An Efficient Simulator for Fault Detection and Recovery in Smart Grids *FDIRSY*

Syrine Ben Meskina[1], Narjes Doggaz[2] and Mohamed Khalgui[3]
[1]*LISI, ISI, University Tunis-El Manar, Tunis, Tunisia*
[2]*LIPAH, FST, University Tunis-El Manar, Tunis, Tunisia*
[3]*LISI, INSAT, University of Carthage, Tunis, Tunisia*

Keywords: Smart Grid, Fault Categorization, Dominant Faults, Equivalent Faults, Local Resolution, Non-local Resolution, Multi-agent System.

Abstract: This research paper deals with failures and faults in power smart grids. We propose an original multi-agent approach for power system recovery based on fault classification. For that, we propose the classification of faults as dominant or equivalent ones. This classification has the advantage of optimizing the task of power system recovery. To test and validate our approach, we develop a simulator, named FDIRSY (Fault Detection, Isolation and Recovery SYstem). The experimental study showed that our approach ensures the search for the best solution from the existing ones thanks to the use of mobile agents. These agents have the advantage of evaluating all the existing alternatives while reducing the communication cost (in terms of exchanged messages). We demonstrate that our approach is gainful in terms of required times, actions to be performed as well as the faults to be resolved thanks to the proposed fault classification.

1 INTRODUCTION

The new generation of power networks, "smart grid", makes power networks intelligent through the main following characteristics; multi-service communication, reliability, security and safety allowing a real-time supervision. It should integrate the actions of all types of users and consumers increasing the importance of low voltage network automation. (Fang et al., 2012) discusses the several actors and factors contributing on the evolution of power grids in their survey. Many types of problems may occur on power grids engendering other faults by propagation. There are many related works dealing with power recovery systems. But the existing methods do not handle the consequent failures and do not investigate the relation between them. Each one of the detected failures is handled separately.

In this paper, we are interested in developing an original software system detecting the faults in power networks, identifying, localizing and investigating the occurred faults as well as the consequent ones. This new simulator is based on the multi-agent approach for power system recovery proposed in (Ben Meskina et al., 2014). The decentralization of the control (one controller agent per sub-grid) restricts the number of the deployed agents in order to avoid the expensive inter-agent communication. The use of mobile agents in power grid present a novelty and allows an accelerated investigation of all the connected neighbor sub-grids to provide efficient recovery. The number of faults to be resolved is reduced thanks to the proposed fault categorization. The use of extensible data bases updated at run-time is advantageous in order to re-use solutions yet found. In this paper, Section 2 summarizes the most important works related to smart grid recovery in the literature. In Section 3, we expose the problems which are not considered in the literature and present the characteristics and architecture of the developed simulator FDIRSY. Then, in Section 4 we detail its functioning and give the principal algorithms of the several agents deployed in the multi agent system(MAS). The last section illustrates the gain, using our simulator, in terms of required time and communication.

2 STATE OF THE ART

There is a big amount of research works, in the literature, relative to several topics about smart power grids. An important number of them work on failure

Ben Meskina S., Doggaz N. and Khalgui M..
An Efficient Simulator for Fault Detection and Recovery in Smart Grids - FDIRSY.
DOI: 10.5220/0005245001320139
In *Proceedings of the 5th International Conference on Pervasive and Embedded Computing and Communication Systems* (PECCS-2015), pages 132-139
ISBN: 978-989-758-084-0

detection and localization, self-healing and system recovery... These related studies investigate these problems and propose several approaches based on different concepts such as; Multi-Agent Systems (MAS), Petri-Nets, smart micro-grids using Distributed Energy Resources (DER), electric devices like Smart Storage Devices (SSD)... (Chertkov et al., 2011) develops an approach to efficiently identify the most probable failure modes in static load distribution for a given power network. (Calderaro et al., 2011) proposes a method using Petri Nets in order to detect and identify failures in transmission power grids. (Russell and Benner, 2010) presents some examples of the types of incipient failures that can be detected from substation electrical waveforms. (Oudalova and Fidigattib, 2009) uses the concept of micro-grids using renewable energy, SSD and load controllers in order to reduce the transmission loads and to resolve failures. (McArthur et al., 2007a) and (McArthur et al., 2007b) investigate and study the higher value-added by MAS to power industry (using open MAS architectures and distributed system platforms). (Jiang et al., 2014b) propose a hierarchical MAS for the self-healing based on IEC 61499/61850. (Jiang et al., 2014a) propose an hierarchical multi-agent architecture for automatic restoration in power grids. They associate an intelligent agent to each electrical station in order to control its functioning. (Massoud and Wollenberg, 2005) presents a modern infrastructure for power grids improving their efficiency, reliability, and safety and making them intelligent by, simply, integrating a software module in some electric components. The goal is to decentralize the control without waiting the response of the central protection system in the case of power network failures.

