Decentralized Transactive Energy Management Framework for Distribution Systems

Yerasyl Amanbek, Electrical and Electronic B.Eng.

Submitted in fulfilment of the requirements for the degree of Master of Science in Electrical and Computer Engineering

School of Engineering Department
Electrical and Computer Engineering
Nazarbayev University

53 Kabanbay Batyr Avenue,
Astana, Kazakhstan, 010000

Supervisor: Hema Satya Venkata Sivanand Kumar Nunna
Co-supervisor: Alexander Ruderman

2020

Abstract

Increased penetration of renewable energy-based plug-able and Distributed Energy Resources (DER) brings new challenges to distribution systems. To address these changes, the power system experts promote Transactive Energy (TE) models. Transactive Energy is a concept of providing control over energy exchange by integrating electricity markets and auction mechanisms. Numbers of studies on TE have shown a positive effect of TE management systems on distribution system efficiency, security, and reliability. However, it is still difficult to suggest TE model that will consider majority of distribution network constraints. The constraints include power allocation, voltage stability, network losses, congestion constraints and others. In the past Optimal Power Flow (OPF) method was used for distribution system management. Therefore, this thesis concentrates on modelling and simulation of feasible TE framework. In addition, more attention will be given for energy scheduling utilizing Distribution Locational Marginal Price (DLMP). The DLMP is key parameter that determine true cost of energy accounting topology, power losses, congestion, and other parameters. Therefore, this work will examine DLMP based Transactive Energy framework for distribution systems with enthusiastic or smart prosumers. The framework uses MAS as the basis on which the proposed Transactive Energy (TE) model, i.e. DLMP based TE Management System (DTEMS), is implemented. DTEMS uses a novel metric known as nodal earning component, which is determined by the optimal power flow (OPF) based smart auction mechanism, to schedule the TE transactions optimally among the stakeholders by alleviating the congestion in the distribution system. Based on the individual contributions to the congestion relief, DTEMS ranks the prosumers and loads as Most Valuable Players (MVPs) and assigns the energy trading price according to the category of the player. The effectiveness of the proposed TE model is verified by simulating the proposed DTEMS for a modified 33 bus radial distribution system fed with various plug-able energy resources, prosumers, and microgrids.

Acknowledgment

Firstly, I would like to express my uttermost gratitude towards my supervisors H. S. V. S. Kumar Nunna and Alexander Ruderman. It has been their supervision and direction throughout the duration of my studies which has allowed me to successfully complete this Master's program. I am appreciative for all the hours of discussion they have offered me, especially in the areas of power systems and power electronics.

Secondly, I would like to show my indebtedness to the administration staff at Nazarbayev University whose efforts a lot of the time go unnoticed. Last but not least, I would like to thank my family and friends.

Table of Contents

List of Abbreviations

DER	Distributed Energy Resources
DG	Distributed Generator
DLMP	Distribution Locational Marginal Price
DR	Demand Response
DSO	Distribution System Operator
ED	Economic Dispatch
ESS	Energy Storage Systems
EV	Electric Vehicles
FA	Flexible Agent
GA	Generation Agent
GBP	Grid Buying Price
GSP	Grid Selling Price
HEMS	Home Energy Management System
IEEE	Institute of Electrical and Electronic Engineering
IoT	Internet of Things
LA	Load Agent
LMP	Locational Marginal Price
MAS	Multi Agent System
MCP	Market Clearing Price
MVP	Most Valuable Player
NVP	Non-Valuable Player
OPF	Optimal Power Flow
PTDF	Power Transfer Distribution Factor
PV	Photo Voltaic
RBB	Risk Based Bidding
SCADA	Supervisory Control and Data Acquisition
SVP	Second Valuable Player
TE	Transactive Energy

TEMS	Transactive Energy Management System
ToU	Time of Use
TVP	Third Valuable Player
V2G	Vehicle to Grid

List of Figures

List of Tables

List of Publications

1. Y. Amanbek, Y. Tabarak, H. S. V. S. K. Nunna, and S. Doolla, "Decentralized transactive energy management system for distribution systems with prosumer microgrids," *2018 19th International Carpathian Control Conference (ICCC)*, Szilvasvarad, 2018, pp. 553-558 **(PUBLISHED, 7 citations in ieeexplore)**
2. H. S. V. S. Kumar Nunna, Y. Amanbek, B. Satuyeva and S. Doolla, "A decentralized transactive energy trading framework for prosumers in a microgrid cluster," *2019 IEEE PES GTD Grand International Conference and Exposition Asia (GTD Asia)*, Bangkok, Thailand, 2019, pp. 824-830. **(PUBLISHED, 1 citation in ieeexplore)**
3. A. Kalakova, Y. Amanbek, H. S. V. S. K. Nunna, P. K. Jamwal, and S. Doolla, "Feasibility Study of Energy Storage Systems for a Wind Farm: A Case of Ereymentau Region in Kazakhstan," *CPE-POWERENG 2020 14th International Conference on Compatibility, Power Electronics and Power Engineering*, Setubal, Portugal, pp. 1-6. **(ACCEPTED for Publication)**
4. Y. Amanbek, H. S. V. S. K. Nunna, and S. Doolla, "Distribution-LMP based Transactive Energy Management in Distribution Systems with Smart Prosumers - A Multi-Agent Approach", *IEEE transactions on sustainable energy* **(SUBMITTED for Publication)**

Chapter 1 - Introduction

1.1. General Information

Increased penetration of renewable energy-based plug-able and Distributed Energy Resources (DER) reshapes modern distribution systems. As a result, multiple microgrid systems are formed. These microgrids can effectively transact energy with each other to obtain positive benefits and enhance distribution system reliability, security, and resiliency. To enable these advancements in a multi microgrid system, the work proposes a novel blockchain-based TE framework that can regulate both the financial and physical layers. The main challenges faced by other TE models in the literature are incapacity to handle distribution network physical constraints such as line congestions, network losses, voltage stability, and power quality. Therefore, the work suggests the methodology of system regulation using distribution locational marginal price (DLMP) allocation for cyber-physical agents. The technique suggests providing additional incentives for agents that help to improve the condition of the distribution network and preventing transactions that cause problems to the system. Overall, the cyber-physical agents can obtain financial benefits by contributing to the betterment of the distribution power system. The distribution system can benefit in terms of energy exchange rates rises, power quality improvements, renewable energy promotion, consumers paying less for energy, and producers paid more. Such results can be achieved by more effective energy scheduling and reduction of operational costs. With the help of the proposed TE framework, the energy exchange volumes rise due to enabling a more flexible transactive market, which is not limited to fixed Time of Use (ToU) and feed-in tariffs. Power quality is improved because of the integration of the DLMP component that prevents transactions that causes voltage, congestion, or power loss problems. Renewable energy promotion can be enabled by providing extra incentives for a specific type of power. The producers of energy can sell energy for a price higher than Grid Buying Price (GBP), whereas consumers can buy energy for a price lower than Grid Selling Price (GSP). This feature will result in financial benefits for market participants, especially prosumers. Besides, to enable credible energy and financial transactions in the suggested TE framework for distribution networks, the blockchain technologies

integrated with smart contracts are investigated. The blockchain is a secure ledger for storing and managing data about transactions and is the vital enabling technology for the TE frame along with cyber-physical agents. More details are provided in subsequent chapters.

The idea behind building the Transactive Energy framework for distribution systems was inspired by the energy trading in electricity markets and modern technologies such as IoT and Blockchain. In the beginning, it was found that more residential buildings started to install small-scale PV panels and connect them with battery storage for local use. Later this business expanded and neighborhoods started to inject electricity to the main grid and obtain incentives in terms of feed-in tariffs. However, people reflected that feed-in tariffs were significantly lower than Grid Selling Price and started to think about the ways to transact energy between neighborhoods and getting maximum profit. The latest idea led to the foundation of Transactive Energy systems where energy buyers and sellers could freely trade energy between each other in a competitive market. Multiple articles related to the Transactive Energy Markets were written in the past decade. In the next subchapter, a detailed literature review on these articles will be provided.

1.2. Problem Definition

The rapid development of distribution systems both in terms of an increasing number of active units and infrastructure leads to the necessity of effective control and management of distribution systems. To address these challenges multiple attempts were made, where Transactive Energy was a general trend in the literature of the past five years that have the potential to address the problem. However, in the literature to the best of the author's knowledge, there is a lack of a complete framework that can integrate four key components: end units (players), physical topology (distribution network), market (regulation, scheduling), and communication (multi-agent systems). In the existing literature typically, there is a focus only on one or two components. Therefore, the primary goal of the work is to propose a complete state-of-the-art TE framework that is able to effectively integrate within four components.

Another problem is that it is often not clear what is the practical benefit of constructing the TE framework in real life. Often, the benefits of end-users from TE frameworks are explained in the literature, whereas, there is no clear benefit for DSO and providers of TE systems. These bring to

the debate of economic feasibility of TE systems. In the current work, the financial advantages of the proposed TE framework for all major stakeholders will be considered.

The last problem is related to the regulation of the market considering physical constraints. In the literature, one of the effective solutions is utilizing the DLMP parameter. The DLMPs are often associated with a fair price regulation mechanism for markets in distribution systems. However, to the best of the author's knowledge, there is no work that absolutely integrates DLMP into the TE framework. Therefore, this work will bring a novel TE framework with highly integrated components such as DLMPs, agents, market, and physical layer.

1.3. Aims and Objectives

The primary aim of the work is to develop an integrated TE framework for distribution systems. It is essential to consolidate four key enabling components of TE systems which are: end units (players), physical topology (distribution network), market (regulation, scheduling), and communication (multi-agent systems). To integrate four components of the TE framework certain steps were made. Firstly, the role of players should be determined, in this work players are cyber-physical (or virtual) agents that can participate in the TE market as energy buyers or sellers. Therefore, the strategy for each player agent should be developed and effective communication with other agents in the multi-agent system should be explained. Another objective is to clearly demonstrate the communication of agents when they are actively participating in the TE market. Since one of the challenges written in the previous subchapter was the integration and benefit of DSO and other stakeholders their role, communications, and benefit should also be explained. Overall, the players, market, and communication should be highly integrated. Furthermore, physical network constraints should also be integrated into the whole system. To integrate the physical system DLMP parameter have to be introduced. Unlike, in another system, DLMPs are directly associated with player agents and accounted in the TE market.

Despite, the integration of four key elements there are still some limitations that can be solved intuitively. The constrain defining one agent for each energy unit limits subjects that control multiple energy units, besides this constraint disable multiple energy units in one node. These constraints can be overcome by consolidated strategies and the addition of shadow nodes.

1.4. Novelty and Contribution

Most of the proposed TE systems in the literature [1]-[9] are based on decentralized power exchange and offers benefits for all market participants while utilizing the energy harvested from renewables efficiently. However, those TE systems are only feasible when the major parties like DSO and TE service providers altruistically refuse to have profit. Therefore, this work introduces a novel method by utilizing a special "nodal earn component" that supports building an effective TE framework. The nodal earn component indicates profit made by each node per one kWh of energy while participating in the electricity market. The node is usually supervised by one of the player agents: Load Agent (LA), Generator Agent (GA), and Flexible Agent (FA).

The work proposes a method of distributing this nodal earn component to satisfy major TE network stakeholders. Also, individual risk-based bidding strategies are developed for player agents. Besides, the concept of Value of Player (MVP) is proposed as an additional market regulation tool to discourage under-bidding and over-asking. This tool also supplements the idea of rescheduling the nodal earning component. The proposed TE system offers congestion management by utilizing DLMP metrics.

The major contributions of the work are listed below:

- Enhanced Multi-Agent-Based TE trading architecture with a high level of integration of the energy market to energy scheduling.
- A customized Risk-Based bidding strategy for trading agents such as consumers, DGs, and energy storage systems Transactive energy scheduling with congestion management, and loss reduction.
- DLMP based energy market with 3 cost components that encourage fair process, loss and congestion reduction in distributed systems.
- A novel TE profit (earning) management, called MVP based earning distribution, with the inclusion of the share of TE stakeholders.

1.5. Thesis Outline

The rest of the master thesis work is organized as follows. Chapter II proposes the literature

review on the points of issues related to the following work. Chapter III represents the details of the proposed DTEMS architecture and the designated roles of various agents. The energy market structure and the TE model followed also represented in Chapter III. The case study system is presented in Chapter IV, whilst simulation results and discussion are provided in Chapter V. Lastly, Chapter VI shows the insights of the overall work.

Chapter 2 – Literature Review

2.1. Agent Theory and Multi-Agent System

The trend related to the integration of the smart systems for simplification processes related to the system control, interaction of the separate systems, as well as providing interaction between the systems and giving the self-control for the different units leads to the improvement of the systems. The creation of the agents is used to solve problems related to the systems' configuration and moving in the direction of those improvements. The fact of creation agents for internal control of the systems solves challenges related to the collision between the control unit related to the centralized control of the system. On the other side, implementation of the agent-based systems limits information collection to one point or saying in particular to location limitation [10] – [12].

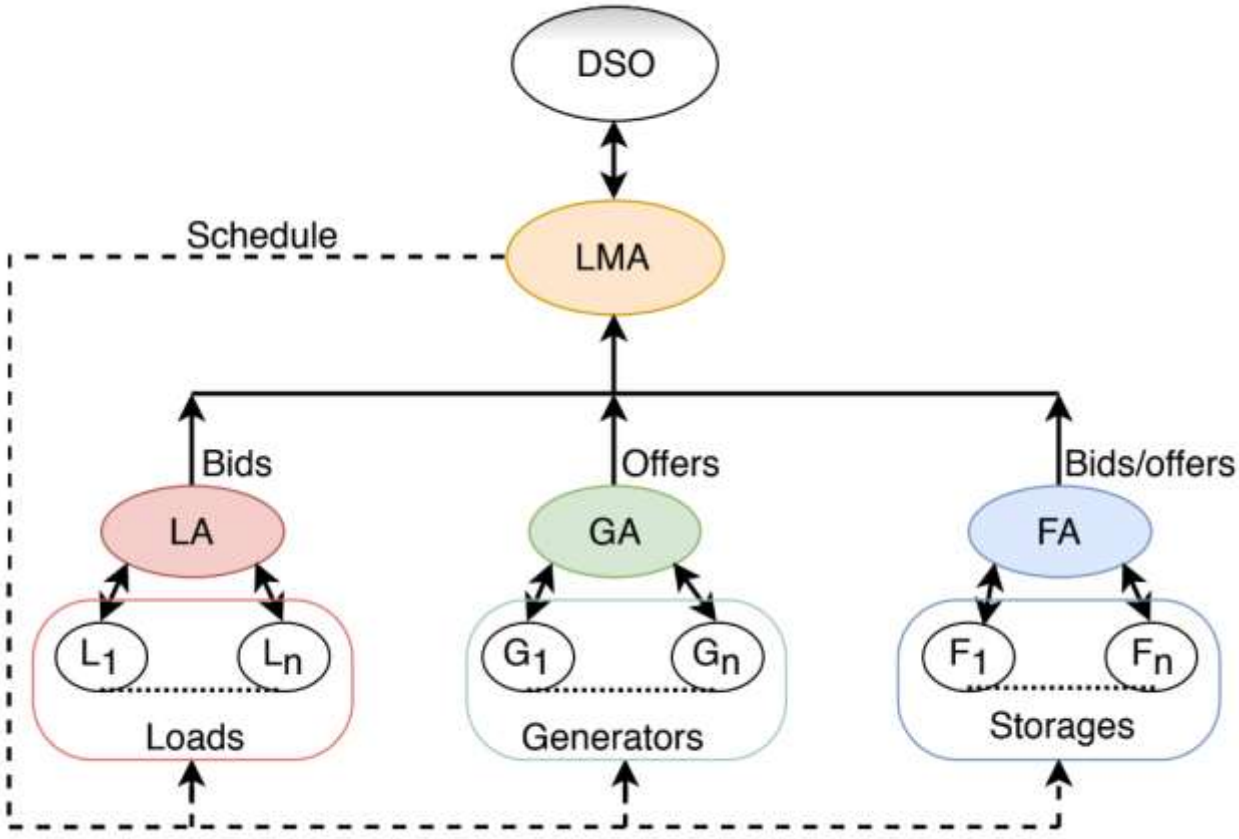

Figure 2.1.1. Multi-agent system's communication

The agent-based systems are giving the ability to transition to the human-free self-sustainable models, where agents are acting as controlling models and used to represent the system for the outside world. The application of the agents also includes an agent to agent communication, where agents will follow the assigned roles in the communication process. Following the instructions preassigned for the agents gives the ability to maintain the system in the working conditions [13]. The example of the agent's communication is represented in Figure 2.2.1, which represents the system represented in the following work. The FAs, Las and Gas represented in the figure are represented by the agents for the other parts on the system, where DSO and LMA also represented by their agents.

