View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Elsevier - Publisher Connector

Available online at www.sciencedirect.com

ScienceDirect

Procedia - Social and Behavioral Sciences 148 (2014) 153 – 160

Procedia
Social and Behavioral Sciences

ICSIM

Negotiation Models for the European Energy Market – Heletel Fast Prototype

N. Protogeros[a], C.G. Yakinthos[b],*

[a]*University of Macedonia, 156 Egnatia, Thessaloniki 54006, Greece*
[b]*Merchant Marine Academy of Macedonia, Nea Mihaniona 57004, Greece*

Abstract

For the purpose of the EU-DEEP project, a European co-funded project, which focuses on the European Energy market, a prototype was studied and developed for the management of resources that is based on transaction techniques. That is to say, the prototype examines all the possible cases and in order to make the decision, for the energy management system, which generator to use at a particular time depending from the result of negotiation with market terms using agent technologies between the energy managing system and the participating generator. The purpose of this study was to investigate the models of negotiation, the Software Agents with platforms as the JADE, as well as the requirements of the system and to develop a prototype and evaluate it.

© 2014 Elsevier Ltd. This is an open access article under the CC BY-NC-ND license
(http://creativecommons.org/licenses/by-nc-nd/3.0/).

Selection and peer-review under responsibility of the 2nd International Conference on Strategic Innovative Marketing.

Keywords: electronic auction; Power Systems; Energy Market

---

1. Agent technology - JADE

1.1. Agent Technology

If we have to give a definition to the word agent, we could define an agent as a software component that can run autonomously and accurately in order to carry out the commands that the user gave it. It is better to say that the

---

* Corresponding author. Tel.: +30 6944774577; fax: +2392031222.
E-mail address: cgyak@yahoo.com

1877-0428 © 2014 Elsevier Ltd. This is an open access article under the CC BY-NC-ND license
(http://creativecommons.org/licenses/by-nc-nd/3.0/).

Selection and peer-review under responsibility of the 2nd International Conference on Strategic Innovative Marketing.
doi:10.1016/j.sbspro.2014.07.029

definition agent works like an umbrella that covers a specific range of software agents that can accurately be enumerated and defined.

The field of applications that is being applied now is surprising: it ranges from simple application management user notes up to complicated system applications. The application of a statistic management system for communication and the program for visitors at a university are considered to be the best (CMU). To accomplish this, firstly the agents have direct access to sources of information so that they can clarify the fields of interest of the visitors, the name and organization of the visitors so that they can solve the unavoidable inconsistencies and vagueness. The system interacts at the end with the organizer and it aims at the confirmation or denial of the decision. This demonstration is considerably clever but we must pay attention that the intelligence of the application does not come from the event that the isolated agents work well but from the unexceptionable coordination of their actions, as well as working cooperatively the profit is bigger than that by working with any number of independent and isolated software agents .At this level are the applications where the technology of the agents actually demonstrates its capabilities.

1.2. Jade

Java Agent Development Framework is a sector of agent software fully developed in Java. It simplifies the development of multi–agent systems by a middleware which runs based on the Foundation of Intelligent Physical Agent specifications (FIPA) and with graphical tools that support the phases of correction and extension. The agent platform can be distributed to different machines, which do not necessarily run with the same operational system, and a remote GUI can control the configuration. It engages with all the issues for the implementation of the internal agents, but it remains independent from the rest of the application such as the transfer of messages from one agent to another, the codification, the parsing of data, and their life cycle. The configuration of JADE can even change during the time of enforcement transferring agents from one machine to another when it is requested (agent mobility). JADE is fully developed in Java language and its minimum system requirements is the JAVA 1.4 version (the JDK run time environment).

JADE software is freely distributed from TILAB, who is the owner of the copyright, in association with the LGPL rules for the open source software (Lesser General Public License Version 2)

2. Agent technology – JADE

In the system the LTS agent has the major role. This agent receives from the buyer an requirement for a certain amount of energy for a specified period within the next 24 hours. Besides the LTS agent is a SCADA agent who is the generator agent. The system can have different types of generators. According to the types of generators the producer has at his disposal. For the needs of this project we assume that the energy producer can have at his disposal energy that comes either from an atom generator that uses diesel, lignite, crude oil as fuel or from a wind generator.

