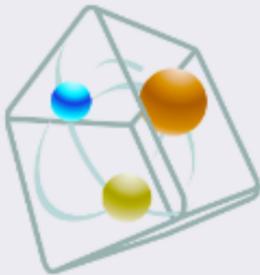


Works 2013



xlOptimizer

Version 2.0.0

© 2012 TechnoLogismiki

USER GUIDE

www.technologismiki.com

 | **TECHNO logismiki**

Advanced Technical Software

5 Imitou str, 15561, Cholargos, Athens, Greece
tel: ++30 210 65 64 147 - fax: ++30 210 65 48 461
www.technologismiki.com - info@technologismiki.com

xOptimizer

TechnoLogismiki

xlOptimizer

© 2012 TechnoLogismiki

Publisher

TechnoLogismiki

Editors

Fotis Fotopoulos

Aristotelis Charalampakis

Technical Assistance

Antigoni Egglezou

All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher. You are entitled to one (1) paper copy for your own reference.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Printed: September 2012 in Athens, Greece.

Table of Contents

Chapter I About the program

1	What does the program do?.....	8
2	Minimum requirements.....	9
3	Technical support.....	10

Chapter II User Interface

1	Main Window	12
2	Status/Progress Window.....	14

Chapter III File

1	File menu	17
2	New project.....	17
3	Open project.....	17
4	Save project.....	18
5	Save project as.....	18
6	Exit	19

Chapter IV Edit

1	Edit menu.....	21
2	Undo.....	21
3	Redo.....	21
4	Options.....	22
	General preferences	22
	Customize toolbar	24

Chapter V Data

1	Data menu.....	26
2	Variables.....	26
	Add	26
	Delete	27
	Properties	27
	Move top	28
	Move up	28
	Move down	29
	Move bottom	29
3	Constraints.....	29
	Add	29
	Delete	30
	Properties	30

Move top	31
Move up	31
Move down	31
Move bottom	32
4 Scenarios.....	32
Add	32
Wizard: Standard GA [1 solution]	32
Wizard: Customizable GA	33
Wizard: Sawtooth GA	39
Wizard: Micro GA	45
Wizard: Simple PSO	50
Wizard: Enhanced PSO	54
Wizard: DE	58
Delete	62
Properties	62
Clone	62
Move top	63
Move up	63
Move down	63
Move bottom	64

Chapter VI Window

1 Window menu.....	66
2 Auto arrange.....	66
3 Arrange.....	66
4 Status/progress window.....	66

Chapter VII Help

1 Help menu.....	68
2 Contents.....	68
3 User guide.....	68
4 Tutorials.....	68
5 Tip of the day.....	69
6 Unit conversion.....	70
7 TechnoLogismiki website.....	70
8 Buy products.....	70
9 TechnoLogismiki NOMOS.....	70
10 TechnoLogismiki Live!	70
11 About the program.....	70

Chapter VIII Appendix

1 Genetic Algorithms (GA).....	72
2 Sawtooth GA.....	74
3 Micro GA.....	74
4 Simple Particle Swarm Optimization (PSO).....	75

5	Enhanced PSO.....	76
6	Differential Evolution (DE).....	77
	Keyword Index	79

Chapter

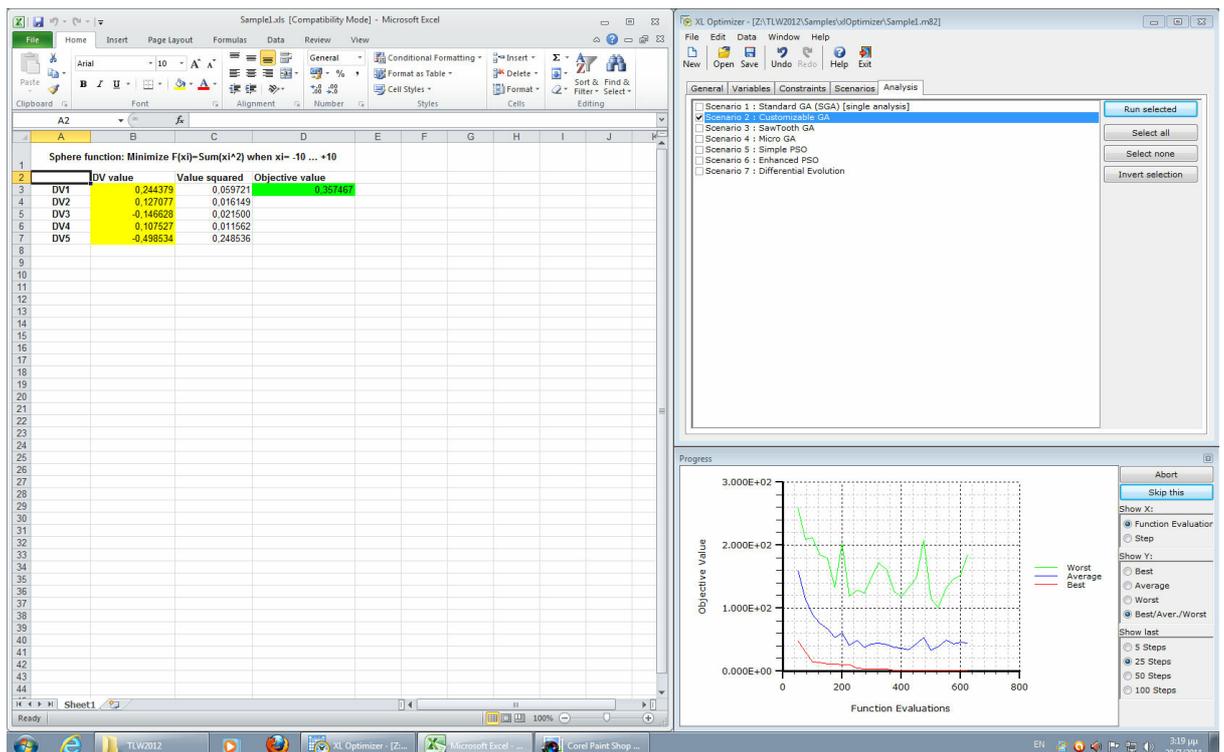


1 About the program

1.1 What does the program do?

This program is a **generic optimization tool** that uses Microsoft Excel (versions 2003 and newer) as a platform for the definition of the problem at hand. Practically any problem that can be formulated in a spreadsheet can be tackled by this program. Examples include **problems in finance, engineering, resource allocation, scheduling, manufacturing, route finding**, etc.

This program is ideal for practitioners and researchers alike.



xlOptimizer differs radically from the built-in "solver" pack of Microsoft Excel in the sense that it implements **a host of customizable, state-of-the-art Evolutionary Algorithms (EAs)**, homogenized and arranged in an intuitive interface. These algorithms are considered to be very good "global optimizers", i.e. they are able to find very good solutions (from a practical point of view) with few function evaluations. This renders them **ideal for very difficult** (multi-parametric, non-differentiable, discontinuous, combinatorial, deceptive, etc) **and/or expensive problems** (i.e. each function evaluation may require several minutes).

Apart from the built-in functions of Excel, which are readily available, **external functions can be easily implemented through dynamic-link-libraries (dlls)**, so that virtually any problem can be formulated efficiently within a spreadsheet.

xlOptimizer is suitable for **practitioners with small or even no knowledge at all** of the inner workings of EAs. As long as the problem is formulated in a spreadsheet, it is very easy to set-up a number of scenarios and optimize, which is translated into

significant profit or improvement in performance. All the parameters controlling the algorithms are set automatically to commonly used values. Concise tips are included in each step of appropriate wizards, which guide the user. Extensive on-line and off-line support is also available.

xOptimizer is also ideal for **researchers in any scientific field** that want to use a simple yet powerful tool for optimization. All parameters controlling the behavior of each algorithm are customizable, so that experimentation can reveal the optimum set of values. Moreover, **a simple, Excel-like, built-in function evaluator is included**, so that the parameters can be adaptive and not fixed. For example, the optimum population size of a Genetic Algorithm (GA) depends on the complexity of the problem. This can be easily set-up into a function, that produces the population size based on e.g. the chromosome length. Finally, automation tools are included that facilitate the examination of the performance of the Algorithms. These include the capability of performing multiple independent runs, using specific or random seeds, and storing the results into appropriate folders. The log of each run contains all the necessary information regarding the evolution, and can be easily analyzed further using text editors. **The core modules of xOptimizer have been used in many papers published in peer-reviewed international scientific journals and proceedings of international conferences.** For more information, please contact TechnoLogismiki.

Some of the unique features of xOptimizer are:

- Handles practically **any problem** that can be formulated in a spreadsheet (Microsoft Excel 2003 or newer required), using either built-in or external functions.
- Handles **any number of variables, constraints (through customizable penalty functions) and scenarios**. These are limited only by hardware.
- The problem can be single-objective **minimization, maximization or target-value**.
- Multiple fully-customizable, state-of-the-art Evolutionary Algorithms are available: **Custom Genetic Algorithm (GA), Micro-GA, Sawtooth-GA, Simple Particle Swarm Optimization (PSO), Enhanced PSO, Differential Evolution (DE)**. All algorithms are homogenized and arranged within an intuitive interface.
- Full logs are produced, which can be further viewed and analyzed using simple text editors.
- Optional **graphical representation of the evolution** reveals valuable information such as current best/mean/worst objective values in the population, current function evaluations, evolution progress in the last evolutionary steps, etc.

If you are interested in using xOptimizer in the research or the industry, please contact TechnoLogismiki.

1.2 Minimum requirements

The minimum requirements for the usage of the programs are the following:

- Windows 2000/ XP/ 2003/ Vista/ 7 (for each case, the latest service packs,

- updates & patches must be installed)
- Pentium III 800 MHz
- 800x600 with 256 color palette
- 700 MB free disk space
- CD-Rom

If your system does not meet one or more of the above requirements, it is highly recommended that you upgrade it before installing the programs. The recommended system configuration is the following:

- Windows 2000/ XP/ 2003/ Vista/ 7 (for each case, the latest service packs, updates & patches must be installed)
- Pentium IV 2.0 GHz
- 1280x768 with 16-bit color palette
- 1.2 GB free disk space
- CD-Rom
- Internet connection

1.3 Technical support

Support through the Internet

TechnoLogismiki offers technical support 24 hours per day, 365 days per year, through the web site where you can get information on the latest programs and services.

Support by e-mail

Please use the dedicated e-mail addresses for better customer service:

- for questions regarding sales: sales@technologismiki.com
- for questions regarding the usage of programs: support@technologismiki.com
- for any other question or comment: info@technologismiki.com

The normal response time is within two business days. If your inquiry cannot be answered by e-mail, a customer service representative will contact you by telephone.

Interactive Support

Business days, 09:00 - 17:00 Eastern European Time:

- Telephone [3 lines]: ++30-210-656-4147
- Fax: ++30-210-654-8461
- Address: 5, Imittou street, Cholargos, 15561, Athens, Greece.

Chapter

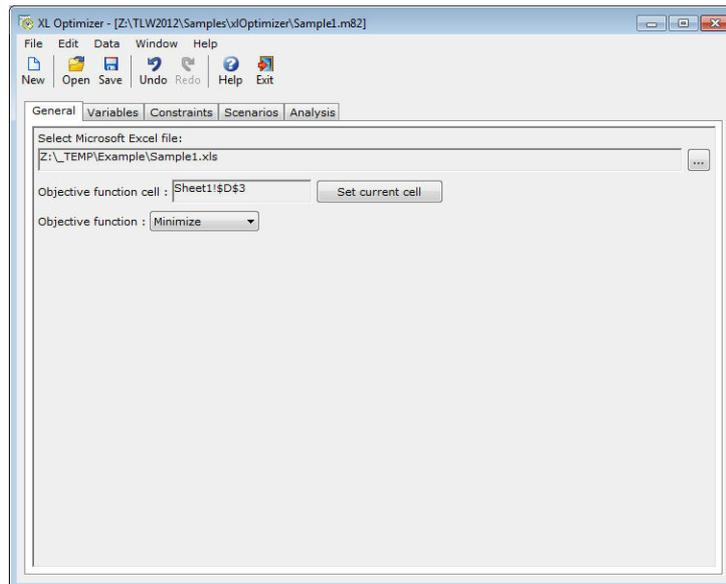


2 User Interface

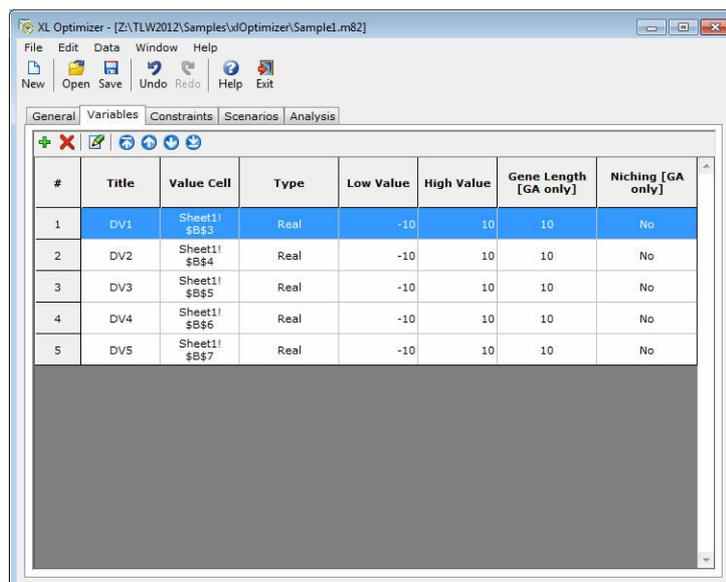
2.1 Main Window

The main window consists of a standard toolbar and five tabs that organize the data. Data input should follow the order of the tabs, as follows:

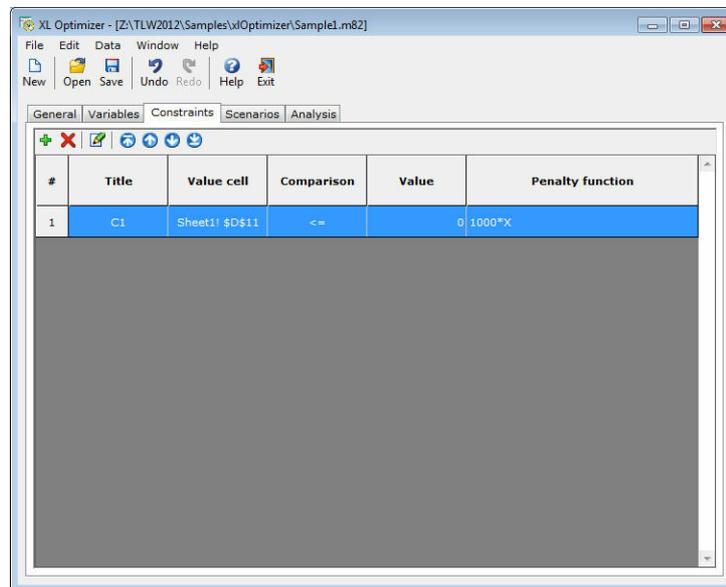
1. General: provides access to basic problem parameters, such as the associated Microsoft Excel file, the cell of the objective value, and whether the objective function is to be minimized, maximized or reach a target value.



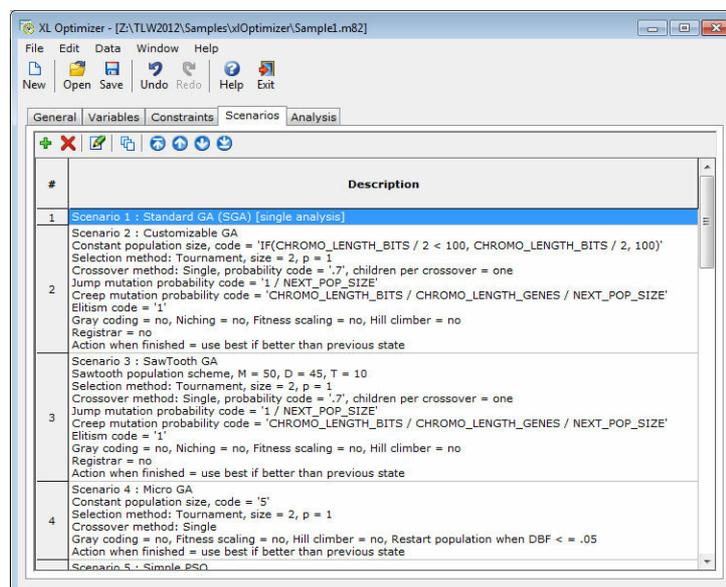
2. Variables: a list of the variables, which are to be modified in order to obtain good solutions. Each variable may be of real or integer type. Appropriate low and high boundaries also need to be set.



