# VidyutVanika: A Reinforcement Learning Based Broker Agent for a Power Trading Competition

by

Susobhan Ghosh, Easwar Subramanian, Sanjay P. Bhat, Sujit Prakash Gujar, Praveen Paruchuri

in

*Thirty-Third AAAI Conference on Artificial Intelligence*
(*AAAI-2019*)

Hilton Hawaiian Village, Honolulu, Hawaii, USA

Report No: IIIT/TR/2019/-1

Centre for Visual Information Technology
International Institute of Information Technology
Hyderabad - 500 032, INDIA
January 2019

# VidyutVanika: A Reinforcement Learning Based Broker Agent for a Power Trading Competition

**Susobhan Ghosh [1], Easwar Subramanian [2], Sanjay P. Bhat [2], Sujit Gujar [1], Praveen Paruchuri [1]**
[1] Machine Learning Lab, IIIT Hyderabad, India {susobhan.ghosh@research.iiit.ac.in}, {sujit.gujar, praveen.p}@iiit.ac.in
[2] Tata Consultancy Services, TCS Innovation Labs, Hyderabad, India {easwar.subramanian, sanjay.bhat}@tcs.com

## Abstract

A smart grid is an efficient and sustainable energy system that integrates diverse generation entities, distributed storage capacity, and smart appliances and buildings. A smart grid brings new kinds of participants in the energy market served by it, whose effect on the grid can only be determined through high fidelity simulations. Power TAC offers one such simulation platform using real-world weather data and complex state-of-the-art customer models. In Power TAC, autonomous energy brokers compete to make profits across tariff, wholesale and balancing markets while maintaining the stability of the grid. In this paper, we design an autonomous broker VidyutVanika, the runner-up in the 2018 Power TAC competition. VidyutVanika relies on reinforcement learning (RL) in the tariff market and dynamic programming in the wholesale market to solve modified versions of known Markov Decision Process (MDP) formulations in the respective markets. The novelty lies in defining the reward functions for MDPs, solving these MDPs, and the application of these solutions to real actions in the market. Unlike previous participating agents, VidyutVanika uses a neural network to predict the energy consumption of various customers using weather data. We use several heuristic ideas to bridge the gap between the restricted action spaces of the MDPs and the much more extensive action space available to VidyutVanika. These heuristics allow VidyutVanika to convert near-optimal fixed tariffs to time-of-use tariffs aimed at mitigating transmission capacity fees, spread out its orders across several auctions in the wholesale market to procure energy at a lower price, more accurately estimate parameters required for implementing the MDP solution in the wholesale market, and account for wholesale procurement costs while optimizing tariffs. We use Power TAC 2018 tournament data and controlled experiments to analyze the performance of VidyutVanika, and illustrate the efficacy of the above strategies.

## Introduction

A *smart grid* is an evolved electrical system that manages electricity demand in a sustainable, reliable and economical manner, built on advanced infrastructure and tuned to facilitate the integration of all the entities involved. [1] With the efforts to move to sustainable energy sources, smart grids in theory, offer a stable and efficient mechanism to manage such systems (Speer et al. 2015). Smart grids also offer the flexibility of dynamically changing tariffs to the customers. Electricity distributing agencies operating in the smart grid, which we refer to as *brokers*, can signal the supply-demand imbalance to the market through dynamic pricing strategies, while simultaneously reducing the overhead costs of their customers by buying energy in bulk from generating companies. However, there are multiple challenges in the operationalization of smart grids, like managing highly fluctuating supply-demand scenarios, engaging stakeholders with ulterior motives, and handling automation failures of participating entities.

In order to foresee such problems and examine potential solutions, Power TAC (Ketter, Collins, and Weerdt 2017) provides an open source simulator platform that replicates crucial elements of a smart grid system and allows large-scale experimentation. The simulation encourages the development of autonomous broker agents that aim at making a profit by offering electricity tariffs to customers in a retail (or tariff) market, and trading energy in a competitive wholesale market, while carefully balancing their supply and demand. To this end, a Power Trading Agent Competition (Power TAC) (Ketter, Collins, and Weerdt 2017) is held annually.

Machine Learning and Game Theory-based strategies are essential for such broker agents to dynamically price tariffs and predict customer usage while simultaneously placing bids in wholesale auctions. In the past, some broker agents have used MDP to model strategies in the tariff market (Cuevas, Rodriguez-Gonzalez, and De Cote 2017; Yang et al. 2018), and wholesale market (Urieli and Stone 2014; 2016a; Reddy and Veloso 2011), while others have employed genetic algorithm, fuzzy-logic and tailored heuristics for the same (Özdemir and Unland 2018a; Rúbio et al. 2015; Liefers, Hoogland, and La Poutré 2014). Less attention has been paid to utilize weather data, having been used only to predict wholesale prices (Chowdhury 2016). Very few contributions have been made towards modeling the entire system as a reinforcement learning problem (Urieli and Stone 2016a), due to its complexity.

The goal of this paper is to design a learning broker with the following objectives: (i) React to competing tariffs (ii) Increase market share, i.e., subscribed customers (iii) Decrease transmission capacity costs (iv) Decrease costs of



[1] http://new.abb.com/smartgrids/what-is-a-smart-grid

energy procurement. We use different MDPs for our tariff and wholesale market strategy. Though the MDPs are motivated by (Urieli and Stone 2014) and (Cuevas, Rodriguez-Gonzalez, and De Cote 2017), our novelty lies in their reward structure, solution, and application of those solutions. These are supplemented by a Neural Network based usage predictor, that also utilizes weather data. Our broker, VidyutVanika, referred as $VV$ throughout the paper, was the runner-up in Power TAC 2018 Finals. We illustrate the efficacy of our strategies through different statistics from the competition as well as controlled offline experiments.

