

Full Length Research Paper

Applications of intelligent multi-agent systems for smart distribution systems

Yu-Hsiu Lin¹ and Men-Shen Tsai^{2*}

¹Graduate Institute of Mechanical and Electrical Engineering, National Taipei University of Technology, 1, Sec. 3, Zhongxiao E. Rd., Taipei 10608 Taiwan, R.O.C.

²Graduate Institute of Automation Technology, National Taipei University of Technology, 1, Sec. 3, Zhongxiao E. Rd., Taipei 10608 Taiwan, R.O.C.

Received 8 August, 2014; Accepted 9 October, 2014

Multi-agent systems (MAS) can be applied to service restoration planning of distribution systems. The main purpose of service restoration planning is to restore as many customers as possible. In this paper, a MAS-based service restoration planner with a Genetic Algorithm (GA) is developed to solve group and zone service restoration problems in power distribution systems. Intelligent software agents embedded inside switches and feeders of a distribution system form distributed MAS. They autonomously co-work in parallel, so that time consumption on service restoration planning is kept into minimum. By transferring de-energized customers to their adjacent supporting distribution feeders, the agents running autonomously and sharing information with each other restore out-of-service loads. The proposed methodology is implemented on a Java Agent Development (JADE) platform. The feasibility of the proposed MAS-based restoration planner is demonstrated by a distribution system.

Key words: Believe-Desire-Intension (BDI) agents, distribution systems, Java Agent Development (JADE), service restoration.

INTRODUCTION

In a typical distribution system/network, loads are partitioned by many switches. There are two types of switches: sectionalized (normally closed, N.C.) switches and tie (normally opened, N.O.) switches. The status of the switches must be properly regulated such that the radial network constraint of distribution systems is satisfied. Normally, the network topology can be re-configured. When an emergency occurs, power service to de-energized loads can be restored by service restoration plans. Several service restoration strategies have been

proposed in Hsu et al. (1992), Tsai (2008), Chen et al. (2002), Parada et al. (2004), Huang (2003) and Kumar et al. (2008) with the following assumptions: 1) a centralized service restoration planner collecting all necessary information from fields is used; 2) the service restoration planning is based on combinatorial search; and 3) the switches are remotely controllable; that is, the service restoration planning made on the centralized service restoration planner can be executed remotely. However, massive data collected from a large-scale distribution

*Corresponding author. E-mail: mstsai@ntut.edu.tw.

Author(s) agree that this article remain permanently open access under the terms of the [Creative Commons Attribution License 4.0 International License](https://creativecommons.org/licenses/by/4.0/)

system for service restoration need to be handled by the centralized service restoration planner. It results in high computational complexity and time complexity. As a result, a distributed intelligent Multi-agent system (MAS)-based service restoration planner is needed for service restoration, where intelligent agents co-work in a parallel fashion so that service restoration planning progresses more efficiently. Hence, the computational complexity and time complexity are de-centralized and reduced (it is not computationally expensive). Several MAS-based service restoration architectures have been used to solve service restoration problems (Nagata and Sasaki, 2002; Solanki et al., 2007; Pan and Tsai, 2011).

In Nagata and Sasaki (2002), a multi-agent approach to power system restoration was presented. In Solanki et al. (2007), a deliberative MAS was proposed. In Pan and Tsai (2011), a Believe-Desire-Intension (BDI) (Pan and Tsai, 2011; Bratman, 1999) based MAS that build a simple decision tree to identify the service restoration plans was proposed. However, the service restoration planning strategy is based on heuristics. In this paper, a distributed intelligent BDI- and MAS-based service restoration planner that plans service restoration strategies based on either heuristics or meta-heuristics by a Genetic Algorithm (GA) in a parallel manner is proposed. By gathering and analyzing information from other intelligent agents, each intelligent agent is capable of autonomously progressing for service restoration planning. The proposed MAS-based service restoration planner verified and validated by a distribution system provides group restoration service and zone restoration service to power system restoration. The experimental results confirm that the MAS-based service restoration planner proposed in this paper works well.

PROBLEM FORMULATION

When a fault happens in a distribution system, power service to customers may be interrupted. The goal of the service restoration planning is to restore power service to de-energized consumers as soon as possible. The service restoration problem can be formulated as follows:

$$\text{Max} \sum_{i=1}^n x_i \cdot \text{load}_i \quad (1)$$

$$\text{Where, } x_i = \begin{cases} 0, & \text{if the } i\text{-th zone is de-energized.} \\ 1, & \text{if the } i\text{-th zone is energized.} \end{cases}$$

In Equation 1, n is the total number of zones. load_i is the load of the i -th zone. The constraints of the service restoration problem solved in this paper include: 1) the line current of all conductors must not exceed the conductor capacity; 2) the distribution network must hold

radial; and 3) the total load by each feeder must not exceed the feeder capacity. Besides, the distribution system considered in this paper has the following characteristics: 1) only the primary distribution feeder (11.4/22.8 kV) is considered; 2) three-phase laterals with switches connected to adjacent feeders are considered; and 3) three-phase currents are considered simultaneously.

METHODOLOGY FOR SERVICE RESTORATION PLANNING

The methodology for service restoration planning proposed in this paper is introduced here. In the proposed multi-agent architecture for the distribution system service restoration problems addressed in this paper, two different types of intelligent agents are defined.

1) Feeder agents: Feeder agents only respond to a query for the feeder ampacity. With responding to a query, they provide total served load current information and present available ampacity information. Each feeder agent maintains the information of exactly one adjacent switch that is directly connected to it (the feeder). Feeder agents do not make any service restoration planning.

2) Switch agents: Each switch agent, who records the current through the switch, is responsible for monitoring and controlling the status of the switch it is located at. This type of intelligent agents responds to queries from other intelligent agents. The topological information of adjacent intelligent agents of each intelligent agent is held by each switch agent. Adjacent intelligent agents can directly exchange their information. To gather more necessary information from non-adjacent intelligent agents, the adjacent intelligent agents have to rely on a message passing mechanism. Whenever a fault is detected and isolated occurs, switch agents start to plan the service restoration strategies.

Flowchart of the proposed self-healing approach

Figure 1 shows the flowchart of the proposed service restoration planning. The restoration planning starts when a fault detected occurs. Whenever a fault occurs, intelligent agents that belong to switch-type agents and are responsible for service restoration planning are identified and are named Leader Agents (LA) first. As soon as the LA(s) is identified, it starts communicating and interacting with other intelligent switch and feeder agents to explore the whole out-of-service (outage) area enclosed by the tie switches (within the tie switches). When all related information such as the available capacity of feeders is identified through a query process, the Las start planning service restoration strategies.

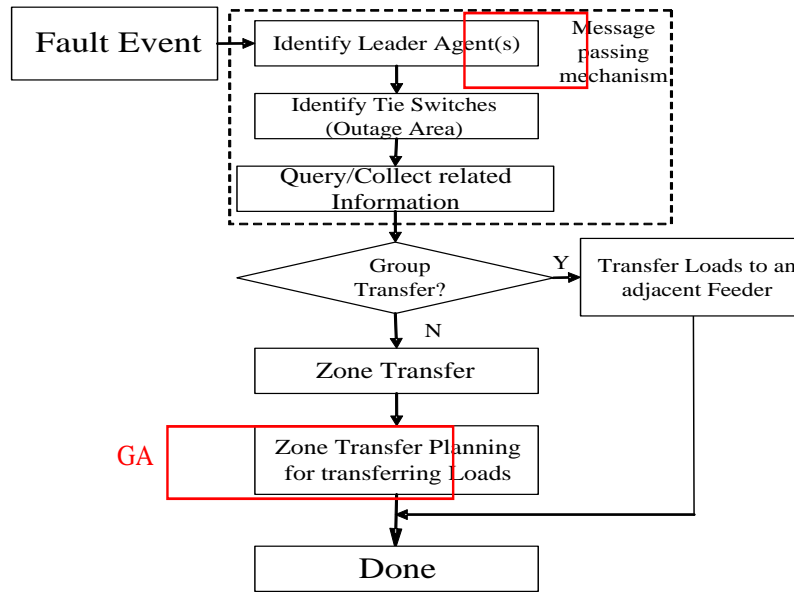


Figure 1. Flowchart of the self-healing strategy proposed in this paper.

