



HeuristicLab

A Paradigm-Independent and Extensible  
Environment for Heuristic Optimization

## Algorithm and Experiment Design with HeuristicLab

An Open Source Optimization Environment for  
Research and Education

S. Wagner, G. Kronberger

Heuristic and Evolutionary Algorithms Laboratory (HEAL)

School of Informatics, Communications and Media, Campus Hagenberg

Upper Austria University of Applied Sciences



HEAL

Heuristic and Evolutionary  
Algorithms Laboratory



Josef Ressel-Zentrum  
**HEUREKA!**

## Instructor Biographies



- Stefan Wagner
  - MSc in computer science (2004)  
Johannes Kepler University Linz, Austria
  - PhD in technical sciences (2009)  
Johannes Kepler University Linz, Austria
  - Associate professor (2005 – 2009)  
Upper Austria University of Applied Sciences
  - Full professor for complex software systems (since 2009)  
Upper Austria University of Applied Sciences
  - Co-founder of the HEAL research group
  - Project manager and chief architect of HeuristicLab
  - <http://heal.heuristiclab.com/team/wagner>
- Gabriel Kronberger
  - MSc in computer science (2005)  
Johannes Kepler University Linz, Austria
  - PhD in technical sciences (2010)  
Johannes Kepler University Linz, Austria
  - Research assistant (since 2005)  
Upper Austria University of Applied Sciences
  - Member of the HEAL research group
  - Architect of HeuristicLab
  - <http://heal.heuristiclab.com/team/kronberger>



# Agenda



- Objectives of the Tutorial
- Introduction
- Where to get HeuristicLab?
- Plugin Infrastructure
- Graphical User Interface
- Available Algorithms & Problems
- Demonstration
- Some Additional Features
- Planned Features
- Team
- Suggested Readings
- Bibliography
- Questions & Answers

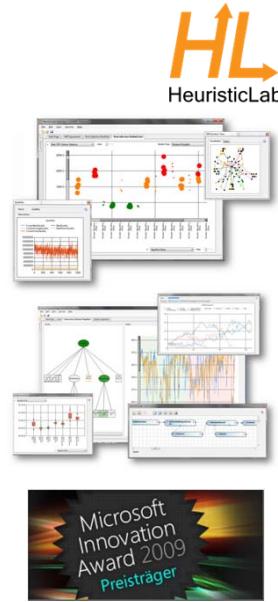
# Objectives of the Tutorial



- Introduce general motivation and design principles of HeuristicLab
- Show where to get HeuristicLab
- Explain basic GUI usability concepts
- Demonstrate basic features
- Demonstrate editing and analysis of optimization experiments
- Demonstrate custom algorithms and graphical algorithm designer
- Demonstrate data-based modeling features
- Outline some additional features

# Introduction

- Motivation and Goals
  - graphical user interface
  - paradigm independence
  - multiple algorithms and problems
  - large scale experiments and analyses
  - parallelization
  - extensibility, flexibility and reusability
  - visual and interactive algorithm development
  - multiple layers of abstraction
- Facts
  - development of HeuristicLab started in 2002
  - based on Microsoft .NET and C#
  - used in research and education
  - second place at the *Microsoft Innovation Award 2009*
  - open source (GNU General Public License)
  - version 3.3.0 released on May 18th, 2010
  - latest version 3.3.4 released on May 4th, 2011



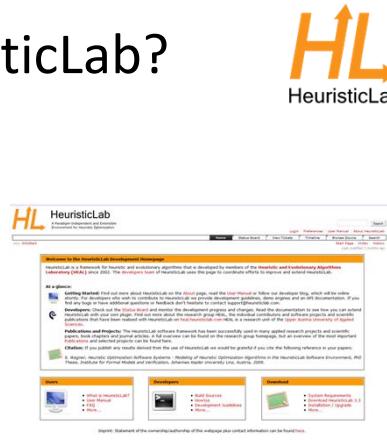
ICCGI 2011

<http://dev.heuristiclab.com>

5

# Where to get HeuristicLab?

- Download binaries
  - deployed as ZIP archives
  - latest stable version 3.3.4
    - released on May 4th, 2011
  - daily trunk build
  - <http://dev.heuristiclab.com/download>
- Check out sources
  - SVN repository
  - HeuristicLab 3.3.4 tag
    - <http://dev.heuristiclab.com/svn/hl/core/tags/3.3.4>
  - current development trunk
    - <http://dev.heuristiclab.com/svn/hl/core/trunk>
- License
  - GNU General Public License (Version 3)
- System requirements
  - Microsoft .NET Framework 4.0 Full Version
  - enough RAM and CPU power ;-)



ICCGI 2011

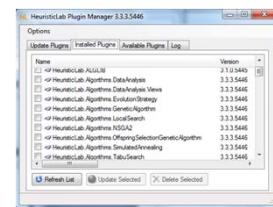
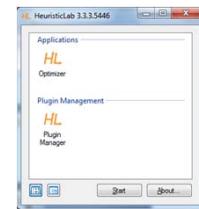
<http://dev.heuristiclab.com>

6

# Plugin Infrastructure



- HeuristicLab consists of many assemblies
  - 95 plugins in HeuristicLab 3.3.4
  - plugins can be loaded or unloaded at runtime
  - plugins can be updated via internet
  - application plugins provide GUI frontends
- Extensibility
  - developing and deploying new plugins is easy
  - dependencies are explicitly defined, automatically checked and resolved
  - automatic discovery of interface implementations (service locator pattern)
- Plugin Manager
  - GUI to check, install, update or delete plugins



ICCGI 2011

<http://dev.heuristiclab.com>

7

# Graphical User Interface



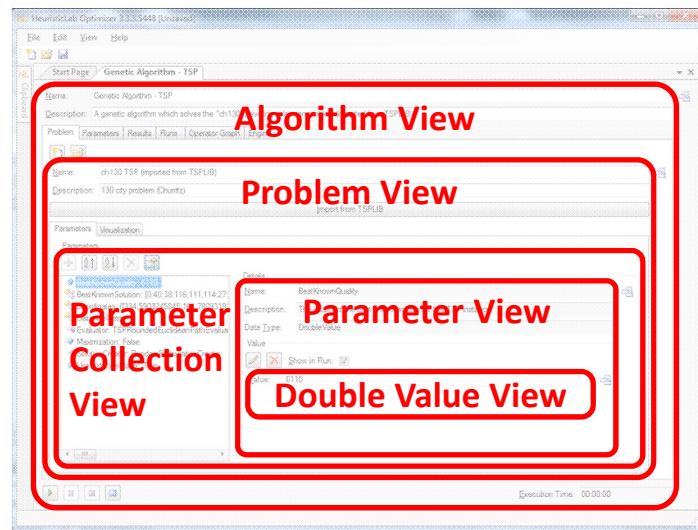
- HeuristicLab GUI is made up of views
  - views are visual representations of content objects
  - views are composed in the same way as their content
  - views and content objects are loosely coupled
  - multiple different views may exist for the same content
- Drag & Drop
  - views support drag & drop operations
  - content objects can be copied or moved (shift key)
  - enabled for collection items and content objects

ICCGI 2011

<http://dev.heuristiclab.com>

8

## Graphical User Interface



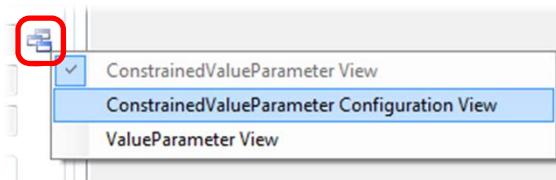
ICCGI 2011

9

## Graphical User Interface



- ViewHost
  - control which hosts views
  - right-click on windows icon to switch views
  - double-click on windows icon to open another view
  - drag & drop windows icon to copy contents



ICCGI 2011

<http://dev.heuristiclab.com>

10

# Available Algorithms & Problems



## Algorithms

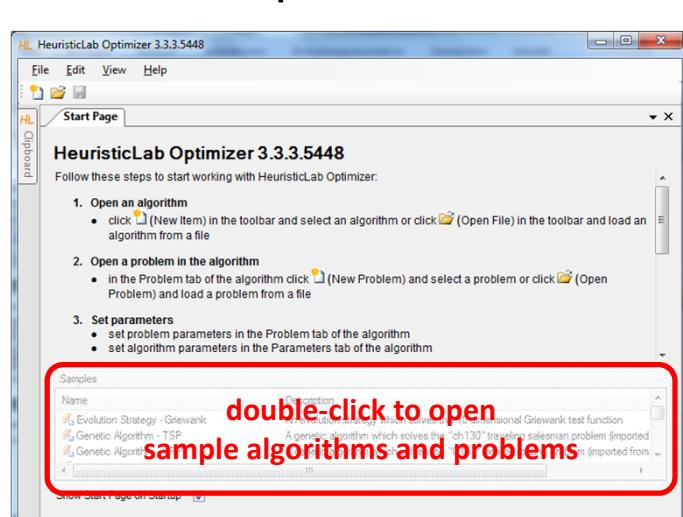
- Genetic Algorithm
- Island Genetic Algorithm
- Offspring Selection Genetic Algorithm
- Island Offspring Selection Genetic Algorithm
- SASEGASA
- Evolution Strategy
- NSGA-II
- Particle Swarm Optimization
- Local Search
- Simulated Annealing
- Tabu Search
- Variable Neighborhood Search
- Linear Regression
- Linear Discriminant Analysis
- Support Vector Machine
- k-Means
- User-defined Algorithm

## Problems

- Single-Objective Test Function
- Traveling Salesman Problem
- Quadratic Assignment Problem
- Vehicle Routing Problem
- Scheduling
- Knapsack
- OneMax
- Data Analysis
- Regression
- Symbolic Regression
- Classification
- Symbolic Classification
- Clustering
- Artificial Ant
- External Evaluation Problem
- User-defined Problem

ICCGI 2011 <http://dev.heuristiclab.com> 11

# HeuristicLab Optimizer



The window shows the 'Start Page' of the HeuristicLab Optimizer. It includes a toolbar with icons for file operations, a menu bar with File, Edit, View, and Help, and a main content area with instructions and a list of samples.

**HeuristicLab Optimizer 3.3.3.5448**

Follow these steps to start working with HeuristicLab Optimizer:

1. Open an algorithm
  - click (New Item) in the toolbar and select an algorithm or click (Open File) in the toolbar and load an algorithm from a file
2. Open a problem in the algorithm
  - in the Problem tab of the algorithm click (New Problem) and select a problem or click (Open Problem) and load a problem from a file
3. Set parameters
  - set problem parameters in the Problem tab of the algorithm
  - set algorithm parameters in the Parameters tab of the algorithm

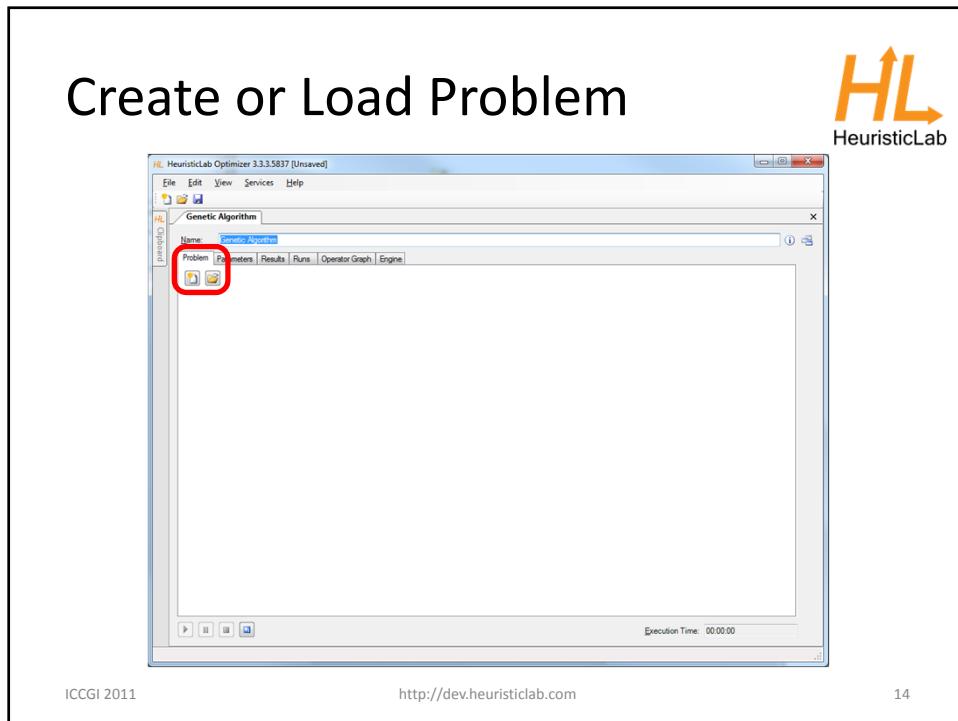
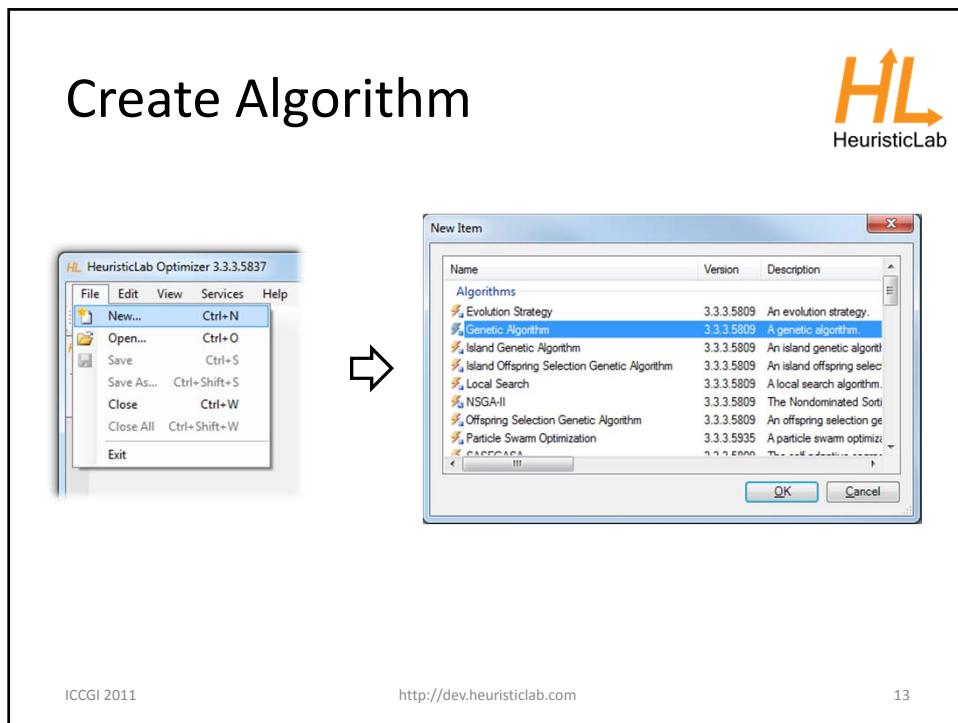
**Samples**

Name	Description
Evolution Strategy - Griewank	A genetic algorithm which solves the "100-dimensional Griewank test function"
Genetic Algorithm - TSP	A genetic algorithm which solves the "10000-dimensional Traveling Salesman Problem (imported from ...)"
Genetic Algorithm - K-Means	A genetic algorithm which solves the "10000-dimensional K-Means clustering problem (imported from ...)"

