A Novel Bidding Strategy for PDAs using MCTS in Continuous Action Spaces

Sanjay Chandlekar[1,2], Easwar Subramanian[1]

1 TATA CONSULTANCY SERVICES

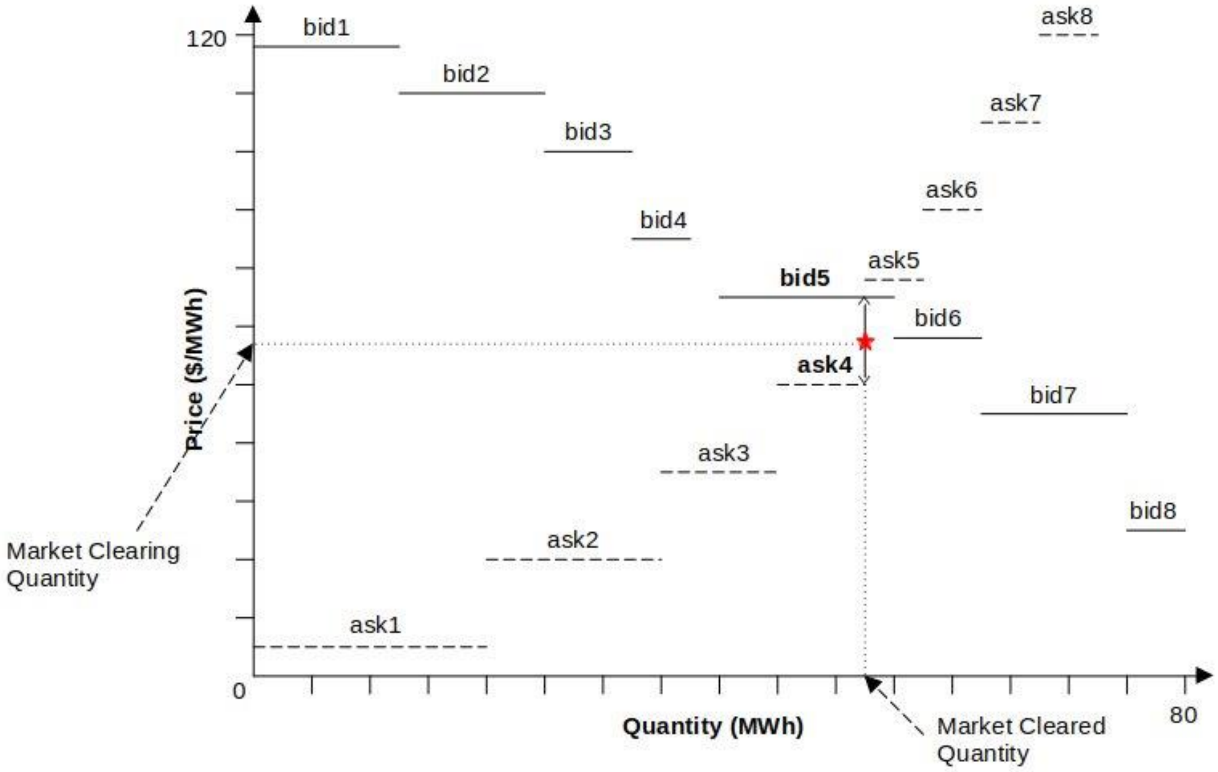
2 INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY HYDERABAD

EMAS 2024 AUCKLAND NEW ZEALAND

Background - PDA

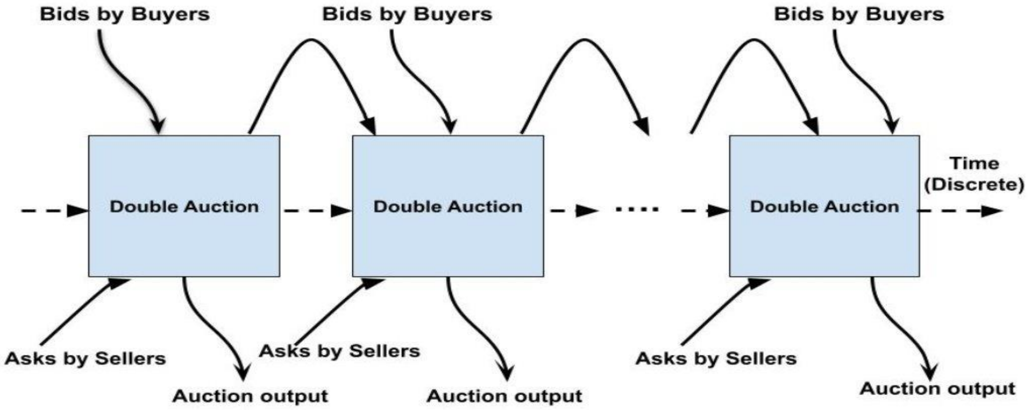
What is Periodic Double Auctions (PDAs)?

Single-shot auction with k-Double Auction



What is Periodic Double Auctions (PDAs)?

