- Bottom Panel:** Includes "Load" and "Save" buttons on the left, and "Save" and "Show" buttons on the right. Below these is an "Error Log" field.

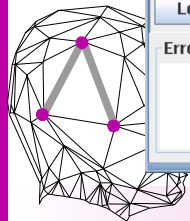

The LIO logo is visible in the bottom right corner of the application window, featuring a stylized head with a network structure and the text "SIMD. Sistemas Inteligentes y Minería de Datos".



Execution configuration

LIO (Library of Optimization)

Search definition	Options	Execution
Task problems.bitchain.OneMax <input type="button" value="Configure"/>	Stop Condition lio.misc.StopCondition <input type="button" value="Configure"/>	<input type="button" value="Run"/>
Search Algorithm lio.search.genetic.StdGeneticAlgorithm <input type="button" value="Configure"/>	Search Output lio.misc.SearchOutput <input type="button" value="Configure"/>	<input type="button" value="Show Progress"/>
Current configuration Kind of Individual: lio.individuals.BitChain Individual size: 100 task=problems.bitchain.OneMax size=100 probMutation=0.05 populationSize=200 probCrossover=0.6 selector=lio.selectors.RouletteWheelSelector ranking=false replacer=lio.replacement.ElitistReplacement generator=lio.generators.bitchain.RandomGenerator mutation=lio.mutation.bitchain.BinaryMutation crossover=lio.crossover.bitchain.OnePointCrossover point=0	Results	
<input type="button" value="Load"/> <input type="button" value="Save"/>	<input type="button" value="Save"/> <input type="button" value="Show"/>	
Error Log		



Execution results

The screenshot displays the LIO (Library of Optimization) software interface. The window title is "LIO (Library of Optimization)".

Search definition

- Task:** problems.bitchain.OneMax [Configure]
- Search Algorithm:** lio.search.genetic.StdGeneticAlgorithm [Configure]

Options

- Stop Condition:** lio.misc.StopCondition [Configure]
- Search Output:** lio.misc.SearchOutput [Configure]

Execution

- [Run]
- [Show Progress]

Current configuration

```
Kind of Individual: lio.individuals.BitChain
Individual size: 100
task=problems.bitchain.OneMax
  size=100

probMutation=0.05
populationSize=200
probCrossover=0.6
selector=lio.selectors.RouletteWheelSelector
  ranking=false
replacer=lio.replacement.ElitistReplacement
generator=lio.generators.bitchain.RandomGenerator
mutation=lio.mutation.bitchain.BinaryMutation
crossover=lio.crossover.bitchain.OnePointCrossover
  point=0
```

Results

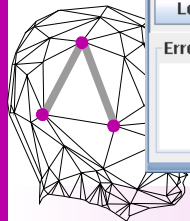
```
-----
02 October 2006 20:00:39

Results for the lio.search.genetic.StdGeneticAlgorithm algorithm:
Task to solve:problems.bitchain.OneMax
  Best fitness: 99.0
  Number of iterations to best: 141.0
  Number of iterations: 155.0
  Number of evaluations to best: 28432.0
  Number of evaluations: 31400.0
  Time to best: 935.0 ms
  Total time: 1017.0 ms
-----
```