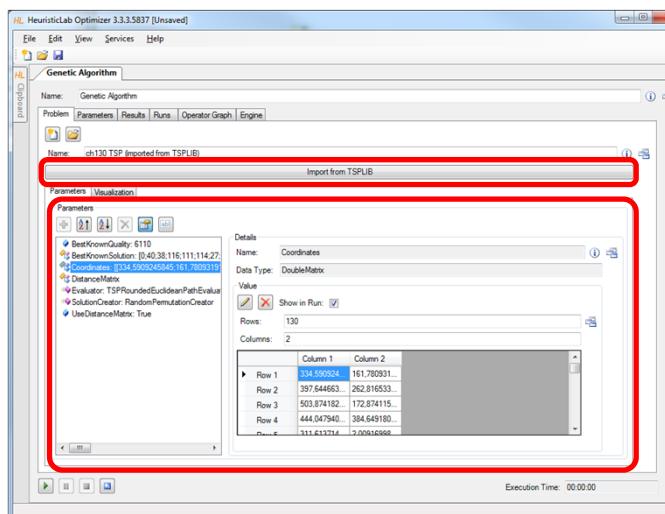
**double-click to open**

**sample algorithms and problems**

ICCGI 2011 <http://dev.heuristiclab.com> 12



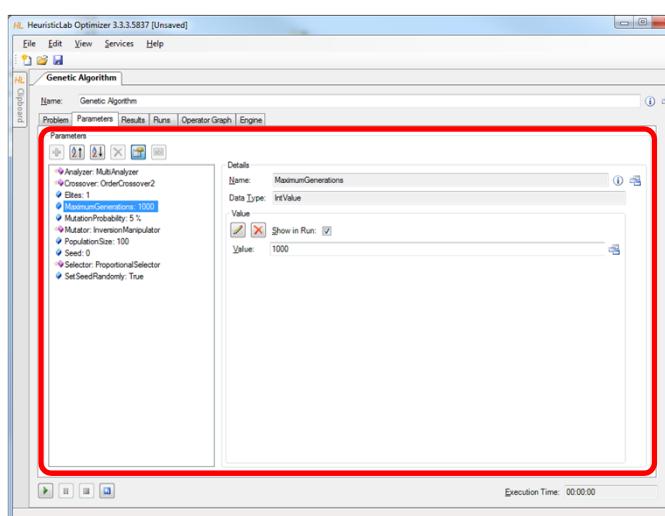
## Import or Parameterize Problem Data



ICCGI 2011

15

## Parameterize Algorithm

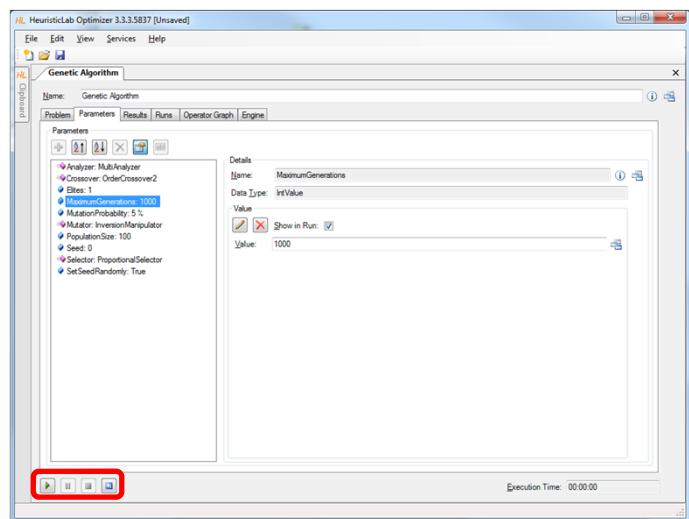


ICCGI 2011

<http://dev.heuristiclab.com>

16

## Start, Pause, Resume, Stop and Reset



ICCGI 2011

<http://dev.heuristiclab.com>

17

## Inspect Results



ICCGI 2011

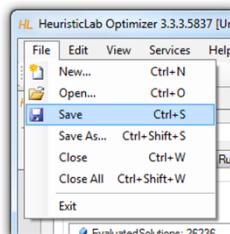
<http://dev.heuristiclab.com>

18

## Save and Load



- Save to and load from disk
  - HeuristicLab items (i.e., algorithms, problems, experiments, ...) can be saved to and loaded from a file
  - algorithms can be paused, saved, loaded and resumed
  - data format is custom compressed XML
  - saving and loading files might take several minutes
  - saving and loading large experiments requires some memory



ICCGI 2011

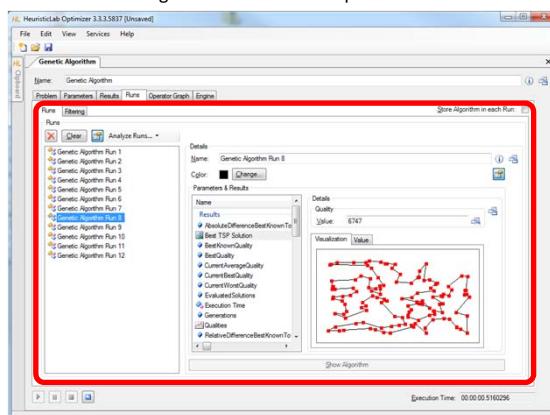
<http://dev.heuristiclab.com>

19

## Compare Runs



- A run is created each time when the algorithm is stopped
  - runs contain all results and parameter settings
  - previous results are not forgotten and can be compared



ICCGI 2011

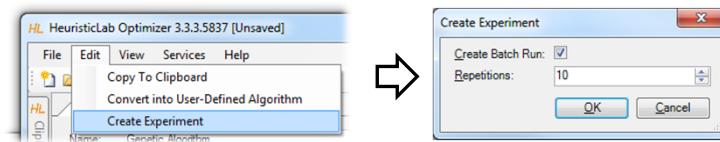
<http://dev.heuristiclab.com>

20

## Create Batch Runs and Experiments



- Batch runs
  - execute the same optimizer (e.g. algorithm, batch run, experiment) several times
- Experiments
  - execute different optimizers
  - suitable for large scale algorithm comparison and analysis
- Experiments and batch runs can be nested
- Generated runs can be compared afterwards



ICCGI 2011

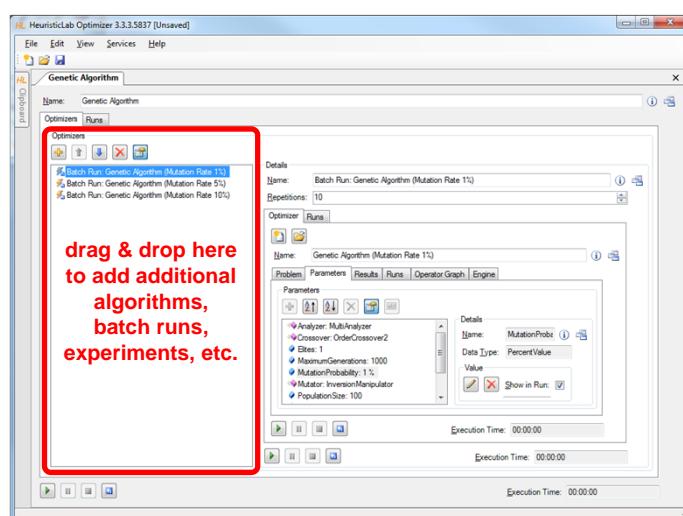
<http://dev.heuristiclab.com>

21

## Create Batch Runs and Experiments



drag & drop here  
to add additional  
algorithms,  
batch runs,  
experiments, etc.

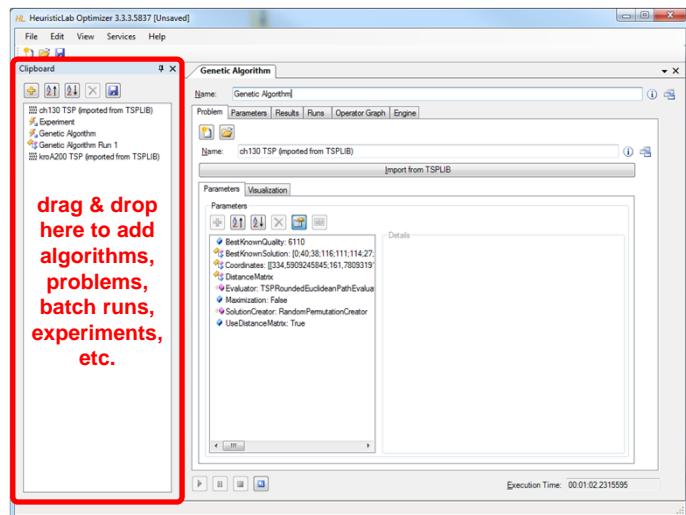


ICCGI 2011

<http://dev.heuristiclab.com>

22

## Clipboard



ICCGI 2011

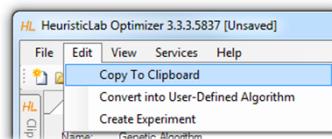
<http://dev.heuristiclab.com>

23

## Clipboard



- Store items
  - click on the buttons to add or remove items
  - drag & drop items on the clipboard
  - use the menu to add a copy of a shown item to the clipboard



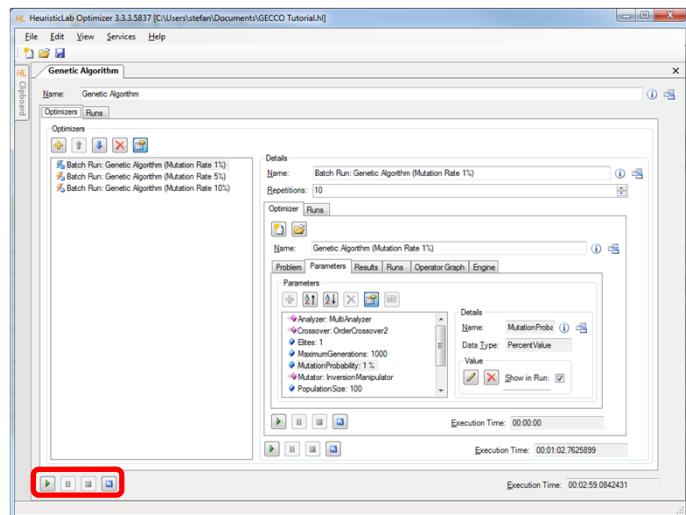
- Show items
  - double-click on an item in the clipboard to show its view
- Save and restore clipboard content
  - click on the save button to write the clipboard content to disk
  - clipboard is automatically restored when HeuristicLab is started the next time

ICCGI 2011

<http://dev.heuristiclab.com>

24

## Start, Pause, Resume, Stop, Reset



ICCGI 2011

<http://dev.heuristiclab.com>

25

## Multi-core CPUs and Parallelization



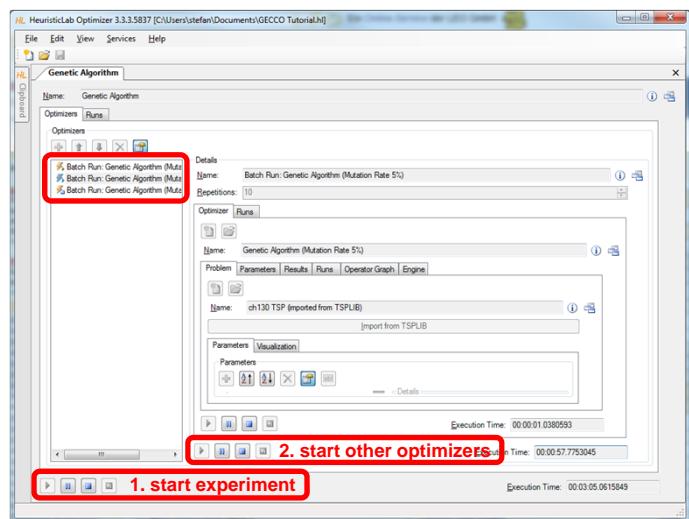
- Parallel execution of optimizers in experiments
  - optimizers in an experiment are executed sequentially from top to bottom per default
  - experiments support parallel execution of their optimizers
  - select a not yet executed optimizer and start it manually to utilize another core
  - execution of one of the next optimizers is started automatically after an optimizer is finished
- Parallel execution of algorithms
  - HeuristicLab provides special operators for parallelization
  - engines decide how to execute parallel operations
  - sequential engine executes everything sequentially
  - parallel engine executes parallel operations on multiple cores
  - Hive engine (under development) executes parallel operations on multiple computers
  - all implemented algorithms support parallel solution evaluation

ICCGI 2011

<http://dev.heuristiclab.com>

26

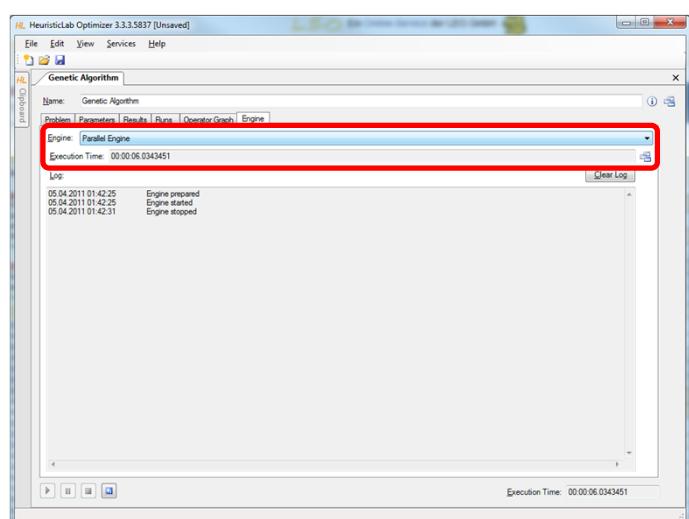
## Parallel Execution of Experiments



ICCGI 2011

27

## Parallel Execution of Algorithms

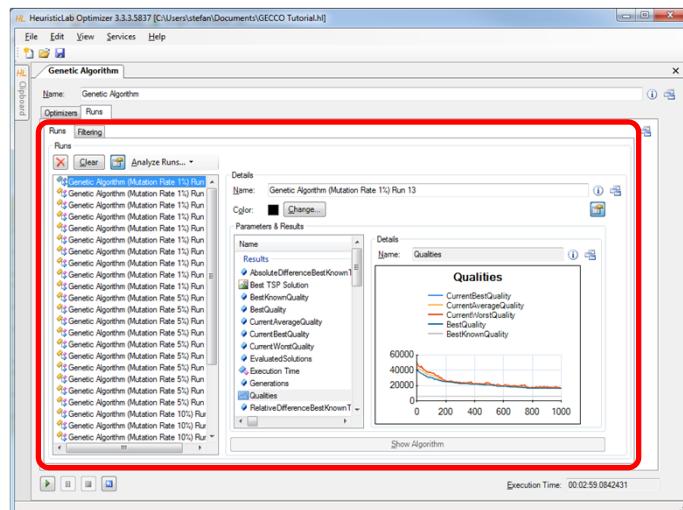


ICCGI 2011

<http://dev.heuristiclab.com>

28

## Compare Runs

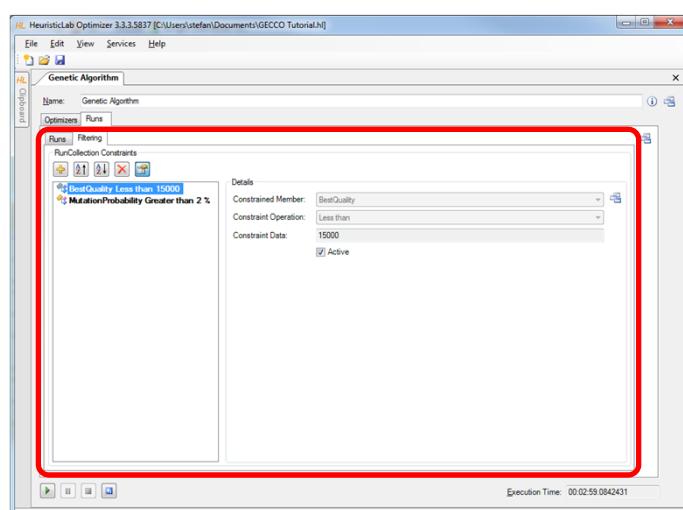


ICCGI 2011

29

<http://dev.heuristiclab.com>

## Filter Runs



ICCGI 2011

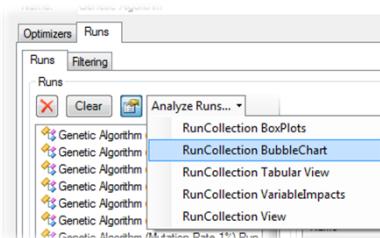
<http://dev.heuristiclab.com>

30

# Analyze Runs



- HeuristicLab provides interactive views to analyze and compare all runs of a run collection
  - textual analysis
    - RunCollection Tabular View
  - graphical analysis
    - RunCollection BubbleChart
    - RunCollection BoxPlots
- Filtering is automatically applied to all open run collection views



ICCGI 2011

<http://dev.heuristiclab.com>

31

# RunCollection Tabular View



HeuristicLab Optimizer 3.3.3.5837 [Unsaved]

