Multi-agent Scheme to Handle Flexible Loads on Low Voltage Distribution Grids

Niels Blaauwbroek
Electrical Energy Systems Group
Eindhoven University of Technology
Eindhoven, Netherlands
n.blaauwbroek@student.tue.nl

Diego Issicaba
INESC P&D Brasil & Labspot
Federal University of Santa Catarina
Florianópolis, Brazil
diego.issicaba@inescbrasil.org.br

João Abel Peças Lopes
INESC Porto & FEUP
University of Porto
Porto, Portugal
jpl@fe.up.pt

***Abstract*—The large scale integration of electric vehicles and distributed energy resources on low voltage grids might cause serious problems related to, for instance, under/over voltages and line overloading. In order to cope with these problems, this paper presents a multi agent system (MAS) developed to dynamically schedule flexible loads on low voltage grids, preventing operation limit violations. Since different geographical positions of the loads in the grid will cause a different impact on the grid, load flow calculations are used to indicate operation limit violations. The application uses a decentralized algorithm which ensures similar chances of being scheduled to the customer loads using a priority scheme. A case study is carried out on a 70-bus feeder where electric vehicle loads are scheduled to prevent under voltages, showing the applicability of the approach.**

Keywords—multi-agent systems, electric vehicles, active network management.

I. Introduction

Distribution networks nowadays undergo some changes in terms of design and operation. On one side, more and more distributed generation (DG) units are envisioned to be integrated in the network, while on the other side possibly large numbers of flexible loads will be connected, like plug-in electric vehicles (EV). The introduction of a large number of DG and EV can influence considerably the distribution grid operation. In case of EV, uncontrolled charging, in which the vehicle will immediately start charging until it is disconnected or fully charged (the so called dumb charging), can cause operation limit violations, like under voltage, line overloading, and increased power losses. In order to address the challenges of preventing the operation limits to be violated, this paper presents the usage of a multi agents system (MAS) for the dynamic scheduling of flexible loads using a priority based approach with a decentralized algorithm. For this accomplishment, Section II gives an overview of the MAS introduced in this paper. Section III presents in more detail the implementation of the MAS. Section IV presents the results of a case study using the presented system, while section V elaborates on technical issues related with the deployment of this MAS approach.

II. System Overview

Fig. 1 gives an overview of the control/management approach presented in this paper. This approach was conceived to deal with 3-phase with neutral low voltage distribution grids, having an unbalanced operation. In each node of this grid, non-flexible and flexible loads can be connected. Non-flexible loads are loads where the amount of power drawn from the grid cannot be adjusted. Conversely, flexible loads can adjust their power or shift the consumption over time. Flexible loads can also have mobile and non-mobile properties, i.e. they can move in the grid like EV, or always stay connected to the same node. Steady-state analysis of this 3-phase unbalanced distribution grid can be performed using an unbalanced load flow algorithm. In this way, operation conditions of the grid can be determined, like the voltage levels for each phase at each node as well as 3-phase power and current flows in each branch of the grid. With this information, grid operation can be checked regarding all kind of operational restrictions like under/over voltage, overloading or unbalanced operation.

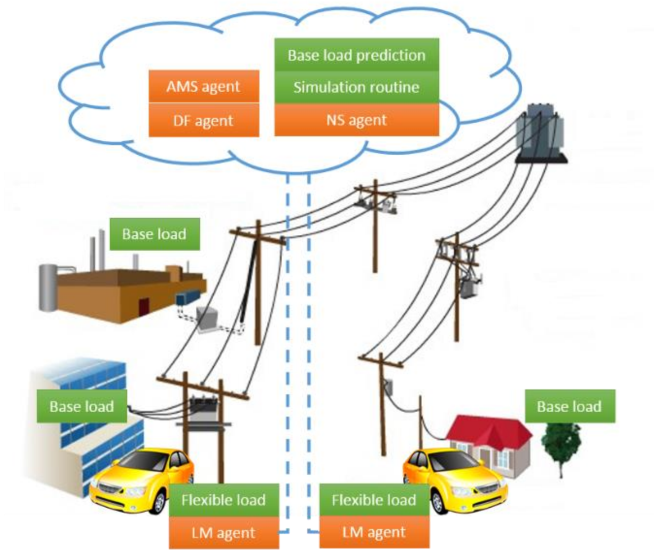

Fig. 1. System overview

In this approach, the distribution grid is managed by a MAS. Each of the flexible loads in the grid is represented by a Load Management (LM) agent. All the LM agents together will run a distributed algorithm in order to determine set points for the loads they represent. The management system makes use of time frames for which the set points will be applied. Each round in which the algorithm runs, the set points of the loads for the next time frame will be determined. Therefore, the algorithm will run ahead of the time frame in which the set points will be applied.