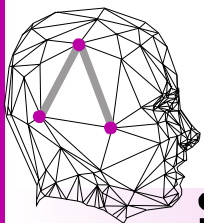
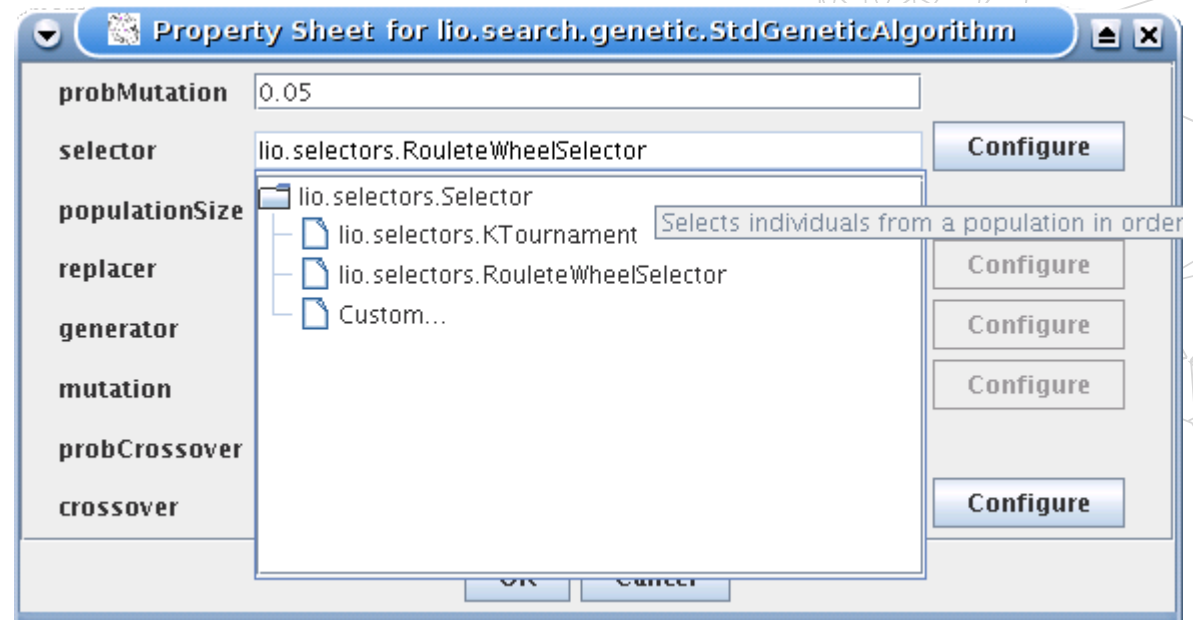
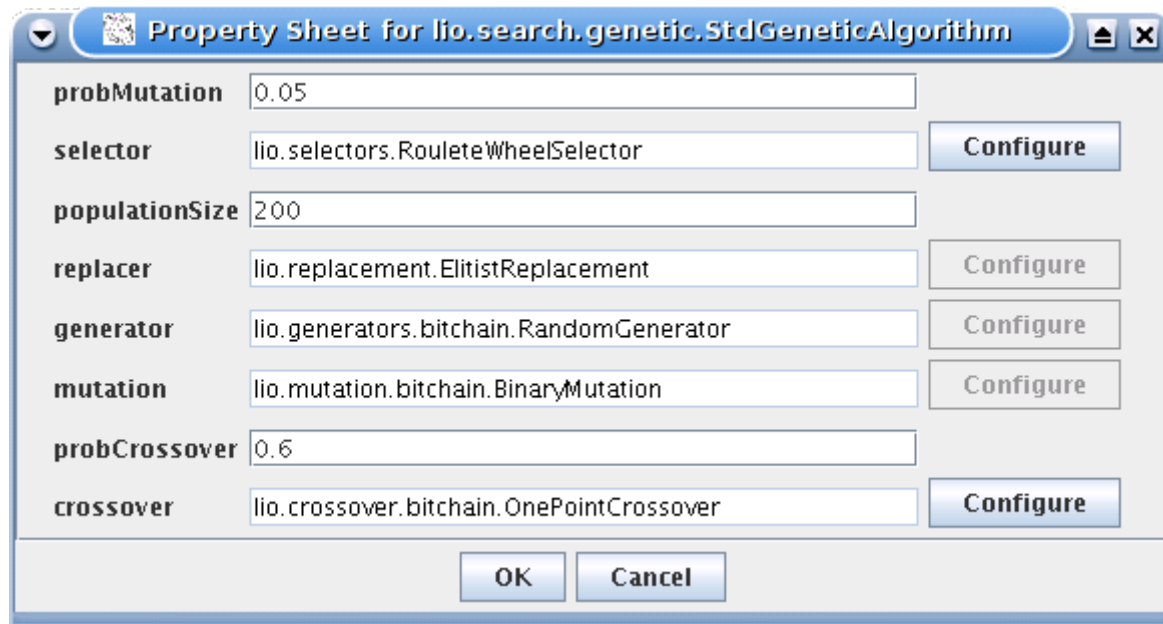
Buttons: Load, Save, Save, Show

