A Genetic Based Service Restoration Algorithm for real-time operated Medium Voltage Distribution Networks using High Performance Computing

Guillermo Bravo[1], Juan Prieto[1], Angel Yunta[2], David Trebolle[2], Stefano Salvini[3] and David Wallom[3]

[1]*INDRA*
{gbravo, jprietov}@indra.es

[2]*Unión Fenosa Distribución*
{ayunta, dtrebolle}@gasnatural.com

[3]*Oxford e-Research Centre*
The University of Oxford
{stef.salvini, david.wallom}@oerc.ox.ac.uk

***Abstract*- This paper presents a new Service Restoration Algorithm for solving the restoration problem of a Medium Voltage Network after a fault. The algorithm is oriented to solve real complex problems dealing with up to thousands of switches. The resulting huge combinatorial problem is solved combining a classical genetic based approach with a probabilistic driven mutation, a smart selection of switches to try, and some heuristic processes to solve constrains when needed. Moreover, performance requirements for real-time operations or a future more sophisticated mutilobjective function are achieved by a High Performance Computing design.**

***Index Terms*- Distribution Network Restoration; Genetic Algorithm; High Performance Computing; Service Restoration;**

I. INTRODUCTION

The recent process of massive deployment of smart meters in Distribution Power Networks, and the associated changes in the ICT[1] infrastructure in secondary (MV/LV[2]) substations and dispatch centres, required to access the smart meters data, is leading progressively to a much higher visibility in real-time of the Medium Voltage Network. This voltage level was traditionally operated off-line and in manual mode, having very little real time information about the actual state of the network. Lacking an accurate network State Estimation, the use of automated functions that are typically applied in higher voltage levels, is very limited to the study of planning scenarios.

Nevertheless, today, as the network is becoming more observable in near to real-time and in terms of the availability of historic data, State Estimation begins to make sense and the results of automated functions over the network are more useful. As a result, most of the algorithms that traditionally have been investigated and developed focused in Transmission Networks, need to be adapted, adjusted or even redesigned to be able to deal with Distribution Networks characteristics: a much larger number of elements and nodes information in Distribution Management Systems and a different and more restricted topology. This effect is especially meaningful for reconfiguration algorithms, which are based on changing the topology of the network, searching for a more stable or efficient state, while fulfilling mandatory constraints in the network. The resultant topology that would lead usually to a more stable and optimum state correspond to a meshed one, which is usually not allowed by design and operation rules in Medium Voltage networks. This feature, along with the huge number of possible combinations, makes the optimum extremely difficult to find, and the process very computationally expensive.

II. SERVICE RESTORATION ALGORITHM

In this paper a special type of reconfiguration algorithm for Medium Voltage Networks, called Service Restoration Algorithm (SRA), is presented. The main purpose of the algorithm is to provide the optimum switching operation list needed to restore as much service as possible after a fault in a primary (HV[3]/MV) substation or in the MV network, leaving the Medium Voltage network in a stable state, and fulfilling mandatory constraints regarding voltage drops, overloads and radial structure of the network.

A. General Approach

In mathematical terms, the problem to solve is a binary integer non linear optimization with the following general formulation:

$$\begin{cases} \min f(x) \\ \text{subject to} \\ G_i(x) = 0 \quad i = 1,\ldots,m_e \\ G_i(x) \le 0 \quad i = m_e+1,\ldots,m \end{cases} \qquad (1)$$

Where:

- 'x' is a binary vector containing the state of each switching element on the network.
- 'f' is the objective function to minimize. Normally, it must achieve a certain compromise between several criteria (total load out of service, number of switching operations, unbalanced load between feeders, etc.)
- 'G_i' are the constraints that must be fulfilled for physical, security or policy reasons (overloads, voltage limits, radial structure, etc.)

[1] ICT: Information and Communications Technologies.
[2] MV: Medium Voltage, LV: Low Voltage.
[3] HV: High Voltage.

- f and/or G_i are non-linear.

Additionally, for this algorithm the following specific characteristics have been fixed:

1) Initial state is such that the original fault is isolated

2) Main objective is to minimize total load out of service (Power not supplied – PNS) with absolute priority, so it is not a multiobjective minimization. Additionally, secondary objectives can be subordinated to the main one (in fact, total number of switching operations has been considered a secondary objective).