File Edit View Services Help

Genetic Algorithm RunCollection Tabular View

Rows: 510 Columns: 48

	BestKnownQuality	BestKnownSolution	BestQuality	Coordinates	Crossover	CurrentAverageQuality
► Genetic Algorithm (Mutation Rate 1%) Run 13	3110	[0 40 38 16 111 114 ...	16405	[334 590 9245 ...	OrderCrossover	16543.13
Genetic Algorithm (Mutation Rate 1%) Run 14	3110	[0 40 38 16 111 114 ...	14783	[334 590 9245 ...	OrderCrossover	15029.02
Genetic Algorithm (Mutation Rate 1%) Run 15	3110	[0 40 38 16 111 114 ...	14252	[334 590 9245 ...	OrderCrossover	14262.89
Genetic Algorithm (Mutation Rate 1%) Run 16	3110	[0 40 38 16 111 114 ...	13243	[334 590 9245 ...	OrderCrossover	13245.95
Genetic Algorithm (Mutation Rate 1%) Run 17	3110	[0 40 38 16 111 114 ...	13703	[334 590 9245 ...	OrderCrossover	13749.98
Genetic Algorithm (Mutation Rate 1%) Run 18	3110	[0 40 38 16 111 114 ...	13564	[334 590 9245 ...	OrderCrossover	13951.09
Genetic Algorithm (Mutation Rate 1%) Run 19	3110	[0 40 38 16 111 114 ...	15421	[334 590 9245 ...	OrderCrossover	15431.74
Genetic Algorithm (Mutation Rate 1%) Run 20	3110	[0 40 38 16 111 114 ...	14409	[334 590 9245 ...	OrderCrossover	15147
Genetic Algorithm (Mutation Rate 1%) Run 21	3110	[0 40 38 16 111 114 ...	13771	[334 590 9245 ...	OrderCrossover	13954.56
Genetic Algorithm (Mutation Rate 1%) Run 22	3110	[0 40 38 16 111 114 ...	14529	[334 590 9245 ...	OrderCrossover	14532.3
Genetic Algorithm (Mutation Rate 5%) Run 13	3110	[0 40 38 16 111 114 ...	13095	[334 590 9245 ...	OrderCrossover	13642.7
Genetic Algorithm (Mutation Rate 5%) Run 14	3110	[0 40 38 16 111 114 ...	12403	[334 590 9245 ...	OrderCrossover	12818.09
Genetic Algorithm (Mutation Rate 5%) Run 15	3110	[0 40 38 16 111 114 ...	14098	[334 590 9245 ...	OrderCrossover	14653.38
Genetic Algorithm (Mutation Rate 5%) Run 16	3110	[0 40 38 16 111 114 ...	12595	[334 590 9245 ...	OrderCrossover	13297.99
Genetic Algorithm (Mutation Rate 5%) Run 17	3110	[0 40 38 16 111 114 ...	12792	[334 590 9245 ...	OrderCrossover	13264.38
Genetic Algorithm (Mutation Rate 5%) Run 18	3110	[0 40 38 16 111 114 ...	12711	[334 590 9245 ...	OrderCrossover	13151.19
Genetic Algorithm (Mutation Rate 5%) Run 19	3110	[0 40 38 16 111 114 ...	12326	[334 590 9245 ...	OrderCrossover	12625.78
Genetic Algorithm (Mutation Rate 5%) Run 20	3110	[0 40 38 16 111 114 ...	13346	[334 590 9245 ...	OrderCrossover	13777.85
Genetic Algorithm (Mutation Rate 5%) Run 21	3110	[0 40 38 16 111 114 ...	12897	[334 590 9245 ...	OrderCrossover	13284.81
Genetic Algorithm (Mutation Rate 5%) Run 22	3110	[0 40 38 16 111 114 ...	12741	[334 590 9245 ...	OrderCrossover	13113.18
Genetic Algorithm (Mutation Rate 10%) Run 13	3110	[0 40 38 16 111 114 ...	15921	[334 590 9245 ...	OrderCrossover	18084.04
Genetic Algorithm (Mutation Rate 10%) Run 14	3110	[0 40 38 16 111 114 ...	16384	[334 590 9245 ...	OrderCrossover	19609.36

ICCGI 2011

<http://dev.heuristiclab.com>

32

## RunCollection Tabular View



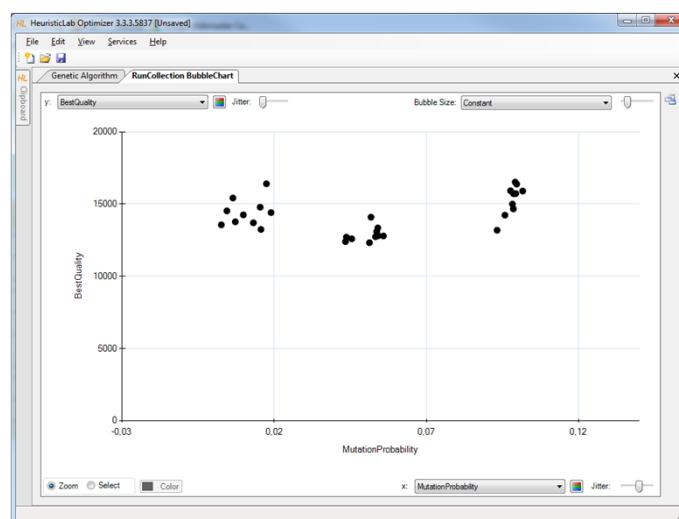
- Sort columns
  - click on column header to sort column
  - Ctrl-click on column header to sort multiple columns
- Show or hide columns
  - right-click on table to open dialog to show or hide columns
- Compute statistical values
  - select multiple numerical values to see count, sum, minimum, maximum, average and standard deviation
- Select, copy and paste into other applications

ICCGI 2011

<http://dev.heuristiclab.com>

33

## RunCollection BubbleChart



ICCGI 2011

<http://dev.heuristiclab.com>

34

# RunCollection BubbleChart



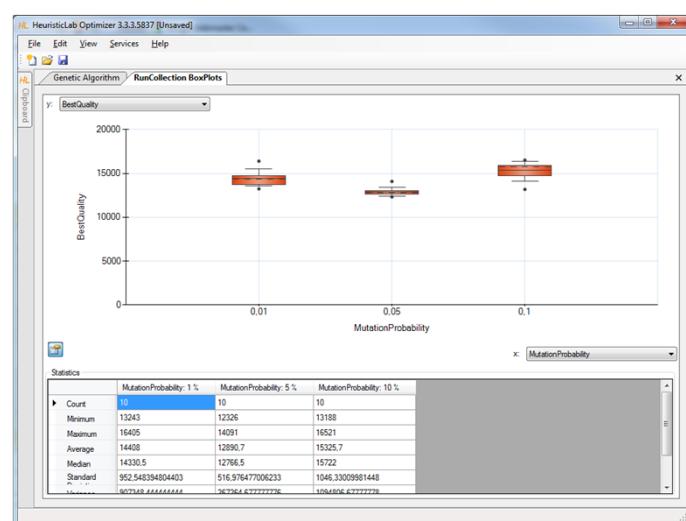
- Choose values to plot
  - choose which values to show on the x-axis, the y-axis and as bubble size
  - possible values are all parameter settings and results
- Add jitter
  - add jitter to separate overlapping bubbles
- Zoom in and out
  - click on Zoom and click and drag in the chart area to zoom in
  - double click on the chart area background or on the circle buttons beside the scroll bars to zoom out
- Color bubbles
  - click on Select, choose a color and click and drag in the chart area to select and color bubbles
  - apply coloring automatically by clicking on the axis coloring buttons
- Show runs
  - double click on a bubble to open its run
- Export image
  - right-click to open context menu to copy or save image
  - save image as pixel (BMP, JPG, PNG, GIF, TIF) or vector graphics (EMF)
- Show box plots
  - right-click to open context menu to show box plots view

ICCGI 2011

<http://dev.heuristiclab.com>

35

# RunCollection BoxPlots



ICCGI 2011

<http://dev.heuristiclab.com>

36

## RunCollection BoxPlots



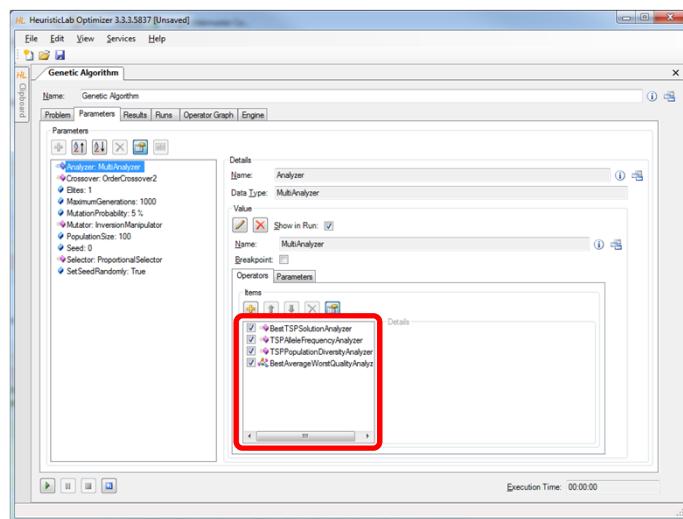
- Choose values to plot
  - choose which values to show on the x-axis and y-axis
  - possible values are all parameter settings and results
- Zoom in and out
  - click on Zoom and click and drag in the chart area to zoom in
  - double click on the chart area background or on the circle buttons beside the scroll bars to zoom out
- Show or hide statistical values
  - click on the lower left button to show or hide statistical values
- Export image
  - right-click to open context menu to copy or save image
  - save image as pixel (BMP, JPG, PNG, GIF, TIF) or vector graphics (EMF)

## Analyzers



- Special operators for analysis purposes
  - are executed after each iteration
  - serve as general purpose extension points of algorithms
  - can be selected and parameterized in the algorithm
  - perform algorithm-specific and/or problem-specific tasks
  - some analyzers are quite costly regarding runtime and memory
  - implementing and adding custom analyzers is easy
- Examples
  - TSPAlleleFrequencyAnalyzer
  - TSPPopulationDiversityAnalyzer
  - SuccessfulOffspringAnalyzer
  - SymbolicDataAnalysisVariableFrequencyAnalyzer
  - SymbolicRegressionSingleObjectiveTrainingBestSolutionAnalyzer
  - ...

## Analyzers

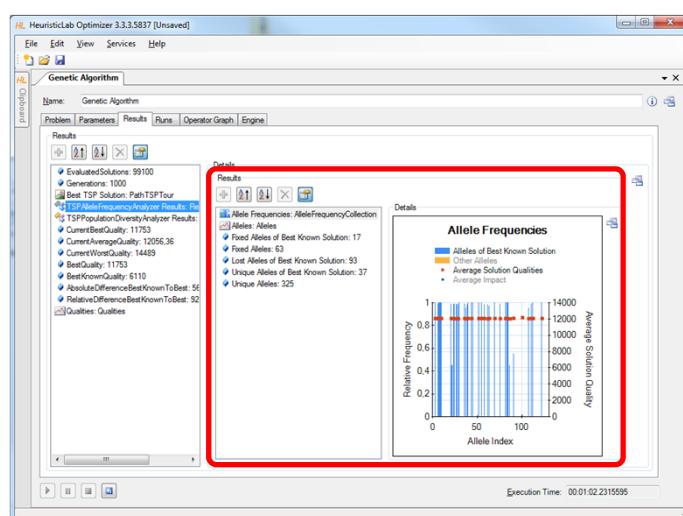


ICCGI 2011

<http://dev.heuristiclab.com>

39

## TSPAlleleFrequencyAnalyzer

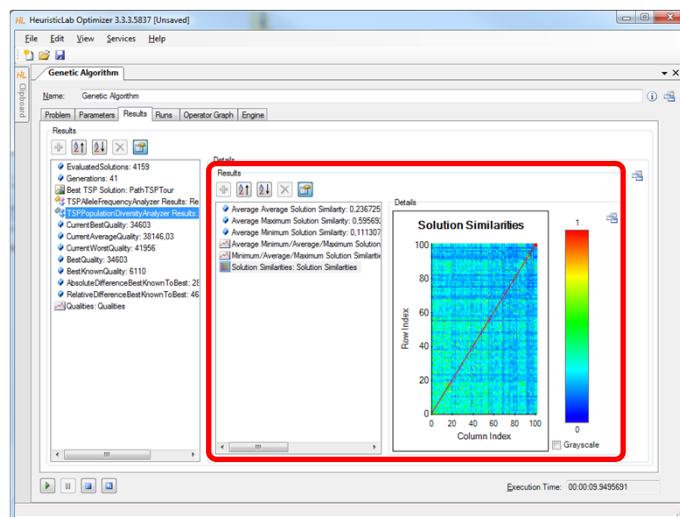


ICCGI 2011

<http://dev.heuristiclab.com>

40

## TSPPopulationDiversityAnalyzer



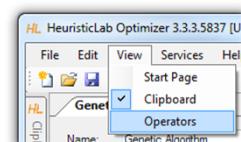
ICCGI 2011

41

## Building User-Defined Algorithms



- Operator graphs
  - algorithms are represented as operator graphs
  - operator graphs of user-defined algorithms can be changed
  - algorithms can be defined in the graphical algorithm designer
  - use the menu to convert a standard algorithm into a user-defined algorithm



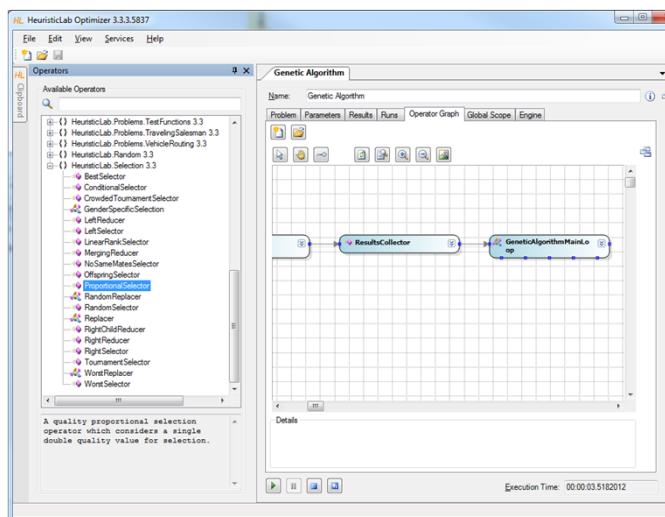
- Operators sidebar
  - drag & drop operators into an operator graph
- Programmable operators
  - add programmable operators in order to implement custom logic in an algorithm
  - no additional development environment needed
- Debug algorithms
  - use the debug engine to obtain detailed information during algorithm execution

ICCGI 2011

<http://dev.heuristiclab.com>

42

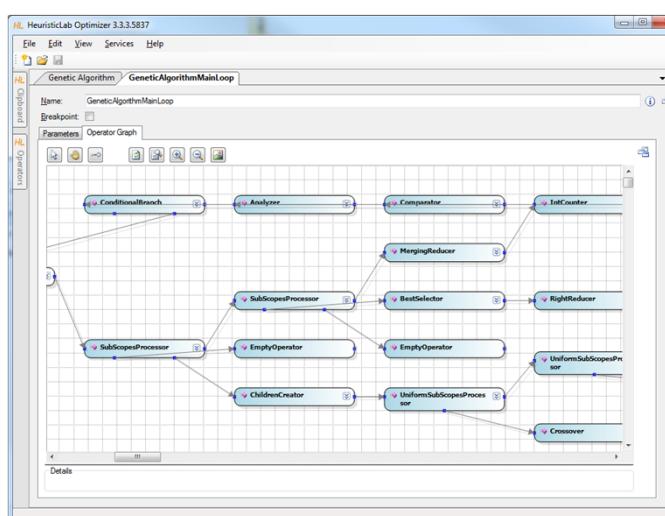
## Building User-Defined Algorithms



ICCGI 2011

43

## Building User-Defined Algorithms

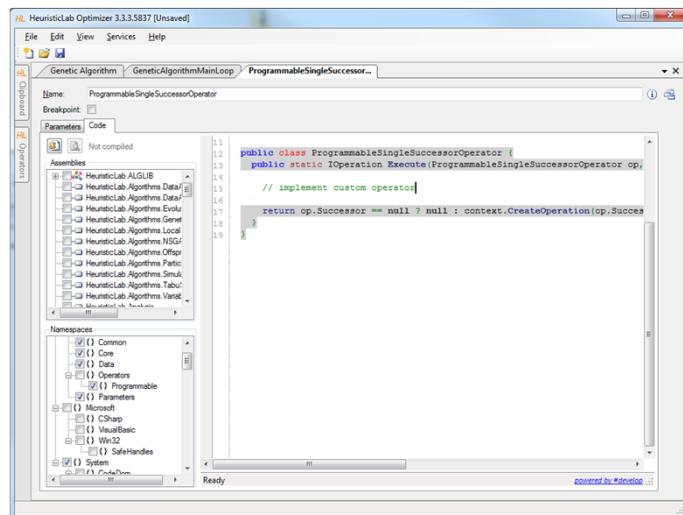


ICCGI 2011

<http://dev.heuristiclab.com>

44

# Programmable Operators

```

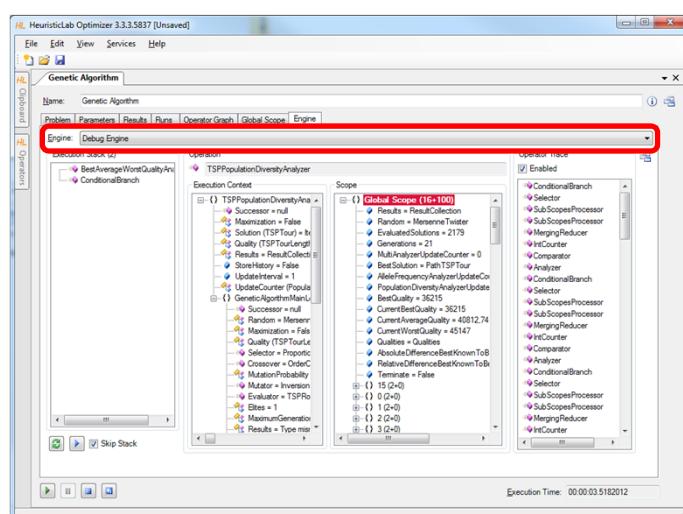
public class ProgrammableSingleSuccessorOperator : public IOperation
{
    public static IOperation Execute(ProgrammableSingleSuccessorOperator op,
        IOperationContext context)
    {
        // implement custom operator
        return op.Successor == null ? null : context.CreateOperation(op.Successor);
    }
}

```

ICCGI 2011

45

# Debugging Algorithms

The screenshot shows the HeuristicLab Optimizer interface with the 'Engine' tab selected. The 'Scope' pane displays the execution context of a 'TSPPopulationDiversityAnalyzer' operation, showing various parameters and their values. The 'Operator Trace' pane shows a detailed trace of the algorithm's execution, listing operations like 'EvaluateSolutions', 'Generations', 'MultiAnalyzerUpdate', 'UpdateCounter', 'PopulationDiversityAnalyzerUpdate', and 'Terminate'. The 'Execution Context' pane shows the current state of the algorithm, including 'BestTourLength' (36215), 'CurrentTourLength' (36215), and 'CurrentWorstQuality' (45147). The 'Execution Time' is listed as 00:00:03.5182012.