Error Log: [Empty text area]

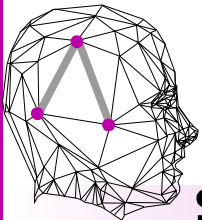
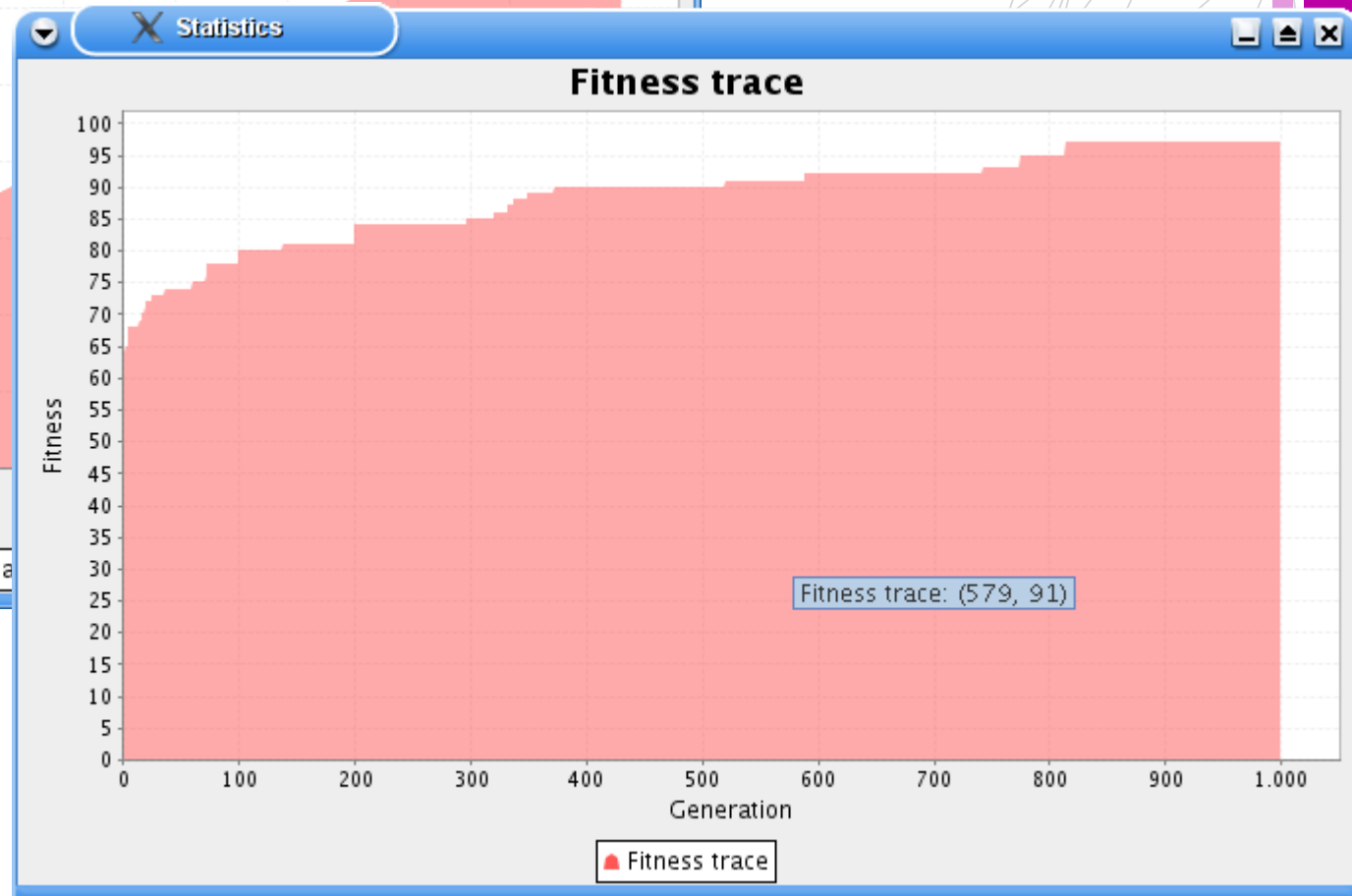