## Power TAC Game Description

This section provides a brief overview of the annual Power Trading Agent Competition (Power TAC) tournament. For the detailed specifications of the various components of Power TAC, please refer (Ketter, Collins, and Weerdt 2017).

In Power TAC, multiple teams deploy autonomous electricity broker agents which have to operate and compete in three smart electricity markets. The tournament consists of numerous games played between participating broker agents in different configurations. The duration of each game is around 60 simulation days. The simulation time is discretized into time slots. Each such time slot represents an hour, and is played out in 5 seconds of real time. Hence, the duration of a single game in the tournament is roughly around two hours of real time.

A broker agent in Power TAC develops a subscriber base by offering attractive bilateral tariff contracts, and simultaneously attempts to fulfill its subscribers' energy requirements by trading in the wholesale market. Typically, a broker agent performs three functions, (i) purchase from, or sell power to, its subscriber base in the *retail* (or *tariff*) *market*; (ii) purchase or sell power in the *wholesale market*; and (iii) rectify any supply-demand imbalance within its portfolio through the *balancing market*.

The *tariff market* consists of customers of three different power types, namely, consumers, producers and storage. Consumers include offices, housing complexes, hospitals and villages. A subset of these consumers accept curtailment of their usage in exchange for discounted tariffs. Producers in the tariff market use renewable sources such as solar or wind to generate electricity. The Storage customers possess storage capacity in the form of batteries or electric vehicles connected to the smart grid. Broker agents compete in the retail market to draw customers into their subscriber base by offering attractive tariffs that are power type specific. Tariffs could be offered with fixed or variable rates, and could be tiered or based on time of use. The *wholesale market* in the Power TAC setting is a 'day-ahead' market that is largely supplied by a single power generation company. At any given time, broker agents participate in twenty-four periodic double auctions to buy or sell power in the wholesale market for a future time slot that could be one to twenty-four hours away. Broker agents participate in the *balancing market* by exercising control over a customer's storage infrastructure to store or withdraw power as needed, and by offering suitable tariffs to customers accepting curtailment.

To support the functioning of brokers agents, the Power TAC environment publishes a variety of information to all the participating broker agents in a game. Before the start of a game, a 14-day simulation exercise, called the bootstrap period, is organized in which the distribution utility (DU) is the only participant. The data generated during the bootstrap period contains the name, characteristics and consumption profile of all retail market customers, wholesale market data pertaining to average cleared price and quantity, and weather data of the geographical location of the customer base, all at an hourly frequency. During the game, the Power TAC environment publishes identities of competing broker agents, tariff updates that includes new, revoked and superseding tariffs published by competing broker agents, wholesale market clearing data, aggregate energy consumption data for every time slot, and weather reports and forecasts.

The goal of a broker agent in a Power TAC game is to deploy suitable strategies in the wholesale, tariff and balancing markets to achieve a healthy *cash position* at the end of the game. Apart from cash flows resulting from trades in the wholesale and retail markets, a broker's final cash position is also affected by *balancing fees* for failure to maintain a balance between supply and demand, tariff publication fees, distribution fees, bank interest and *transmission capacity fees* for contributions to peak demand events by a broker agent's subscriber base. The cash position of a broker is aggregated across all the games played and then normalized in order to determine the winner of the competition.

## Overview of Broker Agent

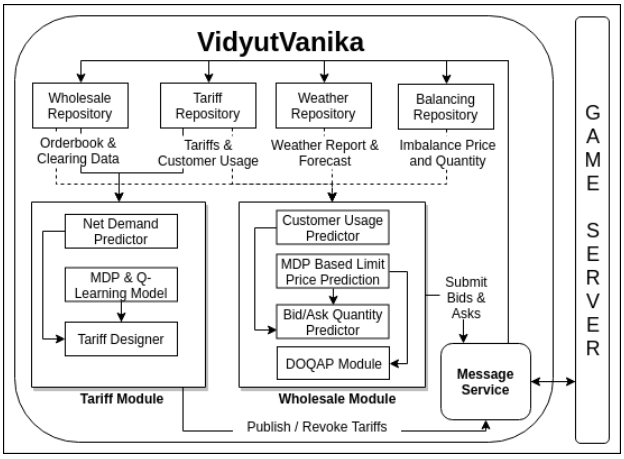


Figure 1: Architecture of VidyutVanika

In this section, we present an overview of our autonomous broker agent, $VV$. As shown in Figure 1, $VV$ consists of two main modules, namely, *Tariff Module* (TM) and *Wholesale Module* (WM). TM is responsible for publishing and revoking tariffs in the tariff (or retail) market. WM generates bids/asks to purchase/sell energy contracts in the wholesale market. $VV$ doesn't actively participate in the balancing market. Tariff design is accomplished by formulating a Markov decision process (MDP) (Puterman 1994),

which we approximately solve using Q-learning (Watkins and Dayan 1992). We model the bidding problem in the wholesale market as a separate MDP, which we solve using dynamic programming (Bellman 2013).

In addition to these two modules, $VV$ incorporates a Customer Usage Predictor (CUP) submodule built using neural networks (NN) to predict the usage of all subscribed customers in a future time slot, by using weather forecasts and past usage pattern of each customer. $VV$ aggregates the predicted usage across all its subscribed customers to estimate the amount of energy to be procured in the wholesale market. Doing so helps $VV$ reduce the imbalance on its portfolio. We note that, to the best of our knowledge, $VV$ is the first broker agent to use NN with the weather data to predict customer usage in Power TAC competition [2]. In the following subsections, we discuss each module of $VV$ in more detail.