The system represented in Figure 2.1.1 depicts also called the multi-agent system, where different parts of the whole structure are represented by their agents, and the communication between the agents is allowed according to the preassigned structure. Authors in [14] state that the formation of the autonomous structure implies of implementation of the multi-agent system with desired coordination and cooperative connectivity between the agents. The implementation of the multi-agent system is also can be used for the autonomous action of the different agents with independent actions without combined work, as represented in [15]. In [16] the low of control includes the requirement of assigning agents to the appropriate units. All of the opportunities gained by the agents achieved for the local maximization of the unit, which as well can lead to system optimization and achieving the goal without any unexpected types of failures.

2.2. Transactive Energy systems

Increased penetration of renewable energy-based pluggable and Distributed Energy Resources (DER) causes distribution systems transformation. The pluggable resources include DERs, Electric Vehicles, microgrids, and prosumers. A promising solution represented as a coordination mechanism among all smart energy resources of the system is widely known as Transactive Energy (TE) [1]. TE is the new effective approach in providing control of energy flow and exchange concerning market-based standard values of the energy. TE improves system reliability, thus enables the optimal integration of green DERs using negotiation contracts among stakeholders and enhances the renewable energy hosting capacity of the distribution system [1]. However, the TE system must be reliable and transparent to all the players and stakeholders in the system. In the

current literature, various TE approaches have been reported [2]-[9], [17]. For instance, the authors of [2] have articulated the shortcomings of the existing methods and proposed a new framework to integrate TE optimally into a coupled natural gas and power system. For this, the overall system is modeled as an agent-based Virtual Power Plant which participates in day-ahead and real-time markets and regulates profit and energy imbalances.

Similarly, another article [3] proposes a framework for the day-ahead transactive market. In this framework, Distribution System Operator (DSO) participates in wholesale electricity market operations to trade the TE and interact with distribution level prosumers, including microgrids, DERs, load aggregators and demand response aggregators. Upon obtaining all the responses from local aggregations, DSO determines the distribution of DLMPs and payoffs. The work in [4] suggests a transactive energy framework for the optimal scheduling of DERs in a virtual power plant. The schedule of DERs obtained as an off-line solution is adjusted to minimize the real-time imbalance.

Reference [5] proposes a smart contract based on Ethereum blockchain. In this approach, the smart contracts enable energy producers to sell the excess of energy to the highest bidder through a Vickrey second price auction. Authors in [6] present a similar approach, where they introduce different contribution metrics to rank the prosumers by their energy production and consumption profiles. The TE is scheduled so that prosumers with higher metric gets the more substantial benefit. A similar work dealing with assigning priorities to the prosumers is presented in [7], where the priorities were defined based on the energy shortage and game theoretic strategy to maximize the profit for all parties.

It is important to note that the TE models strongly rely on a reliable communication network and require distributed computing environments such as Multi-Agent System (MAS). Power system experts widely discuss the application of MAS to TE market management systems and some significant approaches have been reported in [8] - [9], [17]. For example, transactive energy management systems (TEMS) to control flexible loads, EVs, generators, and energy storage systems were proposed in [8]. In which a heuristic iterative multi-agent method was used for TEMS, where the authority to make a decision is given to all customers. The work in [9] has proposed a hybrid model for energy scheduling for multiple microgrids. Similarly, in [17] a comprehensive agent-based energy management system for multi-microgrid networks is

presented, which aims at reducing the energy imbalances using DERs, such as Demand Response (DR) and Distributed Energy Storage Systems (DESSs). Another approach in [18] utilizes DLMP for transactive energy management in distribution systems.

Multiple attempts towards the establishment of TE markets were made by authors working in power engineering, information technology, and data science fields. Some works are focused on power infrastructure, other related to communications, also some works emphasize auctions and game theory. In the literature review, different TE approaches have been found for Research Proposal [19]-[24]. For example, the authors in [19] have proposed a structure to incorporate TE optimally into a coupled distribution system with multiple microgrids. The architecture includes inter-Microgrid market trading in day-ahead and continuous markets. However, the centralized approach used in the article indicates a single point of failure and potential malfunction in the continuous market clearing.

To address a single point of failure, reference [20] proposes a smart contract based on the decentralized blockchain platform. In this approach, the smart contracts enable energy producers to sell the excess of energy to the highest bidder through a Vickrey second price auction. Authors in [6] present a similar approach, where they introduce different contribution metrics to rank the prosumers by their energy production and consumption profiles. The TE is scheduled so that prosumers with higher parameter gets the more substantial benefit. However, the current work is limited to a small system with only a few players and does not consider prosumers like battery systems or electric vehicles.

Optimally scheduling prosumers like electric vehicles in the distribution network is a major challenge in TE model development. The work in [21] suggests a transactive energy framework for the optimal scheduling of Electric Vehicles (EVs). The objective is to quickly respond for power imbalances and adjust Electric Vehicle schedules accordingly. In contrast, reference [22] presents a multi-agent TE management system that schedules not only electric vehicles but renewable energy sources and batteries. The scheduling of these DERs is performed on the IEEE-37 bus feeder. However, these methods ignore some constraints related to flow congestion and power losses.

A similar approach was used by authors of reference [23], where they proposed a multi-agent based TE model for managing Virtual Power Plants (VPP). They have adopted the standard day

ahead market to estimate the VPP schedule and bidding in the market. Another article [24] proposes an optimization model with the integration of Distributed Energy Resources (DERs) into distribution systems with consideration of real-world network constraints such as losses and voltage violations. However, authors have adopted the traditional Economic Dispatch algorithm which is not as flexible as the auction model. There are also other articles about the TE environment, some of them talk about modeling TE systems.

The existing literature is lack of integrity regarding modeling feasible TE environment. The primary reason is that authors who write articles have different backgrounds: math, computer science, electrical engineering. Where some authors advance on modeling DER schedule optimization algorithms, others prefer to model on network optimization. To address this problem, some article has suggested multi-agent based TE models [19], [24]. In the multi-agent approach, different agents perform different tasks related to optimization. However, these articles are lack of novelty regarding modeling. Modeling was adopted from existing standard ED or OPF optimization models.

To address abovementioned limitations authors, suggest focusing on intelligent multi-agent based TE system with advanced modeling for different agents. In [6] authors have already adopted the multi-agent approach with auction models. However, some network constraints like power lose were ignored, and the internal optimization model for agents is lack of complexity. Therefore, there is still a demand for both a feasible and advanced TE model.

For a feasible TE model, authors suggest including the real-world constraints and provide proof-of-concept by simulating in existing distribution systems like IEEE 14, 30, 37, 118 bus systems. Moreover, the TE model must be fair for every stakeholder in the system. In addition, the proposed model should be resilient and scalable. The model should be able to handle the complexities brought by higher DERs penetration in distribution systems.

2.3. Bidding Strategies

Restructuring of the traditional power industry, which is started at the beginning of the 90s, was aimed to introduce transparent competition in the power markets. The creating of the mechanisms to trade openly gave the ability to improve economic efficiency and stabilize the electricity exchange. Ideally, the mechanism of the market used to manage and control energy trading and

tries to maximize the social welfare of the market participants by avoiding possible loopholes for cheating. To say in a more simple form, the well-designed market should be able to identify the monopolistic unit and try to concentrate on the small units of energy production. The barrier to enter the market can be categorized, where the large size of the trader can be not accepted [25].

From the side of the market participants, the main objective is proper identification of the bidding strategy for achieving the benefit of market participation. Many types of research propose different bidding strategies, where some of the states that investigation of the sealed bid auctions with the steady level pricing can be considered as an optimal strategy [26]. The first work, which raised the question of the bidding was the [27], where the analysis of the power generation units was provided for the proper identification of the optimal bid selection pattern. The authors of the work assumed that each generation unit can be considered as the block with the constant price for the energy generation, and the system for simulation was selected with the symmetrical demand variation. The authors identified the bidding model with modification of the random generation, where the bid prices were varied according to the dynamic parameters of the generation units, where the unit commitment costs also were taken into account. The work, represented in [28] proposed simple suboptimal bidding, where only two auction participants were taken into account. Unfortunately, after a short analysis, it is able to identify, that the number of market participants cannot be increased, and it is not possible to extend the proposed bidding and market architecture to the general market system with several participants.

Some works aimed to introduce the bidding based on the implementation of artificial intelligence, where the purpose was to ensure that the bidding history of the market participant will play into the hands. An example of this is the use of neural network technology, where the data bank is the main engine of progress. The first trading round for every market's participant, in this case, should be done as the random bidding, till the moment when the neural network will be able to generate a new price for bidding [29]. Authors in [30] criticized the idea of the implementation of the random bidding, even as the first steps of the market rounds, and proposed the two-level optimized bidding. The first level in the bidding optimization is the identification of the constraints of the system and limits of the possible bidding. The second level of bidding optimization is depending on the objectives of the units. Thus, in case if every unit wants to maximize the social welfare, the bid values are identified in the consolidated form, where the

estimation of the bids is made after consultation with other units in the system. Those systems main assumption is that every unit will bid honestly, without trying to deceive other market participants.

The bidding model for the units participated in the energy market can also have a form of the uniform price clearing, where the probability of winning in the auction depends is identified according to the previous MCP. The bidding, in this case, is still done as the random generation between the acceptable limits, while the main change is done in the limits directly. Thus, auction participants will not try to submit bids that will be lower than the previous MCP, as it will not result in any positive aspect for the participants [31].

In the case of the Continuous Double auction, where generation units submit their bis prices and load units submit their ask prices the financial intuition for the market participants should be taken into account for the development of the bidding model. The authors of [32] propose the use of the Zero Intelligence bidding model, where agents participating in the market will ignore the situation in the market and identify bid and ask prices as a randomly generated, based on the uniform distribution between the given limits. The proposed zero intelligence approach is extremely simple and does not provide high efficiency, however, it can be used as a profitable trading tactic. Modification of the Zero Intelligence bidding strategy is called a Zero Intelligence Plus bidding strategy and mainly aims to determine the more profitable bid and ask prices. The margins of the uniform distribution of the improved strategy depend mainly on the market conditions and can be modified using different learning functions [33].

Other bidding models for the energy auctions can include belief-based bidding or using fuzzy logic for identification of the bid and ask prices. Besides, authors in [33] proven that Zero Intelligence Plus based bidding is more profitable compared with the abovementioned techniques. Unfortunately, the Zero Intelligence Plus bidding does not consider any risk associated with the bidding and characteristics of the auction participants bid and ask prices do not include an eagerness to win, rather only participate in the trading. The strategy that solved this issue is called as the Risk-Based bidding (RBB) strategy [34]. The asks and bids in the RBB developed with considering the risk related to the agents, and selection depends on the past experience of the players. Results of [35] proven that the RBB can demonstrate the effectiveness of the bidding

model-based prices selection for the double auction, and moreover shows that RBB brings experimental balance for the trading process.

2.4. DLMP fundamentals

Distribution Locational Marginal Pricing concept usually refer to formulation of energy prices in distribution systems for both energy buyers (loads, demand responses, HVAC) and energy sellers (distributed generations, batteries, etc.). The price formulation considers not only energy costs (λ^r) but also other costs associated with losses (λ^{loss}), congestion (λ^{Con}), voltages (λ^{Volt}), and reactive power (λ^Q). The DLMPs are subject to continues change due to dependencies of its components to any change in distribution systems. E.g. when one node in distribution system starts to inject more energy, it affects the losses in the system and maybe cause of congestion in some lines. Therefore, DLMPs despite having potential to become advanced incentive mechanism for energy players in distribution systems are difficult to effectively track and utilize.

In the literature, authors from [36] propose DLMP decomposition into 3 sectors. Namely, Energy Cost Sector (ECS), Voltage Cost Sector (VCS), and Loss Cost Sector (LCS). They have analyzed economic effect of DLMPs on distributed resources in case of Demand Response and Voltage stabilization. The case study on 10 bus feeders was conducted, the primary finding was that DLMP based power markets can help in efficient operation of distribution systems. Where DRs are critical causes of reduction of DLMPs in network.

Another interesting work reported in [37] suggest DLMP computation not only at each bus, but for each phase. The rationale behind separate DLMPs for each phase comes from unbalanced nature of distribution systems. The authors proposed new method of phase DLMP decomposition method using three-phase current injection robust method (TCIM). The method initially uses power flow to obtain voltage values then optimal power flow is conducted to get DLMPs. To verify the results IEEE 13-bus feeder was selected. The results have shown that three-phase DLMPs can be beneficial for active market participants like distributed generations and demand responses.

The complete day-ahead market model that utilizes DLMPs for congestion management and voltage control were suggested by authors in [38]. Initially authors have developed market clearing and scheduling model using optimization function with certain constraints. Then in market pricing part DLMPs were obtained using sensitivity factors. Finally, they managed to decompose DLMP

into five costs components associated with: active power, reactive power, loss, voltage, and congestion. The proposed model was verified in IEEE 33 bus radial feeder, the effect of five components of DLMPs was demonstrated. From the results it can be noticed that DLMPs have capacity to alleviate congestions and provide voltage support.

There were some other articles [39]-[41] related to the role of DLMP in distribution systems. E.g. authors from [39] attempted to trace DLMP using Lagrange multipliers from three-phase AC optimal power flow problem, the results on 60-bus system have shown that DLMPs have potential to drive the cost of energy down and assist in formation of power marketplace in distribution level. In works [40] and [41] alternative approach of derivation of DLMPs were proposed. Where, in [40] Jain's fairness index based on DLMPs were utilized for regulation of social welfare in distribution systems. The model was simulated in IEEE 37-bus system and it was found that the model convergence without knowledge of generation or load parameters. Similarly, authors in [41] has also proposed alternative method of tracing DLMPs using quadratic programming. The results have shown that congestion in distribution system can be alleviated using the approach. Moreover, quadratic programming for decentralized aggregators optimization results converges with centralized DSO optimization, therefore, decentralized applications are possible.

Chapter 3 – Methodology

3.1. Multi-Agent System Architecture for TE system

Conventional SCADA systems alone cannot handle the complexity added by high penetration of pluggable DERs to the distribution system operation. To take complex decisions, Multi-Agent System (MAS) based approaches are widely used. A combination of multiple intelligent agents, which are operating interactively, becomes a powerful tool to improve the performance of complex systems [42]. Since decentralized systems can compute tasks in parallel and do not have a single point of failure. Moreover, the systems built using MAS are independent, highly reactive, pro-active, and scalable. The proposed DTEMS also uses MAS as the underlying architecture over which the TE model is implemented.

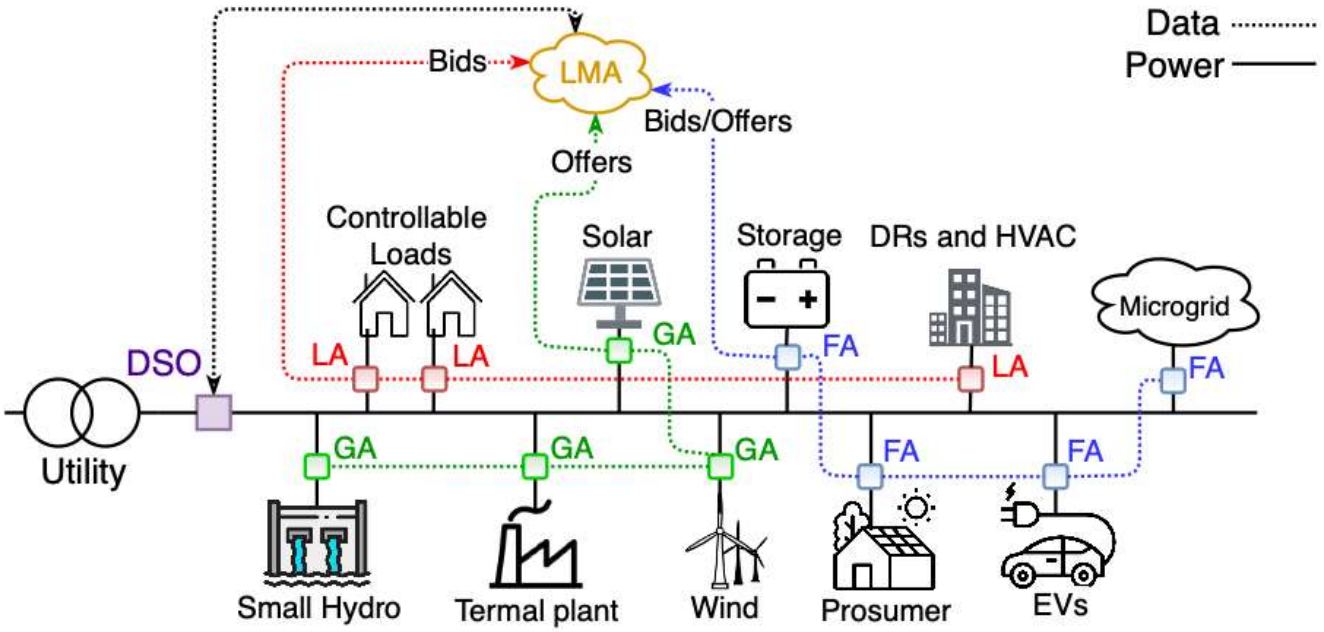

Figure 3.1.1. Multi-agent based transactive energy management framework

The agent architecture of the DTEMS is presented in Figure 3.1.1. In this architecture, the pluggable resources including prosumers are delegated and controlled by individual agents. These agents target to achieve specific goals set by the owners, say prosumers and DERs. Therefore, as shown in Fig. 1 the distinct loads, generators and storage units are represented by LAs, GAs, and

FAs respectively. These agents broadcast their bidding strategies to TE Market Agent (TEMA), which is accountable for scheduling the TE in the system. TEMA consolidates the received data and passes to the Distribution System Operator (DSO) to obtain required DLMPs.