Periodic Double Auction
(Sequence of auctions)



Applications: Stock market auction and Energy auction

*Image Credits: A Nash Equilibrium Solution for Periodic Double Auctions, Bharat Manvi and Easwar Subramanian

Challenges in Periodic Double Auctions (PDAs)

- Strategic planning across current and future auctions
- Each decision influences subsequent steps
- Need to be real-time
- Need to account for the updated requirements based on factors like weather

Consequently, the decision-making process for bidding strategies becomes intricate, demanding innovative approaches to navigate the complexities of real-time bidding in PDAs.

For such sequential decision-making problems $\xrightarrow{\text{Perfect Fit}}$ **Monte Carlo Tree Search (MCTS)**

Background - MCTS

Introduction of MCTS

- Monte Carlo Tree Search (MCTS) framework is capable of constructing trees
  - That encompass entire decision-making trajectories
  - Comprehensively capturing process dynamics
- Ability to blend the precision of tree search with the generality of random sampling

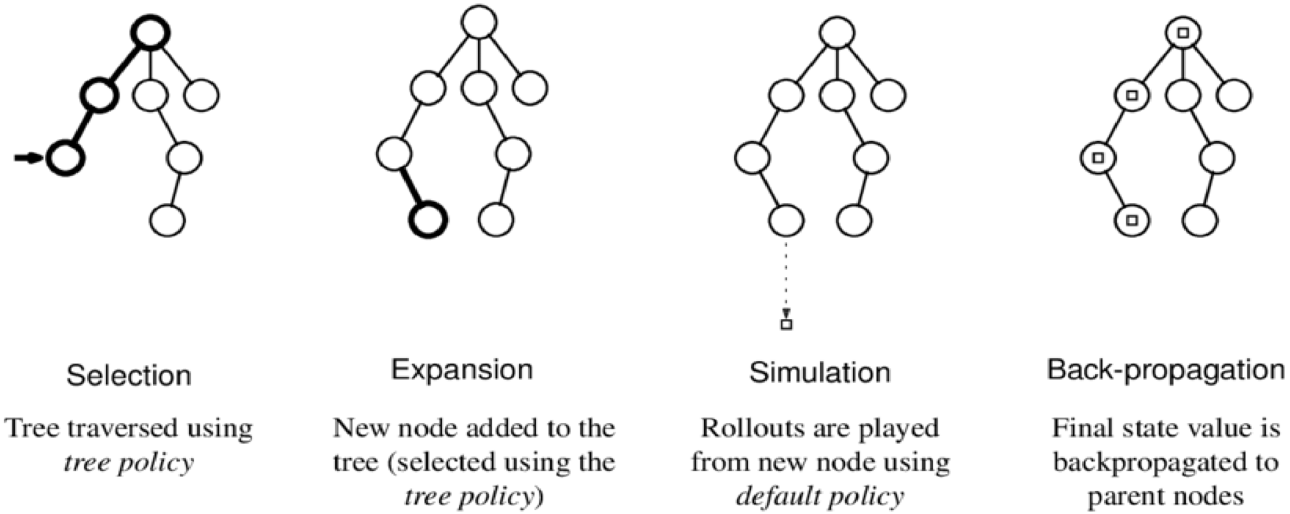


*Image Credits: Monte Carlo tree search experiments in hearthstone, Andre Santos, Pedro A. Santos and Francisco S. Melo

Types of MCTS

- Discrete MCTS

- Continuous MCTS
  - Discretization
  - Progressive Widening

Limitation: They do not leverage insights gained from explored actions to perform the exploration of unexplored actions or enhance knowledge about previously explored ones

- Some previous work (mostly for game playing, not directly applicable to other domains)
  - Policy gradient with MCTS

Our Contributions

- We investigate the efficacy of MCTS in continuous action spaces, specifically for the context of placing bids in a periodic double auction.

- Introducing Regression-MCTS (R-MCTS), a novel method designed to navigate the continuous action space of bid prices, harnessing insights gleaned from explored actions, to generalize the action space and accelerating MCTS learning.

- We demonstrate the effectiveness of R-MCTS by evaluating against several state-of-the-art MCTS methods, as well as various PDA bidding strategies, leveraging a simulated PDA environment as our experimental test bed.

Elements of Regression-MCTS (R-MCTS)

- **UCT Selection:** $argmax_a \left[\bar{v}_a + C\sqrt{\frac{log \sum_b n_b}{n_a}} \right]$

- **SPW:** $n(v_c)^\alpha \geq |children(v_c)|$

- **Clearing Probability:** $p_{cleared}(s, action) = \frac{\sum_{ac \in auction[s], ac.CP < action} ac.cleared\_amount}{\sum_{ac \in auction[s]} ac.cleared\_amount}$

Regression-MCTS (R-MCTS)