The system is supported by a database that contains information about the weather for the next three days and in conclusion covers the next 24 hours we are interested in. This information comes from a forecast station, which is imported into the database of the system, which in turn makes it available to different units (agents) that need it. Also at the base there is information about different types of generators. Next we will go on to describe the base.

The system begins when the LTS agent receives a request for a certain amount of energy for a specified time interval. The direct action of the system is to look at the yellow pages with the available agents so that it can get o list of SCADA agents that are active and available for the supply of energy. Once it acquires the list it starts to ask all the active generator agents how much energy they can supply in the specified period and their production cost. The LTS agent waits for a reply from the SCADA agents and in the case that one of the agent's delays to reply then the agent is excluded from the procedure.

Each generator agent finally replies to the LTS the characteristics of the generator for each hour that is required in the time interval. The characteristics include the coefficients a, b, c of the generator, the maximum and minimum energy it can supply, the least working time, the least starting time, the cost of cold and warm start, the starting time for cold start and the total working time to the generator.

When the LTS collects all the agents replies it carries out an algorithm that calculates the best combination of generators that have to be used to cover all the available time. The algorithm is explained below.

Once the LTS agent reaches the combination energy it adds the total cost and the extra cost of the TSO/DSO who offers the transportation line. The transportation cost is a well-known curve, which keeps the ratio euro/kwh constant. The time taken for energy transfer is not related to its cost. The LTS agent calculates the total production cost and the energy transfer cost and then replies to the buyer. At this point the agent can add any extra cost to the price. It can include depreciation of equipment variable costs that ensures the most expensive pricing of energy at peak periods or it can follow an energy sales strategy so that it can acquire the best profit for the owner's account. The buyer can choose the best available price for energy from the different LTS agents that exist in the system.

Finally a reply is sent to the LTS agent, whether the energy will be bought from the LTS agent or if there are other alternative offers. If the reply is positive then the LTS agent informs the SCADA about the generators that are involved in the best-chosen solution so that they can supply the energy. In the case of a negative reply no further communication is required between the agents.

In the next stage the LTS and the other agents await until they get a new application for energy.

2.1. Description of the scada generators

Each generator replies to an LTS request separating the initial time to one-hour intervals. For each interval each agent sends to the LTS all the information that is required to calculate the solution and to find the best combination.

When the generator starts, each generator agent collects from the data base parameters and characteristics of the generator. Each generator has a name and a specified type. The generators types that the system supports are steam, diesel, natural gas, hydroelectric, wind and solar generators. The type plays a major role since the specification is defined by its parameters.

When the application for energy supply comes each generator consults the database to check how much energy it supplied from previous requests and to calculate the amount of energy available so that it can reply to the LTS. Once the generator sends a reply for the amount of energy and the parameters for each hour of the initial time period it awaits a command from the LTS whether it will supply energy or not. When the command is positive then the LTS specifies the exact amount of energy that each machine will supply. For each hour the final supply is made from SCADA agents that in turn inform the specified data base table.

The Pmin and Pmax energy that machines can supply each hour comes from a data base table that has two prices for each generator. In this way the maximum energy prices that a generator can supply can vary at the base for example some units are taken for maintenance, or if some units come from maintenance or the increase of production of a certain factory.

2.2. Description of the incorporated algorithm unit

The first function of the algorithm is to check how much the SCADA agent can fulfill the energy requests for the required time intervals. If the system can produce the required energy then the algorithm starts by calculating the best combination hence saving the unit incorporated problem.

One widely used method to the solution of the unit-incorporated problem is dynamic programming. We go on to describe dynamic programming.