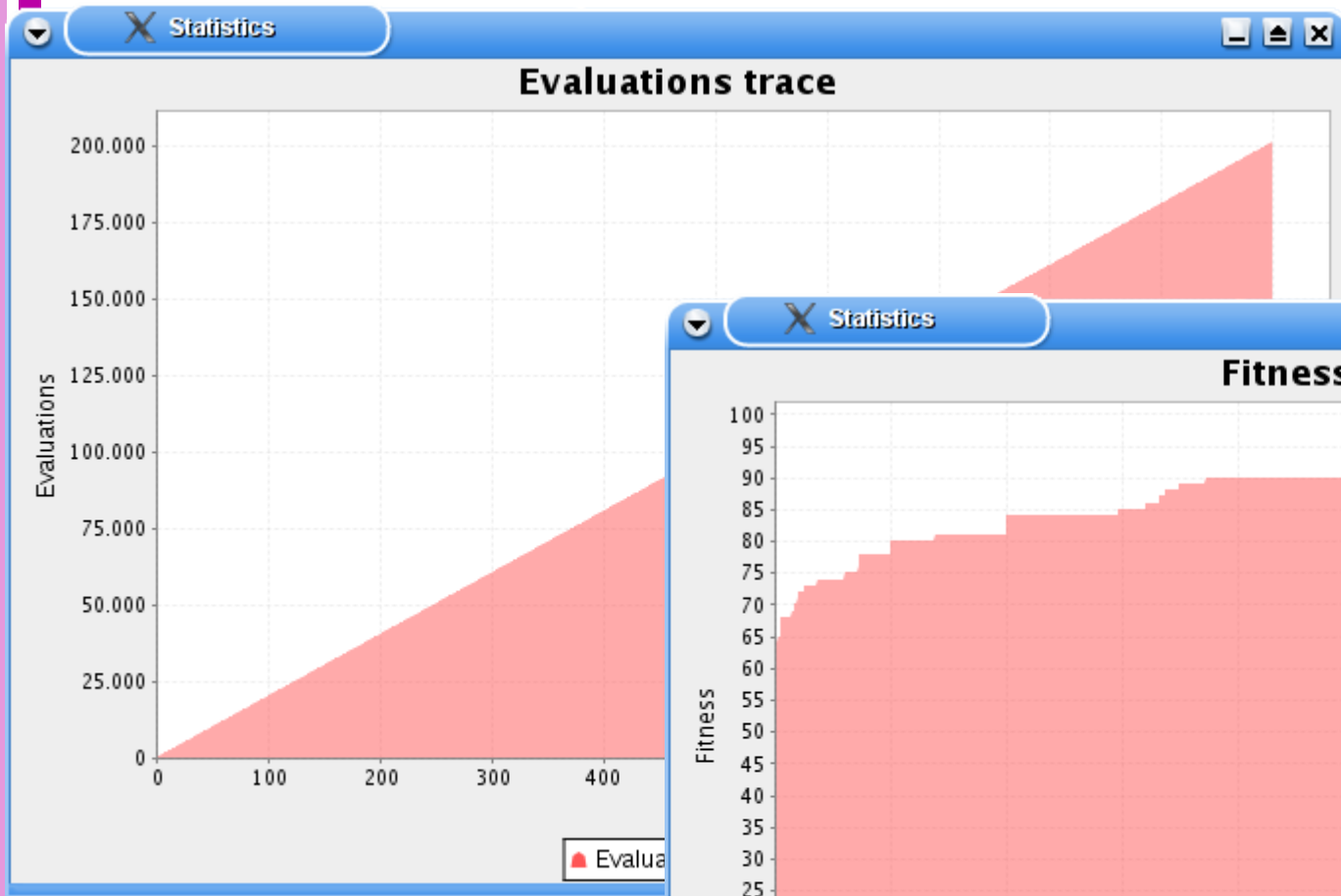
Logo: SIMD (Sistemas Inteligentes y Minería de Datos)