3) The constraints taken into account are: voltage in every node inside a given range, thermal ratings within the security standards and radial structure (no meshing network).

4) Non linearity of the problem comes from the fact that computing both voltage and current require a previous non-linear load flow, and from the topology graph that relates switches states and PNS.

It is well known that this type of problems are between the most difficult to solve. They are classified as nondeterministic polynomial time (*NP*-complete) and combine the difficulties of both Integer Programming (IP) and Nonlinear Programming (NLP). They have been widely studied in the last years, as an issue still under discussion [6].

In essence this is a combinatorial problem, which complexity in mathematical terms depends mainly on the number of switching elements (n) that can be operated in the network (2^n combinations), and in performance terms, also on how expensive (in computation time) is objective function evaluation and constraints testing.

In this case, both criteria lead to an extremely complex problem:

1) The algorithm is intended to solve real complex problems, dealing with tens of feeders, hundreds of loads and thousands of switches (only with 1,000 switches there are over 10^{300} combinations)

2) Evaluation of objective function and constraints is expensive, because they need a previous complete load flow computation on the network.

The only deterministic methods for solving discovered up to date are those based on Branch and Bound techniques [3, 5, 6], or its derivatives in most complex cases of non convex functions in objective and constraints. Nevertheless, these methods are usually very expensive in computational terms for real complex problems. For this reason, most of the solving efforts have been oriented to the alternative heuristic methods, which usually offer a better convergence, but in general do not guarantee that the final solution is the global optimum [5, 6, 10].

Consequently, it is clear at this point that some type of heuristic technique should be used to solve the problem. Moreover, previous experience and initial testing with real networks suggested using evolutionary techniques and specifically, a Genetic Algorithm (GA) approach.

A first initial testing procedure was applied using real problems and comparing convergence and performance, in order to select the specific type of GA to apply, and its main parameters. The final selection was a GA with no crossover (only mutation), an expansion-contraction phase on each generation and a random selection of survivals on a certain proportion.

Nevertheless, this initial testing also revealed that simply using GA is not enough to solve in a reasonable time (in order of some minutes). This objective requires a combination of an enhanced GA (taking into account the physics, or electric nature of the problem), and optimized programming techniques. Finally, depending on the results (in performance terms) of previous efforts, and how close to real time the solution is needed, High Performance Computing (HPC) specific technologies may be an option to reduce computation time as needed. This consideration also reinforces the selection of GA, because these algorithms are usually suitable for a deep parallelization compared with deterministic methods or even other heuristic options.

B. Enhanced Genetic Algorithm

The required enhance in GA consist on a way to achieve the same optimal solution with significant less time spent. The strategy followed use two different lines, but both based on using *physical* information of the problem: reduce the available Search space, and enhance the search itself.

Reducing the search space

Reducing the search space consists on avoiding mutations over the majority of the switches, without losing ability to find optimal solution. The key idea for this purpose is that we only need to *close* some switches to restore service: the switching elements to try (the Mutation List Switching Elements – MLSE) are reduced to those that are normally opened (NOSE).

Nevertheless, this dramatic reduction makes the search excessively rigid, because sometimes an interchange between a NOSE and a NCSE (normally closed ones) of the same MV/LV or HV/MV substation is a desirable operation. For this reason, the trying subset of switching elements should be incremented with those NCSE on HV/MV substations or in MV/LV substations with more than two switches, which in fact are switches designed for more flexible operation of the network. In practice, MLSE is restricted at the beginning to NOSE, but also a list of elements NCSE suitable for manoeuvres (NCSE-SM) is calculated with the above criterion. When one of these NCSE-SM is selected for opening in meshing constraint solving heuristic (see below), it is added to MLSE. This is only done for using the complete MLSE in fact, but reducing at most trying list, adding each NCSE-SM element only when it becomes necessary.

