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

2017 Competition
Evaluating the Performance of Modern Heuristic Optimizers on Smart Grid Operation Problems

Test bed 1:
Stochastic OPF based active-reactive power dispatch

Problem Definitions and Implementation Guidelines

Sergio Rivera[1], Andres Romero[2], José Rueda[3], Kwang Y. Lee[4], István Erlich[5]

[1]Laboratorio de Metrología (LABE+i) and ElectroMagnetic Compatibility research group (EMC-UN), Department of Electrical Engineering, Universidad Nacional de Colombia, Bogotá, Colombia
[2]Instituto de Energía Eléctrica, Universidad Nacional de San Juan, San Juan, Argentina
[3]Department of Electrical Sustainable Energy, Delft University of Technology, Delft, Netherlands
[4]Department of Electrical and Computer Engineering, Baylor University, Waco, USA
[5]Electrical Power Systems, University Duisburg-Essen, Duisburg, Germany

srriverar@unal.edu.co, aromero@iee.unsj.edu.ar, j.l.ruedatorres@tudelft.nl, Kwang_Y_Lee@baylor.edu, istvan.erlich@uni-due.de

Contents

1. AIM OF THE COMPETITION [1]

The application of heuristic optimization algorithms to solve power system optimization problems is receiving great 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 [0]. Knowing this, the Working Group on Modern Heuristic Optimization under the IEEE PES Power System Analysis, Computing, and Economics Committee organized a special panel in the 2014 IEEE PES General Meeting, which consisted on a competition focusing on application of these tools for solving optimal power flow (OPF) problems [0]. That was the first step towards the development of power system optimization test beds, which are aimed at ascertaining and performing comparative analysis on the general applicability and effectiveness of emerging tools in the field of heuristic optimization [0].

Along this spirit, it has been decided to organize a new competition focused on optimization problems related to smart grid operation. Especial emphasis is put on the consideration of stochastic factors associated to increasing penetration of renewables. This document introduces the first test bed, which concerns with stochastic OPF based active-reactive power dispatch, the implementation for problem evaluation (i.e. calculation of objective function and constraints) is built upon the active-reactive OPF for the IEEE 57-bus system, including the stochastic behavior of Wind, Solar and Small-Hydro Generation.

In a similar way of the 2014 contest, the competition aims to perform a comparative assessment of the search capability of different heuristic optimization algorithms [0], but now considering stochastic behavior of some variables. The assessment will base on statistical tests performed on results submitted by interested participants [0]. For this purpose, an encrypted file has been prepared based on functionalities of Matlab (developed with version R2015a) and MATPOWER toolbox (version used: matpower6.0b1) in order to perform automatic evaluation of active-reactive power dispatch problem trough an objective function and constraints as well as to collect and store automatically the results [0]. The objective function is modified with respect to the 2014 competition, so it is now defined in terms of minimization of expected value of the operational cost as impacted by uncertainty associated to renewable generators (detailed explanation can be found in subsection 2.1.2).

In this way, the problems to be solved are treated as black box tasks, which should be solved for different stochastic scenarios based on probability distributions of wind speed, solar irradiation and river flow over a test network. Therefore, the participants are requested to exclusively work on implementation of the particular heuristic optimization algorithm to be used, which could include any special strategy for constraint handling, strategy for consideration of stochastic variables, or treatment of discrete/binary optimization variables associated to transformer and compensation devices [0].

---

[1] Please note that the structure and content of this section is similar to (it is updated the new considerations for the 2017 competition): 2014 Competition Application of Modern Heuristic Optimization Algorithms for Solving Optimal Power Flow Problems by **István Erlich, Kwang Y. Lee, José L. Rueda, Sebastian Wildenhues**

The encrypted file named *test_bed_OPF.p* along with other exemplary Matlab m-files, which are intended for easy adaptation to any heuristic optimization algorithm, are included in the zipped folder named *test_bed_OPF_2017.zip*. This folder also provides complete details (in MATPOWER format) and updated diagram (in Microsoft Visio format) of the test system in the subfolders named *input_data*, and *Docs*, respectively. Please read carefully the instructions given in every m-file and in *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 ASCII-files. Each participant is kindly requested to put the files corresponding to final results, obtained by using a specific optimizer, into a zipped folder named *output_data_case_implementation_name.zip* (e.g. *output_data_OPF_PSOAlgorithm_Smith*). These files are needed for statistical tests to be performed in the competition, so this folder should be sent to srriverar@unal.edu.co and aromero@iee.unsj.edu.ar by 30th March 2017 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 (i.e. they should also be inside the zipped folder) for full consideration in the evaluation. The submitted codes will be used for further tests, which are intended to crosscheck the submitted results. The submitted codes will be in the public domain and no intellectual property claims should be made.

2. DEFINITION OF OPTIMAL ACTIVE-REACTIVE POWER DISPATCH PROBLEM

2.1. Overview

2.1.1. Competition files structure based on the 2014 contest [0][2]

The formulation of optimal active-reactive power dispatch (OARPD) problem, i.e. [1]-[2], consists in an optimization problem with an objective cost-function with system operation constraints. In this way, in *test_bed_OPF.p* is implemented the calculation of the objective function and the constraints for a set of decision variables. Additionally, *test_bed_OPF.p* has been developed for automatic collection and storage of results in formatted ASCII-files in a similar way of the 2014 contest [0]. It uses the functions for modeling the power flow calculation available in MATPOWER toolbox [3], which can be freely downloaded from http://www.pserc.cornell.edu/matpower/.

The zipped folder *test_bed_OPF_2017.zip* contains this code along with instructions on how to use it as well as an implementation example with basic particle swarm optimization (PSO) algorithm. The code is considered as a black box, so it cannot be modified by participants.

[2] Please note that the structure and content of this subsection is similar to (it is updated the new considerations for the 2017 competition): 2014 Competition Application of Modern Heuristic Optimization Algorithms for Solving Optimal Power Flow Problems by **István Erlich, Kwang Y. Lee, José L. Rueda, Sebastian Wildenhues**

In the zip file, there is a folder called *test_bed_OPF_2017*, in this folder you can find the different files of the competition. Please read the *readme2017Contest.txt* in order to understand the competition files structure.

Each participant is encouraged to work exclusively on the particular optimization algorithm to be used like in the 2014 contest [0]. The use of any type of constraint handling technique is allowed. An exemplary routine for constraint handling is provided in the file *constraint_handling.m*. In the way it is implement in the m-file, this routine does not affect the calculations done in *test_bed_OPF.p*, which calculates internally the set of fitness as a function of the different combinations of decision variables by using (1).

$$fitness\ (decision\ variables)\text{: } objective\ function + \rho \sum_{i=1}^{constraints\ number} \max[0, violation\ of\ constrain_i]^2 \quad (1)$$

where ρ is a penalty factor that is set to a value of 1E+7.