## Tariff Module (TM)

Throughout a game, $VV$ maintains two active time-of-use (TOU) tariffs in the tariff market, namely (i) MDPTOU and (ii) WeeklyTOU. MDPTOU is the result of solving an MDP problem using Q-learning, and is revised every twenty-four hours. WeeklyTOU is an empirically determined, fixed weekly TOU tariff, which remains active throughout the duration of the game. If MDPTOU makes losses for a sustained period of time, $VV$ revokes it and falls back upon WeeklyTOU, as it is empirically proven to be reliable.

Generating MDPTOU is a two-step process - (1) Generate a Fixed Price Tariff (FPT) by solving an MDP using Q-learning; (2) Convert the FPT to a TOU tariff for consumption customers by predicting the overall demand profile for the tariff market over the next 24 time slots. Both the steps are described in detail in the following sub-subsections.

### MDP & Q-Learning Model (MDPQLM)

Our Tariff MDP formulation is primarily motivated from the work of (Cuevas, Rodriguez-Gonzalez, and De Cote 2017). At any simulation time $t$, the state $s_t$ of the MDP is a quadruple that captures four features of the tariff market. The first feature is rationality of the tariff market which is decided based on whether the highest production tariff is lower or higher than the lowest consumption tariff. The second is the portfolio status of our broker agent $VV$ which could be surplus, balanced or deficit depending on the difference between the amount of energy acquired and committed in the tariff market at time $t$. The third and fourth features rank the $VV$'s current consumption and production tariffs with respect to prevailing tariffs of other competing broker agents. In total, there are 96 possible states in the MDP. The action $a_t$, at time $t$, is chosen from a set $A$ of 8 actions, each of which lets $VV$ modify its previous production and consumption tariff in a specific fashion. These include different combinations of increasing, decreasing or tempering the latest prevailing production or consumption tariff of broker $VV$. A detailed description of the state and action space of the MDP can be found in the supplement.

[2] based on past Power TAC agent publications

The key novelty in our MDP formulation is the reward structure c.f. Cuevas, Rodriguez-Gonzalez, and De Cote. The idea behind the reward structure is to capture the net profit made by $VV$ when it incurs no balancing charge. Thus, the reward at time $t$ is given by:

$$r_t = \theta_{t,C} P_{t,C} - \theta_{t,P} P_{t,P} - \theta_{t,W} W_t \quad (1)$$

The first term in Equation 1 represents the revenue generated by selling energy $\theta_{t,C}$ at the tariff $P_{t,C}$ to consumers of $VV$ at time $t$. Similarly, the second term represents the amount paid to producers of $VV$ for procuring energy $\theta_{t,P}$ at the tariff $P_{t,P}$. The third term in represents the amount paid in the wholesale market to satisfy the net unfulfilled demand $\theta_{t,W} = \theta_{t,C} - \theta_{t,P}$ at unit wholesale procurement cost $W_t$.

We construct a Q-table using Q-learning to solve the aforementioned MDP. For a state-action pair $(s_t, a_t)$, the Q-learning update rule with learning rate $\alpha$ and discount rate $\gamma$ is given by

$$\hat{Q}(s_t, a_t) \leftarrow (1-\alpha_t)\hat{Q}(s_t, a_t) + \alpha[r_t + \gamma \max_a \hat{Q}(s_{t+1}, a)],$$

where $r_t$ is the reward obtained at time $t$ for taking action $a_t$ in state $s_t$. A Q-table is constructed through a training process in which $VV$ plays 100 games of every configuration against broker agents from past the Power TAC tournaments across 9 different game configurations. For each configuration, $VV$ starts with a zero-initialized Q-table and updates the Q-table entries, across 100 games according to the update rule specified above. While playing a game in the real tournament, at any tariff publication time $t$, being in state $s_t$, $VV$ simply chooses an action $a_t$ greedily according to $a_t = \text{argmax}_{a \in A} Q(s_t, a)$. The action thus chosen translates into a production FPT $P_{t,P}$ and a consumption FPT $P_{t,C}$. While the production FPT is published without any change, the consumption FPT is modified to generate MDPTOU as explained in *Tariff Designer* (TaD).

### Net Demand Predictor (NDP)

Before converting the consumption FPT into MDPTOU, $VV$ first estimates the overall net usage/demand of the tariff market for all future twenty-four time slots. To this end, at a simulation time slot $t$, $VV$ estimates the net demand $\hat{\mathrm{D}}_{t+k}, k \in \{1, \ldots, 24\}$ as a weighted average of two historical net demand values, namely, net demand $\mathrm{D}_{t+k-24}$ observed at the same time slot of the previous day and the net demand $\mathrm{D}_{t+k-168}$ observed during the same time-slot of the same day of the previous week. This enables $VV$ to capture the recent customer usage patterns in such a sensitive market while also utilizing the weekly trends. More specifically, we have,

$$\hat{\mathrm{D}}_{t+k} = \beta \mathrm{D}_{t+k-24} + (1-\beta)\mathrm{D}_{t+k-168} \quad (2)$$

where $\beta \in [0, 1]$ is a fixed parameter.

### Tariff Designer (TaD)

Once the estimate of the net demand for the next twenty-four time slots is obtained from NDP, MDPTOU for a time slot $k$ hours ahead of the current time slot $t$ is computed as:

$$\pi_{t+k} = P_{t,C} + \rho \left( \hat{\mathrm{D}}_{t+k,T} - \frac{\sum_{j=1}^{24} \hat{\mathrm{D}}_{t+j,T}}{24} \right), \quad (3)$$

where $\rho$ is an empirically determined constant and $k \in \{1, \ldots, 24\}$. Equation (3) proposes MDPTOU for a twenty-four hour time horizon. Observe that the tariff rate in Equation (3) at a time slot $t$ modifies the fixed price consumption tariff $P_{t,C}$ provided by the Q-learning algorithm by an amount that is proportional to the excess estimated demand in that time slot over the mean estimated demand over the 24-hour period starting at $t$. The second term in Equation (3) closely resembles the manner in which the *transmission capacity fees* are calculated in the Power TAC simulation (see section 7.2 of (Ketter, Collins, and Weerdt 2017)). As a result, MDPTOU serves to mitigate the effect of transmission capacity fees that the broker incurs in two ways. First, it encourages the customers to shift some of their usage away from expected peak demand time-slot(s). Second, the excess over the consumption FPT charged to a customer is in proportion to that customer's contribution to the expected net demand profile, and this helps offset some of the transmission capacity fees that will actually result from that customer's usage profile.