After these short observations on the literature, we count a multitude of research works working on failure detection and power system recovery. In fact, the majority of them handle only one type of faults. We remark, also, that the related works do not investigate the relation between the detected fault and the consequent ones to resolve both of them. It is important to establish a relation between the detected faults and the engendered ones in order to define an optimizing strategy solving the maximum of the occurred faults or, even, all of them. In the other hand, we note that the majority of the proposed approaches, in the literature, are based on MAS deploying a big number of agents ((Ramchurn et al., 2011) assigns one agent to each consumer). In fact, this makes the process of inter-agent communication very expensive in terms of required time and, thus, the procedure of search for solutions performs too slowly. (Rahman et al., 2007) proposes an efficient approach for power systems self-healing based on specialized micro-grids. They use 7 agent types making the communication process costly. Another disadvantage of the proposed MAS in the literature lies in the centralized control. (Vyatkin et al., 2010) deploys only a unique one master agent Facilitator Agent (FAG) and multiple Bus Agents (BAGs) leading to a centralized control power system and to a big number of BAGs. The investigation of the history of the encountered and solved problems in power grids is important to avoid the resolution of problems solved beforehand. (Pipattanasomporn et al., 2009) used an inextensible fact bases.

3 SMART GRID SIMULATOR

In this section, we begin at first by exposing the reasons motivating the development of FDIRSY. Second, we detail our multi-agent approach for fault detection and recovery. Finally, we present, FDIRSY, our developed simulator.

3.1 Problems and Motivations

The robustness of electrical grids lies on their ability of managing and facing all eventual submerging problems. We model a problem in a power network by a fault or a fault set. In fact, a fault occurs because of violating one or more constraints knowing that such fault may engender other consequent ones. We proposed, in (Ben Meskina et al., 2014), a set of operating conditions that must be respected in smart power grids. In order to optimize the required time for power system recovery, we look for optimizing the cost of the resolution procedure while trying to solve the maximum of the occurred faults. For that, we proposed new definitions for dominant and equivalent faults in (Ben Meskina et al., 2014). Dominant faults correspond to faults engendering other ones in the connected components (belonging to the voltage level below or the same one with smaller rank). The rank presents the order of a component apparition in a power line. Equivalent faults correspond to faults occurring on connected components belonging to the same voltage level and having the same rank. These new relations facilitate the failure recovery and allow the control and the reduction of the fault recovery time.

In order to illustrate our fault categorization, we consider, in Figure 1, a power sub-grid. Let us assume that the emergency Medium Voltage Line $eMVL1$ is activated and that the Medium Voltage Line $MVL1$ is deactivated. Let us suppose, in addition, that there is an instability voltage problem observed on $MVL2$.

We denote by F_1 the resulting fault occurred on the Low Voltage Transformer $LVT1$ which is propagated to the connected devices. We denote, respectively, by F_2, F_3, F_4 and F_5 the faults, consequently, occurred on the Low Consumers $LC1$, $LC5$, $LC8$, and on the Medium Consumer $MC1$. Based on the new proposed definitions, we categorize F_2, F_3 and F_4 as faults dominated by F_1 and both of F_1 and F_5 as equivalent ones. The resolution of the dominant fault resolves all the problem (including the dominated ones). When a dominant fault can not be resolved, we proceed to the resolution of the dominated faults after categorizing them. On the other hand, the resolution of only one of the equivalent faults can resolve all the problem. The advantage is, particularly, observed in the case of multiple occurring faults, as we are focussed on resolving only one or, at most, a subset of them. Thus, in this example, the resolution of either $F1$ or $F5$ solves all the problem. This strategy facilitates, accelerates and guides pertinently the procedure of search for solution.