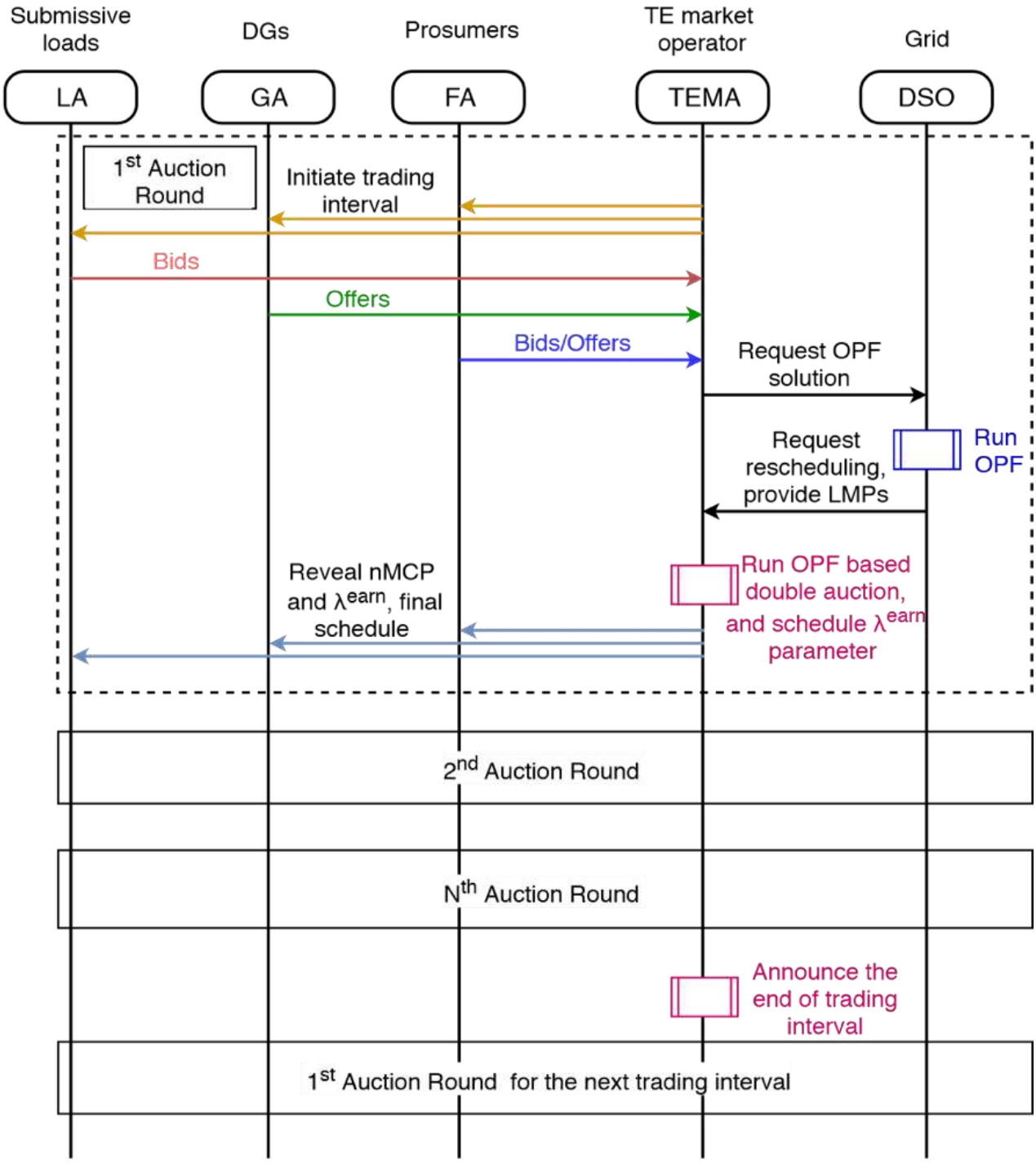

Figure 3.1.2. Data exchange layout for agents in TE system

DSO determines DLMP value for each node by solving the Optimal Power Flow (OPF) described in detail in the latter section. Upon on solving the OPF, the congestion (λ^{Con}) and loss (λ^{loss}) components of DLMP are determined and shared with TEMA including the nodal DLMPs. Upon receiving the data from DSO, TEMA implements the proposed TE mechanism in which it organizes the energy auction market using the bidding strategies collected from LAs, GAs, and FAs, and determines the TE trading contracts. Figure 3.1.2 depicts the information exchange process in the proposed DTEMS. It is worthy to note that all communication occurs through TEMA.

The latter parts of the section detail the roles of the agents and their rational bidding strategies to interact with the TE market.

3.1.1. Load Agent

LAs act as retailers or aggregators for the end-users with controllable low priority loads and without on-site generation. In general, there can be multiple LAs representing end-users across the distribution system. It is assumed that the end-users have Home Energy Management System (HEMS), such as the system proposed in [44], to coordinate with LAs for effective integration and trading of DR (NegaWatts). HEMS of the corresponding end-user executes the load operation commands directed by LAs. The end-users submit the information about their flexible load's information to LA via HEMS.

Upon registering with an LA, the HEMS submits the sub-hourly load profile of the premises along with the flexible load information set by the owner. This information includes power ratings, operating time, target time before which the device must be operated. Using this information, LAs clusters the loads into ($P_c(i,t)$), low priority or flexible loads $P_f(i,t)$), and least priority or super flexible loads $P_{sf}(i,t)$). Therefore, the total amount of power $P_k(i,t)$) required by LA is given by Equation (3.1.1).

$$P_k(t) = \sum_{i=1}^{N} P_k(i,t), \quad k\epsilon\{c,f,sf\} \tag{3.1.1}$$

where, N is the number of loads registered with the LA. After identifying the overall power requirement, LAs determine the bids for each load group by following a bidding strategy called

risk-based bidding (RBB). The bids for each load category have different levels of risk, i.e. the bid for high priority loads carries low risk whereas the bid for the least priority load group carries high risk.

3.1.2. Generator Agent

In the proposed DTEMS, GAs delegate the generators as energy sellers in the TE market. Based on the data provided by the Distributed Generator (DG) owners, GAs develop energy offering strategies. In the case of conventional generators, the energy generation is a quadratic function as represented in Equation (3.1.2).

$$C_i(t, P_g) = A_{i,t} * P_g^2 + B_{i,t} * P^g + C_{i,t} \quad \forall i \epsilon \{1, 2, \dots, N_{DG}\} \tag{3.1.2}$$

where, $C_i(t, P_g)$ is the cost of energy generation by DG in (¢/kWh) during the market interval t and $A_{i,t}$, $B_{i,t}$, and $C_{i,t}$ are the corresponding coefficients of the cost model. GAs convert the cost function into piece-wise-linear form and submit data as (E_j, ask_j) pairs to TEMA at the beginning of each trading interval. In general, the energy offers may differ from the incremental costs of the generators as DGs are profit-motivated and the model of the market followed by DTEMS is a stable energy price market. Therefore, GAs change offers using a strategy that maximizes the benefit of DGs in the market. Similar to LAs, GAs are assumed to follow the Risk-Based Bidding (RBB) strategy explained in subsection 3.1.4 to choose and revise the asks progressively in the proposed TE market.

3.1.3. Flexible Agent

FAs represent the pluggable resources that can draw the power from the distributed system or feed the power into the system. Therefore, these agents express prosumers in the TE market. The prosumers include end-users with on-site energy generation, Building to Grid (B2G), Distributed Energy Storage Systems, such as V2G enabled Electric Vehicle (EV2G), and smart microgrids. Prosumers such as EVs was found to be highly effective for DR programs in energy markets [44], and transactive systems [45]. In case of supplying energy to the grid, prosumers act as the energy sources and the corresponding FAs takes the role of a GA and calculates the energy offers based on the incremental costs provided by the end-users or the cost of accumulated energy in case of storage systems.

In the case of drawing energy from the system, the FAs take the role of the LAs. If the prosumers have flexible loads, then accordingly FAs choose the priorities and bids as described in subsection 3.1.1 of Chapter 3. In the absence of flexible loads, the whole load demand is assigned high priority and place in the market with the corresponding bid. In both cases, FAs follow the risk-based bidding strategy described in subsection 3.1.4 of Chapter 3 to quantify the bids.

3.1.4. Risk Based Bidding

The TE trading agents participate in continuous double auctions conducted by TEMA and use RBB as a bidding strategy. This strategy allows agents to place bids according to the degree of risk. In this strategy, player agents do risk-return tradeoff. Players that are looking for higher momentum profits has a lower probability of winning transaction. This strategy is determined by target price τ, which is derived from the risk model and price estimate $p*$.

Each TE trading agent that participates in TEMA has a limit price l_{ik} that indicates the maximum price the buyer is willing to pay or c_{jk} minimum price seller agrees to sell for. The first auction round starts when all agents submit their bid and ask prices. When the bid of the buyer is higher than the ask of a seller the transaction pairs are formed. In every auction, round buyers and sellers submit bids according to Equations (3.1.3) and (3.1.4) respectively.

$$bid_i = \begin{cases} o_b + (\min\{l_{ik}, o_a\} - o_b)/\eta & if\ 1st\ round \\ o_b + (\tau - o_b)/\eta & otherwise \end{cases} \tag{3.1.3}$$

$$ask_j = \begin{cases} o_b - (o_b - \max\{c_{jk}, o_b\})/\eta & if\ 1st\ round \\ o_a + (o_a - \tau)/\eta & otherwise \end{cases} \tag{3.1.4}$$

where $\eta \in [1, \infty)$ is a constant, that determines convergence of bids toward transaction price and τ is the target price.

In other auction rounds, LAs, GAs, and FAs estimate their target price τ for the loads based on risk - *r(t)* and eagerness - *Eag*. The current risk factor for the next auction round *r(t+1)* is estimated by Equation (3.1.5).

$$r(t+1) = r(t) + Eag * (\delta(t) - r(t)) \quad \forall i \epsilon \{1, 2, \dots, N_{DG}\} \tag{3.1.5}$$

where, $Eag \in (0, 1)$ is an eagerness to secure round, and δ(t) is the desired risk factor, which found by Equation (3.1.6).

$$\delta(t) = (1 + \Delta)\rho, \quad \Delta = (-k, k) \tag{3.1.6}$$

where, ρ is risk factor for the last bid, and k is the step size.

In case when a player fails to win transactions, he becomes more eager and rises a bid to secure the chance of winning in the next transactions. The proposed eagerness model, represented in the Equation (3.1.7), is subject to current trading interval t_i, deadline when the player must finalize bid/ask t_D, and operation time for the load/generator t_{op}.

$$Eag = EF * (t_i/(t_D - t_{op} + 1)) \tag{3.1.7}$$

where, $EF \in (0, 1)$ is an eagerness factor indicating traders' bias towards eagerness. Lower priority loads can rise priority when load switches the state to a higher priority as the deadline approaches. E.g. low priority load - a washing machine that has to run for 3 hours in any interval from 1:00 pm to 11:00 pm, does raise its priority when a deadline (8:00 pm) is approaching. If the LA did not manage to win auction rounds for any period from 1:00 pm to 7:00 pm it automatically rises priority of the load in auction rounds from 8:00 pm to 11:00 pm interval. The sample bidding strategy for 3 categories of the load is provided in Table 3.1.

Table 3.1. Bidding strategy for LAs

Load	Bid in 1st round	Risk	Eagerness Factor	Priority Rise
C1	High	Low	High	Impossible
C2	Medium	Medium	Depends on load	Possible
C2	Low	High	Depends on load	Possible

Target price τ is estimated based on the risk model, where the estimated price p∗ is found according to the moving average method with respect to the history of past transactions. More details about the moving average method are presented in [46]. Target price for buyers' adjustments reflecting past sales is given in Equation (3.1.8).

$$\tau = \begin{cases} p^* * (1 - r * e^{\theta(r-1)}) & if\ r \in (0,1) \\ (l_{ik} - p^*) * (1 - (r+1) * e^{rk}) + p^* & if\ r \in (-1,0) \end{cases} \tag{3.1.8}$$

where $k = (p^* * e^{-\theta})/(l_{ik} - p^*) - 1$

Target price for sellers is given in Equation (3.1.9).

$$\tau = \begin{cases} p^* * (a^{max} - p^*) r * e^{\theta(r-1)} & if\ r \in (0,1) \\ p^* + (p^* - c_{jk}) r * e^{k(r+1)} & if\ r \in (-1,0) \end{cases} \quad (3.1.9)$$

where $k = \log[(a^{max} - p^*)/(p^* - c_{jk})] - \theta$, and $\theta \in [-1, \infty)$ is function indicating the rate of change concerning risk. A lower value of θ means higher the cost of the gradient

3.2. Transactive Energy Model

This section presents the novel TE scheduling method with the congestion management model. It also includes DLMP based auction algorithm for scheduling TE and application of nodal earning component.

3.2.1. Problem formulation

The primary objective suggested to follow by DSO for TE scheduling is to reduce costs by minimizing losses, maintaining power balance, and avoiding congestions and voltage drops. In addition, the network topology should be operating with a radial topology. The objective function is given by Equation (3.2.1).

$$F_{obj} = \min \sum_{i=1}^{N} C_i * P_{gi} \quad (3.2.1)$$

The constraints for the objectives are provided from (3.2.2) - (3.2.11).

The system balance constraints:

$$\sum_{i=1}^{N_s} P_i^D - \sum_{k=1}^{N_b} P_k^L - P_{loss} = 0 \quad (3.2.2)$$

Line limits constraints:

$$\sum_{i=1}^{N} S_{ki} * P_i \leq |PL_l| \quad (3.2.3)$$

The GA capacity constraints:

$$P_{i,min}^{GA} \leq P_i^{GA} \leq P_{i,max}^{GA} \quad (3.2.4)$$

The LA demand response constraints:

$$P_{i,min}^{LA} \leq P_i^{LA} \leq P_{i,max}^{LA} \tag{3.2.5}$$

The FA operation constraints:

$$0 \leq P_{i,t}^{FA,+} \leq P_{i,max}^{FA,c} \tag{3.2.6}$$

$$0 \leq P_{i,t}^{FA,-} \leq P_{i,max}^{FA,d} \tag{3.2.7}$$

$$E_{i,t} = E_{i,t-1} + \eta_i^c P_{i,t}^{FA,+} - (1/\eta_i^d) P_{i,t}^{FA,-} \tag{3.2.8}$$

$$E_i^R * SOC_{i,min} \leq E_{i,t} \leq E_i^R * SOC_{i,min} \tag{3.2.9}$$

$$E_{i,t=T} = E_{i,t=0} \tag{3.2.10}$$

The voltage constraint:

$$V_i^{min} \leq V_i \leq V_i^{max} \tag{3.2.11}$$

For TE scheduling it is necessary to ensure that physical constraints are not violated. Moreover, it is practical to support nodes that help to reduce losses and congestions in the network. Therefore, in this work, DLMP components derived from objective function outcomes are integrated into the auction mechanism. More details are provided in the following sections.

3.2.2. Transactive Energy Scheduling

TEMA collects strategies from LA, GA, and FA to conduct the electricity market by negotiating with DSO regarding nodal prices and running a double-sided auction.

For the proposed TE scheduling model the first step is DSO conducting Economic Dispatch after receiving LA, GA, and FA strategies from TEMA. After this step, the network is ready to respond to potential contingencies by utilizing obtained DLMP metrics by rescheduling LA, GA, and FA strategies. In the case of an uncongested network, the energy dispatch remains unaltered. In the case of congestion, the proposed TE model deploys demand response through the rescheduling of resources of LAs, GAs, and FAs.

3.2.3. DLMP based market formulation

From OPF results conducted by DSO Locational Marginal Prices for each node in the network can be derived. Locational Marginal Price modeling is used for economic scheduling of energy considering congestions and losses in a distribution network. When DSO identifies congestion in a system, it can request TEMA to deploy the DR program by rescheduling LA, GA, and FA. Therefore, energy for these agents is scheduled in a way to reduce congestions in lines according to their Power Transfer Distribution Factor (PTDF). PTDF identifies critical nodes that can assist in relieving congestion in the monitored line. The requested energy capacity - P^E from individual LA, GA, or FA is used to relieve congestion by ΔPL_l is:

$$P^E = \Delta PL_l * PTDF_l \tag{3.2.12}$$

$$\Delta PL_l = |PL_l| - PL_l^{max} \tag{3.2.13}$$

$$PTDF_l = \Delta P_{i,j}/\Delta P_m \tag{3.2.14}$$

where, PL_l is power flow on congested line, PL_l^{max} is the capacity of the branch *l*. ΔP_m is power change at bus *m* caused power to change on the branch $\Delta P_{i,j}$.