Together, MDPQLM, NDP and TaD enable $VV$ to select actions in the large decision space of TOU tariffs by solving an MDP with a much smaller action space.

## Wholesale Module (WM)

In order to balance the future net usage in its tariff portfolio, $VV$ participates in the wholesale market auctions by placing bids/asks of the form (*energy amount, limit-price*). To predict this net usage for a future time-slot, it uses a Neural Network (NN) based Customer Usage Predictor (CUP). $VV$ then determines the *limit-price* using the Limit Price Predictor (LPP), and the *energy amount* using the Bid/Ask Quantity Predictor (BAQP). Overall, the wholesale market strategy of $VV$ comprises of four major submodules - (i) Customer Usage Predictor (CUP), (ii) Limit Price Predictor (LPP), (iii) Bid/Ask Quantity Predictor (BAQP), and (iv) Dummy Order Quantity and Price (DOQAP) Module

### Customer Usage Predictor (CUP)

CUP is responsible for predicting the net usage of the broker's tariff portfolio for a future target time-slot $t$, by summing over the predicted usage of each customer subscribed to the broker for that target time-slot $t$. To predict the usage of each customer, it uses a NN with two hidden layers of size 7 each, and 10 epochs of training over the training data. The input data consists of the weather report, time of day (0-23), and day of week (1-7), while the target variable is the actual usage of the customer. During prediction, the weather forecast is used in place of the weather report to predict the usage for the next 24 hours. A fresh model is initialized every game for each customer, and then trained on the 336 data points obtained from the bootstrap data. The model is then continuously updated via online training throughout the game, as the broker gets more data points from the usage reports for each subscribed customer.

### Limit Price Predictor (LPP)

$VV$'s Limit Price Predictor is primarily motivated by the work of (Urieli and Stone 2014) on MDP-based wholesale bidding strategy, which in turn is based on (Tesauro and Bredin 2002). Although we use a similar MDP structure, the novelty lies in the reward, solution and application to place bids. The limit-prices generated from the MDP solution are used to bid for small energy quantities specified by BAQP across multiple auctions, as opposed to bidding the entire predicted energy requirement in a single auction as proposed by Urieli and Stone.

$VV$ maintains two instances of the MDP at all times - one for bids, another for asks. Going forward, we describe the components with respect to the bid MDP. At any given time-slot $t$, the predictor computes 24 limit prices for 24 wholesale auctions for the time-slots $t+1, \ldots, t+24$. The MDP components are hence defined as:

1. **States**: $s \in S = \{0, 1, \ldots, 24, success\}$, $s_0 := 24$
2. **Actions**: $limit\text{-}price \in \mathbb{R}$
3. **Transition**: Same state transition as Urieli and Stone; can be found in supplement. The transition function $p_{cleared}(s, limit\text{-}price)$ is determined by Equation (5).
4. **Reward**: At any state $s \in \{1, \ldots, 24\}$, the reward is 0. At terminal state $s = 0$, the reward is the negative of *balancing price* per unit energy. At terminal state $s = success$, the reward is the negative of the *limit-price* of the cleared bid. Since, for bids, we take the reward to be negative, maximizing reward results in minimizing costs.
5. **Terminal States**: $\{0, success\}$

The solution to the MDP is a sequential bidding strategy that minimizes the cost per unit energy procured. It is given by a value function which equals the *balancing-price* at state $s = 0$, and is recursively defined at all states by

$$V(s) = \begin{cases} balancing\text{-}price, & \text{if } s = 0 \\ \min\limits_{limit\text{-}price} \{p_{cleared} \times limit\text{-}price & \\ +(1 - p_{cleared}) \times V(s-1)\}, & \text{if } s \in [1, 24] \end{cases} \quad (4)$$

The value function in Equation (4) is computed recursively using dynamic programming. The solution gives an optimal *limit-price* for each state $s \in S$. By definition, $VV$ is always in the states $\{1, \ldots, 24\}$ of 24 concurrent auctions. Thus, $VV$ solves the MDP once every time-slot, and places 24 optimal *limit-prices* as bids to 24 auctions. The *balancing-price* and the transition function $p_{cleared}$ are both initially unknown. The former is estimated by averaging the *balancing-prices* across past time-slots separately for positive and negative imbalance (for ask and bid MDPs respectively). The transition function $p_{cleared}(s, limit\text{-}price)$ is estimated from the past auction statistics as:

$$p_{cleared} = \frac{\sum_{ac \in auction[s], ac.\text{LCP} < limit\text{-}price} ac.\text{cleared-amount}}{\sum_{ac \in auction[s]} ac.\text{cleared-amount}} \quad (5)$$

where $auction[s]$ is the set of all past auctions in the state $s$, and LCP is the *Last Clearing Price*, which is estimated in the DOQAP submodule described below. Since $VV$ iterates over the same sequence of states S, auction statistics for each state $s$ gets re-used in the future for estimating $p_{cleared}$.

**Bid/Ask Quantity Predictor (BAQP)**
For a target time-slot $t + 24$, $VV$ aims to procure its predicted energy amount from the wholesale market across all 24 possible auctions from $\{t, \ldots, t+23\}$. BAQP is responsible for distributing the predicted energy requirement across 24 auctions, with the aim of buying more and selling less in those auctions in which the prices are expected to be low, and vice-versa.