Besides the LM agent, in each distribution grid one or more Network Simulation (NS) agents can exist. The NS agent runs simulations to determine whether operation limits are violated and returns this information to the LM agent. Besides these agents, complying with the Foundation for Intelligent Physical Agents (FIPA) agent framework specification [1], the system will also have an Agent Management System (AMS) and a Directory Facilitator (DF).

All the agents used in the system are implemented in Java, on top of the Java Agent Development Framework (JADE) agent simulation platform [2]. JADE is completely compliant with the FIPA agent framework specifications.

A. MAS

The IEEE Power Engineering Society's (PES) Intelligent System Subcommittee has published contributions [3], [4] about the potential values of MAS for the power industry, as well as guidance and recommendations on how MAS can be designed and implemented in the power and energy sector. In [3], a MAS is defined as a system comprising two or more intelligent agents, where each agent will try to accomplish its own goal. As performed in our approach, to realize the overall intention of the system, multiple agents will be included with local goals corresponding to subparts of that intention. As stated in [4], the FIPA standards have become the *de facto* standards for developing MAS. For our implementation, the JADE was selected. This is a Java based tool for developing MAS compliant to the FIPA standards. In JADE, the AMS and DF are implemented as being agents themselves as well, running in the parent container of the platform. The AMS agent controls the platform and is the only agent who can create and destroy other agents or destroy containers. The DF agent provides a directory serving information about which agents are available on the platform.

B. Priority based scheme

Ahead of every time frame, all the agents together will decide which LM agents are allowed to draw power from the grid in the next time frame. Therefore, they will run a distributed algorithm, which will schedule the loads of all LM agents that take part in that round. Each LM agent will do a simulation request for the amount of power that it will draw from the grid in the next time frame. If the requests are feasible and not violating any operation limit constraints, the LM agents will schedule the power for the next time frame. If not, the LM agent will wait for the next round to do a new request. Improvements could be made to the algorithm in order to allow the LM agent for another request for the same time frame. The order in which the LM agents will do the request is determined by a priority scheme aimed at establishing a "fair" scheduling scheme. The LM agent with the highest priority will do the first request, followed by the other LM agents with a lower priority. Each round, the priorities of all LM agents will be updated. One may think of several schemes for updating the priority indicators. In this paper, a LM agent decreases its priority if it was able to draw power from the grid and increases its priority if it was not able to draw power from the grid.

When applying the priority scheme mentioned above, the LM agent that was able to draw power from the grid the lowest number of times, has the highest priority. The LM agent that was able to draw power from the grid the highest number of times has the lowest priority. Since the presented system is supposed to be fair for all LM agents taking part, the absolute differences between all priorities of the LM agents should stay within a certain bandwidth. If all loads were able to draw power from the grid a similar number of times, they will also have similar priorities. Note that for this reason, overflow or underflow of the priorities after a very long time should not be an issue, since they can wrap around and can be compared using serial number arithmetic.

C. Operation limit constraints

In principle, it is possible to apply operation limit constraints related to steady-state analysis. One may think of over or under voltage limitations, current limitations, power factor, preventing unbalanced operation or congestion and more. In our case, the system is designed to avoid under voltages.

D. IEEE Reliablility Test System

The IEEE Reliability Test System [5] presents a load model having hourly percentages of the peak load during a whole year. This data is applied to simulate the behavior of the system during a whole year. It is used for the base load forecast that the NS agent will take into account for simulating the network, as well as for the requests of the LM agents themselves.

III. DETAILED SYSTEM IMPLEMENTATION

This section describes the implementation of the management/control system. In our implementation, the components of the system overview of Fig. 1 colored in red (all the agents) are implemented in JADE. The components colored in green (distribution network, loads and simulation routine) are implemented in MATLAB. Between those two platforms, a communication link is set up in order to establish communication between the LM agents and the loads modeled in MATLAB and between the NS agent and the network simulation routine. In our case, we established a TCP/IP connection between each LM agent in JADE and its corresponding load modeled in MATLAB, as well as between the NS agent and the network simulation routine.