Once the MLSE have been defined and they are used in mutation phase of the GA, the problem now is what to do if as a result of closing operations, some constraints are violated. In order to overcome this difficulty, and at the same time, allowing operations over the rest of NCSE not available to mutate, the constraints are tried to solved, instead of directly rejecting the case. For this purpose some heuristic processes have been developed to solve the constraints

violated after a closing operation, by means of opening one or more switches of one or more of these types:

- NCSE that are involved in the constraint violation.
- NOSE that were previously closed, and are involved in the constraint violation.

Any of these opening operations could lead, of course, to leave some loads without service, incrementing the total PNS, but the different alternatives are compared and the one with less PNS is selected.

Enhancing the search itself

In order to enhance the search itself taking advantage of the physics of the problem, the random selection of the switch to mutate has been modified. In this case, not all the switches have the same mutation probability, but it depends on the electric distance (impedance) from the switch to the origin of the fault.

Solving the constraints

The three considered constraints can be violated after closing operations on the mutation phase. In the case of overload and voltage constraints, a selective load shedding is applied. In the case of meshed network constraint, an opening manoeuvre for breaking the cycle is searched combined if necessary with voltage/overload constraint solving. In both cases, the criterion for selection is trying to find the solution with the least PNS.

C. Software Design

The software design is another issue that helps in the final objective of having a useful solver for complex real problems. In this sense the design has been oriented to speed optimization. For this purpose, first, a high optimized compiled language has been selected (C++). Second, memory management has been carefully designed, e.g, using special memory structures for representing each state of switching elements, in terms of closing and opening manoeuvres from an initial reference state. Third, a shared memory scheme has been used for the main algorithm and the load flow solver, thus avoiding of re-building of the whole admittance matrix for each change in switching elements state. And fourth, high optimized libraries has been used for solving basic math operations (boost for graph operations and superLU as linear system solver for load flow [12, 13])

D. HPC Integration

High Performance Computing technologies represent a final option to reduce significantly computing times, taking advantage of the total available computing power on the hardware platform.

Several techniques are available in HPC, some of them focussed on partition of input data and parallel (and independent) computation of packages, and some others more of low level ones, based on parallelization of the processes themselves. In this case a low level parallelization is needed, because the computing performance here is not focused on input data, but in the combinatorial problem that arise managing these data.

For this reason, OpenMP and MPI standards have been adopted in the development of SRA, as the better options for adding HPC capabilities to the original SRA.

Open Multi-Processing (OpenMP)

OpenMP is a portable, scalable model which provides a simple and flexible interface for developing parallel applications through a shared memory application programming interface (API) [15, 16]. It supports multi-platform shared memory multiprocessing programming in C, C++ and Fortran and multiple operating systems such as Linux, Unix, AIX, Solaris, Mac OS X and Microsoft Windows platforms.

With the introduction of multi-core CPU into commodity systems its utilisation is increasing due to this being the best manner in which applications can utilise all of the resources within a single system. OpenMP provides a simple way of developing multithreading applications. In this sense, it is suitable for those algorithms which apply different operations using the same input data or those in which synchronization between the different computations is very expensive (extremely large data transfer). Additionally, has provided a way to take advantage of the full computational power of modern hardware architectures that includes multi-core.

Message Passing Interface (MPI)

Message Passing is a paradigm of application design in which the problem may be broken down into small components each of which after each time-step in calculation need to exchange some result from this calculation with other parallel processes, normally the neighbours in problem space.

MPI has been developed by a consortia of academic, research and industry partners to help to develop a standard and its associated implementations, such that application designers do not have to continually develop these low level communications supporting libraries or components. There are currently two versions of the MPI standard, V1.1 which has a significant number of implementations that are installed on a significant number of commodity HPC cluster systems, and V2.0 which is an extension of V1.1 that has a significantly smaller number of implementations. These standards are available from the MPI Forum5. Significant implementations are MPICH (MPICH consortia) and Open MPI (Open MPI consortia) [14, 17, 18], which are both open source.

Within the HiPerDNO project, Open MPI implementation of MPI is supported through its installation on the cluster [19], using only the V1 features of V2.0 .This is used to support message passing applications when running within pipelines on the cluster environment that have been implemented within the project [19, 20].

