Bidding Strategy for Periodic Double Auctions Using Monte Carlo Tree Search

Extended Abstract

Moinul Morshed Porag Chowdhury
The University of Texas at El Paso
mchowdhury4@miners.utep.edu

Christopher Kiekintveld
The University of Texas at El Paso
cdkiekintveld@utep.edu

Tran Cao Son
New Mexico State University
tson@cs.nmsu.edu

William Yeoh
Washington University in St. Louis
wyeoh@wustl.edu

ABSTRACT

Bidding strategies for *Periodic Double Auctions* (PDAs) are complicated because they need to predict and plan for future auctions, which may affect the bidding strategy in the current auction. We present a general bidding strategy for PDAs based on forecasting clearing prices and using *Monte Carlos Tree Search* (MCTS) to plan a bidding strategy across multiple time periods. We developed a controlled simulator by isolating Power Trading Agent Competition's wholesale market to evaluate bidding strategies in a realistic PDA energy market. We show that our MCTS bidding strategy is cost effective in buying energy compared to other baseline and state-of-the-art strategies and it's performance improves with increasing number of MCTS simulations.

KEYWORDS

Bidding Strategy; Monte Carlo Tree Search; Machine Learning

ACM Reference Format:

Moinul Morshed Porag Chowdhury, Christopher Kiekintveld, Tran Cao Son, and William Yeoh. 2018. Bidding Strategy for Periodic Double Auctions Using Monte Carlo Tree Search. In *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), Stockholm, Sweden, July 10–15, 2018,* IFAAMAS, 3 pages.

1 INTRODUCTION

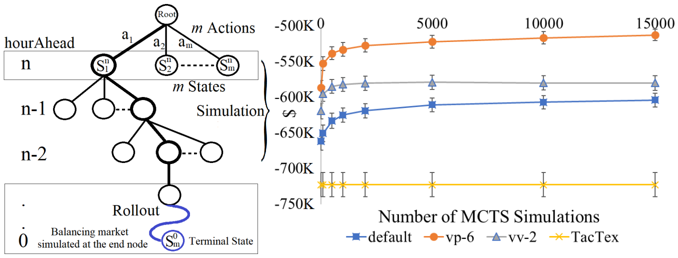
The evolving smart energy grid has the potential to improve many problems of our current traditional energy grid [12]. By incorporating more advanced sensing capabilities and artificial intelligence for better decision making, it will provide a more intelligent energy infrastructure. However, as the grid becomes more decentralized, automated, and capable of providing much greater volumes of sensor data, we need to develop the economic structures and decision-making algorithms to manage these grids efficiently. To this end, an important area of research for smart grids is to understand the market mechanisms that can coordinate buying and selling decisions in energy markets, and to develop automated bidding agents that can represent individuals in these markets. One of the major academic efforts to develop such strategies centers on the *Power Trading Agent Competition* (Power TAC) [11], a competition with over a decade of history [18], which supports a competitive benchmarking research model [13], building on the experience of competitions in 2012-2017 such as the Trading Agent Competition for Supply-Chain Management (TAC SCM) [5] and Trading Agent Competition for Ad Auctions (TAC AA) [10].

The *Periodic Double Auction* (PDA) wholesale energy market is one of the common real energy exchange protocols, e.g., NordPool, FERC, or EEX [6, 8, 11, 15]. The PDA is a general type of auction that can be used to trade many other types of goods beyond energy. Bidding in a single double auction is strategically complex and having periodic auctions adds the need to reason about future auctions for the same good, including predicting future clearing prices and planning a future bidding strategy. In this context, the search space of a wholesale broker is too large for a systematic search approach to be applicable. Inspired by the successes of *Monte Carlo Tree Search* (MCTS) [2], a statistical anytime algorithm for finding optimal decisions that combines the precision of tree search and the generality of random sampling, we develop a MCTS bidding strategy for PDAs. In this research, by developing a controlled testbed for a PDA-based realistic wholesale power market, we test PDA bidding strategies and propose a novel MCTS-based bidding strategy for autonomous energy broker agents. We present the design, implementation, and empirical evaluation of this strategy. Empirical studies show that MCTS outperforms benchmark strategies as well as a state-of-the-art bidding policy from a champion Power TAC agent.

2 MCTS BIDDING STRATEGY

Since bidding in a PDA is essentially a sequential planning problem for any particular sequence of auctions for trading energy in a specific time period, we proposed an approach based on the successful MCTS family of algorithms. To implement an MCTS strategy, we first need to forecast the market clearing prices for the current and future auctions for different time slots. We experiment with two different price prediction methods: *MDP Price Predictor* [17] and *RepTree Price Predictor* [3, 7]. We set REPTree as our default price predictor because we have found that REPTree has a better accuracy (avg. error 54.05%) in predicting 24 hour ahead auction prices than the MDP predictor (avg. error of 66.28%); previous empirical studies also show that REPTree performs better in predicting market clearing prices comparing to other machine learning strategies [3, 4].

Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. André, S. Koenig (eds.), July 10–15, 2018, Stockholm, Sweden.