It is clarified that the fitness calculation performed by *test_bed_OPF.p* is exclusively intended for ascertaining fulfillment of constraints in the competition. The values of are automatically recorded at a predefined rate of 100 function evaluations, i.e. power flow calculations, and stored in a formatted ASCII-file, which will be used later in algorithms' performance evaluation.

The *rounding.m* file is an exemplary external function that can be employed for rounding the real numbers used to code discrete/binary optimization variables. You are allowed to modify this file to include your own rounding strategy, but the function syntax, i.e. x_out=rounding(x_in), should be kept, because it is called internally in *test_bed_OPF.p* before every function evaluation. x_in denotes one individual component of the sequence of discrete/binary variables from the vector of optimization variables to be generated using the offspring creation scheme of your optimization algorithm.

The *test_bed_OPF.p* is configured to automatically round the values corresponding to the discrete/binary coded variables to the nearest integer, so this rounding approach will be internally used regardless of whether your algorithm uses a rounding strategy or not. If a rounded variable violates its boundary, it will be automatically fixed in *test_bed_OPF.p* to the corresponding limit.

Please read the instructions given in main_SOPF2017Commented.m to determine the indexes (elements) of the vector of optimization variables defined as discrete/binary variables. Please also refer to main_SOPF2017Commented.m file to gather how to obtain all power system and optimization related information, e.g. location of controllable transformer and compensation devices, problem dimensionality, bounds of optimization variables, steps of discrete variables. In the main_SOPF2017Commented.m file, it is possible to realize where the competitor can update the code in order to prematurely stop running the procedure in terms of independent trials or update the size of the population of your implementation.

The IEEE 57-bus test system is used to evaluate the OARPD problem. Based on details given in [4] for system buses and branches, the data of the system has been structured in

MATPOWER data format. Branch thermal limits were defined based on reference values given in [5]. A summary of the characteristics of the test system is shown in Table 1, whereas description of the optimization test cases to be performed for the system is given in the following subsections.

Please note that a MATPOWER folder must be in the matlab work path since the codes use some MATPOWER functions.

Table 1: Composition of test system

IEEE 57 bus system	
Generators	7
Loads	42
Lines/cables	63
Transformers Stepwise	15
Transformers Fixed tap	2
Shunt compensation Binary On/Off	3

2.1.2. New considerations regarding stochastic behavior of Wind, Solar and Small-Hydro Generation.

Normally, the target in the ORAPD is to minimize the total fuel cost while fulfilling constraints (associated to nodal balance of power, nodal voltages, allowable branch power flows, generator reactive power capability, and maximum active power output of slack generator) for normal (non-contingency) and selected N-1 conditions [0].

In this competition the target is to minimize the total fuel cost of the traditional generators plus the expected cost of an uncertainty cost function for renewable generators. In this way, each renewable generator is considered to be a dispatchable generator; and depending of the available real power, it is considered an underestimated or overestimated condition [6]. These conditions are understood in the following way [6]:

- Underestimated condition

  The scheduled power (Ps_i) in the renewable generator i is lesser than the available real power (Pa_i), and there will be a cost for underestimate given by: $Cu=c_u(Pa_i-Ps_i)$ due to the total available power that is not used in the system (only it is used Ps_i). It would be a kind of power wasted, but in the real life could be considered as a power directed to an energy storage system with a related cost for using the system (c_u).

- Overestimated condition

  The scheduled power (Ps_i) in the renewable generator i is bigger than the available real power (Pa_i), and there will be a cost for overestimate given by: $Co=c_o(Ps_i-Pa_i)$

due to the total available power is not enough to get the power to be scheduled in the system (Ps_i). In this case the network operator must turn on or request more power to another energy source with a related cost (c_o).

The available real power of a renewable generator is not known with certainty. Nevertheless, in some cases, it is possible to know the probability distribution of the primary energy source like the wind speed, solar irradiance or the river flow. In this way, considering the relation between the primary energy source and the available real power, it is possible to get the probability distribution of Pa_i.

In order to obtain the probability distribution of the available power from the known primary energy source probability distribution, it is proposed to develop montecarlo simulations. That is to say, through scenarios of wind speed, solar irradiance or river flow, given by aleatory scenarios from the primary energy source probability distribution. Using the relation between the primary energy source and the injected power in the network, it is possible to get scenarios of the available real power, and through a histogram its probability distribution (see details in the subsections: 2.2.1, 2.2.2, and 2.2.3).

Using the Understimated and Overestimated condition, it is proposed in this competition to calculate through montecarlo simulations an uncertainty cost function given by the different costs for the different available real power scenarios. In this way, it is possible to get the histogram of the uncertainty cost function considering the following steps:

i) Generate a random primary energy source value (following the probability distribution of the wind speed, solar irradiance or the river flow) of scenario j.

ii) Calculate the available real power of scenario j for renewable generator i (Pa_{i_j}) using the relation between the primary energy source and Pa_{i_j} (cf. next section).

iii) Verification of the underestimated ($Ps_i < Pa_{i\_j}$) or overestimated ($Ps_i > Pa_{i\_j}$) condition in scenario j. Ps_i corresponds to the variable decision for renewable generator i.

iv) Calculate the uncertainty cost for scenario j:

$C_{i\_j}=c_u(Pa_{i\_j}-Ps_i)$ if $Ps_i < Pa_{i\_j}$

or

$C_{i\_j}=c_o(Ps_i-Pa_{i\_j})$ if $Ps_i > Pa_{i\_j}$

v) Repeat the steps i) to iv) N (in this competition N is set to 5000 times).

vi) Build the histogram of the uncertainty cost function for the N scenarios.

vii) Calculate the expected cost of the uncertainty cost function for renewable generator i.

2.2. IEEE 57 bus system

The IEEE 57 bus system has 7 generators, in this competition 3 of them are considered renewable generators in the buses 2, 6 and 9; respectively

- **Target:** Minimize the total fuel cost of traditional generators (buses: 1, 3, 8, 12) plus the expected cost of the uncertainty cost function for renewable generators (buses: 2, 6, 9).