A. Load flow algorithm

The load flow routine used for the simulation of the grid is described in [6]. The load flow algorithm is based on back-forward sweep technique, considering not only 3-phase wires, but also the neutral wire and ground, taking into account unbalanced connected loads and component mutual impedances.

B. Distributed algorithm

All the LM agents together run a distributed algorithm in order to schedule as much loads as possible for the next time frame, but without violating the operation limits of the grid. The scheduling is subjected to the priorities of the different loads. Based on these priorities, the LM agents will determine the order in which they are allowed trying to schedule the corresponding loads for the next time frame. The agent with the highest priority will first try to schedule its load. Therefore, it will do a request to the NS agent. If the NS agent returns that the request is feasible, the requesting LM agent will schedule the corresponding load. If not feasible, it will decide not to schedule its load and make a new request for the next round. After this, the LM agent with the next lower priority will do a request, up to the LM agent with the lowest priority.

When joining the MAS, according to the FIPA standard, the LM agent needs to register with the AMS agent. If a LM agent wants to draw power from the grid and try to schedule its load for the next time frame, it also needs to register with the DF agent with its own unique agent ID number (AID). As long as it is registered there, it is known to be present and take part in the management algorithm. When it does not want to continue being managed, it needs to deregister from the DF before the distributed algorithm starts running for the next time frame. Otherwise it is expected to take part in the next time frame.

The diagram in Fig. 2 shows the steps taken by the LM agent in each simulation round. The algorithm incorporates the following steps:

1. In the first step, each LM agent contacts the DF agent in order to ask which other LM agents are registered and so which loads take part in the negotiation for this round. The DF agent will return a list containing those LM agents.

2. Each LM agent sorts this list of LM agents received from the DF agent in the order of their AID number. The two LM agents that turn out to have the lowest and the highest ID numbers will take the initiative to initialize the next step, while all the other LM agents will wait for this.

3. The LM agent with the lowest and highest ID numbers will initialize the exchange of the actual priorities. They will perform the initialization by sending a list with only one item, namely an object containing its AID and priority to the next LM agent in the sorted list of participating LM agents. The LM agent with the lowest AID number will send it to the LM agent that has the next higher AID number and the LM agent with the lowest AID number will send it to the LM agent that has the next lower AID number. Every LM agent that receives this list containing the objects with the AID's and priorities will append its own object with its own AID and priority to the list and pass it on to the next device. This process was created for the sake of establishing an ordering of interactions, since in the field this might depend dynamically to the local proximity of the agents.

   As soon as a LM agent has received two messages with the lists, they will merge the lists so that this single list will contain all objects with AID's and priorities from all other LM agents taking part in the procedure. This is not the case for the LM agents with the lowest and highest AID numbers, since they will only receive one list, which they will merge with their own object with AID and priority in order to also have the complete list. Each LM agent will sort the list in the order of the priorities of the LM agents. If more than one LM agent has the same priority, they will need to agree on a secondary scheme (e.g. randomly sorted or based on ID number), but they need to make sure they will all sort it in the same order as the other LM agents do.