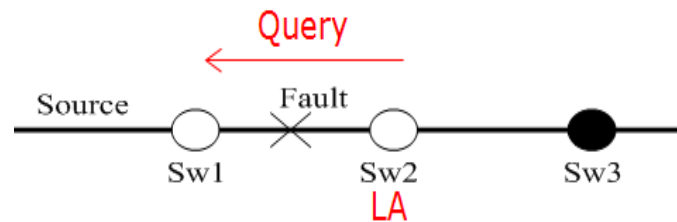


Figure 2. Occurrence of fault between two switches.

Finally, to restore power service to de-energized customers, intelligent agents perform necessary actions.

Message passing mechanism and service restoration planning

The message passing flow and the service restoration planning are discussed as follows:

Stage 1): Elect the LAs for service restoration planning

When a fault occurs, the status of the switches that surround the faulted section is marked "LOCKED_OPEN." When the intelligent agents installed inside the switches detect the change of the status, queries toward the power source side are sent by the intelligent agents. If the switch agent that received the query responds with "LOCKED_OPEN," the intelligent agent that sent the query is elected as the LA. The main goal of the LA(s) is to make an adequate service

restoration plan based on the retrieved information from other agents. For example, suppose that a fault occurred between Sw1 and Sw2, as shown in Figure 2, the status of the two switches is set to "LOCKED_OPEN." Messages from Sw2 and Sw1 are sent toward the power source side. In this case, Sw2 is elected as the LA. If the fault is surrounded by more than two switch agents as shown in Figure 3, messages from Sw2, Sw4, and Sw1 are sent towards the power source side. Since the source of Sw2 and Sw4 is Sw1 and the status of Sw1 is "LOCKED_OPEN", both switch agents, Sw2 and Sw4, are elected as the LAs. The two LAs co-work for service restoration planning in a parallel manner, such that the time consumption on service restoration planning is improved.

Stage 2): Acquire necessities for the outage area

The LAs send a query (queries) toward the down-stream sector, in order to identify all tie switches within the outage area. When a switch agent receives the query, it responds the topology and loading information to the LA.

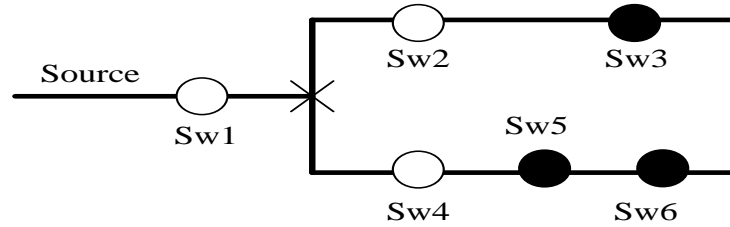


Figure 3. Occurrence of fault surrounded by three switches. In this situation, Sw2 and Sw4 are elected as the LAs; the service restoration strategies by the two LAs are separately and simultaneously planned (in a parallel way). Similarly, the service restoration strategies are planned by LAs in parallel such that the computational complexity and the time complexity are distributed and reduced, if there are multiple faults that occurred (at the same time) in a distribution system.

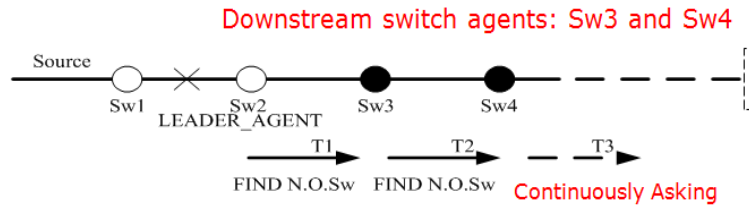


Figure 4. Search for tie switches.

If the status of the switch agent is not open, the query is continually passed to the switch agent’s downstream switch agent(s). For example, as shown in Figure 4, the LA, Sw2, sends a query to the downstream switch agent, Sw3. Since the status of Sw3 is not open, the switch agent passes the query to its downstream switch agent. In the meantime, it also passes the topology and loading information to the LA. The query process is continued until a tie switch or the feeder is met. As shown in Figure 5, when a tie switch is queried, it responds its identity to the leader agent. If the number of downstream switch agents of a switch agent is more than one, the query is passed to the downstream switch agents. A special message “a branch has been identified” is sent to the LA. Thus, the LA is able to keep track of the number of branches that exist at its downstream sector. When all the tie switches send their identity to the LA, the search process terminates.

Stage 3): Identify the available ampacity of adjacent feeders

The intelligent agent located at the tie switch found in Stage 2 sends the feeder capacity query message to its source-side intelligent agent. If the intelligent agent that receives the message is a feeder agent, it responds with the available capacity information to the tie switch agent; otherwise, it passes the query message to its source-side intelligent agent until a feeder agent is met. Finally, the

adjacent feeder’s available capacity information is sent to the LA. Figure 6 shows the search process.

Stage 4): LA(s) starts planning service restoration strategies

When all the related information is retrieved, the LA(s) starts planning service restoration strategies as follows:

Group restoration planning: The margin of one of all the adjacent feeders is sufficient enough for fully supporting all the de-energized customers in the out-of-service area.

Zone restoration planning: The zone restoration planning process starts when the group restoration planning process fails to restore all de-energized customers in an outage area(s). The zone restoration planning is composed of a heuristic approach and a GA. The GA is triggered for service restoration planning, when the heuristic approach is not able to identify an adequate service restoration plan. The heuristic approach is based on the Maximum capacity principle. During the service restoration planning, outage areas that are adjacent to the feeders are energized by the feeders. The feeders’ available capacity is updated. Then, the power service to the rest of the areas is restored in turn by the feeder with the maximum available capacity at the current run. It progresses until each feeder’ available capacity is not

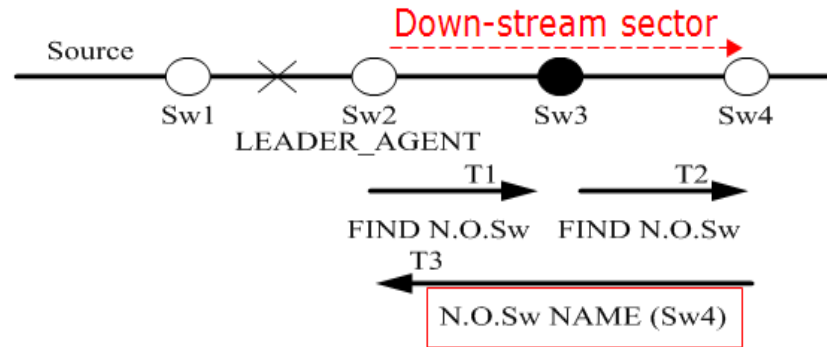


Figure 5. Information response from a tie-switch agent.

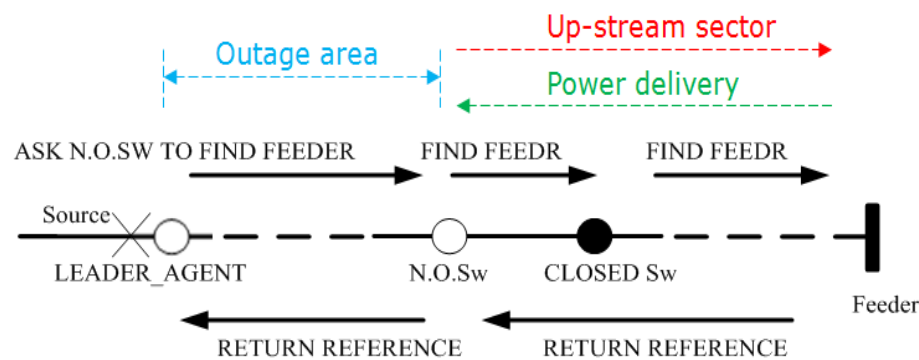


Figure 6. Search for an adjacent feeder.