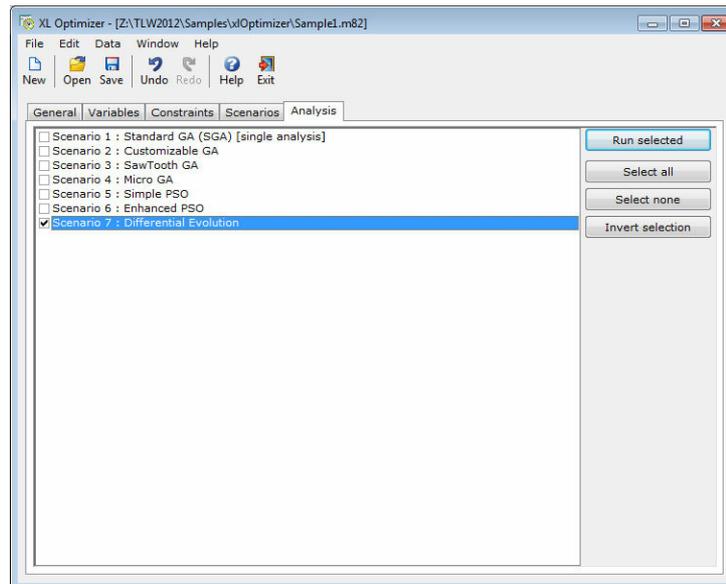
3. Constraints: a list of the constraints which are implemented in the form of customizable penalty functions.



4. Scenarios: a list of the scenarios that are set for analysis. Each scenario uses a certain EA with specific parameters. Cloning and re-arrangement of the scenarios is available.



5. Analysis: enables the user to analyze one or more scenarios that are defined in the previous tab.

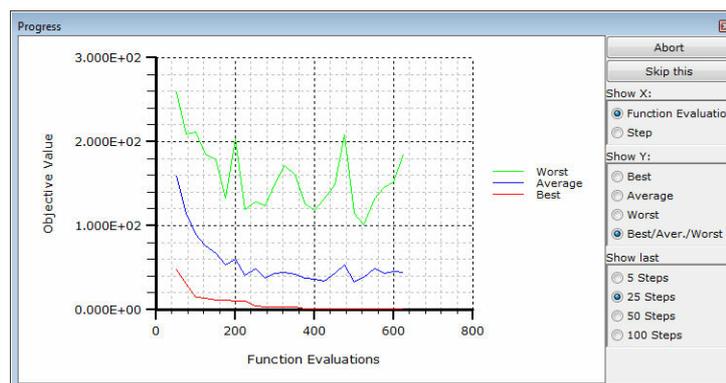


Most lists are accompanied by a small dedicated toolbar with the following commands:

- **Add:** adds an object in the specified list.
- **Delete:** deletes the selected object(s).
- **Edit:** edits the selected object.
- **Clone:** (for scenarios only) clones the selected scenario.
- **Move Top:** moves the selected object to the top of the list.
- **Move Up:** moves the selected object by one slot upwards.
- **Move Down:** moves the selected object by one slot downwards.
- **Move Last:** moves the selected object to the bottom of the list.

2.2 Status/Progress Window

The status/progress window consists of a graphical area and a column with options on the right:



The options available are the following:

- **Abort:** when an analysis is performed, this aborts the procedure.
- **Skip this:** the current analysis is skipped, and the process continues with the next analysis in line.

-
- **Show X:** selects the data that is represented in the X axis. This can be either the function evaluations so far, or the current evolutionary step.
 - **Show Y:** selects the data that is represented in the Y axis. This can be the best objective value in the current population, the average, the worst, or all together.
 - **Show last:** controls the number of past evolutionary steps that are included in the graph.

Chapter



3 File

3.1 File menu

With this menu, you can perform file operations and print reports. In the **File** menu you can select one of the following options:

- New project
- Open project
- Save project
- Save project as
- Exit

3.2 New project

With this option, a new project is started. All data, results, graphs, titles etc. of the previous project are erased.

To create a new project:

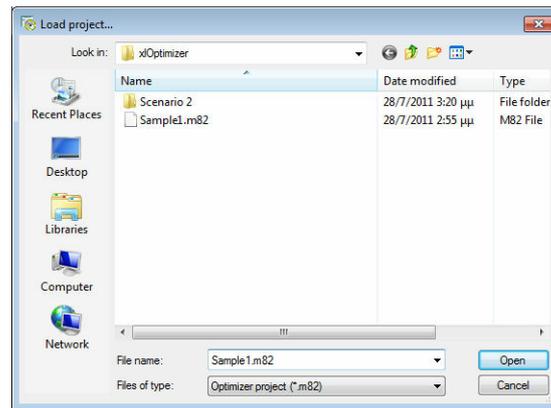
1. Select **New project** from the **File** menu.
2. If a project is already loaded and changes have been made, a warning message will appear that asks the user whether to save the changes or not.
3. The current project is erased and a new project is started.

3.3 Open project

With this option, an existing project is loaded. The project may be located locally, in a network or in an external media device such as a CD-Rom. If a project is already loaded and changes have been made, a warning message will appear that asks whether to save the changes or not. When a project is loaded, all data of the previous project are lost.

To open an existing project:

1. Select **Open project** from the **File** menu.
2. Select the path of the file.
3. Select the file type from the **Files of type** drop-down list. The default option is "Optimizer project" with the extension .m82.
4. Select the file by clicking on it.
5. Select **Open** to open the selected file. Select **Cancel** to cancel the operation.



NOTE: You can find sample projects in the installation folder of the program:
C:\Program Files\TechnoLogismiki\TLW2013\Samples\xLOptimizer

Supported file types

- **M82** (Optimizer project): Files created by versions 2012 and 2013 of xLOptimizer.
- **BCK** (Backup files): If you have selected from program options the creation of backup copy when a file is loaded, then the file can be loaded by selecting Backup files (*.bck) from the Files of type drop-down list.
- ***.*** (All files): Displays all files in the current folder.

NOTE: If a message "Could not load project. File may be corrupt or saved by an unknown or incompatible version of the program" then either you are trying to load a project that does not belong to this program or the file is used (and locked) by another process in your computer.

3.4 Save project

With this option, you can save all data of a project into a file. The file may be saved locally, in a network location or in an external media device such as a disk.

The filename and path will be asked only the first time you are saving a project. When the filename and path are set, all subsequent saves will be made to the same file.

When you want to rename a file or save it in a new location, use Save project as... from the **File** menu.

To save the current project:

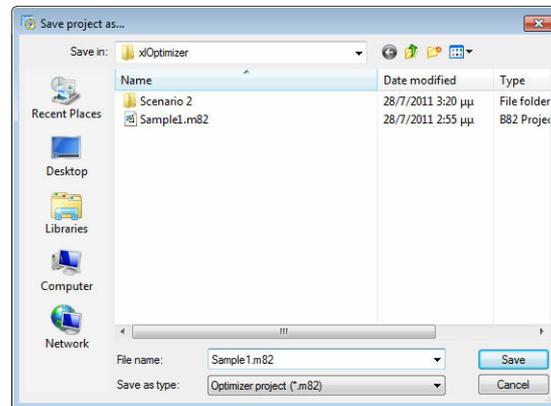
1. Select **Save project** from the **File** menu.
2. If the location of the file is already set, the project is saved to this file without any messages. If the filename is not set, a dialog box will appear that allows the selection of the filename and path.

3.5 Save project as

With this option, the current project is saved just as in the case of Save project, but with the difference that the name and/or location of the file can be changed. In this way, you can create backup files or move a project to another media device.

To save a project with another name and/or to another location:

1. Select **Save project as** from the **File** menu.
2. Select the path of the file.
3. Type the filename in the **File name** text box.
4. Select **Save** to save the project with the selected filename and path. Select **Cancel** to cancel the operation.



NOTE: If a file with the same name and in the same path already exists, a warning message will appear that asks whether to overwrite the file or not. If you answer Yes, then the existing file is erased and the new file takes its place. If you answer No, the existing file remains intact but NO changes of the current project are saved.

3.6 Exit

With this option, you can exit the program. If there are changes in the current project that have not been saved then the program will:

- either ask the user to save the changes
- or save the changes
- or ignore the changes

depending on what you have selected in General preferences.

To exit the program:

1. Select **Exit** from **File** menu.
2. If you are asked whether to save the changes or not, you can save changes or ignore them.
3. The program is terminated.

Chapter

IV

4 Edit

4.1 Edit menu

With this menu, you can perform basic operations regarding data. In the **Edit** menu you can select one of the following options:

- Undo
- Redo
- Options
 - General preferences
 - Customize toolbar

4.2 Undo

Undo cancels the last committed change in the project.

To cancel the last committed change:

1. Select **Undo** from the **Data** menu.
2. The last committed change is canceled.

To cancel an undo command, you may use the redo function which is described below. Redo becomes available once undo is used.

It is possible to undo more than one recent changes and to redo them, by following the step described above. The number of actions that are kept in memory and may be undone or redone is 20 by default. This means that the program is able to keep track of up to 20 successive changes and undo them. This number may change for all programs, using the option in the main menu. For more information, please consult main menu user guide.

NOTE: Some changes cannot be undone like the new project or the save project functions.

4.3 Redo

Redo cancels the latest undo command.

To redo the latest change that was undone:

1. Select **Redo** from the **Data** menu.
2. The latest undone change is redone.

To undo a redo, you may use the undo command.

It is possible to redo more than one changes that were previously undone by following the steps described above. The number of actions that are kept in memory and may be undone or redone is 20 by default. This means that the program is able to keep track of up to 20 successive changes that are undone and redo them. This number may change for all programs, using the option in the main menu. For more information, please consult main menu user guide.

4.4 Options

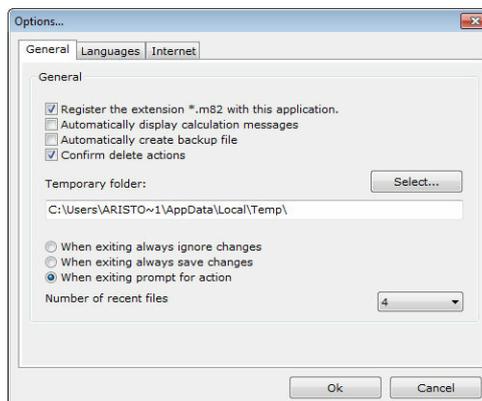
4.4.1 General preferences

With this option, you can modify the general preferences of the program.

To modify the general preferences:

1. Select **Options** from the **Data** menu.
2. Select **General preferences** from the **Options** menu.
3. The general preferences dialog box appears. The preferences are grouped into four tabs. You can select a tab by clicking on its name.

General Tab



This tab contains general preferences regarding the usage of the program.

Check **Register the extension *.m82 with this application** to associate the extension .m82 with this program. This extension is used by the program when saving a project. In this way, you will be able to run the program and load a project by double-clicking on the project filename in Windows Explorer.

Check **Automatically display calculation messages** if you want the report details to be automatically displayed when you calculate the results.

Check **Automatically create backup file** if you want a backup file (with the extension .bck) to be created every time a project is loaded. By default, this file is created in the temporary folder of Windows.

Check **Confirm delete actions** if you want to be asked for confirmation each time an object is about to be deleted. This setting affects the behaviour of all delete actions, for example the deletion of an object.

You can also modify the temporary folder that will be used for the creation of backup files. By default, this folder is the temporary folder of Windows.

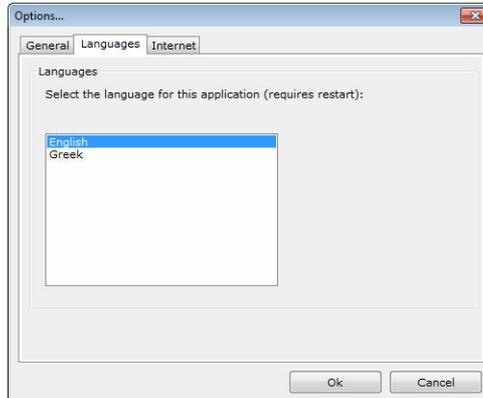
Finally, there are three options regarding the termination of the program:

- **When exiting always ignore changes** - All changes since the last save of the project are ignored.
- **When exiting always save changes** - All changes in the current project are automatically saved. If the filename of the project is not set, a dialog box will

appear that allows the selection of the filename, as when selecting Save project as from the **File** menu.

- **When exiting prompt for action** - If there are changes in the current project, then a dialog box will appear. You can choose to save or ignore the changes. If the filename of the project is not set, a dialog box will appear that allows the selection of the filename, as when selecting Save project as from the **File** menu.

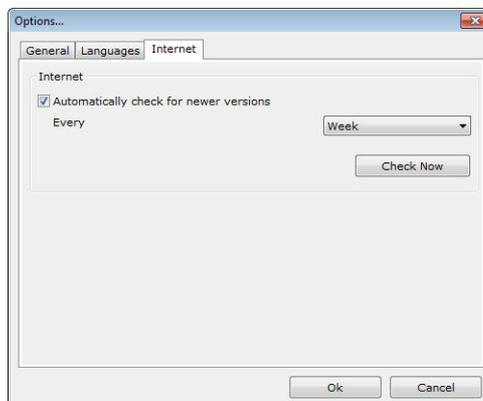
Languages Tab



If more than one language packs have been installed, then you can choose the language of the program. In the above case, there are two language packs; English (that are already selected) and Greek. If you change the language, all forms, menus, messages, help files will reflect the chosen language.

In order for the changes to take effect, you must restart the program.

Internet Tab



The program can automatically check for newer versions over the Internet. Check **Automatically check for newer versions** to enable this feature. The check is automatically performed at an interval specified in the **Every** drop-down list. Select **Check now** to manually check for newer versions.

When a newer version is found, you will be prompted to download and install the latest version.

NOTE: TechnoLogismiki protects your privacy. During the check for newer versions,

no data is transferred from your computer to the Internet.

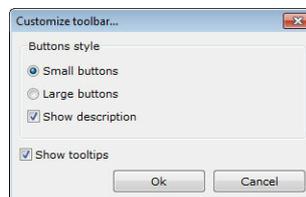
Select **Ok** to apply the changes and close the dialog box. Select **Cancel** to close the dialog box without applying any changes.

4.4.2 Customize toolbar

With this option, you can customize the toolbar of the main form.

To customize the toolbar of the main form:

1. Select **Options** from the **Data** menu.
2. Select **Customize toolbar** from the **Options** menu.
3. Make the appropriate changes.
4. Select **Ok** to apply the changes and close the dialog box. Select **Cancel** to close the dialog box without applying any changes.



The toolbar may contain small or large buttons.

Check **Show description** if you want a small description to be displayed under the buttons.

Check **Show tooltips** if you want tooltips to be displayed when the mouse pointer hovers over a button for 2-3 seconds.

NOTE: These preferences affect all projects, both old and new.

Chapter



5 Data

5.1 Data menu

With this menu, you can add and modify data. In the **Data** menu you can select one of the following options:

- Variables
 - Add
 - Delete
 - Properties
 - Move top
 - Move up
 - Move down
 - Move bottom
- Constraints
 - Add
 - Delete
 - Properties
 - Move top
 - Move up
 - Move down
 - Move bottom
- Scenarios
 - Add
 - Wizard: Standard GA [1 solution]
 - Wizard: Customizable GA
 - Wizard: Sawtooth GA
 - Wizard: Micro GA
 - Wizard: Simple PSO
 - Wizard: Enhanced PSO
 - Wizard: DE
 - Delete
 - Properties
 - Clone
 - Move top
 - Move up
 - Move down
 - Move bottom

5.2 Variables

5.2.1 Add

With this option, you can add variables. These variables control the state of the problem and the program will modify their values in order to obtain better solutions. Prior to using this option, you must first select the corresponding cells in Microsoft Excel.

To add variables:

1. Select the cells that contain the **values** (not the description) of the variables in Microsoft Excel.
2. Select **Variables > Add** from the **Data** menu. All currently selected cells in Microsoft Excel are associated with a variable. Common prefix of their names is DV (Design Variable) and initial settings are automatically assigned to each variable, e.g. low value equal to 0, high value equal to 1 etc. These properties can change easily at a later stage.