The proposed OPF based auction algorithm utilizes DLMP, which is a nodal price model after including 1) $\lambda_r -$ the cost of energy in the node; 2) $\lambda_i^L -$ the price of losses paid by the node; 3) $\lambda_i^{Con} -$ the congestion price paid by the node. i.e.

$$\lambda_i^{DLMP} = \lambda_r + \lambda_i^L + \lambda_i^{Con} \tag{3.2.15}$$

$$\lambda_r = dC_p(P_p)/dP_k \tag{3.2.16}$$

where Cp is cost functions submitted by GAs, and FAs, P_k is the energy received at bus *k*. Loss component of LMP is derived:

$$\lambda_i^L = -\lambda_r \times LF_k = -\lambda_r \times (dL_t/dPL_l) \tag{3.2.17}$$

$$L_t = \sum_{i=1}^{n} F_l^2 \, Z_l \tag{3.2.18}$$

where:

$$F_l = \sum_{i=1}^{n} a_{l,i} * P_{ti} \quad (3.2.19)$$

$$a_{l,i} = (z_{ni} - z_{mi})/z_l \quad (3.2.20)$$

where LF_k is the loss factor at bus k, PL_l is the line flow at branch *l*. α is the line sensitivity factor of line *l*, where z_{ni} and z_{mi} are self-impedance of two ends of the line. Congestion component of DLMP can be derived:

$$\lambda_i^{Con} = \sum_{i=1}^{n} (a_{l,i} \times TL_l) \quad (3.2.21)$$

$$TL_l = \pi^+ - \pi^- \quad (3.2.22)$$

where TL_l is total cost variation with π^+ and without π^- line limits in line *l*. Where π are Lagrange multipliers of line limit constraints.

To model the DLMP based electricity market, a concept of nodal Market Clearing Price (λ_i^{Price}) is introduced. Where nodal MCP indicates real price paid by the node to receive one unit of energy. This price is calculated by considering market clearing price and DLMP parameters like loss and congestion components.

$$\lambda_i^{Price} = \frac{MCP}{\lambda_r} \lambda_i^{DLMP} \quad (3.2.22)$$

where, λ_i^{Price} is the market-clearing price at node *i*, λ_i^{DLMP} is the locational marginal price at bus *i*, λ_r is the cost of energy in bus *r*.

Market Clearing Rate - *R* is a scale factor obtained by scaling prices till last accepted ask equals to its nodal price.

$$R = max_i(o_i^{p,LA}/\lambda_r) \quad (3.2.23)$$

where, $o_i^{p,LA}$ is the price of last accepted energy ask price from seller *i*, λ_r is the marginal energy price at bus *r*.

3.2.4. Nodal earning component

TEMA conducts DLMP based double-sided auction (Figure 3.2.1), where buyers and sellers get earning through participation in the market. Nodal earning component - λ_i^{earn} represents the

difference between nodal MCP and offer (or bid). The area of $Es_i * \lambda_i^{earn}$ in Figure 3.2.1 is additional revenue made by seller *i* after participating in the market.

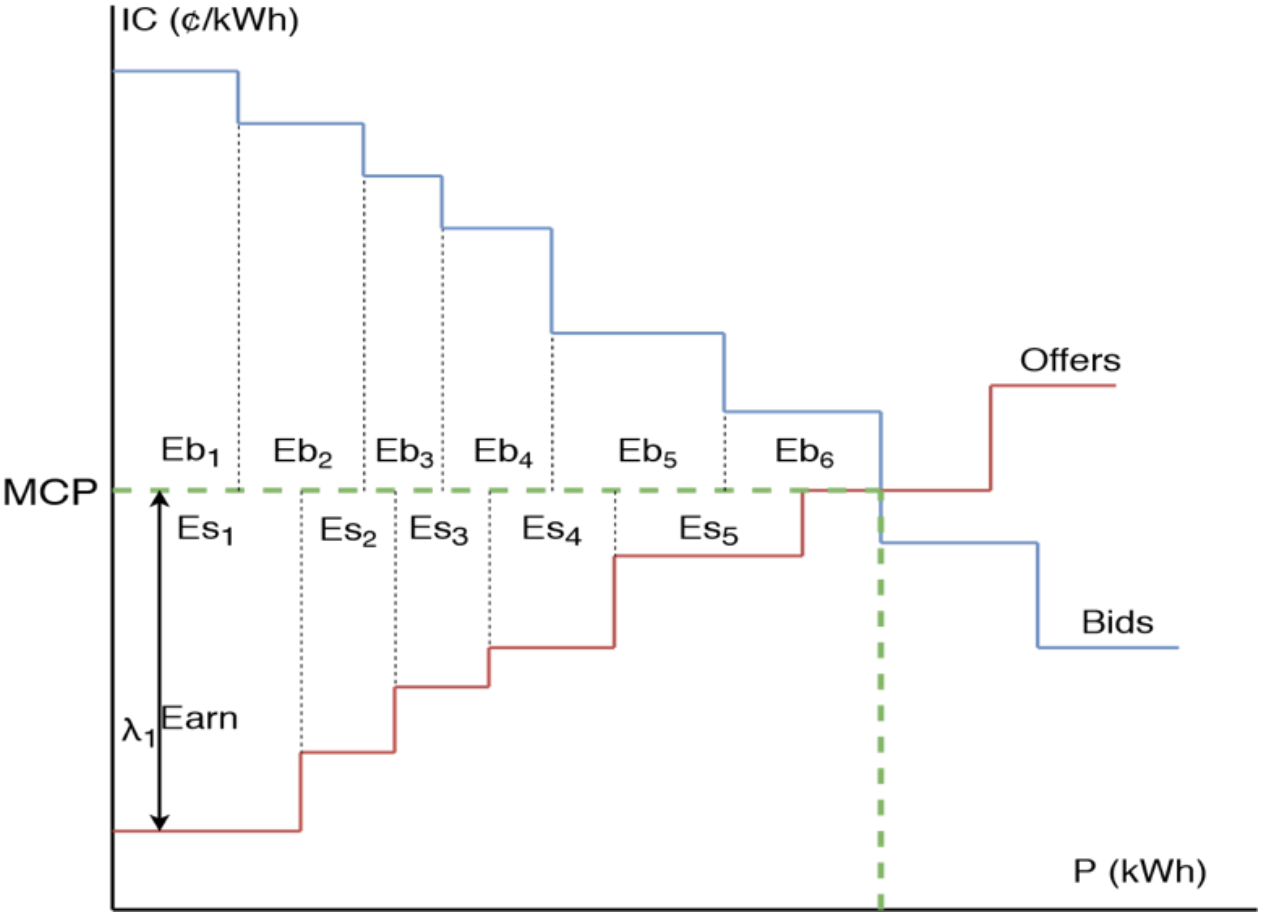

Figure 3.2.1. Double auction and earning component

Therefore, in the proposed approach these extra earnings can be used as a promotion tool for transactive energy implementation.

$$\lambda_i^{earn} = \frac{MCP}{\lambda_r} \lambda_i^{LMP} - \lambda_r \tag{3.2.24}$$

where, λ_i^{LMP} is the locational marginal price at bus *i*, λ_r is energy price at bus *r*.

It is essential to encourage agents to contribute to system betterment by providing incentives to valuable contributors and depriving non-contributing agents. Therefore, evaluation of the performance of agents is necessary to identify real contributors to the market. λ_i^{earn} parameter evaluates individual contributions of LAs, GAs, and FAs to the whole system. In the proposed

system personal contribution is considered high when $|\lambda_i^{earn}|$ is relatively large compared to other inputs. There are five levels of agents utility according to λ_i^{earn}: most valuable player (MVP), the second valuable player (SVP), the third valuable player (TVP), the least valuable player (LVP) and non-valuable player (NVP).

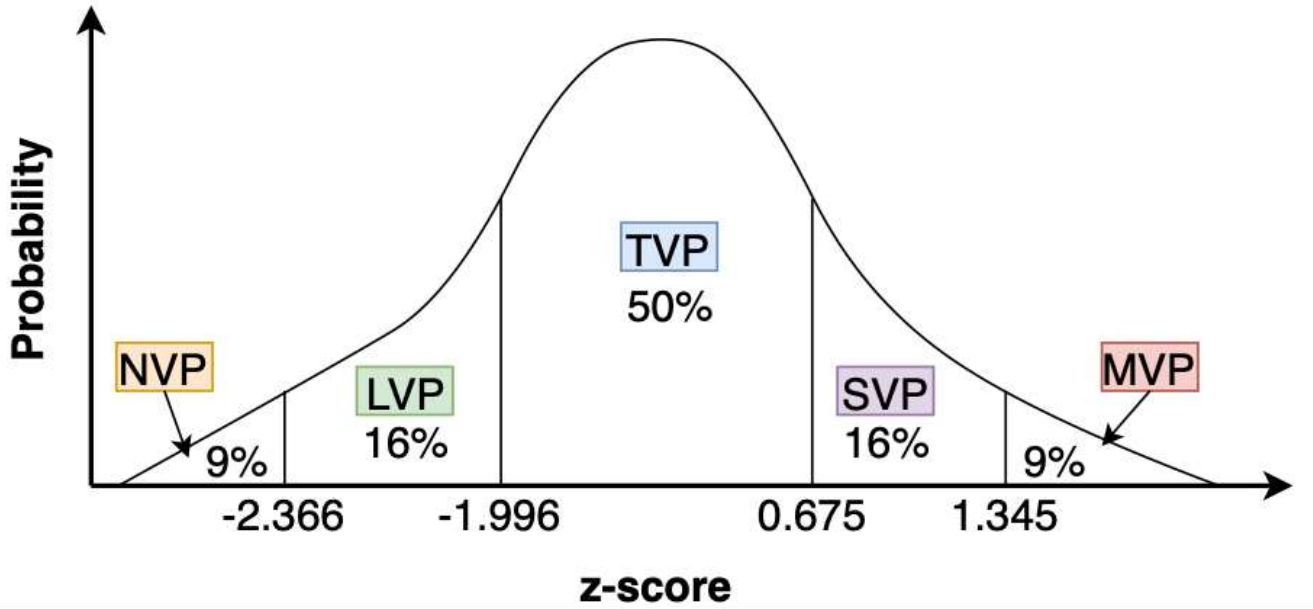

Figure 3.2.2. Value of agents based on contribution

Allocation of all agents is performed using the normal distribution to categorize contribution (Figure 3.2.2).

$$z = (|\lambda_i^{earn}| - \mu)/\sigma \tag{3.2.25}$$

where z is normally distributed random variable used for standard normal distribution function, λ_i^{earn} is the value that is being standardized, μ is the mean of the distribution and σ is the standard deviation of the distribution.

After ranking agents to a particular player value level, λ_i^{earn} are assigned as 100% honored for MVP, 75% for SVP and with a decrease of 25% for every other player value, so for agents located in NVP range; there is no extra profit. The ones, who submitted low bids or help to clear congestion, have a higher chance to get a high value of earning component and, thus, to be elected as MVP.

Through this method, a system has money for stable TE framework operation, due to an adequate amount of revenue to be distributed among stakeholders such as DSO and TE service providers.

For nodal earning components the share is decided individually by each node:

$$\lambda_i^{earn} = (a_i + b_i + c_i)\lambda_i^{earn}, \forall\, i \in n \tag{3.2.26}$$

where, $a_i, b_i, c_i \in [0.00, 1.00]$ are elements of TE stakeholders that have shared in earning, where, a is a share of players, b is a share of DSO, and c is a share of TE service providers. The primary objective is to build a reasonable and reliable TE framework that satisfies all sides. TEMA is the governing agent for optimally scheduling incentives among stakeholders and executing the MVP algorithm.

Chapter 4 – Case Study

IEEE 33 bus feeder is selected as a physical topology of the system. Optimal locations for generator and storage elements in the distribution feeder is determined using the autoadd command in OpenDSS.

A wide variety of distributed energy sources with different power profiles that are comparable with the radial network were selected to demonstrate the feasibility of the TE framework. In the given system some nodes have implemented MAS and eligible for participating in the proposed transactive energy framework. Player agents supervise renewable energy sources, in particular, LAs supervises EV, and various DESS, whereas GAs oversee HVAC and other controllable loads, and FAs manage microgrids.

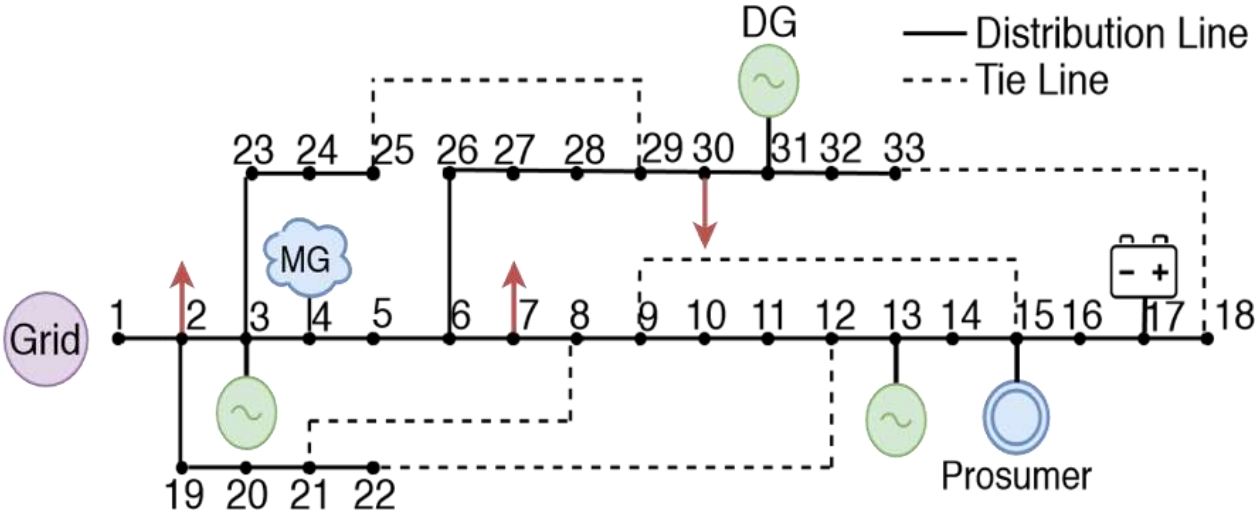

Figure 4.1. Modeled IEEE 33 bus system with generation, flexible and prosumers

In trading time interval Δt LAs, GAs, and SAs are submitting bids and offers to TEMA based on their strategy. Table 4.1 shows generator offers, and Table 4.2 shows bids submitted by loads.