- **Constraints:**

There are 3 types of constraints:

i) Power flow constraints

the constraints are associated to nodal balance of power (these are equality constraints)

ii) Constraints penalized in the fitness function

nodal voltages for load buses (42 + 42)
allowable branch power flows (80)
generator reactive power capability (7 +7)
maximum active power output of slack generator (1)

for normal (non-contingency) and selected N-1 conditions, that is to say 179 for non-contingency conditions, and 178 for each N-1 condition.

iii) Minimum and maximum levels of optimization variable

- **Optimization variables:** 31, comprising 13 continuous variables associated to generator active power outputs (6, the slack is not considered here) and generator bus voltage set-points (7), 15 discrete variables associated to stepwise adjustable on-load transformers' tap positions, and 3 binary variables associated to switchable shunt compensation devices.

- **Considered contingencies (N-1 conditions):** outages at branches 8 and 50.

- **Number of function evaluations:** 50000.

- **Cases:** 6, to different cases of stochastic scenarios (The competitor must select the case in the *main_SOPF2017.m* file, lines 69 to 81)

2.2.1. Stochastic OPF for IEEE 57 bus system considering Wind generators (Cases 1 and 4)

For the cases 1 and 4 of this competition, it is considered that the 3 renewable generators are wind generators. It is well known that the wind speed probability distribution follows a Weibull distribution [7]-[8]. Additionally, there is a relation between the wind speed and the available real power. In this way, it is possible to get the probability distribution of the available real power. The file named

WindStochastic.m has the mentioned process in order to get the 5000 montecarlo scenarios for the available real power. An example for the 3 wind generators is showed in the figures 1 to 3.

The case 1 for this competition consider that each participant must use the same montecarlo scenarios. In this way, there is a file called *WindPowerScenarios.mat* with a predetermined montecarlo scenarios. The montecarlo scenarios for case 1 can be represented in the figure 4 (the encrypt file has a control to verify that the scenarios are the same in case 1).

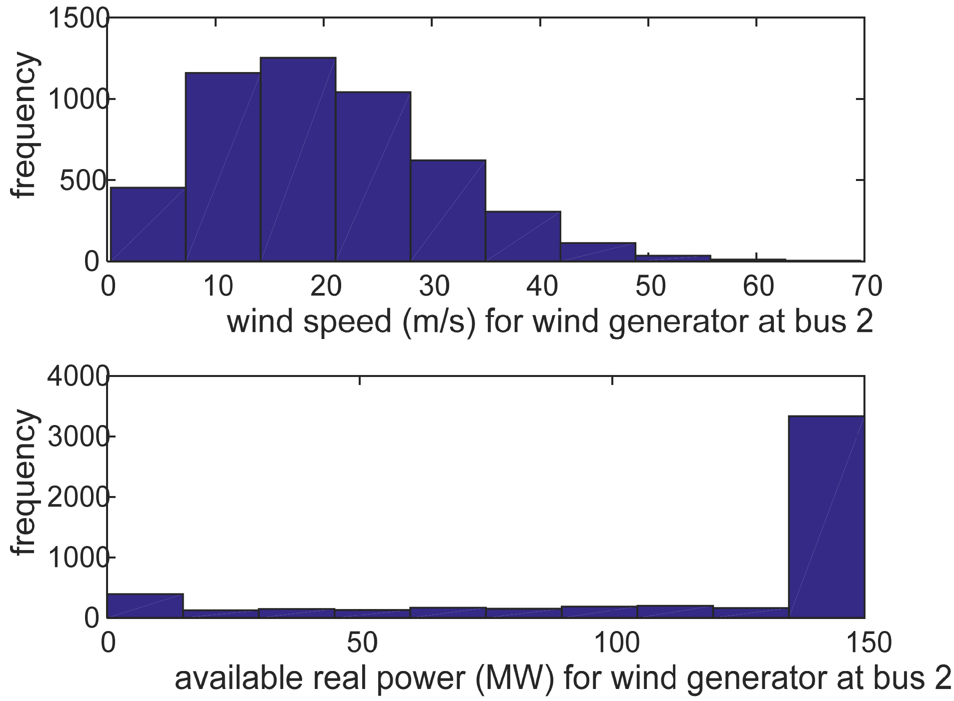

Figure 1. Wind Generator at bus 2.

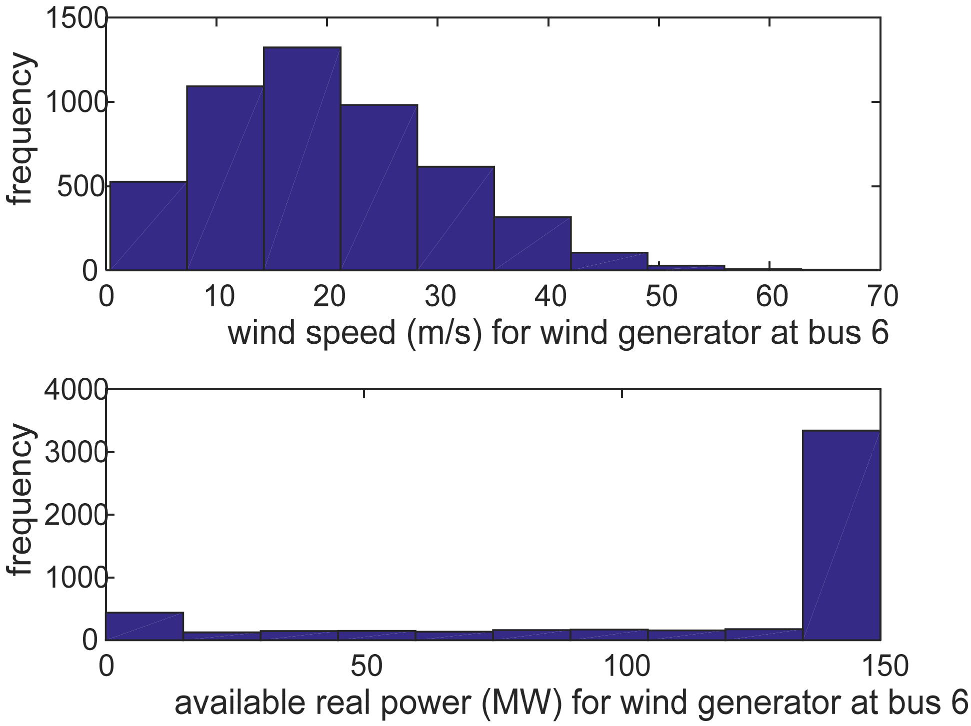

Figure 2. Wind Generator at bus 6.