Application to SRA

Evolutionary heuristic algorithms of genetic type, which are used in the main process of SRA, are specially suitable for parallelizing using MPI: within the main genetic process

there are some expensive operations (in computational terms) that are applied independently to each individual (the testing solution) of the generation (the set of testing solutions), so synchronization is only needed once in each generation for selection of survivals, and the information to interchange is small (only costs and individuals definitions).

Regarding multithreading and shared memory parallelization (OpenMP), it has not a direct application in main genetic process, because all the individuals of the generation refer to a specific configuration of the same distribution network. Therefore all electrical computations use the same static and dynamic data (except switching states, which are defined precisely by the individuals). Nevertheless, it can be applied if initial data in each generation is replicated, and it is expected that performance lost by replication will be much compensated by additional computational power that become accessible applying OpenMP in a multi-core system. Additionally, it is possible to apply multithreading to other more low level computations used by SRA like linear solvers (superLU) or basic vector and matrix operations (BLAS). In these cases, using of multithreading versions of these libraries enhances automatically the whole performance of SRA.

Both OpenMP and Open MPI approaches have been implemented in the genetic process of SRA, providing to the user the possibility of using or not any of them.

III. Development within HiPerDNO Project

The software has been developed as a part of the research project HiPerDNO [1], which has received funding from EC Seventh Framework Programme 2007-2013. The final result is due to three partners of HiPerDNO consortium: *INDRA* experience on ICT and Electrical Engineering for the general design and software development, DNO *Unión Fenosa Distribución* (UFD) experience on how to solve manually this type of problems for the design of the heuristics and other non standard modifications to the main genetic process, and the *University of Oxford* for the application of HPC technologies to the initial sequential version.

IV. Initial Testing

A first prototype of the Service Restoration Algorithm without HPC (sequential version) was released by February 2011 [2]. During year 2011 and part of 2012 HPC features were added, and extensive testing phase is scheduled for the second half of 2012. Nevertheless continuous testing has been applied during development phase, as a way to select between alternatives and explore new ones. These tests have used two real examples of MV network manually extracted from UFD Corporate Information Systems and simulating a wide range of loads and failures over them. Also MATLAB environment has been used for load simulation, launch of SRA and evaluation of results.

In parallel with the main algorithm development, a full interface with these UFD systems, based on files, has been released, to allow large-scale field trials during year 2012. This interface involves a way to generate files from the corporate system, a launcher of SRA from command-line and a way to retrieve back the results of restoration on the system. The integration will provide a very flexible way of varying the conditions in which the algorithm is tested, both changing the input data (network, load scenario) and the parameters for SRA (voltage limits, overload admissible, HPC use, etc.). It also will facilitate the work of evaluating the quality of the solutions proposed.

Regarding performance tests, a complete HPC environment is required. Oxford e-Research Centre (OeRC) has provided the needed hardware infrastructure, and configured the software environment for accessing of the HiPerDNO consortium members [19].

A. *Performance tests design*

One key factor about SRA performance is that it is based on heuristic evolutionary algorithm of genetic type. This means that on every execution, even using the same inputs, the result could be slightly different, and the time needed can change significantly, depending on chance trying different solutions. For this reason, a unique execution is not enough to characterize performance on solving a specific input, but some statistical average values must be considered for a complete benchmarking of different versions of SRA. In this sense, a special statistical mode has been developed on SRA Launcher that performs the following operations:

1) Calls SRA library n times, where n is a parameter defined in the call by the user.
2) Retrieves the solution of SRA in each call to output files, and records general parameters of execution: time, number of iterations, final objective function achieved (cost and number of maneuvers), number of load flows computed, etc.
3) Computes some statistics over general parameters, in other to retrieve some average indicators about performance and quality of results, and outputs these results to a file.
4) Repeat all the previous operations varying GA parameters that can affect the performance (population size, contraction and expansion factors)

This set of results, regarding four different versions of the algorithm (no HPC, OpenMP, MPI, both OpenMP and MPI) will be used to fill in a two-entry table of execution times which allow detailed comparisons.

B. *Validation tests design*

The main purpose of validation tests is to assure that SRA has an adequate level of fulfilment of the DNO needs.