Selection and Expansion

Simulation

Backpropogation

Best Move

Algorithm 1 R-MCTS(*rem_quant*, *cur_ts*, *delv_ts*)

```
bids ← [], root ← Node()                                   # initialise bids list and root node
rem_auctions ← delv_ts − cur_ts                             # number of auctions remaining in a PDA
if rem_quant > 0 then
    while i in NUMBER_OF_ROLLOUTS do
        visited ← [], rewards ← [], cur ← root              # initialise lists of visited, rewards
        visited ← [visited; root]
        list_of_sellers ← generate_sellers()
        list_of_buyers ← clone_buyers()
        while not cur.is_leaf(rem_quant) do
            cur ← select(rem_quant)                         # see algorithm 2
            cur.p_cleared ← get_pcleared(rem_auctions, cur.action)
            cp, cq, rem_quant ← perform_auction(cur.action, rem_quant,
            list_of_sellers, list_of_buyers)                # clearing the current auction round
            rewards ← [rewards; cp]
            visited ← [visited; cur]
        end while
        cur_cost, cur_quant ← cur.simulation(rem_quant, rem_auctions)
        root ← backpropogation(rewards, visited, cur_cost, cur_quant)
    end while
    lp ← root.best_action()
    bids ← [bids; Bid(buyer_ID, lp, rem_quant)]             # limit order
else
    bids ← [bids; Bid(buyer_ID, NULL, rem_quant)]           # market order
end if
return bids
```

SPW Condition

Variable Risk Tactic

Algorithm 2 select($node$, rem_quant)

if $number\_of\_visits(node)^{\alpha} \leq number\_of\_children(node)$ **then**

    $A \leftarrow$ actions considered in $node$

    $action \leftarrow argmax_{a \in A} \mathbb{E}(v|a) + C\sqrt{\frac{log \sum_{b \in A} number\_of\_visits(b)}{number\_of\_visits(a)}}$ # UCB-select

else

    new $action \leftarrow$ child of $node$ by taking an action using $p\_cleared$ data

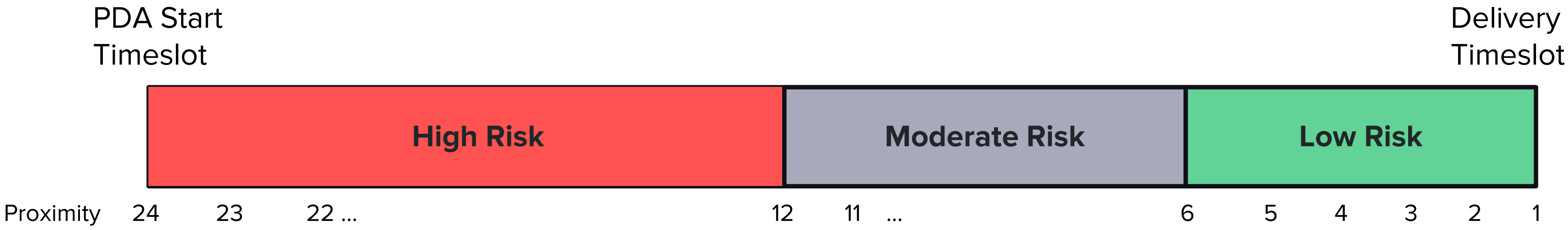
    $node.children \leftarrow [node.children; action]$ # SPW-select: expanding action space

