TNA - A Deep-Q Learning broker for the PowerTAC tariff market

João Macedo and Raul Viana
{up201704464, up201208089}@edu.fe.up.pt

Faculdade de Engenharia da Universidade do Porto, Porto , PT

Abstract. PowerTAC is a competitive simulation that models a liberalized electrical energy market [1]. Future sustainable energy systems will have to comply not only with a smart grid but also to employ a smart market approach [2]. Brokers are entities interested in reducing costs and gaining profits operating in the smart grid. And so, they face several problems, such as balancing demand and supply from customers and coexisting with other competing brokers [3]. In this paper we will present the TemporaryNameAgent (TNA) broker, specialized in the tariff market. The TNA broker aims to apply Deep Q-learning techniques in the tariff specs, gaining market share and profit. TNA uses a neural network to predict the best action to perform at what tariff type at each moment to achieve the best market share and profit. We trained it against the TUCTAC broker [4], the winner agent for the 2020 edition of the PowerTAC competition. The results show that the TNA broker consistently improved its outcomes with the training games. Its performance was substantially better after the 700 training games, even managing to achieve a marginal profit competing with previous competition champion TUCTAC broker, and decisively out-preforming the *sampleBroker*.

Keywords: PowerTAC competition · autonomous agent · machine learning · deep q-learning.

1 Introduction

Climate changes, pollution, and CO2 emissions had become the main drives for innovation in the energy sector. Access to clean, affordable and reliable energy has been a cornerstone of the world's increasing prosperity and economic growth [5]. These demands enforce improvements in the energy market as a whole, including the physical electricity grid, the related services and the energy markets. PowerTAC simulates these new upcoming changes and provides a framework for testing new smart grid market approaches, as a virtual competition [1]. In this context, competitors tune their brokers to simulate market energy companies and compete with each other for profit and market share [6].

The PowerTAC simulation environment is composed of three markets: the tariff market, the wholesale market and a market for balancing resources [1]. In the tariff market, the brokers try to acquire energy generation capacity from local producers and load capacity from energy consumers. Our broker TNA, will operate in this market only, optimizing the tariffs made available to the customers.

In this market, there are different customer types, defined by its PowerType [1]. Each one has its own characteristics, needs and preferences, so the broker has to define different tariffs for different customer types.

The tariff design has to provide the conditions required by customers and also be profitable to brokers. Customers select the best tariff based on their self-interests, determined by its customer model [1]. And brokers try to maximize profit through improving market share and maximizing tariff profit.

This balance, between meeting the customer's expectations and maximizing broker gains, is the key factor for succeeding in PowerTAC simulation, although its complexity and difficulty, alongside with the competitive factor amongst brokers is difficult to accomplish.

Nevertheless, the best tariff does not always guarantee the client's loyalty. The customer models have an inertia built-in utility function to model the real-world customer's habits. It expresses the human aversion to change and complexity, that may retard or limit the decision to change to better tariffs [7]. This inertia factor grows, from zero, through the simulation time, making the customer's decision of changing tariff harder and harder, as simulation time passes. All customers evaluate tariff offerings in the first publication cycle, but their interest tails off with time. Additionally, the choice of the better tariff has some probabilistic degree, which makes customers not always choose the better tariff, modelling human errors and apathetic attitudes [1].

A tariff can formally be defined as the tuple:

$$\mathbf{C = \{pavg, `p, bp, ewp, cl\}}$$

where:

- **pavg** = simple average rate of all tariff rates (€/kWh);
- **'p** = daily payment from customers (€);
- **bp** = sign-up bonus paid by the broker to the subscribed customer (€);
- **ewp** = early withdrawal penalty, charged to the customer if they leave the tariff earlier than the contract length;
- **cl** = contract length, the minimum period which a customer must stay in the tariff.

Each tariff may be as complex or simple as the broker desires though each tariff can only target a single power type such as consumption or production [11]. From this, the key question we aim to answer through iterative tariff manipulation is evident.

Is it possible to find the perfect tariff for each customer group, at a determined time, by manipulating these components and guarantee the biggest of the competitors market share?

2 Related Work

This complex problem of finding the best tariff at all times has been studied for some time. Due to its complexity or to its specificities, it had been handled in a myriad of different approaches. None of these approaches is demonstrably the best, and none stands out, as can be seen by the competition podium heterogeneity in the last six years [1].

Although all these different approaches have two common final objectives, maximize the profit and have the highest market share, the means to achieve them are composed of a wide diversity of methodologies.

Some approaches, like the one by Reddy and Veloso, try to focus on the tariff and its properties. In their study, Reddy and Veloso tried to predict the attraction of a tariff to a client based on its characteristics [8]. At the same time, they tried to negotiate private tariff characteristics with other brokers to make a better prediction.