NOTE: Alternatively, you can use the left-most toolbar button (denoted with a green plus sign) in the **Variables** tab. In this way, the background color of the selected cells becomes temporarily red in order to avoid confusion due to the lack of focus in Microsoft Excel window.

5.2.2 Delete

With this option, you can delete variables from the list.

To delete variables:

1. Select one or more rows in the list of variables that correspond to the objects you wish to delete.
2. Select **Variables > Delete** from the **Data** menu. If you have enabled the confirmation of deletions (in General preferences), then a confirmation dialog appears.

NOTE: Alternatively, you can use the second-from-the-left toolbar button (denoted with a red x sign) in the **Variables** tab.

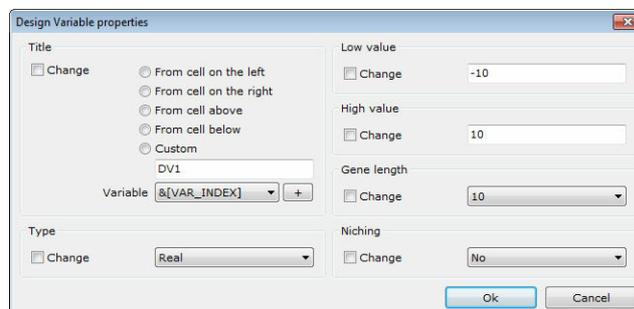
NOTE: During this action, the spreadsheet is not modified in any way.

5.2.3 Properties

With this option, you can modify the properties of one of more variables in the list.

To modify the properties of one of more variables in the list:

1. Select one or more rows in the list of variables that correspond to the objects you wish to modify.
2. Select **Variables > Properties** from the **Data** menu. The following form appears:



3. Make the appropriate modifications according to the following list. Check the

corresponding **Change** check box to actually apply the modification to all selected variables.

- **Title:** it is the title of the variable, which is used to avoid confusion. This can be obtained from the contents of the adjacent cell in Microsoft Excel (the cell on the left, right, above, below), or set to a custom value based on a script. For instance, using 'DV&[VAR_INDEX]' will assign the name 'DV4' to the fourth design variable etc. The list of available reserved names (such as &[VAR_INDEX]) is included below.
- **Type:** select the type of the variable. A variable can be either real or integer.
- **Low value:** select the lower boundary for the variable.
- **High value:** select the higher boundary for the variable.
- **Gene length (for GAs and real variables only):** select the number of bits that will be used for the representation of the variable. The higher the number, the more accurate is the algorithm but also the more difficult becomes the problem. Using a gene length of 10 means that the initial range of the variable, i.e. high value minus low value is divided into $2^{10}-1 = 1023$ segments. Moderate values of gene length are recommended.
- **Niching (for GAs):** select this if you wish the variable to be used in a niching strategy.

4. Press **Ok** to close the form and apply the modifications. Press **Cancel** to close the form without any modifications.

NOTE: Alternatively, you can use the corresponding toolbar button in the **Variables** tab.

NOTE: During this action, the spreadsheet is not modified in any way.

5.2.4 Move top

With this option, you can move the selected variables to the top of the list.

To move the selected variables to the top of the list:

1. Select one or more rows in the list of variables.
2. Select **Variables > Move top** from the **Data** menu.

NOTE: Alternatively, you can use the corresponding toolbar button in the **Variables** tab.

NOTE: During this action, the spreadsheet is not modified in any way.

5.2.5 Move up

With this option, you can move the selected variables upwards by one slot.

To move the selected variables upwards by one slot:

1. Select one or more rows in the list of variables.
2. Select **Variables > Move up** from the **Data** menu.

NOTE: Alternatively, you can use the corresponding toolbar button in the **Variables** tab.

NOTE: During this action, the spreadsheet is not modified in any way.

5.2.6 Move down

With this option, you can move the selected variables downwards by one slot.

To move the selected variables downwards by one slot:

1. Select one or more rows in the list of variables.
2. Select **Variables > Move down** from the **Data** menu.

NOTE: Alternatively, you can use the corresponding toolbar button in the **Variables** tab.

NOTE: During this action, the spreadsheet is not modified in any way.

5.2.7 Move bottom

With this option, you can move the selected variables to the bottom of the list.

To move the selected variables to the bottom of the list:

1. Select one or more rows in the list of variables.
2. Select **Variables > Move bottom** from the **Data** menu.

NOTE: Alternatively, you can use the corresponding toolbar button in the **Variables** tab.

NOTE: During this action, the spreadsheet is not modified in any way.

5.3 Constraints

5.3.1 Add

With this option, you can add constraints. All constraints are implemented as soft constraints using customizable penalty functions. These can be as strict or relaxed as desired. Very strict penalty functions act similarly to hard constraints. Prior to using this option, you must first select the corresponding cells in Microsoft Excel.

To add constraints:

1. Select the cells that contain the **values** (not the description) related to the penalty function in Microsoft Excel.
2. Select **Constraints > Add** from the **Data** menu. All currently selected cells in Microsoft Excel are associated with a constraint. Common prefix of their names is C (Constraint) and initial settings are automatically assigned to each constraint, e.g. the value must less or equal to zero etc. These properties can change easily at a later stage.

NOTE: Alternatively, you can use the left-most toolbar button (denoted with a green plus sign) in the **Constraints** tab. In this way, the background color of the selected cells becomes temporarily red in order to avoid confusion due to the lack of focus in Microsoft Excel window.

5.3.2 Delete

With this option, you can delete constraints from the list.

To delete constraints:

1. Select one or more rows in the list of constraints that correspond to the objects you wish to delete.
2. Select **Constraints > Delete** from the **Data** menu. If you have enabled the confirmation of deletions (in General preferences), then a confirmation dialog appears.

NOTE: Alternatively, you can use the second-from-the-left toolbar button (denoted with a red x sign) in the **Constraints** tab.

NOTE: During this action, the spreadsheet is not modified in any way.

5.3.3 Properties

With this option, you can modify the properties of one of more constraints in the list.

To modify the properties of one of more constraints in the list:

1. Select one or more rows in the list of constraints that correspond to the objects you wish to modify.
2. Select **Constraints > Properties** from the **Data** menu. The following form appears:

3. Make the appropriate modifications according to the following list. Check the corresponding **Change** check box to actually apply the modification to all selected constraints.

- **Title:** it is the title of the constraint, which is used to avoid confusion. This can be obtained from the contents of the adjacent cell in Microsoft Excel (the cell on the left, right, above, below), or set to a custom value based on a script. For instance, using 'C&[CONSTR_INDEX]' will assign the name 'C4' to the fourth constraint etc. The list of available reserved names (such as &[CONSTR_INDEX]) is included below.
- **Comparison:** select the type of the comparison.

- **Value:** select the comparison value.
- **Penalty function:** enter a normal mathematical expression which will be used to evaluate the penalty function. Within the expression, X is violation of the constraint. Note that the code is not even executed if there is no violation. Therefore, for a minimization problem and a constraint of the type 'C1<=0', the expression "1000*X+100" will **not** add 100 to the objective value if there is no violation (even if X=0). If there is a violation, even by little, e.g. X=1E-6, then the code is executed and the objective value is increased by $1000*X+100 = 100.001$. Also note that the result of the penalty function is always added to the objective value, which is natural in case of minimization where the objective function usually represents the cost. The user needs to use appropriate constraints that return negative values in case of a maximization problem.

4. Press **Ok** to close the form and apply the modifications. Press **Cancel** to close the form without any modifications.

NOTE: Alternatively, you can use the corresponding toolbar button in the **Constraints** tab.

NOTE: During this action, the spreadsheet is not modified in any way.

5.3.4 Move top

With this option, you can move the selected constraints to the top of the list.

To move the selected constraints to the top of the list:

1. Select one or more rows in the list of constraints.
2. Select **Constraints > Move top** from the **Data** menu.

NOTE: Alternatively, you can use the corresponding toolbar button in the **Constraints** tab.

NOTE: During this action, the spreadsheet is not modified in any way.

5.3.5 Move up

With this option, you can move the selected constraints upwards by one slot.

To move the selected constraints upwards by one slot:

1. Select one or more rows in the list of constraints.
2. Select **Constraints > Move up** from the **Data** menu.

NOTE: Alternatively, you can use the corresponding toolbar button in the **Constraints** tab.

NOTE: During this action, the spreadsheet is not modified in any way.

5.3.6 Move down

With this option, you can move the selected constraints downwards by one slot.

To move the selected constraints downwards by one slot:

1. Select one or more rows in the list of constraints.

2. Select **Constraints > Move down** from the **Data** menu.

NOTE: Alternatively, you can use the corresponding toolbar button in the **Constraints** tab.

NOTE: During this action, the spreadsheet is not modified in any way.

5.3.7 Move bottom

With this option, you can move the selected constraints to the bottom of the list.

To move the selected constraints to the bottom of the list:

1. Select one or more rows in the list of constraints.
2. Select **Constraints > Move bottom** from the **Data** menu.

NOTE: Alternatively, you can use the corresponding toolbar button in the **Constraints** tab.

NOTE: During this action, the spreadsheet is not modified in any way.

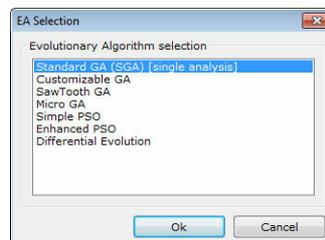
5.4 Scenarios

5.4.1 Add

With this option, you can add scenarios. These scenarios are actually a selection of Evolutionary Algorithm and its accompanying parameters that will be applied to the problem at hand.

To add a scenario:

1. Select **Scenarios > Add** from the **Data** menu. The following form appears:



2. Select the Evolutionary Algorithm that will be used with the new scenario. This cannot change at a later stage, i.e. a new scenario needs to be created. Depending on the choice of EA, the corresponding wizard appears.

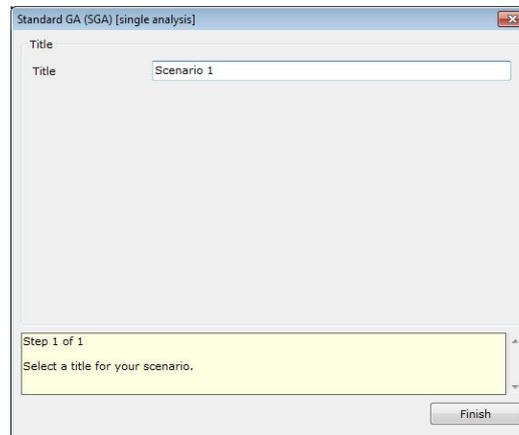
3. Make the appropriate modifications following the steps of the wizard. At the final step, the **Next** button becomes **Finish**. Press **Finish** to complete the definition of the new scenario, or close the form to cancel the process.

NOTE: Alternatively, you can use the left-most toolbar button (denoted with a green plus sign) in the **Scenarios** tab.

5.4.2 Wizard: Standard GA [1 solution]

Step 1 of 1

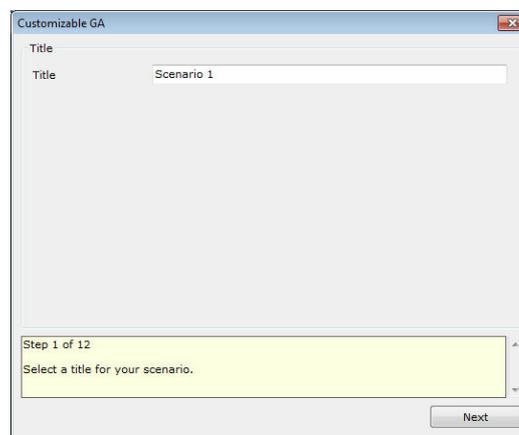
Enter a short title for the scenario. All the other parameters of the algorithm are set automatically.



5.4.3 Wizard: Customizable GA

Step 1 of 12

Enter a short title for the scenario. The log files will be saved to a subfolder with this name.



Step 2 of 12

Enter the population size for the GA. Usually the optimum value depends on the complexity and the nature of the problem. A mathematical expression may also be used.

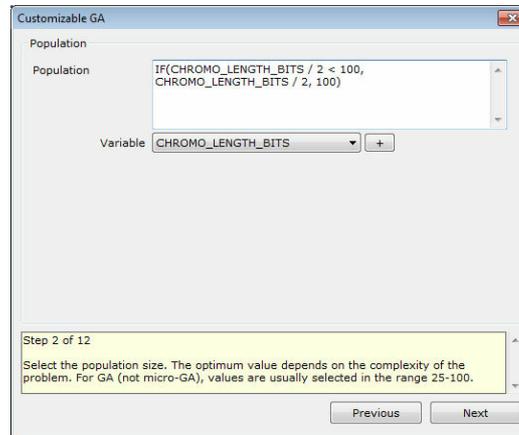
Example 1: entering '50' sets the population size to 50 individuals.

Example 2: entering 'IF(CHROMO_LENGTH_BITS / 2 < 100, CHROMO_LENGTH_BITS / 2, 100)' sets the population size to the chromosome length divided by two if the result is less than 100, or 100 otherwise.

The variables that may be used in the expression are:

CHROMO_LENGTH_BITS: the chromosome length in bits.

CHROMO_LENGTH_GENES: the chromosome length in genes. This equals the number of design variables.



Step 3 of 12

Choose the selection method. The method can be either tournament selection or biased roulette wheel.

Usually, tournament selection with pool size equal to 2 and probability equal to 1 is adequate. The roulette wheel method requires the definition of a fitness function, based on the objective function.

The evaluation of the fitness can be performed automatically, or based on custom code. In the latter case, the variables that can be used are the following:

MIN_OV: the minimum objective value within the members of the population.

MAX_OV: the maximum objective value within the members of the population.

AVER_OV: the average objective value within the members of the population.

BEST_OV: the best objective value within the members of the population.

WORST_OV: the worst objective value within the members of the population.

OV: the objective value of the current individual for which the fitness is being evaluated.

TARGET: the target objective value for target - problems.

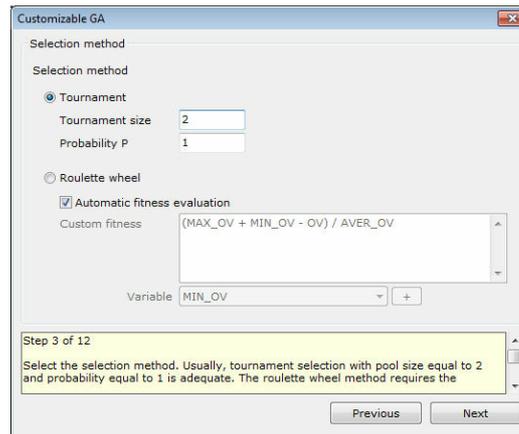
The automatic evaluation is based on the following code:

'(OV - MIN_OV) / AVER_OV' for maximization problems,

'(MAX_OV + MIN_OV - OV) / AVER_OV' for minimization problems,

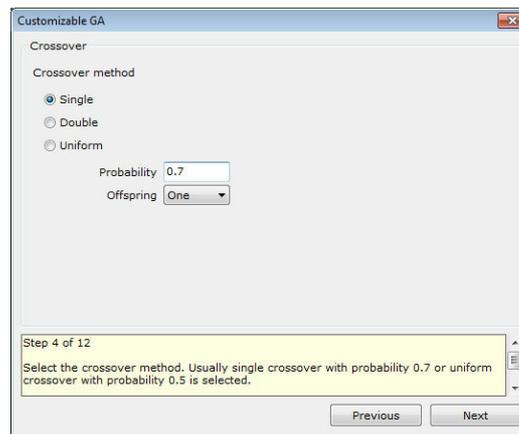
'(ABS(MAX_OV - TARGET) + ABS(MIN_OV - TARGET) - ABS(OV - TARGET)) / ABS(AVER_OV - TARGET)' for target-value problems.

Based on the above, a first level of normalization is achieved with the division by the average objective value.



Step 4 of 12

Select the crossover method. Usually single crossover with probability 0.7 or uniform crossover with probability 0.5 is selected. Also, choose whether the selected pair of chromosomes will produce one or two offspring.