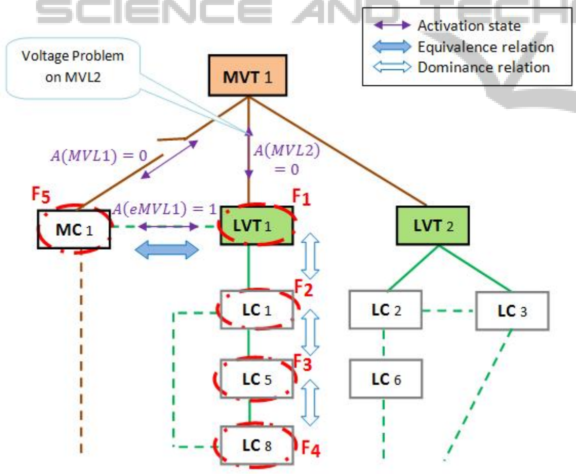

Figure 1: Fault F_1 engendering other faults (F_2, F_3, F_4 and F_5) on the connected components

3.2 The Proposed Architecture

To achieve the development of our system, we use the multi-agent paradigm based on distributed architecture decentralizing the control hence the decomposition of the overall electric network into sub-nets. This architecture is, essentially, composed of three types of software agents which are Reconfiguration Agent, Mobile Agent and Data Base Agent. These agents interact and collaborate together in order to maintain the stability and the effective functioning of the power grid. Our system detects and classifies the occurred faults to facilitate the system recovery and looks for finding the best solution as we use a solution updated at run-time contrary to the related works. The UML activities diagram, illustrated by Figure 2, describes the behavior of our simulator. This schema indicates the used software entities composing our system and summarizes the main considered functionalities.

- **Reconfiguration Agent (RAgent)**

A Reconfiguration Agent is responsible of the detection and the recovery of the faults occurred in the sub-grid under its scope. We propose to associate only one RAgent to each power smart sub-grid. When it detects an anomaly due to a violated constraint, it isolates first the component or line responsible of the anomaly. Then, it searches locally for a solution. When no local solution is found, it requests the Data Base Agent about a solution already found for the encountered problem. If no stored solution, the Reconfiguration Agent searches for a cooperative solution from the neighbor sub-grids. For that, it creates a Mobile Agent(s) to obtain information about the components belonging to the connected sub-grids.

- **Mobile Agent (MAgent)**

A Mobile Agent is a software entity moving dynamically, through electric lines, to visit the components belonging to the neighbor smart sub-grids. It is created and destructed by an RAgent. Its task consists on collecting and communicating useful information for a given search for fault recovery. When the visited component has more than one outgoing line (connected to more than one components), the MAgent is duplicated in order to analyze all the existing alternatives or paths. The duplicated agents are called clones and takes only one line (path) at a time. All the clones operate simultaneously in order to decrease the required time. Each component should be visited at most once by the same Mobile Agent to ensure that the resolution terminates.

- **Data Base Agent (DBAgent)**

The DBAgent is responsible of the management and the storage of all the information relative to the whole power grid structure. It contains, also, the history of all the occurred faults and all information relative to the problems (like problem type, failed components, CPU...) as well as the corresponding solutions found by MAgents. It may be requested by both Reconfiguration and Mobile Agents about information relative to the power system. It should update its knowledge for each new solution found by Mobile Agents. To avoid searching solutions for problems already resolved, the Reconfiguration Agent may request the Data Base Agent when it encounters a problem for which it did not found a local solution.