Table I shows the criteria for testing cases selection. These testing cases result from the combination of some variables. The first choice is the target networks, the 1st one inside a big city area (Madrid, 4 million people) and the 2nd one located in the country. Additionally, some other variables to consider relate to the electricity demand for different characteristic days (normal working day, holiday), seasons and hours, types of failures and conditions for electrical variables.

TABLE I
CRITERIA FOR TEST SCENARIOS DEFINITION

Target networks	- One urban network - One rural network	
Load Scenarios	- Realistic load on an average day - Realistic maximum load on a peak day - Synthetic load: peak load on each feeder - Realistic valley load on a low demand day	
Restoration problems	Failures	- Loss of each individual feeder - Loss of several feeders - Loss of each individual transformer - Loss of the entire substation - Loss of transformers and feeders
	Conditions	- Overload (%) - Maximum voltage drop (%)

In each case the portion of MV network used is determined using a central primary (HV/MV) substation, and the total network that can be considered to reach other HV/MV substations.

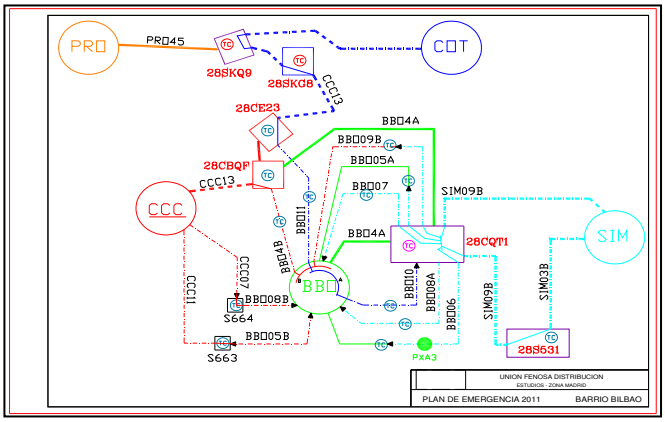

Fig. 1. Emergency plan for BBO substation in Madrid. It shows the choices made to get the supply back. Dashed lines refer to the ones used in the restoration process. It is a summary of the solution to restore the energy after a total supply loss in every feeder. BBO has 2 45/15 kV transformers installed, 25 MVA each and the peak winter load is about 33.2 MVA.

The main criterion for evaluating SRA solution is to check it against the periodic calculations made by engineers over these scenarios, in which are called *emergency plans* (Fig.1) of the DNO. These plans consist on predefined solutions for predefined faults on predefined load scenarios, and are considered as *good enough* solutions. In this sense, results from SRA applied to the same scenarios, should be as least as good as those ones.

C. *Preliminary results*

The results available at this time are still preliminary, because the complete field trials are scheduled for the second half of year 2012. They include results from the development phase, first validation use cases, and raw data from performance test on the HPC cluster.

For validation tests, the two target networks selected are:

- An urban network, located in *Madrid*, and defined by BBO HV/MV substation. Includes 68 feeders from 8 different HV/MV Substations, 1,305 line segments, 1,846 switching elements and 706 MV/LV transformers (loads).
- A mixed rural-urban network, located in *Segovia* province, and defined by CDH HV/MV substation. Includes 20 feeders from 5 different HV/MV substations, 2,182 line-segments, 1,164 switching elements and 454 MV/LV transformers (loads).

Table II shows the complete failure list for BBO substation including the 2 main transformers. For each feeder, the algorithm indicates both the energy source before the failure and the new one after the restoration, coming from a different substation: e.g. , BBO708B was connected to CCC707 to get 1.53 MW from the substation CCC. In the use case analyzed, for the peak demand in 2012, there is a certain power that would still be disconnected (0.46 MW). The solution should be analyzed from a wider perspective looking, maybe, for planning choices instead of optimal operation topologies.