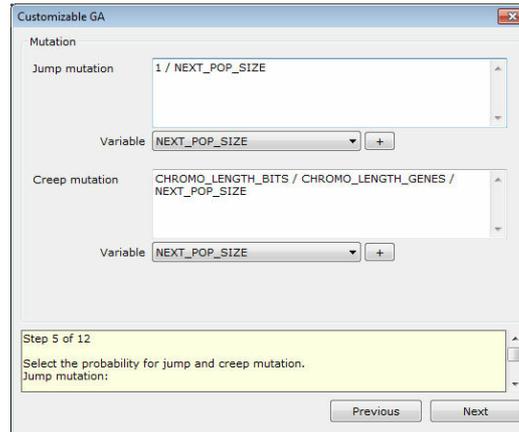


Step 5 of 12

Select the probability for jump and creep mutation.

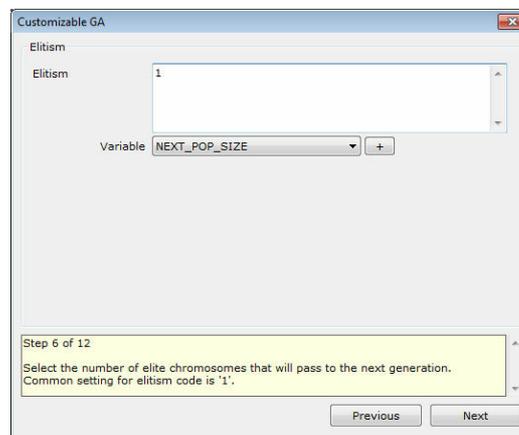
Common setting for jump mutation code is '1 / NEXT_POP_SIZE'. In this case the random number generator is invoked $\text{NEXT_POP_SIZE} * \text{CHROMO_LENGTH_BITS}$ times.

Common setting for creep mutation code is ' $\text{CHROMO_LENGTH_BITS} / \text{CHROMO_LENGTH_GENES} / \text{NEXT_POP_SIZE}$ '. In this case the random number generator is invoked $\text{NEXT_POP_SIZE} * \text{CHROMO_LENGTH_GENES}$ times.



Step 6 of 12

Select the number of elite chromosomes that will pass to the next generation. Common setting for elitism code is '1', the single best individual is transferred to the next generation.



Step 7 of 12

Select other algorithm settings. The options are the following:

Gray code: Choose whether the chromosome is encoded using Gray code or not.

Niching: Select the alpha coefficient (power law exponent for sharing function; typically = 1.0) and the normalized distance ($1/\text{normalized distance}$ can be viewed as the number of regions over which the sharing function should focus). This affects the design variables that have been set to participate in a niching strategy.

Hill climber: Choose whether after the termination of the GA a local optimizer, namely steepest ascend hill climber, is invoked. Also select the maximum number of function evaluations for the local optimizer.

Customizable GA

Other settings

Use Gray code

Use niching

Alpha: 1

Normalized distance: .1

Fitness normalization

C multiplier: 2

After termination of the EA, use hill climber

Maximum hill climbing evaluations: 1000

Step 7 of 12

Select other algorithm settingshe options are the following:
- Gray code. Choose whether the chromosome is encoded using Gray code or not.

Previous Next

Step 8 of 12

Choose whether to use the Registrar operator with the algorithm. This is recommended only for expensive objective functions.

Customizable GA

Registrar

Use Registrar operator

Similarity criteria in phenotype

Initial distance: .25

Maximum rate: .15

Multiplier: .95

Impose similarity in all dimensions

Similarity criteria in genotype

Initial max different bits fraction: .25

Maximum rate: .99

Multiplier: .99

Use accelerator with: 5

Step 8 of 12

Choose whether to use the Registrar operator with the algorithm.
This is recommended only for expensive objective functions.

Previous Next

Step 9 of 12

Select the termination criteria. At least one termination criterion must be active. Usually one selects a maximum number of function evaluations and/or a time limit.

Customizable GA

Termination criteria

Compare best result

Best: <= 0

Max generations

Value: 200

Max function evaluations

Value: 20000

Best remains for max generations

Value: 25

Min different bits fraction

Value: .01

Max time elapsed

Minutes: 5

Step 9 of 12

Select the termination criteria. At least one termination criterion must be active.
Usually one selects a time limit and/or a maximum number of function evaluations.

Previous Next

Step 10 of 12

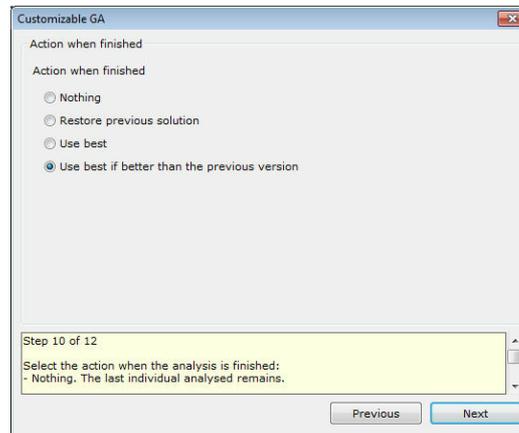
Select the action when the analysis is finished:

Nothing: The last individual analysed remains.

Restore previous solution: The solution that was visible before the analysis is restored.

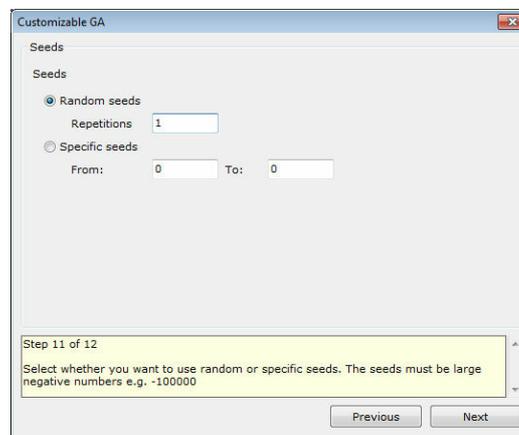
Use best: The best solution that was discovered during the analysis is applied.

Use best if better than the previous version (recommended): The best solution that was discovered during the analysis is applied only if it is better than the previous solution. In the opposite case the previous solution is restored.



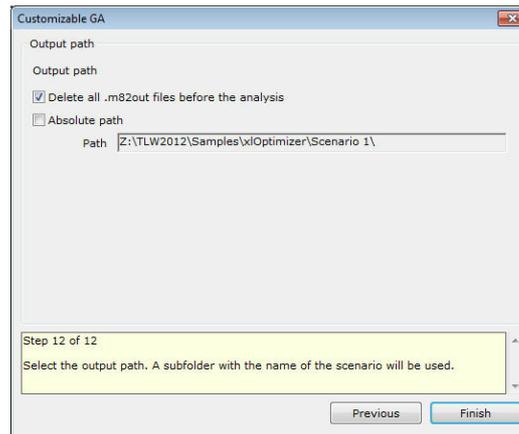
Step 11 of 12

Select whether you want to use random or specific seeds for the random number generator. The seeds must be large negative numbers e.g. -100000. Usually a number of analyses with random seeds is performed. The filename of the log is the seed used in the specific analysis.



Step 12 of 12

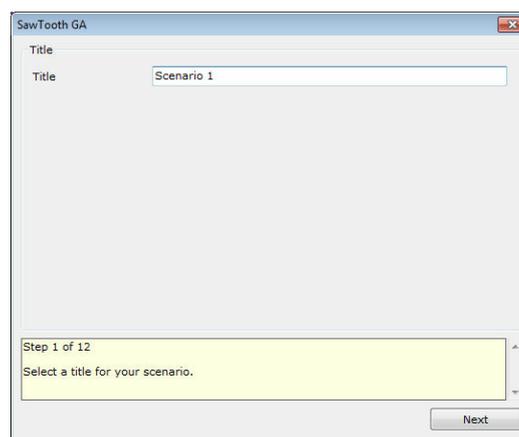
Select the output path. A subfolder with the name of the scenario will be used. Usually a relative path is used i.e. the **absolute path** check box is not checked. This results in the output logs being saved in the same path as the project file.



5.4.4 Wizard: Sawtooth GA

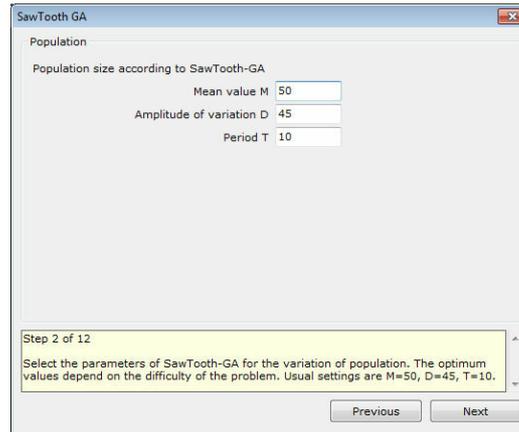
Step 1 of 12

Enter a short title for the scenario. The log files will be saved to a subfolder with this name.



Step 2 of 12

Select the parameters of SawTooth-GA for the variation of population. The optimum values depend on the difficulty of the problem. Usual settings are $M=50$, $D=45$, $T=10$. This means that the mean value is 50 individuals, the amplitude of variation is 45, i.e. the population varies from $50+45=95$ individuals to $50-45=5$ individuals, and the period of variation is 10 generations.



Step 3 of 12

Choose the selection method. The method can be either tournament selection or biased roulette wheel.

Usually, tournament selection with pool size equal to 2 and probability equal to 1 is adequate. The roulette wheel method requires the definition of a fitness function, based on the objective function.

The evaluation of the fitness can be performed automatically, or based on custom code. In the latter case, the variables that can be used are the following:

MIN_OV: the minimum objective value within the members of the population.

MAX_OV: the maximum objective value within the members of the population.

AVER_OV: the average objective value within the members of the population.

BEST_OV: the best objective value within the members of the population.

WORST_OV: the worst objective value within the members of the population.

OV: the objective value of the current individual for which the fitness is being evaluated.

TARGET: the target objective value for target - problems.

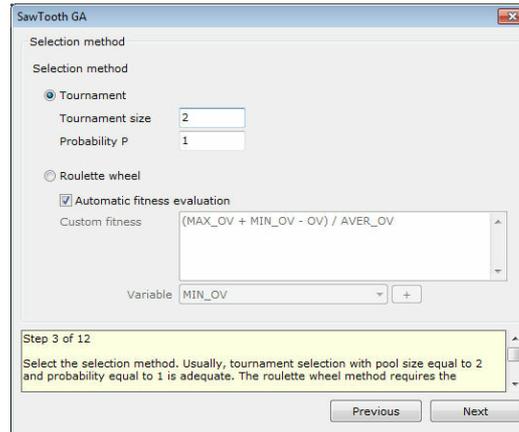
The automatic evaluation is based on the following code:

' $(OV - MIN_OV) / AVER_OV$ ' for maximization problems,

' $(MAX_OV + MIN_OV - OV) / AVER_OV$ ' for minimization problems,

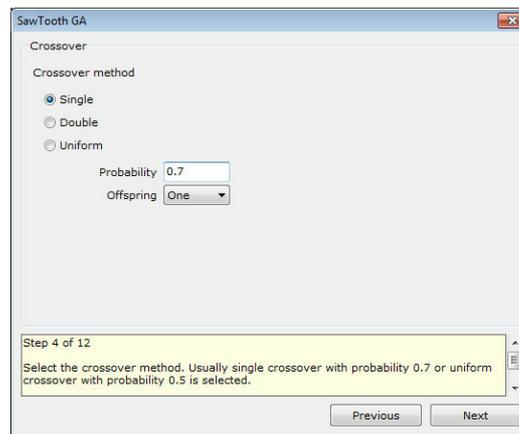
' $(ABS(MAX_OV - TARGET) + ABS(MIN_OV - TARGET) - ABS(OV - TARGET)) / ABS(AVER_OV - TARGET)$ ' for target-value problems.

Based on the above, a first level of normalization is achieved with the division by the average objective value.



Step 4 of 12

Select the crossover method. Usually single crossover with probability 0.7 or uniform crossover with probability 0.5 is selected. Also, choose whether the selected pair of chromosomes will produce one or two offspring.

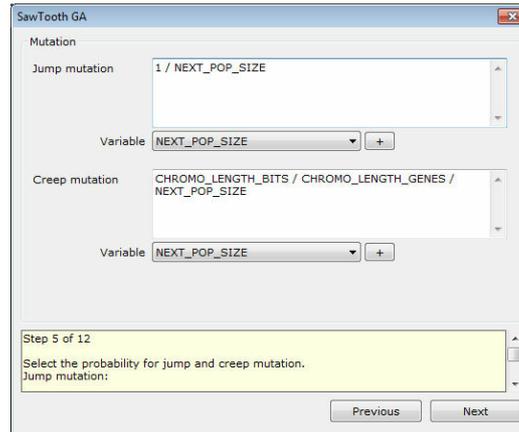


Step 5 of 12

Select the probability for jump and creep mutation.

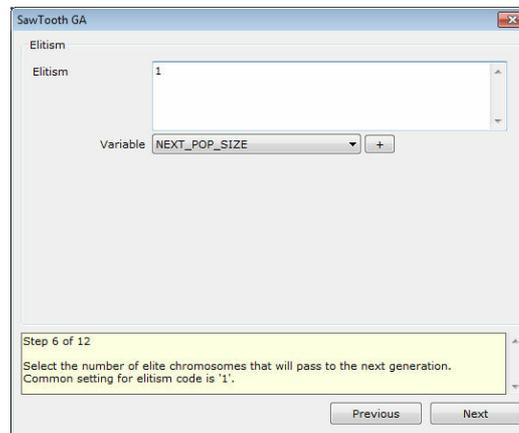
Common setting for jump mutation code is '1 / NEXT_POP_SIZE'. In this case the random number generator is invoked $\text{NEXT_POP_SIZE} * \text{CHROMO_LENGTH_BITS}$ times.

Common setting for creep mutation code is ' $\text{CHROMO_LENGTH_BITS} / \text{CHROMO_LENGTH_GENES} / \text{NEXT_POP_SIZE}$ '. In this case the random number generator is invoked $\text{NEXT_POP_SIZE} * \text{CHROMO_LENGTH_GENES}$ times.



Step 6 of 12

Select the number of elite chromosomes that will pass to the next generation. Common setting for elitism code is '1', the single best individual is transferred to the next generation.



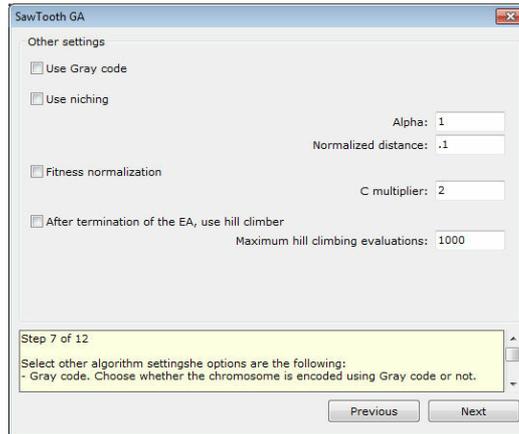
Step 7 of 12

Select other algorithm settings. The options are the following:

Gray code: Choose whether the chromosome is encoded using Gray code or not.

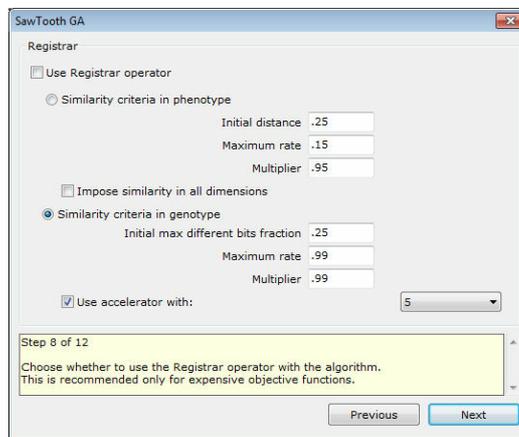
Niching: Select the alpha coefficient (power law exponent for sharing function; typically = 1.0) and the normalized distance ($1/\text{normalized distance}$ can be viewed as the number of regions over which the sharing function should focus). This affects the design variables that have been set to participate in a niching strategy.

Hill climber: Choose whether after the termination of the GA a local optimizer, namely steepest ascend hill climber, is invoked. Also select the maximum number of function evaluations for the local optimizer.