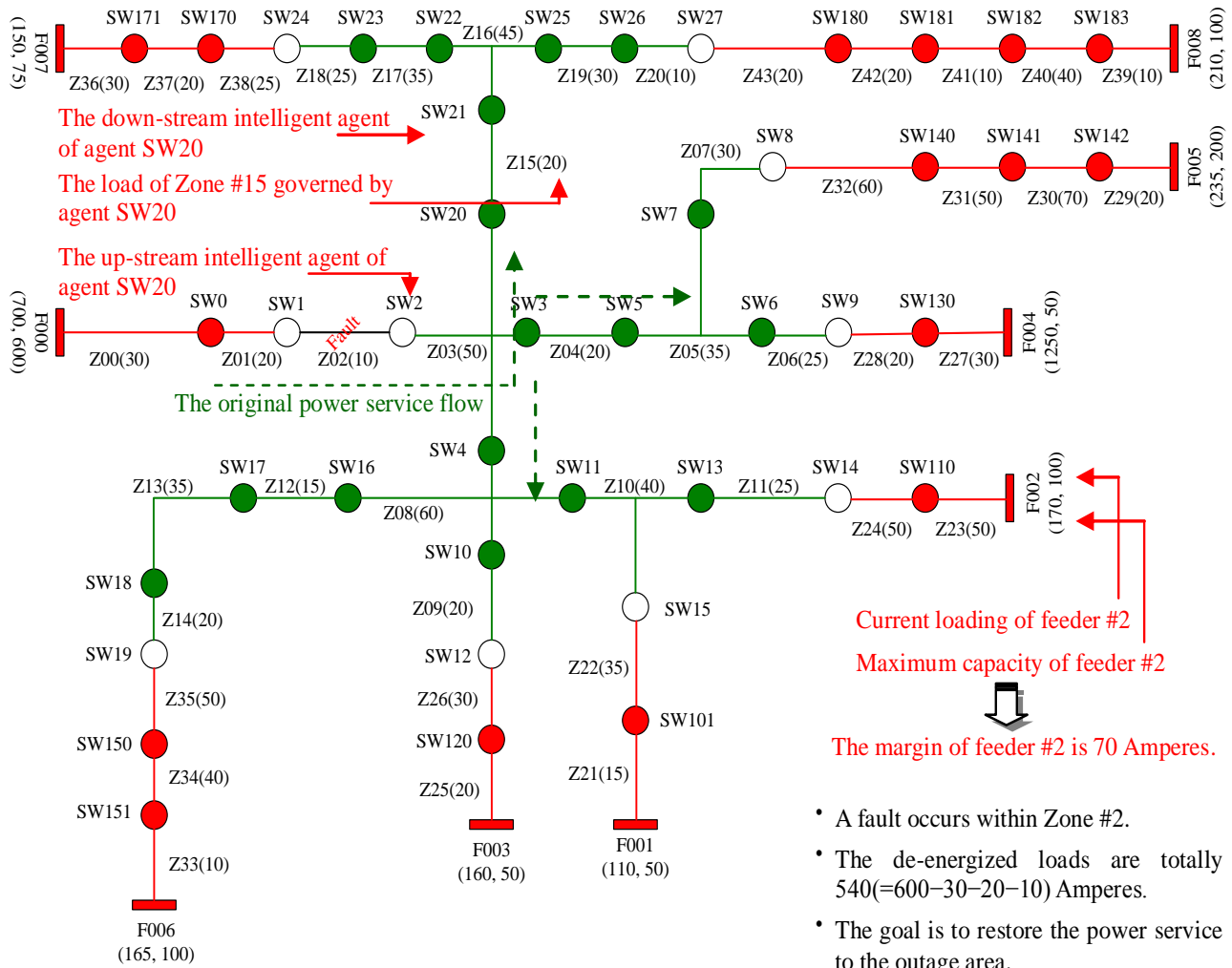
sufficient for supporting the area that is located at the downstream sector of the feeder. In some cases, the heuristic approach is not able to restore power service to all de-energized zones. As a result, the population-based stochastic GA (Gen and Cheng, 1997) is conducted. It globally searches for the optimal network topology. With use of the GA planning service restoration strategies, one of the most important steps is to encode network topologies as chromosomes (genotype). Each gene, a binary number (0: Opened/1: Closed), of a chromosome presents the status of a switch on a feeder. To evaluate the quality of each chromosome, a fitness function, which guides the GA to search for the optimal solution, is needed. Since the service restoration to all de-energized zones is the final goal, the fitness function is defined as (1). During evaluation, the GA decodes each chromosome into the corresponding network topology (phenotype), checks whether the constraints are satisfied or not, and evaluates the fitness of each legal chromosome. The constraints have been mentioned earlier. If any violation exists, the reject mechanism that deletes the illegal ones is performed. After all legal chromosomes in the current population are evaluated, the natural (roulette wheel) selection process that mimics the process of Darwinian evolution to create populations

from generation to generation is performed. The genetic operators facilitate an efficient search and guide the search into new solution regions. One-point crossover and bit-wise flipping mutation operators are used in this paper.

EXPERIMENTATION

The proposed MAS-based service restoration planner with the GA in this paper is implemented on the Java Agent Development (JADE) platform (Bellifemine et al., 2007) which is a platform for developing MASs and running the agents. It complies with the Foundation for Intelligent Physical Agents (FIPA) standards (Zidan and El-Saadany, 2012; Nguyen and Flueck, 2012). FIPA is an IEEE Computer Society Standards Organization, which promotes agent-based technology and the interoperability of its standards with other technologies. A test distribution system used to demonstrate the feasibility of the proposed multi-agent approach using the object-oriented design is shown in Figure 7. The distribution system is constructed by 9 feeders with 43 switches. In this paper, three different scenarios were simulated. During the experimentation, the following two assumptions are made:

- 1) The distribution systems with load variations are not covered; and
- 2) All load zones (customers) to be restored have the same load priority.



- A fault occurs within Zone #2.
- The de-energized loads are totally $540(=600-30-20-10)$ Amperes.
- The goal is to restore the power service to the outage area.

Figure 7. A large-scale distribution system used during the experiments conducted in this paper.

In Simulation 1, a fault occurred between Sw1 and Sw2 (within zone Z02). The leader agent elected was Sw2. In this simulation, the de-energized loads are totally 540 A. The goal is to restore the power service to the whole outage area shown in green color in Figure 7. According to the network typology shown in Figure 7, intelligent switch agents and feeder agents are set up as shown in Figure 8. Figure 9 shows the JADE remote agent management GUI showing the set-up intelligent switch agents and feeder agents. Figure 10 shows the message passing and querying process. The interaction of intelligent switch and feeder agents leads to a dynamic and effective system allowing an efficient service restoration to a solution. Intelligent switch agents and feeder agents exchange information and autonomously co-work in a parallel fashion so that the service restoration planning progresses more efficiently. The outage area in this simulation is finally figured out through the message passing mechanism. The group restoration result obtained in this simulation is reported in Figure 11. The margin of feeder #4 (F004) was very sufficient for supporting all the out-of-service zones in the outage area. Hence, the LA, Sw2, made a decision that Sw9 is switched on; the group restoration was made, where all the de-energized customers were served. The computational time is less than 10 s. Figure 12 shows the

finally-configured topology of the distribution system. No any constraint was violated and the goal was achieved.

In Simulation 2, the heuristics-based zone restoration planning process is executed by the MAS proposed in this paper for service restoration planning, since the margin of one of all the adjacent feeders is not sufficient enough for fully supporting the out-of-service customers in the outage area as seen in Figure 13(a). The message passing mechanism has been presented earlier. Also, based on the heuristic-based zone restoration planning strategy-Maximum capacity principle introduced earlier, the MAS fully restore a total of 540 Amperes. During the service restoration process, all the constraints are satisfied. The computational time is less than 10 s. The final topology of the distribution system is shown in Figure 13(b).