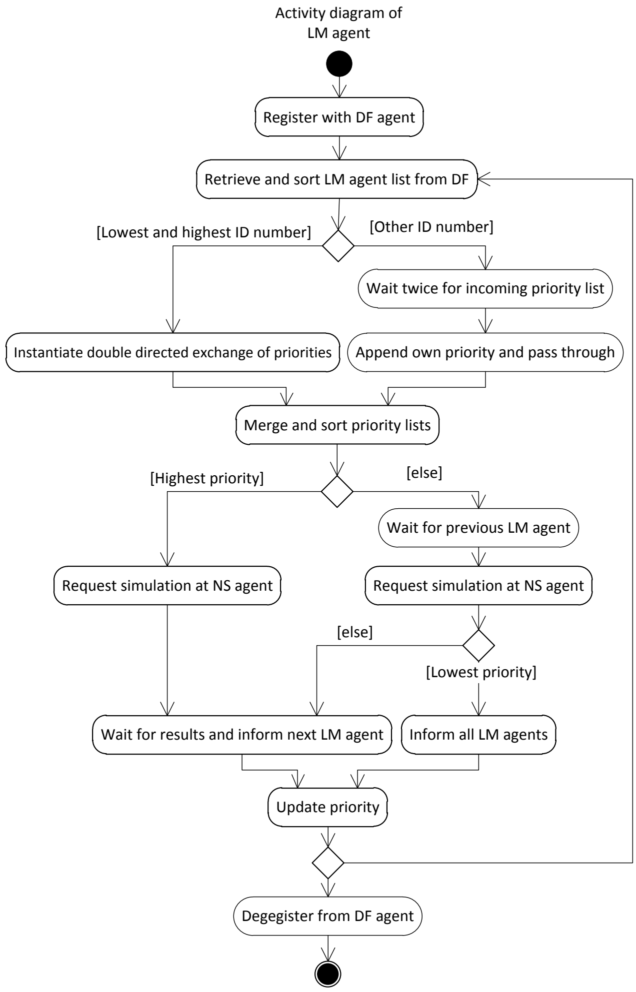

Fig. 2. Activity diagram of LM agent

4. The LM agent that is first in this list, i.e. has the highest priority, is allowed to do a request for drawing power in the next time slot to the NS agent. Therefore, it sends a request with its request and the bus to which it is connected to the NS agent.

5. The NS agent will run a power flow simulation of the grid. This simulation is based on the predicted power that will be drawn at each node in the next time frame. Together with this base load and the requests of the LM agent, the complete load flow will be calculated, after which the NS agent can determine whether there are any operation limit violations. These results will be returned to the requesting LM agent.

6. Once received, the LM agent will decide whether the request is a feasible request or not. If operation limits are violated, the LM agent will decide not to draw the requested power and make a new request in the next time frame. If there are no operation limit violations, the LM agent will schedule the request for the next time frame. The LM agent will update its priority according to the result of the simulation (i.e. increase its priority when it is not able to draw power and decrease its priority other ways).

7. After that, the next step for the LM agent is to inform the next LM agent in the priority list that it is its turn to do a request, as described in step 4. The NS agent will do the load flow calculations as described in step 5, based on the predicted load at each node, all previously loads that did a feasible request and the new request of the requesting load. Of course, the LM agents need to keep track of which loads have been scheduled for the next iteration, because all previously scheduled loads need to be incorporated in the simulation requested by the next LM agent. Therefore, when informing the next LM agent to do a request to the NS agent, the informing LM agent also needs to inform the next LM agent about which loads are scheduled until now. The next LM agent needs to include this information in its simulation request to the NS agent, in order to get the right simulation results. In order to achieve a simulation speedup, the NS agent could cache the results of this simulation, since putting the final node voltages of the previous simulation as the initial voltages of the next simulation can yield a dramatically simulation speedup.

8. After the last LM agent has finished its request to the NS agent, it will inform all other LM agents about this. Now, all LM agents have scheduled their power for the next time frame and they can return to step 1 in order to start the algorithm for the next round. If a device no longer wants to continue taking part in this management procedure, it needs to deregister from the DF agent.

C. Fairness

The system developed was designed to be as fair as possible for all participating LM agents, taking into account their geographical position in the grid. In this case, fairness means that each load after some time has been allowed to draw power from the grid approximately the same number of time frames as every other load, unless some constraints make this scheme unfeasible. This means that the system is fair in terms of time and not necessarily in terms of energy. This because each device can do a different request for the amount of power it wants to draw from the grid. Therefore, it is specifically useful for loads that might demand similar amounts of power, like charging electric vehicles. If not dealing with devices drawing similar amounts of power, a different priority scheme may be considered. In case it is expected that the requests of (a group of) LM agents will be accepted likely, one may think of doing a request for several agents together. If the combined request is not feasible, new combined requests need to be done for sub groups of the originally participating LM agents. This way, if the combined request of a group of LM agents is likely being feasible, it can drastically decrease the simulation time.

Since the priority scheme is supposed to be fair, it is expected that all the different priorities of the LM agents participating, will not have large deviations from each other. In other words, they are expected to be within a certain range from each other. This because if a certain LM agent has a significantly higher priority, it means it was able to draw power from the grid in significantly more time frames than other LM agents. In case the priority is significantly lower, is was able to draw power from the grid less often than other LM agents. So, the smaller the band of priorities at each moment in time, the more fair the system is.