TABLE II
SRA RESULTS SUMMARY FOR BBO SUBSTATION

FEEDER	Source Before Failure	Load (MW)	Source After Failure	Power Restored (MW)	Power not supplied (MW)
BBO708B	TRAFO2	1,53	CCC707	1,53	0
BBO704B	TRAFO2	2,21	CCC713	2,21	0
BBO711	TRAFO1	2,48	CCC713	2,48	0
BBO709B	TRAFO2	3,05	SIM735B	3,05	0
BBO705A	TRAFO1	1,36	SIM709B	1,36	0
BBO708A	TRAFO1	1,21	SIM709B	1,21	0
BBO710	TRAFO2	1,12	SIM735B	1,12	0
BBO706	TRAFO1	2,18	SIM709B	2,18	0
BBO707	TRAFO1	2,43	SIM709B	1,97	0,46
BBO705B	TRAFO2	0,55	CCC711	0,55	0

Checking against emergency plan (Fig. 1), the results have been quite comparable, especially in the worst scenarios studied that refer mainly to the failure in several feeders together, i.e., that go through the same street gallery and become unavailable after some team work.

For performance tests, one of the above cases (CDH) has been selected. Results in terms of average time of execution, depending on how many HPC features are included and the computing power used, are shown in Fig. 2. These results show that fixing the case and genetic parameters, performance evolves almost linearly with the number of nodes at the beginning, and that using OpenMP has similar effect than using MPI if the number of cores is similar to the number of nodes. It seems also that saturation in MPI effect depends on genetic parameters like size of population. A more detailed analysis will be done during field trials.

V. FUTURE WORK AND CONCLUSIONS

Service Restoration Plans and its automation using an algorithm is one of the most useful tools for a DNO. In this sense, different departments in the utility could benefit for SRA results, from Planning to Operation. Today current

policies in architectural design can be compared quite fast for different network structures. Regulation from different administrations (national and regional) can be considered and solutions obtained quickly for big MV networks. In the case of UFD, just to give an idea of the complexity of this work, the total MV network covers line length of 37,149 km., with 47,587 secondary substations and 2,215 MV feeders. Analyzing the best electrical solution for every area can be quite cumbersome if you lack automatic tools.

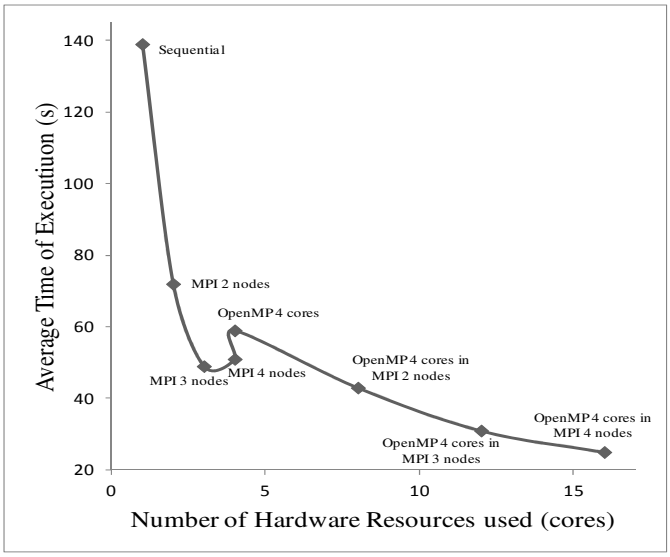

Fig. 2. Average time of execution for SRA versus hardware resources (total number of cores, taking into account how many nodes are used).

Current version of the software is very promising but it is still in a prototype phase. At present, all basic developments planned are available, performance tests have been finished and validation tests are going on. Additionally, some enhancements are currently on development, mainly oriented to add more optimization secondary criteria, in order to reduce to the number of possible solutions. For this purposes, fuzzy modelling is being considered as proposed in [9].

Initial tests on development phase showed a good behaviour in terms of ability to find satisfying solutions. Nevertheless, these tests were limited, so final extensive test phase is needed to carry out a final adjustment, and eventually clarifying the features that need to improve. This phase is planned for the second half of 2012 year, as a part of HiPerDNO project.

Performance tests revealed an almost linear scalability relationship between hardware resources used by HPC and final reduction on computing time, at least on the reasonable range of cores/nodes available on a computer cluster. This feature provides a way to achieve the needed computing time for real-time applications, just adding the enough hardware resources. Moreover, provide a way for solving more complex problems in the future, or even adding new capacities to SRA (e.g. compute optimal switching operation sequence) without severely compromising computing times. Anyway, even computing times for sequential version are small enough for network planning applications, and they can be reduced in a factor of at least two simply using OpenMP with regular multi-core computing systems available at present, without any new investment need.