In Simulation 3, the GA-based zone restoration planning where switches in a distribution system are encoded and evolved through the genetic operations-crossover and mutation and the evolution operation-selection was adopted by the proposed MAS, because the heuristics fail in this case. The GA globally searches for the optimal network topology. The distribution system tested is illustrated in Figure 14. The message passing mechanism and the meta-heuristic-based zone restoration planning strategy by a GA


```

aaabat - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
set JADE_HOME=C:\JADE-3.7.android
set JADE_LIB=%JADE_HOME%\lib\commons-codec\commons-codec-1.3.jar;%JADE_HOME%\lib\http.jar;%JADE_HOME%\lib\iiop.jar;%JADE_HOME%\lib\jade.jar;
%JADE_HOME%\lib\jadeTools.jar;
start "sw2" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Switch2 sw2:SwitchAgent(close 50 sw1 sw3 sw4 sw20)
start "sw3" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Switch3 sw3:SwitchAgent(close 20 sw2 sw5)
start "sw4" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Switch4 sw4:SwitchAgent(close 60 sw2 sw10 sw11 sw16)
start "sw5" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Switch5 sw5:SwitchAgent(close 35 sw3 sw6 sw7)
start "sw6" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Switch6 sw6:SwitchAgent(close 25 sw5 sw9)
start "sw7" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Switch7 sw7:SwitchAgent(close 30 sw5 sw8)
start "sw8" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Switch8 sw8:SwitchAgent(open 0 sw7 sw140)
start "sw9" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Switch9 sw9:SwitchAgent(open 0 sw6 sw130)
start "sw10" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Switch10 sw10:SwitchAgent(close 20 sw4 sw12)
start "sw11" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Switch11 sw11:SwitchAgent(close 40 sw4 sw13 sw15)
start "sw12" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Switch12 sw12:SwitchAgent(open 0 sw10 sw120)
start "sw13" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Switch13 sw13:SwitchAgent(close 20 sw12 sw120)
start "sw14" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Switch14 sw14:SwitchAgent(close 20 sw12 sw120)
start "sw15" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Switch15 sw15:SwitchAgent(close 20 sw12 sw120)
start "sw16" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Switch16 sw16:SwitchAgent(close 20 sw12 sw120)
start "sw17" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Switch17 sw17:SwitchAgent(close 35 sw16 sw18)
start "sw18" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Switch18 sw18:SwitchAgent(close 20 sw17 sw19)
start "sw19" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Switch19 sw19:SwitchAgent(open 0 sw18 sw150)
start "sw20" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Switch20 sw20:SwitchAgent(close 20 sw2 sw21)
start "sw21" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Switch21 sw21:SwitchAgent(close 45 sw20 sw22 sw25)
start "sw22" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Switch22 sw22:SwitchAgent(close 35 sw21 sw23)
start "sw23" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Switch23 sw23:SwitchAgent(close 25 sw22 sw24)
start "sw24" /MIN /I java -classpath %JADE_LIB% jade.Boot -co -container-name Switch24 sw24:SwitchAgent(close 25 sw23 sw24)
start "sw25" /MIN /I java -classpath %JADE_LIB% jade.Boot -co -container-name Switch25 sw25:SwitchAgent(close 25 sw23 sw24)
start "sw26" /MIN /I java -classpath %JADE_LIB% jade.Boot -co -container-name Switch26 sw26:SwitchAgent(close 25 sw23 sw24)
start "sw27" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Switch27 sw27:SwitchAgent(open 0 sw26 sw100)

rem
rem

```

The upstream switch of sw20

The status of sw20

The downstream switch of sw20

The loading of the zone governed by sw20

```

aaabat - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
rem
rem
start "sw150" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Switch150 sw150:SwitchAgent(close 50 sw151 sw19)
start "sw151" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Switch151 sw151:SwitchAgent(close 40 f006 sw150)
start "f006" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Feeder006 f006:FeederAgent(165 100)

rem
rem
start "sw101" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name sw101:SwitchAgent(close 35 f001 sw15)
start "f001" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Feeder001 f001:FeederAgent(110 50)

rem
rem
rem
rem
start "sw110" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Switch110 sw110:SwitchAgent(close 50 f002 sw14)
start "f002" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Feeder002 f002:FeederAgent(170 100)

rem
rem
rem
rem
start "sw120" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Switch120 sw120:SwitchAgent(close 30 f003 sw12)
start "f003" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Feeder003 f003:FeederAgent(160 50)

rem
rem
rem
rem
start "sw130" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Switch130 sw130:SwitchAgent(close 20 f004 sw9)
start "f004" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Feeder004 f004:FeederAgent(1250 50)

rem
rem
rem
rem
start "sw140" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Switch140 sw140:SwitchAgent(close 60 sw141 sw8)
start "sw141" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Switch141 sw141:SwitchAgent(close 50 sw142 sw140)
start "sw142" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Switch142 sw142:SwitchAgent(close 20 f005 sw141)
start "f005" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Feeder005 f005:FeederAgent(235 200)

rem
rem
rem
rem
start "sw170" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Switch170 sw170:SwitchAgent(close 25 sw171 sw24)
start "sw171" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Switch171 sw171:SwitchAgent(close 20 f007 sw170)
start "f007" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Feeder007 f007:FeederAgent(210 100)

rem
rem
rem
rem
start "sw180" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Switch180 sw180:SwitchAgent(close 20 sw181 sw27)
start "sw181" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Switch181 sw181:SwitchAgent(close 20 sw182 sw180)
start "sw182" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Switch182 sw182:SwitchAgent(close 10 sw183 sw181)
start "sw183" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Switch183 sw183:SwitchAgent(close 40 f008 sw182)
start "f008" /MIN /I java -classpath %JADE_LIB% jade.Boot -container -container-name Feeder008 f008:FeederAgent(210 100)

```

The maximum capacity of f005

The current loading of f005

Figure 8. Intelligent switch agents for planning service restoration strategies and intelligent feeder agents are set up (through a *.bat file).

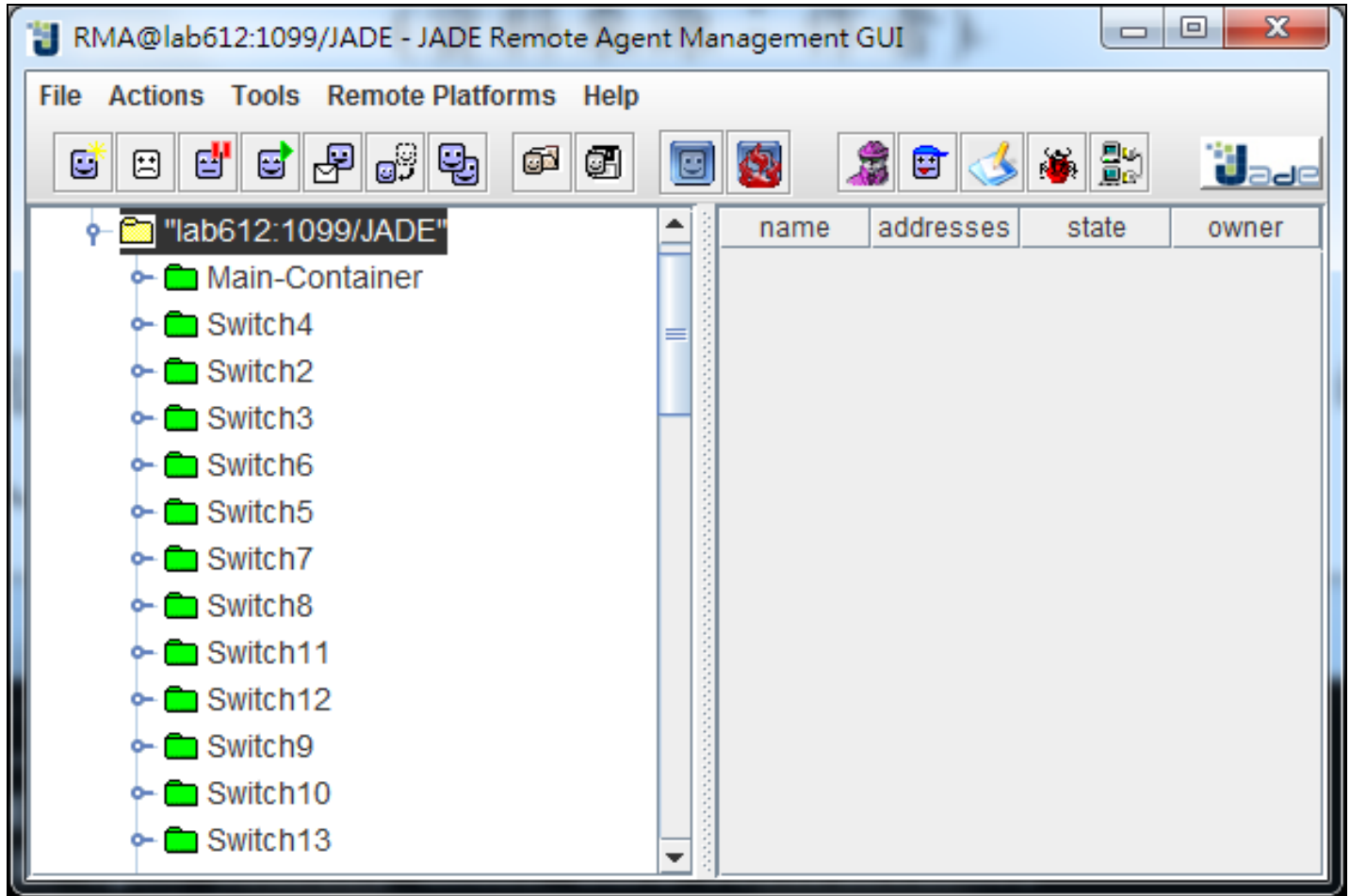


Figure 9. JADE remote agent management GUI showing the set-up intelligent switch agents and feeder agents. The Remote Monitoring Agent (RMA) is a graphical console to monitor the agent platform activity and visualize container and agent states. The number of switch agents/feeder agents is the same as the number of switches/feeders in a distribution system. Intelligent agents in JADE communicate by the message passing mechanism, where the messages adhere to Agent Communication Language (ACL) standards.

have been presented earlier. For the GA, the population size is 20. The maximum generation is 10; and the crossover rate and the mutation rate are 0.45 and 0.05, respectively. Figure 15 shows the evolution by the GA. The computational time is less than 10 s. The final topology of the distribution system re-configured by the proposed MAS in this simulation is shown in Figure 16. As shown in Figure 16, 3 out-of-service zones, Z3 (load: 50), Z4 (load: 20) and Z8 (load: 60), existed. ~75.93 of power service (410 of 540 A) was served. The search accuracy/efficiency of the GA employed in this simulation is improvable.