Some approaches rely upon simplifying the environment reality, trying to achieve a higher level of efficiency. Liefers et. al developed a broker with a simple structure, publishing only one tariff at a time. This tariff is fixed-price, single-rate, with no sign-up bonuses or fees, no early-exit bonuses and no periodic payments. They chose the best opponent's tariff and simplified it to their tariff representation. They argue that this final tariff is expected to be the most relevant one, hence attracting the most customers [9].

Others, on the other hand, rely on not simplifying the model, trying to make a conceptual analysis, developing a tariff generator that understands the linguistic genesis of the constraints. This is achieved through the use of fuzzy logic, where variables are described using conceptual values, for example, the variable temperature could be specified as "cold", "normal" or "hot" [7].

Finally, there are other different approaches. For example, CrocodileAgent evaluates the market properties, mainly its scarcity, balanced and oversupply items. Then generates the best-fitted tariff to those conditions. It tries to gather an environment's wider perspective, to make more supported decisions [10]. Silva tried to tackle the PowerTAC complexity by using a Multi-Agent System, breaking his agent into sub-agents working independently, but communicating its solutions to a central agent. Each sub-agent is responsible to deal with specific parts of the tariff calculation. There is a sub-agent to deal with the prediction of consumption and production, other to financial analysis, and other possible features [6]. This way, avoiding simplifications and narrowing focus, this approach tries to implement a broader strategy, hoping for superior results.

3 The TNA strategy

As seen before, there is no proven best methodology to handle the tariff market complexity and the other brokers' competitiveness. So, we have tried to implement the TNA with a mix and an improvement of well-succeeded strategies.

3.1 Tariff Creation

The TNA was developed with a *sample broker* as a starting basis. This *sample broker* is intended to provide developers with a foundation that interfaces correctly with the PowerTAC infrastructure and an example of a working broker agent [12]. From there on, developers can design their broker with their preferred strategies.

As such, and for simplicity purposes, TNA's tariff creation strategy is similar to the Sample broker's method. A tariff is created for each Power Type with fixed rates for all time-slots with the average energy market price as the cost. Although this simplifies the tariff creation process, the implementation of TNA allows for the update of more complex tariffs, as it alters all rates connected to that tariff. It also allows for updating more than one tariff for each Power Type. We believe that future work in this area could allow for relevant improvement in achieving a competitive tariff offer earlier.

3.2 Observed States

The state observed by the DQN's policy network every six time-slots, right before the customer's reevaluation period is a normalized vector of the variation of several simulation variables in relation to the previous reevaluation period. Normalization was done through a Gaussian distribution based conversion function used by the implemented ObservationGenerator, based on statistics of the means and standard deviations of the population. The mean of all the observed variables is 0, as they represent a variation of the state.

These variables and standard deviation (std) are as follows:

- Δ **balance** = variation of balance *(std = 5000)*;
- Δ **subscribers** = variation of subscribers *(std = 1000)*;
- Δ **storageConsumption** = variation of consumption/production of Storage Power-Type Customers *(std = 10000)*;
- Δ **storageSubscriptions** = variation of subscribers of Storage Power-Type Customers *(std = 100)*;
- Δ **productionConsumption** = variation of consumption/production of Production Power-Type Customers *(std = 10000)*;
- Δ **productionSubscriptions** = variation of subscribers of Production Power-Type Customers *(std = 100)*;
- Δ **consumptionConsumption** = variation of consumption/production of Consumption Power-Type Customers *(std = 10000)*;
- Δ **consumptionSubscriptions** = variation of subscribers of Consumption Power-Type Customers *(std = 100)*;

Additionally, as the objective function in the following section is based on the current time-slot, we added a ninth variable to the observed state. This variable, in opposition to the previous ones is not a variation but rather the current simulation time-slot. We also normalize it through the same process, with the mean being half of the expected total simulation time-slot count (as the true value is variable to avoid end-game strategies), and the standard deviation being 34.1% of the expected number of time-slots.

- Δ **timeSlot** = Current simulation Time-Slot;

3.3 Actions

We opted for a simplifying strategy over a strategy that tries to handle all the simulation variables [9]. We simplified the tariff tuple, defaulting some of its variables, seeding just one tariff per power type, and applied a q-learning model that decides the actions to apply to tariffs in each superseding cycle, as did Chowdhury [11].