D. Base load prediction

In order to be able of making simulations for a certain period in time, for example a day or week of the year or the whole year, the load model presented in [5] is used. Here, for each hour in a year, the hourly peak load in percentage of the annual peak load is listed. This information is used to “predict” the base load for each hour in a year. This prediction of the base load will be used by the NS agent in order to simulate the network in a simulation round, each time a LM agent requests for a simulation. So ahead of each simulation round, a prediction of the base load will be made for the next time frame, after which the NS agent will use this information.

E. Final simulation

Out of the scope of the scheduling process, it is interesting whether there are indeed no operation limit violations occurred anymore. Since the prediction of the base load will never be perfect, it may still be the case that the operation limit will slightly exceeded. In order to simulate this, a normalized distribution sampling factor is added to the power drawn by the base load after the scheduling process has been completed. It is applied to the flexible loads as well since the load will never draw exactly the amount of power as scheduled by its LM agent. Now, because after the scheduling process not every load will draw the exact amount of power as scheduled, due to the normalized distribution sampling factor, small deviations in the power flow will occur. Since the scheduling process is implemented in a simulation environment, one more final simulation of the distribution grid is needed to check for violations due to these deviations. Therefore, after all set points of the scheduling process have been applied for the upcoming time frame, the network will be simulated one more time, including the normalized distribution sampling factor. By applying the normalized distribution sampling factor, statistical data may be generated out of the simulation environment.

IV. SIMULATION RESULTS

In order to see how the system behaves, a case study was defined in which the management system is tested extensively for various scenarios on the 70-bus distribution network of Fig. 3.

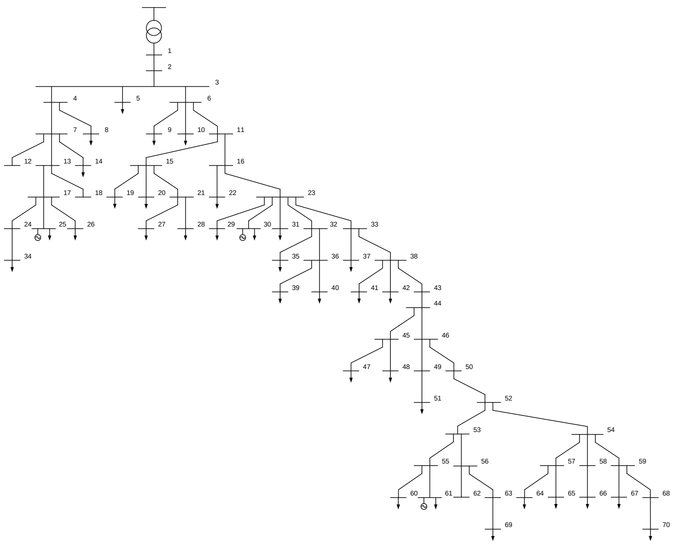

Fig. 3. 70-bus distribution grid

The goal of this case study was to prevent under voltages in the grid due to the presence of too many flexible loads. Therefore, the criterion for operation limit violation was set to be that none of the absolute individual phase voltages at every node are allowed to go below a certain value. For this study case, the value was set to be 0.95 pu.

A. Base load with 25 flexible loads

The first results were obtained by simulating the network with the base load and 25 flexible devices spread over the network. All LM agents were doing the same request for power consumption in each time frame. The simulation is for one week, having for each time frame a duration of one hour.

1) Prevention of violations

The graphs in Fig. 4 show the voltages occurrences within the distribution network during the simulation period of one week. It displays one of the phase voltages at node 70 during two different simulations. Node 70 is selected because it has the longest electrical distance to the slack node and therefore is expected to have the lowest phase voltage. The upper half of the figure shows a situation in which the MAS management is turned off (or running without constraints), whereas the lower half of the figure shows the situation in which the 0.95 pu phase voltage constrains are applied. In both simulations, the LM agents will do the same request for drawing power from the grid over time. The difference is that in the first simulation, the LM agent will always schedule the power, since there are no constrains. In the results it is clearly visible that in the constrained simulation the voltages never get below the operation limit of 0.95 pu, whereas in the unconstrained simulation this is certainly not the case. Note that this has the simple consequence that the loads of the constrained simulation will shift over time.

2) Fairness

Fig. 5 shows the result of a simulation for one week incorporating 25 different devices, each presented by a LM agent and all willing to draw the same amounts of power, spread over the 70-bus distribution grid. Besides those 25 devices, the grid has a base load that cannot be controlled.