For this service restoration problem, the legal solutions are finite. For the encoding method used by the GA, most of the offspring generated by the genetic operators are illegal. A lot of time is wasted, due to the large number of illegal solutions. If an appropriate encoding method is conducted, the search accuracy/efficiency of the GA can be improved. And, the power service to all the outage areas can be fully restored.

Conclusions

This paper presents a completely distributed MAS-based

service restoration planner with the heuristics and the GA for distribution systems service restoration problems. Compared with the centralized one, the proposed MAS-based service restoration planner has several advantages. For instance, service restoration planning is made in a parallel fashion. The experimentation reported in this paper shows that the proposed distributed MAS-based service restoration planner using the agent- and object-oriented design and reporting the adequate service restoration plan is feasible. The quality of service restoration planning is improved. This is because the MAS is conducted to the smart distribution systems. In a MAS, each intelligent agent merely collects local information, while communicating and interacting with other intelligent agents to figure out the whole outage area(s). Then, the service restoration planning to a large-scale distribution system that suffers from at least one fault (multiple faults may occur) can be de-centralized into a series of sub-tasks (it can be tackled in a divide-and-conquer manner).


```

sw2
Received duplicated Switch : sw14
1290498355779 Recv INFORM:Switch from sw10
1290498355779 Recv INFORM:Feeder from sw15
Removing sw10 from downSwitchTable
downSwitchTable size : 17
Received duplicated Switch : sw10
Removing f001 from waitFeederTable
Size of waitFeederTable 4
Adding feeder f001 to feederTable
1290498355794 Recv INFORM:Switch from sw24
An open switch sw24 is received
size of waitFeeder : 5
Removing sw24 from downSwitchTable
downSwitchTable size : 16
Received duplicated Switch : sw24
1290498355794 Recv INFORM:Switch from sw6
Removing sw6 from downSwitchTable
downSwitchTable size : 15
Received duplicated Switch : sw6
1290498355794 Recv INFORM:Feeder from sw8
Removing f005 from waitFeederTable
Size of waitFeederTable 4
Adding feeder f005 to feederTable
1290498355810 Recv INFORM:Switch from sw7
Removing sw7 from downSwitchTable
downSwitchTable size : 14
Received duplicated Switch : sw7
1290498355826 Recv INFORM:Switch from sw5
Removing sw5 from downSwitchTable
downSwitchTable size : 13
Received duplicated Switch : sw5
1290498355826 Recv INFORM:Feeder from sw14
Removing f002 from waitFeederTable
Size of waitFeederTable 3

```

```

sw2
1290498355904 Recv INFORM:Switch from sw20
Removing sw20 from downSwitchTable
downSwitchTable size : 0
Update through current from 375.0 to 540.0
Outage Area Information :
-----
1 : Id: sw4 St: close Src: sw2 Id.C: 60 th.C: 215 Nb: sw10 sw11 sw16
2 : Id: sw3 St: close Src: sw2 Id.C: 20 th.C: 110 Nb: sw5
3 : Id: sw2 St: LOCKED Src: sw1 Id.C: 50 th.C: 540 Nb: sw3 sw4 sw20
4 : Id: sw19 St: open Src: sw18 Id.C: 0 th.C: 0 Nb: sw150
5 : Id: sw18 St: close Src: sw17 Id.C: 20 th.C: 20 Nb: sw19
6 : Id: sw17 St: close Src: sw16 Id.C: 35 th.C: 55 Nb: sw18
7 : Id: sw16 St: close Src: sw4 Id.C: 15 th.C: 70 Nb: sw17
8 : Id: sw15 St: open Src: sw11 Id.C: 0 th.C: 0 Nb: sw101
9 : Id: sw14 St: open Src: sw13 Id.C: 0 th.C: 0 Nb: sw110
10 : Id: sw13 St: close Src: sw11 Id.C: 25 th.C: 25 Nb: sw14
11 : Id: sw12 St: open Src: sw10 Id.C: 0 th.C: 0 Nb: sw120
12 : Id: sw11 St: close Src: sw4 Id.C: 40 th.C: 65 Nb: sw13 sw15
13 : Id: sw10 St: close Src: sw4 Id.C: 20 th.C: 20 Nb: sw12
14 : Id: sw27 St: open Src: sw26 Id.C: 0 th.C: 0 Nb: sw180
15 : Id: sw26 St: close Src: sw25 Id.C: 10 th.C: 10 Nb: sw27
16 : Id: sw25 St: close Src: sw21 Id.C: 30 th.C: 40 Nb: sw26
17 : Id: sw24 St: open Src: sw23 Id.C: 0 th.C: 0 Nb: sw170
18 : Id: sw23 St: close Src: sw22 Id.C: 25 th.C: 25 Nb: sw24
19 : Id: sw22 St: close Src: sw21 Id.C: 35 th.C: 60 Nb: sw23
20 : Id: sw9 St: open Src: sw6 Id.C: 0 th.C: 0 Nb: sw130
21 : Id: sw21 St: close Src: sw20 Id.C: 45 th.C: 145 Nb: sw22 sw25
22 : Id: sw8 St: open Src: sw7 Id.C: 0 th.C: 0 Nb: sw140
23 : Id: sw20 St: close Src: sw2 Id.C: 20 th.C: 165 Nb: sw21
24 : Id: sw7 St: close Src: sw5 Id.C: 30 th.C: 30 Nb: sw8
25 : Id: sw6 St: close Src: sw5 Id.C: 25 th.C: 25 Nb: sw9
26 : Id: sw5 St: close Src: sw3 Id.C: 35 th.C: 90 Nb: sw6 sw7

1 : Id : f006 Asoc : sw19 Amp : 165.0 IdC : 100.0
2 : Id : f008 Asoc : sw27 Amp : 210.0 IdC : 100.0
3 : Id : f001 Asoc : sw15 Amp : 110.0 IdC : 50.0
4 : Id : f002 Asoc : sw14 Amp : 170.0 IdC : 100.0
5 : Id : f004 Asoc : sw9 Amp : 1250.0 IdC : 50.0
6 : Id : f007 Asoc : sw24 Amp : 150.0 IdC : 75.0
7 : Id : f005 Asoc : sw8 Amp : 235.0 IdC : 200.0
8 : Id : f003 Asoc : sw12 Amp : 160.0 IdC : 50.0

Outage set:
SET : sw10 sw11 sw13 sw16 sw17 sw18 sw2 sw20 sw21 sw22 sw23 sw25 sw2
6 sw3 sw4 sw5 sw6 sw7
-----
Building restoration area for tie sw17 with ampacity of 165.0
Not enough capacity to pickup sw11
Q : sw18 sw17 sw16 sw4 sw10
SET : sw10 sw16 sw17 sw18 sw4
Building restoration area for tie sw27 with ampacity of 210.0
Not enough capacity to pickup sw2
Q : sw26 sw25 sw21 sw22 sw20 sw23
SET : sw20 sw21 sw22 sw23 sw25 sw26
Building restoration area for tie sw15 with ampacity of 110.0
Not enough capacity to pickup sw4
Q : sw11 sw13
SET : sw11 sw13
Building restoration area for tie sw14 with ampacity of 170.0
Not enough capacity to pickup sw2
Q : sw13 sw11 sw4 sw10 sw16
SET : sw10 sw11 sw13 sw16 sw4
Building restoration area for tie sw9 with ampacity of 1250.0
Building restoration area for tie sw24 with ampacity of 150.0