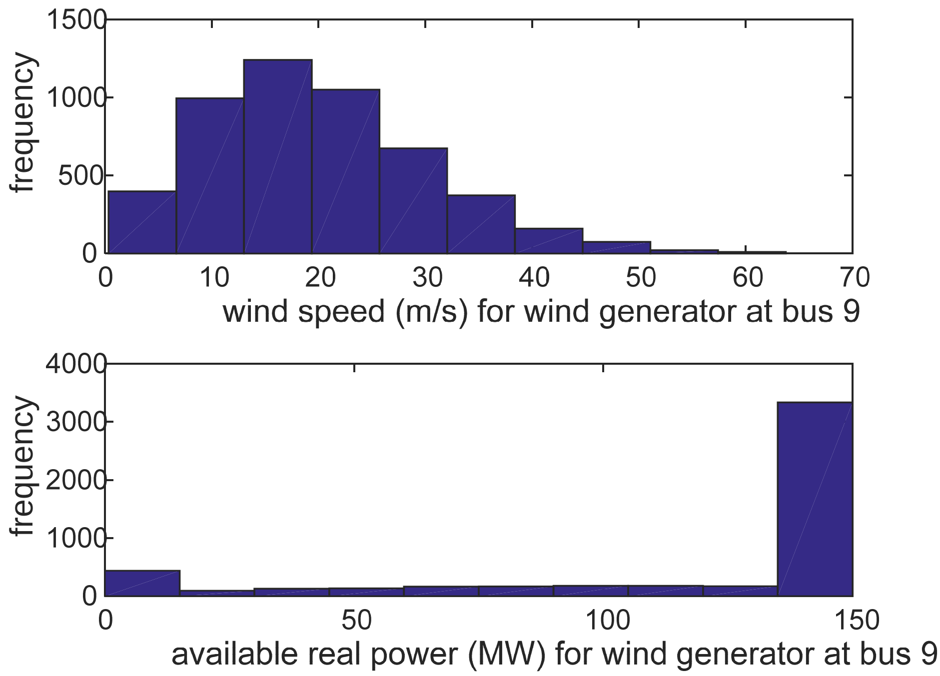

Figure 3. Wind Generator at bus 9.

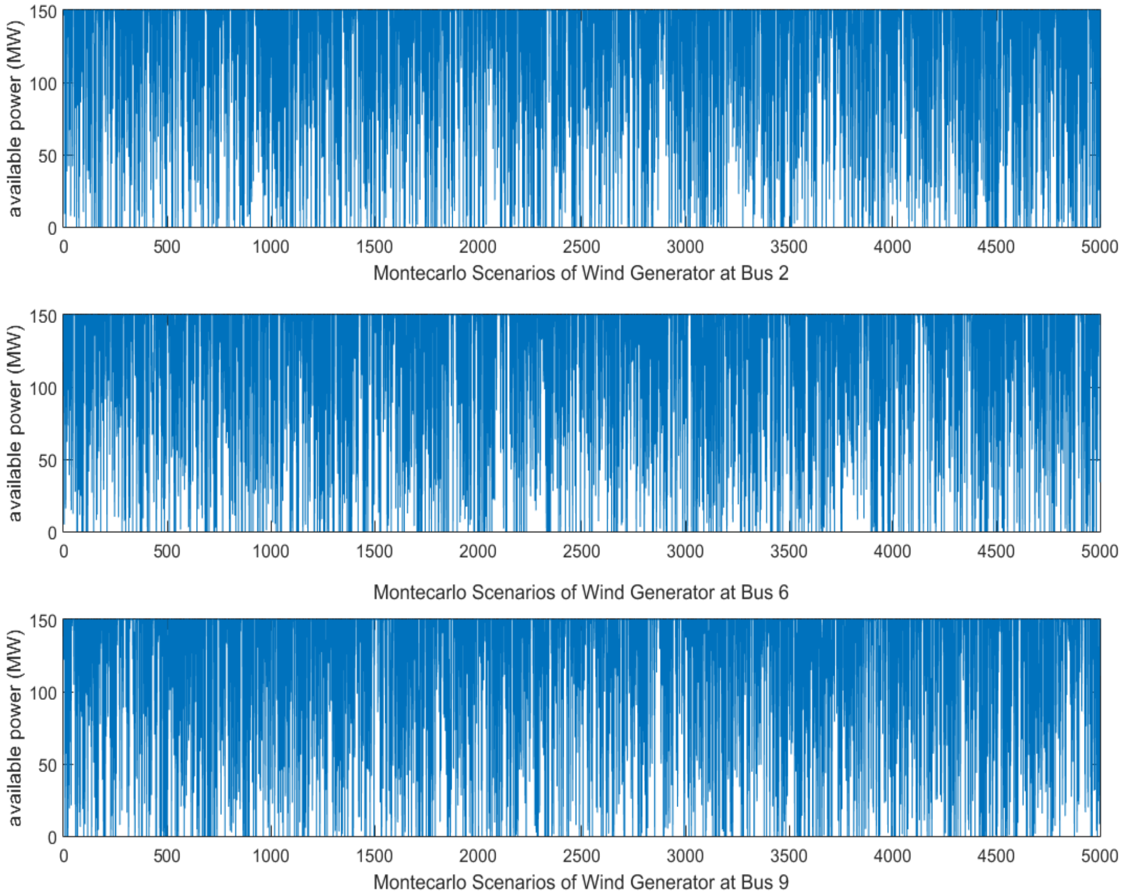

Figure 4. Montecarlo Scenarios of Wind Generators for case 1.

For the case 4, each competitor will have his own montecarlo scenarios generated with the *WindStochastic.m*. In addition to the zip file, it is required that the competitors send the 3 figures printed by the codes in case 4, in order to validate that a Weibull distribution was used for the wind speed.

2.2.2. Stochastic OPF for IEEE 57 bus system considering Wind and Solar generators (Cases 2 and 5)

For the cases 2 and 5 of this competition, it is considered that the 2 renewable generators are wind generators and other solar generator. It is well known that in several parts of the world the solar irradiance probability distribution follows a lognormal distribution [6], [9]. Additionally, there is a relation between the solar irradiance and the available real power. In this way, it is possible to get the probability distribution of the available real power. The file named *SolarWindStochastic.m* has the mentioned process in order to get the 5000 montecarlo scenarios for the available real power. An example is showed in the figures 5 to 7:

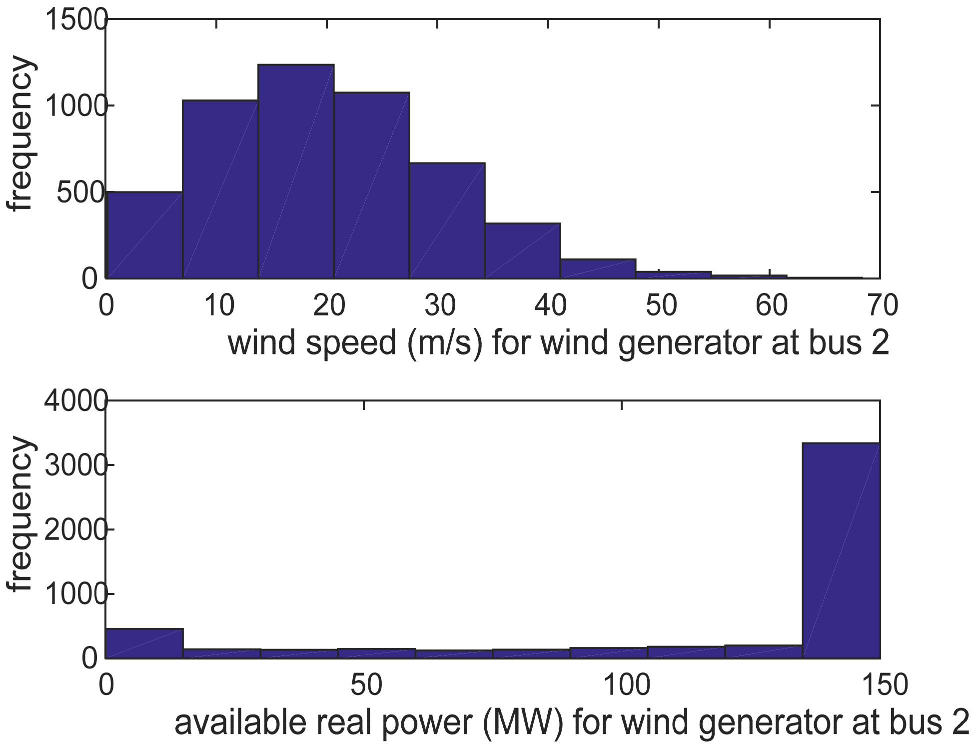

Figure 5. Wind Generator at bus 2.

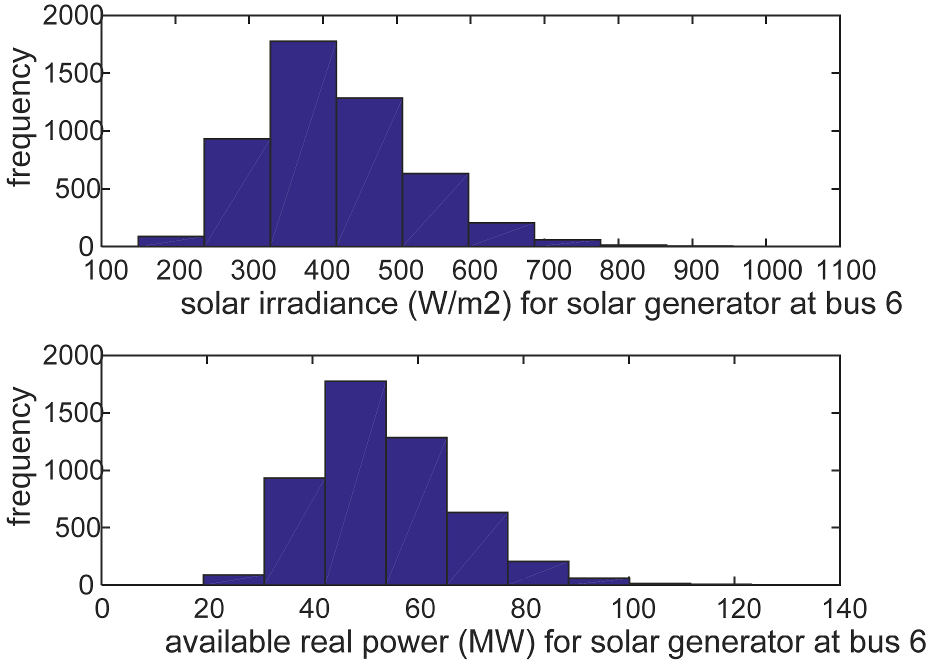

Figure 6. Solar Generator at bus 6.

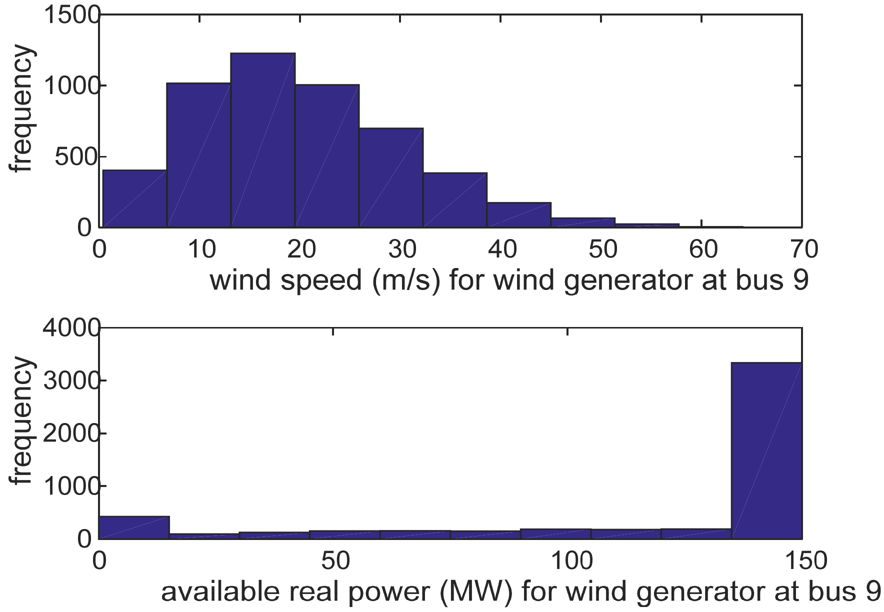

Figure 7. Wind Generator at bus 9.