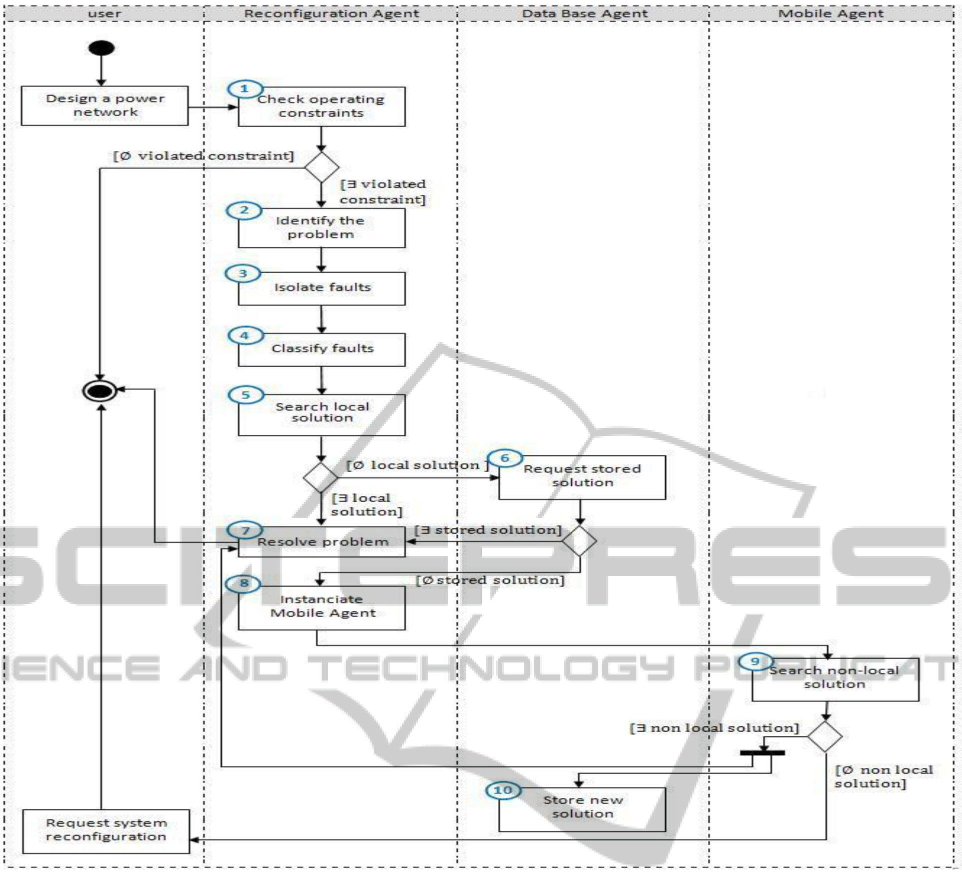

Figure 2: Activities diagram relative to the proposed MAS.

3.3 FDIRSY

To test, validate and evaluate our approach, we have developed in Java, FDIRSY, a *Fault Detection, Isolation and Recovery SYstem*. It provides the following services: *(i)* design and simulation of smart grids composed of electrical components (generators, transformers, consumers) and lines (principal and emergency ones) belonging to the high, medium and low voltage levels, *(ii)* checking operating conditions to detect, localize and identify problems, eventually, occurred in the simulated smart grid, *(iii)* isolation of the detected problem to avoid propagation, *(iv)* classification of the occurred faults relative to the encountered problem and *(v)* searching for solution to recover the failed power grid.

4 IMPLEMENTATION

In this section, we describe the functioning of the proposed simulator *FDIRSY*. We begin by detailing and giving some algorithms relative to the tasks ensured by RAgents followed by those ensured by MAgents.

4.1 RAgent Implementation

An RAgent is responsible of supervising the proper functioning of the sub-grid under its scope. Let RA_i be the RAgent supervising the i^{th} smart sub-grid denoted by SSG_i. RA_i ensures the following tasks:

- **Check Operating Constraints**

FDIRSY allows RA_i to investigate the healthy operation of SSG_i. In fact, RA_i checks the operating conditions over SSG_i; the activation, stability and flowing load constraints. Algorithm 1 provides the set of the violated constraints ($VC(SSG_i)$), below, the description of the several used functions:

- ***A(X)***: returns the activation state of an electrical component or line X as a boolean value (true if activated and false if deactivated),

- ***Freq(C)***: returns the frequency of an electrical component C,

- ***Volt(L)***: returns the voltage value of an electrical line L,

- ***ReqL(C)*** and ***ReceivL(C)***: returns, respectively, the required and received loads by a consumer C.

Algorithm 1: Checking Operating Constraints.

Require: SSG_i
Ensure: $VC(SSG_i)$
$VC(SSG_i) \leftarrow \emptyset$
if ($\exists C \in SSG_i \backslash A(C) = false$ **or** $Freq(C) \neq DefaultFreq$ **or** C does not respect its capacities **or** $ReqL(C) < ReceivL(C)$) **then**
add $violated\ constraint(C)$ **to** $VC(SSG_i)$
if ($\exists pL \in SSG_i \backslash A(L) = false$ **or** $Volt(L) \neq DefaultVolt$ **or** L does not respect its capacities) **then**
add $violated\ constraint(pL)$ **to** $VC(SSG_i)$

- **Identify Problem**

When there is, even at least, one violated constraint ($VC(SSG_i) \neq \emptyset$), RA_i proceeds, in 2, to identify the encountered problem $ep(VC(SSG_i))$. In fact, it is an important step to localize the failed components ($fcL(SSG_i)$) and lines ($flL(SSG_i)$) in order to guide the recovery procedure since each problem type is resolved differently. Moreover, it is a useful step to update the data base. We note that each failed component corresponds to a fault. It should be noted that several faults can occur simultaneously following to more than one violated constraint (one problem).