end if

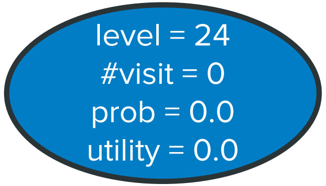
$new\_state = node.action$

return new_state

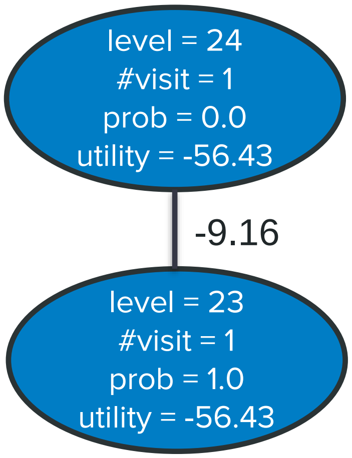
Variable Risk Tactic



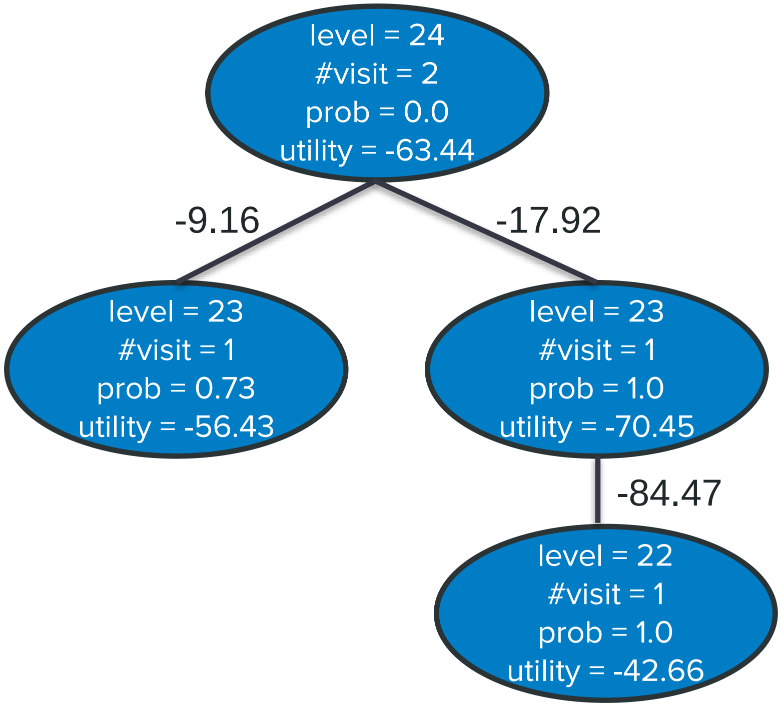
At the start



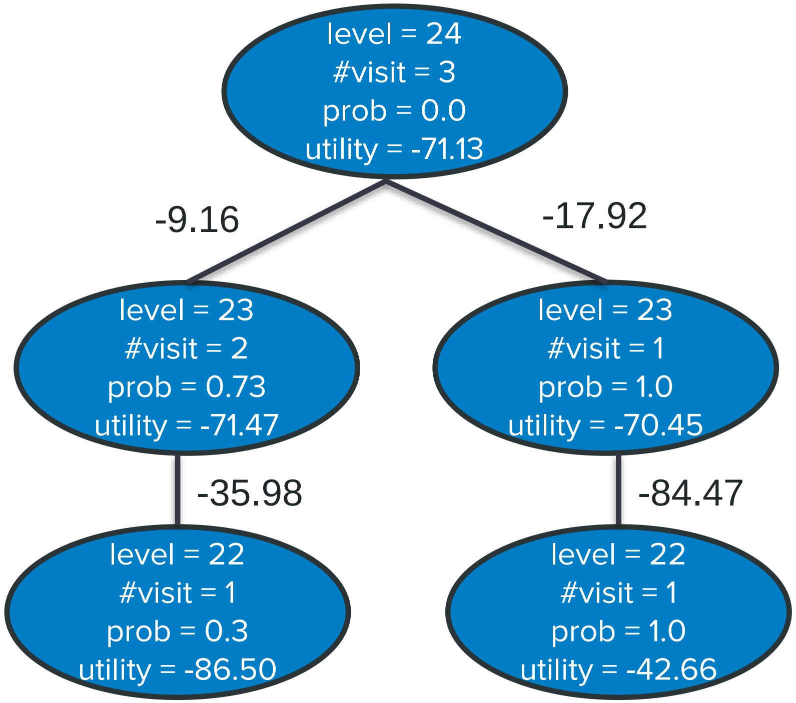
1st Iteration



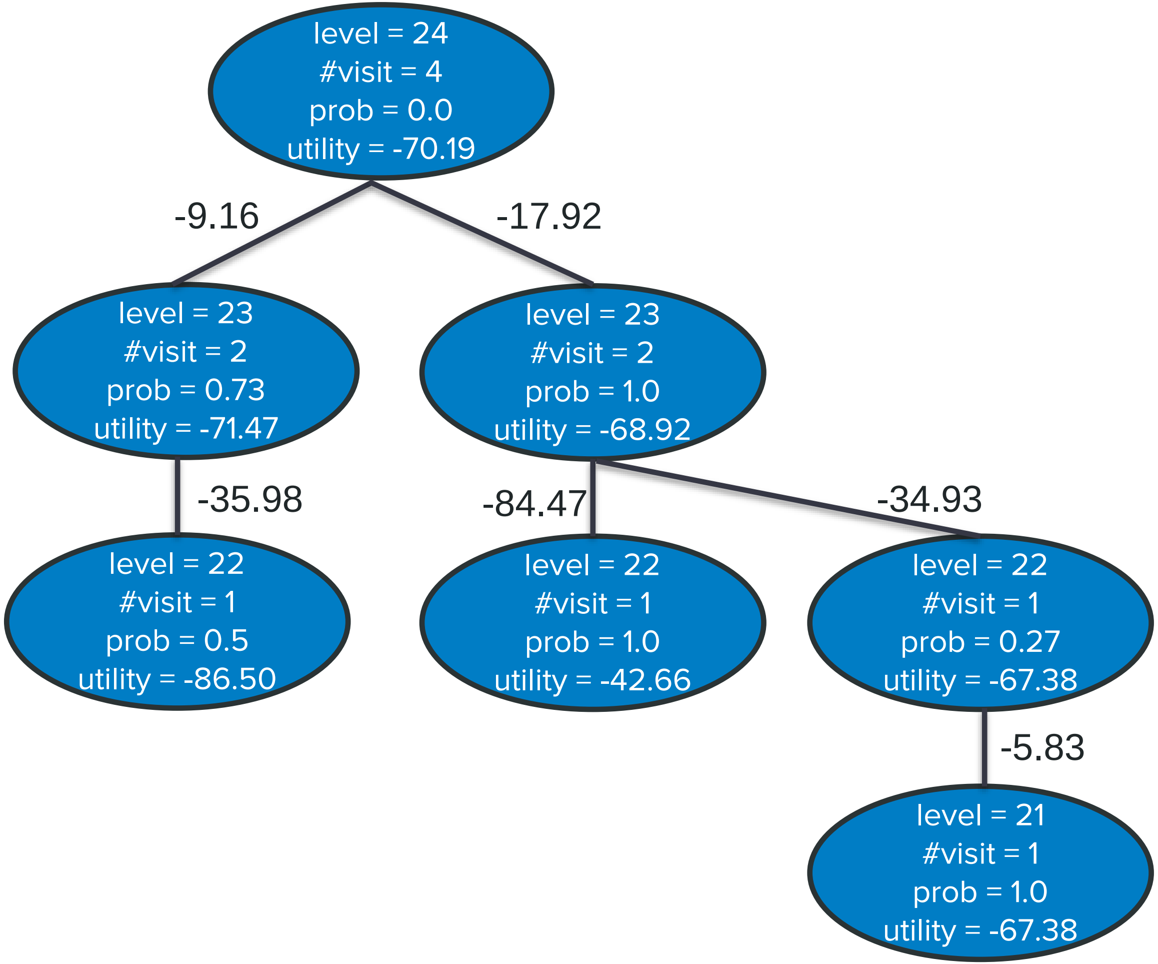
2nd Iteration



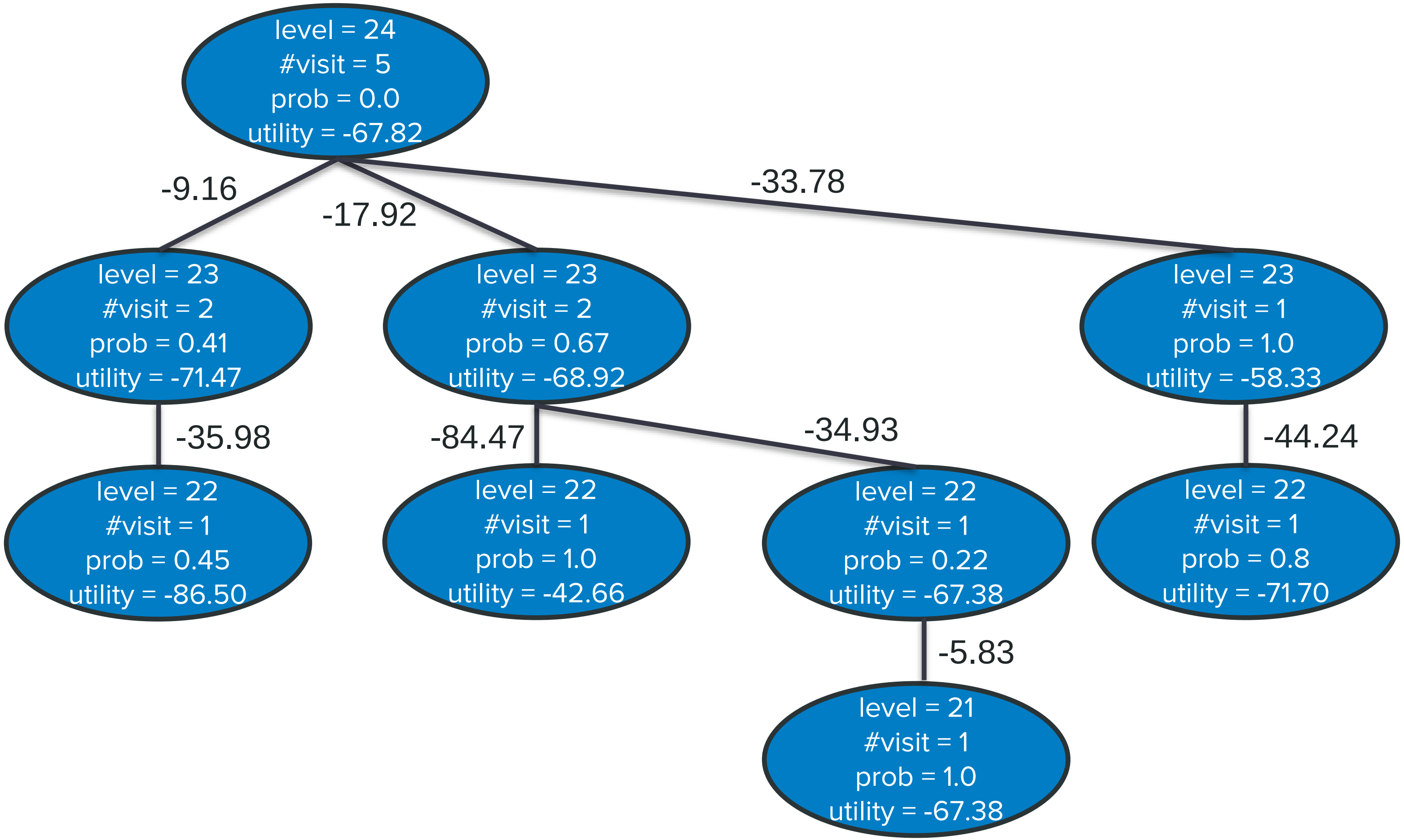
3rd Iteration



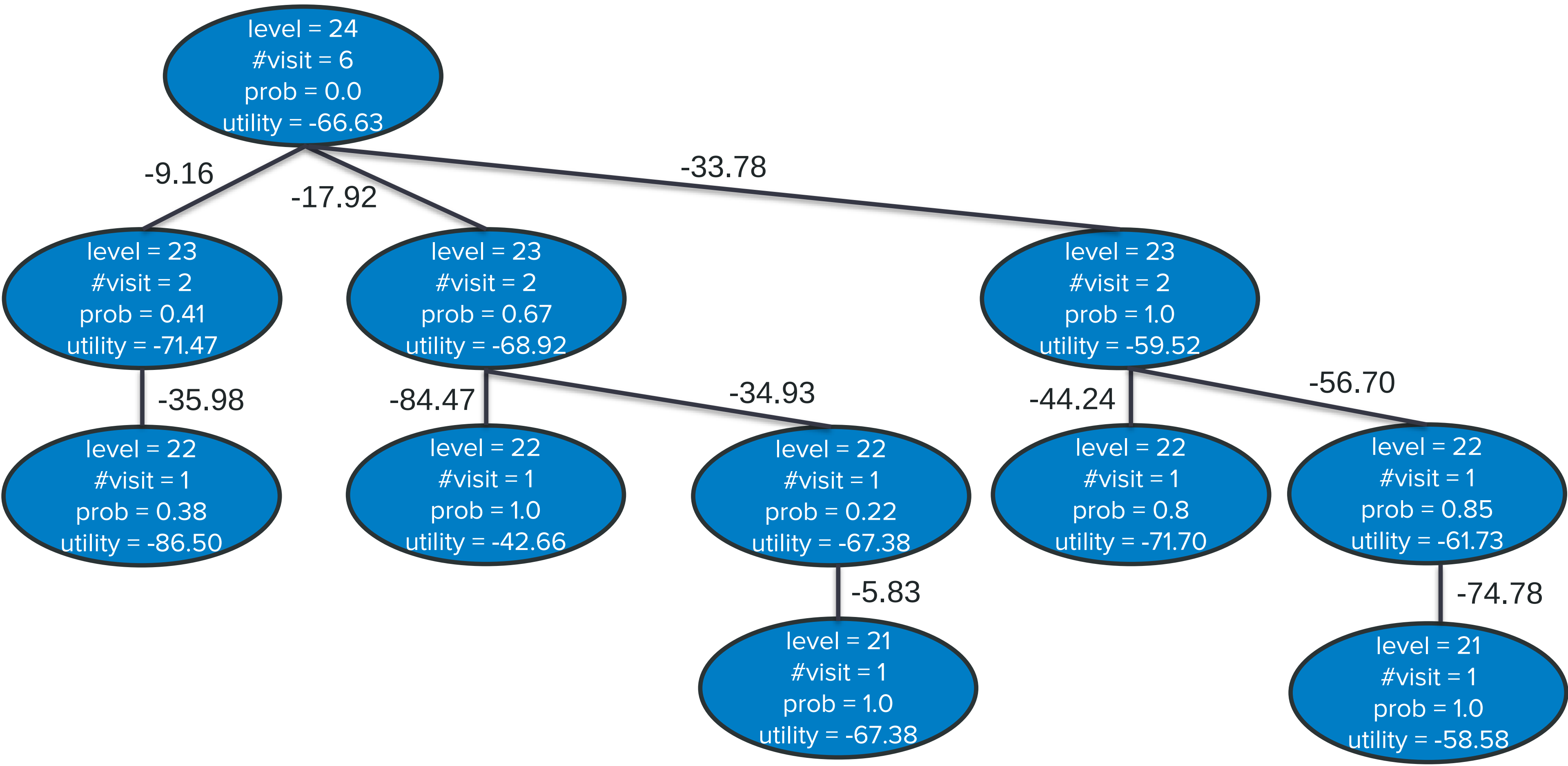
4th Iteration



5th Iteration



5th Iteration



Experiments and Results

Baseline Bidding Strategies for Comparison

MCTS-based Strategies:

MCTS-Vanilla: Discretizes the action space of bid prices. Each action within this space is defined by two multipliers, α_1 and α_2, applied to the lower and upper bounds of feasible bid prices, respectively.

MCTS-SPW: Initially starts with a discrete action space and dynamically grows its size with the number of rollouts as more bid prices are randomly sampled by balancing between exploration and exploitation.

SPOT: Integrates several heuristic techniques to optimize bid prices and strategically place multiple bids in auctions. Specifically, it calculates the standard deviation of clearing prices σ offline and incorporates an external limit price predictor that provides limit-price μ.