One nexus of the table is specified by the coordinated pair (stage, situation). Stage is the time and situation is the different combinations of the units that work. There are 2N–1 different situations for each hour. If we use the symbols I and J for the unit combination and t for the time of day then the retrospective algorithm for back to end dynamic programming to the solution of the unit incorporated problems is:

$TC(t,I) = \min_j \{PC(t,I) + SC(t-1,J ; t, I) + TC(t-1,J)\}$

Where:

TC (t, I) is the minimum working cost for the first t working hours of the system, if at the time t we are at the situation I (unit combination).

PC (t, I) working cost at time t, if at that time, the unit combination that works is I.

SC (t-1, J; t, I) transfer cost from the situation (t-1, J) to the situation (t, I).

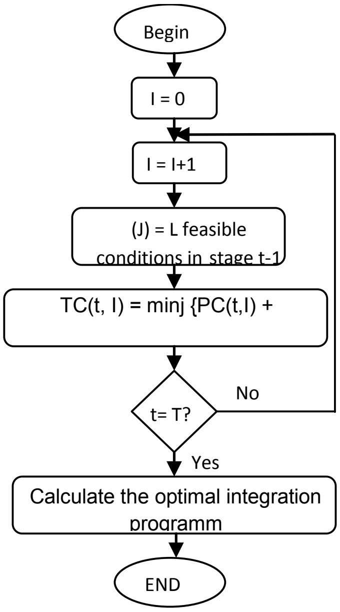

Fig. 1. Algorithm

The cost PC (t, I) is calculated by the economic situation of the load L (t) of the units that work at situation I. The unit's transfer cost SC includes the initial unit cost that is included at the time t and can be defined if the combinations J (t-1) and I (t) are known.

The economic distribution of the load that is required at cost PC (t, I), is calculated the repeated algorithm λ

An example of dynamic programming is show below. For the time internal t=2 from which 600 MW are required from the system, we calculate all situations that satisfy the condition Pmax > 600. There exist seven situations. For each situation, we calculate the transfer costs and the working cost TC (t, I) from all the available situations of the previous time interval. From the calculations we found, the best transfer is the one that gives the least working cost TC. In the diagram below the best transfer at situation 3 for t=2 is from situation 6 at t=1.

In continuance we calculate the best transfer for all the possible situations at the present (t=2) from the previous time (t=1) as shown by the fault. The best transfers for each interval are stored in a table that is used at the end of the method to find the best combination for the econometric satisfaction of the energy application.

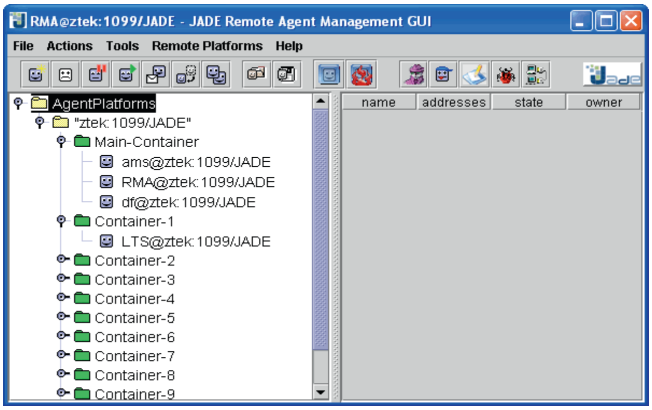

Fig. 2. Jade remote agent management GUI

The calculation of the most economic route starts at the end of the method. When all the best transfers have been calculated, the procedure finds the most economic situation. It begins from the economic situation that was calculated at the last time interval and goes on to follow the best transfers as calculated during the formation of the situation table. The working cost TC of the most economic situation of the last time interval consists of the production cost for the ideal route.

The amount of energy we will receive from each unit has already been calculated for each situation form the repeated algorithm λ.

2.3. Work description

The application is divided into different actions that have to be carried out. Firstly we have to open the JADE interface environment (Fig. 3). We also have to activate an LTS agent and an agent for each SCADA type.