We opted for a reduced set of actions, in an attempt to reduce the training time needed for the convergence of the policy network to a consistent and competitive strategy. These actions, when taken, translate into superseding a subset of the published tariffs. While the expansion of the set of actions would be easily implemented through the current implementation design, the set of action used to obtain the following results is as follows:

- **Stay** = No tariffs are superseded;
- **StorageUP** = Storage Power-Type Tariffs (rates UP by 10%);
- **StorageDOWN** = Storage Power-Type Tariffs (rates DOWN by 10%);
- **ProductionUP** = Production Power-Type Tariffs (rates UP by 10%);
- **ProductionDOWN** = Production Power-Type Tariffs (rates DOWN by 10%);
- **ConsumptionUP** = Consumption Power-Type Tariffs (rates UP by 10%);
- **ConsumptionDOWN** = Consumption Power-Type Tariffs (rates DOWN by 10%);

3.4 Objective function

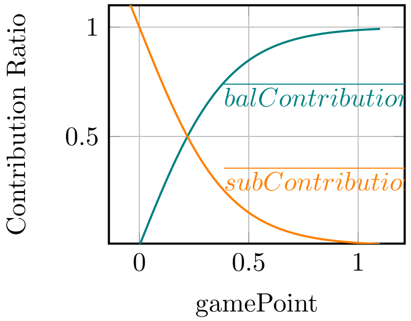
The broker approach to the retail tariff market can be aggressive or cooperative. In a cooperative approach, all brokers agree to negotiate the tariffs prices high, allowing to maximize the profit for all. In an aggressive approach, there's a broker, or more, who keeps seeding best lower tariffs than the others, relegating the profit and focusing on getting the largest market share.

We opted for this aggressive market strategy, tailoring our objective function to prioritize securing market share as Rubio et. al [7], in the early stages of the simulation, and transitioning to prioritising balance increase later.

The objective function is then dependent on the current time-slot, the ΔBalance and the ΔSubscribers. With the current time-slot we first map this to a range between 0 and 1, with the expected time-slot count, creating a gamePoint variable. We then make use of the hyperbolic tangent function *tanh* to reach the contributions of ΔBalance and ΔSubscribers to the reward function. We define an additional constant to moderate the transition from early subscriber prioritization and later balance focus.

$$tanhFactor = 2.5$$

$$balContribution = tanh(gamePoint \times tanhFactor)$$
$$subContribution = 1 - balContribution$$

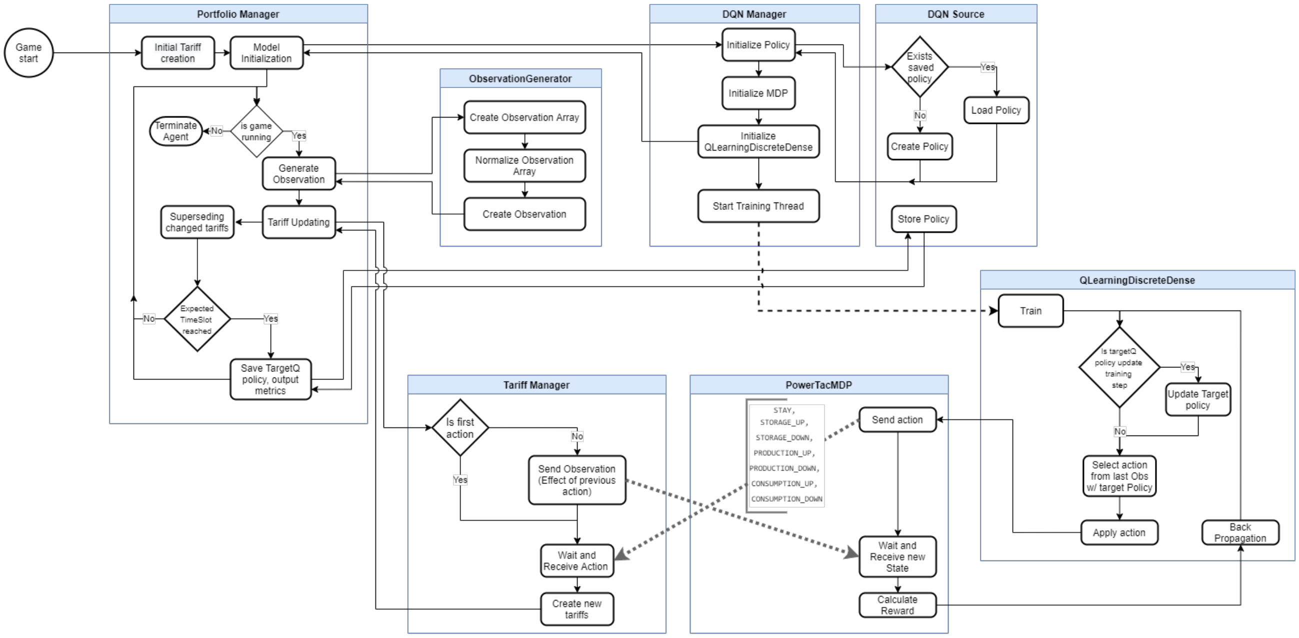


$$Reward(R) = subContribution \times \varDelta Subscribers + balContribution \times \varDelta Balance$$

4 Implementation

The implementation of the TNA was based on the *Sample Broker* made available by the team behind PowerTAC.

