RMOP-GUI Manual

Espinoza H. and Coma M. and Cervello J.

December 17, 2012

Contents

1 Introduction

RMOP, the Robust Multi-objective and Multidisciplinary Optimization Platform, is a user-friendly and easy-to-use optimization tool. Based on the Hybrid -games techniques developed by CIMNE researchers, it provides the best environment for solving optimization problems. Fully compatible with all type of solvers, RMOP provides an amazing optimization experience leading to optimal results.

Now we describe basic concepts of optimization.

Optimization Problem: cosists in finding the design variables values such that the objective functions values are optimum. By optimum we mean maximum or minimum. In RMOP we always pose the optimization problem as a minimization problem, so we always want to minimize the objective functions.

Individual: it is a given simulation case and it is fully characterized by a set of design variable values.

Design Variable: it is a value (real or integer) that modifies the behavior of the individual in terms of the objective function values.

Objective Function: it is a real valued function. Its value represents how good is an individual. In RMOP we always want to minimize the objective function, so an individual with a lower value of the objective function will be better.

RMOP is capable of dealing with an unlimited number of objective functions and design variables. The RMOP Graphical User Interface (RMOP GUI) let you define these two important parameters.

2 Pre-Processing

This section describes the pre-processing features of the Graphical User Interface (GUI) of the Robust Multi-objective Optimization Platform (RMOP)

2.1 Polulation

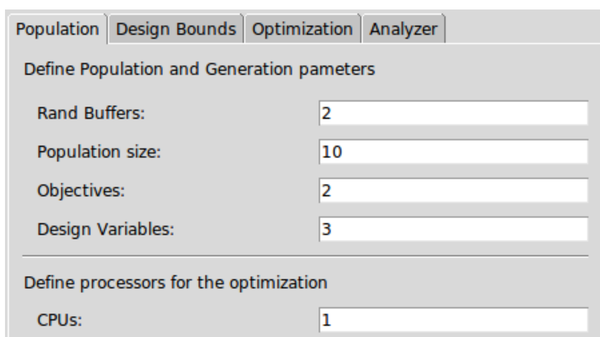


Figure 1: Population

Rand Buffers: it is a calculation buffer to evaluate individuals. Should be greater than the population size

Population Size: it is the number of individuals in the population. Should be an even number greater or equal than 2.

Objectives: it is the number of objective functions involved in the optimization. These objective functions are going to be minimized.

Design Variables: it is the number of design variables to use in the optimization.

CPUs: RMOP is designed to run in parallel, this field defines the number of processors assigned to the optimizer.

2.2 Design Bounds

This part of the GUI let us inform the type of design variables: Integer or Real and their bounds. The order in which the design variables are informed is important, because that is the order in which they are passed to the analyzer.

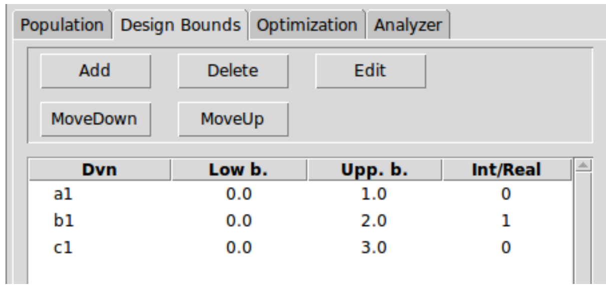


Figure 2: Design Bounds

Buttons Add, Delete and Edit let modify one design variable at a time. Buttons Move Up and Move Down let change the order of appearance the design variables. The Design Variable Identifier is just informative and is used only within the GUI and the optimization project.

In figure 3 it can be seen the window that appears when you hit the add or edit button. As can be seen, there you can choose the type of design variable (integer or real), assign a identifier to it and its bounds.

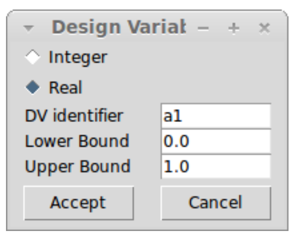


Figure 3: Design Bounds Window

2.3 Optimization

Game Strategy

- **Pareto Game**: uses the classical hierarchical multi-population Pareto optimality.
- **Hybrid Game (Pareto & Nash)**: uses a combination of Pareto and Nash game strategies. It consists in one Pareto Player and many Nash Players and can produce a Nash-equilibrium and Pareto non-dominated solutions simultaneously

Optimization Method

- **Genetic Algorithm (GA)**: uses a modified version of the Non-domitated Sorting Genetic Algorithm II (NSGA-II)