Table 4.1. Generator offers

Bus	Type	Block 1 kW || kWh	Block 2 kW || kWh	Block 3 kW || kWh
1	GSP	Inf || 5.0	-	-
3	Solar	96 || 3.0	-	-

4	EV	12 || 2.0	24 || 4.2	24 || 8.0
17	Battery	12 || 2.0	24 || 4.4	24 || 9.0
13	Wind	96 || 2.0	-	-
31	Controllable	50 || 2.0	50 || 6.0	50 || 8.0

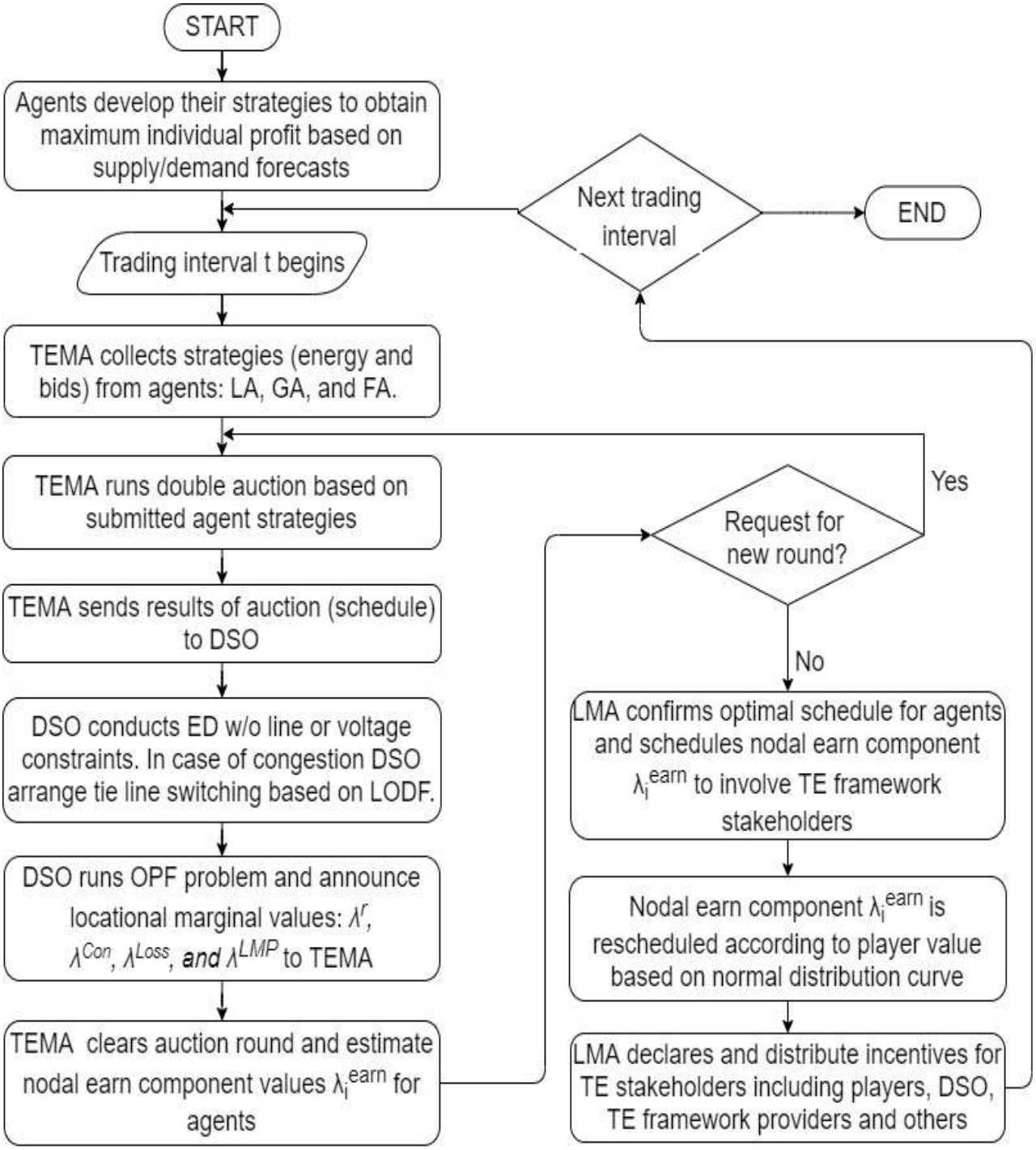

Figure 4.2. Flowchart for the TE framework

Table 4.2. Load bids

Bus	Type	Block 1 kW || kWh	Block 2 kW || kWh	Block 3 kW || kWh
7	HVAC	40 || 10.0	30 || 7.0	30 || 6.0
15	EV	20 || 10.0	20 || 5.0	20 || 2.0
30	Industry	100 || 10.0	50 || 6.0	50 || 5.0
2	GBP	inf || 3.0	-	-

Based on the submitted bids and offers TE runs OPF-based smart market to determine winner and loser blocks.

Chapter 5 – Results and Discussion

Market simulation for the 33-bus radial feeder is conducted using Matlab. Branch 2-3 is set to have capacity $PL_{2,3}$ = 3.06 MW.

OPF based market results are shown in Table 5.1.

Table 5.1. Market results

Bus	Pg (kWh)	λ^{Price} (¢//kWh)	Revenue (¢)	Cost (¢)	Earning (¢)	λ^{earn} (¢//kWh)
1	2760.7	5.000	13803.5	13803.6	0.0	0.000
3	96.5	5.671	547.2	289.5	257.7	2.671
4	36.0	5.715	205.7	120.0	85.7	2.382
17	36.0	6.056	218.0	120.0	98.0	2.722
13	96.5	6.019	580.9	193.0	387.9	4.019
31	83.9	6.000	503.4	471.5	31.9	0.381
7	-100.0	5.884	-588.4	-999.8	411.4	-4.114
15	-20.0	5.000	-100.0	-200.0	100.0	-5.001
30	-100.0	6.000	-600.0	-1000.0	400.0	-4.000
2	0.0	5.017	0.0	0.0	0.0	0.000

OPF results include locational marginal components λ_r, λ_i^L, λ_i^{Con}. These marginal components of DLMP for agent deployed buses are shown in Table 5.2.

Table 5.2. Nodal marginal components (DLMP)

Bus	λ_r	λ_i^{Con}	λ_i^L	λ_i^{LMP}	λ_i^{Price}	λ^{earn}
1	5.000	0.000	0.000	5.000	5.000	0.000
3	3.000	0.343	0.059	3.402	5.671	2.671
4	3.333	0.383	0.093	3.810	5.715	2.382
17	3.333	0.412	0.291	4.037	6.056	2.722
13	2.000	0.245	0.163	2.408	6.019	4.019
31	5.619	1.943	-0.819	6.743	6.000	0.381

7	9.998	1.186	0.582	11.766	5.884	-4.114
15	10.001	-0.001	0.001	10.001	5.000	-5.001
30	10.000	0.790	1.210	12.000	6.000	-4.000
2	3.000	0.000	0.010	3.010	5.017	0.000

Results of the OPF-based market provide the optimal schedule for players. Earnings of market participants are considered as pure profit. However, in the real world, these earnings should be redistributed across multiple stakeholders. The significant stakeholders' such as Players, DSO, and TE providers were selected as in Figure 5.1.

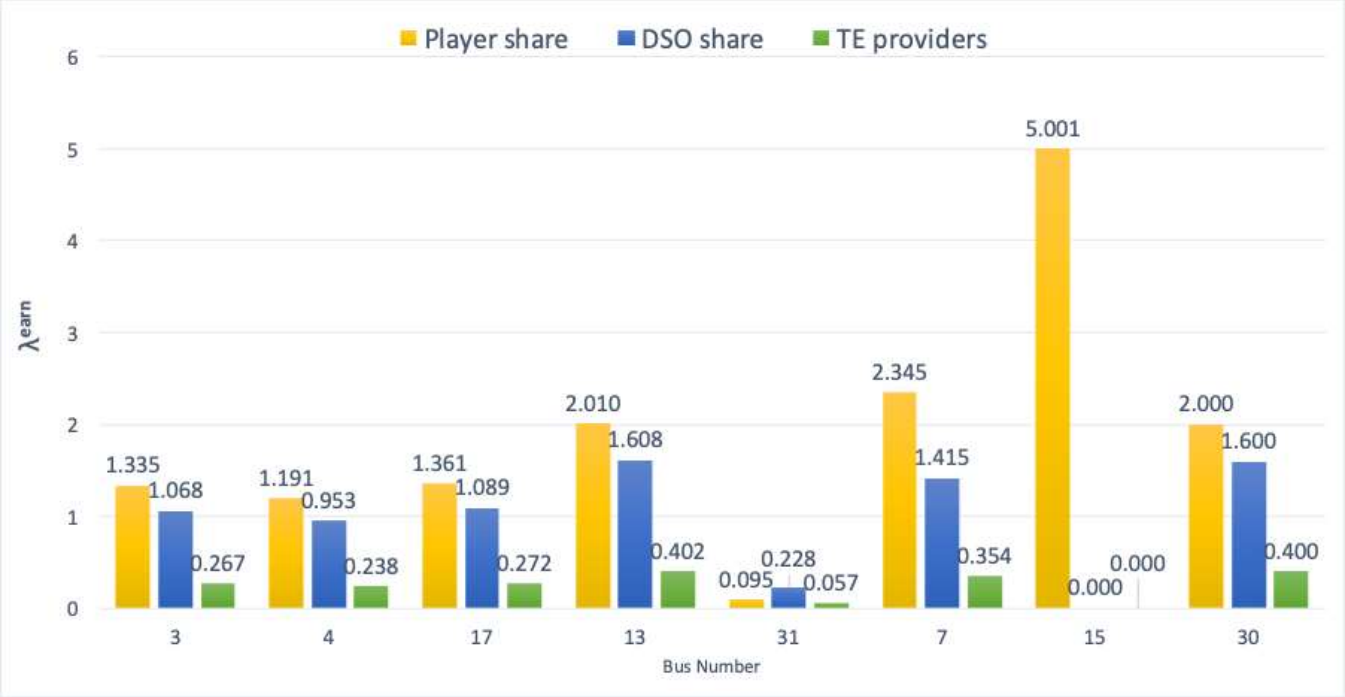

Figure 5.1. Distribution of marginal earn component

The schedule for distributed energy sources that have participated in the market includes the λ^{earn} component that is subject to rescheduling.

When rescheduling the λ^{earn} parameter, Players' incentives were subject to the value of a player in the market. The player is considered as valuable when it has a relatively large λ^{earn} parameter. Higher λ^{earn} parameter for GA indicates lower offers, and for LA, it is higher bids. Therefore, these players are considered valuable for the market and eligible to become MVP. The scheduling

λ^{earn} component is given in Fig.8. From these results it is seen that Bus 15 is MVP and has 100% share for the profit, Bus 31 is LVP and has a 25% share, and other players are TVP with share 50%. Therefore, $a_{15} = 1$, $a_{31} = 0.25$, $a_i = 0.5$, where $i \notin$ Bus (15, 31) is inserted to Bus 17. The share of DSO and TE service providers was chosen as 80% and 20% of the remaining schedule. $b_{15} = c_{15} = 0$, $b_{31} = 0.6$, $c_{31} = 0.15$, $b_i = 0.4$, $c_i = 0.1$, where $i \notin$ Bus (15, 31).

The proposed TE framework is a feasible solution to implement it within existing radial networks with the support of DSO. The proposed player value management system attracts higher bids and lower offers that lead to an overall decrease in electricity prices. This TE framework ensures that agents submit an honest bid and offers since underbidding results in getting Non-Valuable-Player (NVP) status with 0% player-share for incentives.

Chapter 6 – Conclusion and Future Works

The proposed DLMP-based transactive energy framework is simulated on a modified IEEE 33-bus radial feeder. The results have shown, that LMP based auction markets can be applied for optimal dispatch and can assist in congestion management and loss minimization for the system. The congestion relieving method using agent-controlled demand response was proposed. A novel approach of utilizing λ^{earn} parameter is offered, this method is developed in a way to promote transactive energy in distribution systems. The proposed method considers incentives for major parties that are involved in the transactive energy network.

To regulate the market operations, the player value parameter was introduced to award the most valuable market participants. It was proven that dishonest players that practice underbidding are losing due to the risk of becoming LVP or NVP with low λ^{earn} player-share. Also, the results of the proposed framework have shown that MAS can assist distributed energy resource owners to form onsite energy markets and participate.

In future work, the proposed TE framework should be implemented in laboratory-scale nano-grid to test the feasibility of the model. Besides, it is also important to conduct simulation for a larger period to verify the system in the long run. Moreover, more focus on FA modeling should be provided, that could cover more effective battery utilization. However, the most important step to further develop the current work is to start testing in special laboratories or small physical regions. The author believes that the project can be executed in Kazakhstan's distribution systems. With the help of a local automated commercial energy meter manufacturer "Saiman company," it is possible to co-develop smart cyber-physical smart-meter. In addition, it is possible to cooperate with Sigma Telas (Lithuania) company that provides a framework for Automatic Commercial Energy Accounting (ACEA) for power transmission systems of Kazakhstan for the development of decentralized TE framework suitable for the distribution system of Kazakhstan and CIS region countries.

Bibliography

[1]. N. Laboratory, “Transactive energy: An overview,” Apr. 2017.

[2]. J. Qiu and J. Zhao, “Optimal scheduling for prosumers in coupled transactive power and gas systems,” IEEE Transactions on Power Systems, vol. 33, DOI 10.1109/TPWRS.2017.2715983, no. 2, pp. 1970 – 1980, Mar. 2018.

[3]. Y. K. Renani and M.Ehsan, “Optimal transactive market operations with distribution system operators,” IEEE Transactions on Smart Grid, DOI 10.1109/TSG.2017.2718546, Jun. 2017.

[4]. J. Qui and K. Meng, “Optimal scheduling of distributed energy resources as a virtual power plant in a transactive energy framework,” IET Generation, Transmission & Distribution, vol. 11, DOI 10.1049/ietgtd.2017.0268, no. 13, pp. 3417 – 3427, Oct. 2017.

[5]. A. Hahn and R. Singh, “Smart contract-based campus demonstration of decentralized transactive energy auctions,” IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), DOI 10.1109/ISGT.2017.8086092, Oct. 2017.

[6]. Y. Amanbek, Y. Tabarak, H. S. V. S. K. Nunna, and S. Doolla, “Decentralized transactive energy management system for distribution systems with prosumer microgrids,” Proc. 2018 19th Int. Carpathian Control Conf. ICCC 2018, pp. 553–558, 2018.

[7]. P. Divshali and B. Choi, “Multi-agent transactive energy management system considering high levels of renewable energy source and electric vehicles,” IET Generation, Transmission & Distribution, vol. 11, DOI 10.1049/iet-gtd.2016.1916, no. 15, pp. 3713 – 3721, Nov. 2017.

[8]. M. Akter and M. Mahmud, “Comparative analysis of energy trading priorities based on open transactive energy markets in residential microgrids,” Universities Power Engineering Conference (AUPEC), DOI 10.1109/AUPEC.2017.8282400, Feb. 2018.

[9]. N. Liu, J. Wang, and L. Wang, “Hybrid energy sharing for multiple microgrids in an integrated heat-electricity energy system,” IEEE Transactions on Sustainable Energy, vol. 10, DOI 10.1109/TSTE.2018.2861986, no. 3, pp. 1139–1151, Jul. 2019.

[10]. Q. Wang, H. Fang, J. Chen, Y. Mao and L. Dou, "Flocking with obstacle avoidance and connectivity maintenance in multi-agent systems," *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, Maui, HI, 2012, pp. 4009-4014.

[11]. K. Meissner and H. Hensel, "Agent based assistance and support in process control systems," *18th International Conference on Systems Engineering (ICSEng'05)*, Las Vegas, NV, USA, 2005, pp. 159-163.

[12]. N. Badr, "An agent-based architecture for intelligent decision support system," *2010 10th International Conference on Intelligent Systems Design and Applications*, Cairo, 2010, pp. 1213-1217.

[13]. K. Huang, S. Srivastava and D. Cartes, "Decentralized Reconfiguration for Power Systems Using Multi Agent System," *2007 1st Annual IEEE Systems Conference*, Honolulu, HI, 2007, pp. 1-6.

[14]. A. Halinka, P. Rzepka and M. Szablicki, "Agent model of multi-agent system for area power system protection," *2015 Modern Electric Power Systems (MEPS)*, Wroclaw, 2015, pp. 1-4.

[15]. N. K. Nagwani, "Performance measurement analysis for multi-agent systems," *2009 International Conference on Intelligent Agent & Multi-Agent Systems*, Chennai, 2009, pp. 1-4.

[16]. Z. Guessoum, "Adaptive agents and multiagent systems," in *IEEE Distributed Systems Online*, vol. 5, no. 7, 2004.

[17]. H. K. Nunna and D. Srinivasan, "Multiagent-based transactive energy framework for distribution systems with smart microgrids," *IEEE Transactions on Industrial Informatics*, vol. 13, DOI 10.1109/TII.2017.2679808, no. 5, pp. 2241 – 2250, Mar. 2017.

[18]. A. K. Zarabie, S. Das, and M. N. Faqiry, "Fairness-regularized dlmp-based bilevel transactive energy mechanism in distribution systems," *IEEE Transactions on Smart Grid*, vol. 10, DOI 10.1109/TSG.2019.2895527, no. 6, pp. 6029–6040, Nov. 2019.

[19]. H. S. V. S. K. Nunna and D. Srinivasan, "Multiagent-Based Transactive Energy Framework for Distribution Systems with Smart Microgrids*," IEEE Trans. Ind. Informatics*, vol. 13, no. 5, pp. 2241–2250, 2017.

[20]. A. Hahn, R. Singh, and C. Liu, "Smart Contract-based Campus Demonstration of Decentralized Transactive Energy Auctions."

[21]. J. Hu, G. Yang, and H. W. Bindner, "Network constrained transactive control for electric vehicles integration," *IEEE Power Energy Soc. Gen. Meet.*, vol. 2015–Septe, pp. 1–5, 2015.

[22]. P. H. Divshali, B. J. Choi, and H. Liang, "Multi-agent transactive energy management system considering high levels of renewable energy source and electric vehicles," *IET Gener. Transm. Distrib*., vol. 11, no. 15, pp. 3713–3721, 2017.

[23]. J. Qiu, K. Meng, Y. Zheng, and Z. Y. Dong, "Optimal scheduling of distributed energy resources as a virtual power plant in a transactive energy framework," *IET Gener. Transm. Distrib*., vol. 11, no. 13, pp. 3417–3427, 2017.

