Task Force on Modern Heuristic Optimization Test Beds
Working Group on Modern Heuristic Optimization
Intelligent Systems Subcommittee
Analytic Methods in Power Systems (AMPS) Committee

Competition

on

Emerging heuristic optimization algorithms for expansion planning and flexibility optimization in sustainable electrical power systems

Problem Definitions and Implementation Guidelines

Santiago P. Torres[1], José L. Rueda[2], Sergio R. Rivera[3], Rolando M. Pringles[4]

[1]Head of Department of Electrical Engineering, Universidad de Cuenca, Ecuador
[2]Department of Electrical Sustainable Energy, Delft University of Technology, Delft, Netherlands
[3]Head of Electrical Engineering Doctoral and Master Programs, Universidad Nacional de Colombia, Colombia
[4]Instituto de Energía Eléctrica, Universidad Nacional de San Juan, Argentina

santiago.torres@ucuenca.edu.ec, J.L.RuedaTorres@tudelft.nl, srriverar@unal.edu.co, rpringles@gmail.com

November 2018

TABLE OF CONTENTS

1. AIM OF THE COMPETITION

Like in other engineering fields, the application of heuristic optimization algorithms to solve power system optimization problems is attracting researchers' attention due to their potential to deal with inherent mathematical complexities, such as high dimensionality, non-linearity, non-convexity, multimodality, and discontinuity of the search space. Moreover, the experience gained in the development and use of different types of heuristic optimization concepts motivates further research efforts to devise novel mechanisms for improved search exploration and exploitation.

One of the targets of the Working Group on Modern Heuristic Optimization under the IEEE PES Analytic Methods in Power Systems (AMPS) Committee is to develop power system optimization test beds where the general applicability and effectiveness of emerging tools in the field of heuristic optimization can be evaluated and compared with one another (as demonstrated in 2014, 2017 and 2018 competitions [1]). Thus, as an initial step, it was decided to organize a special panel at the 2019 IEEE PES General Meeting, which will be preceded by a competition focused on the application of these tools for solving proposed transmission expansion planning (TEP) problems.

The competition aims to perform a comparative assessment of the search capability of different heuristic optimization algorithms. The assessment will be based on statistical tests performed on the results submitted by interested participants. To this end, an encrypted file has been prepared, based on the functionalities of the Matlab and MATPOWER toolboxes, in order to perform an automatic evaluation of the objective function and the constraints of the Optimal Power Flow (OPF), as well as to automatically collect and store the results. In this way, the OPF problems are treated as black box tasks, which should be solved for different test cases based on four selected test networks. Preliminary tests have proven that all optimization test cases are solvable, but some of them constitute very hard-to-solve tasks, in which finding feasible solutions is a key challenge. Therefore, the participants are requested to focus exclusively on the implementation of the particular heuristic optimization algorithm to be used, which could include any special strategy for constraint handling or treatment of discrete/binary optimization variables associated to transformer and compensation devices.

Some encrypted (.p) and .m Matlab files, which are intended for easy adaptation to any heuristic optimization algorithm, are included in the zipped folder named *test_bed_TEP_V01.zip*. This folder also provides complete details (in MATPOWER format) of each test system. Please read carefully the instructions given in every m-file and in the *readme.txt* file, which provide precise indications about Matlab based procedural and implementation aspects.

Final results, which are automatically saved for each optimization test case over 31 independent optimization trials in formatted .mat files contained in a zipped folder named *output_data_implementation_name.zip*, are needed for statistical tests to be performed in the competition, so this folder must be submitted to santiago.torres@ucuenca.edu.ec, J.L.RuedaTorres@tudelft.nl, srriverar@unal.edu.co, by March 1st, 2019 in accordance with the guidelines provided in this technical report. The implementation codes of each algorithm entering the competition must also be submitted along with final results for full consideration in the evaluation. The submitted codes will be in the public domain, and no intellectual property claims thereon be made.

2. DEFINITION OF TEP PROBLEMS

2.1. Overview

A well-known formulation of an optimal TEP problem, i.e. [2], is implemented in the file named *evaluacion_particulas.p*, in such way that different TEP test cases can be performed for three selected test systems with different sizes and structural complexities.

The formulation contained in *evaluacion_particulas.p* has been developed for the calculation of the objective function and the constraints of all TEP tasks, as well as for the automatic collection and storage of the results in .mat files. It uses the functions to model an optimal load flow calculation available in the MATPOWER toolbox [3]-[5], which can be freely downloaded from the web site http://www.pserc.cornell.edu/matpower/. In order to solve the Optimal Power Flow, it is also necessary to download the OPF solver from the website http://www.pserc.cornell.edu/tspopf/. The zipped folder *test_bed_TEP_V01.zip* contains this code along with instructions on how to use it as well as an implementation example with a basic particle swarm optimization (PSO) algorithm. The code (except for the PSO code) is treated as a black box, so it cannot be modified by the participants.