Algorithm 2: Problem Identification.

Require: $VC(SSG_i)$
Ensure: $fcL(SSG_i)$, $flL(SSG_i)$, $ep(VC(SSG_i))$
$fcL(SSG_i) \leftarrow \emptyset$, $flL(SSG_i) \leftarrow \emptyset$
for each $vc \in VC(SSG_i)$
switch (constraint type of vc)
case activation**:**
$ep(VC(SSG_i)) \leftarrow$ deactivation problem
case stability**:**
$ep(VC(SSG_i)) \leftarrow$ instability problem
case flowing load**:**
$ep(VC(SSG_i)) \leftarrow$ problem on flowing loads
end switch
update($fcL(SSG_i)$, $ep(VC(SSG_i))$)
update($flL(SSG_i)$, $ep(VC(SSG_i))$)
end for

- **Isolate Failed Components / Lines**

Once the encountered problem is identified, RA_i should isolate the failed components and lines ($fcL(SSG_i)$ and $flL(SSG_i)$) -by deactivating them- in order to avoid the failure propagation to the non-failed connected components.

- **Classify Faults**

We detail, in Algorithm 3 how, RA_i classifies the identified faults by dominance or by equivalence ($ft(fcL(SSG_i))$). We note that the function $VoltLevel(X)$ returns the voltage level of an electric component X (1, 2 or 3 for, respectively, the high, medium or low voltage levels).

Algorithm 3: Fault Classification.

Require: $fcL(SSG_i)$
Ensure: $ftfcL(SSG_i)$
$min \leftarrow minimum\ VoltLevel(fcL(SSG_i))$
$C_{min} \leftarrow \{c \in fcL / VoltLevel(c) = min\}$
if ($|C_{min}| = 1$) **then**
$ft(fcL(SSG_i)) \leftarrow$ dominance
dominant fault($(fcL(SSG_i)) \leftarrow C_{min}$
dominated faults($(fcL(SSG_i)) \leftarrow (fcL(SSG_i) \setminus C_{min}$
elsif ($|C_{min}| > 1$) **then**
$ft(fcL(SSG_i)) \leftarrow$ equivalence
equivalent faults $(fcL(SSG_i) \leftarrow C_{min}$

- **Search Local Solution**

According to the identified type of the occurred faults ($ft(fcL(SSG_i))$), RA_i is guided to resolve pertinently the encountered problem based on the strategy of fault management described in Section 3.1. In order to optimize the search for solution as well as the network route, RA_i follows a flexible strategy. It begins by searching a local solution that consists of a deactivated local emergency line ($\in SSG_i$). This line should procure sufficient load to supply the component(s) outputted by Algorithm 3. If there exist more than one solution, RA_i choose the one providing the bigger Remaining Load (RL = ReceivLoad(x) - ReqLoad(x)).

- **Search Non-local Solution**

When no local solution is found, RA_i begins by looking for a solution previously stored on the data base by requesting the DBAgent about the encountered problem relative to SSG_i. In fact, our system may have encountered and solved this problem beforehand. If there is no stored solution, RA_i looks for new non-local solution from the other connected sub-grids through MAgent(s).

4.2 MAgent Implementation: Non-local Solution Search

In case of neither local solution nor stored solution, RA_i creates a MAgent MA_j on the failed component C. MA_j visits all the connected devices to C through the existing outgoing lines called paths in order to collect useful information about; *(i)* Cumulative Remaining Load (*CRL*), *(ii)* Cumulative Priority (*CPr*) and *(iii)* End of the taken Path (*EoP* is equal to true if there is no longer component to be visited and false otherwise). MA_j is destructed when it reaches EoP (there is no longer component to be visited) or when the calculated *CRL* is negative (invalid sub-solution). If MA_j reaches its *EoP* with positive *CRL*, the taken path presents a solution. Before the destruction, MA_j notifies its creator by sending a message containing the collected information. Algorithm 4 describes how MA_j operates. Thus, all the existing alternatives are investigated since the created MAgent(s) visit(s) all the neighboring power sub-grids.