Baseline Bidding Strategies for Comparison

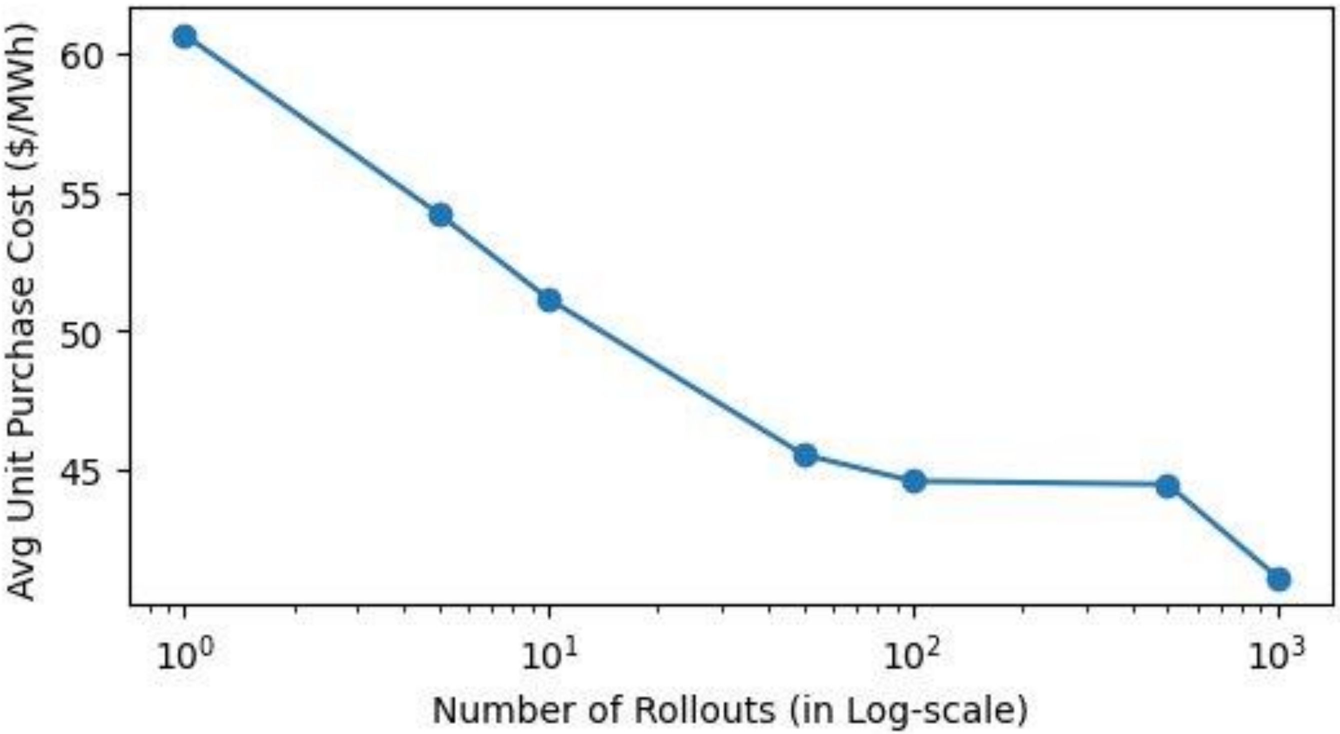
PDA Bidding Strategies:

ZI: Strategy to place random bids within upper and lower bounds

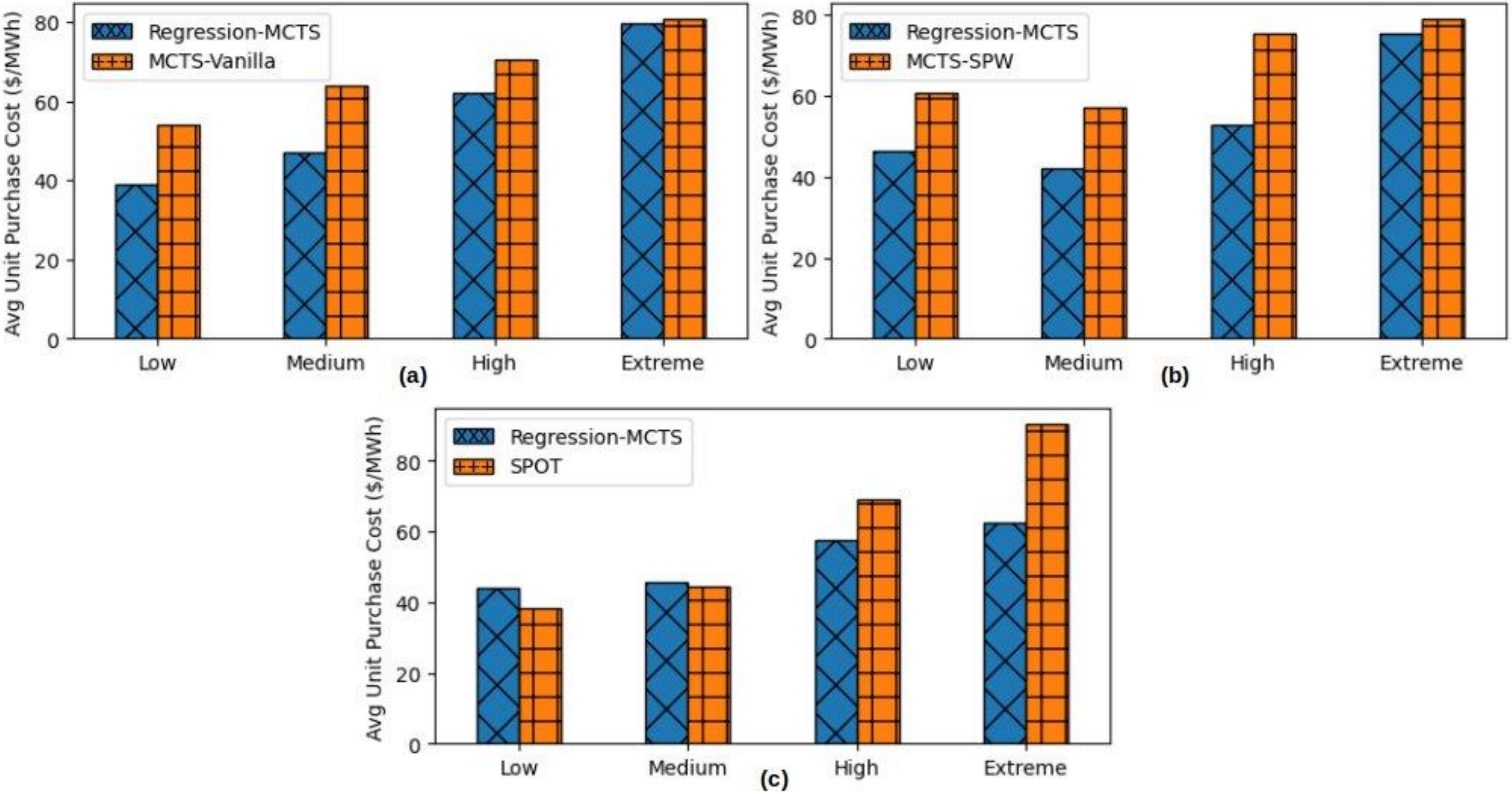
ZIP: Heuristic strategy to adjust the profit margins while placing bids