Additionally, we used the libraries DeepLearning4j for the policy network and ND4j for the Epsilon-greedy Double Deep-Q learning implementation, including its Markov Decision Process definition, extended for the PowerTAC Scenario.



The Policy Network is generated from the specifications available in the *mlmodel/DQNSource.java* file. The layers of the policy network, all fully connected, used to obtain the following results are as follows:

- **Input** = *Number of nodes: 9*, *Activation Function: RELU*;
- **Hidden 1** = *Number of nodes: 8*, *Activation Function: RELU*;
- **Hidden 2** = *Number of nodes: 8*, *Activation Function: RELU*;
- **Output** = *Number of nodes: 7*, *Activation Function: SOFTMAX*;

Bellow we present the resulting topology of the policy network.

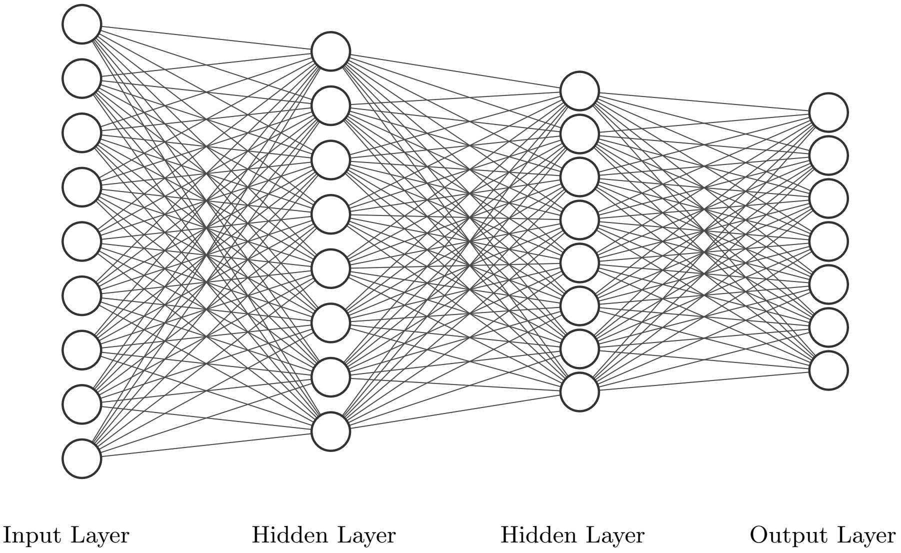


Fig. 1. Tested Policy network

5 Training Process

The training process involves firstly the creation of the target policy file. Following the creation of this network, the weights of which where initially randomized and updated at every training game, we proceeded to the sequential simulation of games. For this, we started by generating a set of bootstrap files, for initializing the simulations, For $N = numberOfTrainingGames$ we generate $N \times 0.1$ bootstrap files, randomizing the chosen one for each simulation.

The training games, and the following validation games for the result's section were shortened to around 130 time-slots, for convenience during the training process. This is of particular relevance for interpreting the following training process and outcomes, as it entails that, even if the approach is validated, a training process for complete simulations would have to take place for TNA to be ready for a classic PowerTAC tournament environment.

We trained our broker for 700 games against the TUCTAC Broker. As can be seen in Figure 2, although we kept the *epsilon* (ϵ) value for the implemented Epsilon-Greedy Q-learning strategy quite high, the model cumulative reward had suffered a light improvement through the 700 games set. This ϵ value was kept high enough so the model could explore a greater percentage of the solution space allowing the model to sometimes chose a non-optimal decision. This way the trend line has a slight slope of 0.0001, which demonstrates the model slow evolution throughout the entire solution space, even with a high degree of action randomization.

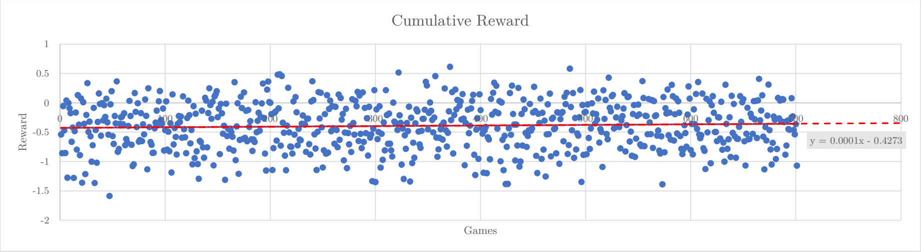


Fig. 2. Cumulative Reward trend line over training

Overall balance and subscribers also had some improvement during the 700 games set. The trend line in Figures 3 and 4 shows an improvement. The trend line of the total number of subscribers at the end of each game has a 0.27 slope, denoting a slight improvement. On the other hand, the trend line for the overall balance at the end of each game has a slope of 14.9, which denotes an impressive evolution.