ICCGI 2011

46

## Introduction to Data-based Modeling



- Dataset: Matrix  $(x_{i,j})_{i=1..N, j=1..K}$ 
  - N observations of K input variables
  - $x_{i,j}$  = i-th observation of j-th variable
  - Additionally: Vector of labels  $(y_1 \dots y_N)^T$
- Goal: learn association of input variable values to labels
- Common tasks
  - Regression (real-valued labels)
  - Classification (discrete labels)
  - Clustering (no labels, group similar observations)

## Data-based Modeling Algorithms in HeuristicLab



- Symbolic regression and classification based on genetic programming
- External Libraries:
  - Support Vector Machines for Regression and Classification
  - Linear Regression
  - Linear Discriminate Analysis
  - K-Means clustering

## Case Studies



- Regression
  - Artificial benchmark problem dataset *Poly-10*
  - Algorithms:
    - Linear regression
    - Symbolic regression using Genetic Programming
- Classification
  - Real world medical *Mammographic Mass* dataset from the UCI Machine Learning Repository
  - Algorithms:
    - Symbolic classification

## Case Study: Regression

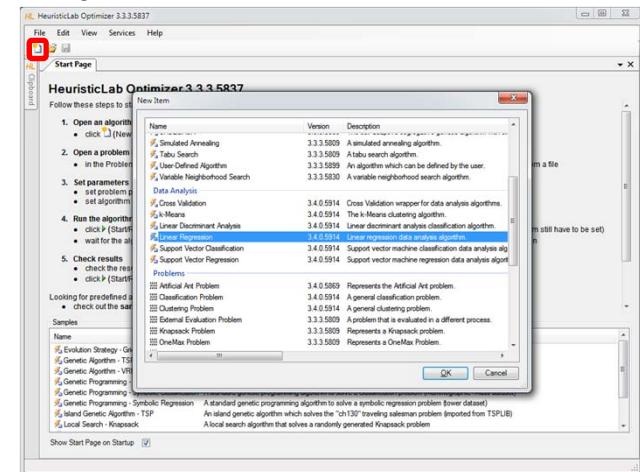


- Poly-10 benchmark problem dataset
  - 10 input variables  $x_1 \dots x_{10}$
  - $y = x_1 \cdot x_2 + x_3 \cdot x_4 + x_5 \cdot x_6 + x_1 \cdot x_7 \cdot x_9 + x_3 \cdot x_6 \cdot x_{10}$
  - Non-linear modeling approach necessary
  - Frequently used in GP literature
  - Download:
    - <http://dev.heuristiclab.com/AdditionalMaterial#ICCGI2011>

