Multi-stage Smart Grid Optimisation with a Multiagent System (Demonstration)

Christopher-Eyk Hrabia, Francisco Denis Pozo Pardo, Tobias Küster and Sahin Albayrak
Technische Universität Berlin, DAI-Labor
Ernst-Reuter-Platz 7
10587 Berlin, Germany
firstname.lastname@dai-labor.de

Categories and Subject Descriptors

I.2 [**Computing Methodologies**]: Distributed Artificial Intelligence—*Multiagent systems*

Keywords

multiagent systems; distributed information systems; optimisation; simulation; smart grids

1. INTRODUCTION

Increasing the usage of renewable energy and electromobility are high priority goals for several countries and organisations. Energy from renewable energy sources, like wind and solar, have disadvantages of being highly variable and volatile. Furthermore, the occurrence of renewable energy and energy demand is asynchronous. This aggravates any successful integration of renewable energy sources into today's energy infrastructures.

For this reason smart grid solutions became an appropriate approach for integrating renewable energy resources [2]. According to Gharavi [1] a smart grid can be defined as a electric system that uses information, two-way, cyber-secure communication technologies, and computational intelligence in an integrated fashion across electricity generation, transmission, substations, distribution and consumption to achieve a system that is clean, safe, secure, reliable, resilient, efficient, and sustainable.

In general smart grids are leading to the goal of more environment-friendly energy production. Moreover they allow the integration of electric vehicles as mobile energy storages that gives rise to less expensive transportation by using electromobility concepts at the same time. The problem domain was investigated in several applied research projects [4]. In this work we focus on the experience made in three German research projects from the Electromobility Showcase initiative Berlin-Brandenburg or "Schaufenster Elektromobilität Berlin-Brandenburg".

2. PROBLEM STATEMENT

The concerned ongoing projects "Micro Smart Grid EUREF", "Smart E-User" and "NaNu!" are all dealing with charging schedule optimisation in general, but have different constraints. These constraints are specific for the application scenarios, namely multiple car-sharing fleet companies as a floating service, dynamic routes with delivery vehicles and delivery trucks with exchangeable batteries.

All these projects are going to be integrated into real world test-beds with different smart grid configurations. Furthermore, the project "Micro Smart Grid EUREF" is going to support different operation modes that aim for different optimisation targets like increased CO_2 efficiency, reduced short-term energy costs or islanding. As a consequence, we are dealing with a large amount of different system setups.

In addition to the given optimisation problem and the specific application goals for each project, it is a major goal to deploy a common distributed agent-based solution. This solution is based on a common energy domain model [4] and takes into account required forecasts, resource allocation, integration with other Supervisory Control and Data Acquisition (SCADA) components and the demand scheduling.

3. SYSTEM ARCHITECTURE

As stated in the problem description, it was an important requirement to design a common solution, able to adapt to dynamic environments and varying operation modes. Therefore concepts such as modularity and versatility are highly present in every design decision. In addition to this, the optimisation module has to be placed in the context of larger software projects with several other depending and influencing modules. In this concern we decided to follow a multiagent approach. Firstly due to the fact that, after analysing the requirements of the projects, it was an intuitive process to split the different modules into roles or behaviour units, e.i. agent descriptions. Secondly available frameworks for multiagent development are providing the low-level infrastructure and allowing the designer to focus on the high abstraction level ones. Our system was developed with the *Java Intelligent Agent Componentware* (*JIAC*) [3], a Java-based multiagent development framework and runtime environment and consists of several agents, which communicate with each other using the JIAC messaging capabilities.

In the following we are focusing on the core component of our system the optimisation module. The motivation of splitting the optimisation process in several stages lies mainly on two objectives. First, to structure and simplify this complex process. Since we are dealing with the so-called multi-objective optimisation problem plus a wide spectrum

Appears in: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), Bordini, Elkind, Weiss, Yolum (eds.), May, 4–8, 2015, Istanbul, Turkey.*

of scenario descriptions, this level of complexity is difficult to handle. By having 4 different phases, the complexity of the whole problem can be divided into smaller and more specific domains. And second, to enable a flexible configuration and orchestration at runtime. Since the stages are specified via interfaces, they are absolutely implementation-independent, i.e., improvements or simply different approaches can be deployed in order to cope with a solution, required maybe by a different scenario definition.

These stages are specified with interfaces that are implemented by one or more agents for each stage. Thereby, some stages are mandatory and some are optional.