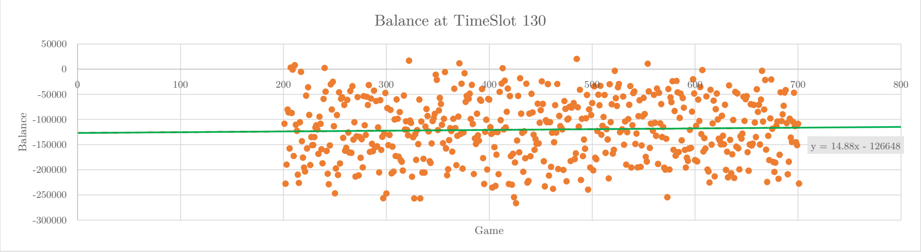


Fig. 3. Overall Subscribers trend over training

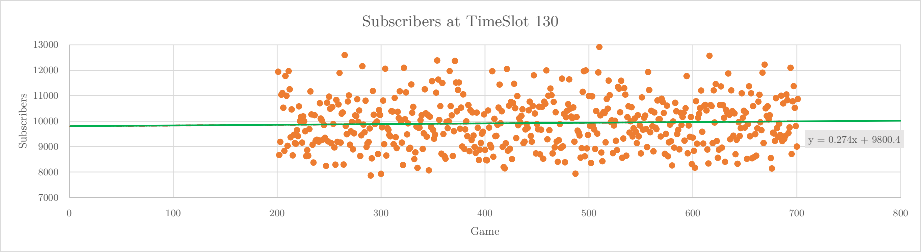


Fig. 4. Overall Subscribers trend over training

More relevant for the evaluation of the training process is the relation between these results, with a high exploratory component, and the following 20 games, executed with $\epsilon = 0$, translating to a complete exploitation of the perceived optimal actions. As we can see from Figures 5, 6 and 7, reflecting the variables explored in figures 2, 3 and 4 respectively, the games executed following the training were consistently more successful than the previously calculated trend, showing a clear convergence to a mode effective strategy.

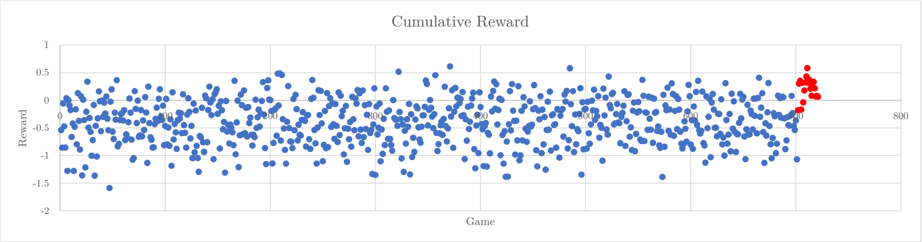


Fig. 5. Cumulative Reward (exploration/exploitation)

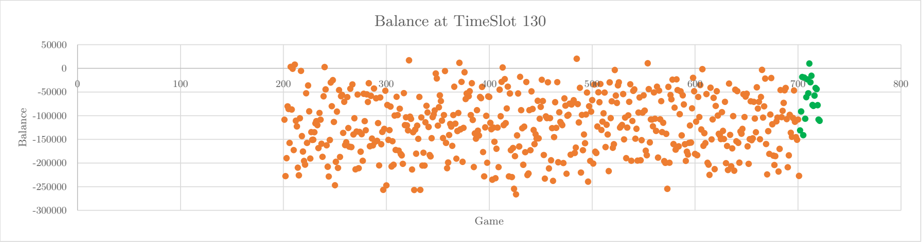


Fig. 6. Overall Balance (exploration/exploitation)

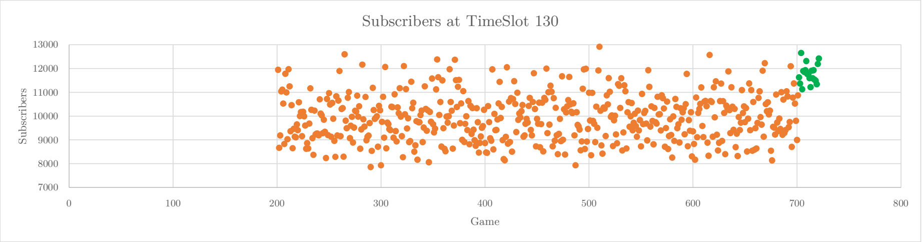


Fig. 7. Overall Subscribers (exploration/exploitation)

The evolution from the initial random games to these final 20 games with $\epsilon = 0$ and a trained policy network constitutes the results of our model for the purposes of this paper, validating the general approach as an effective strategy for tariff publication management. This evolution will be further explored in the following section.

6 Results

The results were produced playing games against TUCTAC and *sample broker*. The following figures were obtained with the PowerTAC visualizer, which displays graphically the game evolution and its variables. Figure 8 and 9 shows TNA performance without any training.