# Linear Regression



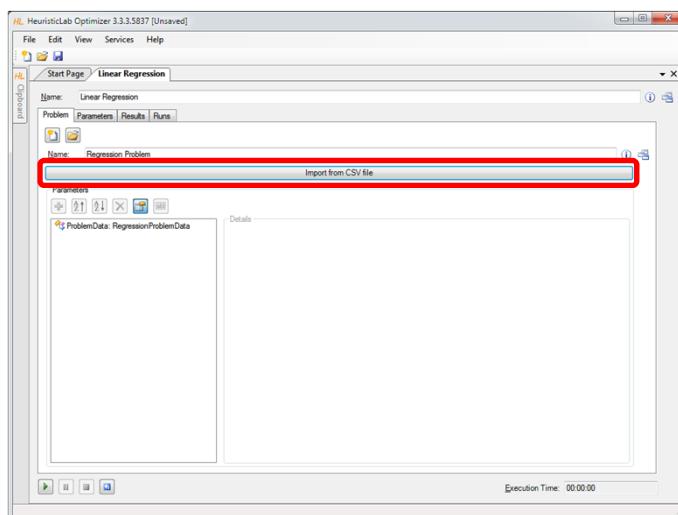
- Create new algorithm



ICCGI 2011

51

# Import Data from CSV-File

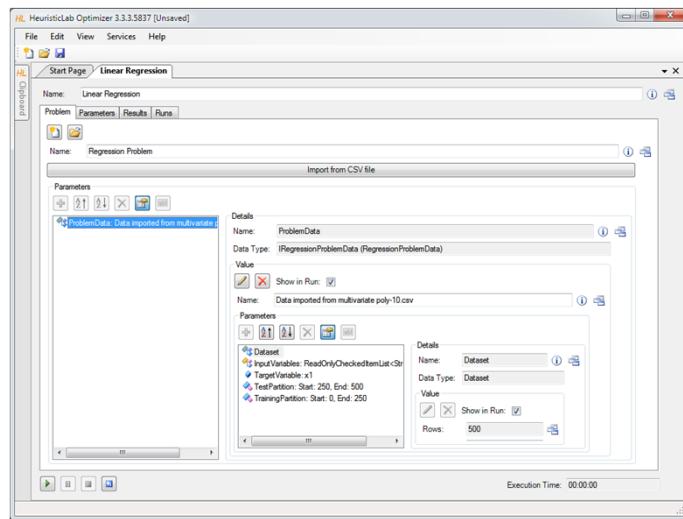


ICCGI 2011

http://dev.heuristiclab.com

52

## Inspect and Configure Dataset

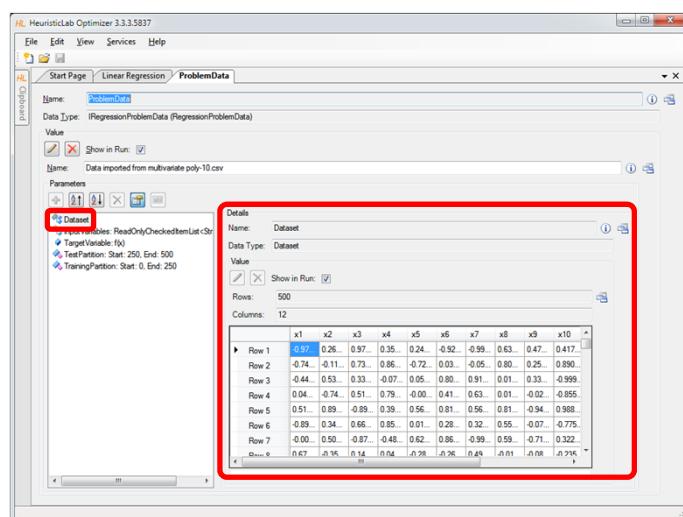


ICCGI 2011

<http://dev.heuristiclab.com>

53

## Inspect Imported Data

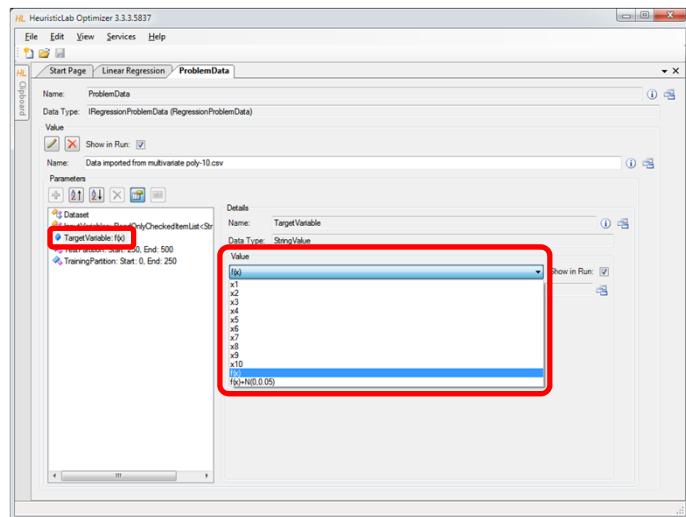


ICCGI 2011

<http://dev.heuristiclab.com>

54

## Set Target Variable

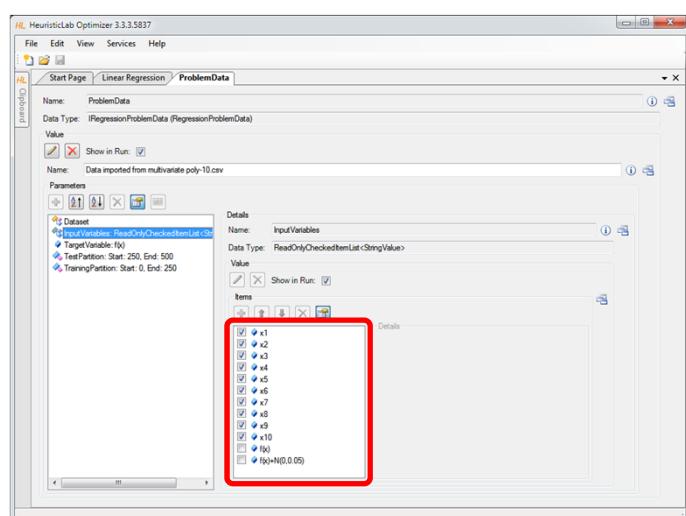


ICCGI 2011

<http://dev.heuristiclab.com>

55

## Select Input Variables

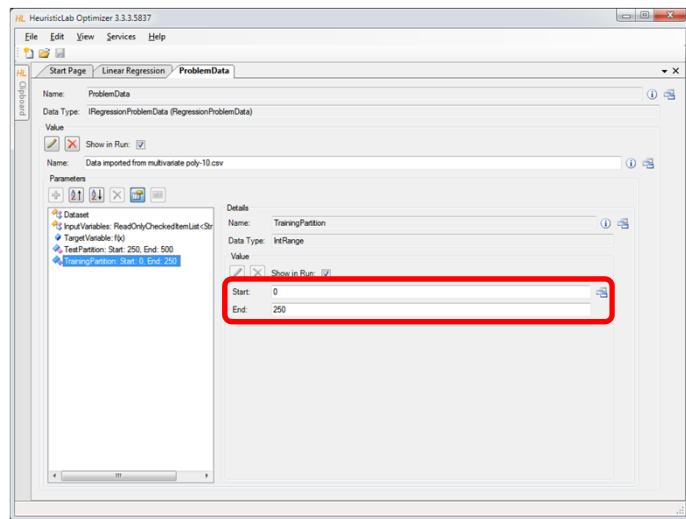


ICCGI 2011

<http://dev.heuristiclab.com>

56

## Configure Training and Test Partitions

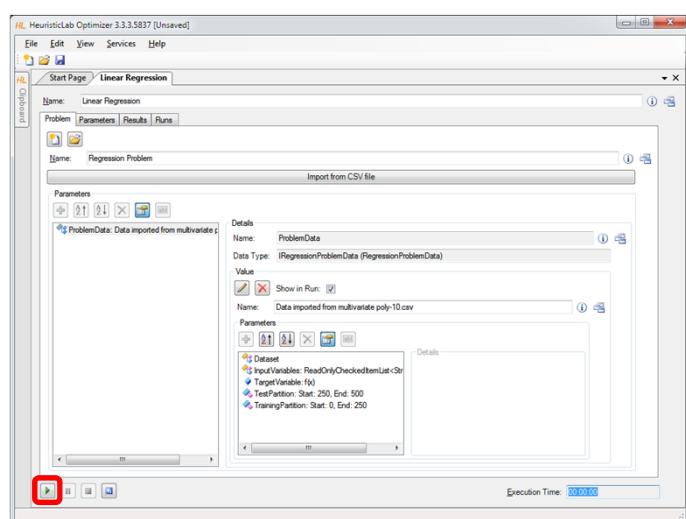


ICCGI 2011

<http://dev.heuristiclab.com>

57

## Run Linear Regression

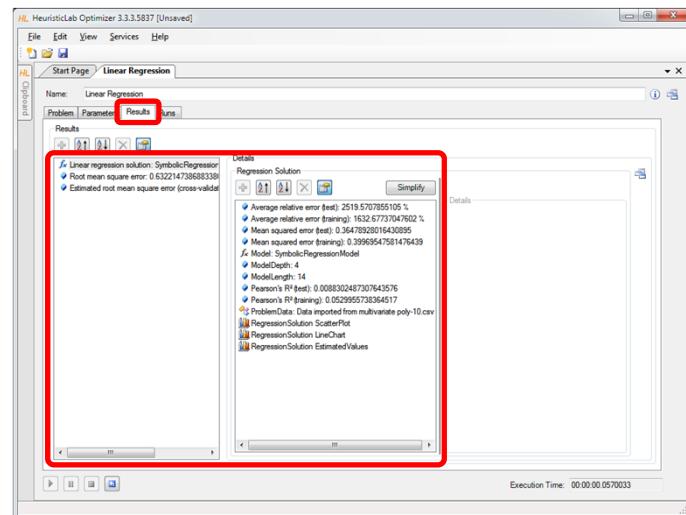


ICCGI 2011

<http://dev.heuristiclab.com>

58

## Inspect Results

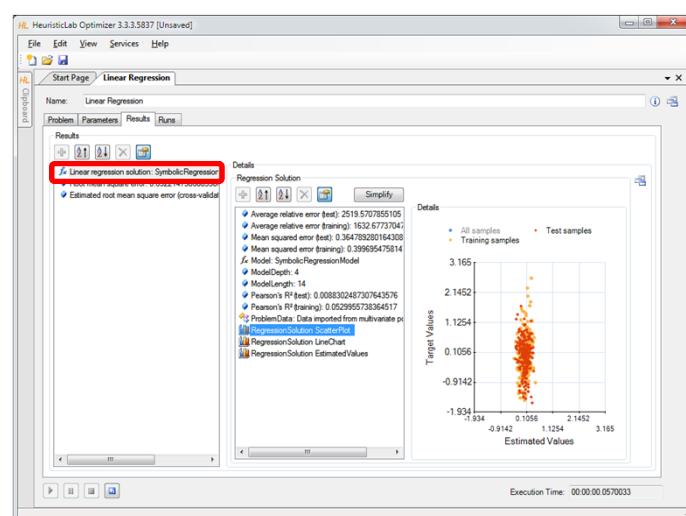


ICCGI 2011

<http://dev.heuristiclab.com>

59

## Inspect Scatterplot of Predicted and Target Values

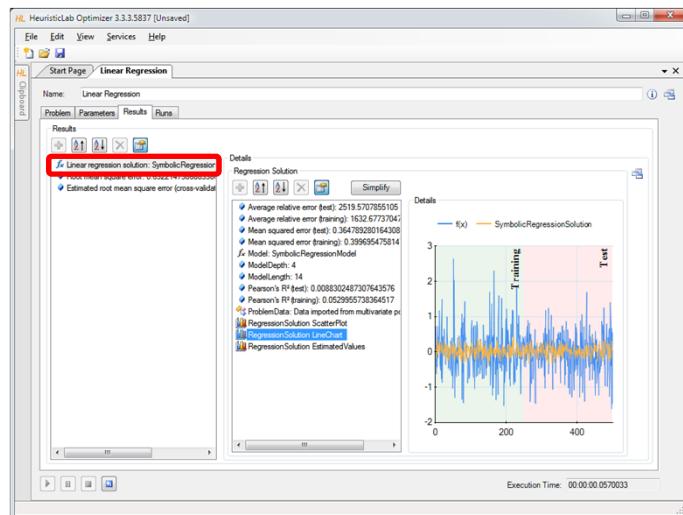


ICCGI 2011

<http://dev.heuristiclab.com>

60

## Inspect Linechart

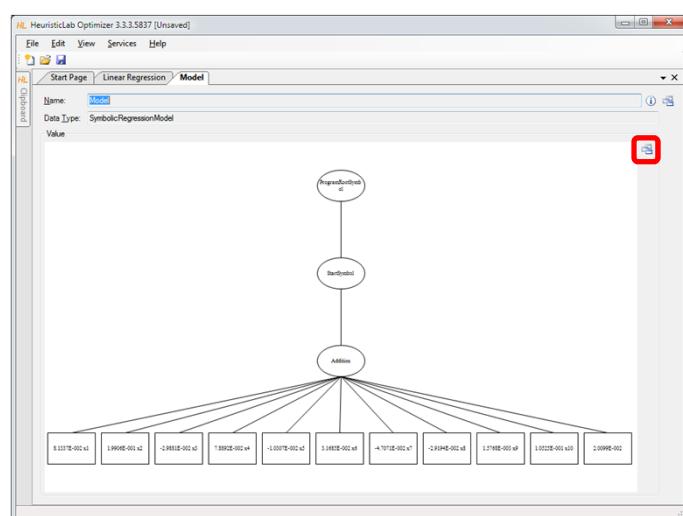


ICCGI 2011

<http://dev.heuristiclab.com>

61

## Inspect Graphical Representation of Model



ICCGI 2011

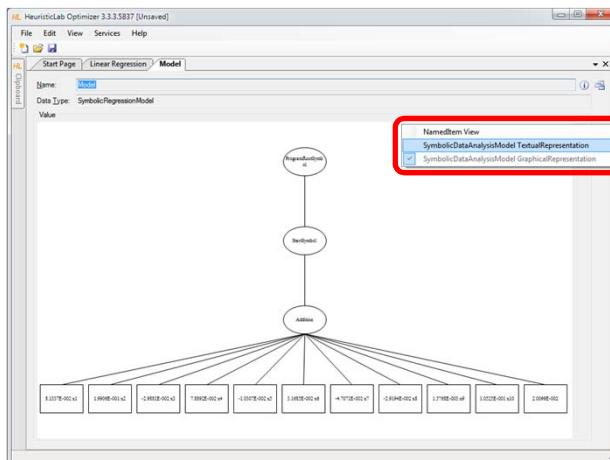
<http://dev.heuristiclab.com>

62

## Textual Representations Are Also Available



- Use *ViewHost* to switch to textual representation view



ICCGI 2011

<http://dev.heuristiclab.com>

63

## Default Textual Representation for Model Export



```

HeuristicLab Optimizer 3.3.3.5837 [Unsaved]
File Edit View Services Help
HL Start Page | Linear Regression | Model
Name: Model
Data Type: SymbolicRegressionModel
Value
Formatter: Default String Formatter
(Prog-Rng-Symbol
(Star-Symbol
(Addition
(3.1337E-002*x)
(1.5908E-001*x)
(-2.9881E-002*x)
(7.0302E-002*x)
(-1.0307E-002*x)
(3.1852E-002*x)
(4.2071E-002*x)
(2.5194E-002*x)
(1.9525E-001*x)
(1.0252E-001*x)
(2.0099E-002)
)
)
)

```

ICCGI 2011

<http://dev.heuristiclab.com>

64

## Textual Representation for Export to LaTeX



HeuristicLab Optimizer 3.3.3.5897 [Unsaved]

File Edit View Services Help

Start Page Linear Regression Model

Item Model Data Type: SymbolicRegressionModel Value

Formatter: LaTeX String Formatter

```
% needs \usepackage{amsmath}
\begin{aligned}
\text{Result} = & (c_0 x(t) + c_1 x_2(t) + c_2 x_3(t) + c_3 x_4(t) + c_4 x_5(t) + c_5 x_6(t) + c_6 x_7(t) + c_7 x_8(t) + c_8 x_9(t) + c_9 x_{10}(t) + c_{10}) \\
& \cdot (c_{10} t + 0.0813371220642195) \\
& + (c_{10} t^2 + 0.0298811744629839) \\
& + (c_{10} t^3 + 0.0789193541302) \\
& + (c_{10} t^4 + 0.031684953696999) \\
& + (c_{10} t^5 + 0.0200987846232596) \\
& + (c_{10} t^6 + 0.01525443686677) \\
& + (c_{10} t^7 + 0.00157679665070775) \\
& + (c_{10} t^8 + 0.000157679665070775) \\
& + (c_{10} t^9 + 0.0000157679665070775) \\
& + (c_{10} t^{10} + 0.00000157679665070775) \\
\end{aligned}
```

Result =  $(c_0 x(t) + c_1 x_2(t) + c_2 x_3(t) + c_3 x_4(t) + c_4 x_5(t) + c_5 x_6(t) + c_6 x_7(t) + c_7 x_8(t) + c_8 x_9(t) + c_9 x_{10}(t) + c_{10})$

$c_0 = 0.0813371220642195$  (1)

$c_1 = 0.199055016563887$  (2)

$c_2 = -0.0298811744629839$  (3)

$c_3 = 0.0789193541302$  (4)

$c_4 = -0.00103065273366223$  (5)

$c_5 = 0.031684953696999$  (6)

$c_6 = -0.0170707585925129$  (7)

$c_7 = -0.0291939124032144$  (8)

$c_8 = 0.00157679665070775$  (9)

$c_9 = 0.01032504436866577$  (10)

$c_{10} = 0.0000987846232596$  (11)

$c_{10} = 0.00000157679665070775$  (12)

ICCGI 2011

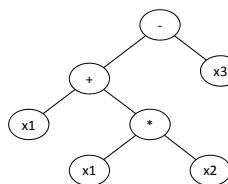
<http://dev.heuristiclab.com>

65

## Nonlinear Modeling: Symbolic Regression



- Linear regression produced an inaccurate model.
- Next: produce a nonlinear symbolic regression model using genetic programming
- Genetic programming
  - Evolve variable-length models
  - Model representation: symbolic expression tree
  - Structure and model parameters are evolved side-by-side
  - White-box models

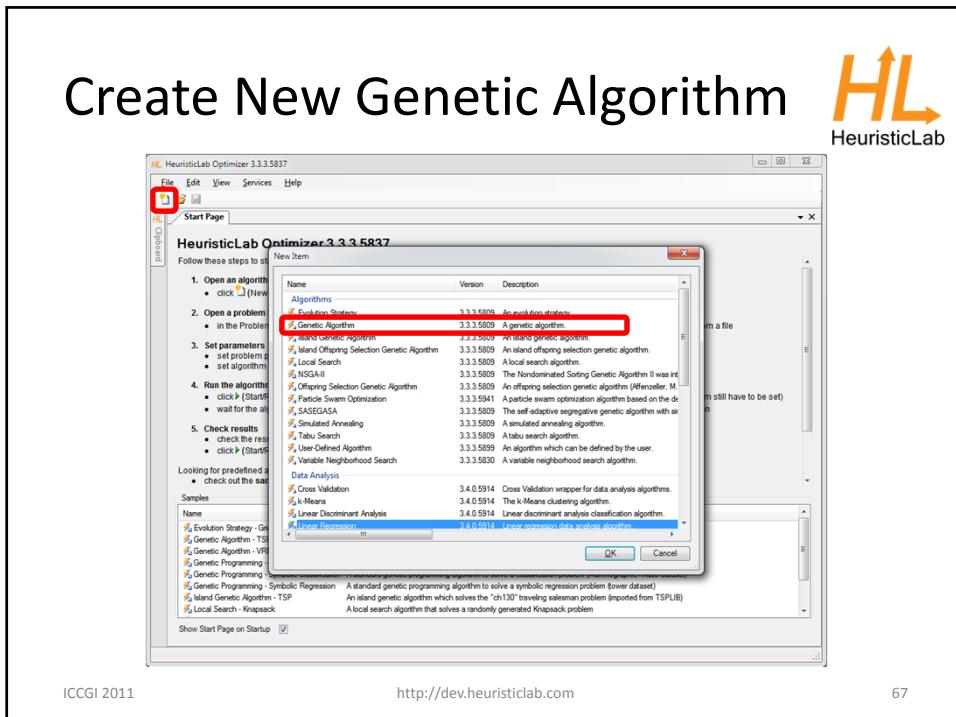


ICCGI 2011

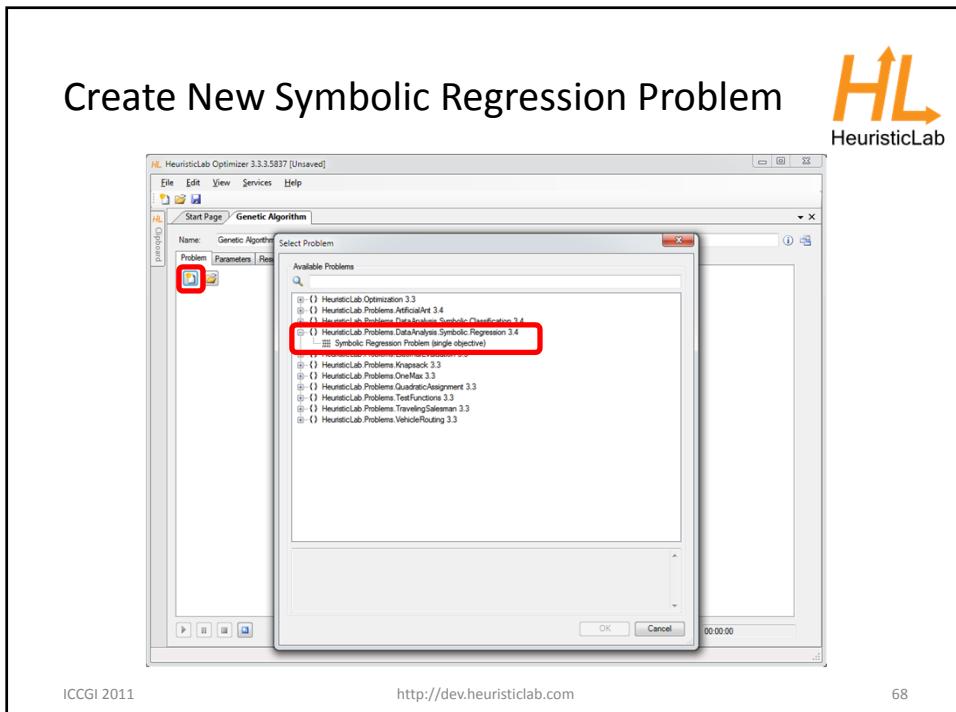
<http://dev.heuristiclab.com>

66

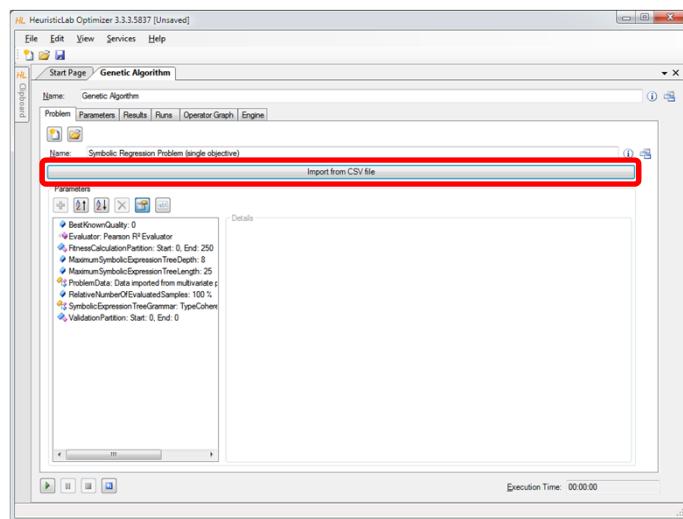
## Create New Genetic Algorithm



## Create New Symbolic Regression Problem



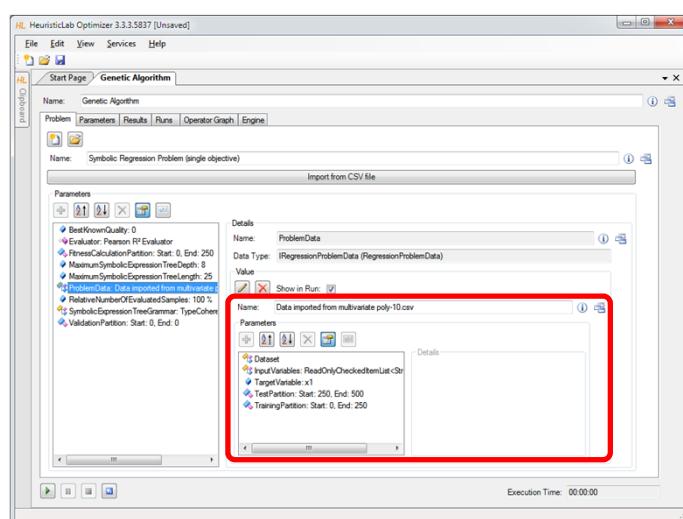
## Import Data



ICCGI 2011

69

## Inspect Data and Configure Dataset

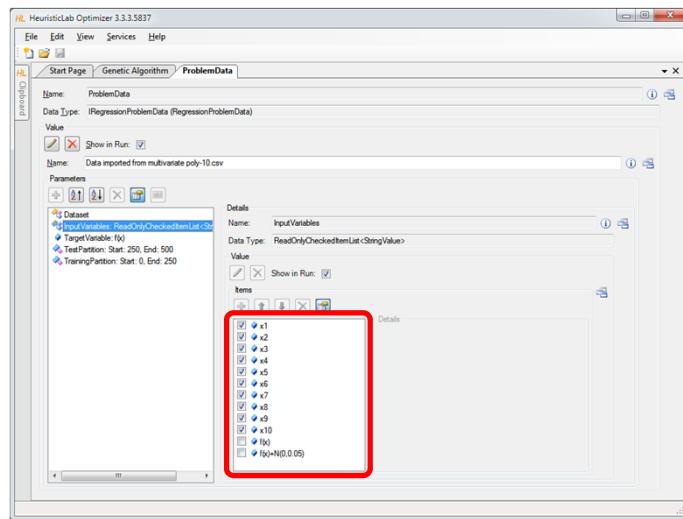


ICCGI 2011

<http://dev.heuristiclab.com>

70

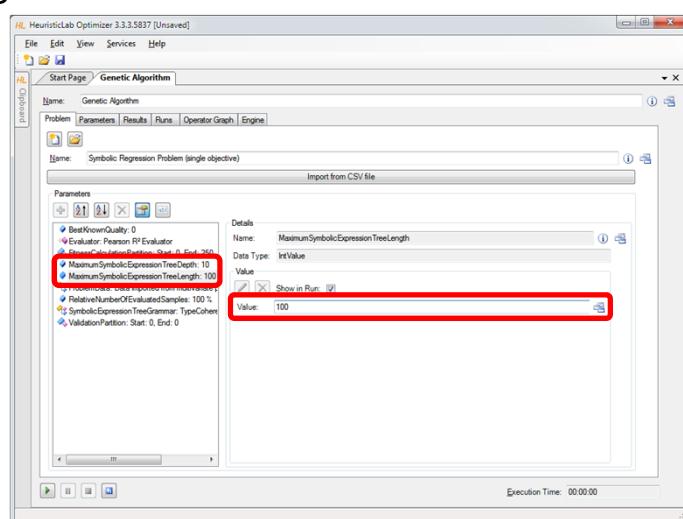
## Set Target and Input Variables



ICCGI 2011

71

## Configure Maximal Model Depth and Length

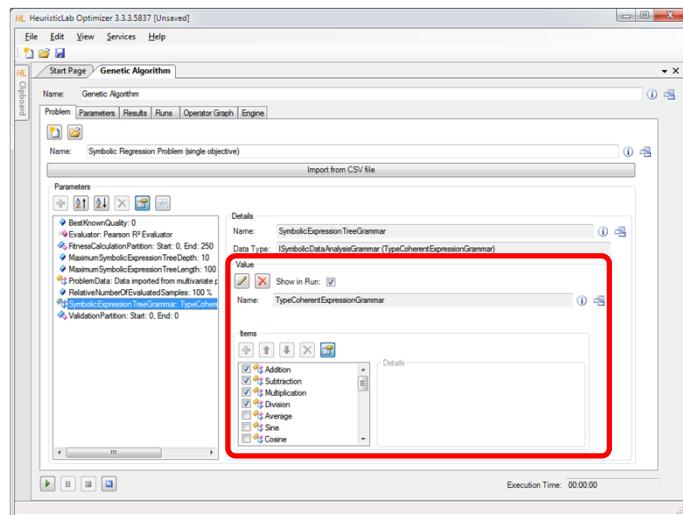


ICCGI 2011

<http://dev.heuristiclab.com>

72

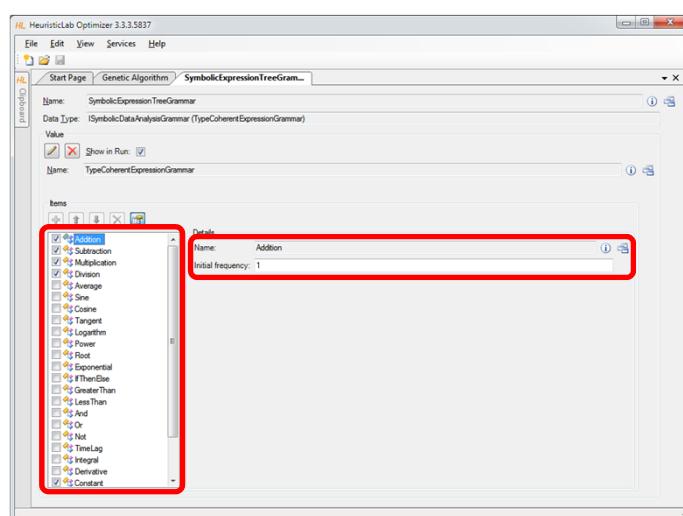
## Configure Function Set (Grammar)