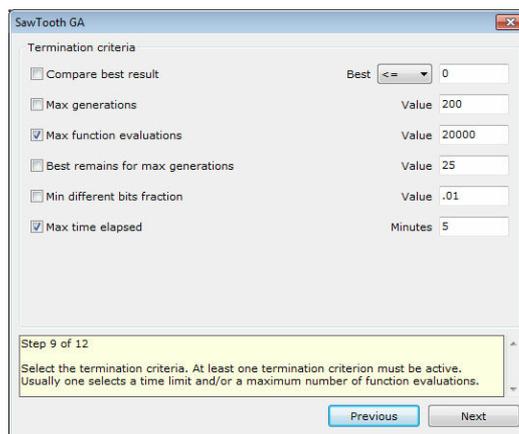
Step 8 of 12

Choose whether to use the Registrar operator with the algorithm. This is recommended only for expensive objective functions.



Step 9 of 12

Select the termination criteria. At least one termination criterion must be active. Usually one selects a maximum number of function evaluations and/or a time limit.



Step 10 of 12

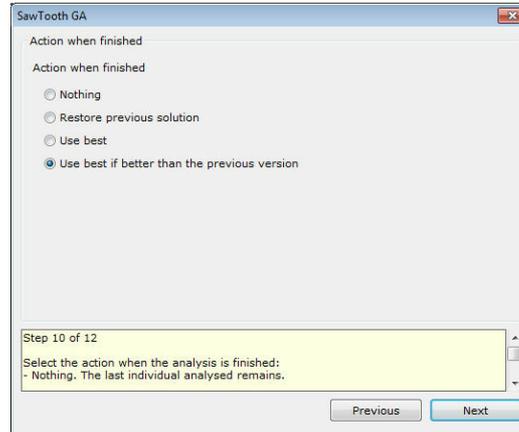
Select the action when the analysis is finished:

Nothing: The last individual analysed remains.

Restore previous solution: The solution that was visible before the analysis is restored.

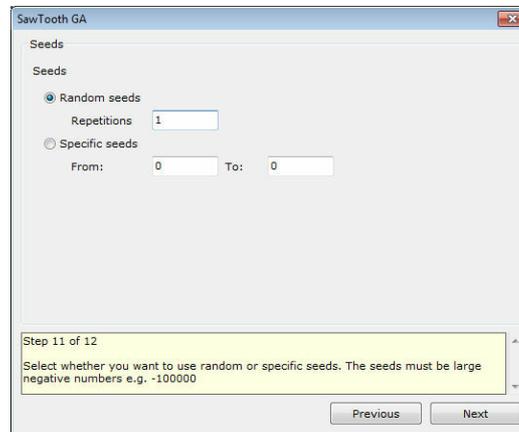
Use best: The best solution that was discovered during the analysis is applied.

Use best if better than the previous version (recommended): The best solution that was discovered during the analysis is applied only if it is better than the previous solution. In the opposite case the previous solution is restored.



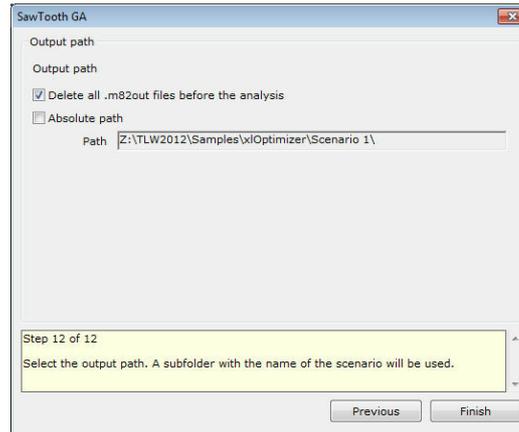
Step 11 of 12

Select whether you want to use random or specific seeds for the random number generator. The seeds must be large negative numbers e.g. -100000. Usually a number of analyses with random seeds is performed. The filename of the log is the seed used in the specific analysis.



Step 12 of 12

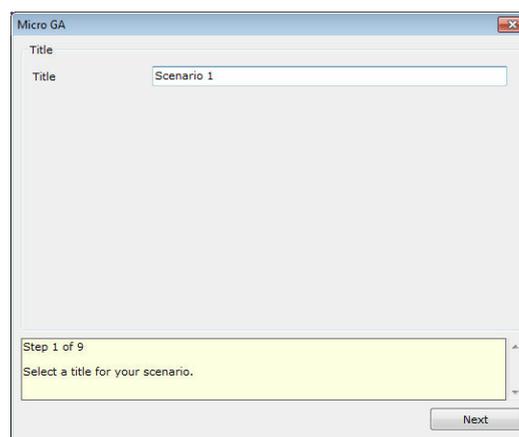
Select the output path. A subfolder with the name of the scenario will be used. Usually a relative path is used i.e. the **absolute path** check box is not checked. This results in the output logs being saved in the same path as the project file.



5.4.5 Wizard: Micro GA

Step 1 of 9

Enter a short title for the scenario. The log files will be saved to a subfolder with this name.



Step 2 of 9

Enter the population size for the GA. For Micro GA, usually 5-10 individuals is adequate.

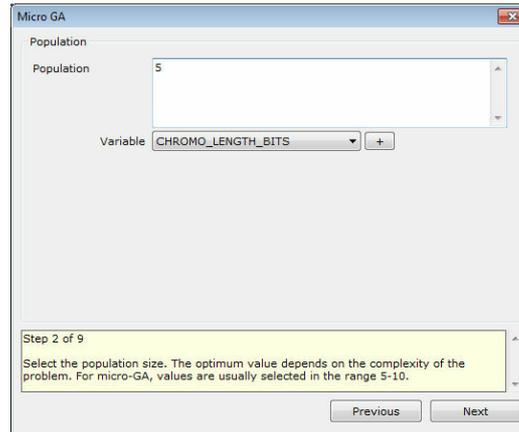
Example 1: entering '50' sets the population size to 50 individuals.

Example 2: entering 'IF(CHROMO_LENGTH_BITS / 2 < 100, CHROMO_LENGTH_BITS / 2, 100)' sets the population size to the chromosome length divided by two if the result is less than 100, or 100 otherwise.

The variables that may be used in the expression are:

CHROMO_LENGTH_BITS: the chromosome length in bits.

CHROMO_LENGTH_GENES: the chromosome length in genes. This equals the number of design variables.



Step 3 of 9

Choose the selection method. The method can be either tournament selection or biased roulette wheel.

Usually, tournament selection with pool size equal to 2 and probability equal to 1 is adequate. The roulette wheel method requires the definition of a fitness function, based on the objective function.

The evaluation of the fitness can be performed automatically, or based on custom code. In the latter case, the variables that can be used are the following:

MIN_OV: the minimum objective value within the members of the population.

MAX_OV: the maximum objective value within the members of the population.

AVER_OV: the average objective value within the members of the population.

BEST_OV: the best objective value within the members of the population.

WORST_OV: the worst objective value within the members of the population.

OV: the objective value of the current individual for which the fitness is being evaluated.

TARGET: the target objective value for target - problems.

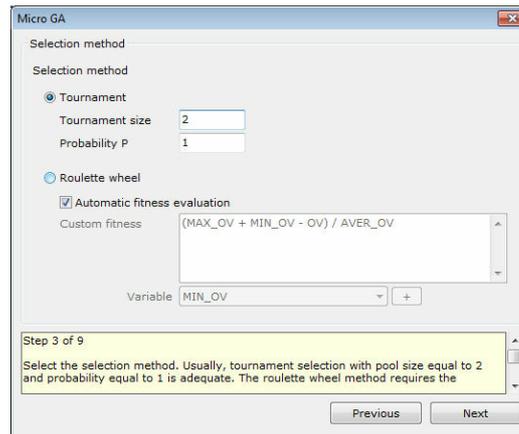
The automatic evaluation is based on the following code:

' $(OV - MIN_OV) / AVER_OV$ ' for maximization problems,

' $(MAX_OV + MIN_OV - OV) / AVER_OV$ ' for minimization problems,

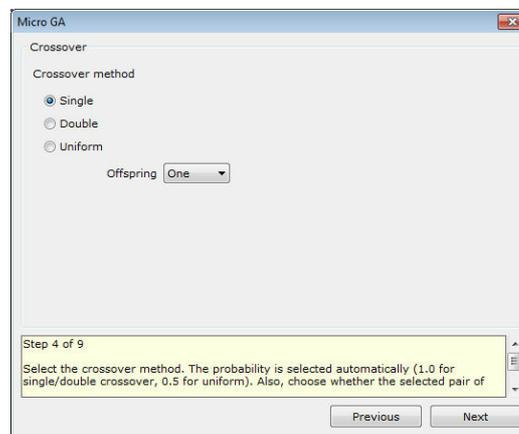
' $(ABS(MAX_OV - TARGET) + ABS(MIN_OV - TARGET) - ABS(OV - TARGET)) / ABS(AVER_OV - TARGET)$ ' for target-value problems.

Based on the above, a first level of normalization is achieved with the division by the average objective value.



Step 4 of 9

Select the crossover method. Usually single crossover is selected. Also, choose whether the selected pair of chromosomes will produce one or two offspring.



Step 5 of 9

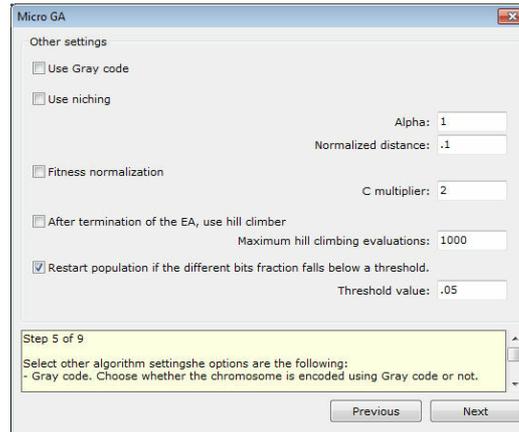
Select other algorithm settings. The options are the following:

Gray code: Choose whether the chromosome is encoded using Gray code or not.

Niching: Select the alpha coefficient (power law exponent for sharing function; typically = 1.0) and the normalized distance (1/normalized distance can be viewed as the number of regions over which the sharing function should focus). This affects the design variables that have been set to participate in a niching strategy.

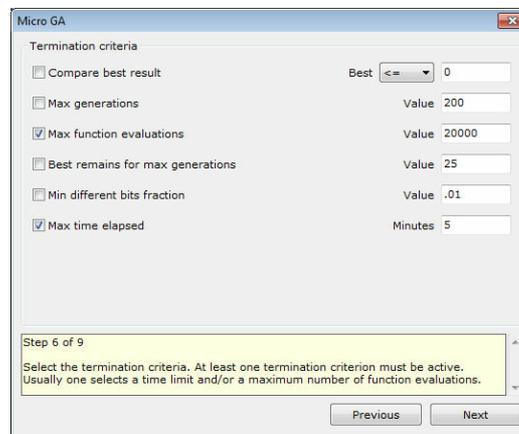
Hill climber: Choose whether after the termination of the GA a local optimizer, namely steepest ascend hill climber, is invoked. Also select the maximum number of function evaluations for the local optimizer.

Restart population after different bits fraction falls below a threshold: Choose the threshold of different bits fraction (with respect to the current best individual) under which the population must be restarted. This is imperative in micro-GA. Usual value is 0.05, i.e. 5%.



Step 6 of 9

Select the termination criteria. At least one termination criterion must be active. Usually one selects a maximum number of function evaluations and/or a time limit.



Step 7 of 9

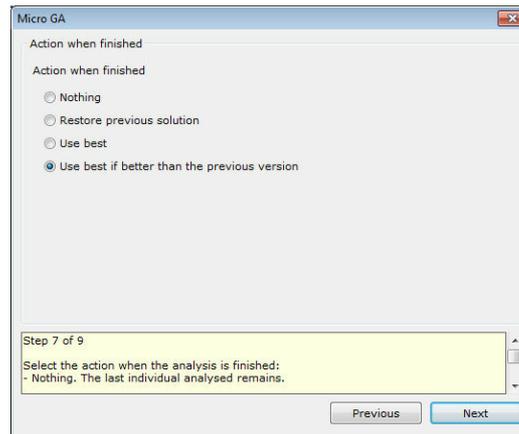
Select the action when the analysis is finished:

Nothing: The last individual analysed remains.

Restore previous solution: The solution that was visible before the analysis is restored.

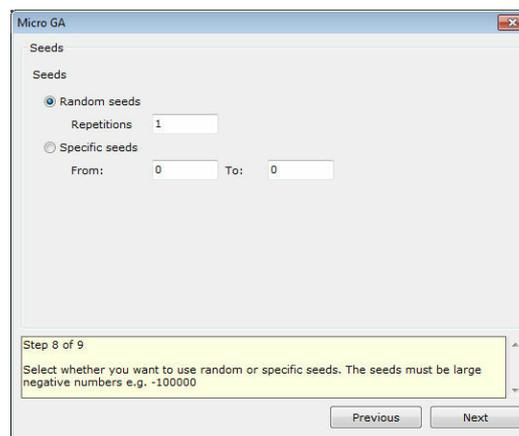
Use best: The best solution that was discovered during the analysis is applied.

Use best if better than the previous version (recommended): The best solution that was discovered during the analysis is applied only if it is better than the previous solution. In the opposite case the previous solution is restored.



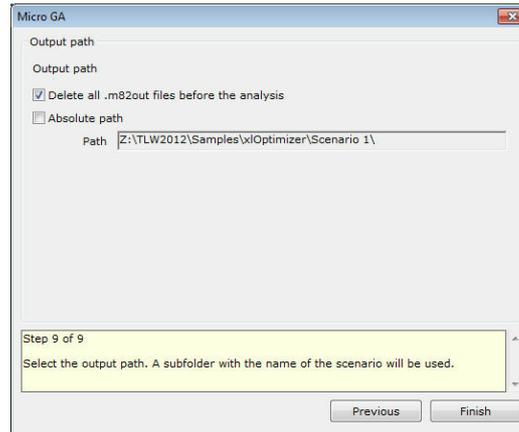
Step 8 of 9

Select whether you want to use random or specific seeds for the random number generator. The seeds must be large negative numbers e.g. -100000. Usually a number of analyses with random seeds is performed. The filename of the log is the seed used in the specific analysis.



Step 9 of 9

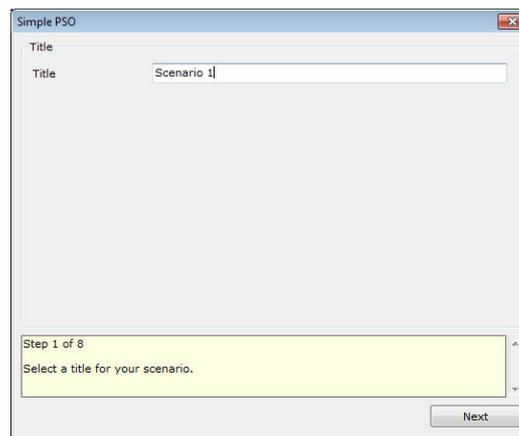
Select the output path. A subfolder with the name of the scenario will be used. Usually a relative path is used i.e. the **absolute path** check box is not checked. This results in the output logs being saved in the same path as the project file.



5.4.6 Wizard: Simple PSO

Step 1 of 8

Enter a short title for the scenario. The log files will be saved to a subfolder with this name.



Step 2 of 8

Select the number of particles. The optimum value depends on the complexity of the problem. For PSO, values are usually selected in the range 10-30.

The variables that may be used in the expression are:

VARIABLES: the number of variables.

Simple PSO

Population

Population 20

Variable VARIABLES

Step 2 of 8

Select the number of particles. The optimum value depends on the complexity of the problem. For PSO, values are usually selected in the range 10-30.

Previous Next

Step 3 of 8

Select the parameters of the PSO algorithm:

Cognitive parameter C1: controls the weight of the pull of each particle to the best ever position of the same particle. Usual value for simple PSO is 2.

Social parameter C2: controls the weight of the pull of each particle to the best ever position of the whole swarm. Usual value for simple PSO is 2.

Inertia parameter W0: controls the inertia of the particle. Usual values for simple PSO are in the range 0.8-1.4.

Parameter gamma: controls the initial maximum velocities of the particles, as the fraction of the design space (per variable) that can be travelled in one time step. Usual value is 0.4.