Following the state notation from the MDP from LPP, for each auction state $s \in \{1, \ldots, 24\}$ in which the broker participates at time $t$, it takes its own corresponding predicted net demand $\hat{\theta}^{VV}_{t+s,T}$ from CUP and *market position* $\Phi_{t+s}$ (the amount it has already procured for $t+s$ in the wholesale market) to find the energy left to procure $E_{t+s} = \hat{\theta}^{VV}_{t+s,T} - \Phi_{t+s}$. Then the 24 *limit-prices* from LPP are used to distribute the required energy among the bids to be placed in the remaining auctions. The energy quantity to bid for each state $s$ at time $t$ is given as:

$$e(s) = \begin{cases} \frac{E_{t+s}}{\sum_{j=s}^{24} \frac{limit\text{-}price[j]}{limit\text{-}price[s]}}, & \text{if } E_{t+s} > 0 \\ \frac{E_{t+s}}{\sum_{j=s}^{24} \frac{limit\text{-}price[s]}{limit\text{-}price[j]}}, & \text{if } E_{t+s} < 0 \\ 0, & \text{if } E_{t+s} = 0 \end{cases} \quad (6)$$

where $s \in \{1, \ldots, 24\}$, $limit\text{-}price[s]$ is the limit-price for state $s$ from *Limit Price Predictor*. The first case in Equation (6) occurs when $VV$ has to sell energy, and thus the energy to be sold in an auction is directly proportional to the predicted limit-price of that auction, i.e. sell more at high price. On the other hand, the second case occurs when $VV$ has to procure energy, and thus the energy to be bought in a target auction is inversely proportional to its predicted limit-price i.e. buy more at less price. The final bids of all states are of the form $(e(s), limit\text{-}price[s])$.

**Dummy Order Quantity and Price (DOQAP) Module**
In each cleared auction, the *Last Clearing Price* (LCP) for bids (asks) is higher (lower) than or equal to the *clearing price* of the auction. However, the LCP is unknown to every broker agent. Thus, having an good LCP estimation results in a better $p_{cleared}$ estimation. To estimate the LCP in an auction in state $s$, $VV$ places a fixed number of bids and asks with the least tradable energy amount in the market (0.01 MwH), and with prices equally spaced in the range $[\beta \times limit\text{-}price[s], balancing\text{-}price]$, where $\beta$ is a fixed parameter. Such bids and asks are called *dummy orders*. After clearance, the estimated LCP for bids in an auction in state $s$ is given by:

$$LCP(s) = \min(dummy\text{-}bids_{cleared}, limit\text{-}price[s]_{cleared}) \quad (7)$$

where $dummy\text{-}bids_{cleared}$ is the set of bid prices of all dummy bids which got cleared in the state $s$, and $limit\text{-}price[s]_{cleared}$ is the limit-price for the cleared final bid made by the broker in state $s$. The latter is set to infinity if the final bid doesn't clear. To estimate LCP for asks, we replace $\min$ by $\max$, and $dummy\text{-}bids_{cleared}$ by $dummy\text{-}asks_{cleared}$ in Equation (7). LCP is then used to update the transition function $p_{cleared}$ in Equation (5).

# Results

We analyze the performance of our broker $VV$ in Power TAC 2018 and show the efficacy of certain sub-modules of $VV$ using controlled off-line experiments.

## Power TAC 2018 Finals Results

The Power TAC 2018 Finals had 7 brokers from research groups across the globe. The tournament had a total of 324 games, with all possible combinations of 7-broker games (100 games), 4-broker games (140 games; 80 games for each broker), and 2-broker games (84 games; 24 games for each broker). Table 1 shows the net profit of all brokers across different game configurations, percentage of profit in comparison to the winning agent, AgentUDE, and the corresponding normalized scores. Despite winning more games than AgentUDE, $VV$ was placed next to AgentUDE in overall ranking of Power TAC 2018. This is because, the determination of the winner is made based on normalized cumulative profits in each configuration across all games in the tournament. Specifically, AgentUDE netted high profits against competing agents (excluding $VV$) in 2-player games that helped in cementing its place as the winner of the tournament.

Table 2 shows the number of $1^{st}$ and $2^{nd}$ place finishes by each broker across all three configurations. As seen, $VV$ won the most number of games in the tournament with 112 wins out of the 204 it participated in, with AgentUDE coming second with 92 wins out of 204. $VV$ had the most wins in 7-broker and 4-broker games, and had the second highest number of wins, behind AgentUDE, in 2-broker games. It is important to note that, overall, $VV$ finished in the top two, 72% of the time whenever it played in a game with more than 2 brokers. In comparison, AgentUDE stood at 65%. On a head-to-head comparison with AgentUDE, out of 100 7-broker games, AgentUDE and $VV$ both shared 39 wins each. However in 4-Broker games in which both $VV$ and AgentUDE participated, $VV$ won 31 times out 40, with AgentUDE winning the remaining 9. In the four 2-broker games involving both brokers, AgentUDE ended up winning three games. $VV$ led in all these three lost games almost till the end, only to fall behind finally due to transmission capacity fees. Figure 2 shows the number of games in which each broker ended up with a negative profit. CrocodileAgent had the fewest games with negative profits, with $VV$ coming second in this category with four times the average market share. Thus, $VV$ managed to make up for its losses on a consistent basis, and rarely ended up being non-profitable.