```

The LA:
sw2

Figure 10. Message passing and querying. The outage area information is collected through the message passing mechanism.

```
sw2
Not enough capacity to pickup sw4
Q : sw11 sw13
SET : sw11 sw13
Building restoration area for tie sw14 with ampacity of 170.0
Not enough capacity to pickup sw2
Q : sw13 sw11 sw4 sw10 sw16
SET : sw10 sw11 sw13 sw16 sw4
Building restoration area for tie sw9 with ampacity of 1250.0
Building restoration area for tie sw24 with ampacity of 150.0
Not enough capacity to pickup sw20
Q : sw23 sw22 sw21 sw25
SET : sw21 sw22 sw23 sw25
Building restoration area for tie sw8 with ampacity of 235.0
Not enough capacity to pickup sw20
Q : sw7 sw5 sw6 sw3 sw2 sw4
SET : sw2 sw3 sw4 sw5 sw6 sw7
Building restoration area for tie sw12 with ampacity of 160.0
Not enough capacity to pickup sw2
Q : sw10 sw4 sw11 sw16
SET : sw10 sw11 sw16 sw4
Tie SW : sw19 Ampacity : 165.00 Outage : 390.00
Served : sw18 sw17 sw16 sw4 sw10
Boundary : sw11 sw12 sw19 sw4

Tie SW : sw15 Ampacity : 110.00 Outage : 475.00
Served : sw11 sw13
Boundary : sw11 sw14 sw15

Tie SW : sw27 Ampacity : 210.00 Outage : 375.00
Served : sw26 sw25 sw21 sw22 sw20 sw23
Boundary : sw20 sw24 sw27

Tie SW : sw14 Ampacity : 170.00 Outage : 380.00
Served : sw13 sw11 sw4 sw10 sw16
Boundary : sw12 sw14 sw15 sw17 sw4

-----
Tie SW : sw9 Ampacity : 1250.00 Outage : 0.00
Served : sw6 sw5 sw7 sw3 sw2 sw4 sw20 sw10 sw11 sw16
sw21 sw13 sw17 sw22 sw25 sw18 sw23 sw26
Boundary : sw12 sw14 sw15 sw19 sw2 sw24 sw27 sw8 sw9

-----
Tie SW : sw12 Ampacity : 160.00 Outage : 405.00
Served : sw10 sw4 sw11 sw16
Boundary : sw12 sw13 sw15 sw17 sw4

Tie SW : sw8 Ampacity : 235.00 Outage : 320.00
Served : sw7 sw5 sw6 sw3 sw2 sw4
Boundary : sw10 sw11 sw16 sw2 sw20 sw8 sw9

Tie SW : sw24 Ampacity : 150.00 Outage : 405.00
Served : sw23 sw22 sw21 sw25
Boundary : sw21 sw24 sw26

-----
Restoration plan :
Tie SW : sw9 Ampacity : 1250.00 Outage : 0.00
Served : sw6 sw5 sw7 sw3 sw2 sw4 sw20 sw10 sw11 sw16
sw21 sw13 sw17 sw22 sw25 sw18 sw23 sw26
Boundary : sw12 sw14 sw15 sw19 sw2 sw24 sw27 sw8 sw9
```

No outage

Figure 11. Group restoration result in this simulation. The message passing mechanism and the group restoration planning strategy is presented in the work. The group restoration decision was made, where all the de-energized customers were served.

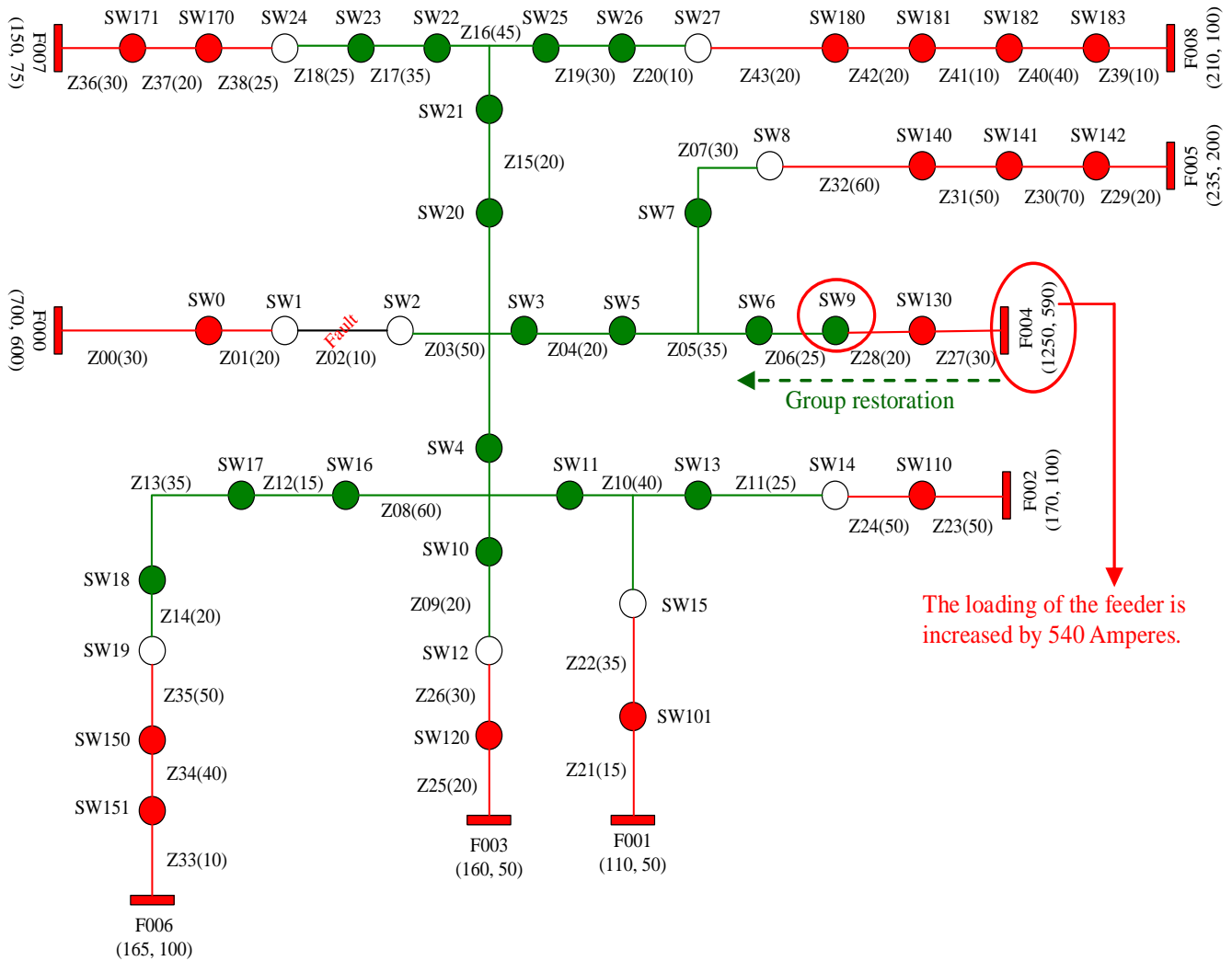


Figure 12. The finally-configured network topology of the distribution system given in Figure 7 in this simulation is obtained through the self-healing process by the proposed MAS. The network topology is re-configured through *switching*. In this simulation, the margin of F004 was very sufficient for supporting all the out-of-service zones.