Algorithm 4: MAgent Movement.

for each visited component Y **do**
  calculate ($CRL; CPr$)
  if ($CRL(Y) < 0$) **then** Send Message to $RA_i(CRL, CPr)$
  elsif ($\nexists$ outgoing lines from Y) **then** $EoP \leftarrow$ true
  Send Message to $RA_i(CRL, CPr, EOP)$
  else $EOP \leftarrow$ false
  send message to $RA_i(RL, CPr, EoP)$
  switch (number of outgoing lines from Y)
  case =1:
    Move on Z, the unique connected component, to Y
    call Mobile Agent Movement on Z)
  case > 1:
    clone on each connected component
    for each clone **do**
      call MAgent Movement
    end for
  end switch
end for

5 SIMULATIONS

In this section, we present an example of an electrical grid including the three voltage levels (high, medium and low). The first contains 2 power generators and 4 lines. The second one comprises 4 transformers, 7 consumers and 18 lines including 5 emergency ones. Finally, the third level involves 6 transformers, 17 consumers and 24 lines including 7 emergency ones. We circled, in red, the electric components to which *FDIRSY* assigned RAgents and we drawn the emergency lines in yellow. We begin by presenting two fault cases; the first one is locally resolved while the second requires a solution from the other connected sub-grids. Then, we study the behavior of *FDIRSY* in terms of required time, exchanged messages and recovery rate by running on multiple injected problems. Figure 3 illustrates the simulation of the studied smart grid by FDIRSY as well as the principal interface of FDIRSY allowing user to design and parameterize a power grid (all types of electrical components and lines).

- **Local Resolution**

Let us consider, in the first case locally resolved, that there is a voltage instability problem p on the *LVL* going from *LVT6* to *LC13*. The concerned RAgent looks for the existence of solution to *LC*13 on which the dominant fault occurs. The solution consists of activating the emergency line going from *LC*12 to *LC*13. The required time for resolving p, denoted by *TG(p)*, is smaller than 1.8 μs (Table 1). Thanks to the proposed fault categorization, *FDIRSY* resolves only the fault on *LC*13 (corresponding to the dominant fault) instead of searching for solutions (for *LC*13 and *LC*16). Thus, *TG(p)* is reduced. The recovery is performed at 100% as all the occurred faults are resolved.

Table 1: Local Recovery for Dominant Fault on *LC13*

Task	Description	CPU
Detection	detects voltage problem on LVL	1 ηs
Isolation	deactivates the failed LVL	< 1 ηs
Fault Categorization	Dominant Fault on $LC13$ and dominated fault on $LC16$	$1,6$ ηs
Local search (onLC13)	$\exists$ emergency line between $LC12$ and $LC13$	$1,76$ μs
Resolution	activate the found emergency line	< 1 ηs
TG(p)	p is a voltage instability problem	$< 1,8$ μs

- **Non Local Resolution**

Let us consider, in the second case, that there is a frequency problem p on $LVT5$ which is resolved thanks to the instantiation of the MAgents. The new found solution consists on activating the non-local emergency line connecting *LC*17 to *LC*3. The required time, $TG(p)$, is smaller than 20 μs (2). It presents, also, a comparison between CPUs when p occurs for the first time and for the second or n^{th} time. The use of data base updated at run-time makes our approach gainful in terms of exchanged messages and CPUs. The recovery is performed at 75% as there are 3 repaired components from 4 failed ones. In fact, we can not propose a software solution for both frequency