Each participant is encouraged to work exclusively on the particular optimization algorithm to be used**.** The constraint handling (except for the number of ***n*** circuits allowed in each right-of-way) is managed internally using a penalty technique based on artificial generators as explained in [1]. Therefore, participants only have to worry about the number of **n** circuits when implementing their own heuristic algorithm.

The values of f' and its solutions are automatically recorded in a .mat file, which will be used later in the evaluation of the performance of the algorithms. This .mat file is created by the corresponding code in the TEP.m file. It is also possible to set the following parameters in the *TEP.m* file:

- Population value (*Particle_Number*)
- Maximum number of Generations/Iterations (Gen_Max)
- Maximum number of allowed circuits per right-of-way (Xmax)
- Maximum number of trials (nexp)

The Garver 6-bus test system, and slightly modified versions of IEEE 24 and 300-bus test systems are used to evaluate the TEP problems. Based on details given in [1], [2], [6] for system buses and branches, the data of each system has been structured in MATPOWER data format. Branch thermal limits were defined based on reference values given in [1], [2], [6]. Table 1 shows a comparative summary of the characteristics of all test systems, whereas and the following subsections contain the description of the optimization test cases to be performed for each system. In the annex A, you can find the codes structure and the files description.

Table 1: Composition of test systems

Item/System	**Garver 6 bus system**	**IEEE 24 bus system**	**IEEE 118 bus system**
Generators	9	36	182
Loads/Buses	6	24	118
Lines/cables	15	41	186
Candidate right-of-ways	15	41	186
Maximum number of circuits (**Set *Xmax in TEP.m***)	5	5	8

2.2. Garver 6 bus system

2.2.1. Dispatchable generation TEP test case

- **Target:** to minimize the total investment cost of a number of transmission circuits added to the electric system.
- **Constraints:** only the maximum number of circuits set in the TEP.m file (the other constraints are internally handled).
- **Optimization variables:** 15 discrete variables.
- **Scenarios:** 1, corresponding to a long-term loading condition.
- **Considered contingencies (N-1 conditions):** not at the moment.
- **Maximum Number of function evaluations:** 5,000.

2.2.2. Non - Dispatchable generation TEP test case

- **Target:** to minimize the total investment cost of a number of transmission circuits added to the electric system.
- **Constraints:** only the maximum number of circuits set in the TEP.m file (the other constraints are internally handled).
- **Optimization variables:** 15 discrete variables.
- **Scenarios:** 1, corresponding to a long-term loading condition.
- **Considered contingencies (N-1 conditions):** not at the moment.
- **Maximum Number of function evaluations:** 5,000.

2.3. IEEE 24 bus system

2.3.1. Dispatchable generation TEP test case

- **Target:** Minimize the total investment cost of a number of transmission circuits added to the electric system.
- **Constraints:** only maximum number of circuits set in the TEP.m file (the other constraints are internally handled).
- **Optimization variables:** 41 discrete variables
- **Scenarios:** 1, corresponding to a long-term loading condition.
- **Considered contingencies (N-1 conditions):** not at the moment.
- **Maximum Number of function evaluations:** 12,000.

2.4. IEEE 118 bus system

2.4.1. Dispatchable generation TEP test case

- **Target:** to minimize the total investment cost of a number of transmission circuits added to the electric system.
- **Constraints:** only the maximum number of circuits set in the TEP.m file (the other constraints are internally handled).
- **Optimization variables:** 186 discrete variables
- **Scenarios:** 1, corresponding to a long-term loading condition.
- **Considered contingencies (N-1 conditions):** not at the moment.
- **Maximum Number of function evaluations:** 100,000.

3. IMPLEMENTATION ASPECTS

The *TEP.m* file contained in *test_bed_TEP_V01.zip* allows participants to select the TEP test case and scenario to be solved, as well as to call the implementation routine written for their optimization algorithm and to decide whether or not to employ the shared-memory parallel computing functionality of Matlab's Parallel Computing Toolbox.

3.1. Experimental setting

- **Trials/problem:** it is fixed to 31 in *TEP.m*. For initial testing purposes, the participant is allowed to change the value of this variable to a lower one (see line 35 in *TEP.m*). However, please remember that 31 trials are mandatory for performance evaluation in the competition.
- **Stop criterion:** *TEP.m* is configured to terminate automatically an optimization trial upon completion of the maximum number of function evaluations. Lines 185-196 of *PSO.m* provide an example on how to stop a current trial in the participant's implemented algorithm.
- **Initialization:** uniform random initialization within the search space.

- **Encoding:** if the algorithm requires encoding, the encoding scheme should be independent of the specific optimization tasks, and governed by generic factors such as search ranges, dimensionality of the problems, etc.
- **Algorithm tuning:** the participants are allowed to tune their algorithms. The details of the tuning procedure, corresponding dynamic ranges of the algorithm's parameters, and the final parameter values used should be provided to the organizers and thoroughly discussed in the panel as well.

3.2. Results to be submitted

Four (4) .mat files, corresponding to each test system, contained in a zipped folder named *output_data_implementation_name.zip* must be submitted along with the implemented algorithm.