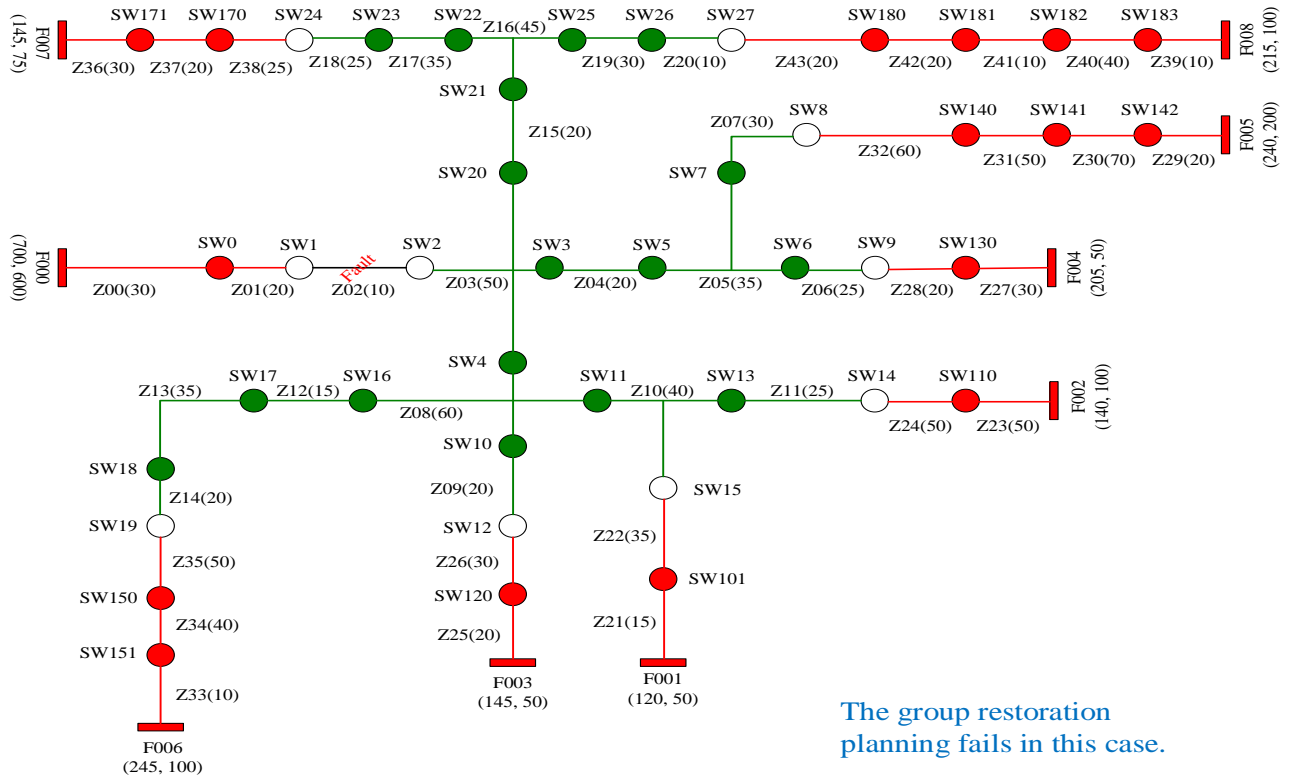
With a parallel, asynchronous, and autonomous process of planning service restoration strategies, the computational complexity and the time complexity are distributed and reduced. Besides, the MAS have a high degree of scalability and flexibility. Overall, for fault detection and isolation and service restoration, a decentralized multi-agent framework could outperform the centralized ones requiring a powerful central computer that may lead to a single point of failure.

In the future, if an appropriate encoding method is conducted, the search accuracy and efficiency of the GA can be improved. It is expected that the power service to all de-energized consumers in an outage area(s) is fully restored. The next step of this work is to improve the performance of the MAS proposed in this paper, in order for coping with the difficulty in self-healing distribution

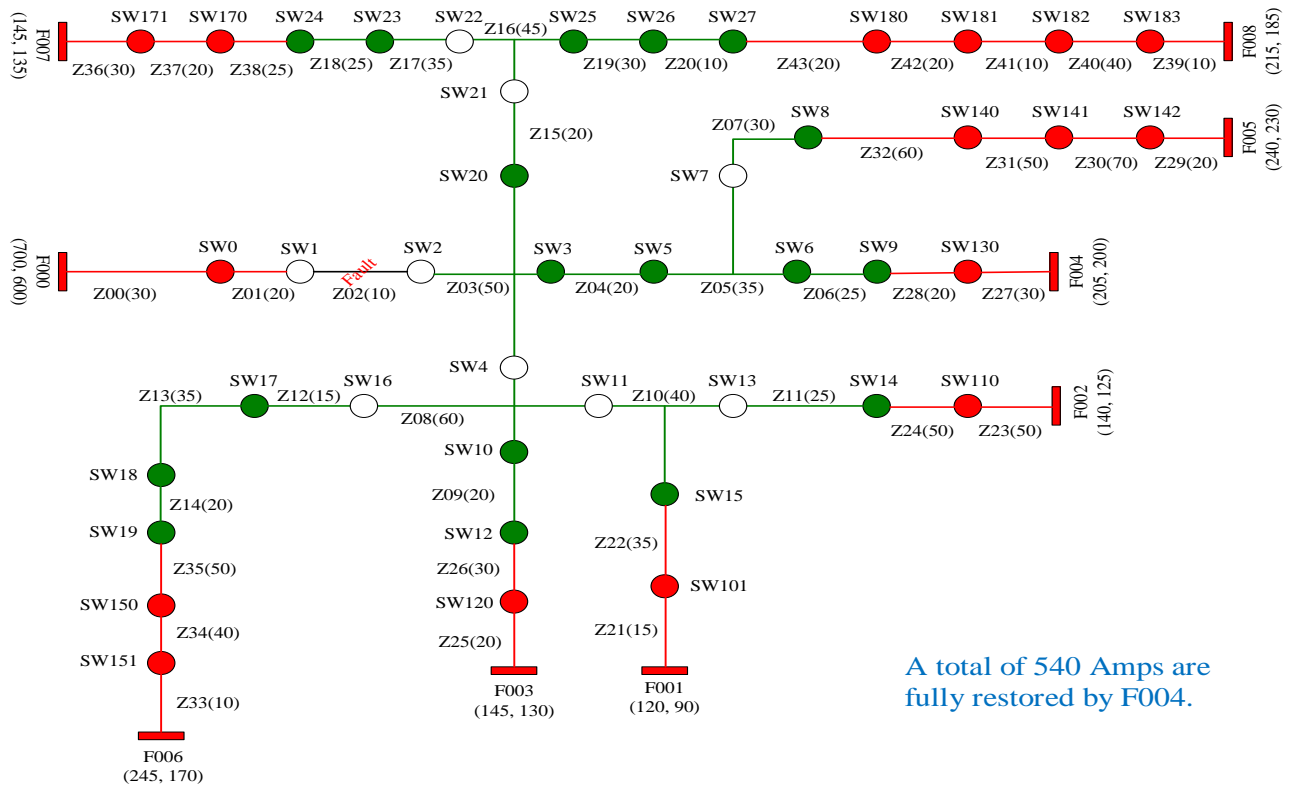
systems under multiple faults. The enhanced MAS will be a promising approach to more complex large-scale distribution networks that suffer from multiple faults.

ACKNOWLEDGEMENTS AND LEGAL RESPONSIBILITY

This paper was supported in part by the National Science Council, Taiwan, under the Grant No. NSC 101-2221-E-027-097-MY2- and MOST 103-3113-E-006 -011, and was sponsored in part by the Ministry of Education (MOE) with Technological University Paradigms, Taiwan, under the Grant No. 7021401-4. This study is an original work that has neither appeared elsewhere for publication, nor which is under review for another refereed publication.



(a)



(b)

Figure 13. A large-scale distribution system used in Simulation 2 (a) and the final topology configured by the proposed MAS for power service restoration (b).

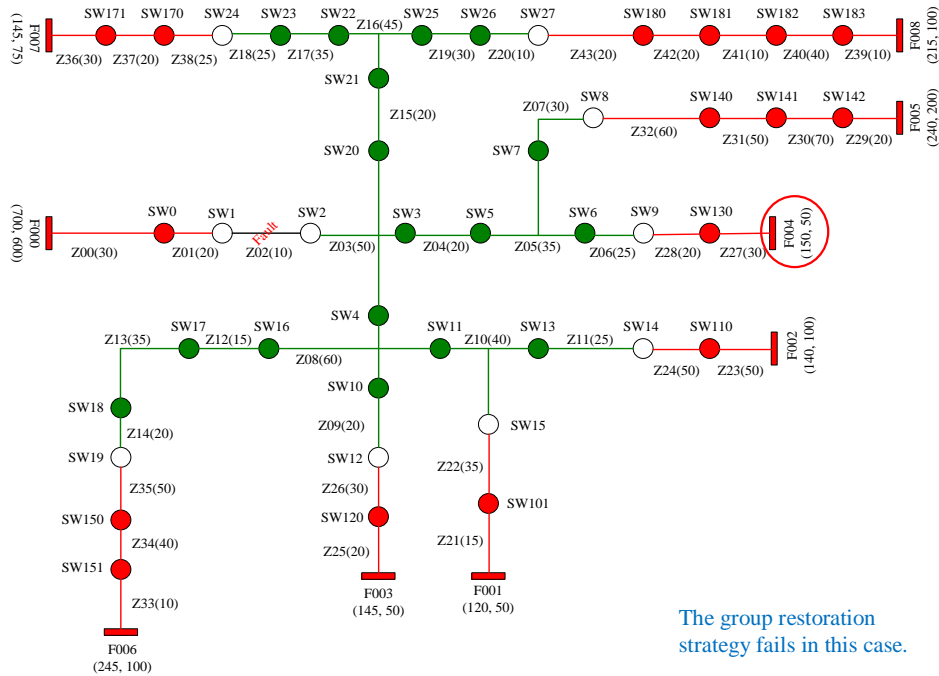


Figure 14. A large-scale power distribution system used in Simulation 3.