Simple PSO

Parameters

Cognitive parameter C1 2

Social parameter C2 2

Inertia weight W0 .8

Parameter gamma .4

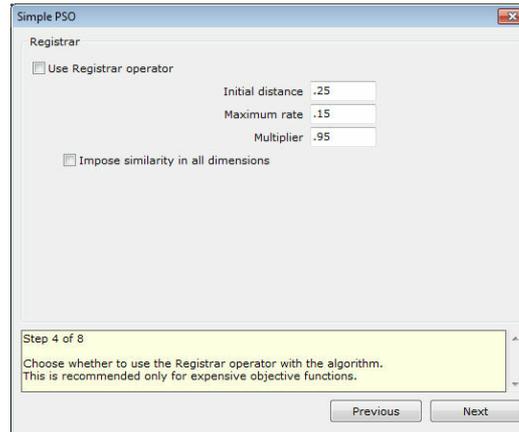
Step 3 of 8

Select the parameters of the PSO algorithm:
- Cognitive parameter C1: controls the weight of the pull of each particle to the best

Previous Next

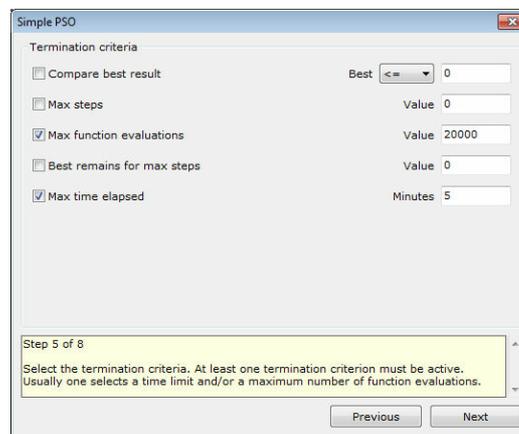
Step 4 of 8

Choose whether to use the Registrar operator with the algorithm. This is recommended only for expensive objective functions.



Step 5 of 8

Select the termination criteria. At least one termination criterion must be active. Usually one selects a maximum number of function evaluations and/or a time limit.



Step 6 of 8

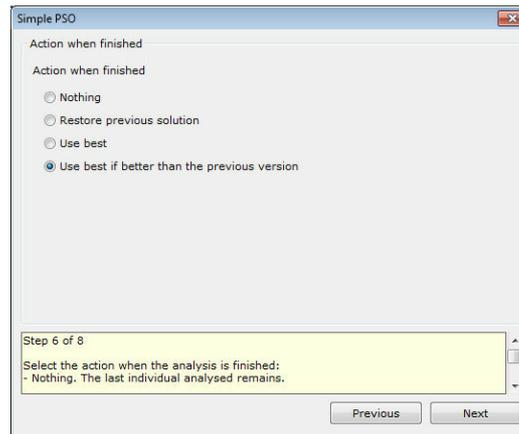
Select the action when the analysis is finished:

Nothing: The last individual analysed remains.

Restore previous solution: The solution that was visible before the analysis is restored.

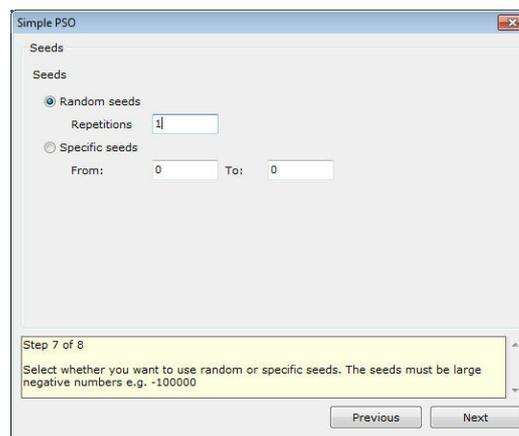
Use best: The best solution that was discovered during the analysis is applied.

Use best if better than the previous version (recommended): The best solution that was discovered during the analysis is applied only if it is better than the previous solution. In the opposite case the previous solution is restored.



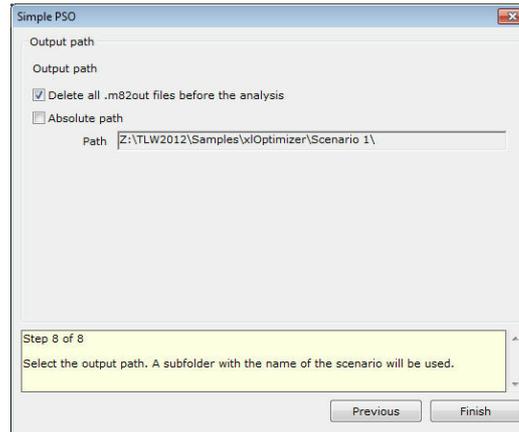
Step 7 of 8

Select whether you want to use random or specific seeds for the random number generator. The seeds must be large negative numbers e.g. -100000. Usually a number of analyses with random seeds is performed. The filename of the log is the seed used in the specific analysis.



Step 8 of 8

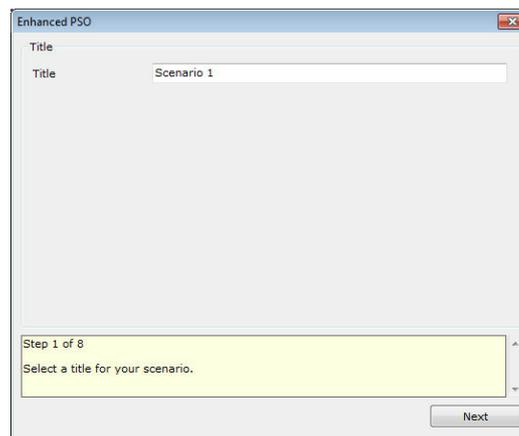
Select the output path. A subfolder with the name of the scenario will be used. Usually a relative path is used i.e. the **absolute path** check box is not checked. This results in the output logs being saved in the same path as the project file.



5.4.7 Wizard: Enhanced PSO

Step 1 of 8

Enter a short title for the scenario. The log files will be saved to a subfolder with this name.

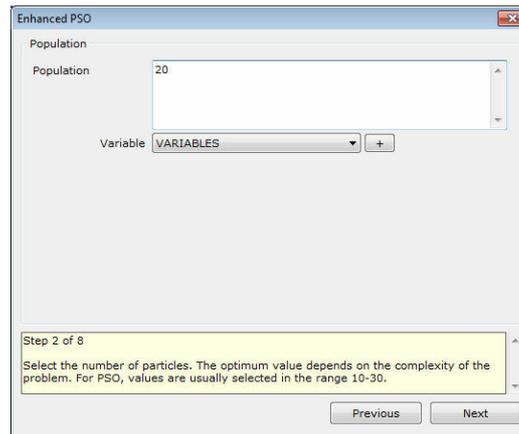


Step 2 of 8

Select the number of particles. The optimum value depends on the complexity of the problem. For PSO, values are usually selected in the range 10-30.

The variables that may be used in the expression are:

VARIABLES: the number of variables.



Step 3 of 8

Select the parameters of the PSO algorithm:

Cognitive parameter C1: controls the weight of the pull of each particle to the best ever position of the same particle. Usual value for simple PSO is 2.

Social parameter C2: controls the weight of the pull of each particle to the best ever position of the whole swarm. Usual value for simple PSO is 2.

Inertia parameter W0: controls the inertia of the particle. Usual values for simple PSO are in the range 0.8-1.4.

Parameter gamma: controls the initial maximum velocities of the particles, as the fraction of the design space (per variable) that can be travelled in one time step. Usual value is 0.4.

Parameter h (consecutive steps without improvement of the global optimum): If more than h steps are observed without improvement, then it is assumed that the velocities are large and $w = \alpha * w$, $v_{max} = v_{max} * \beta$. Use zero to disable. Usual value is 3.

Parameter alpha : If the h-parameter is activated, the inertia parameter is decreased according to $w = \alpha * w$. Use 1 to disable. Usual value is 0.99.

Parameter beta: If the h-parameter is activated, the maximum velocities are decreased according to $v_{max} = \beta * v_{max}$. Use 1 to disable. Usual value is 0.95.

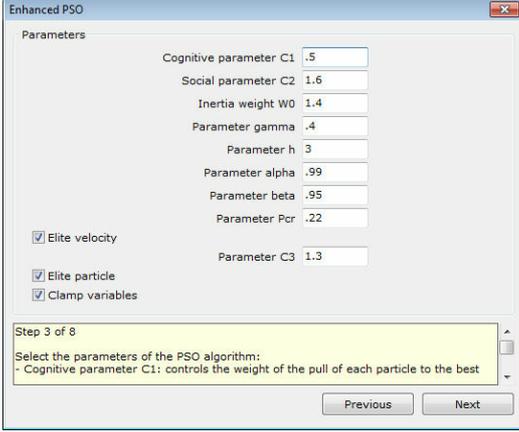
Parameter Pcr: 'Craziness' probability. When activated, the velocity of a particle is initialized randomly. It is a safeguard against premature convergence. Use 0 to disable. Usual value is 0.22.

Elite velocity: if the velocity of a particle improved the best ever position of the swarm, then it is used again, multiplied by a random number in [0.1] and also multiplied by parameter C3.

Elite velocity coefficient C3: used when a particle improved the best ever position of the swarm. Usual value is 1.3.

Elite particle: the worst particle in each step is transferred to the best ever position of the swarm.

Clamp variables: the variables are clamped within their boundaries prior to evaluation.



Enhanced PSO

Parameters

Cognitive parameter C1 .5

Social parameter C2 1.6

Inertia weight W0 1.4

Parameter gamma .4

Parameter h 3

Parameter alpha .99

Parameter beta .95

Parameter Pcr .22

Parameter C3 1.3

Elite velocity

Elite particle

Clamp variables

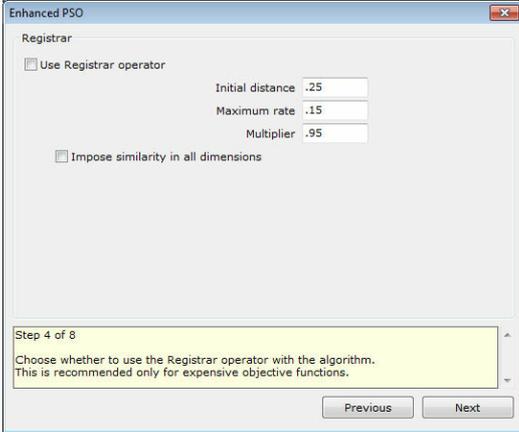
Step 3 of 8

Select the parameters of the PSO algorithm:
- Cognitive parameter C1: controls the weight of the pull of each particle to the best

Previous Next

Step 4 of 8

Choose whether to use the Registrar operator with the algorithm. This is recommended only for expensive objective functions.



Enhanced PSO

Registrar

Use Registrar operator

Initial distance .25

Maximum rate .15

Multiplier .95

Impose similarity in all dimensions

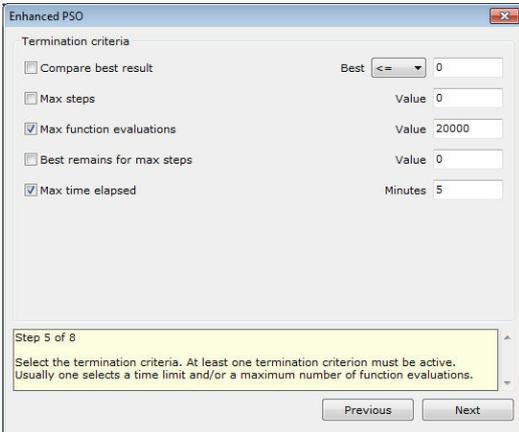
Step 4 of 8

Choose whether to use the Registrar operator with the algorithm.
This is recommended only for expensive objective functions.

Previous Next

Step 5 of 8

Select the termination criteria. At least one termination criterion must be active. Usually one selects a maximum number of function evaluations and/or a time limit.



Enhanced PSO

Termination criteria

Compare best result Best <= 0

Max steps Value 0

Max function evaluations Value 20000

Best remains for max steps Value 0

Max time elapsed Minutes 5

Step 5 of 8

Select the termination criteria. At least one termination criterion must be active.
Usually one selects a time limit and/or a maximum number of function evaluations.

Previous Next

Step 6 of 8

Select the action when the analysis is finished:

Nothing: The last individual analysed remains.

Restore previous solution: The solution that was visible before the analysis is restored.

Use best: The best solution that was discovered during the analysis is applied.

Use best if better than the previous version (recommended): The best solution that was discovered during the analysis is applied only if it is better than the previous solution. In the opposite case the previous solution is restored.

Enhanced PSO

Action when finished

Action when finished

Nothing

Restore previous solution

Use best

Use best if better than the previous version

Step 6 of 8

Select the action when the analysis is finished:
- Nothing. The last individual analysed remains.

Previous Next

Step 7 of 8

Select whether you want to use random or specific seeds for the random number generator. The seeds must be large negative numbers e.g. -100000. Usually a number of analyses with random seeds is performed. The filename of the log is the seed used in the specific analysis.

Enhanced PSO

Seeds

Seeds

Random seeds

Repetitions: 1

Specific seeds

From: 0 To: 0

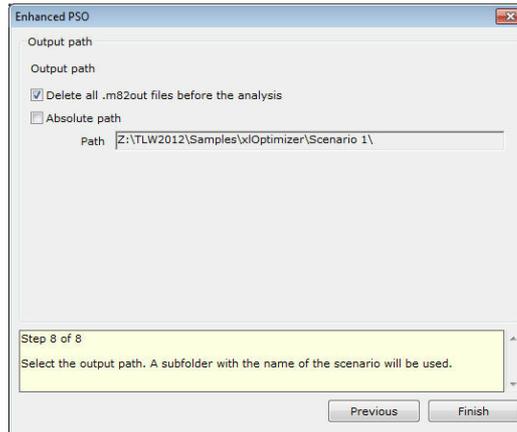
Step 7 of 8

Select whether you want to use random or specific seeds. The seeds must be large negative numbers e.g. -100000

Previous Next

Step 8 of 8

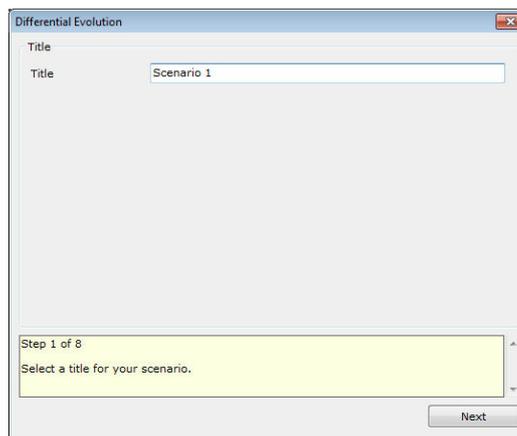
Select the output path. A subfolder with the name of the scenario will be used. Usually a relative path is used i.e. the **absolute path** check box is not checked. This results in the output logs being saved in the same path as the project file.



5.4.8 Wizard: DE

Step 1 of 8

Enter a short title for the scenario. The log files will be saved to a subfolder with this name.

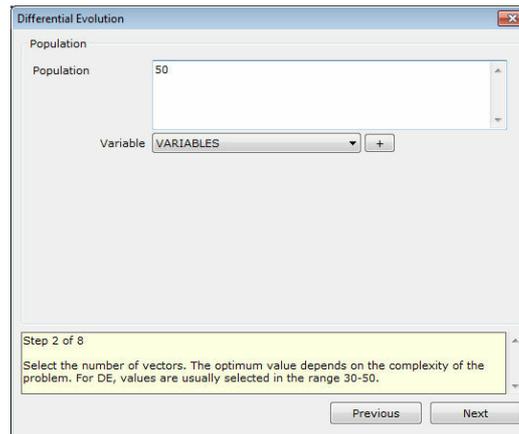


Step 2 of 8

Select the number of vectors. The optimum value depends on the complexity of the problem. For DE, values are usually selected in the range 30-50.

The variables that may be used in the expression are:

VARIABLES: the number of variables.



Step 3 of 8

Select the parameters of the DE algorithm:

Method: select rand/1/bin ('classic' DE, more robust) in which the base vector is selected randomly, or best/1/bin (faster, but less robust) in which the base vector is the current best vector of the population. Note that a random base can be selected stochastically in the latter case (according to random base probability parameter).