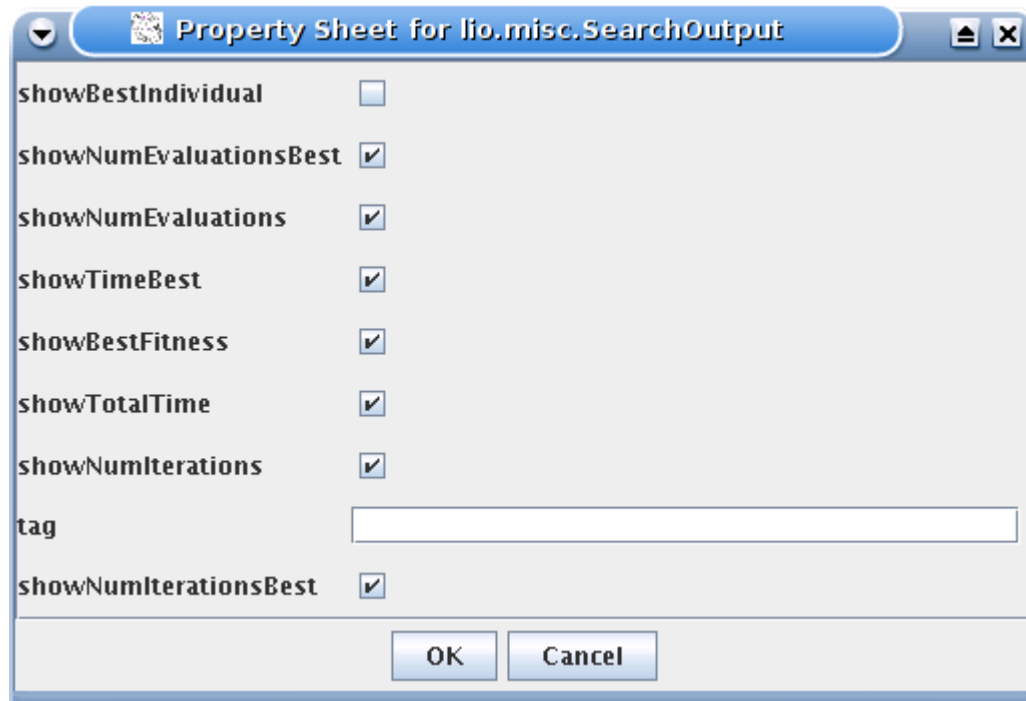
Configuring the objects



Graphical results: Charts



Stopping and output options



Property Sheet for lio.misc.SearchOutput

showBestIndividual

showNumEvaluationsBest

showNumEvaluations

showTimeBest

showBestFitness

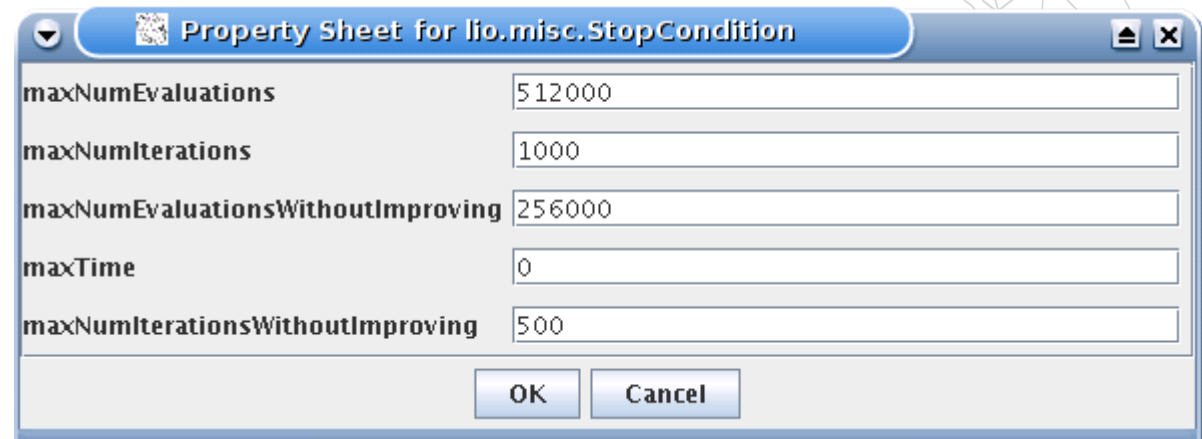
showTotalTime

showNumIterations

tag

showNumIterationsBest

OK Cancel



Property Sheet for lio.misc.StopCondition

maxNumEvaluations 512000

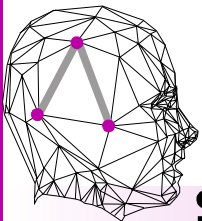
maxNumIterations 1000

maxNumEvaluationsWithoutImproving 256000

maxTime 0

maxNumIterationsWithoutImproving 500

OK Cancel



Configuration file

It can be generated from the graphical interface

```
probMutation=0.05