[24]. J. Hu, G. Yang, H. W. Bindner, and Y. Xue, "Application of Network-Constrained Transactive Control to Electric Vehicle Charging for Secure Grid Operation," *IEEE Trans. Sustain. Energy*, vol. 8, no. 2, pp. 505–515, 2017

[25]. Wen, F. and David, A., 2001. "Optimal bidding strategies and modeling of imperfect information among competitive generators," *IEEE Transactions on Power Systems*, 16(1), pp.15-21.

[26]. Zhang, D., Wang, Y. and Luh, P., 2010. "Optimization based bidding strategies in the deregulated market," *IEEE Transactions on Power Systems*, 15(3), pp.981-986.

[27]. Bompard, E., Lu, W. and Napoli, R., 2006. "Network Constraint Impacts on the Competitive Electricity Markets Under Supply-Side Strategic Bidding, " *IEEE Transactions on Power Systems*, 21(1), pp.160-170.

[28]. Vahidinasab, V. and Jadid, S., 2009. "Multiobjective environmental/techno-economic approach for strategic bidding in energy markets," *Applied Energy*, 86(4), pp.496-504.

[29]. Zakeri, G., 2016. "Pricing Wind: A Revenue Adequate, Cost Recovering Uniform Price for Electricity Markets with Intermittent Generation," *SSRN Electronic Journal*.

[30]. Weber, J. and Overbye, T., 2002. "Generation Bidding Strategies Based on Two-Level Optimnization and Bids Sensitivities," *IEEE Power Engineering Review*, 22(2), pp.44-46.

[31]. Hao, S., 2000. "A study of basic bidding strategy in clearing pricing auctions," *IEEE Transactions on Power Systems*, 15(3), pp.975-980.

[32]. Vytelingum, P., Cliff, D. and Jennings, N., 2008. "Strategic bidding in continuous double auctions," *Artificial Intelligence*, 172(14), pp.1700-1729.

[33]. Hong, Y., Tsai, S. and Weng, M., 2001. "Bidding strategy based on artificial intelligence for a competitive electric market," *IEE Proceedings - Generation, Transmission and Distribution*, 148(2), p.159.

[34]. Yermish, I., Miori, V., Yi, J., Malhotra, R. and Klimberg, R., 2010. “Business Plus Intelligence Plus Technology Equals Business Intelligence,” *International Journal of Business Intelligence Research,* 1(1), pp.48-63.

[35]. Yang, C., 2013. “Projects Bidding Decision Risk Analysis Based on Multi-factor Clustering Analysis,” *Information Technology Journal*, 12(21), pp.6164-6168.

[36]. N. Tang and B. Wang, "The research on the value of distributed resources based on the decomposition of distribution LMP(DLMP)," *2017 IEEE Conference on Energy Internet and Energy System Integration (EI2)*, Beijing, 2017, pp. 1-6.

[37]. J. Wei, Y. Zhang, F. Sahriatzadeh and A. K. Srivastava, "DLMP using three-phase current injection OPF with renewables and demand response," *IET Renewable Power Generation*, vol. 13, no. 7, pp. 1160-1167, 20 5 2019.

[38]. L. Bai, J. Wang, C. Wang, C. Chen and F. Li, "Distribution Locational Marginal Pricing (DLMP) for Congestion Management and Voltage Support," *IEEE Transactions on Power Systems*, vol. 33, no. 4, pp. 4061-4073, July 2018.

[39]. R. Yang and Y. Zhang, "Three-phase AC optimal power flow based distribution locational marginal price," *2017 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, Washington, DC, 2017, pp. 1-5.

[40]. A. K. Zarabie, S. Das and M. N. Faqiry, "Fairness-Regularized DLMP-Based Bilevel Transactive Energy Mechanism in Distribution Systems," *in IEEE Transactions on Smart Grid*, vol. 10, no. 6, pp. 6029-6040, Nov. 2019.

[41]. S. Huang, Q. Wu, S. S. Oren, R. Li and Z. Liu, "Distribution Locational Marginal Pricing Through Quadratic Programming for Congestion Management in Distribution Networks," *in IEEE Transactions on Power Systems*, vol. 30, no. 4, pp. 2170-2178, July 2015.

[42]. A. M. Farid, “Multi-agent system design principles for resilient operation of future power systems*,” in 2014 IEEE International Workshop on Intelligent Energy Systems (IWIES)*, 18–25, Oct. 2014.

[43]. J. Han, C. Choi, W. Park, I. Lee, and S. Kim, “Smart home energy management system including renewable energy based on zigbee and plc,” *IEEE Transactions on Consumer Electronics*, vol. 60, no. 2, pp. 198–202, May. 2014.

[44]. F. Rassaei, W. Soh, and K. Chua, "Demand response for residential electric vehicles with random usage patterns in smart grids," *IEEE Transactions on Sustainable Energy*, vol. 6, no. 4, pp. 1367–1376, Oct. 2015.

[45]. J. Hu, G. Yang, H. W. Bindner, and Y. Xue, "Application of network constrained transactive control to electric vehicle charging for secure grid operation," *IEEE Transactions on Sustainable Energy*, vol. 8, no. 2, pp. 505–515, Apr. 2017.

[46]. P. Vytelingum, R. K. Dash, E. David, and N. R. Jennings, "A risk-based bidding strategy for continuous double auctions," *in ECAI*, 2004.

Appendix

Part A. Matlab codes

```
function mpc = bus33_new

mpc.version = '2'; %using 2nd version of case format

%%Endering data in CDF
mpc.baseMVA = 100; %setting base MVA


%% bus data
%   bus_i   type    Pd  Qd  Gs  Bs  area    Vm  Va  baseKV  zone    Vmax    Vmin
mpc.bus = [  %% (Pd and Qd are specified in kW & kVAr here, converted to MW & MVAr below)
    1   3   0   0   0   0   1   1   0   12.66   1   1   1;
    2   2   0   0   0   0   1   1   0   12.66   1   1.1 0.9; %100 60
    3   2   0   0   0   0   1   1   0   12.66   1   1.1 0.9; %90 40
    4   2   0   0   0   0   1   1   0   12.66   1   1.1 0.9; %120 80
    5   1   60  30  0   0   1   1   0   12.66   1   1.1 0.9; %60 30
    6   1   60  20  0   0   1   1   0   12.66   1   1.1 0.9;
    7   2   100 100 0   0   1   1   0   12.66   1   1.1 0.9; %HVAC 200 100
    8   1   200 100 0   0   1   1   0   12.66   1   1.1 0.9;
    9   1   60  20  0   0   1   1   0   12.66   1   1.1 0.9;
    10  1   60  20  0   0   1   1   0   12.66   1   1.1 0.9;
    11  1   45  30  0   0   1   1   0   12.66   1   1.1 0.9;
    12  1   60  35  0   0   1   1   0   12.66   1   1.1 0.9;
    13  2   0   0   0   0   1   1   0   12.66   1   1.1 0.9; %60    35
    14  1   120 80  0   0   1   1   0   12.66   1   1.1 0.9;
    15  2   0    10 0   0   1   1   0   12.66   1   1.1 0.9; %HVAC 60*0 10*0
    16  1   60  20  0   0   1   1   0   12.66   1   1.1 0.9;
    17  2   0   20  0   0   1   1   0   12.66   1   1.1 0.9; %60    20
    18  1   90  40  0   0   1   1   0   12.66   1   1.1 0.9;
    19  1   90  40  0   0   1   1   0   12.66   1   1.1 0.9;
    20  1   90  40  0   0   1   1   0   12.66   1   1.1 0.9;
    21  1   90  40  0   0   1   1   0   12.66   1   1.1 0.9;
    22  1   90  40  0   0   1   1   0   12.66   1   1.1 0.9;
    23  1   90  50  0   0   1   1   0   12.66   1   1.1 0.9;
    24  1   420 200 0   0   1   1   0   12.66   1   1.1 0.9;
    25  1   420 200 0   0   1   1   0   12.66   1   1.1 0.9;
    26  1   60  25  0   0   1   1   0   12.66   1   1.1 0.9;
    27  1   60  25  0   0   1   1   0   12.66   1   1.1 0.9;
    28  1   60  20  0   0   1   1   0   12.66   1   1.1 0.9;
    29  1   120 70  0   0   1   1   0   12.66   1   1.1 0.9;
    30  2   0   600 0   0   1   1   0   12.66   1   1.1 0.9; %200   600
    31  2   0   0   0   0   1   1   0   12.66   1   1.1 0.9; %150   70
    32  1   210 100 0   0   1   1   0   12.66   1   1.1 0.9;
    33  1   60  40  0   0   1   1   0   12.66   1   1.1 0.9;
];
```

```
%% generator data
%   bus Pg  Qg  Qmax    Qmin    Vg  mBase stat Pmax Pmin   Pc1  Pc2 Qc1min  Qc1max  Qc2min  Qc2max  ramp_agc    ramp_10 ramp_30 ramp_q  apf
%l.gen=[5   0   0   0   0   1   100     1   0   -0.0001];
mpc.gen = [
    1   1000    0   60000   -15000  1   100 1   6000    -3500; %GSP

    3   960 0   0   0   1   100 1   960 940; %Solar
    4   600 0   0   0   1   100 1   600 120; %EV discharge
    17  600 0   0   0   1   100 1   600 120; %Battery discharge
    13  960 0   0   0   1   100 1   960 940; %Wind

    31  1500    70  70  0   1   100 1   1500    500; %Controlable
    7   -100    0   0 0     1    100    1   0   -100; %HVAC
    15  -60 0   0       0   1    100    1   0   -60; %EV charge
    30  -200    -100     0  -100       1    100     1   0   -200; %Industry

    2   -1000   0   0   0   1   100     1   0   -60000; %GBP

    %l.gen

];
```

```
%% branch data
%   fbus    tbus    r   x   b   rateA   rateB   rateC   ratio   angle   status  angmin  angmax
mpc.branch = [  %% (r and x specified in ohms here, converted to p.u. below)
    1   2   0.0922  0.0470  0   0   0   0   0   0   1   -360    360;
    2   3   0.4930  0.2511  0   1*3.06  0   0   0   0   1   -360    360;
    3   4   0.3660  0.1864  0   0   0   0   0   0   1   -360    360;
    4   5   0.3811  0.1941  0   0   0   0   0   0   1   -360    360;
    5   6   0.8190  0.7070  0   0   0   0   0   0   1   -360    360;
    6   7   0.1872  0.6188  0   0   0   0   0   0   1   -360    360;
    7   8   0.7114  0.2351  0   0   0   0   0   0   1   -360    360;
    8   9   1.0300  0.7400  0   0   0   0   0   0   1   -360    360;
    9   10  1.0440  0.7400  0   0   0   0   0   0   1   -360    360;
    10  11  0.1966  0.0650  0   0   0   0   0   0   1   -360    360;
    11  12  0.3744  0.1238  0   0   0   0   0   0   1   -360    360;
    12  13  1.4680  1.1550  0   0   0   0   0   0   1   -360    360;
    13  14  0.5416  0.7129  0   0   0   0   0   0   1   -360    360;
    14  15  0.5910  0.5260  0   0   0   0   0   0   1   -360    360;
    15  16  0.7463  0.5450  0   0   0   0   0   0   1   -360    360;
    16  17  1.2890  1.7210  0   0   0   0   0   0   1   -360    360;
    17  18  0.7320  0.5740  0   0   0   0   0   0   1   -360    360;
    2   19  0.1640  0.1565  0   0   0   0   0   0   1   -360    360;
    19  20  1.5042  1.3554  0   0   0   0   0   0   1   -360    360;
    20  21  0.4095  0.4784  0   0   0   0   0   0   1   -360    360;
    21  22  0.7089  0.9373  0   0   0   0   0   0   1   -360    360;
    3   23  0.4512  0.3083  0   0   0   0   0   0   1   -360    360;
    23  24  0.8980  0.7091  0   0   0   0   0   0   1   -360    360;
    24  25  0.8960  0.7011  0   0   0   0   0   0   1   -360    360;
    6   26  0.2030  0.1034  0   0   0   0   0   0   1   -360    360;
    26  27  0.2842  0.1447  0   0   0   0   0   0   1   -360    360;
    27  28  1.0590  0.9337  0   0   0   0   0   0   1   -360    360;
    28  29  0.8042  0.7006  0   0   0   0   0   0   1   -360    360;
    29  30  0.5075  0.2585  0   0   0   0   0   0   1   -360    360;
    30  31  0.9744  0.9630  0   0   0   0   0   0   1   -360    360;
```

```
    31  32  0.3105  0.3619  0   0   0   0   0   0   1   -360    360;
    32  33  0.3410  0.5302  0   0   0   0   0   0   1   -360    360;
    21  8   2.0000  2.0000  0   0   0   0   0   0   0   -360    360;
    9   15  2.0000  2.0000  0   0   0   0   0   0   0   -360    360;
    12  22  2.0000  2.0000  0   0   0   0   0   0   0   -360    360;
    18  33  0.5000  0.5000  0   0   0   0   0   0   0   -360    360;
    25  29  0.4512  0.3083  0   0   0   0   0   0   0   -360    360;
];

%%-----  OPF Data  -----%%
%% generator cost data
%   1   startup shutdown    n   x1  y1  ... xn  yn
%   2   startup shutdown    n   c(n-1)  ... c0


%l=[1   0   0   4   -200    -10000  -100    -5000   -50 -2500   0   0];
mpc.gencost = 0.001*[
    1   0   0   4   0   0   50  2500    100  5000   200 10000; %GBP

    1   0   0   4   0   0   940 28200   950 28500   960 28800; %Solar
    1   0   0   4   0   0   120 2400    360 12000   600 24000; %EV discharge
    1   0   0   4   0   0   120 2400    360 12000   600 24000; %battery discharge
    1   0   0   4   0   0   940 18800   950 19000   960 19200; %wind

    1   0   0   4   0   0   500 20000   1000    40000   1500    70000; %Controlable
    1   0   0   4   -100    -10000  -70 -7000   -30 -3000   0   0; %HVAC
    1   0   0   4   -60 -6000   -40 -4000   -20 -2000   0   0;%EV charge
    1   0   0   4   -30 -3000   -20 -2000   -10 -1000   0   0; %Industry
    1   0   0   4   -200    -2000   -100    -1000   -50 -500    0   0; %GSP

     %l
];
```

```
%% convert branch impedances from Ohms to p.u.
[PQ, PV, REF, NONE, BUS_I, BUS_TYPE, PD, QD, GS, BS, BUS_AREA, VM, ...
    VA, BASE_KV, ZONE, VMAX, VMIN, LAM_P, LAM_Q, MU_VMAX, MU_VMIN] = idx_bus;

[F_BUS, T_BUS, BR_R, BR_X, BR_B, RATE_A, RATE_B, RATE_C, ...
    TAP, SHIFT, BR_STATUS, PF, QF, PT, QT, MU_SF, MU_ST, ...
    ANGMIN, ANGMAX, MU_ANGMIN, MU_ANGMAX] = idx_brch;

[GEN_BUS, PG, QG, QMAX, QMIN, VG, MBASE, GEN_STATUS, PMAX, PMIN, ...
    MU_PMAX, MU_PMIN, MU_QMAX, MU_QMIN, PC1, PC2, QC1MIN, QC1MAX, ...
    QC2MIN, QC2MAX, RAMP_AGC, RAMP_10, RAMP_30, RAMP_Q, APF] = idx_gen;

[PW_LINEAR, POLYNOMIAL, MODEL, STARTUP, SHUTDOWN, NCOST, COST] = idx_cost;


Vbase = mpc.bus(1, BASE_KV) * 1e3;      %% in Volts
Sbase = mpc.baseMVA * 1e6;              %% in VA
mpc.branch(:, [BR_R BR_X]) = mpc.branch(:, [BR_R BR_X]) / (Vbase^2 / Sbase);
```

```
%% convert loads from kW to MW
mpc.bus(:, [PD, QD]) = mpc.bus(:, [PD, QD]) / 1e3;
mpc.gen(:, [PG, QG, QMAX, QMIN, PMAX, PMIN]) = mpc.gen(:, [PG, QG,QMAX, QMIN, PMAX, PMIN]) / 1e3;

mpc.gencost(:, [PW_LINEAR, NCOST]) = mpc.gencost(:, [PW_LINEAR, NCOST]) * 1e3;

res=mpc.gencost
%runopf('bus33_new')
%runpf('bus33_new')
```

```
mkt.OPF = 'AC';
mkt.auction_type = 1;
mpc = loadcase('bus33_new');
offers.P.qty = 0.00099999999*[
        0 2000 4000;
        960 0 0;
        120 240 240;
        120 240 240;
        960 0 0;
        500 500 500];
offers.P.prc = [
        50 50 50;
        30 0  0;
        20 42 80;
        20 44 90;
        20 0 0;
        20 45 70 ];
   %l.qty=[60 0 0];
   %l.prc=[40 0 0];
bids.P.qty = 0.001*[
        40 30 30;
        20 20 20;
        100 50 50;
        4000 2000 0;
        % 0.000001 0 0
        ];
bids.P.prc = [
        100 70 60;
        100 50 20;
        100 60 50;
        30  30 30;
       % 0 0 0
        ];
[mpc_out, co, cb, f, dispatch, success, et] = runmarket(mpc, offers, bids, mkt);
res=dispatch
res2=mpc
```