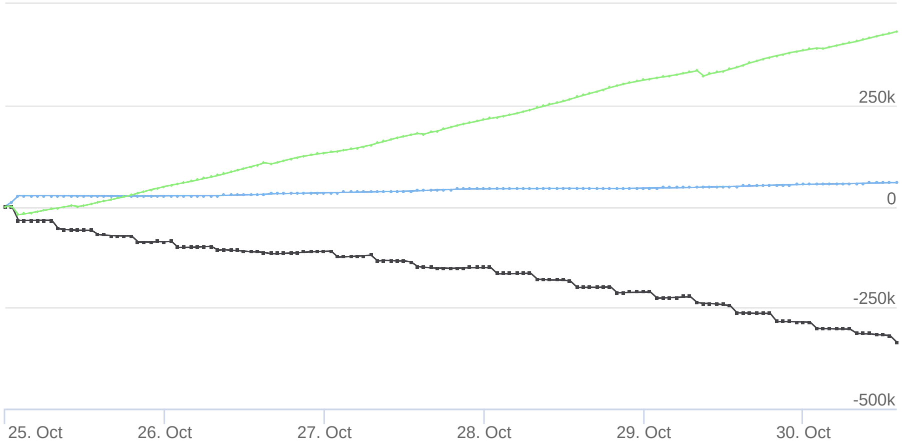


Fig. 8. Balance [nTrainingGames = 0] - TNA (black) VS TucTac (green)

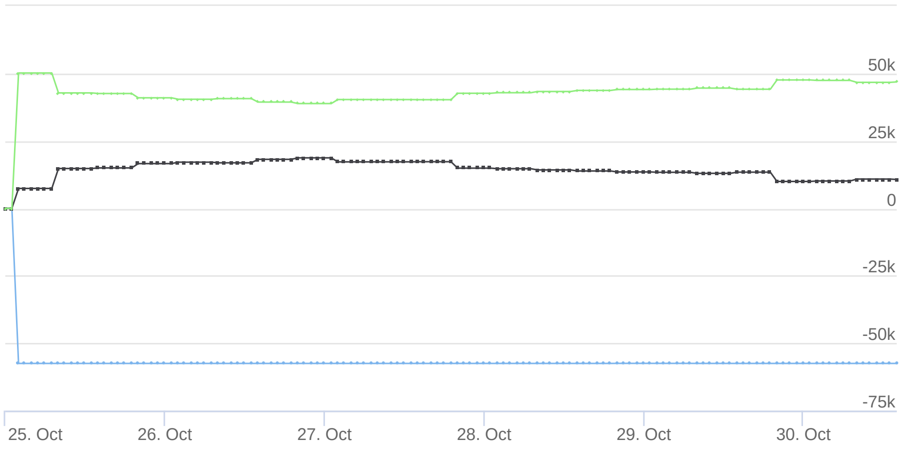


Fig. 9. Subscribers [nTrainingGames = 0] - TNA (black) VS TucTac (green)

From the two former figures, it's quite obvious that without training our broker shows a deplorable performance against TUCTAC. It has a poorly market share, but most importantly the balance has a continuous drop-down, reaching alarmingly low negative values. This poor balance performance it's due to its random decisions updating tariffs, but mostly because it's superseding new tariffs at each publishing time slot. This is a very expensive behaviour, one that has diminished with the following training.

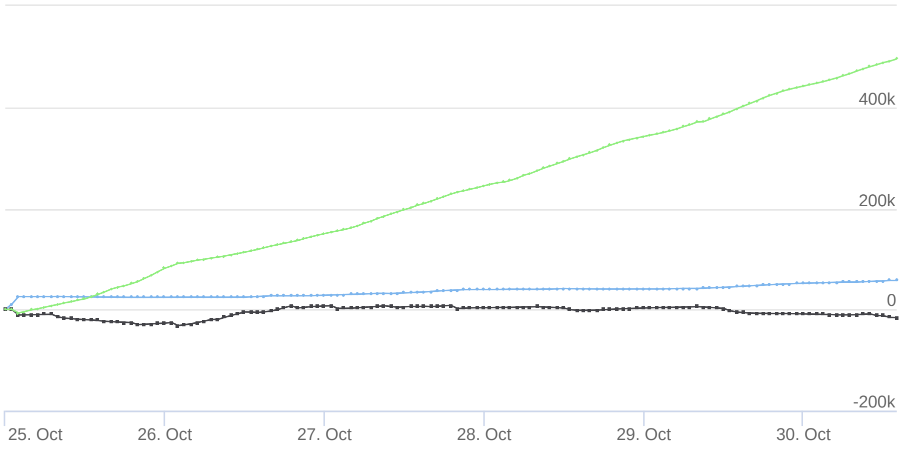


Fig. 10. Balance [nTrainingGames = 700] - TNA (black) VS TucTac (green)