ICCGI 2011

73

## Configure Function Set (Grammar)

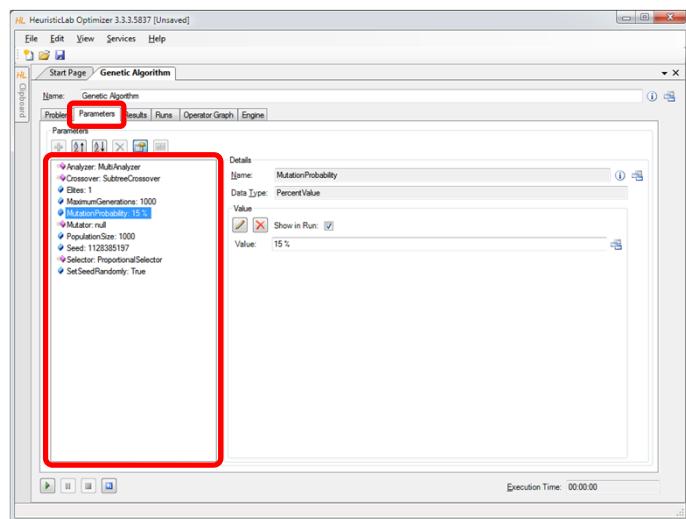


ICCGI 2011

<http://dev.heuristiclab.com>

74

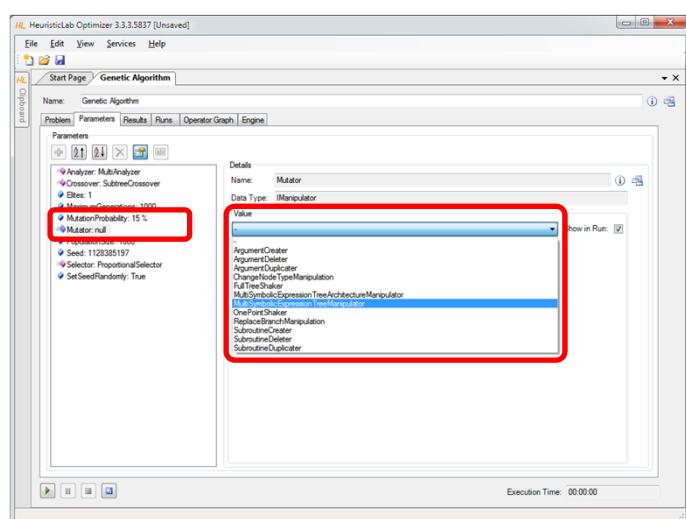
## Configure Algorithm Parameters



ICCGI 2011

75

## Configure Mutation Operator

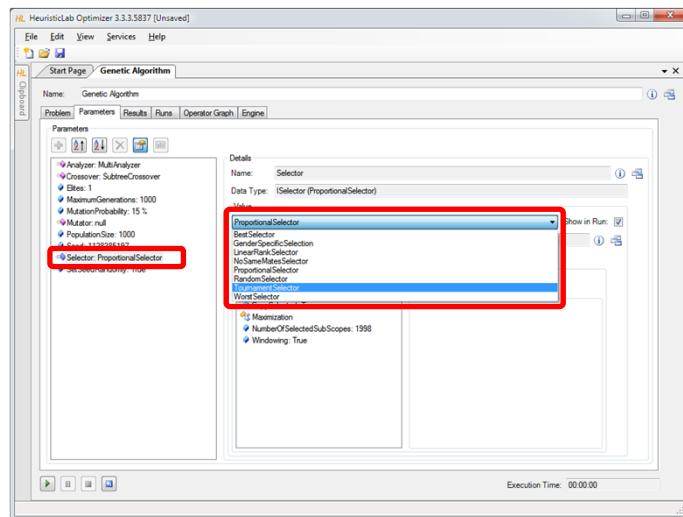


ICCGI 2011

http://dev.heuristiclab.com

76

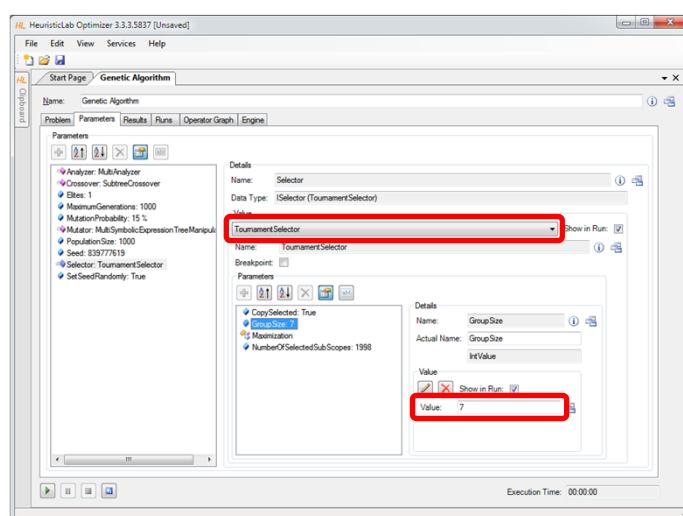
## Configure Selection Operator



ICCGI 2011

77

## Configure Tournament Group Size

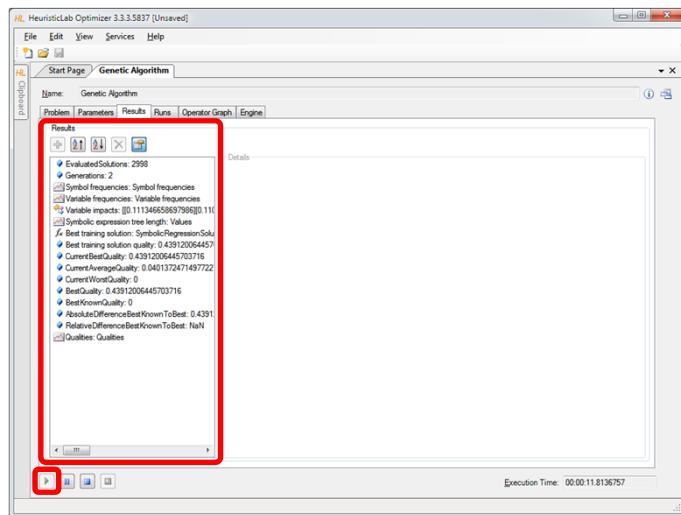


ICCGI 2011

<http://dev.heuristiclab.com>

78

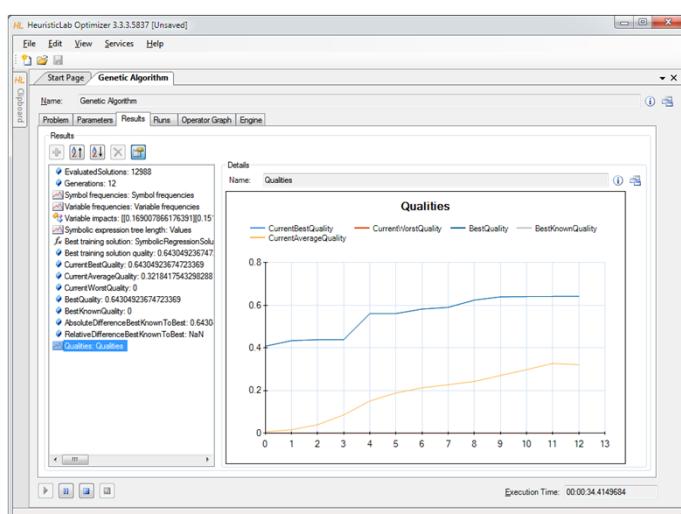
## Start Algorithm and Inspect Results



ICCGI 2011

79

## Inspect Quality Chart

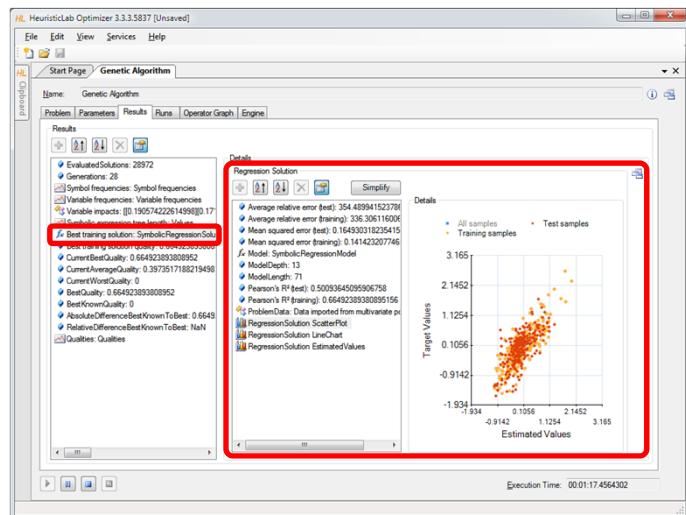


ICCGI 2011

<http://dev.heuristiclab.com>

80

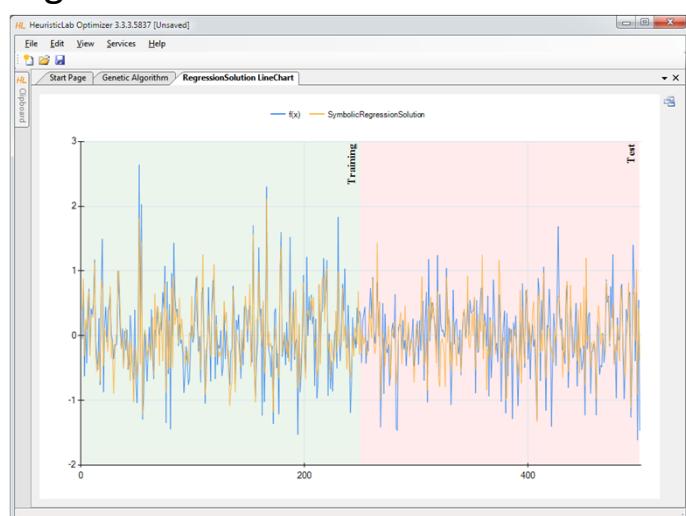
## Inspect Best Model on Training Partition