Part B. JADE codes

B.1. DSO code

```
package Tema;

import java.io.File;

public class DSO extends Agent
{
    float GSP, GBP;
    String gsp, gbp;
    float OPF = 0;
    String opf;
    protected void gbp()
        {
          ParallelBehaviour parallelBehaviour = new ParallelBehaviour();
            parallelBehaviour.addSubBehaviour(new receive_start(this));
            parallelBehaviour.addSubBehaviour(new run_OPF(this));
            parallelBehaviour.addSubBehaviour(new send_OPF(this));
            this.addBehaviour(parallelBehaviour);
        }
    class receive_start extends CyclicBehaviour{
        public receive_start(Agent a){
            super(a);}
        @Override
        public void action() {
            MessageTemplate mt=MessageTemplate.not(MessageTemplate.MatchSender(myAgent.getAID()));
            ACLMessage msg=myAgent.receive(mt);
            if(msg != null){
                if (msg.getConversationId() == "DSOPrices"){
                    gsp = Float.toString(GSP);
                    gsp = Float.toString(GBP);

                    ACLMessage msg1= new ACLMessage(ACLMessage.INFORM);
                    msg1.addReceiver(new AID("TEMA", AID.ISLOCALNAME));
                    msg1.setConversationId("GSP");
                    msg1.setContent(gsp);
                    myAgent.send(msg1);

                    ACLMessage msg2= new ACLMessage(ACLMessage.INFORM);
                    msg2.addReceiver(new AID("TEMA", AID.ISLOCALNAME));
                    msg2.setConversationId("GBP");
                    msg2.setContent(gbp);
                    myAgent.send(msg2);
                }}}}
    class run_OPF extends CyclicBehaviour{
        public run_OPF(Agent a){
            super(a);}
        @Override
        public void action() {
            MessageTemplate mt=MessageTemplate.not(MessageTemplate.MatchSender(myAgent.getAID()));
            ACLMessage msg=myAgent.receive(mt);
            if(msg != null){
                if (msg.getConversationId() == "OPF"){
                    //Running OPF in the Matlab
                    OPF = 1;
                    opf = Float.toString(OPF);
                }}}}
    class send_OPF extends CyclicBehaviour{
        public send_OPF(Agent a){
            super(a);}
        @Override
        public void action() {
            //Sending OPF results to the MAtlab (actually data is already in Matlab,
            //and this code is just to show communication)
            if (OPF != 0) {
                ACLMessage msg= new ACLMessage(ACLMessage.INFORM);
                msg.addReceiver(new AID("TEMA", AID.ISLOCALNAME));
                msg.setConversationId("OPFresults");
                msg.setContent(opf);
                myAgent.send(msg);
                block();
            }}}
}
```

B.2. FA code

```
package Tema;
import java.io.File;
public class FA1 extends Agent
{
    int level = 0;
    int val = 0;
    float bid, load;
    String Bid, Load;
    String results;
    protected void setup()
        {
          ParallelBehaviour parallelBehaviour = new ParallelBehaviour();
            parallelBehaviour.addSubBehaviour(new receive_start(this));
            parallelBehaviour.addSubBehaviour(new send_bid(this));
            parallelBehaviour.addSubBehaviour(new send_load(this));
            parallelBehaviour.addSubBehaviour(new rec_bid_Ok(this));
            parallelBehaviour.addSubBehaviour(new rec_load_Ok(this));
            parallelBehaviour.addSubBehaviour(new rec_result(this));
            this.addBehaviour(parallelBehaviour);
        }
    class receive_start extends CyclicBehaviour{
        public receive_start(Agent a){
            super(a);}
        @Override
        public void action() {
            MessageTemplate mt=MessageTemplate.not(MessageTemplate.MatchSender(myAgent.getAID()));
            ACLMessage msg=myAgent.receive(mt);
            if(msg != null){
                if (msg.getConversationId() == "TEMAStart"){
                    System.out.println("FA1 received message from TEMA regarding start");
                    level = 1;
                    val = 1;
                    block();
                }}}}
    class send_bid extends CyclicBehaviour{
        public send_bid(Agent a){
            super(a);}
        @Override
        public void action() {
            if(level == 1) {
                ACLMessage msg1= new ACLMessage(ACLMessage.INFORM);
                msg1.addReceiver(new AID("TEMA", AID.ISLOCALNAME));
                msg1.setConversationId("FA1Bid");
                Bid = String.valueOf(bid);
                msg1.setContent(Bid);
                myAgent.send(msg1);
            }
            else if(level == 2) {
                block();}
            }
        }
    class send_load extends CyclicBehaviour{
        public send_load(Agent a){
            super(a);}
        @Override
        public void action() {
            if(val == 1) {
                ACLMessage msg1= new ACLMessage(ACLMessage.INFORM);
                msg1.addReceiver(new AID("TEMA", AID.ISLOCALNAME));
                msg1.setConversationId("FA1val");
                Load = String.valueOf(load);
                msg1.setContent(Load);
                myAgent.send(msg1);
            }
            else if(val == 2) {
                block();}
            }
        }
    class rec_bid_Ok extends CyclicBehaviour{
        public rec_bid_Ok(Agent a){
            super(a);}
        @Override
        public void action() {
            MessageTemplate mt=MessageTemplate.not(MessageTemplate.MatchSender(myAgent.getAID()));
            ACLMessage msg=myAgent.receive(mt);
            if(msg != null){
                if (msg.getConversationId() == "FA1bidok"){
                    level = 2;
                    block();
                }}}}
        class rec_result extends CyclicBehaviour{
            public rec_result(Agent a){
                super(a);}
            @Override
            public void action() {
                MessageTemplate mt=MessageTemplate.not(MessageTemplate.MatchSender(myAgent.getAID()));
                ACLMessage msg=myAgent.receive(mt);
                if(msg != null){
                    if (msg.getConversationId() == "FA1result"){
                        results = msg.getContent();
                        block();
                    }}}}
    }
```

B.3. GA code

```
package Tema;
import java.io.File;
public class GA1 extends Agent
{
    int level = 0;
    int val = 0;
    float offer, gen;
    String Offer, Gen;
    String results;
    protected void setup()
        { ParallelBehaviour parallelBehaviour = new ParallelBehaviour();
            parallelBehaviour.addSubBehaviour(new receive_start(this));
            parallelBehaviour.addSubBehaviour(new send_offer(this));
            parallelBehaviour.addSubBehaviour(new send_gen(this));
            parallelBehaviour.addSubBehaviour(new rec_offer_Ok(this));
            parallelBehaviour.addSubBehaviour(new rec_gen_Ok(this));
            parallelBehaviour.addSubBehaviour(new rec_result(this));
            this.addBehaviour(parallelBehaviour);
        }
    class receive_start extends CyclicBehaviour{
        public receive_start(Agent a){
            super(a);}
        @Override
        public void action() {
            MessageTemplate mt=MessageTemplate.not(MessageTemplate.MatchSender(myAgent.getAID()));
            ACLMessage msg=myAgent.receive(mt);
            if(msg != null){
                if (msg.getConversationId() == "TEMAStart"){
                    System.out.println("GA1 received message from TEMA regarding start");
                    level = 1;
                    val = 1;
                    block();
                }}}
    class send_offer extends CyclicBehaviour{
        public send_offer(Agent a){
            super(a);}
        @Override
        public void action() {
            if(level == 1) {
                ACLMessage msg1= new ACLMessage(ACLMessage.INFORM);
                msg1.addReceiver(new AID("TEMA", AID.ISLOCALNAME));
                msg1.setConversationId("GA1Offer");
                Offer = String.valueOf(offer);
                msg1.setContent(Offer);
                myAgent.send(msg1);
            }
            else if(level == 2) {
                block();}
            }}
    class send_gen extends CyclicBehaviour{
        public send_gen(Agent a){
            super(a);}
        @Override
        public void action() {
            if(val == 1) {
                ACLMessage msg1= new ACLMessage(ACLMessage.INFORM);
                msg1.addReceiver(new AID("TEMA", AID.ISLOCALNAME));
                msg1.setConversationId("GA1val");
                Gen = String.valueOf(gen);
                msg1.setContent(Gen);
                myAgent.send(msg1);
            }
            else if(val == 2) {
                block();}
            }}
    class rec_offer_Ok extends CyclicBehaviour{
        public rec_offer_Ok(Agent a){
            super(a);}
        @Override
        public void action() {
            MessageTemplate mt=MessageTemplate.not(MessageTemplate.MatchSender(myAgent.getAID()));
            ACLMessage msg=myAgent.receive(mt);
            if(msg != null){
                if (msg.getConversationId() == "GA1offerok"){
                    level = 2;
                    block();
                }}}
    class rec_gen_Ok extends CyclicBehaviour{
        public rec_gen_Ok(Agent a){
            super(a);}
        @Override
        public void action() {
            MessageTemplate mt=MessageTemplate.not(MessageTemplate.MatchSender(myAgent.getAID()));
            ACLMessage msg=myAgent.receive(mt);
            if(msg != null){
                if (msg.getConversationId() == "GA1genok"){
                    val = 2;
                    block();
                }}}
    class rec_result extends CyclicBehaviour{
        public rec_result(Agent a){
            super(a);}
        @Override
        public void action() {
            MessageTemplate mt=MessageTemplate.not(MessageTemplate.MatchSender(myAgent.getAID()));
            ACLMessage msg=myAgent.receive(mt);
            if(msg != null){
                if (msg.getConversationId() == "GA1result"){
                    results = msg.getContent();
                    block();
                }}}
}
```

B.4. LA code

```
package Tema;
import java.io.File;
public class LA1 extends Agent
{
    int level = 0;
    int val = 0;
    float bid, load;
    String Bid, Load;
    String results;
    protected void setup()
        {
          ParallelBehaviour parallelBehaviour = new ParallelBehaviour();
            parallelBehaviour.addSubBehaviour(new receive_start(this));
            parallelBehaviour.addSubBehaviour(new send_bid(this));
            parallelBehaviour.addSubBehaviour(new send_load(this));
            parallelBehaviour.addSubBehaviour(new rec_bid_Ok(this));
            parallelBehaviour.addSubBehaviour(new rec_load_Ok(this));
            parallelBehaviour.addSubBehaviour(new rec_result(this));
            this.addBehaviour(parallelBehaviour);
        }
    class receive_start extends CyclicBehaviour{
        public receive_start(Agent a){
            super(a);}
        @Override
        public void action() {
            MessageTemplate mt=MessageTemplate.not(MessageTemplate.MatchSender(myAgent.getAID()));
            ACLMessage msg=myAgent.receive(mt);
            if(msg != null){
                if (msg.getConversationId() == "TEMAStart"){
                    System.out.println("LA1 received message from TEMA regarding start");
                    level = 1;
                    val = 1;
                    block();
                }}}}
    class send_bid extends CyclicBehaviour{
        public send_bid(Agent a){
            super(a);}
        @Override
        public void action() {
            if(level == 1) {
                ACLMessage msg1= new ACLMessage(ACLMessage.INFORM);
                msg1.addReceiver(new AID("TEMA", AID.ISLOCALNAME));
                msg1.setConversationId("LA1Bid");
                Bid = String.valueOf(bid);
                msg1.setContent(Bid);
                myAgent.send(msg1);
            }
            else if(level == 2) {
                block();}
            }}
     class send_load extends CyclicBehaviour{
         public send_load(Agent a){
             super(a);}
         @Override
         public void action() {
             if(val == 1) {
                 ACLMessage msg1= new ACLMessage(ACLMessage.INFORM);
                 msg1.addReceiver(new AID("TEMA", AID.ISLOCALNAME));
                 msg1.setConversationId("LA1val");
                 Load = String.valueOf(load);
                 msg1.setContent(Load);
                 myAgent.send(msg1);
             }
             else if(val == 2) {
                 block();}
             }}
     class rec_bid_Ok extends CyclicBehaviour{
         public rec_bid_Ok(Agent a){
             super(a);}
         @Override
         public void action() {
             MessageTemplate mt=MessageTemplate.not(MessageTemplate.MatchSender(myAgent.getAID()));
             ACLMessage msg=myAgent.receive(mt);
             if(msg != null){
                 if (msg.getConversationId() == "LA1bidok"){
                     level = 2;
                     block();
                 }}}}
     class rec_load_Ok extends CyclicBehaviour{
         public rec_load_Ok(Agent a){
             super(a);}
         @Override
         public void action() {
             MessageTemplate mt=MessageTemplate.not(MessageTemplate.MatchSender(myAgent.getAID()));
             ACLMessage msg=myAgent.receive(mt);
             if(msg != null){
                 if (msg.getConversationId() == "LA1loadok"){
                     val = 2;
                     block();
                 }}}}
     class rec_result extends CyclicBehaviour{
         public rec_result(Agent a){
             super(a);}
         @Override
         public void action() {
             MessageTemplate mt=MessageTemplate.not(MessageTemplate.MatchSender(myAgent.getAID()));
             ACLMessage msg=myAgent.receive(mt);
             if(msg != null){
                 if (msg.getConversationId() == "LA1result"){
                     results = msg.getContent();
                     block();
                 }}}}
}
```