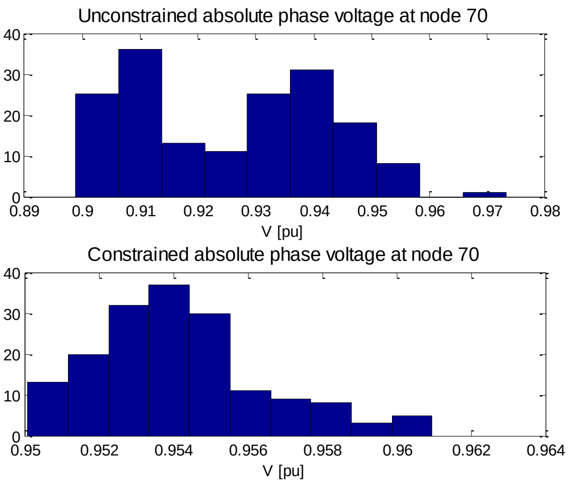

Fig. 4. Voltage occurrences in one week at node 70

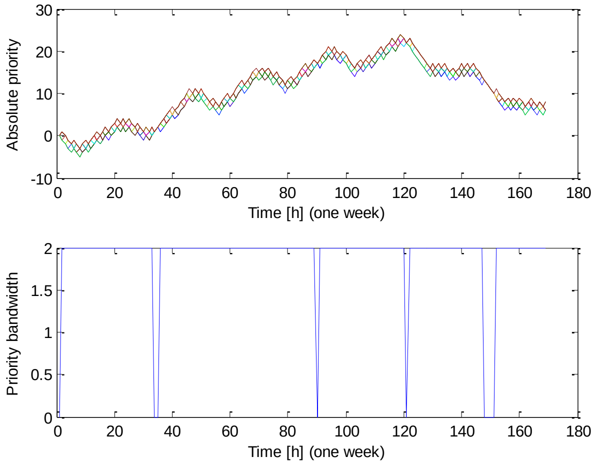

Fig. 5. Simulation of one week with 25 different loads

The upper graph in Fig. 5 shows the development of the absolute priorities of all 25 devices over time. Note that not all 25 devices are distinguishable because the lines are overlapping. During the first five work days of this particular week, on average the priorities rise a little bit. During the day, they go up, during the night they go down. This is easily explained by the fact that the base load during the day is higher than during the night, leaving more space for the flexible loads during the night. In the weekend on average the priorities go down, also due to the fact that the base load demand is lower during weekends than on work days.

From the priorities over time, one can derive the fairness of the system for this simulation. It can be seen that the difference in priority of the LM agent with maximum priority and minimum priority at each point in time is small. This priority bandwidth is set out against time in the lower half of Fig. 5. From this graph, one can see that the priority bandwidth at each point in time is at most two. This means that for each moment in time, a LM agent has at most one more time scheduled its load compared to the other LM agents, since the difference in priority will always be a factor of two.

B. Base load with 9 large loads

Another simulation incorporates 9 different loads, spread over the distribution grid, but each drawing a lot more power than in the previous simulation. The loads are spread in such a way, that there are relatively more loads downstream the network and relatively less loads close to the slack node. One load is connected directly to the node after the slack node.

1) Fairness

In Fig. 6, the results of this simulation are presented. Again, each LM agent is represented by a different line. The upper graph of Fig. 6 again shows the development of absolute priority against time, while the lower graph shows the development of relative priorities, having the offset removed. That is, the lowest priority is subtracted from the other priorities at each moment in time. Again, not all lines may be distinguishable because lines are overlapping.