The activation of the LTS agent activates the “application energy production” interface environment, which is shown in the diagram. Here anybody can set the time interval and the amount of energy as given to the LTS agent from a buyer or an agent who is part of a wider buyer system. The user can create simultaneous time intervals with amounts of energy for each hour and add them to the list of application energy programmes, in turn the LTS will read them simultaneously to SCADA so that LTS can receive a reply related with its availability and character. If the user-buyer does not want to create many different time intervals but is interested in a specific linear distributed time interval for the required energy he can fill the submission button.

With this button LTS starts the procedure as described above. After the calculations the LTS shows a message to the buyer with the total cost and he can either accept or deny the offer. During the calculation of the cost messages that concern the development of the stage will appear in the list at the bottom of the interface.

3. System evaluation

The system reacts differently in relation to the different parameters that are given to it. Our tests consist of different types of generators, the use of the steam generator or not, energy applications for big and small timeintervals, so that we can gain the maximum possible responses and in turn judge its effectiveness.

Table 1. Test Units

ID	a	b	c	Min operational time	Min Hold time	Internal Starting Cost	External Starting Cost	Hold Time for internal starting
1st	1000	16	0.00048	8	8	4500	9000	5
2nd	970	17.26	0.00031	8	8	5000	10000	5

3rd	700	16.6	0.002	5	5	550	1100	4
4th	680	16.5	0.00211	5	5	560	1120	4
5th	450	19.7	0.00398	6	6	900	1800	4
6th	370	22.26	0.00712	3	3	170	340	2
7th	480	27.74	0.00079	3	3	260	520	2
8th	660	25.92	0.00413	1	1	30	60	0
9th	665	27.27	0.00222	1	1	30	60	0
10th	670	27.79	0.00173	1	1	30	60	0

In the platform in which the system's agents were developed we did not consider their evaluation and exchange of messages necessary because JADE consists of well-known and developing framework that is set as a standard for agent development. The generators that have been evaluated are shown in Table 1 with the working time for each generator, which depends on the time interval that energy is required will be given for each example.

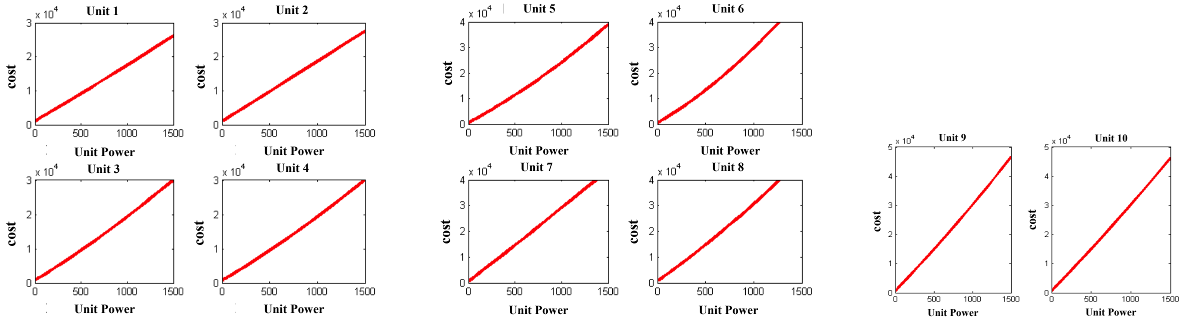

Fig. 3. (a) cheap unit 1 – 4 (b) medium unit 5 – 8 (c) Expensive with cheap starting cost 9-10

The first evaluation uses all the available units to receive a solution from the system. 2190 MW of energy for three hours were required from the LTS. The system divides the required time interval into hours and demands parts of the energy departments that are equipment to each interval, the energy is distributed linearly.

For the initial conditions the hours from the shutting down of the generator were represented with a negative sign and with a positive sign the working hours up to its shut down.

In this evaluation we used the application of energy shown below for the demand of energy in a 24 hours time interval. The system has the potential to give results for any time interval up to 24 hours as long as the load for each hour is given. In the evaluation the load of the Table 2 consists of the daily consumption for a typical load.