The case 2 for this competition consider that each participant must use the same montecarlo scenarios. In this way, there is a file called *WindSolarPowerScenarios.mat* with a predetermined montecarlo scenarios. The montecarlo scenarios for case 2 can be represented in the figure 8 (the encrypt file has a control to verify that the scenarios are the same in case 2):

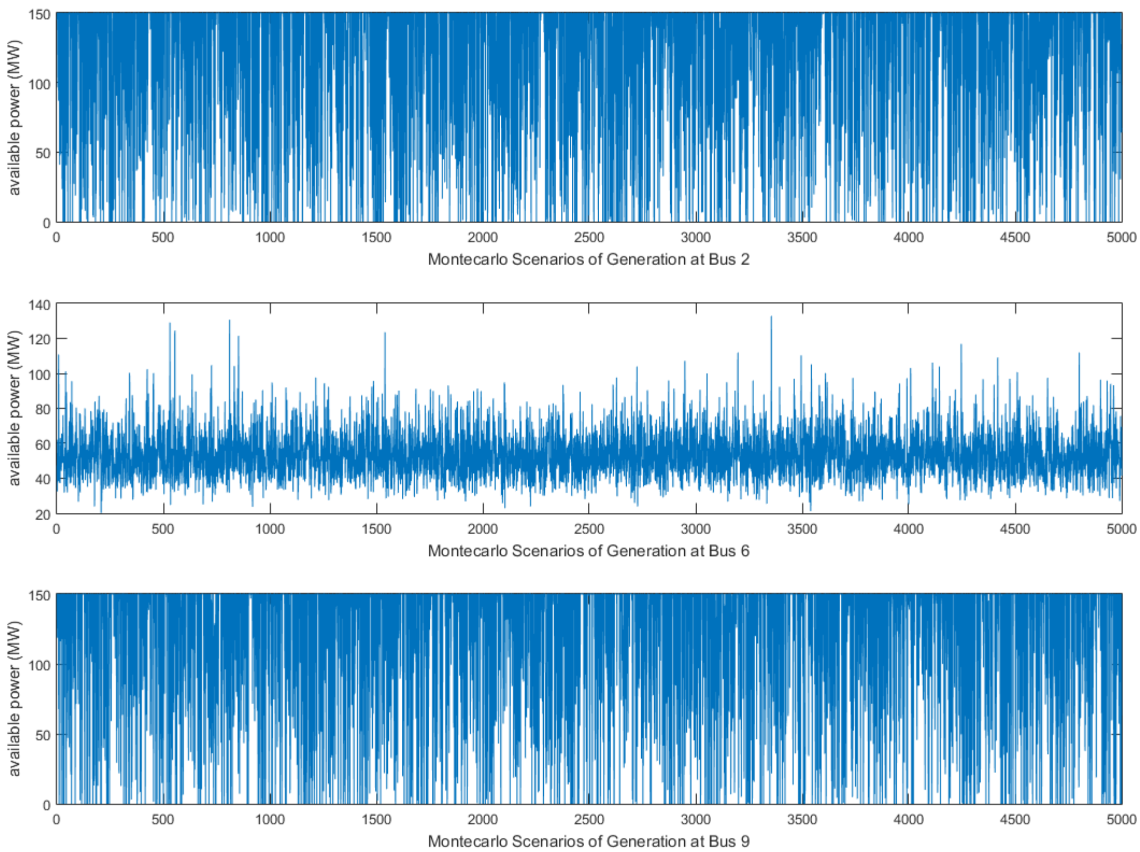

Figure 8. Montecarlo Scenarios of Wind and Solar Generators for case 2.

For the case 5, each competitor will have his own montecarlo scenarios generated with the *SolarWindStochastic.m*. In addition to the zip file, it is required that the competitors send the 3 figures printed by the codes in case 5, in order to validate that a Weibull and lognormal distributions were used for the wind speed and solar irradiance, respectively.

2.2.3. Stochastic OPF for IEEE 57 bus system considering Wind, Solar and Small-Hydro generators (Cases 3 and 6)

For the cases 3 and 6 of this competition, it is considered that at bus 2 there is a wind generator and that at buses 6 and 9 there are two generators, a solar generator and a small-hydro generator. It is well known that the solar irradiance probability distribution follows a lognormal distribution and the river flow follows a gumbel distribution [10]-[11]. Additionally, there is a relation between the solar irradiance and the available real power; and between the river flow and the available real power. In this way, it is possible to get the probability distribution of the available real power. The file named *SolarWindHydroStochastic.m* has the mentioned process in order to get the 5000 montecarlo scenarios for the available real power. An example is showed in figures 9 to 11:

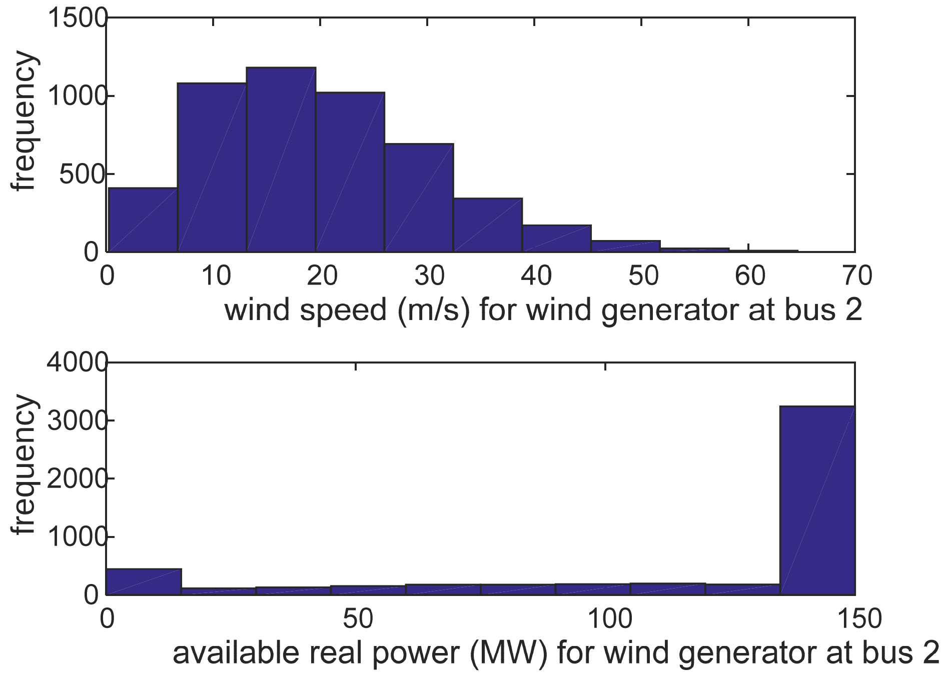

Figure 9. Wind Generator at bus 2.