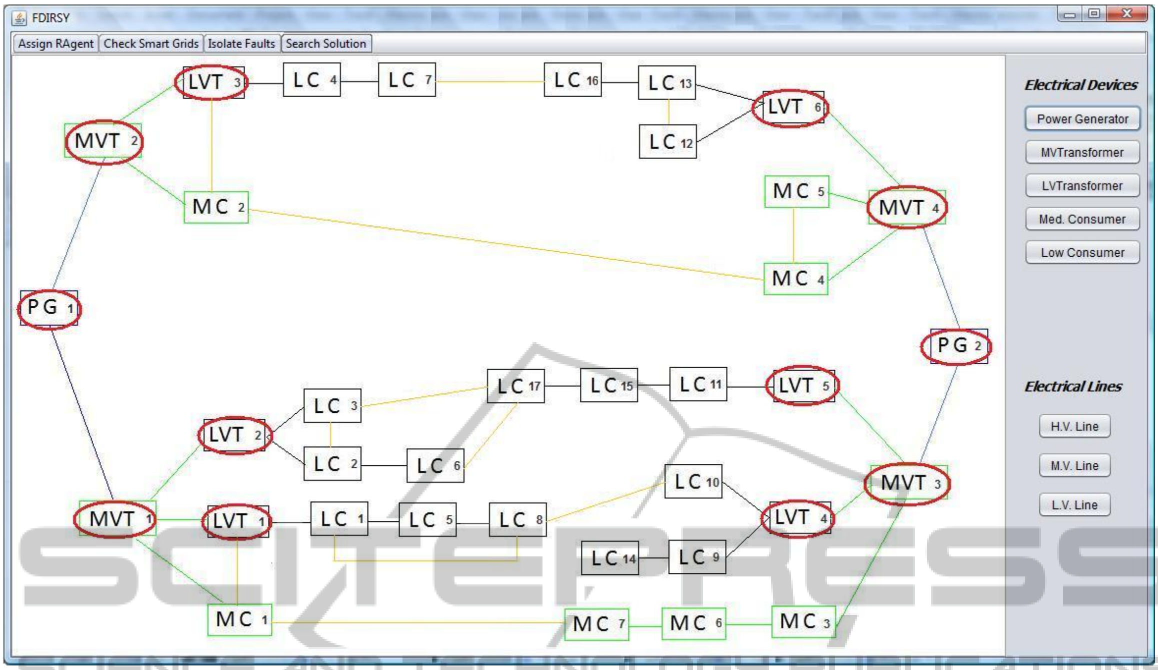

Figure 3: An example of smart power grid simulated by FDIRSY.

and instability problems as they need a physical human intervention. Thus, we look for resolving, only, the engendered faults (like in this case).

- **Multiple Problem Resolution**

In the last part of this section, we begin by estimating the required time *TG* for recovery in the studied power network. For that, we execute our system on multiple problems at the same time and at different times. In order to estimate the best and the worst *TG*, we run *FDIRSY* with multiple problems over the smallest and the bigger sub-grids at different times. These sub-grids are, respectively, composed of (3 components, 3 lines) and (6 components and 5 lines). In fact, the best and worst $TG(p)$ represent, respectively, the time spent to recover locally the first one and non-locally the second one (respectively equal to $1,45\ \eta s$ and $21\ \mu s$). We run, also, FDIRSY over the studied smart grid - which is composed of 12 sub-grids - on multiple faults at the same time. Figure 4 illustrates the TG spent to resolve p problems at the same time (at most $p = 8$).

It is, also, important to investigate the communication process between MAgents, RAgents and the DBAgent. For a given problem p at a time t; *(i)* if p is locally resolved, there is 0 exchanged messages, *(ii)* if p requires a non-local solution yet found, there is 1 exchanged message and *(iii)* if p requires a new non-local solution, the number of exchanged messages is equal to the number of the existing paths to which we add two messages (for the DBAgent request and for updating data base).

Table 2: Non-Local Recovery for Dominated Faults on *LC11*, *LC15* and *LC17*

Task	**Occurrence of p**	
	1^{st}	n^{th}
Detection	detect frequency problem on $LVT5$ $1\ \eta s$	$1\ \eta s$
Isolation	deactivate the failed line between $LVT5$ and $LC11$ $< 1\ \eta s$	$< 1\ \eta s$
Fault Categorization	dominant fault on $LVT5$ and dominated faults on $LC11$, $LC15$ and $LC17$ $4,8\ \mu s$	$4,8\ \mu s$
Local search	$\emptyset$ local solution for both of dominated and dominant faults $3\ \eta s$	$3\ \eta s$
DBAgent Request	$\emptyset$ stored solution $2\ \mu s$	$2,8\ \mu s$
Search non local solution	create MAgent on $LC11$ moving to the sub-grid supplied by $LVT2$ $13\ \mu s$	$0s$
Resolution	activate the found emergency line $< 1\ \eta s$	$< 1\ \eta s$
TG(p)	p is a frequency problem on $LVT5$ $< 20\ \mu s$	$< 7,6\ \mu s$

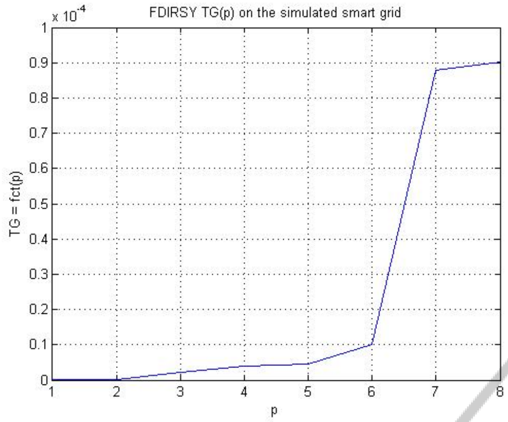

Figure 4: TG for resolving multiple problems.