(a) n^{th} hour ahead auction **(b) Net cost comparison.**

Figure 1: MCTS Tree and Costs of Best Candidate Agents

The main components of our MCTS bidding strategy are:

Actions: We represent the main actions of the MCTS strategy as prices relative to the predicted distribution of clearing prices for the current auction. Each action ($action_m$) is represented by $\{\mu, \sigma, \{\Delta_{min}, \Delta_{max}\}, \gamma\}$ where μ represents the limit price, σ is the observed standard deviation of the clearing price distribution, $\{\Delta_{min}, \Delta_{max}\}$ is the minimum and maximum price multiplier tuple, and γ is the volume (in %) of the current demand δ. We estimate μ using the price predictor and estimate σ by running 30 four-broker simulations. The Δ_{min} and Δ_{max} are multiplied with σ and used by the agent to create μ^{mcts}_{min} and μ^{mcts}_{max} from μ. Using μ^{mcts}_{min} and μ^{mcts}_{max}, the MCTS strategy is able to bid in different price ranges. For example, if $\{\Delta_{max}, \Delta_{min}\}$: {1, -1}, our MCTS strategy varies its bid prices in the first standard deviation range. An action is a *NO-BID* action when $\gamma = 0$.

States: Each state keeps a memory of its corresponding action id, visit-count, and avgUnitCost (i.e., total avg. unit cost incurred by the agent in this auction and all future auctions). The agent selects the state that has the highest UCT value while doing simulations. Each action leads to a specific state, so an agent with an *action-space*$_m$ size of m actions can go into m different states from a specific state.

Transition: A state S^n_m transitions into one of $S^{n-1}_0, \ldots, S^{n-1}_m$ states.

Terminal States: $\{S^0_0, \ldots, S^0_m\}$. If there are m actions and n hour ahead auctions, the search tree will have m^n terminal states.

Reward: While doing rollouts/simulations, if the agent reaches a terminal state, it gets a balancing cost (C_{bal}) as a reward, which corresponds to the price the agent would pay for energy on the fall-back balancing market. Otherwise, at each state, it gets a simulated cost (C_{sim}) that is the summation of the cost paid for energy in all of the auctions.

Simulation: While running a MCTS simulation for t timeslot's n^{th} hour ahead auction by selecting the m^{th} action, the agent first gets the current demand $\delta^n_{m,t}$ and tries to clear γ% of $\delta^n_{m,t}$ using a simulation of the market clearing process. It generates a simulated market clearing price $\chi^n_{m,t}$ from a Gaussian distribution, where the mean is equal to μ^n_t and standard deviation is σ. If the bid's limit price μ^{mcts} is greater than $\chi^n_{m,t}$, then the bid gets cleared. If $v^n_{m,t}$ volume is cleared in this process, the agent updates its $\delta^n_{m,t}$ for the rest of the hour ahead auction simulation by deducting $v^n_{m,t}$ from $\delta^n_{m,t}$ and repeats the same process until it reaches a terminal state or a state where $\delta^n_{m,t}$ is zero. At each level of the hour ahead auction, we get a C_{sim} as follows: $C^n_{sim,m,t} = \chi^n_{m,t} * v^n_{m,t}$.

Rollout: If the agent reaches a state where there are no children, then the agent selects an action randomly, creates a state and adds it to the tree. Then, it does a random rollout process (i.e., picking actions randomly from the action space and traversing from the newly added state to the terminal state). When it reaches a terminal state, it calculates the C_{bal} by multiplying $\delta^n_{m,t}$ with the simulated unit balancing cost. At timeslot t, the unit balancing cost is calculated by doubling the maximum unit ask price for that specific timeslot's hour ahead auctions. After repeating N_{sim} number of MCTS iterations for the n^{th} hour ahead auction, the agent builds the tree as illustrated in Figure 1a. It selects the action that leads to the highest UCT value [14] state S^n_m from the *root*. After bidding according to the best action, the agent discards the whole MCTS tree and builds it again from the scratch when it needs to bid for the $(n-1)^{\text{th}}$ hour ahead auction.

3 EXPERIMENTAL RESULTS

We conducted an empirical analysis of four bidding strategies: ZI [9], ZIP [16], TacTex [17], and MCTS. Our default MCTS has five bid actions and one *NO-BID* action. Default action-space properties are as follows: *Number-of-iterations* (N^d_{sim}): 10,000. *Bid-Volume* (γ_d): 100% of the current demand (except for NO-BID). *Price-Multipliers* $\{\Delta^d_{min}, \Delta^d_{max}\}$: Five actions have five price multiplier tuples $\{-1, 0\}$, $\{0, 1\}$, $\{-1, 1\}$, $\{0, 1\}$, and $\{0, 2\}$. *Number-of-Bids* (N^d_{bid}): 10 bids. 9 bids are minimum bandwidth bids with limitprice starting from the μ^{mcts}_{min} to μ^{mcts}_{max} and the 10^{th} bid is the main bid that is submitted at μ^{mcts}_{max} price. We run experiments varying $\Delta^d_{min,max}$ (vp) and varying γ_d (vv) property, select the two best candidate agents $\text{MCTS}_{\text{vv-2}}$ and $\text{MCTS}_{\text{vp-6}}$, and run 30 four-broker (ZI, ZIP, TacTex, and MCTS) games for both agents by varying N_{sim}. MCTS is a statistical anytime algorithm, so more computation time should lead it to a better performance [1]. Figure 1b demonstrates that $\text{MCTS}_{\text{vp-6}}$ does very well compared to other strategies with increasing N_{sim}, where it is able to bid successfully and procure the necessary volume at a lower price. $\text{MCTS}_{\text{vp-6}}$ has a reasonably wide range of pinpoint price selection options (where $\Delta^{vp}_{min} = \Delta^{vp}_{max}$) which can be considered as the best variation of MCTS. Following this policy, when this agent simulates the auctions with a larger number of simulations, it makes good decisions to bid at the right moment to procure the full demand.