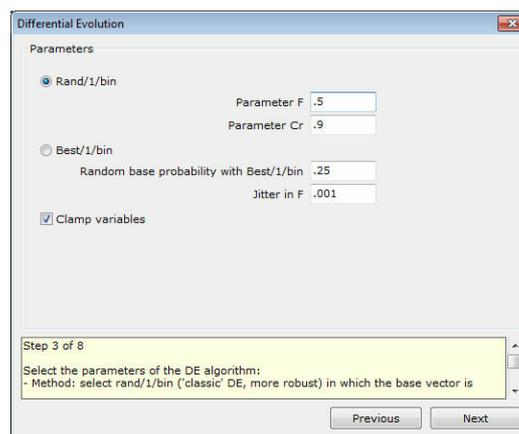
Parameter F: used to scale the difference between two vectors to form a mutated vector according to $v = x_0 + F \times (x_1 - x_2)$ (for random base). Usual value is 0.5.

Parameter Cr: used to form the trial vector based on the mutated vector. Usual value is 0.9.

Random base probability: when using best/1/bin, this is the probability that actually a random base is used instead of the best vector. This increases the robustness of best/1/bin. Usual value is 0.25.

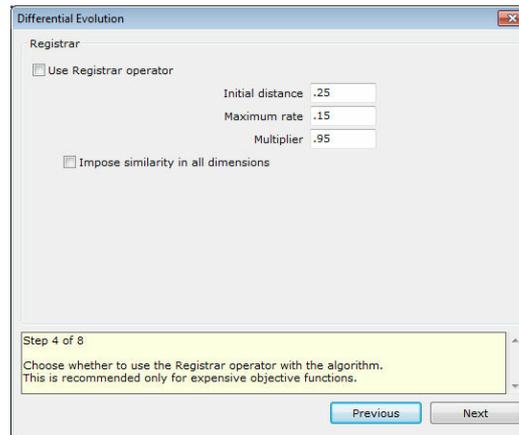
Jitter F: used to introduce a jitter in parameter F with best/1/bin. Usual value is 0.001.

Clamp variables: if checked, the final vectors are clamped within the boundaries of each variable.



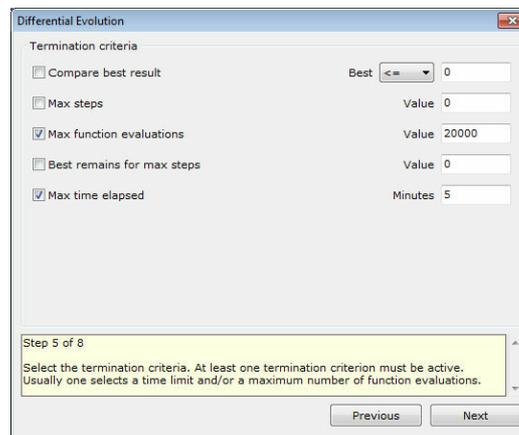
Step 4 of 8

Choose whether to use the Registrar operator with the algorithm. This is recommended only for expensive objective functions.



Step 5 of 8

Select the termination criteria. At least one termination criterion must be active. Usually one selects a maximum number of function evaluations and/or a time limit.



Step 6 of 8

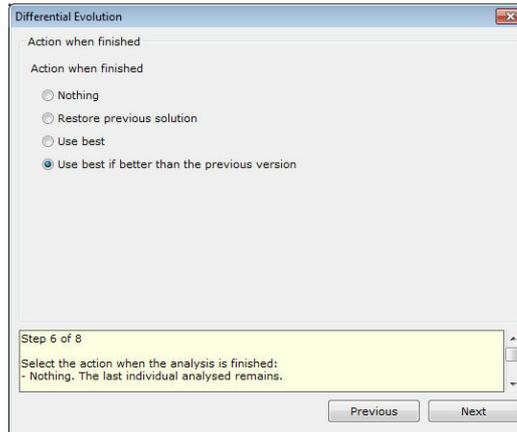
Select the action when the analysis is finished:

Nothing: The last individual analysed remains.

Restore previous solution: The solution that was visible before the analysis is restored.

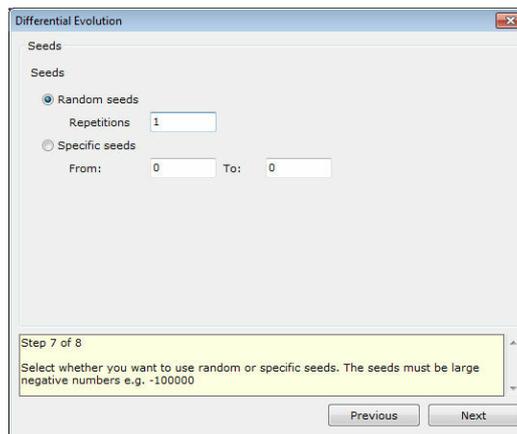
Use best: The best solution that was discovered during the analysis is applied.

Use best if better than the previous version (recommended): The best solution that was discovered during the analysis is applied only if it is better than the previous solution. In the opposite case the previous solution is restored.



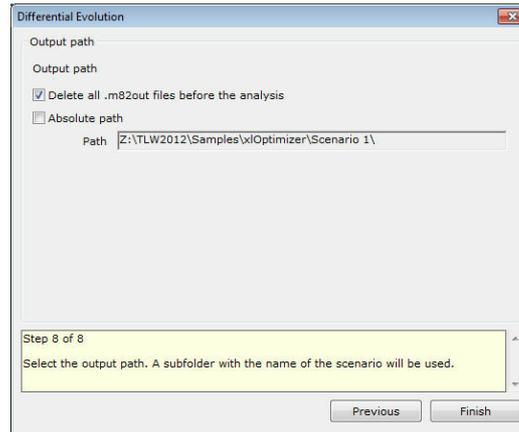
Step 7 of 8

Select whether you want to use random or specific seeds for the random number generator. The seeds must be large negative numbers e.g. -100000. Usually a number of analyses with random seeds is performed. The filename of the log is the seed used in the specific analysis.



Step 8 of 8

Select the output path. A subfolder with the name of the scenario will be used. Usually a relative path is used i.e. the **absolute path** check box is not checked. This results in the output logs being saved in the same path as the project file.



5.4.9 Delete

With this option, you can delete scenarios from the list.

To delete scenarios:

1. Select one or more rows in the list of scenarios that correspond to the objects you wish to delete.
2. Select **Scenarios > Delete** from the **Data** menu. If you have enabled the confirmation of deletions (in General preferences), then a confirmation dialog appears.

NOTE: Alternatively, you can use the second-from-the-left toolbar button (denoted with a red x sign) in the **Scenarios** tab.

NOTE: During this action, the spreadsheet is not modified in any way.

5.4.10 Properties

With this option, you can modify the properties of the currently selected scenario.

To modify the properties of the selected scenario:

1. Select the row corresponding to the scenario you wish to modify in the **Scenarios** tab.
2. Select **Scenarios > Properties** from the **Data** menu.
3. Make the appropriate modifications following the steps of the wizard. The settings differ, depending on the type of the EA that is used in the certain scenario.

NOTE: Alternatively, you can use the corresponding toolbar button in the **Variables** tab.

NOTE: During this action, the spreadsheet is not modified in any way.

5.4.11 Clone

With this option, you can clone an already defined scenario. This option is very useful when you want to analyze similar versions of a certain EA configuration.

To clone an already defined scenario:

1. Select the row corresponding to the scenario you wish to clone in the **Scenarios** tab.
2. Select **Scenarios > Clone** from the **Data** menu.
3. A clone of the selected scenario is created at the end of the list. By default, its name is "Clone of X", where X the title of the source scenario.

NOTE: Alternatively, you can use the corresponding toolbar button in the **Scenarios** tab.

NOTE: The type of the EA used in a scenario cannot be changed, even if cloned.

NOTE: During this action, the spreadsheet is not modified in any way.

5.4.12 Move top

With this option, you can move the selected scenarios to the top of the list.

To move the selected scenarios to the top of the list:

1. Select one or more rows in the list of scenarios.
2. Select **Scenarios > Move top** from the **Data** menu.

NOTE: Alternatively, you can use the corresponding toolbar button in the **Scenarios** tab.

NOTE: During this action, the spreadsheet is not modified in any way.

5.4.13 Move up

With this option, you can move the selected scenarios upwards by one slot.

To move the selected scenarios upwards by one slot:

1. Select one or more rows in the list of scenarios.
2. Select **Scenarios > Move up** from the **Data** menu.

NOTE: Alternatively, you can use the corresponding toolbar button in the **Scenarios** tab.

NOTE: During this action, the spreadsheet is not modified in any way.

5.4.14 Move down

With this option, you can move the selected scenarios downwards by one slot.

To move the selected scenarios downwards by one slot:

1. Select one or more rows in the list of scenarios.
2. Select **Scenarios > Move down** from the **Data** menu.

NOTE: Alternatively, you can use the corresponding toolbar button in the **Scenarios** tab.

NOTE: During this action, the spreadsheet is not modified in any way.

5.4.15 Move bottom

With this option, you can move the selected scenarios to the bottom of the list.

To move the selected scenarios to the bottom of the list:

1. Select one or more rows in the list of scenarios.
2. Select **Scenarios > Move bottom** from the **Data** menu.

NOTE: Alternatively, you can use the corresponding toolbar button in the **Scenarios** tab.

NOTE: During this action, the spreadsheet is not modified in any way.

Chapter



VI

6 Window

6.1 Window menu

With this option, you can arrange the windows and set the visibility of the status/progress window. In the **Window** menu you can select one of the following options:

- Auto arrange
- Arrange
- Status/progress window

6.2 Auto arrange

With this option, you can toggle the auto-arrangement of the windows.

To activate/deactivate the auto-arrangement of the windows:

1. Select **Auto-arrange** from the **Window** menu.

When active, auto-arrange changes the dimensions of the Microsoft Excel window (on the left of the screen) and the status/progress window (bottom right, if visible) according to the dimensions of the main program window.

6.3 Arrange

With this option, you can arrange the dimensions and position of the windows.

To arrange the dimensions and position of the windows:

1. Select **Arrange** from the **Window** menu.

As a result, the dimensions of the Microsoft Excel window (on the left of the screen) and the status/progress window (bottom right, if visible) are modified, according to the dimensions of the main program window.

6.4 Status/progress window

With this option, you can toggle the visibility of the status/progress window.

To toggle the visibility of the status/progress window:

1. Select **Status/progress window** from the **Window** menu.

When visible, the status/progress window shows information about the progress of the current analysis, the overall progress etc.

Chapter

VII

7 Help

7.1 Help menu

In the **Help** menu you can select one of the following options:

- Contents
- User guide
- Tutorials
- Tip of the day
- Unit conversion
- TechnoLogismiki website
- Buy products
- TechnoLogismiki NOMOS
- TechnoLogismiki Live!
- About the program

7.2 Contents

With this option, you can access the online help which contains detailed information regarding the usage of the program.

To view the online help:

1. Click **Contents** from the **Help** menu.
2. The online help appears.

NOTE: If an error message appears then the online help has not been installed. You can install the online help from the installation CD or the Internet.

7.3 User guide

With this option, you can access the user guide which contains detailed information regarding the usage of the program.

To view the user guide:

1. Click **User Guide** from the **Help** menu.
2. The user guide appears.

NOTE: If an error message appears then the online help has not been installed. You can install the online help from the installation CD or the Internet.

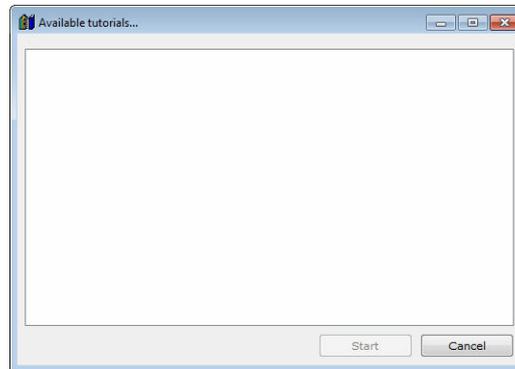
NOTE: Adobe Acrobat Reader or a similar program that can display pdf files is required in order to view or print the user guide.

7.4 Tutorials

With this option, you can access the tutorials of the program. The tutorials are step-by-step examples that allow you to decrease the learning cycle of the programs dramatically.

To access the tutorials:

1. Click **Tutorials** from the **Help** menu.
2. The tutorial selection dialog box appears.
2. Select the appropriate tutorial and click **Start** to proceed. Click **Cancel** to close the dialog box.



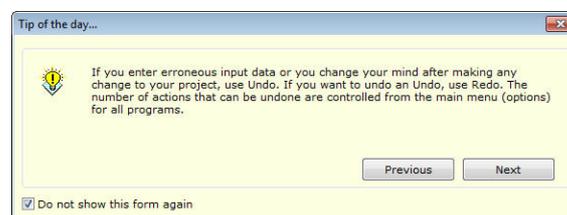
NOTE: The number and content of the tutorials is changed frequently. Use the live update system of TechnoLogismiki's products to download the latest tutorials.

7.5 Tip of the day

With this option, you can access the tip database of the program. The tips are short guidelines regarding the usage of the programs which may be of great help to the user.

To access the tips:

1. Click **Tip of the day** from the **Help** menu.
2. The tip of the day form appears.
3. Check **Do not show this form again** to prevent the program from showing the tip of the day when starting. Press the **Previous/Next** buttons to browse all available tips.
4. Press **Esc** to close the form.



NOTE: The number and content of the tips is changed frequently. Use the live update system of TechnoLogismiki's products to download the latest tips.

7.6 Unit conversion

With this option, you can access the unit conversion tool. You can find more information about its usage in its help system.

To launch the unit conversion tool:

1. Click **Unit conversion** from the **Help** menu.
2. The unit conversion tool is launched.



NOTE: If an error message appears then the unit conversion tool has not been installed. You can install the unit conversion tool from the installation CD or the Internet.

7.7 TechnoLogismiki website

With this option, you can load on your Internet browser the website of TechnoLogismiki's.

7.8 Buy products

With this option, you can load on your Internet browser the main product page of TechnoLogismiki's website.

7.9 TechnoLogismiki NOMOS

With this option, you can load on your Internet browser the **NOMOS** service of TechnoLogismiki.

7.10 TechnoLogismiki Live!

With this option, you can load on your Internet browser the **Live!** service of TechnoLogismiki.

7.11 About the program

With this option, a form containing the name, version and licence information of the program appears.

To show this form:

1. From the **Help** menu, select **About the program**.
2. The form appears.
3. Click anywhere on the form or hit ESC to close the form.

Chapter



8 Appendix

8.1 Genetic Algorithms (GA)

Evolutionary Algorithms (EAs) are population-based, biologically inspired stochastic algorithms that have been successfully applied to a wide variety of problems, including optimization, system identification and simulation [1]. The most widely known type of EAs are the **Genetic Algorithms (GAs)**, which were initially conceived by John Holland [2] and later employed in a plethora of problems.

GAs mimic the process of natural evolution, which is based on the survival of the fittest. They are routinely used to generate useful solutions to optimization and search problems.

In a GA, a population of strings (called chromosomes), which encode candidate solutions to a problem, evolves toward better solutions. The evolution starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness), and modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. The fitness function measures the quality of the represented solution and is problem dependent.

The main components of a GA are described below. For more information, one can refer to [3].

Initialization

Initially individual solutions are randomly generated to form an initial population. The population size depends on the nature of the problem. Traditionally, the population is generated randomly, covering the entire range of possible solutions (the search space).

Selection

During each successive generation, a proportion of the existing population is selected to breed a new generation. Individual solutions are selected through a fitness-based process, where fitter solutions (as measured by a fitness function) are typically more likely to be selected. Certain selection methods rate the fitness of each solution and preferentially select the best solutions.

Reproduction: Crossover and Mutation

The next step is to generate a second generation population of solutions from those selected through genetic operators: crossover (also called recombination), and/or mutation.

For each new solution to be produced, a pair of "parent" solutions is selected for breeding from the pool selected previously. By producing a "child" solution using the

above methods of crossover and mutation, a new solution is created which typically shares many of the characteristics of its "parents". Mutation is also stochastically applied to the offspring, which allows the GA to avoid entrapment into local optima and explore new regions of the search space. New parents are selected for each new child, and the process continues until a new population of solutions of appropriate size is generated.

These processes ultimately result in the next generation population of chromosomes that is different from the initial generation. Generally the average fitness will have increased by this procedure for the population, since only the best organisms from the first generation are selected for breeding, along with a small proportion of less fit solutions, for reasons already mentioned above.

Elitism

Elitism is used in order to ensure that (at least) the best individual of the current generation survives to the next one. Typically one individual, i.e. the best, is transferred as-is to the next generation.