VV21: Heuristic strategy that models the cost curve of the generating company derived from uncleared ask information available in PDAs. Then, uses the cost curve information to place bids.

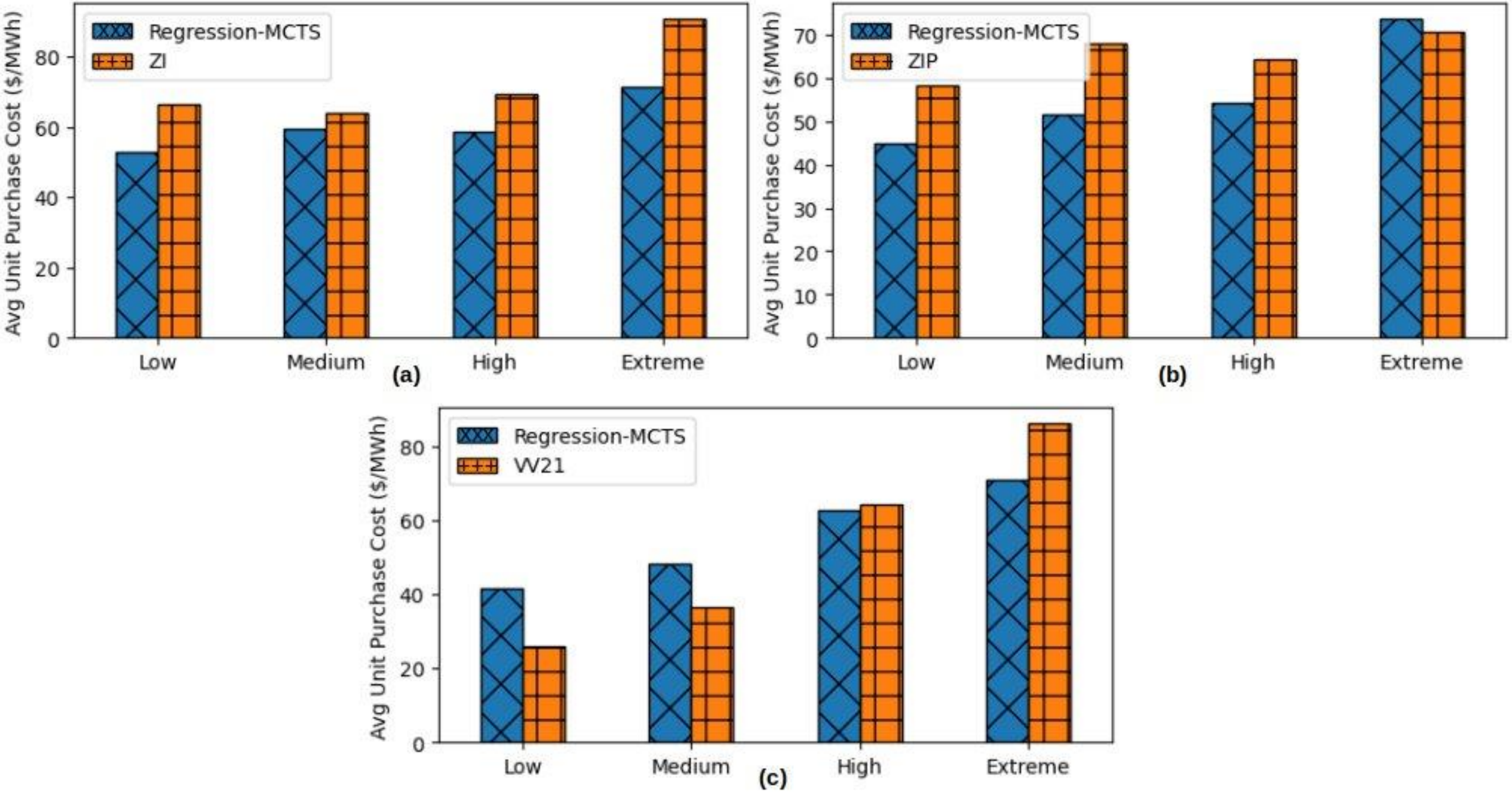
Set-1: Performance R-MCTS with #rollouts



Set-2: Performance R-MCTS against state-of-the-art MCTS methods



Set-3: Performance R-MCTS against PDA bidding strategies



Thank you !!!

Questions … ?