6 CONCLUSION

In this paper, we propose an original approach for efficient smart power grid recovery. In order to evaluate the proposed approach we develop a simulator for smart grids. It ensures the detection of the faults, the identification of the encountered problem and the localization of the failed electrical components (even the consequent ones) thanks to the proposed fault categorization. It resolves the problems and searches the existing solutions according to the defined strategy. The experimental study showed that our approach is gainful in terms of faults to be resolved, CPU and communication. In future works, we look for introducing a learning module in order to deduce new solutions from other existing ones. We are, also, interested in large scale tests for larger power grids.

REFERENCES

Ben Meskina, S., Doggaz, N., and Khalgui, M. (2014). New solutions for fault detections and dynamic recoveries of flexible power smart grids. In 9^{th} *International Conference ICINCO in Informatics in Control, Automation and Robotics*.

Calderaro, V., Hadjicostis, C. N., Piccolo, A., and Siano, P. (2011). Failure identification in smart grids based on petri net modeling. *IEEE Transactions on Industrial Electronics*, pages 4613 – 4623.

Chertkov, M., Pan, F., and Stepanov, M. G. (2011). Predicting failures in power grids: The case of static overloads. *IEEE Transactions on Smart Grid*.

Fang, X., Misra, S., Xue, G., and Yang, D. (2012). Smart grid - the new and improved power grid: A survey. *IEEE Communications Surveys and Tutorials*.

Jiang, Z., Khalgui, M., Mosbahi, O., and Jaouadi, A. (2014a). A novel hierarchical multi-agent architecture for automatic restoration of smart grids. *International Journal of Control and Automation*.

Jiang, Z., Mosbahi, O., and Khalgui, M. (2014b). A multi-agent architecture for the self-healing of sgs based on iec 61499/61850. *Energy Education Science and Technology Part A. Energy Science and Research*.

Massoud, A. and Wollenberg, B. (2005). Toward a smart grid: power delivery for the 21st century. In *IEEE Power and Energy Magazine*, Minneapolis, MN, USA.

McArthur, S. D. J., Davidson, E. M., Catterson, V. M., Dimeas, A. L., Hatziargyriou, N. D., Ponci, F., and Funabashi, T. (2007a). Multi-agent systems for power engineering applications - part i: Concepts, approaches, and technical challenges. In *IEEE Transactions on Power Systems*.

McArthur, S. D. J., Davidson, E. M., Catterson, V. M., Dimeas, A. L., Hatziargyriou, N. D., Ponci, F., and Funabashi, T. (2007b). Multi-agent systems for power engineering applications part ii: Technologies, standards, and tools for building multi-agent systems. In *IEEE Transactions on Power Systems*.

Oudalova, A. and Fidigattib, A. (2009). Adaptive network protection in microgrids. *International Journal of Distributed Energy Resources*.

Pipattanasomporn, M., Feroze, H., and Rahman, S. (2009). Multi-agent systems in a distributed smart grid: Design and implementation. In *IEEE/PES Power Systems Conference and Exposition*, Adv. Res. Inst., Virginia Tech, Arlington, VA.

Rahman, S., Pipattanasomporn, M., and Teklu, Y. (2007). Intelligent distributed autonomous power systems (idaps). In *IEEE Power Engineering Society General Meeting*, Adv. Res. Inst. of Virginia Tech, Arlington, VA.

Ramchurn, S. D., Vytelingum, P., Rogers, A., and Jennings, N. (2011). Agent-based control for decentralised demand side management in the smart grid. In *The 10th International Conference on Autonomous Agents and Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems.

Russell, B. D. and Benner, C. L. (2010). Intelligent systems for improved reliability and failure diagnosis in distribution systems. *IEEE Transactions on Smart Grid*.

Vyatkin, V., Zhabelova, G., Ulieru, M., and McComas, D. (2010). Toward digital ecologies: Intelligent agent networks controlling interdependent infrastructures. In *First IEEE International Conference on Smart Grid Communications (SmartGridComm)*.