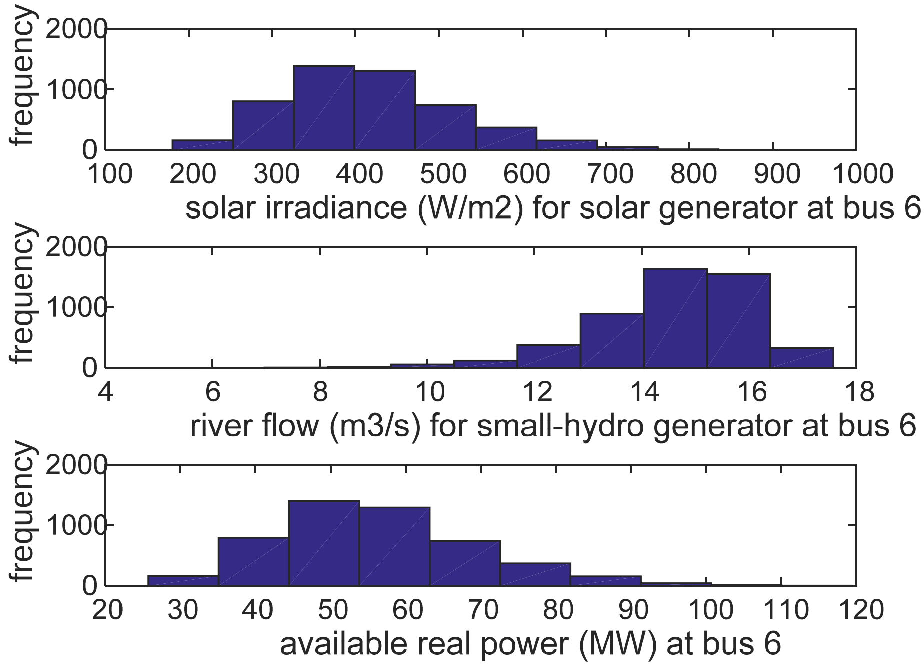

Figure 10. Solar and Small-Hydro Generators at bus 6.

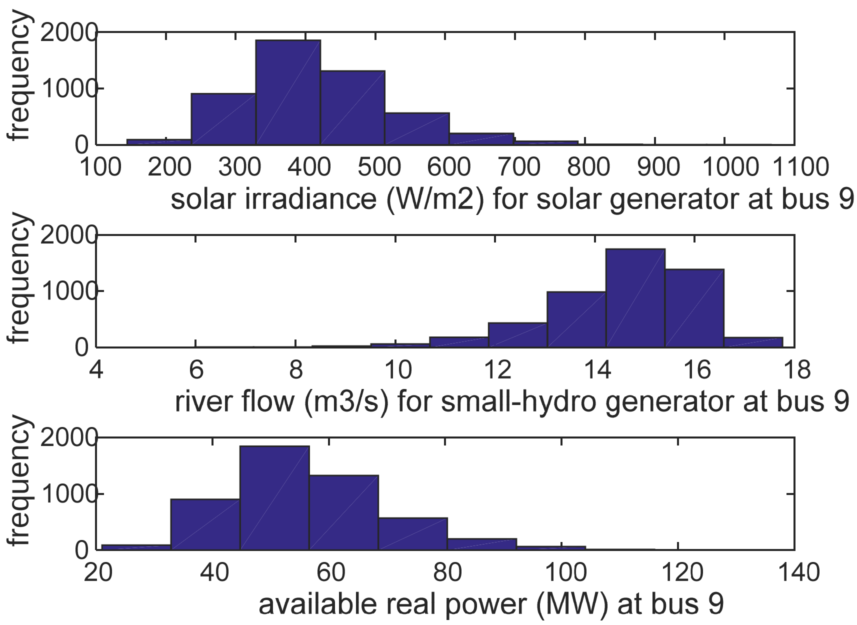

Figure 11. Solar and Small-Hydro Generators at bus 9.

The case 3 for this competition consider that each participant must use the same montecarlo scenarios. In this way, there is a file called *WindSolarHydroPowerScenarios.mat* with a predetermined montecarlo scenarios. The montecarlo scenarios for case 3 can be represented in the figure 12 (the encrypt file has a control to verify that the scenarios are the same in case 3):

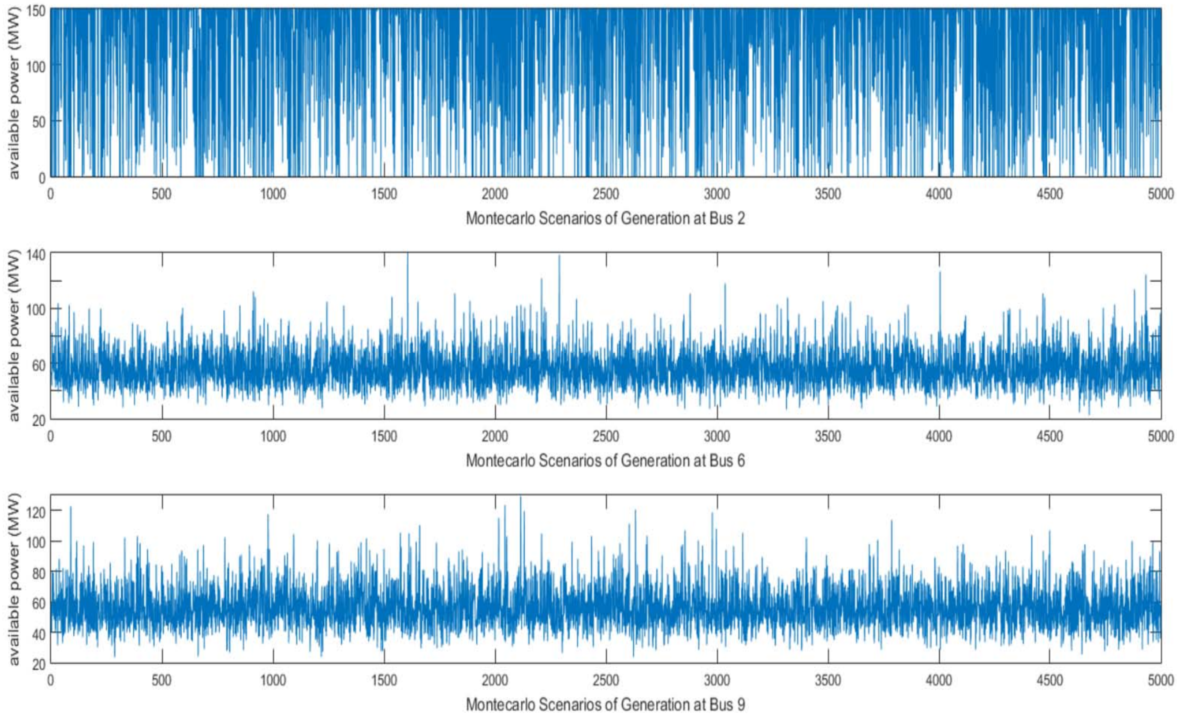

Figure 12. Montecarlo Scenarios of Wind and Solar Generators for case 6.