Table 2. Hourly load for the incorporation of 10 units

Time	Load (MW)	Time	Load (MW)
1st	700	13th	1400
2nd	750	14th	1300
3rd	850	15th	1200
4th	950	16th	1050
5th	1000	17th	1000
6th	1100	18th	1100
7th	1150	19th	1200
8th	1200	20th	1400
9th	1300	21st	1300
10th	1400	22nd	1100
11th	1450	23rd	900
12th	1500	24th	800

Figure 4 is a graphical representation of the load.

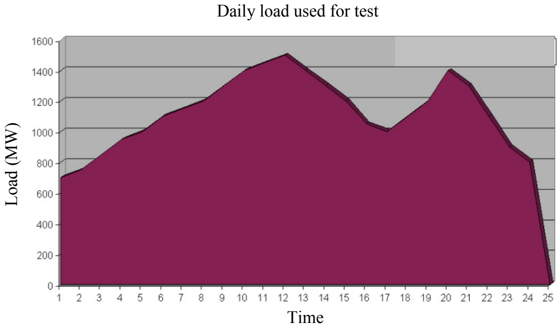

Fig. 4. Load and system evaluation

With the introduction of this load as the entrance to the dynamic programming algorithm that was used the result of the economic incorporation was calculated in 55 seconds. The time is considered to be suitable for this algorithm. The total cost that has been calculated by the dynamic programming algorithm is 553.190. in comparison to other algorithms it is considered quite good, as dynamic programming shows that it can be used for the incorporation of a small amount of units. With the increase of the units a problem arises with the large storage memory that is required by the algorithm. The number of possible situations of the algorithm can be decreased for more units, which in turn gives us a worse result as compared to the other economic incorporated methods such as genetic algorithms, and the Lagrange slitting method. Another disadvantage of the algorithm is that the incorporated units have to have the graphical form (f(x)=α+b*x+c*x^2) of the variable costs. In reality this does not occur for most units of energy production.

4. Conclusions

Generally the effectiveness of the algorithm is considered satisfiable. By playing initial values each time at the base so that a better control of the results was feasible and testing the purchase of different amounts of energy and at different times of the day but also at repeated time intervals with a total of 37 experiments the algorithm gave accurate results of the amount of purchased energy as to the least cost of the 32 experiments. At one stage 2 experiments gave non-feasible results. 3 experiments did not give the best cost. Below are the results.

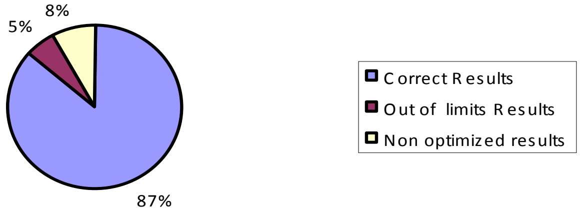

Fig. 5. Results

Far from the fact that there were a large amount of accurate results we are not in the situation to say that the algorithm does not need improvement. In conclusion we can use the algorithm by excluding one or more of the acceptances that were made.

References

Anastasios Bakirtzis – 'Power System Economic Operation'
http://benli.bcc.bilkent.edu.tr/~omer/research/dynprog.html
http://jade.tilab.com
http://www.catalase.com/dprog.htm
http://www.csc.liv.ac.uk/~ped/teachadmin/algor/dyprog.html
http://www.csse.monash.edu.au/
http://www.dei.gr
http://www.rae.gr
http://www.sce.carleton.ca/netmanage/docs/AgentsOverview/ao.html
Hyacinth S. Nwana - Software Agents: An Overview
I.G. Damousis, A.G. Bakirtzis, P.S. Dokopoulos – 'A Solution to the Unit Commitment problem using integer-coded GeneticAlgorithm'
I.G. Damousis, A.G. Bakirtzis, P.S. Dokopoulos – 'Network-Constrained Economic Dispatch using real-coded Genetic Algorithm'
Reticular Systems, Inc. – 'Using Intelligent Agents to Implement an Electronic Auction for Buying and Selling Electric Power'