populationSize=200
probCrossover=0.6

selector=lio.selectors.RouletteWheelSelector
selector.ranking=false

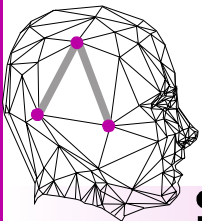
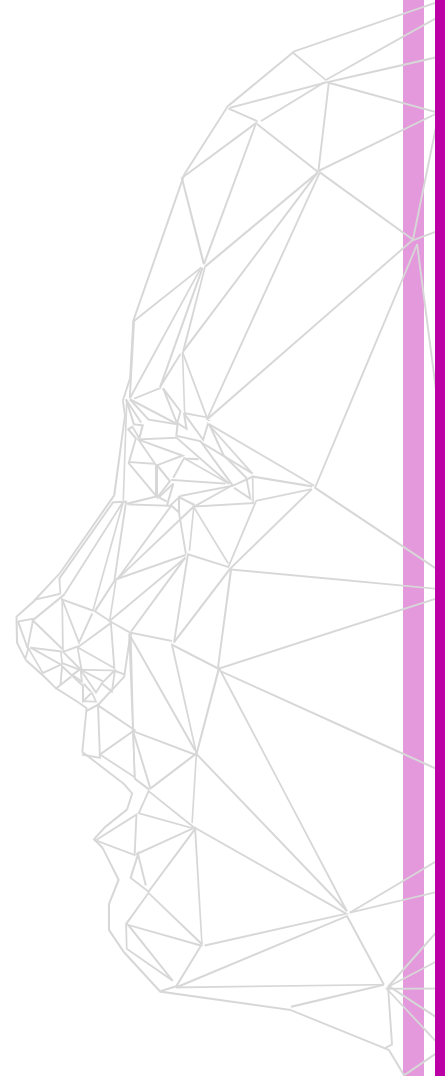
replacer=lio.replacement.SimpleElitistReplacement
replacer.maxPreserved=1
replacer.keepIndividuals=false

generator=lio.generators.bitchain.RandomGenerator

mutation=lio.mutation.bitchain.BinaryMutation

crossover=lio.crossover.bitchain.OnePointCrossover
crossover.point=0

task=problems.bitchain.OneMax
task.size=100
```



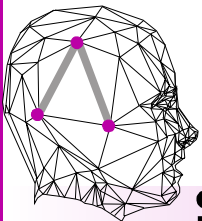
Execution in command line I

```
# Show available parameters
```

```
$java -cp LiO.jar lio.search.genetic.StdGeneticAlgorithm -h
```

Generic options for the search.