ICCGI 2011

81

## Inspect Linechart of Best Model on Trainingset

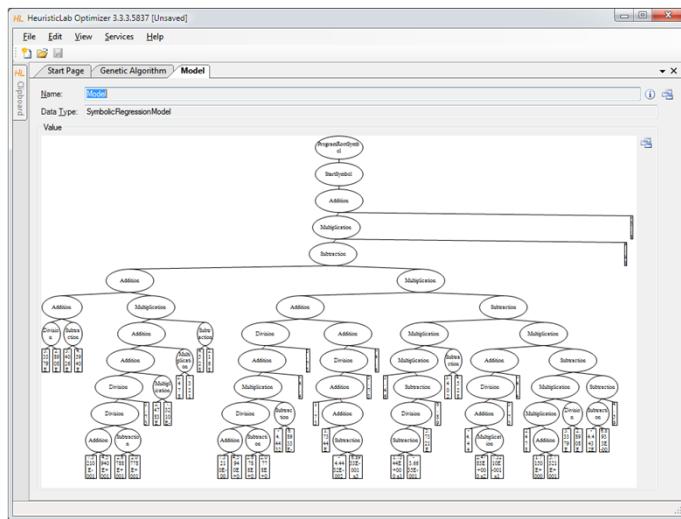


ICCGI 2011

<http://dev.heuristiclab.com>

82

## Inspect Structure of Best Model on Trainingset

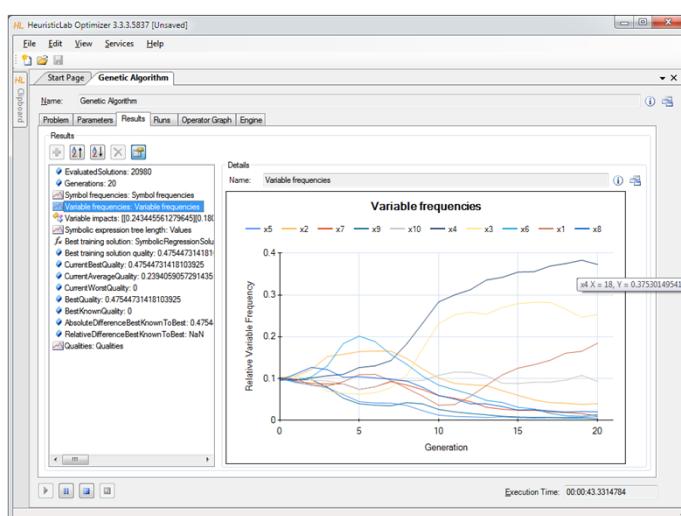


ICCGI 2011

<http://dev.heuristiclab.com>

83

## Inspect Variable Frequency Chart

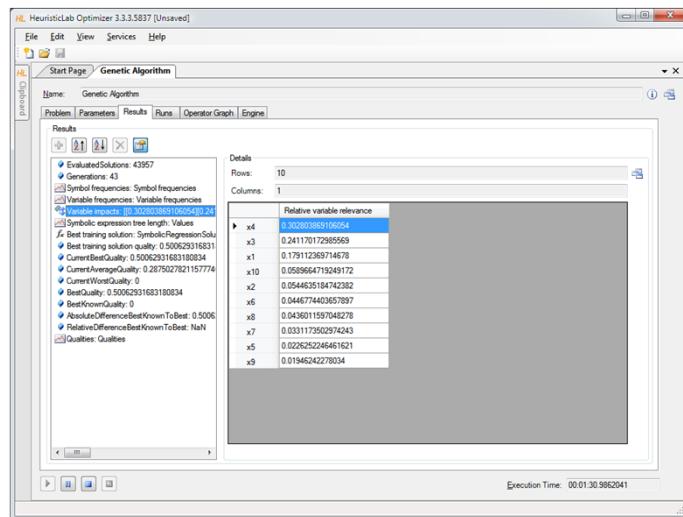


ICCGI 2011

<http://dev.heuristiclab.com>

84

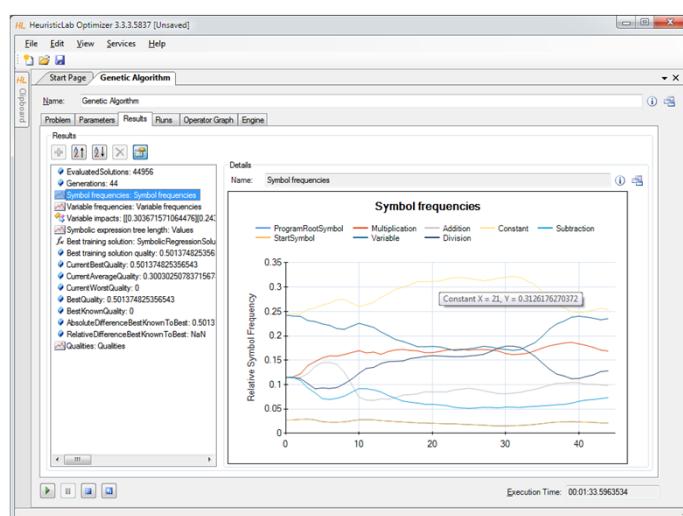
## Inspect Variable Impacts



ICCGI 2011

85

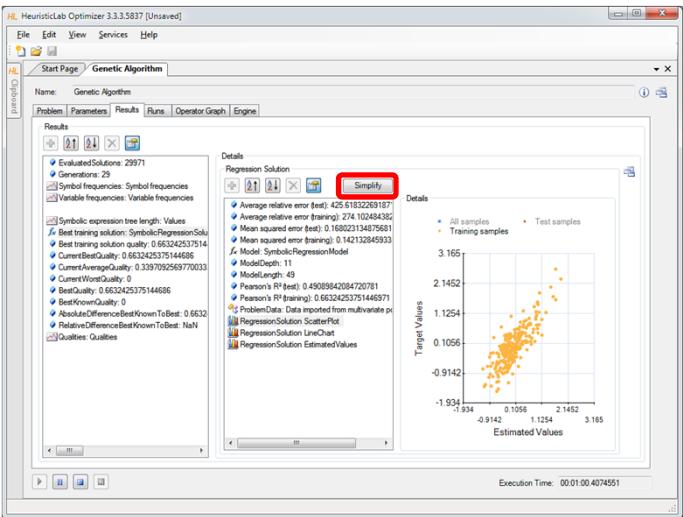
## Inspect Symbol Frequencies



ICCGI 2011

86

## Detailed Model Analysis and Simplification



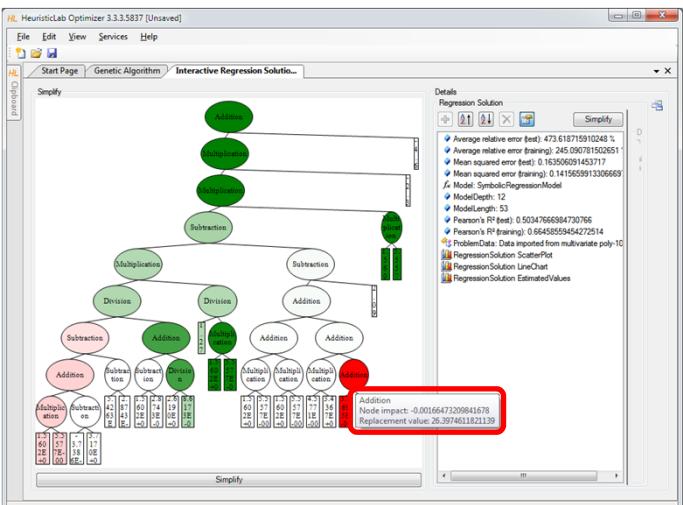
The screenshot shows the HeuristicLab Optimizer interface. On the right, a scatter plot displays 'Target Values' on the y-axis (ranging from -0.934 to 3.165) against 'Estimated Values' on the x-axis (ranging from -0.934 to 3.165). The plot includes data points for 'All samples', 'Training samples', and 'Test samples'. On the left, a tree structure shows the 'Regression Solution' with various nodes like 'Addition', 'Subtraction', 'Multiplication', and 'Division'. A red box highlights the 'Simplify' button in the toolbar. The 'Details' pane on the right lists numerous model statistics, including:

- EvaluatedSolutions: 29571
- Generations: 29
- Symbol frequencies: Symbol Frequencies
- Variable frequencies: Variable Frequencies
- Symbolic expression tree length: Values
- Best training solution: SymbolicRegressionSolution
- Best training solution quality: 0.653242537514
- CurrentBestQuality: 0.653242537514486
- CurrentWorstQuality: 0.339705259770033
- CurrentWorstQuality: 0
- BestKnownQuality: 0
- AbsoluteDifferenceBestKnownToBest: 0.6532
- RelativeDifferenceBestKnownToBest: NaN
- Qualities: Qualities

Execution Time: 00:01:00.4074551

ICCGI 2011 <http://dev.heuristiclab.com> 87

## Symbolic Simplification and Node Impacts



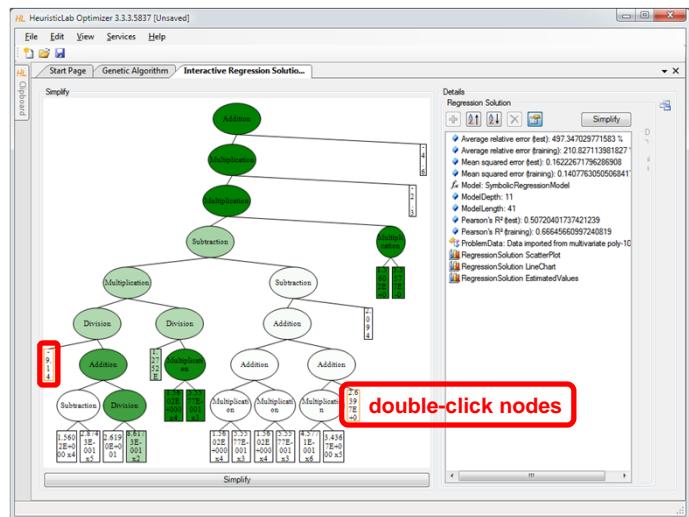
The screenshot shows the HeuristicLab Optimizer interface with an 'Interactive Regression Solution' window. On the left, a symbolic regression tree is displayed with nodes labeled 'Addition', 'Subtraction', 'Multiplication', and 'Division'. A red box highlights a specific 'Addition' node at the bottom right of the tree. The 'Details' pane on the right provides statistics for this node, including:

- Average relative error (test): 473.618715910248 %
- Average relative error (train): 245.09781502651
- Mean squared error (test): 0.11606091453717
- Mean squared error (train): 0.1415639913306689
- Model: SymbolicRegressionModel
- ModelDepth: 12
- ModelLength: 53
- Pearson's R<sup>2</sup> (test): 0.504765884730795
- Pearson's R<sup>2</sup> (train): 0.6454655454272514
- ProblemData: Data imported from multivariate poly-1C
- RegressionSolution ScatterPlot
- RegressionSolution LineChart
- RegressionSolution EstimatedValues

The highlighted node has a 'Node impact: -0.00166473209841678' and a 'Replacement value: 26.3974651821139'.

ICCGI 2011 <http://dev.heuristiclab.com> 88

## Manual Simplification

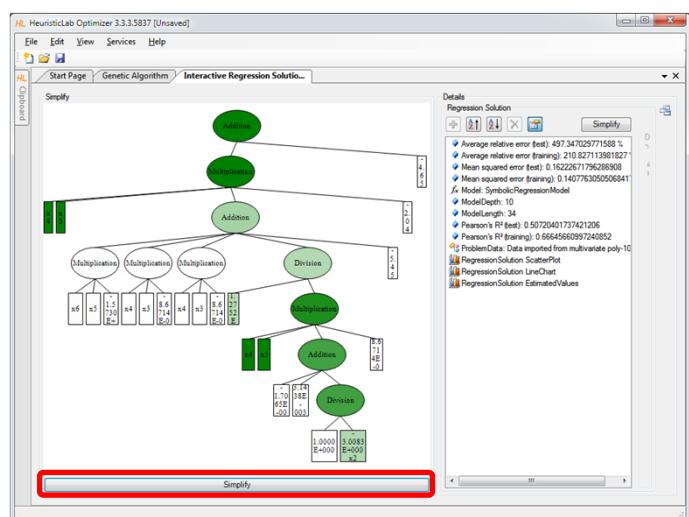


ICCGI 2011

<http://dev.heuristiclab.com>

89

## Automatic Symbolic Simplification



ICCGI 2011

<http://dev.heuristiclab.com>

90

## LaTeX Export



HeuristicLab Optimizer 3.3.3.5837 [Unsaved]

Start Page | Genetic Algorithm | Interactive Regression Solution S... | Model

File Edit View Services Help

Model

Data Type: SymbolicRegressionModel

Value

Formatter: LaTeX String Formatter

```
% needs \usepackage{amsmath}
\begin{aligned}
\text{Result} &= \text{Vebl}(c_{[0]}x44)\text{cdot}c_{[1]}x34)\text{cdot}c_{[2]}x54)\text{cdot}c_{[3]}x54)\text{cdot}c_{[4]}x44)\text{cdot}c_{[5]}x34)\text{cdot}c_{[6]}x34)\text{cdot}c_{[7]}x34)\text{cdot}c_{[8]}x44)\text{cdot}c_{[9]}x34)\text{cdot}c_{[10]}x44)\text{cdot}c_{[11]}x44)\text{cdot}c_{[12]}x44)\text{cdot}c_{[13]}x34)\text{cdot}c_{[14]}x44)\text{cdot}c_{[15]}x44) + \text{cdot}c_{[16]}x44)\text{cdot}c_{[17]}x24)\text{cdot}c_{[18]}x44) \\
&+ \text{Vebl}(c_{[11]}x44)\text{cdot}c_{[12]}x44)\text{cdot}c_{[13]}x34)\text{cdot}c_{[14]}x44) + \text{cdot}c_{[15]}x44) + \text{cdot}c_{[16]}x44)\text{cdot}c_{[17]}x24)\text{cdot}c_{[18]}x44) \\
&+ c_{[4]}x44 + 1.57302367616477 \\
&+ c_{[7]}x44 + 0.86713725013337 \\
&+ c_{[10]}x44 + 0.003832012161288 \\
&+ c_{[17]}x44 + 0.86713725013337 \\
&+ c_{[18]}x44 + 0.003832012161288 \\
&+ c_{[20]}x44 + 0.20448330755849 \\
&+ c_{[21]}x44 + 0.0465339907207764
\end{aligned} \quad (13)

```

*Result* =  $x4(t) \cdot x3(t) \cdot c_{20}$

$$\left( x6(t) \cdot x5(t) \cdot c_4 + x4(t) \cdot x3(t) \cdot c_7 + x4(t) \cdot x3(t) \cdot c_{10} + \frac{c_{11}x1(t)}{x4(t) \cdot x3(t) \cdot (c_{14}x4(t) + c_{15}x5(t) + \frac{1}{c_{17}x2(t)}) \cdot c_{18}} + c_{19} \right) + c_{21}$$

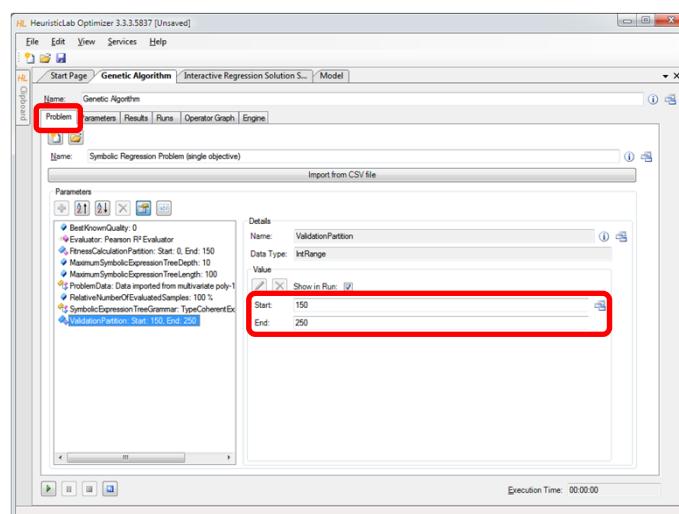
$c_4 = -1.57302367616477$  (15)  
 $c_7 = -0.86713725013337$  (16)  
 $c_{10} = 1.57302367616477$  (17)  
 $c_{11} = -0.003832012161288$  (18)  
 $c_{14} = 0.86713725013337$  (19)  
 $c_{15} = 0.003832012161288$  (20)  
 $c_{17} = -0.86713725013337$  (21)  
 $c_{18} = -0.86713725013337$  (22)  
 $c_{19} = -5.45190909490249$  (23)  
 $c_{21} = -0.20448330755849$  (24)  
 $c_{20} = -0.0465339907207764$  (25)

ICCGI 2011

91

<http://dev.heuristiclab.com>

## Configuration of Validation Partition

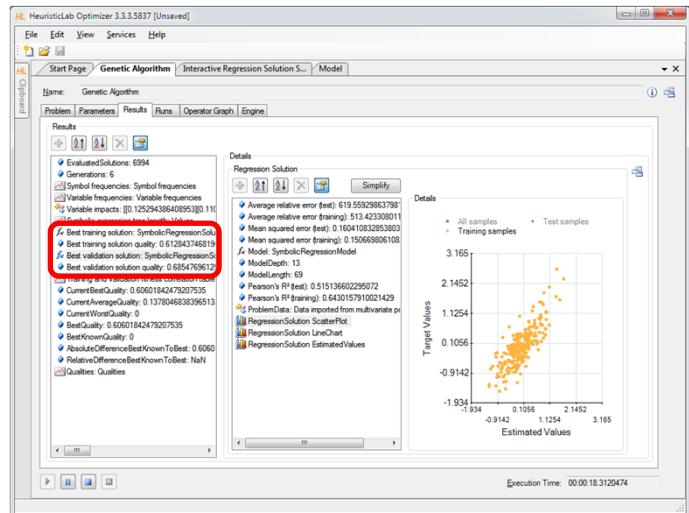


ICCGI 2011

92

<http://dev.heuristiclab.com>

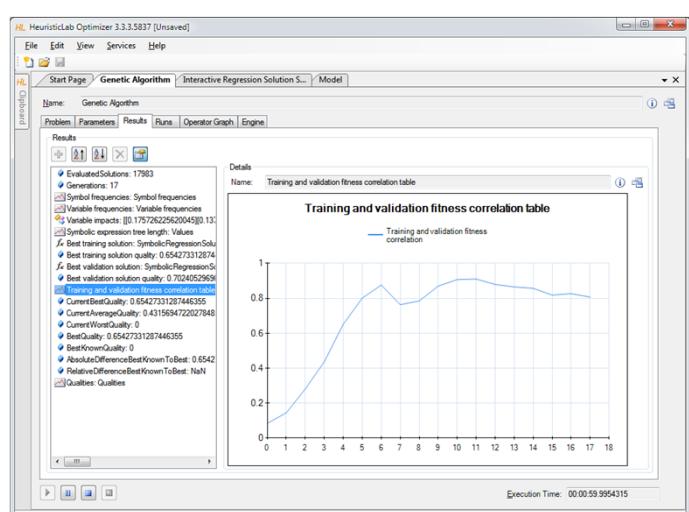
## Inspect Best Model on Validation Partition



ICCGI 2011

93

## Inspect Linechart of Correlation of Training and Validation Fitness



ICCGI 2011

http://dev.heuristiclab.com

94

## Case Study: Classification



- Real world medical dataset (*Mammographic Mass*) from UCI Machine Learning Repository (Frank & Asuncion)
  - data from non-invasive mammography screening
  - variables:
    - patient age
    - visual features of inspected mass lesions: shape, margin, density
  - target variable: severity (malignant, benign)
- Download:
 

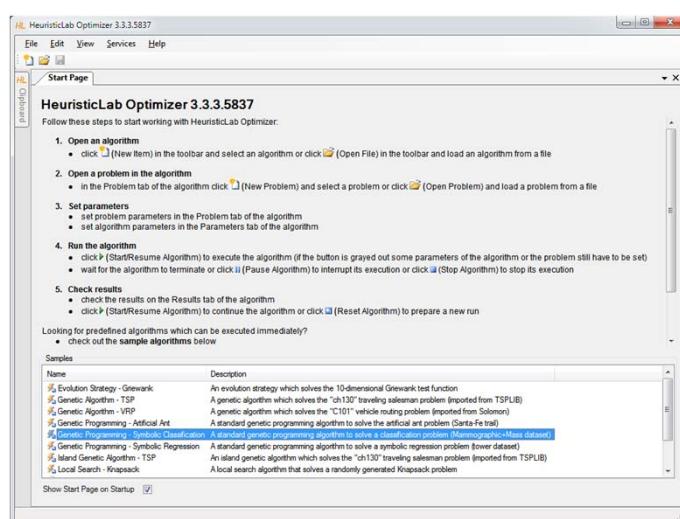
<http://dev.heuristiclab.com/AdditionalMaterial#ICCGI2011>