A full integration with UFD corporate systems is planned after ending extensive test phase. We expect that practical use of the algorithm will be the better way of showing in which way it could be evolved to enhance the real-time and planning operations of a DNO.

ACKNOWLEDGEMENTS

The authors wish to thank the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement number 248135 for their support of the HiPerDNO research project.

REFERENCES

[1] http//:www.hiperdno.eu, *HiPerDNO: High Performance Computing Technologies for Smart Distribution Network Operation*, EU FP7/2007-2013 ICT Energy Funded Research Project.

[2] G. Bravo and J. Prieto, *Functional Design, Specification and Prototype Development of an Algorithm for Distribution Network Service Restoration*, INDRA, HiPerDNO report D2.3.1 , February 2011.

[3] A. Ravindran, K. M. Ragsdell and G. V. Reklaitism, *Engineering optimization, Methods and Applications,* John Wiley & Sons, Inc. 2006.

[4] J. Zhu, *Optimization of Power System Operation,* IEEE Press Series on Power Engineering, John Wiley & Sons, Inc. 2009.

[5] K. Y. Lee and M. A. El-Sharkawi, *Modern Heuristic Optimization Technique. Theory and Applications to Power Systems,* IEEE Press Series on Power Engineering, John Wiley & Sons, Inc. 2008.

[6] J. Branke, K. Deb, K. Miettinen, R. Słowiski, *Multiobjective Optimization. Interactive and Evolutionary Approaches,* Springer, 2008.

[7] M. Ehrgott, *Multicriteria Optimization*, Springer, 2005.

[8] W. Chen, M. Tsai, H. Kuo. *Distribution System Restoration Using the Hybrid Fuzzy-Grey Method,* IEEE Transactions on Power Systems, Vol.20, No.1, February 2005.

[9] A. Augugliaro, L. Dusonchet, E. R. Sanseverino, *Multiobjective service restoration in distribution networks using an evolutionary approach and fuzzy sets,* Electrical Power and Energy Systems 22 (2000) 103–110.

[10] D. Shin, J. Kima, T. Kimb, J. Choob, C. Singh, *Optimal service restoration and reconfiguration of network using Genetic-Tabu algorithm,* Electric Power Systems Research 71 (2004) 145–152.

[11] G. Taylor, D. Wallon, A. Yunta, C. Axon, *Recent developments towards novel high performance computing and communications solutions for smart distribution network operation.* 2011. IEEE ISGT

[12] J. W. Demmel, S. C. Eisenstat , J. R. Gilbert, X. S. Li, J. W. H. Liu, *A supernodal approach to sparse partial pivoting.* SIAM J., Matrix Analysis and Applications, 1999, Vol. 20, No. 3, 720-755.

[13] J. G. Siek, L. Lee, A. Lumsdaine, *The Boost Graph Library. User Guide and Reference Manual.* C++ in Depth Series. Addison-Wesley Professional, 2002.

[14] OpenMPI, *Open Message Passing Interface*, http://www.open-mpi.org/

[15] OpenMP, *Open Multi Processing,* http://openmp.org/

[16] OpenMP, http://en.wikipedia.org/wiki/OpenMP

[17] W. Gropp, E. Lusk and A. Skjellum, *Using MPI: Portable Parallel Programming with the Message-Passing Interface*, MIT Press, 1999.

[18] M. J. Quinn, *Parallel Programming in C with MPI and OpenMP*, McGraw-Hill, 2003.

[19] S. Salvini, P. Lopatka, and D. Wallom, *A hardware and software computational platform for the HiPerDNO project*. In *Proceedings of the first international workshop on High performance computing, networking and analytics for the power grid* (HiPCNA-PG '11). ACM, New York, NY, USA, 75-82. DOI=10.1145/2096123.2096139.

[20] *Working paper on proto-standards for data acquisition for processing with an HPC toolset*, Brunel University, University of Oxford, IBM, Elecktro Gorenska, UK Power Networks, INDRA and Unión Fenosa Distribución. HiPerDNO report D3.2.3, January 2012.