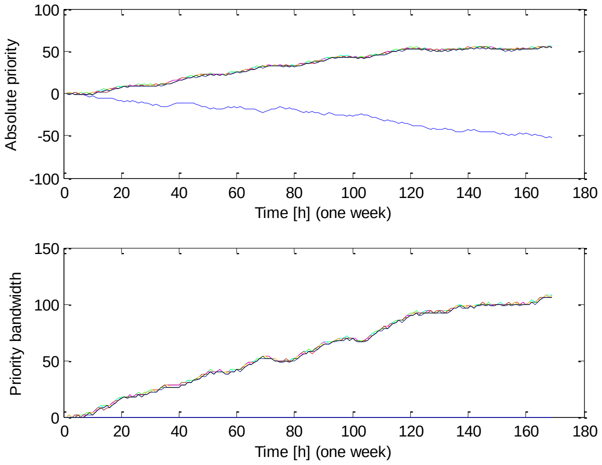

Fig. 6. Simulation of one week with 9 heavy loads

From these graphs, one can see that almost all priorities (except one) are rising most of the time. This denotes that for these LM agents, it was less often possible to schedule the loads without violating the voltage constraints. This is because of their large loads, who have a higher impact on the grid operation. It is visible that 8 priorities are within a bandwidth of 2 at each point in time, while the last LM agent has a priority significantly below the others and even going down instead of up (blue line). This LM agent corresponds to the load that is connected closer to the slack node. The impact of this load on the distribution grid is much lower than for a load that is connected to the end of the distribution grid. Hence, despite having the last chance to do a request to the NS agent, it was quite often still possible to schedule this load, without violations. This shows that loads will be scheduled when possible, and that the priority scheme achieves similar frequencies for each load of being scheduled, as long as the constraints allow it.

V. Final Remarks

In this paper, a MAS for managing flexible loading of 3-phase unbalanced low voltage distribution grids is presented, where flexible loads that do not cause operation limit violations are scheduled according to a priority scheme. One of the important aspects of the management system presented in this paper is the fact that the procedure runs in a distributed manner. For initializing the algorithm, the system relies on the DF agent to know which LM agents take part in the procedure. The DF agent serves like a local centralized service that keeps track of all participating LM agents, introducing a week point in the system. One of the solutions to overcome this problem could be the usage of a distributed hash table (DHT) for finding all the devices in the network that want to take part in the scheduling process. Because of the decentralized approach, communication acknowledgements need to be built in, in order to verify the correct functioning of each LM agent. When a certain LM agent does not properly respond when it is its turn to request a power flow simulation of the NS agent, the procedure will stop if there is no verification. Therefore, in case no acknowledgement is received, the previous LM agent needs to contact the next LM agent in the queue in order to let the process continue. Another drawback of the decentralized approach, is that relatively large amounts of data need to be exchanged between the LM agents.

When using the priority scheme as presented in this paper, it means that the system will be fair in terms of the frequency for which the LM agents are allowed to draw power from the grid. However, it is not fair in terms of the amount of power that each LM agent is drawing power from the grid, since every LM agent can do a different request for the amount of power it will draw. Therefore, this priority scheme is particularly interesting for devices that will all draw similar amounts of power, like EV charging. For other applications, other priority schemes may be considered, for example applying a correction factor to the priority in proportion to the amount of power.

VI. References

[1] "The Foundation for Intelligent Physical Agents," IEEE Foundation for Intelligent Physical Agents, [Online]. Available: fipa.org.

[2] "Java Agent DEvelopment Framework," Telecom Italia SpA, [Online]. Available: jade.tilab.com.

[3] S. D. J. McArthur, E. M. Davidson, V. M. Catterson, A. L. Dimeas, N. D. Hatziargyriou, F. Ponci and T. Funabashi, "Multi-Agent Systems for Power Engineering Applications—Part I: Concepts, Approaches, and Technical Challenges," *Power Systems, IEEE Transactions on,* vol. 22, no. 4, pp. 1743,1752, 2007.

[4] S. D. J. McArthur, E. M. Davidson, V. M. Catterson, A. L. Dimeas, N. D. Hatziargyriou, F. Ponci and T. Funabashi, "Multi-Agent Systems for Power Engineering Applications—Part II: Technologies, Standards, and Tools for Building Multi-agent Systems," *Power Systems, IEEE Transactions on,* vol. 22, no. 4, pp. 1753,1759, 2007.

[5] P. M. Subcommittee, "IEEE Reliability Test System," *Power Apparatus and Systems, IEEE Transactions on,* Vols. PAS-98, no. 6, pp. 2047,2054, 1979.

[6] R. M. Ciric, A. Padilha Feltrin and L. F. Ochoa, "Power flow in four-wire distribution networks-general approach," *Power Systems, IEEE Transactions on,* vol. 18, no. 4, pp. 1283,1290, 2003.

[7] A. L. Dimeas and N. D. Hatziargyriou, "Operation of a Multiagent System for Microgrid Control," *Power Systems, IEEE Transactions on,* vol. 20, no. 3, pp. 1447,1455, 2005.

[8] R. M. Ciric, L. F. Ochoa and A. Padilha, "Power flow in distribution networks with earth return," *International Journal of Electrical Power & Energy Systems,* vol. 26, no. 5, pp. 373,380, 2004.