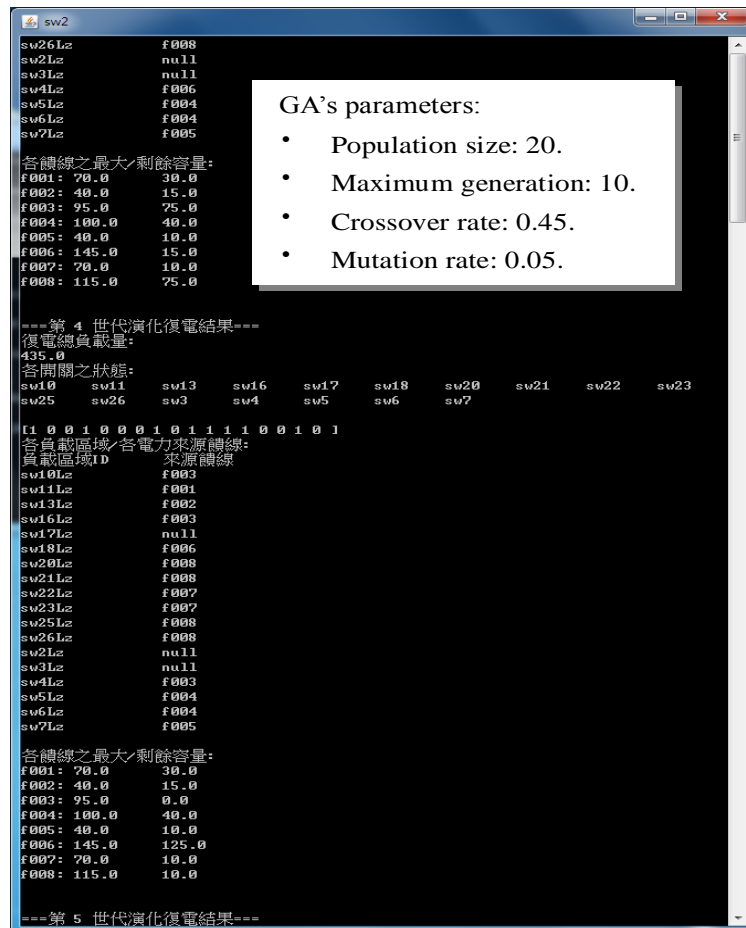


Figure 15. Evolution by the GA.

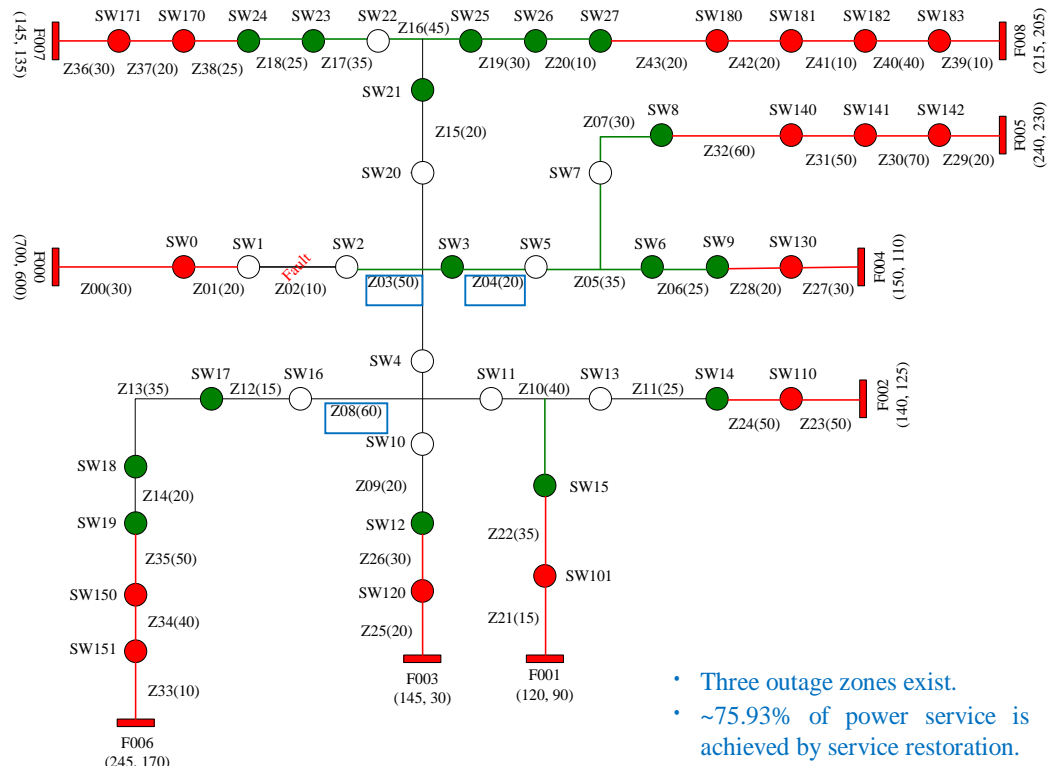


Figure 16. The finally-configured topology obtained by the proposed MAS for power service restoration in this simulation.

Conflict of Interest

The author(s) have not declared any conflict of interest.

REFERENCES

- Bellifemine FL, Caire G, Greenwood D (2007). Developing multi-agent systems with JADE. Wiley.
- Bratman ME (1999). Intention, plans, and practical reason. Harvard University Press: Cambridge, Massachusetts, USA.
- Chen CS, Lin CH, Tsai HY (2002). A rule-based expert system with colored Petri net models for distribution system service restoration. IEEE Trans. Power Syst. 17(4):1073-1080.
- Gen M, Cheng R (1997). Genetic algorithms and engineering design. Wiley-Interscience.
- Hsu YY, Huang HM, Kuo HC, Peng SK, Chang CW, Chang KJ, Yu HS, Chow CE, Kuo RT (1992). Distribution system service restoration using a heuristic search approach. IEEE Trans. Power Deliv. 7(2):734-740.
- Huang CM (2003). Multiobjective service restoration of distribution systems using fuzzy cause-effect networks. IEEE Trans. Power Syst. 18(2):867-874.
- Kumar Y, Das B, Sharma J (2008). Multiobjective, multiconstraint service restoration of electric power distribution system with priority customers. IEEE Trans. Power Deliv. 23(1):261-270.
- Nagata T, Sasaki H (2002). A multi-agent approach to power system restoration. IEEE Trans. Power Syst. 17(2):457-462.
- Nguyen CP, Flueck AJ (2012). Agent based restoration with distributed energy storage support in smart grids. IEEE Trans. Smart Grid. 3(2): 1029-1038.
- Pan YT, Tsai MS (2011). Application of BDI-based intelligent multi-agent systems for distribution system service restoration planning. European Trans. Elect. Power. 21(1):1783-1801.
- Parada V, Ferland JA, Arias M, Daniels K (2004). Optimization of electrical distribution feeders using simulated annealing. IEEE Trans. Power Deliv. 19(3):1135-1141.
- Solanki JM, Khushalani S, Schulz NN (2007). A multi-agent solution to distribution systems restoration. IEEE Trans. Power Syst. 22(3):1026-1034.
- Tsai MS (2008). Development of an object-oriented service restoration expert system with load variations. IEEE Trans. Power Syst. 23(1):219-225.
- Zidan A, El-Saadany EF (2012). A cooperative multiagent framework for self-healing mechanisms in distribution systems. IEEE Trans. Smart Grid. 3(3):1525-1539.