B.5. TEMA code

```
package Tema;
import java.io.File;
public class TEMA extends Agent
{
    int l1 = 0, l2 = 0, l3 = 0, g1 = 0, g2 = 0, g3 = 0, g4 = 0, g5 = 0;
    int b1 = 0, b2 = 0, b3 = 0, o1 = 0, o2 = 0, o3 = 0, o4 = 0, o5 = 0;
    float bid1, load1, bid2, load2, bid3, load3;
    float offer1, gen1, offer2, gen2, offer3, gen3, offer4, gen4, offer5, gen5;
    String Bid1, Load1, Bid2, Load2, Bid3, Load3;
    String Offer1, Gen1, Offer2, Gen2, Offer3, Gen3, Offer4, Gen4, Offer5, Gen5;
    float GSP = 0, GBP = 0;
    String gsp, gbp;
    float OPF;
    String opf;
    float rbid1, rbid2, rbid3, roffer1, roffer2, roffer3, roffer4, roffer5;
    String Rbid1, Rbid2, Rbid3, Roffer1, Roffer2, Roffer3, Roffer4, Roffer5;
    protected void setup()
        {
          ParallelBehaviour parallelBehaviour = new ParallelBehaviour();
            parallelBehaviour.addSubBehaviour(new send_start(this));
            parallelBehaviour.addSubBehaviour(new rec_load_LA1(this));
            parallelBehaviour.addSubBehaviour(new rec_load_LA2(this));
            parallelBehaviour.addSubBehaviour(new rec_load_FA1(this));
            parallelBehaviour.addSubBehaviour(new rec_bid_LA1(this));
            parallelBehaviour.addSubBehaviour(new rec_bid_LA2(this));
            parallelBehaviour.addSubBehaviour(new rec_bid_FA1(this));
            parallelBehaviour.addSubBehaviour(new rec_gen_GA1(this));
            parallelBehaviour.addSubBehaviour(new rec_gen_GA2(this));
            parallelBehaviour.addSubBehaviour(new rec_gen_GA3(this));
            parallelBehaviour.addSubBehaviour(new rec_gen_FA2(this));
            parallelBehaviour.addSubBehaviour(new rec_offer_FA3(this));
            parallelBehaviour.addSubBehaviour(new rec_offer_GA1(this));
            parallelBehaviour.addSubBehaviour(new rec_offer_GA2(this));
            parallelBehaviour.addSubBehaviour(new rec_offer_GA3(this));
            parallelBehaviour.addSubBehaviour(new rec_offer_FA2(this));
            parallelBehaviour.addSubBehaviour(new rec_offer_FA3(this));
            parallelBehaviour.addSubBehaviour(new send_Grid(this));
            parallelBehaviour.addSubBehaviour(new rec_GSP(this));
            parallelBehaviour.addSubBehaviour(new rec_GBP(this));
            parallelBehaviour.addSubBehaviour(new intrn_clc(this));
            parallelBehaviour.addSubBehaviour(new rec_OPF(this));
            parallelBehaviour.addSubBehaviour(new run_int_OPF(this));
            parallelBehaviour.addSubBehaviour(new send_results(this));
            this.addBehaviour(parallelBehaviour);
        }
    class send_start extends CyclicBehaviour{
        public send_start(Agent a){
            super(a);}
        @Override
        public void action() {
            if(l1 == 0 && b1 == 0) {
                ACLMessage msg= new ACLMessage(ACLMessage.INFORM);
                msg.setConversationId("TEMAStart");
                myAgent.send(msg);
                block();
            }}}
    class rec_load_LA1 extends CyclicBehaviour{
        public rec_load_LA1(Agent a){
            super(a);}
        @Override
        public void action() {
            MessageTemplate mt=MessageTemplate.not(MessageTemplate.MatchSender(myAgent.getAID()));
            ACLMessage msg=myAgent.receive(mt);
            if(msg != null){
                if (msg.getConversationId() == "LA1val"){
                    l1 = 1;
                    load1 = msg.getContent();
                    load1 = Float.valueOf(load1);

                    ACLMessage msg1= new ACLMessage(ACLMessage.INFORM);
                    msg1.addReceiver(new AID("LA1", AID.ISLOCALNAME));
                    msg1.setConversationId("LA1loadok");
                    myAgent.send(msg1);
                    block();
                }}}}
    class rec_load_LA2 extends CyclicBehaviour{
        public rec_load_LA2(Agent a){
            super(a);}
        @Override
        public void action() {
            MessageTemplate mt=MessageTemplate.not(MessageTemplate.MatchSender(myAgent.getAID()));
            ACLMessage msg=myAgent.receive(mt);
            if(msg != null){
                if (msg.getConversationId() == "LA2val"){
                    l2 = 1;
                    load2 = msg.getContent();
                    load2 = Float.valueOf(load2);

                    ACLMessage msg1= new ACLMessage(ACLMessage.INFORM);
                    msg1.addReceiver(new AID("LA2", AID.ISLOCALNAME));
                    msg1.setConversationId("LA2loadok");
                    myAgent.send(msg1);
                    block();
                }}}}
    class rec_load_FA1 extends CyclicBehaviour{
        public rec_load_FA1(Agent a){
            super(a);}
        @Override
        public void action() {
            MessageTemplate mt=MessageTemplate.not(MessageTemplate.MatchSender(myAgent.getAID()));
            ACLMessage msg=myAgent.receive(mt);
            if(msg != null){
                if (msg.getConversationId() == "FA1val"){
                    l3 = 1;
                    load3 = msg.getContent();
                    load3 = Float.valueOf(load3);

                    ACLMessage msg1= new ACLMessage(ACLMessage.INFORM);
                    msg1.addReceiver(new AID("FA1", AID.ISLOCALNAME));
                    msg1.setConversationId("FA1loadok");
                    myAgent.send(msg1);
                    block();
                }}}}
```

```
class rec_bid_LA1 extends CyclicBehaviour{
    public rec_bid_LA1(Agent a){
        super(a);}
    @Override
    public void action() {
        MessageTemplate mt=MessageTemplate.not(MessageTemplate.MatchSender(myAgent.getAID()));
        ACLMessage msg=myAgent.receive(mt);
        if(msg != null){
            if (msg.getConversationId() == "LA1Bid"){
                b1 = 1;
                Bid1 = msg.getContent();
                bid1 = Float.valueOf(Bid1);

                ACLMessage msg1= new ACLMessage(ACLMessage.INFORM);
                msg1.addReceiver(new AID("LA1", AID.ISLOCALNAME));
                msg1.setConversationId("LA1bidok");
                myAgent.send(msg1);
                block();
            }}}
class rec_bid_LA2 extends CyclicBehaviour{
    public rec_bid_LA2(Agent a){
        super(a);}
    @Override
    public void action() {
        MessageTemplate mt=MessageTemplate.not(MessageTemplate.MatchSender(myAgent.getAID()));
        ACLMessage msg=myAgent.receive(mt);
        if(msg != null){
            if (msg.getConversationId() == "LA2Bid"){
                b2 = 1;
                Bid2 = msg.getContent();
                bid2 = Float.valueOf(Bid2);

                ACLMessage msg1= new ACLMessage(ACLMessage.INFORM);
                msg1.addReceiver(new AID("LA2", AID.ISLOCALNAME));
                msg1.setConversationId("LA2bidok");
                myAgent.send(msg1);
                block();
            }}}
class rec_bid_FA1 extends CyclicBehaviour{
    public rec_bid_FA1(Agent a){
        super(a);}
    @Override
    public void action() {
        MessageTemplate mt=MessageTemplate.not(MessageTemplate.MatchSender(myAgent.getAID()));
        ACLMessage msg=myAgent.receive(mt);
        if(msg != null){
            if (msg.getConversationId() == "FA1Bid"){
                b3 = 1;
                Bid3 = msg.getContent();
                bid3 = Float.valueOf(Bid3);

                ACLMessage msg1= new ACLMessage(ACLMessage.INFORM);
                msg1.addReceiver(new AID("FA1", AID.ISLOCALNAME));
                msg1.setConversationId("FA1bidok");
                myAgent.send(msg1);
                block();
            }}}
class rec_gen_GA1 extends CyclicBehaviour{
    public rec_gen_GA1(Agent a){
        super(a);}
    @Override
    public void action() {
        MessageTemplate mt=MessageTemplate.not(MessageTemplate.MatchSender(myAgent.getAID()));
        ACLMessage msg=myAgent.receive(mt);
        if(msg != null){
            if (msg.getConversationId() == "GA1val"){
                g1 = 1;
                Gen1 = msg.getContent();
                gen1 = Float.valueOf(Gen1);

                ACLMessage msg1= new ACLMessage(ACLMessage.INFORM);
                msg1.addReceiver(new AID("GA1", AID.ISLOCALNAME));
                msg1.setConversationId("GA1genok");
                myAgent.send(msg1);
                block();
            }}}
class rec_gen_GA3 extends CyclicBehaviour{
    public rec_gen_GA3(Agent a){
        super(a);}
    @Override
    public void action() {
        MessageTemplate mt=MessageTemplate.not(MessageTemplate.MatchSender(myAgent.getAID()));
        ACLMessage msg=myAgent.receive(mt);
        if(msg != null){
            if (msg.getConversationId() == "GA3val"){
                g3 = 1;
                Gen3 = msg.getContent();
                gen3 = Float.valueOf(Gen3);

                ACLMessage msg1= new ACLMessage(ACLMessage.INFORM);
                msg1.addReceiver(new AID("GA3", AID.ISLOCALNAME));
                msg1.setConversationId("GA3genok");
                myAgent.send(msg1);
                block();
            }}}
class rec_gen_FA2 extends CyclicBehaviour{
    public rec_gen_FA2(Agent a){
        super(a);}
    @Override
    public void action() {
        MessageTemplate mt=MessageTemplate.not(MessageTemplate.MatchSender(myAgent.getAID()));
        ACLMessage msg=myAgent.receive(mt);
        if(msg != null){
            if (msg.getConversationId() == "FA2val"){
                g4 = 1;
                Gen4 = msg.getContent();
                gen4 = Float.valueOf(Gen4);

                ACLMessage msg1= new ACLMessage(ACLMessage.INFORM);
                msg1.addReceiver(new AID("FA2", AID.ISLOCALNAME));
                msg1.setConversationId("FA2genok");
                myAgent.send(msg1);
                block();
            }}}
```

```
class rec_gen_FA3 extends CyclicBehaviour{
    public rec_gen_FA3(Agent a){
        super(a);}
    @Override
    public void action() {
        MessageTemplate mt=MessageTemplate.not(MessageTemplate.MatchSender(myAgent.getAID()));
        ACLMessage msg=myAgent.receive(mt);
        if(msg != null){
            if (msg.getConversationId() == "FA3val"){
                g5 = 1;
                Gen5 = msg.getContent();
                gen5 = Float.valueOf(Gen5);

                ACLMessage msg1= new ACLMessage(ACLMessage.INFORM);
                msg1.addReceiver(new AID("FA3", AID.ISLOCALNAME));
                msg1.setConversationId("FA3genok");
                myAgent.send(msg1);
                block();
            }}}
class rec_offer_GA1 extends CyclicBehaviour{
    public rec_offer_GA1(Agent a){
        super(a);}
    @Override
    public void action() {
        MessageTemplate mt=MessageTemplate.not(MessageTemplate.MatchSender(myAgent.getAID()));
        ACLMessage msg=myAgent.receive(mt);
        if(msg != null){
            if (msg.getConversationId() == "GA1Offer"){
                o1 = 1;
                Offer1 = msg.getContent();
                offer1 = Float.valueOf(Offer1);

                ACLMessage msg1= new ACLMessage(ACLMessage.INFORM);
                msg1.addReceiver(new AID("GA1", AID.ISLOCALNAME));
                msg1.setConversationId("GA1offerok");
                myAgent.send(msg1);
                block();
            }}}
class rec_offer_GA2 extends CyclicBehaviour{
    public rec_offer_GA2(Agent a){
        super(a);}
    @Override
    public void action() {
        MessageTemplate mt=MessageTemplate.not(MessageTemplate.MatchSender(myAgent.getAID()));
        ACLMessage msg=myAgent.receive(mt);
        if(msg != null){
            if (msg.getConversationId() == "GA2Offer"){
                o2 = 1;
                Offer2 = msg.getContent();
                offer2 = Float.valueOf(Offer2);

                ACLMessage msg1= new ACLMessage(ACLMessage.INFORM);
                msg1.addReceiver(new AID("GA2", AID.ISLOCALNAME));
                msg1.setConversationId("GA2offerok");
                myAgent.send(msg1);
                block();
            }}}
class rec_offer_GA3 extends CyclicBehaviour{
    public rec_offer_GA3(Agent a){
        super(a);}
    @Override
    public void action() {
        MessageTemplate mt=MessageTemplate.not(MessageTemplate.MatchSender(myAgent.getAID()));
        ACLMessage msg=myAgent.receive(mt);
        if(msg != null){
            if (msg.getConversationId() == "GA3Offer"){
                o3 = 1;
                Offer3 = msg.getContent();
                offer3 = Float.valueOf(Offer3);

                ACLMessage msg1= new ACLMessage(ACLMessage.INFORM);
                msg1.addReceiver(new AID("GA3", AID.ISLOCALNAME));
                msg1.setConversationId("GA3offerok");
                myAgent.send(msg1);
                block();
            }}}
class rec_offer_FA2 extends CyclicBehaviour{
    public rec_offer_FA2(Agent a){
        super(a);}
    @Override
    public void action() {
        MessageTemplate mt=MessageTemplate.not(MessageTemplate.MatchSender(myAgent.getAID()));
        ACLMessage msg=myAgent.receive(mt);
        if(msg != null){
            if (msg.getConversationId() == "FA2Offer"){
                o4 = 1;
                Offer4 = msg.getContent();
                offer4 = Float.valueOf(Offer4);

                ACLMessage msg1= new ACLMessage(ACLMessage.INFORM);
                msg1.addReceiver(new AID("FA2", AID.ISLOCALNAME));
                msg1.setConversationId("FA2offerok");
                myAgent.send(msg1);
                block();
            }}}
class rec_offer_FA3 extends CyclicBehaviour{
    public rec_offer_FA3(Agent a){
        super(a);}
    @Override
    public void action() {
        MessageTemplate mt=MessageTemplate.not(MessageTemplate.MatchSender(myAgent.getAID()));
        ACLMessage msg=myAgent.receive(mt);
        if(msg != null){
            if (msg.getConversationId() == "FA3Offer"){
                o5 = 1;
                Offer5 = msg.getContent();
                offer5 = Float.valueOf(Offer5);

                ACLMessage msg1= new ACLMessage(ACLMessage.INFORM);
                msg1.addReceiver(new AID("FA3", AID.ISLOCALNAME));
                msg1.setConversationId("FA3offerok");
                myAgent.send(msg1);
                block();
            }}}
```

```
class send_Grid extends CyclicBehaviour{
    public send_Grid(Agent a){
        super(a);}
    @Override
    public void action() {
        if(GSP == 0 && GBP == 0) {
            ACLMessage msg= new ACLMessage(ACLMessage.INFORM);
            msg.addReceiver(new AID("DSO", AID.ISLOCALNAME));
            msg.setConversationId("DSOPrices");
            myAgent.send(msg);
        }
        else {
            block();
        }}}
class rec_GSP extends CyclicBehaviour{
    public rec_GSP(Agent a){
        super(a);}
    @Override
    public void action() {
            MessageTemplate mt=MessageTemplate.not(MessageTemplate.MatchSender(myAgent.getAID()));
            ACLMessage msg=myAgent.receive(mt);
            if(msg != null){
                if (msg.getConversationId() == "GSP"){
                    gsp = msg.getContent();
                    GSP = Float.valueOf(gsp);
                    block();
                }}}}
class rec_GBP extends CyclicBehaviour{
    public rec_GBP(Agent a){
        super(a);}
    @Override
    public void action() {
            MessageTemplate mt=MessageTemplate.not(MessageTemplate.MatchSender(myAgent.getAID()));
            ACLMessage msg=myAgent.receive(mt);
            if(msg != null){
                if (msg.getConversationId() == "GBP"){
                    gbp = msg.getContent();
                    GBP = Float.valueOf(gbp);
                    block();
                }}}}
class intrn_clc extends CyclicBehaviour{
    public intrn_clc(Agent a){
        super(a);}
    @Override
    public void action() {
        if (l1 != 0 && l2 != 0 && l3 != 0 && g1 != 0 && g2 != 0 && g3 != 0
                && g4 != 0 && 5 != 0 && b1 != 0 && b2 != 0 && b3 != 0 && o1 != 0
                && o2 != 0 && o3 != 0 && o4 != 0 && o5 != 0) {
            ACLMessage msg= new ACLMessage(ACLMessage.INFORM);
            msg.addReceiver(new AID("DSO", AID.ISLOCALNAME));
            msg.setConversationId("OPF");
            msg.setContent(gsp);
            myAgent.send(msg);
            if(OPF!=0) {
                block();
            }}}}
class rec_OPF extends CyclicBehaviour{
    public rec_OPF(Agent a){
        super(a);}
    @Override
    public void action() {
        MessageTemplate mt=MessageTemplate.not(MessageTemplate.MatchSender(myAgent.getAID()));
        ACLMessage msg=myAgent.receive(mt);
        if(msg != null){
            if (msg.getConversationId() == "OPFresults"){
                //Receiving OPF results
                opf = msg.getContent();
                OPF = Float.valueOf(opf);
            }}}}
class run_int_OPF extends CyclicBehaviour{
    public run_int_OPF(Agent a){
        super(a);}
    @Override
    public void action() {
        if (OPF == 1) {
            //Running OPF based double auction
            //Schedule alpha parameter
            //Simulations done in the Matlab
            OPF = 2;
            block();
        }}}
```

```
    class send_results extends CyclicBehaviour{
        public send_results(Agent a){
            super(a);}
        @Override
        public void action() {
            if (OPF == 2) {
            ACLMessage msg1= new ACLMessage(ACLMessage.INFORM);
            msg1.addReceiver(new AID("LA1", AID.ISLOCALNAME));
            msg1.setConversationId("LA1result");
            Rbid1 = Float.toString(rbid1);
            msg1.setContent(Rbid1);
            myAgent.send(msg1);
            ACLMessage msg2= new ACLMessage(ACLMessage.INFORM);
            msg2.addReceiver(new AID("LA2", AID.ISLOCALNAME));
            msg2.setConversationId("LA2result");
            Rbid2 = Float.toString(rbid2);
            msg2.setContent(Rbid2);
            myAgent.send(msg2);
            ACLMessage msg3= new ACLMessage(ACLMessage.INFORM);
            msg3.addReceiver(new AID("FA1", AID.ISLOCALNAME));
            msg3.setConversationId("FA1result");
            Rbid3 = Float.toString(rbid3);
            msg3.setContent(Rbid3);
            myAgent.send(msg3);
            ACLMessage msg4= new ACLMessage(ACLMessage.INFORM);
            msg4.addReceiver(new AID("GA1", AID.ISLOCALNAME));
            msg4.setConversationId("GA1result");
            Roffer1 = Float.toString(roffer1);
            msg4.setContent(Roffer1);
            myAgent.send(msg4);
            ACLMessage msg5= new ACLMessage(ACLMessage.INFORM);
            msg5.addReceiver(new AID("GA2", AID.ISLOCALNAME));
            msg5.setConversationId("GA2result");
            Roffer2 = Float.toString(roffer2);
            msg5.setContent(Roffer2);
            myAgent.send(msg5);
            ACLMessage msg6= new ACLMessage(ACLMessage.INFORM);
            msg6.addReceiver(new AID("GA3", AID.ISLOCALNAME));
            msg6.setConversationId("GA3result");
            Roffer3 = Float.toString(roffer3);
            msg6.setContent(Roffer3);
            myAgent.send(msg6);
            ACLMessage msg7= new ACLMessage(ACLMessage.INFORM);
            msg7.addReceiver(new AID("FA2", AID.ISLOCALNAME));
            msg7.setConversationId("FA2result");
            Roffer4 = Float.toString(roffer4);
            msg7.setContent(Roffer4);
            myAgent.send(msg7);
            ACLMessage msg8= new ACLMessage(ACLMessage.INFORM);
            msg8.addReceiver(new AID("FA3", AID.ISLOCALNAME));
            msg8.setConversationId("FA3result");
            Roffer5 = Float.toString(roffer5);
            msg8.setContent(Roffer5);
            myAgent.send(msg8);
        }}}
}
```