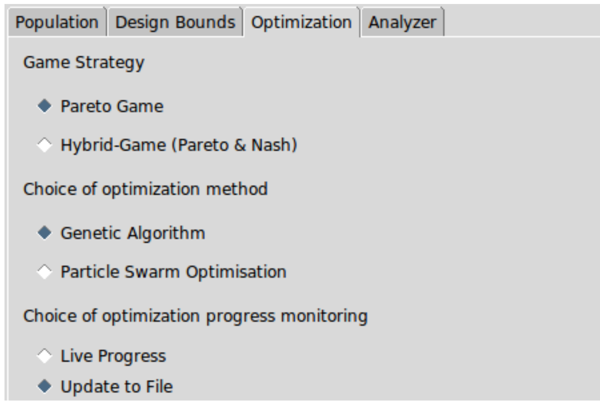


Figure 4: Optimization

- **Particle Swarm Optimization (PSO)**: this algorithm treats each individual in the population as a particle, then it moves each particle in the search space to a given position and with a certain velocity (thereby the name of swarm). The swarm moves to local minima and search for other minima if there are better ones.

Choice of Optimization Progress

- **Live progress**: does the same as Update to file and additionally it shows a pop-up window with a live progress of the optimization.
- **Update to file**: logs all the outputs from the analyzer and RMOP to a file named HPRMOP.log located in the project directory.

2.4 Analyzer

Analyser file The analyser is an executable file (in Windows it can be a .bat or .exe file, in Linux it can be a .sh or binary executable). It is in charge of evaluating one individual. It will be called several times by HPRMOP in order to evaluate one individual for a given set of design variables. Finally, it will write the file Eval.individual which contains the values of the objective functions for that individual.

Termination Criteria Stablishes the criteria to stop the optimization process. Function evaluation: the optimization process will stop when a given number of function evaluations is reached. It represents the number of individuals to be analysed during the optimization process.

Elapsed Time: the optimization process will stop when it has run during the specified period of time. Using this criteria you will know exactly when the optimization will finish, independently of the number of function evaluations or the value reached by the objective functions.

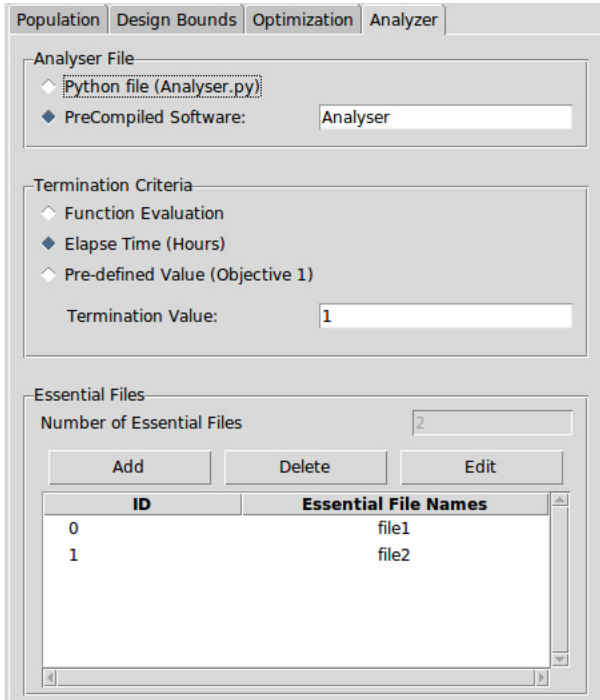


Figure 5: Analyzer

Pre-defined Value: the optimization process will stop when the objective function reaches a given value. This criteria might lead to an infinite optimization process. If you want to stop the optimization you can hit the stop button.

Termination Value: it is either the number of function evaluations (integer), the elapsed time in hours (real) or the pre-defined value (real)

Essential Files Essential files are files needed in the CPU working folder by the Analyser in order to run a case. For example: geometry files (when geometry is not optimized), data files not modified during optimization, etc. All essential files must be in the project folder, then HPRMOP will copy them to each of the CPU working folders.

3 Post-Processing

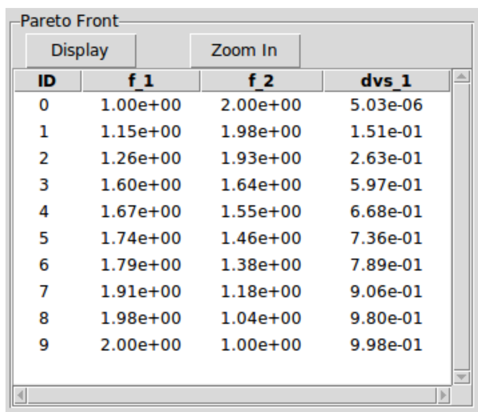
This section describes the post-processing features of the Graphical User Interface (GUI) of the Robust Multi-objective Optimization Platform (RMOP)

Figure 6: Status Icons

3.1 Status Icons

These icons show the status of the optimization. RMOP GUI starts with the first icon. When you launch the calculation it changes to the second icon and automatically switches back to the first icon when the optimization finishes. When the optimization is running (second icon) you can always stop it and the icon will switch to the third one.