Termination

This generational process is repeated until a termination condition has been reached. Common terminating conditions are:

- A solution is found that satisfies minimum criteria
- Fixed number of generations reached
- Allocated budget (computation time/money) reached
- The highest ranking solution's fitness is reaching or has reached a plateau such that successive iterations no longer produce better results
- Combinations of the above

GA process

The GA process can be described by the following pseudo-code:

- Choose the initial population of individuals
- Evaluate the fitness of each individual in that population
- Repeat on this generation until termination (time limit, sufficient fitness achieved, etc.):
 - Select the best-fit individuals for reproduction
 - Breed new individuals through crossover and mutation operations to give birth to offspring
 - Evaluate the individual fitness of new individuals

[1] Eiben, A. E., Smith, J. E.: Introduction to evolutionary computing. Springer, New York (2003)

[2] Holland, J. H.: Adaptation in natural and artificial systems. Univ. of Michigan Press, Ann Arbor, MI (1975)

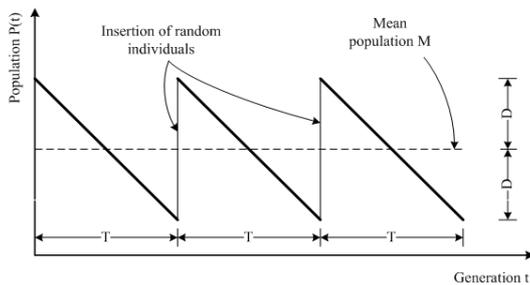
[3] [Online] Wikipedia: http://en.wikipedia.org/wiki/Genetic_algorithms

8.2 Sawtooth GA

Sawtooth Genetic Algorithm (Sawtooth GA) is a GA variant that was proposed by Koumouis and Katsaras [1] at National Technical University of Athens (NTUA).

Sawtooth GA uses variable population size and periodic partial reinitialization of the population in the form of a saw-tooth function. The aim is to enhance the overall performance of the algorithm relying on the dynamics of evolution of the GA and the synergy of the combined effects of population size variation and reinitialization.

The Sawtooth scheme is characterized by mean population M , amplitude of variation D and period of variation T . The population undergoes partial reinitialization every T generations, as shown in the following figure:



It has been shown that Sawtooth GA exhibits improved performance with respect to Standard GA and micro GA for a wide range of problems [1].

[1] Koumouis, V. K., Katsaras, C. P.: A Saw-Tooth Genetic Algorithm Combining the Effects of Variable Population Size and Reinitialization to Enhance Performance. IEEE Trans. Evol. Comput. 10(1), 19–28 (2006)

8.3 Micro GA

Micro GA is a GA variant proposed by Goldberg [1] and first implemented by Krishnakumar [2]. Micro-GA differs from Standard GA in the following aspects:

- The population size is small (typically 5-10 chromosomes)
- There is no mutation
- Crossover is applied with a probability equal to unity
- Elitism is applied with one individual
- The population is reinitialized (except for the best chromosome) whenever the genetic diversity falls below a specified threshold. Genetic diversity can be evaluated based on the different bits ratio of the rest of the chromosomes with respect to the best chromosome of the population.

[1] Goldberg, D.E. (1989) "Sizing populations for serial and parallel genetic algorithms", Proceedings of the third International Conference on Genetic Algorithms (ICGA 89), pp.70-79.

[2] Krishnakumar, K. (1989) "Micro-genetic algorithms for stationary and nonstationary function optimization", Proc. SPIE Intelligent Control Adaptive Systems,

1942-1948.

8.4 Simple Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is a population-based stochastic optimization technique suitable for global optimization with no need for direct evaluation of gradients. The method, introduced by Kennedy and Eberhart [1], mimics the social behavior of flocks of birds and swarms of insects and satisfies the five axioms of swarm intelligence, namely proximity, quality, diverse response, stability and adaptability [2]. The algorithm searches the Design Space (DS) by adjusting the trajectories of individuals, called "particles", viewed as moving points in the DS. These particles are attracted towards the positions of both their personal best solution and the best solution of the swarm in a stochastic manner [3].

In simple PSO, we assume that the population consists of p individuals. Within the DS each individual is characterized by its position and velocity, determined at time instant k by the corresponding vectors \mathbf{x}_k and \mathbf{v}_k . Initially, the particles are distributed randomly in the box-constrained search space, as follows:

$$\mathbf{x}_{LB} \leq \mathbf{x}_0^d \leq \mathbf{x}_{UB} \quad \forall d \in \{1, 2, \dots, p\}$$

where, \mathbf{x}_{LB} and \mathbf{x}_{UB} are vectors defining the lower and upper values of the variables, respectively.

The maximum velocity of the individuals is evaluated so that in one time step an individual may travel up to a certain fraction of the search space:

$$\mathbf{v}_0^{\max} = \gamma(\mathbf{x}_{UB} - \mathbf{x}_{LB})$$

The position vector of individual d at the next time instant $k+1$ is given as:

$$\mathbf{x}_{k+1}^d = \mathbf{x}_k^d + \mathbf{v}_{k+1}^d$$

where the time step Δt between the distinct time instants is assumed to be equal to unity. The velocity vector \mathbf{v}_{k+1}^d is given as:

$$\mathbf{v}_{k+1}^d = w_k \mathbf{v}_k^d + c_1 \mathbf{r}_1 \circ (\mathbf{p}_k^d - \mathbf{x}_k^d) + c_2 \mathbf{r}_2 \circ (\mathbf{p}_g - \mathbf{x}_k^d)$$

where, w_k is the inertia factor at time instant k ; c_1, c_2 are the cognitive and social parameters of the algorithm, respectively; \mathbf{p}_k^d is the best ever position vector of individual d at time instant k ; \mathbf{p}_g is the best ever position vector amongst all individuals at time instant k ; $\mathbf{r}_1, \mathbf{r}_2$, are vectors of random variables with uniform distribution in the interval $[0,1]$ and the (\circ) operator indicates element-by-element multiplication. This formulation is compatible with the diverse "classical" version of PSO [4].

[1] Kennedy J, Eberhart RC. Particle swarm optimization. In: Proceedings, of IEEE International Conference on Neural Networks IV, Perth Australia, IEEE press Piscataway, NJ, 1995; p. 1942–1948.

[2] Millonas MM. Swarms, Phase Transitions and Collective Intelligence. In: C. G.

Langton editor. Artificial Life III, Reading, MA, Addison Wesley, 1994; p. 417–445.
 [3] Clerc M, Kennedy J. The Particle Swarm - Explosion, Stability, and Convergence in a Multidimensional Complex Space. IEEE T Evolut Comput 2002;6(1):58-73.
 [4] Wilke DN, Kok S, Groenwold AA. Comparison of linear and classical velocity update rules in particle swarm optimization: Notes on diversity. Int J Numer Meth Eng 2007;70:962-984.

8.5 Enhanced PSO

The enhanced PSO variant is based on the work of Fourie and Groenwold [1], as follows:

The initial maximum velocity of the individuals is evaluated so that in one time step an individual may travel up to a certain fraction of the search space:

$$\mathbf{v}_0^{\max} = \gamma(\mathbf{x}_{UB} - \mathbf{x}_{LB})$$

If the best solution found in the whole swarm is not improved over a period of h consecutive steps, then it is assumed that the velocities are large and the algorithm cannot locate better solutions due to overshooting. For this reason, both the inertia factor and the maximum velocity are decreased as follows:

$$\text{If } OF(\mathbf{p}_g)|_k = OF(\mathbf{p}_g)|_{k-h} \Rightarrow w_{k+1} = a w_k, \mathbf{v}_{k+1}^{\max} = \beta \mathbf{v}_k^{\max}$$

The craziness operator assigns a random velocity vector to an individual resulting in its moving away from the swarm and thus exploring other regions of the search space. The operator is activated with a probability P_{cr} as follows:

$$\text{If } r < P_{cr} \Rightarrow \text{randomly assign } \mathbf{v}_{k+1} \text{ with } \mathbf{0} < \mathbf{v}_{k+1} \leq \mathbf{v}_{k+1}^{\max} \forall \text{ particle } d$$

where r is a random variable with uniform distribution in the interval $[0,1]$.

The algorithm employs both an elite particle and an elite velocity. The individual with the worst performance is moved to the best ever position of the swarm which implies a gradual shift towards the region where good solutions reside:

$$\mathbf{x}^{pe} = \mathbf{p}_g$$

In addition, if the velocity vector \mathbf{v}_k^d resulted in an improvement of \mathbf{p}_g , then:

$$\mathbf{x}_{k+1}^d = \mathbf{p}_g + C_3 \mathbf{r}_3 \circ \mathbf{v}^{pe}$$

where, $\mathbf{v}^{pe} = \mathbf{v}_k^d$, C_3 is a parameter of the algorithm and \mathbf{r}_3 is a vector of random variables with uniform distribution in the interval $[0,1]$. Note the difference with respect to [1], where a single random variable r is used to multiply the whole vector.

[1] Fourie PC, Groenwold AA. The particle swarm optimization algorithm in size and shape optimization. Struct Multidisc Optim 2002;23(4):259–267.

8.6 Differential Evolution (DE)

Differential Evolution (DE) is a relative new stochastic method which has attracted the attention of the scientific community. It was introduced by Storn and Price [1] and has approximately the same age as PSO. However, it bears no natural paradigm. An early version was initially conceived under the term "Genetic Annealing" and published in a programmer's magazine [2]. The DE algorithm is extremely simple; the uncondensed C-style pseudocode of the algorithm spans less than 25 lines [2].

According to the classic version of DE, a population of p individuals is randomly dispersed within the design space. The population is denoted as $\mathbf{P}_{x,g}$, as follows:

$$\mathbf{P}_{x,g} = (\mathbf{x}_{i,g}), \quad i = 0, 1, \dots, p-1, \quad g = 0, 1, \dots, g_{\max}$$

$$\mathbf{x}_{i,g} = (x_{j,i,g}), \quad j = 0, 1, \dots, N_p - 1.$$

where $\mathbf{P}_{x,g}$ = array of p vectors (individuals), $\mathbf{x}_{i,g}$ = N_p -dimensional vector representing a solution, i =index for vectors, g =index for generations, j =index for design variables, g_{\max} =maximum number of generations and the parentheses indicate an array.

At each evolutionary step (generation), a mutated population $\mathbf{P}_{v,g}$ is formed based on the current population $\mathbf{P}_{x,g}$, as follows:

$$\mathbf{v}_{i,g} = \mathbf{x}_{r_0,g} + F(\mathbf{x}_{r_1,g} - \mathbf{x}_{r_2,g})$$

where, r_0, r_1, r_2 are random indices in $\{0, 1, \dots, p-1\}$ and F is a scalar DE algorithm parameter. Next, a trial population $\mathbf{P}_{u,g}$ is formed, as follows:

$$\mathbf{u}_{i,g} = (\mathbf{u}_{j,i,g}) = \begin{cases} \mathbf{v}_{j,i,g}, & \text{if } (\text{rand}_j(0,1) \leq Cr \text{ or } j = j_{\text{rand}}) \\ \mathbf{x}_{j,i,g}, & \text{otherwise} \end{cases}$$

where, rand_j is a random number with uniform distribution in $(0,1)$ that is sampled anew each time, j_{rand} is a random index in $\{0, 1, \dots, p-1\}$ that ensures that at least one design variable will originate from the mutant vector, and Cr is a scalar DE algorithm parameter in the range $(0,1]$. The final step is the selection criterion, which is greedy:

$$\mathbf{x}_{i,g+1} = \begin{cases} \mathbf{u}_{i,g}, & \text{if } OF(\mathbf{u}_{i,g}) \leq OF(\mathbf{x}_{i,g}) \\ \mathbf{x}_{i,g}, & \text{otherwise} \end{cases}$$

The classic DE variant is denoted "rand/1/bin" [2], where "rand" indicates that base vectors are randomly chosen, "1" means that only one vector difference is used to form the mutated population, and the term "bin" (from binomial distribution) indicates that uniform crossover is employed during the formation of the trial population.

Another variant is denoted "best/1/bin" [2], where "best" indicates that the base vector used is the currently best vector in the population. Thus, the mutated population $\mathbf{P}_{v,g}$ is formed based on:

$$\mathbf{v}_{i,g} = \mathbf{x}_{best,g} + F(\mathbf{x}_{r1,g} - \mathbf{x}_{r2,g})$$

In addition, "jitter" may be introduced to the parameter F and the previous equation is modified as follows:

$$F_j = F + d(\text{rand}_j(0,1) - 0.5)$$

$$\mathbf{v}_{i,g} = \mathbf{x}_{best,g} + F_j(\mathbf{x}_{r1,g} - \mathbf{x}_{r2,g})$$

where, rand_j is a random number with uniform distribution in (0,1) that is sampled anew each time and d is the magnitude of the jitter.

A third variant is also available, denoted as "rand-best/1/bin". This variant is simply a fixed mix of the "rand/1/bin" and "best/1/bin". Define r_b as the expected ratio of evaluations with "rand/1/bin" as opposed to the ones with "best/1/bin". If $\text{rand}_j < r_b$, then evolution is carried out according to "rand/1/bin". In the opposite case, "best/1/bin" is employed. Usual value of r_b is 0.25.

[1] R. Storn, K. Price, "Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces", *Journal of Global Optimization*, 11, 341–359, 1997.

[2] R. Storn, K. Price, J.A. Lampinen, "Differential Evolution – a practical approach to global optimization", Springer, 2005.

Keyword index

- 6 -

696/74 8

- A -

about 70
add 26, 29, 32
address 10
American 8
appearance 22, 24
arrange 66
ASCE 8
auto-arrange 66
automatic 22

- B -

backup 17, 22
bck 17, 22
bottom 29, 32, 64
browser 70
button 24, 70

- C -

calculation 22
CD 68
cell 26, 29, 32
change 18, 19, 22
check 22
chromosome 27
clone 62
close 19
command 21
communicator 70
compatibility 17
computer 22
configuration 9
confirmation 22
constraint 29, 30, 31, 32
content 68, 69
contents 68

conversion 70
create 17
cursor 24
Custom GA 33

- D -

data 24, 26
date 22, 70
decimal 22
delete 17, 22, 27, 30, 62
deletion 22
description 24
desktop 17
detailed 68
Differential Evolution 77
digit 22
disk 18
down 29, 31, 63
download 68, 69

- E -

edit 21
email 10
English 22
Enhanced PSO 54, 58, 76
error 68
Excel 26, 29, 32
execution 22
exit 19, 22
explorer 70

- F -

fax 10
file 17, 18, 19

- G -

GA 72
gene 27
general 22
Genetic Algorithms 72
gravity 8
Greek 8, 22
guide 68

- H -

hard 18
help 68, 69, 70
high 27

- I -

ignore 19
installation 68
interactive 10
Internet 10, 22, 68, 69, 70

- L -

language 22
link 70
Live! 70
liveupdate 68, 69
load 17
local 17, 18
locked 17
low 27

- M -

main 12, 14
manual 22, 68
menu 26, 66
message 22, 68
Micro GA 45, 74
Microsoft 26, 29, 32
minimum 9
modify 18
move 28, 29, 31, 32, 63, 64
mozilla 70

- N -

netscape 70
network 8, 17, 18
new 17, 22
niching 27
NOMOS 70
number 21

- O -

online 68
open 17
opera 70

- P -

pack 22
place 22
plane 8
preference 22
previous 17
privacy 22
program 17, 68, 70
progress 14, 66
project 17, 18
properties 30, 62
property 27

- R -

redo 21
regulation 8
remove 22
requirement 9

- S -

satisfactory 9
save 18, 19, 22
Sawtooth GA 39, 74
scenario 32, 62, 63, 64
sewer 8
Simple PSO 50, 75
Standard GA 32
status 14, 66
store 70
support 10, 70
system 9

- T -

technical 70
TechnoLogismiki 70
telephone 10

temporary 22
terminate 19
tip 69
title 70
tool 70
toolbar 24
tooltip 24
top 28, 31, 63
tutorial 68
type 17, 27

- U -

undo 21
unit 70
up 28, 31, 63
usage 68
user 68

- V -

variable 26, 27, 28, 29
version 17, 22, 70
view 8

- W -

web 70
window 66
wizard 32, 33, 39, 45, 50, 54, 58
WPCF 8

- Z -

zero 22

This page was intentionally left blank.