TM played a crucial role in $VV$'s success, offering tariffs which were attractive to majority of the customers and contributed the most in revenue. Figure 3 shows the average market share [3] to each broker across all three configurations and overall. $VV$ had the highest market share on average in 2-broker games, 7-broker games and overall, and the second highest in 4-broker games. In contrast, AgentUDE had only a quarter of the overall average market share of $VV$. While

[3]Note that the percentage will not sum up to 100 in some configurations. E.g.: In 4-broker games, each broker plays 80 games, where as in total 140 games are played

| Broker | 7-broker | 4-broker | 2-broker | Total | 7-broker (N) | 4-broker (N) | 2-broker (N) | Total (N) |
|---|---|---|---|---|---|---|---|---|
| AgentUDE | 49964603 (100) | 62138484 (100) | 134908672 (100) | 247011760 (100) | 1.091 | 0.634 | 1.565 | 3.291 |
| VidyutVanika | 48197051 (96) | 101942819 (164) | 47541635 (35) | 197681504 (80) | 1.056 | 1.061 | 0.336 | 2.453 |
| CrocodileAgent | 27659543 (55) | 45441732 (73) | 62881837 (47) | 135983111 (55) | 0.648 | 0.455 | 0.552 | 1.655 |
| SPOT | -6979768 (-14) | 32981756 (53) | 49183707 (36) | 75185695 (30) | -0.041 | 0.322 | 0.359 | 0.64 |
| COLDPower18 | 2063729 (4) | 10289982 (17) | 521330 (0.3) | 12875040 (5) | 0.139 | 0.078 | -0.326 | -0.109 |
| Bunnie | -67983216 (-136) | -25049555 (-40) | -19596577 (-15) | -112629348 (-46) | -1.254 | -0.3 | -0.609 | -2.163 |
| EWIIS3 | -87271195 (-175) | -206960249 (-333) | -109800161 (-81) | -404031605 (-164) | -1.638 | -2.25 | -1.878 | -5.766 |

Table 1: Power TAC 2018 – Net profits and normalized scores (denoted by (N)) of each broker

| Brokers | 7-Broker | | 4-Broker | | 2-Broker | | Total | |
|---|---|---|---|---|---|---|---|---|
| | $1^{st}$ | $2^{nd}$ | $1^{st}$ | $2^{nd}$ | $1^{st}$ | $2^{nd}$ | $1^{st}$ | $2^{nd}$ |
| VidyutVanika | 39 | 21 | 54 | 14 | 19 | 5 | 55 | 20 |
| AgentUDE | 39 | 26 | 31 | 21 | 22 | 2 | 45 | 24 |
| CrocodileAgent | 8 | 34 | 13 | 41 | 15 | 9 | 18 | 41 |
| SPOT | 0 | 0 | 16 | 19 | 9 | 15 | 12 | 17 |
| COLDPower18 | 0 | 3 | 5 | 29 | 8 | 16 | 6 | 24 |
| Bunnie | 13 | 15 | 21 | 16 | 9 | 15 | 21 | 22 |
| EWIIS3 | 1 | 1 | 0 | 0 | 2 | 22 | 1 | 11 |

Table 2: Power Tac 2018 – Number of $1^{st}$ and $2^{nd}$ place standings of each broker

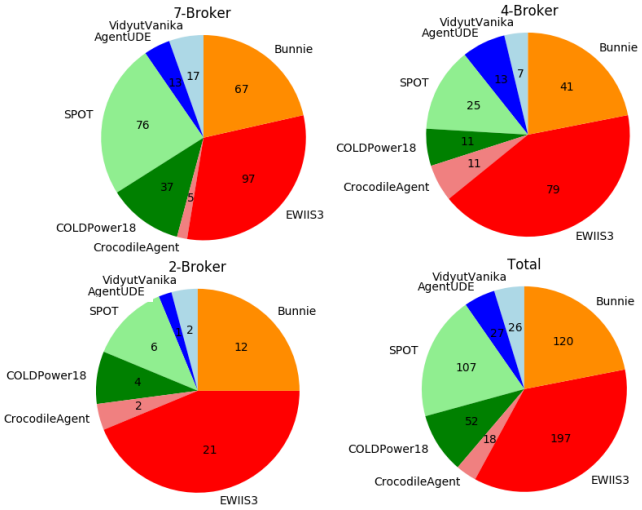


Figure 2: Power TAC 2018 – Number of games with negative profits

one may expect a greater market share to lead to more profits, it usually leads to higher transmission capacity fees and distribution costs, which can cause higher losses unless managed properly. As a result, agents with lower market share often tend to make less losses, and end up winning. Figure 4 represents the average income and costs of all brokers across all three configurations. $VV$ clearly has less imbalance costs while having almost similar number of customers as Bunnie, exhibiting the effectiveness of CUP. $VV$ also had one of the best tariff market income-to-cost ratio (1.14), with only AgentUDE (1.43) and CrocodileAgent (1.32) having better ratios. However, both AgentUDE and CrocodileAgent had very low average market share compared to $VV$. Thus, $VV$ is very efficient at making profits despite having a higher market share.

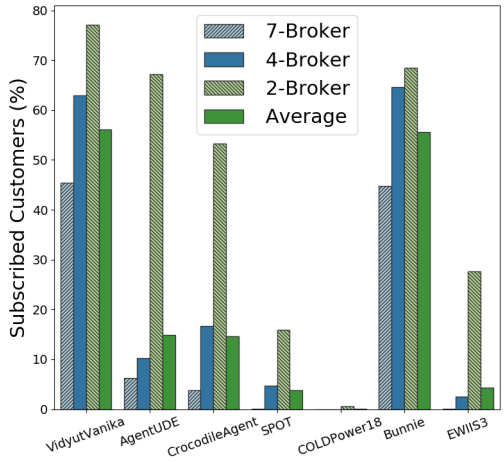


Figure 3: Power TAC 2018 – Average Percentage of customers subscribed (out of 57000), i.e. market share, of each broker