ICCGI 2011

<http://dev.heuristiclab.com>

95

## Open Sample

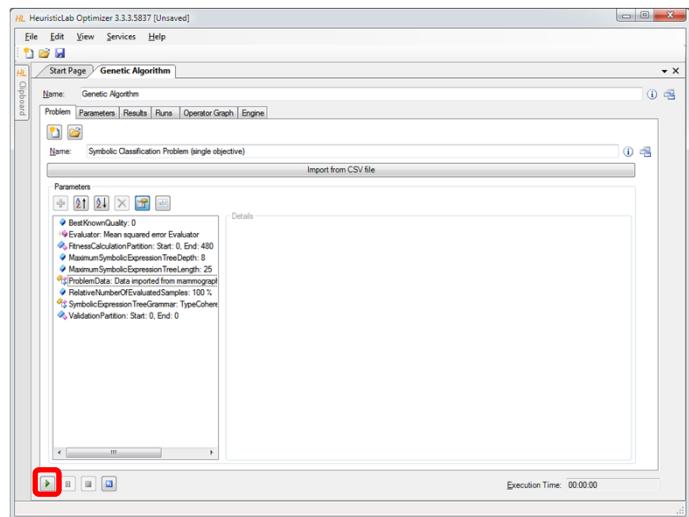


ICCGI 2011

<http://dev.heuristiclab.com>

96

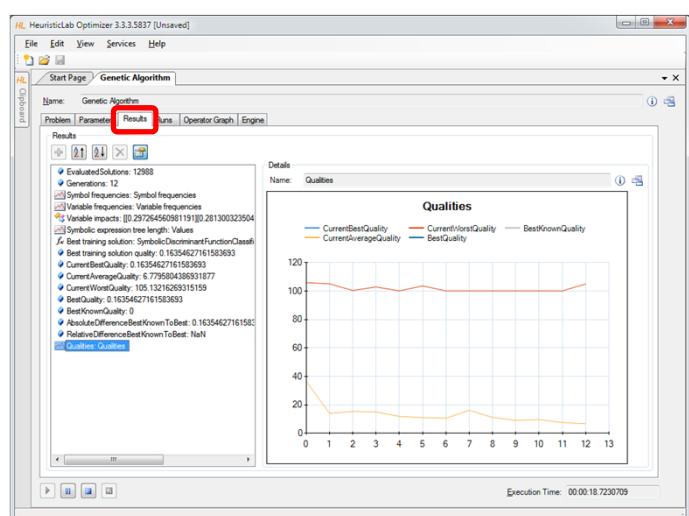
## Configure and Run Algorithm



ICCGI 2011

97

## Inspect Quality Linechart

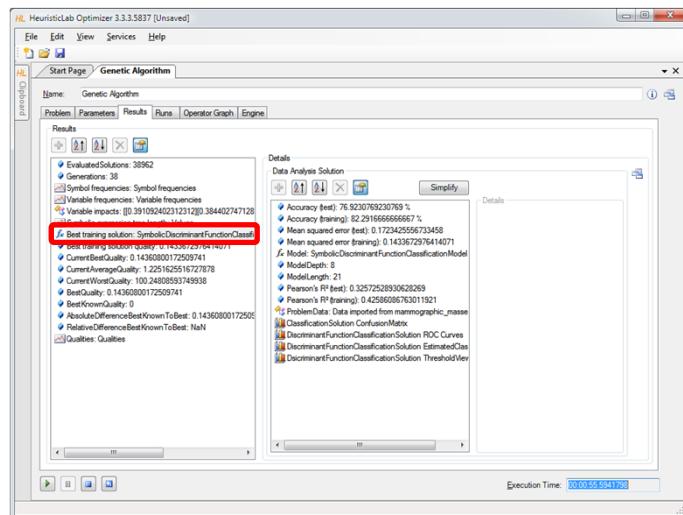


ICCGI 2011

<http://dev.heuristiclab.com>

98

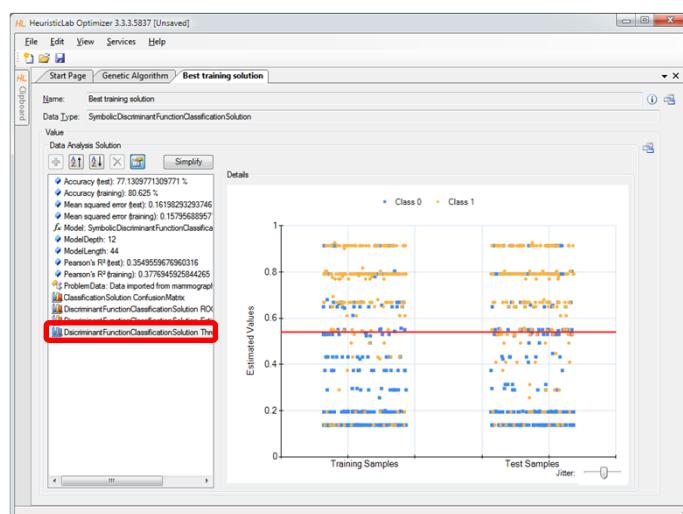
## Inspect Best Training Solution



ICCGI 2011

99

## Inspect Model Output and Thresholds

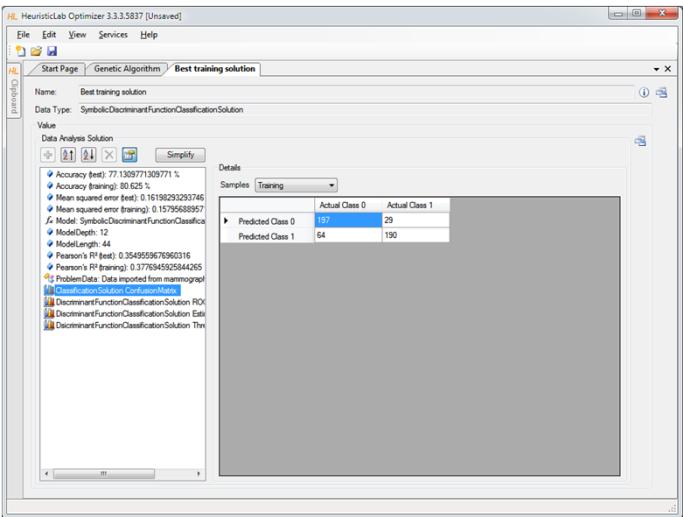


ICCGI 2011

<http://dev.heuristiclab.com>

100

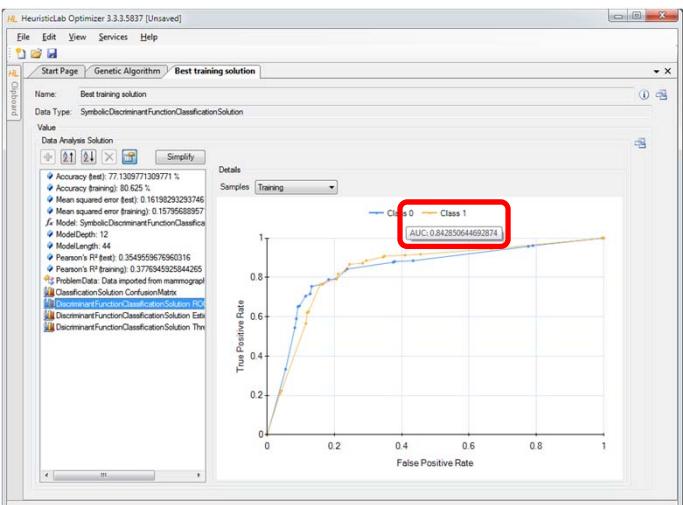
## Inspect Confusion Matrix



		Actual Class 0	Actual Class 1
Predicted Class	0	1	
Predicted Class 0	157	29	
Predicted Class 1	64	190	

ICCGI 2011 <http://dev.heuristiclab.com> 101

## Inspect ROC Curve



True Positive Rate

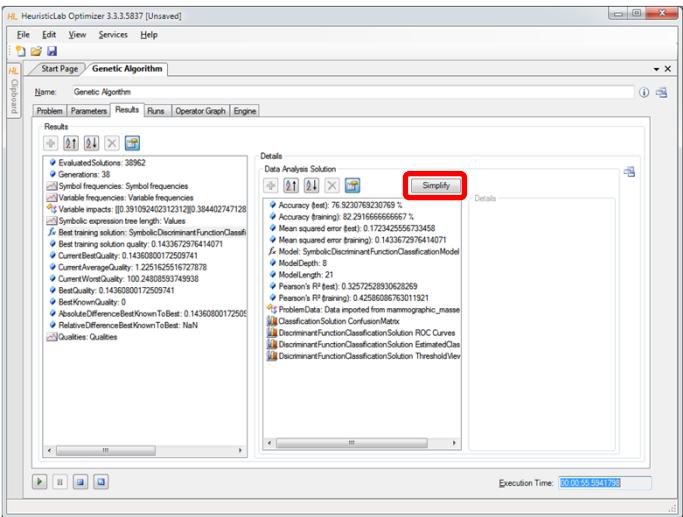
False Positive Rate

Class 0 Class 1

AUC: 0.842850644692874

ICCGI 2011 <http://dev.heuristiclab.com> 102

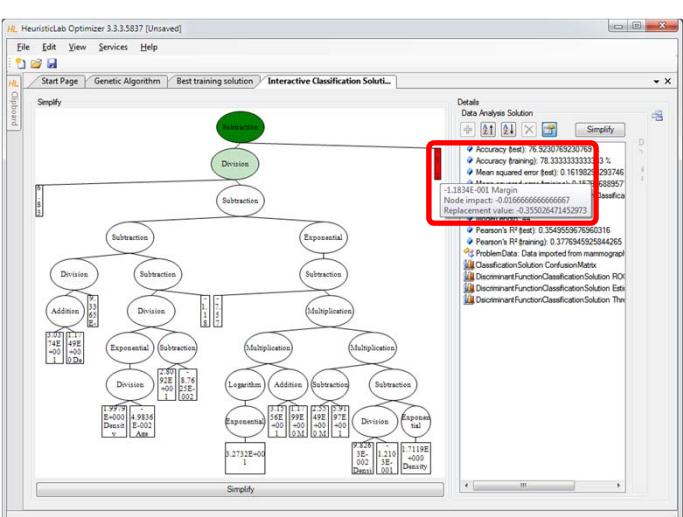
## Analyse and Simplify Best Training Solution



The screenshot shows the HeuristicLab Optimizer interface. The main window title is "HeuristicLab Optimizer 3.3.3.5837 [Unsaved]". The "Start Page" tab is selected. In the center, there is a "Results" pane showing a list of evaluated solutions and their details. A "Simplify" button is highlighted with a red box. To the right, a "Details" pane displays various performance metrics and model statistics. The bottom status bar shows "Execution Time: 00:00:55.934962".

ICCGI 2011 <http://dev.heuristiclab.com> 103

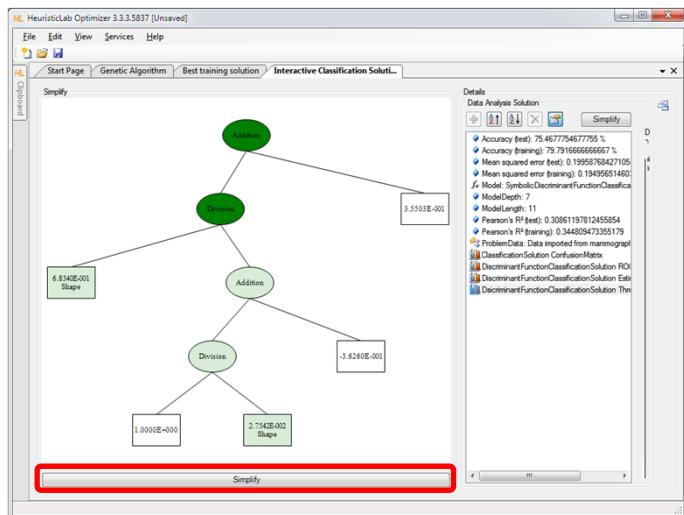
## Analyse and Simplify Model



The screenshot shows the HeuristicLab Optimizer interface with the "Best training solution" tab selected. The main area displays a complex mathematical expression tree. A specific node in the tree is highlighted with a red box. To the right, a "Details" pane shows performance metrics and model statistics. The bottom status bar shows "Execution Time: 00:00:55.934962".

ICCGI 2011 <http://dev.heuristiclab.com> 104

## Symbolically Simplified Model



ICCGI 2011

<http://dev.heuristiclab.com>

105

## Some Additional Features



- HeuristicLab Hive
  - parallel and distributed execution of algorithms and experiments on many computers in a network
- Optimization Knowledge Base (OKB)
  - database to store algorithms, problems, parameters and results
  - open to the public
  - open for other frameworks
  - analyze and store characteristics of problem instances and problem classes
- External solution evaluation and simulation-based optimization
  - interface to couple HeuristicLab with other applications (MatLab, AnyLogic, ...)
  - supports different protocols (command line parameters, TCP, ...)
- Parameter grid tests and meta-optimization
  - automatically create experiments to test large ranges of parameters
  - apply heuristic optimization algorithms to find optimal parameter settings for heuristic optimization algorithms



ICCGI 2011

<http://dev.heuristiclab.com>

106

## Planned Features



- Algorithms & Problems
  - steady-state genetic algorithm
  - unified tabu search for vehicle routing
  - scatter search
  - ...
- Cloud Computing
  - port HeuristicLab Hive to Windows Azure
- Linux
  - port HeuristicLab to run on Mono and Linux machines
- Have a look at the HeuristicLab roadmap
  - <http://dev.heuristiclab.com/trac/hl/core/roadmap>
- Any other ideas, requests or recommendations?
  - please write an e-mail to [support@heuristiclab.com](mailto:support@heuristiclab.com)

ICCGI 2011

<http://dev.heuristiclab.com>

107

## HeuristicLab Team



Heuristic and Evolutionary Algorithms Laboratory (HEAL)  
 School of Informatics, Communications and Media  
 Upper Austria University of Applied Sciences

Softwarepark 11  
 A-4232 Hagenberg  
 AUSTRIA

WWW: <http://heal.heuristiclab.com>



**HEAL**

Heuristic and Evolutionary  
 Algorithms Laboratory



ICCGI 2011

<http://dev.heuristiclab.com>

108

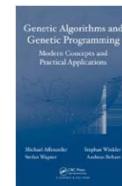
## Suggested Readings



- S. Voß, D. Woodruff (Edts.)  
**Optimization Software Class Libraries**  
 Kluwer Academic Publishers, 2002



- M. Affenzeller, S. Winkler, S. Wagner, A. Beham  
**Genetic Algorithms and Genetic Programming  
Modern Concepts and Practical Applications**  
 CRC Press, 2009



ICCGI 2011

<http://dev.heuristiclab.com>

109

## Bibliography



- S. Wagner, M. Affenzeller  
**HeuristicLab: A generic and extensible optimization environment**  
 Adaptive and Natural Computing Algorithms, pp. 538-541  
 Springer, 2005
- S. Wagner, S. Winkler, R. Braune, G. Kronberger, A. Beham, M. Affenzeller  
**Benefits of plugin-based heuristic optimization software systems**  
 Computer Aided Systems Theory - EUROCAST 2007, Lecture Notes in Computer Science, vol. 4739, pp. 747-754  
 Springer, 2007
- S. Wagner, G. Kronberger, A. Beham, S. Winkler, M. Affenzeller  
**Modeling of heuristic optimization algorithms**  
 Proceedings of the 20th European Modeling and Simulation Symposium, pp. 106-111  
 DIPTEM University of Genova, 2008
- S. Wagner, G. Kronberger, A. Beham, S. Winkler, M. Affenzeller  
**Model driven rapid prototyping of heuristic optimization algorithms**  
 Computer Aided Systems Theory - EUROCAST 2009, Lecture Notes in Computer Science, vol. 5717, pp. 729-736  
 Springer, 2009
- S. Wagner  
**Heuristic optimization software systems - Modeling of heuristic optimization algorithms in the HeuristicLab software environment**  
 Ph.D. thesis, Johannes Kepler University Linz, Austria, 2009.
- S. Wagner, A. Beham, G. Kronberger, M. Kommenda, E. Pitzer, M. Kofler, S. Vonolffen, S. Winkler, V. Dorfer, M. Affenzeller  
**HeuristicLab 3.3: A unified approach to metaheuristic optimization**  
 Actas del séptimo congreso español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB'2010), 2010
- Detailed list of all publications of the HEAL research group: <http://research.fh-oe.at/de/orgunit/detail/356#showpublications>

ICCGI 2011

<http://dev.heuristiclab.com>

110

## Questions & Answers



<http://dev.heuristiclab.com>

[support@heuristiclab.com](mailto:support@heuristiclab.com)