4 CONCLUSIONS

We propose a novel approach for bidding in Periodic Double Auctions (PDAs) using Monte Carlo Tree Search (MCTS). Our strategy shows significant improvements over two widely known baselines and the state-of-the-art PDA bidding strategy. Empirical analyses show (unsurprisingly), MCTS performs better with a larger number of MCTS simulations. An important restriction on all of these MCTS strategies is that they search only a fixed space of possible bidding actions. Future work involves considering dynamic MCTS policies that add promising new actions to the search space over time.

ACKNOWLEDGMENTS

This research is partially supported by NSF grant 1345232.

REFERENCES

[1] Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. 2012. A Survey of Monte Carlo Tree Search Methods. *IEEE Transactions on Computational Intelligence and AI in Games* 4, 1 (2012), 1–43.

[2] Guillaume Chaslot, Sander Bakkes, Istvan Szita, and Pieter Spronck. 2008. Monte-Carlo Tree Search: A New Framework for Game AI. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE).*

[3] Moinul Morshed Porag Chowdhury. 2016. Predicting Prices in the Power TAC Wholesale Energy Market. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI).* 4204–4205.

[4] Moinul Morshed Porag Chowdhury, Russell Y. Folk, Ferdinando Fioretto, Christopher Kiekintveld, and William Yeoh. 2017. Investigation of Learning Strategies for the SPOT Broker in Power TAC. In *Agent-Mediated Electronic Commerce. Designing Trading Strategies and Mechanisms for Electronic Markets*, Sofia Ceppi, Esther David, Chen Hajaj, Valentin Robu, and Ioannis A. Vetsikas (Eds.). Springer International Publishing, 96–111.

[5] John Collins, Wolfgang Ketter, and Norman Sadeh. 2010. Pushing the Limits of Rational Agents: The Trading Agent Competition for Supply Chain Management. *AI Magazine* 31, 2 (2010), 63.

[6] Federal Energy Regulatory Commission. 2017. http://www.ferc.gov. (2017).

[7] Tapio Elomaa and Matti Kaariainen. 2001. An Analysis of Reduced Error Pruning. *Journal of Artificial Intelligence Research* (2001), 163–187.

[8] European Energy Exchange. 2017. http://www.eex.com/en. (2017).

[9] Dhananjay K Gode and Shyam Sunder. 1993. Allocative Efficiency of Markets with Zero-Intelligence Traders: Market as a Partial Substitute for Individual Rationality. *Journal of Political Economy* 101, 1 (1993), 119–137.

[10] P Jordan, Ben Cassell, L Callender, and M Wellman. 2009. *The Ad Auctions Game for the 2009 Trading Agent Competition.* Technical Report.

[11] Wolfgang Ketter, John Collins, and Mathijs de Weerdt. 2017. The 2017 Power Trading Agent Competition. *ERIM Report Series Reference No. ERS-2017-002-LIS* (2017).

[12] Wolfgang Ketter, Markus Peters, and John Collins. 2013. Autonomous Agents in Future Energy Markets: The 2012 Power Trading Agent Competition. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI).* 1298–1304.

[13] Wolfgang Ketter, Markus Peters, John Collins, and Alok Gupta. 2015. Competitive Benchmarking: An IS Research Approach to Address Wicked Problems with Big Data and Analytics. *ERIM Report Series Reference No. ERS-2015-015-LIS* (2015).

[14] Levente Kocsis and Csaba Szepesvári. 2006. Bandit Based Monte-Carlo Planning. In *Proceedings of the European Conference on Machine Learning (ECML).* 282–293.

[15] Nord Pool. 2017. http://www.nordpoolspot.com. (2017).

[16] Gerald Tesauro and Rajarshi Das. 2001. High-performance Bidding Agents for the Continuous Double Auction. In *Proceedings of the ACM Conference on Electronic Commerce (EC).* 206–209.

[17] Daniel Urieli and Peter Stone. 2014. TacTex'13: A Champion Adaptive Power Trading Agent. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS).* 1447–1448.

[18] Michael P Wellman, Amy Greenwald, Peter Stone, and Peter R Wurman. 2003. The 2001 Trading Agent Competition. *Electronic Markets* 13, 1 (2003), 4–12.