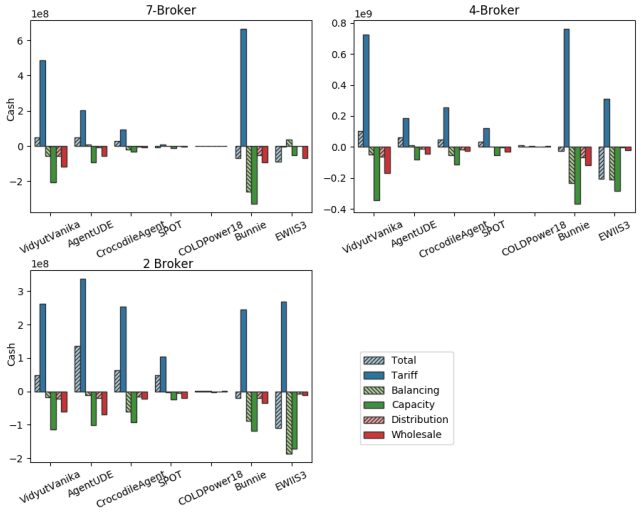


Figure 4: Power TAC 2018 – Average Income/Costs of each broker

## Controlled Offline experiments

For all controlled offline experiments, we played games using randomly chosen weather files from the 324 games

played in the Power TAC 2018 Finals.

To determine the prediction accuracy of NNs used in CUP, we played a set of 30 games with $VV$ being the sole participant. As all the customers end up subscribed to $VV$ in such a game, we determined the accuracy of each customer's usage prediction by comparing it to their actual usage. We got an average prediction accuracy of 84% from these set of games.

Next, in order to identify the contribution of each submodule in $VV$, we performed controlled offline experiments with test agents created by disabling multiple combinations of submodules. Agent F_DOQAP is generated from $VV$ by disabling the DOQAP submodule, and replacing the LCP in the LPP MDP solution with the clearing price, as implemented by (Urieli and Stone 2014). Agent F_BAQP is generated from $VV$ by disabling BAQP, which results in the agent placing the entire predicted net demand in a single bid in the wholesale market. Agent F_DOQAP_BAQP is generated from $VV$ by disabling both the DOQAP and BAQP submodules as above. Agent F_CUP is generated from $VV$ by replacing CUP by the usage predictor provided in the Power TAC *sample broker* which essentially predicts customer usage by exponentially smoothing over the past usage records, incrementally. Agent F_WM is generated by disabling the entire Wholesale Module, and replacing it by the wholesale strategy provided in the Power TAC *sample broker*. F_WM essentially increases the limit prices as the target time-slot gets closer with some randomization in the limit price determination. Agent F_Reward is generated from $VV$ by replacing the MDPQLM reward function by the reward function used by Cuevas, Rodriguez-Gonzalez, and De Cote. Agent F_TaD is generated from $VV$ by disabling TaD and instead offering FPTs from MDPQLM. In theory, this agent has the same tariff strategy as proposed by (Cuevas, Rodriguez-Gonzalez, and De Cote 2017), but with our reward function. Agent F_TaD_CUP is generated by disabling both TaD and CUP as described above. Agent F_TM is generated from $VV$ by disabling TM, but keeping WeeklyTOU active. Finally, F_TM_WM is generated by disabling both TM and WM in the manner described above.

Each of these test agents were made to compete with the full agent $VV$ over 30 games. The results of these experiments are reported in Table 3. Both TM and WM offer significant improvements as compared to the base sample broker strategy, with the former playing the biggest part in $VV$'s success. CUP, DOQAP and BAQP submodules play a crucial role in $VV$'s wholesale market strategy, and cause a significant decrease in profit when removed, as seen from the table. On the other hand, TaD submodule (responsible for generating MDPTOU) is crucial to $VV$'s tariff market strategy, removal of which causes a sharp decline in the broker's profit. Also note that, there is a significant decrease in profit when we used the reward function from (Cuevas, Rodriguez-Gonzalez, and De Cote 2017)[4] in F_Reward.

[4]We used a suitable value for the hyper-parameter in their reward function

| Brokers | % of $VV$'s profit |
|---|---|
| F_Reward | 75 |
| F_TaD | 83 |
| F_CUP | 84 |
| F_TaD_CUP | 75 |
| F_TM | 73 |
| F_DOQAP | 76 |
| F_BAQP | 79 |
| F_DOQAP_BAQP | 71 |
| F_WM | 90 |
| F_TM_WM | 72 |

Table 3: Performance of Test Agents vs Full agent $VV$

## Related Work

Since 2012, several research groups have benchmarked, deployed and published strategies using Power TAC. Özdemir and Unland (2015; 2018a; 2018b), Power TAC 2014 & 2017 Winners, use Genetic Algorithm and aggressive pricing to design tariffs for the tariff market, while using adaptive Q-learning in the wholesale market. They also predict the demand of customers using a combination of SARIMA and AR models. Power TAC 2015 Winners, Urban and Conen, design their TOU Tariff rates using a Hill Climbing algorithm. Past Power TAC participants Rúbio et al. (2015) present a fuzzy-logic based trading mechanism, while Liefers, Hoogland, and La Poutré (2014) use a heuristic inspired from Tit-For-Tat strategy in Iterated Prisoner's Dilemma, to determine tariff rates based on competing tariffs. Chowdhury et al. (2017, 2018) use an MDP & Q-Learning based tariff market strategy and a Monte-Carlo Tree Search based wholesale strategy, with the former incorporating the market share and cash position of the agent into the state space while taking actions on maintaining, incrementing or decrementing tariff rates.