The file *output_data_implementation_name.zip* together with the implementation codes of the algorithm being used, must be submitted to santiago.torres@ucuenca.edu.ec, J.L.RuedaTorres@tudelft.nl, srriverar@unal.edu.co, by March 1st, 2019. Details on the computing system and the programming language used must also be provided. It is discouraged to attempt deliberate manipulation of the .mat files (e.g. replacement of the files corresponding to a given optimization test case with new ones collecting the results of the best 31 trials picked up after performing a myriad of optimization trials).

4. REFERENCES

[1]. 2017 Competition Evaluating the Performance of Modern Heuristic Optimizers on Smart Grid Operation Problems, Test Bed 1: Stochastic OPF based active-reactive power dispatch by Sergio Rivera, Andres Romero, José L. Rueda, Kwang Y. Lee, István Erlich. Available online: http://sites.ieee.org/psace-mho/2017-smart-grid-operation-problems-competition-panel/

[2]. S. P. Torres, C. A. Castro "Expansion planning for smart transmission grids using AC model and shunt compensation" IET Generation, Transmission & Distribution 8 (5), 966-975, 2014

[3]. http://www.pserc.cornell.edu/matpower/

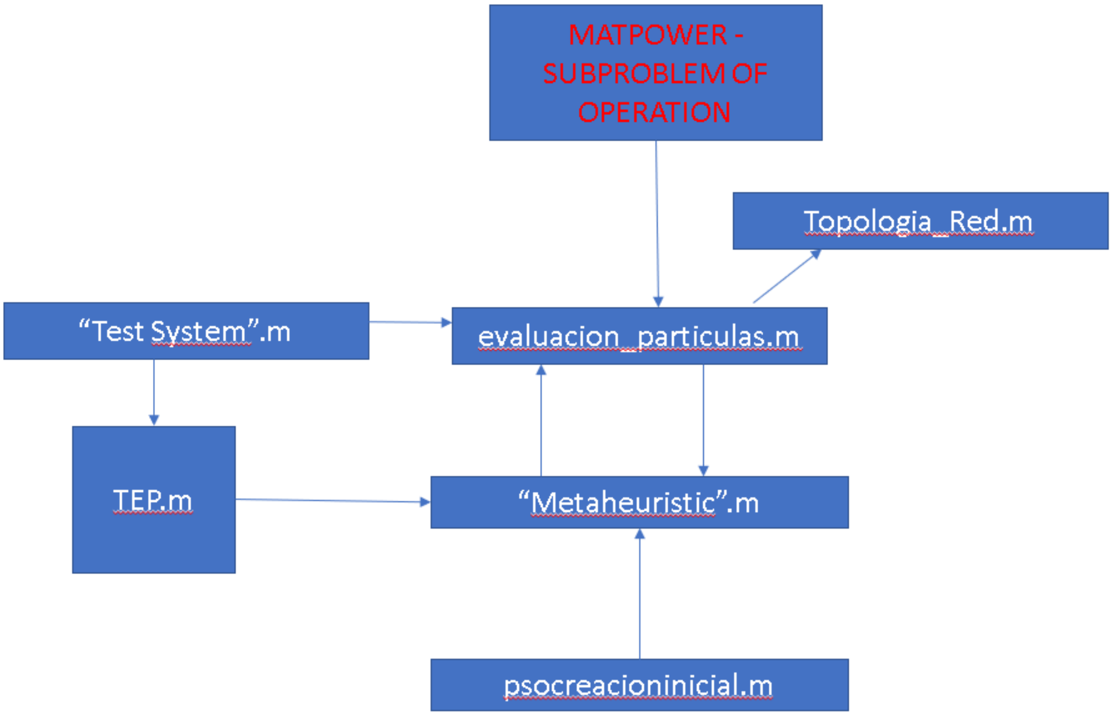
[4]. http://www.pserc.cornell.edu/tspopf/

[5]. R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "MATPOWER: Steady-State Operations, Planning and Analysis Tools for Power Systems Research and Education," IEEE Transactions on Power Systems, vol. 26, no. 1, pp. 12-19, Feb. 2011.

[6]. Rider, M.J.: 'Transmission system expansion planning using DC-AC models and non-linear programming techniques', D.Sc. Thesis in Portuguese, University of Campinas, Sao Paulo, Brazil, 2006

5. ANNEX A

Code Structure



File Descriptions

1.- The file TEP.m is where you configure the parameters of your metahuristic algorithm, e.g. the number of tests you want to perform, the limits of the decision vector X, the metaheuristic function that you want to use.

2.- The PSO.m file is the sample metahuristic file, it must be changed by your algorithm.

4.- The initial psocreacion.m function contains the simple way to initialize the initial population, randomly,

5.- The function evaluacion_particulas.m contains the evaluation of the objective function using the AC model using matpower. It contains the modifications related to the topology, the calculation of the cost of the topology, etc.

6.- The Topologia_Red.m file changes the parameters of the data file according to the transmission topology that evaluates the metaheuristic.

```
%% Test Systems
garver.m (dispatchable generation)
garver_WOrdptch.m (non-dispatchable generation)
case24IEEE_AC.m
case118IEEE_AC20Renew.m
```