Figure 10 shows the balance performance with training. This performance was a massive improvement, as our broker came from huge losses pre-training to a positive balance after training, competing against the TUCTAC broker, a capable opponent. It was a relevant progression, only form 700 training games. As for subscribers, our broker also had a really impressive performance after training. It was able to steal TUCTAC subscribers in the early stages of the game, reaching a 50% market share. Reaching this value, TUTAC becomes more aggressive, due to its minimum 50% market share strategy, and aggressively lowers tariffs to regain subscribers. This aggressive behaviour arise was concomitant with the time when our broker started to prioritize balance variation over the market share, as defined by the reward function. With this being the case, TNA was not able to regain the lost subscriptions, but nevertheless maintained a positive balance almost until the end of the game.

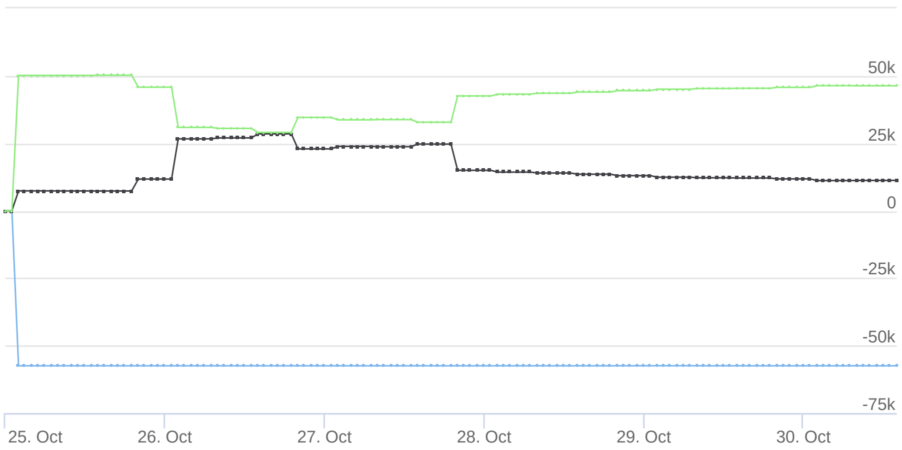


Fig. 11. Subscribers [nTrainingGames = 700] - TNA (black) VS TucTac (green)

Training the model with the TucTac proved effective in improving the broker's performance in games against that opponent. We were interested in knowing if this learned strategy was generalizable, and as such we ran two games against the Sample Broker's base implementation, once before and once after training. The balance progression on these games can be seen bellow.

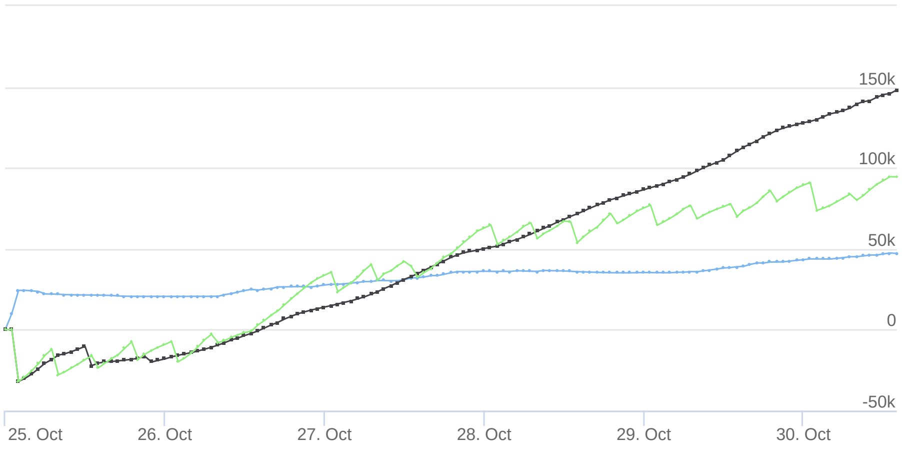


Fig. 12. Balance [nTrainingGames = 0] - TNA (green) VS SampleBroker (black)