- file <conf_file> Name of the file containing the configuration for the search
- param <param>=<value> Name of the parameter and value.
- output <out_file> Name of the file to output results.
- b To avoid show results through standard output.
- h Show this help message.



Execution in command line II

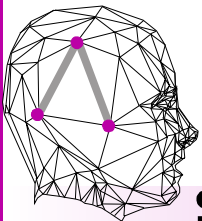
- Using the -param option
- At least we have to set the task to solve, the rest is taken by default

```
# Basic execution
```

```
$java -cp LiO.jar lio.search.genetic.StdGeneticAlgorithm -param  
task=problems.bitchain.OneMax
```

```
Results for the lio.search.genetic.StdGeneticAlgorithm algorithm:
```

```
Task to solve:problems.bitchain.OneMax  
Best fitness: 93.0  
Number of iterations to best: 870.0  
Number of iterations: 1000.0  
Number of evaluations to best: 175233.0  
Number of evaluations: 201200.0  
Time to best: 6706.0 ms  
Total time: 7686.0 ms
```



Execution in command line III

- We can set a file for the results
- Each line represents an execution a every field is separated by a tab character
- This file can be used in several execution because the results are appended to the end

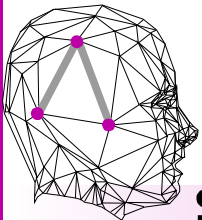
```
# Results are sent to a file
```

```
$java -cp LiO.jar lio.search.genetic.StdGeneticAlgorithm -param  
task=problems.bitchain.OneMax -output OneMax.out
```

```
...
```

```
$cat OneMax.out
```

```
lio.search.genetic.StdGeneticAlgorithm  problems.bitchain.OneMax 91.0  
309.0      809.0      62407.0 162809.0      2426.0 6249.0
```



Execution in command line IV

- If we set many parameters could be easier using a configuration file
- Parameters set in command line have higher priority

```
# Execution with configuration file
```

```
$java -cp LiO.jar lio.search.genetic.StdGeneticAlgorithm -file  
example1.conf -param task.size=5
```

```
Results for the lio.search.genetic.StdGeneticAlgorithm algorithm:
```

```
Task to solve:problems.bitchain.OneMax
```

```
Best fitness: 5.0
```

```
Number of iterations to best: 0.0
```

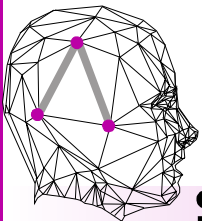
```
Number of iterations: 0.0
```

```
Number of evaluations to best: 38.0
```

```
Number of evaluations: 200.0
```

```
Time to best: 5.0 ms
```

```
Total time: 13.0 ms
```



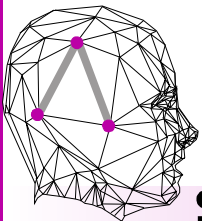
Execution in command line V

■ Using external operators when they are in the CLASSPATH

```
java -cp LiO.jar lio.search.genetic.StdGeneticAlgorithm -param  
task=problems.bitchain.OneMax -param  
crossover=custom.CustomOnePointCrossover
```

■ Using external operators with their .class file

```
java -cp LiO.jar lio.search.genetic.StdGeneticAlgorithm -param  
task=problems.bitchain.OneMax -param  
crossover=custom/CustomOnePointCrossover.class
```



Utility in the command line

■ The `lio.misc.DescribeResource` class show us a description of any resource in LiO

```
$java lio.misc.DescribeResource lio.memetic.HillClimbing
```

```
lio.memetic=lio.memetic.HillClimbing
```

```
Description: Hill Climbing
```

```
Resources:
```

```
    @ neighbourhood:
```

```
    Tip: Neighbourhood operator
```

```
Parameters:
```

```
    & maxNoImprove of type int = 10
```

```
    Tip: Maximum number of steps without improving that can be  
done at each iteration
```

```
    & numCandidates of type int = 1
```

```
    Tip: Number of neighbours considered to choose next move
```

```
    & maxSteps of type int = 2147483647
```

```
    Tip: Maximum number of steps that can be done at each  
iteration
```