3.2 Pareto Front Table



ID	f_1	f_2	dvs_1
0	1.00e+00	2.00e+00	5.03e-06
1	1.15e+00	1.98e+00	1.51e-01
2	1.26e+00	1.93e+00	2.63e-01
3	1.60e+00	1.64e+00	5.97e-01
4	1.67e+00	1.55e+00	6.68e-01
5	1.74e+00	1.46e+00	7.36e-01
6	1.79e+00	1.38e+00	7.89e-01
7	1.91e+00	1.18e+00	9.06e-01
8	1.98e+00	1.04e+00	9.80e-01
9	2.00e+00	1.00e+00	9.98e-01

Figure 7: Pareto Front Table

The Pareto Front Table contains numerial data of the Pareto Front. This table has three main columns and each row represents one individual that belongs to the Pareto Set. The first column is an identification of the individual, for example ID=0 is the individual with the best value of the Objective Function 1. The second column has as many sub-columns as objective functions. It contains the values of the objective functions. The third column has as many sub-columns as design variables. It contains the design variables values characterizing a given individual.

3.3 Pareto Front Graph

It is a graphical representation of the Pareto Set and known as Pareto Front or Pareto Frontier. Given an optimization problem with two objective functions, it is an easy way to asses the solution obtained. For optimization problems with

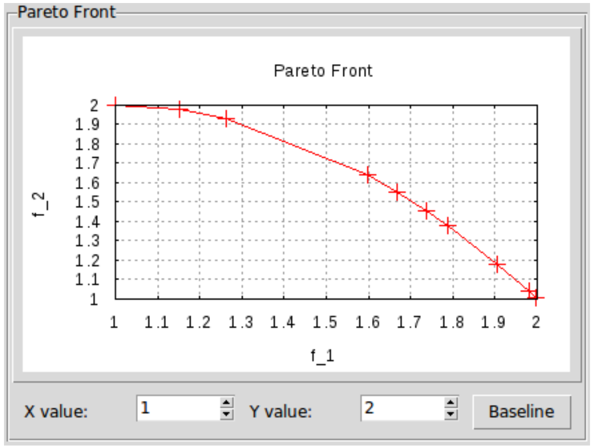


Figure 8: Pareto Front

more than two objective functions, the user can choose the obective function to plot in each axis and replot as desired.

Clicking over the graph opens a bigger graph for easier visualization.

Additionally, a baseline design can be added to the Pareto Front Graph by clicking in the Baseline button. A window as shown in figure 9 appears where you can enter the information. Check the box and refresh the plots to view the baseline design plotted.

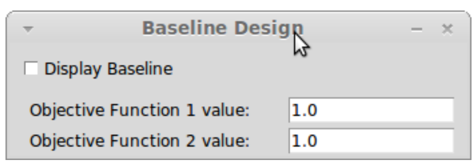


Figure 9: Baseline Design in Pareto Front Graph

3.4 Convergence History Graph

It is a graphical representation of the evolution of the optimization as a function of the number of individuals evaluated. The convergence history graph lets the you know how the optimization problem is converging to the optimal solution.

Let us suppose you are solving an optimization problem and you just put a long time for it to run. After some time you realize the problem is not converging and you need to stop and check what is going wrong. On the other hand, you might realize the problem has converged quite a lot and it is not converging anymore, so you might want to stop.

Clicking over the graph opens a bigger graph for easier visualization.

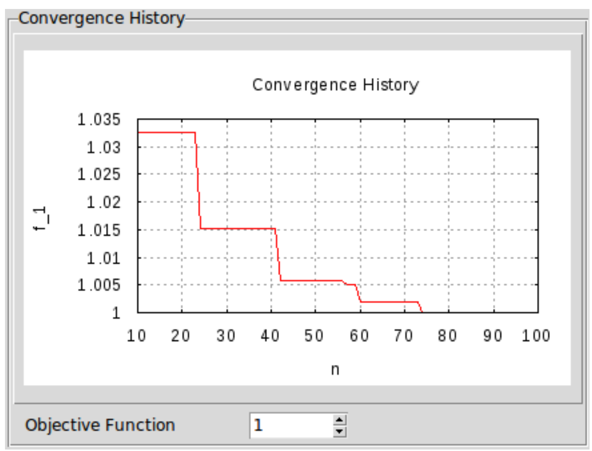


Figure 10: Convergence History