

A Tunable Generator of Instances of Permutation-based Combinatorial Optimization Problems: Manual for the R code

Leticia Hernando, Alexander Mendiburu and
Jose A. Lozano

September 23, 2014

1 Introduction

In this document we include the necessary information for using the generator of instances of Combinatorial Optimization Problems proposed in the paper:

L. Hernando, A. Mendiburu and J.A. Lozano, "A Tunable Generator of Instances of Permutation-based Combinatorial Optimization Problems", IEEE Transactions on Evolutionary Computation, 2014.

The code of the generator is in R-project. We explain in detail the input parameters needed, the aim of each function and the organization of the output file. The *glpkAPI* package is needed to solve the linear programming problem.

2 Input parameters

The input parameters of the generator are:

- Permutation size: $n \in \mathbb{N}$
- Number of Generalized Mallows models in the generator (local optima in the output instance): $m \in \mathbb{N}$
- File with the permutations that will be the local optima in the output instance. Each permutation should be in a different row, and the permutations need to be sorted such that the first one will be the global optimum, the second one will be the local optimum with the highest objective function value, etc., with the last one being the local optimum with the lowest objective function value. An example of an input file with the local optima is *LocalOptima.txt*.
- File with the distances vector of the global optimum to the rest of the local optima. An example of an input file with the distances between the global optimum and the local optima is *Distances.txt*.

- File with the parameters Θ . Each row should contain $\theta_i^1, \theta_i^2, \dots, \theta_i^{n-1}$ for each i . So that the file should have m rows and $n - 1$ columns. Notice that, if the models considered are the Mallows models (not Generalized Mallows models), the parameter θ_i should be repeated $n - 1$ times in each row. An example of an input file with the Θ parameters when considering the Generalized Mallows model is *ThetaGM.txt*, and for the use of the Mallows model *ThetaM.txt*.
- The linear function to optimize in the linear programming problem G . The code is programmed to choose among the three functions explained in the paper. Thus, enter *max*, *min* or *sim* to choose G_{MaxGO} , G_{MinGO} or G_{SimAB} , respectively.
- The type of distance used. It is programmed to choose between the Kendal-tau and the Cayley distances, so K or C , for each distance respectively, must be entered.
- The name of the file where the results will be written.

3 The functions

- **generateCOP**

This is the main function. The input parameters are: $n, m, G, FileSigma, FileDistances, FileTheta, typeDistance$ and $FileOut$ (explained in the Section 2). The files *FileSigma*, *FileDistances* and *FileTheta* are read. The values of the normalization terms of the models $Z(\theta_i)$ are calculated and the linear programming problem is solved. Finally, the results are written in an output file *FileOut*. This function calls: *Zvalue.R* and *LinearProg.R*.

- **Zvalue**

This function calculates the normalization terms $Z(\theta_i)$ of the Generalized Mallows models. It returns the vector: $(Z(\theta_1)Z(\theta_2) \cdots Z(\theta_m))$

- **LinearProg**

Here, the linear programming problem is solved. It uses the function *solveInteriorGLPK* inside the *glpkAPI* package. It chooses among the three linear functions G and it returns an object of class *glpkPtr*.

4 The output

Each row of the *outFile* contains the information of each local optimum of the generated instance: local optimum, objective function value, spread parameters.

σ_1	$f(\sigma_1)$	$\theta_1^1 \theta_1^2 \dots \theta_1^{n-1}$
σ_2	$f(\sigma_2)$	$\theta_2^1 \theta_2^2 \dots \theta_2^{n-1}$
\dots	\dots	\dots
σ_m	$f(\sigma_m)$	$\theta_m^1 \theta_m^2 \dots \theta_m^{n-1}$