For the case 6, each competitor will have his own montecarlo scenarios generated with the *SolarWindHydroStochastic.m*. In addition to the zip file, it is required that the competitors send the 3 figures printed by the codes in case 5, in order to validate that a Weibull, lognormal and gumbel distributions were used for the wind speed, solar irradiance and river flow, respectively.

3. IMPLEMENTATION ASPECTS [0] [3]

The *main.m* file contained in *test_bed_OPF_2017.zip* allows selecting the case to be solved, as well as calling the implementation routine written for your optimization algorithm, please note that this year it is not considered to employ shared-memory parallel computing functionality of Matlab's Parallel Computing Toolbox. The file *main_commented_from_2014Contest.m* provides a thorough description of the overall procedure and adaptation of the provided files for your implementation (based in the 2014 competition).

3.1. Experimental setting

- **Trials/problem:** It is fixed to 31 in *test_bed_OPF.p* by using field *proc.n_run*, which is declared global. For initial testing purposes, you are allowed to change

[3] Please note that the structure and content of this section is similar to (it is updated the new considerations for the 2017 competition): 2014 Competition Application of Modern Heuristic Optimization Algorithms for Solving Optimal Power Flow Problems by **István Erlich, Kwang Y. Lee, José L. Rueda, Sebastian Wildenhues**

the value of this variable to a lower one but please remember that 31 trials are mandatory for performance evaluation in the competition.

- **Stop criterion:** *test_bed_OPF.p* is configured to terminate automatically an optimization trial upon completion of the maximum number of function evaluations. It is possible to prematurely stop running a current trial in your implemented algorithm (please see 2014 contest [0]). Nevertheless, it is pointed out that automatic storage of intermediate results in formatted ASCII files will not be performed in this case, so you may have to add some commands to your implementation for recording the progress of objective function, fitness, constraint fulfillment, and optimization variables. Please remember that the maximum number of function evaluations established in the previous section is mandatory for performance evaluation in the competition, and only the ASCII files created automatically by *test_bed_OPF.p* should be submitted for evaluation.

- **Initialization:** uniform random initialization within the search space.

- **Encoding:** If the algorithm requires encoding, then 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. Details of tuning procedure, corresponding dynamic ranges of algorithm's parameters, and final parameter values used should be provided to the organizers and thoroughly discussed in the panel as well.

3.2. Results to be submitted

test_bed_OPF.p performs automatic saving of results in formatted ASCII-files. Please put them into a zipped folder named *output_data_case_ImplementationName.zip* (e.g. output_data_OPF_PSOAlgorithm_Smith).

Please make sure that the folder for each case (1 to 6) contains a total of 5 files. Each of the 5 associated files should automatically be assigned names according to the following convention:

(Name of your implementation)_(Number of buses denoting the system)_2_1_(xyz).txt
where (xyz) stands for different items to be stored:

- objective: recorded objective function convergence data for each optimization trial. The convergence data is recorded after the first and after every 100 function evaluations.
- fitness: recorded fitness convergence data for each optimization trial. The convergence data is recorded after the first and after every 100 function evaluations.
- variables: final best solution achieved by the optimization algorithm in each optimization trial
- constraint_violation: constraint violation vector corresponding to final best solution in each optimization trial

- complexity: computing time associated to each optimization trial

The file *output_data_case_ImplementationName.zip* shall also contain the implementation codes of the algorithm being used must be summited to srriverar@unal.edu.co, aromero@iee.unsj.edu.ar , and j.l.ruedatorres@tudelft.nl by 30th March 2017. Details on the computing system and the programming language used should also be provided. It is discouraged to attempt deliberate manipulation of the ASCII-files, e.g. replacement of the files corresponding to a given optimization test case by new ones collecting the results of the best 31 trials picked up after performing a myriad of optimization trials.

3.3. Evaluation criteria

Although the submitted results will be mainly assessed in terms of the achieved final fitness values, which are automatically saved in ASCII-files by the *test_bed_OPF.p* file, the fulfillment of the established bounds for the optimization variables will be also considered. Based on these results, a ranking index will be established, which accounts for different problem complexities.

4. REFERENCES

[0]. 2014 Competition Application of Modern Heuristic Optimization Algorithms for Solving Optimal Power Flow Problems by István Erlich, Kwang Y. Lee, José L. Rueda, Sebastian Wildenhues. Available online: http://sites.ieee.org/psace-mho/panels-and-competitions-2014-opf-problems/

[1]. Saraswat, and A. Saini, “Optimal reactive power dispatch by an improved real coded genetic algorithm with PCA mutation,” 2nd International Conference on Sustainable Energy and Intelligent Systems, pp. 310-315, July 2011.

[2]. C.-M. Huang, S.-J. Chen, Y.-C. Huang, and H.-T. Yang, "Comparative study of evolutionary computation methods for active-reactive power dispatch," IET Generation, Transmission & Distribution, vol.6, no.7, pp. 636- 645, July 2012.

[3]. 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.

[4]. http://www.ee.washington.edu/research/pstca/. Last accessed on August 15, 2013.

[5]. Y.H. Song, and X.F. Wang, Operation of Market-Oriented Power Systems. Springer Verlag: London, pp. 146, 2003. ISBN 978-1-85233-670-7.

[6]. S. S. Reddy, P. R. Bijwe, and A. R. Abhyankar, "Real-Time Economic Dispatch Considering Renewable Power Generation Variability and Uncertainty Over Scheduling Period," IEEE Systems Journal, vol. 9, pp. 1440-1451, 2015.

[7]. T. P. Chang, "Performance comparison of six numerical methods in estimating Weibull parameters for wind energy application," Applied Energy, vol. 88, pp. 272-282, Jan. 2011.

[8]. T. Arslan, Y. M. Bulut, and A. Altın Yavuz, "Comparative study of numerical methods for determining Weibull parameters for wind energy potential," Renewable and Sustainable Energy Reviews, vol. 40, pp. 820-825, Dec. 2014.

[9]. T. P. Chang, “Investigation on frequency distribution of global radiation using different probability density functions, Int. J. Appl. Sci. Eng, vol. 8, no. 2, 99–107, 2010.

[10].P. Cabus, "River flow prediction through rainfall–runoff modelling with a probability-distributed model (PDM) in Flanders, Belgium," Agricultural Water Management, vol. 95, no. 7, pp. 859-868, 2008.

[11].N. Mujere, "Flood Frequency Analysis Using the Gumbel Distribution," International Journal on Computer Science and Engineering (IJCSE), vol. 3, no. 7, pp. 2774-2778, 2011.