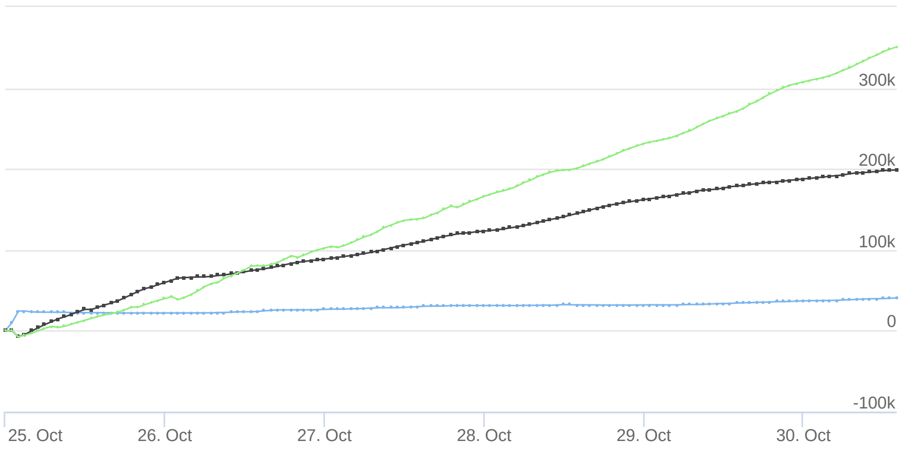


Fig. 13. Balance [nTrainingGames = 700] - TNA (green) VS SampleBroker (black)

While before training the recurrent random tariff superseding actions were very costly, causing TNA to lose decisively in balance total against the Sample Agent, after the training process TNA learned to maintain a tariff specification for longer during the later stages of the game through a set of ”Stay” actions. This proved to be effective in not only in closing the gap seen in the first game but in decisively leading in balance against the Sample Broker.

7 Future Work

As discussed in the training process sub-section, the cumulative reward trending line has a slight positive slope and its' calculated root is at x $\simeq$ 4000. That suggests that the model could continue to gain performance with more training, reinforcing its' accuracy and performance. It would be interesting to evaluate the results of more training games.

Relating to the brokers' performance, it would be interesting to test different model inputs, and how they impact the decision performance.

Also, it would be interesting to measure the performance effect of adding more inputs, augmenting the complexity. On the other hand, testing variations of the model output, like publishing more than one tariff type per time slot, more complex tariffs or changing the degree of tariff variation would also be very interesting, as it would make it possible to tune the model to better performances.

8 Conclusions

The PowerTAC tariff market can be a very complex and competitive environment. Nevertheless, the presented results demonstrate the viability of using Deep-Q Learning in the context of this problem, achieving a relevant market share and positive balance results, even when competing against the winner broker of the 2020 edition of the competition.

These results were achieved with a restricted set of actions and relatively low training time, and we trust that, based on our results, further investment in this approach would prove successful.

We therefore firmly conclude that PowerTAC brokers can achieve a profitable behaviour by using Deep-Q Learning, in the context of the tariff market.

References

1. Ketter, W., P. Reddy, and J. Collins.: "Power TAC: A competitive economic simulation of the smart grid.". Energy Economics (39), 262-270 (2013)
2. Grgic Demijan, Hrvoje Vdovic, Jurica Babic, and Vedran Podobnik: "CrocodileAgent 2018: Robust Agent-based Mechanisms for Power Trading in Competitive Environments.", Computer Science and Information Systems(16), 40-40 (2018)
3. Yang Yaodong, Mingyang Sun, Changjie Fan, and Goran Strbac: "Recurrent Deep Multiagent Q-Learning for Autonomous Brokers in Smart Grid." Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, pp. 569-575. Tianjin (2018)
4. Orfanoudakis, S., Kontos, S., Akasiadis, C., Chalkiadakis, G. Aiming for Half Gets You to the Top: Winning PowerTAC 2020. (2020)
5. Majumdar, Chu S.: "Opportunities and challenges for a sustainable energy future." Nature edition (08), 294-303 (2012)
6. Silva, Jão.: "Agent Strategies in Smart Energy Markets - PowerTAC.", `https://repositorio-aberto.up.pt/handle/10216/85755.2016`. Last accessed 09 Jun 20121

7. Rúbio, Thiago, Jonas Queiroz, Henrique L. Cardoso, Ana P. Rocha, and Eugénio Oliveira.: “TugaTAC Broker: A Fuzzy Logic Adaptive Reasoning Agent for Energy Trading. (2015)
8. Reddy P., Veloso M.: Strategy Learning for Autonomous Agents in Smart Grid Markets. Twenty-second International Joint Conference on Artificial Intelligence. 1446-1451 (2011)
9. Liefers B., Hoogland J., Poutré H.: A successful broker agent for Power TAC. Lecture Notes in Business Information Processing. 187. 99-113. 10.1007/978-3-319-13218-1-8 (2014).
10. Matetic S., Babic J., Matijas M, Petric A., Podobnik V.: The CrocodileAgent 2012: Negotiating Agreements in a Smart Grid Tariff Market (2012)
11. Chowdhury M.M.P., Folk R.Y., Fioretto F., Kiekintveld C., Yeoh W.: Investigation of Learning Strategies for the SPOT Broker in PowerTAC. Designing Trading Strategies and Mechanisms for Electronic Markets - Lecture Notes in Business (2017)
12. Ketter W., Collins J.:“The 2020 Power Trading Agent Competition.” 41 (2020)