1. Scheduling order enhancement by using predictions or heuristics [optional]

2. Allocation of resources

3. Meta-heuristic optimisation of the schedule

4. Post optimisation for special requirements [optional]

First, the process is initiated with a scheduling request, containing information about the availability of the resources belonging to the correspondent smart grid. This scheduling request can be enhanced in the first stage by applying predictions. This step can be necessary in some scenarios because of missing or incomplete information about the bookings of electric-vehicles. We have encountered this problem in real-life integration, especially in the context of floating car-sharing services in combination with multiple operators. In order to deal with the missing booking information we have learned a model of the booking behaviour at our test site that enables long-term scheduling without any vehicle booking information. The learning process is using an artificial neural network with two hidden layers and was trained with historical data from 16 months with a resolution of 3 minutes.

The second stage allocates the resources. This means in particular the assignment of bookings to vehicles, batteries to vehicles and batteries to charging points. This is especially important in one of our projects that is exploring the application of exchangeable batteries for delivery trucks.

The third stage is the main component of the system and processes the actual optimisation algorithm to produce an optimised charging schedule. Our solution is based on evolution strategy and follows the work of Rechenberg [5]. Hence, the optimisation is based on the principles of mutation, recombination and selection to iteratively improve on existing solutions and to converge to a near optimal solution. In more detail, we apply a variant of evolution strategy and use one or more *populations* of charging schedules, from which a number of *parents* are randomly selected, recombined, and mutated. Resulting individuals are simulated and assessed. The best individuals are selected as input for the next generation until the quality converges.

The fourth stage can be used to integrate further post optimisations, which are adjusting the resulting schedule of stage 3 according to fairness requirements among competing car sharing services or integrating further optimisations for components like combined heating and power units.

The multiagent approach of this architecture allows to distribute and orchestrate all agents dynamically. A stage or agent can be added, removed, replaced or reconfigured during runtime at any time. Further, it is possible to compare different configuration against each other with running multiple configurations in parallel in order to determine the most suitable solution.

4. SIMULATION

The multiagent system is integrated into a simulation application. This application provides a web-based user interface that allows to select among different smart grid scenarios at different locations. Further, it is possible to configure the simulation period, executed simulation cycles and optimisation criteria with detailed weight configurations.

It is possible to run the simulation based on artificial or historical input data from our test-beds. The used data consists of weather forecasts, bookings and measured values.

After optimisation and simulation the results are graphically presented in a specialised view. The view allows to explore the single optimisation cycles that represent the repeated execution cycles in a dynamic changing environment, as well as an aggregated result for the whole period. In doing so the influence of imprecise forecasts and the ongoing adoption of the optimisation algorithm is visualised.

Acknowledgements

This work was partially funded by the *German government* under the funding reference numbers 16SBB005C (NaNu!), 16SBB016E (Micro Smart Grid EUREF) and 16SBB014A (Smart E-User).

REFERENCES

[1] H. Gharavi and R. Ghafurian. Scanning the issue: Smart grid: The electric energy system of the future. *Proceedings of the IEEE*, 99(6):917–921, June 2011.

[2] N. Hatziargyriou, H. Asano, R. Iravani, and C. Marnay. Microgrids. *IEEE Power and Energy Magazine*, 5(4):78–94, 2007.

[3] M. Lützenberger, T. Küster, T. Konnerth, A. Thiele, N. Masuch, A. Heßler, J. Keiser, M. Burkhardt, S. Kaiser, and S. Albayrak. JIAC V — A MAS framework for industrial applications (extended abstract). In T. Ito, C. Jonker, M. Gini, and O. Shehory, editors, *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems, Saint Paul, Minnesota, USA*, pages 1189–1190, 2013.

[4] M. Lützenberger, N. Masuch, T. Küster, J. Keiser, D. Freund, M. Voß, C.-E. Hrabia, D. Pozo, J. Fähndrich, F. Trollmann, and S. Albayrak. Towards a holistic approach for problems in the energy and mobility domain. *Procedia Computer Science*, 32(0):780–787, 2014.

[5] I. Rechenberg. *Evolutionsstrategie : Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Number 15 in Problemata. Frommann-Holzboog, Stuttgart-Bad Cannstatt, 1973.

[6] J. Tonn and S. Kaiser. ASGARD – a graphical monitoring tool for distributed agent infrastructures. In *Proceedings of 8th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS 2010)*, Salamanca, Spain, 2010.