Inspired from Reddy and Veloso (2011), Cuevas, Rodriguez-Gonzalez, and De Cote (2017) present an MDP-based strategy to generate FPTs, which forms the base of our TM after the reward structure modification. Based on a similar MDP, a Recurrent Deep Multiagent Reinforcement Learning framework with sequential clustering is presented by Yang et al. (2018). Urieli and Stone (2014; 2016a; 2016b), Power TAC 2013 winners, employ an MDP-based wholesale market strategy, coupled with a Linear Weighted Regression (LWR) based tariff market strategy which chooses the best possible candidate tariff after estimating its long-term utility. They also present the design and optimization of TOU Tariffs from their LWR based FPTs, but it is significantly different from our approach. We improve upon their wholesale strategy by using LCP estimation in DOQAP, and further boost its performance using BAQP. To predict limit prices for the wholesale market auctions, Chowdhury (2016) uses Decision Trees, Linear Regression and NN with weather data. However, none of the past publications use NN with weather data for future customer usage prediction.

## Conclusion

We described the critical elements of the strategy used by our broker VidyutVanika (VV), the runner-up in Power TAC 2018 Finals. In particular, we described details our two modules, TM and WM. TM and WM were responsible for VidyutVanika's actions in the tariff and wholesale market, respectively. The novelty of VidyutVanika lay in (i) defining reward functions for the MDPs, (ii) solving the MDPs, (iii) applying the MDP solutions to actions in the markets, and (iv) NN based usage predictor incorporating available weather data for better customer usage prediction. We illustrated the efficacy of our strategies by providing the detailed analysis of: (i) the comparative market-wise performance of VidyutVanika in the 2018 Power TAC finals and (ii) the offline experiments to demonstrate the contribution of each submodule of VidyutVanika.

## References


Bellman, R. 2013. *Dynamic programming*. Courier Corporation.

Chowdhury, M. M. P.; Folk, R. Y.; Fioretto, F.; Kiekintveld, C.; and Yeoh, W. 2017. Investigation of learning strategies for the SPOT broker in Power TAC. In Ceppi, S.; David, E.; Hajaj, C.; Robu, V.; and Vetsikas, I. A., eds., *Agent-Mediated Electronic Commerce. Designing Trading Strategies and Mechanisms for Electronic Markets*, 96–111. Cham: Springer International Publishing.

Chowdhury, M. M. P.; Kiekintveld, C.; Son, T. C.; and Yeoh, W. 2018. Bidding strategy for periodic double auctions using Monte Carlo tree search. In *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems*, 1897–1899. International Foundation for Autonomous Agents and Multiagent Systems.

Chowdhury, M. M. P. 2016. Predicting prices in the Power TAC wholesale energy market. In *AAAI*, 4204–4205.

Cuevas, J. S.; Rodriguez-Gonzalez, A. Y.; and De Cote, E. M. 2017. Fixed-price tariff generation using reinforcement learning. In *Modern Approaches to Agent-based Complex Automated Negotiation*. Springer. 121–136.

Ketter, W.; Collins, J.; and Weerdt, M. 2017. The 2018 power trading agent competition.

Liefers, B.; Hoogland, J.; and La Poutré, H. 2014. A successful broker agent for Power TAC. In *Agent-Mediated Electronic Commerce. Designing Trading Strategies and Mechanisms for Electronic Markets*. Springer. 99–113.

Özdemir, S., and Unland, R. 2015. Autonomous power trading approaches of a winner broker. In *AAMAS Workshop on Agent-Mediated Electronic Commerce and Trading Agents Design and Analysis (AMEC/TADA 2015)*, 143–156. Springer.

Özdemir, S., and Unland, R. 2018a. AgentUDE17: A genetic algorithm to optimize the parameters of an electricity tariff in a smart grid environment. In *Advances in Practical Applications of Agents, Multi-Agent Systems, and Complexity: The PAAMS Collection*, 224–236. Springer.

Özdemir, S., and Unland, R. 2018b. AgentUDE17: Imbalance management of a retailer agent to exploit balancing market incentives in a smart grid ecosystem.

Puterman, M. L. 1994. Markov decision processes. *Wiley and Sons*.

Reddy, P. P., and Veloso, M. M. 2011. Strategy learning for autonomous agents in smart grid markets. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Two*, IJCAI'11, 1446–1451. AAAI Press.

Rúbio, T. R.; Queiroz, J.; Cardoso, H. L.; Rocha, A. P.; and Oliveira, E. 2015. TugaTAC broker: A fuzzy logic adaptive reasoning agent for energy trading. In *Multi-Agent Systems and Agreement Technologies*. Springer. 188–202.

Speer, B.; Miller, M.; Schaffer, W.; Gueran, L.; Reuter, A.; Jang, B.; and Widegren, K. 2015. Role of smart grids in integrating renewable energy. Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States).

Tesauro, G., and Bredin, J. L. 2002. Strategic sequential bidding in auctions using dynamic programming. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 2*, AAMAS '02, 591–598. New York, NY, USA: ACM.

Urban, T., and Conen, W. 2017. Maxon16: A successful Power TAC broker. In *International Workshop on Agent-Mediated Electronic Commerce and Trading Agents Design and Analysis (AMEC/TADA 2017)*.

Urieli, D., and Stone, P. 2014. TacTex'13: A champion adaptive power trading agent. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 465–471. Association for the Advancement of Artificial Intelligence.

Urieli, D., and Stone, P. 2016a. Autonomous electricity trading using time-of-use tariffs in a competitive market. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)*. Association for the Advancement of Artificial Intelligence.

Urieli, D., and Stone, P. 2016b. An MDP-based winning approach to autonomous power trading: formalization and empirical analysis. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, 827–835. International Foundation for Autonomous Agents and Multiagent Systems.

Watkins, C. J., and Dayan, P. 1992. Q-learning. *Machine learning* 8(3-4):279–292.

Yang, Y.; Hao, J.; Sun, M.; Wang, Z.; Fan, C.; and Strbac, G. 2018. Recurrent deep multiagent Q-learning for autonomous brokers in smart grid. In *IJCAI*, 569–575.