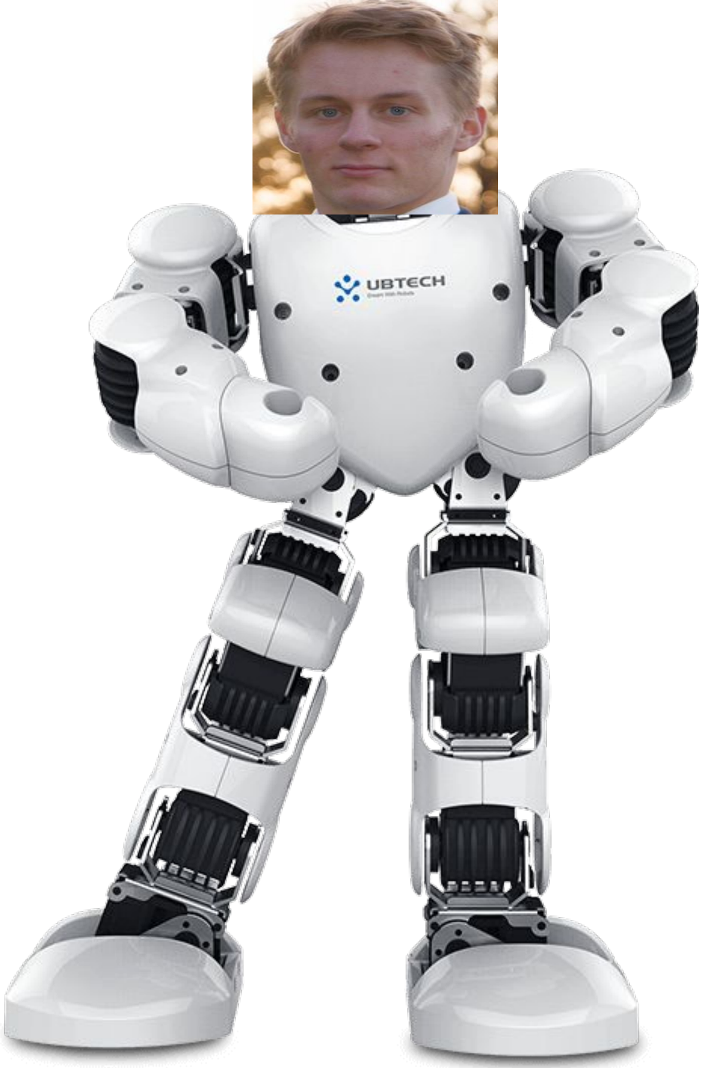
Knut Agent



Concept

- Time-based
    1. Maximum
    2. Average Tit 4 Tat based on opponents change
    3. Random above minimum target

- Always look to accept

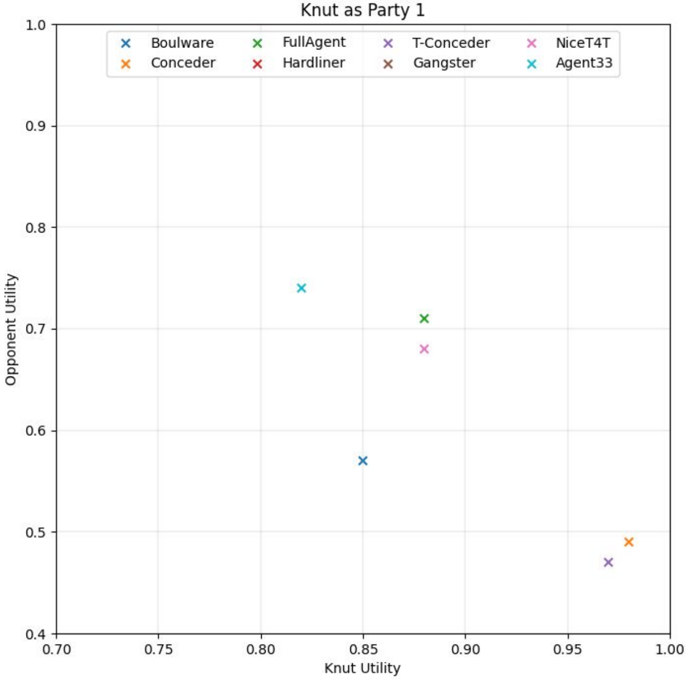
Average T4T

$$\bar{\Delta} = \frac{\sum_{i=1}^{n} (u_i - u_{i-1})}{n-1}$$

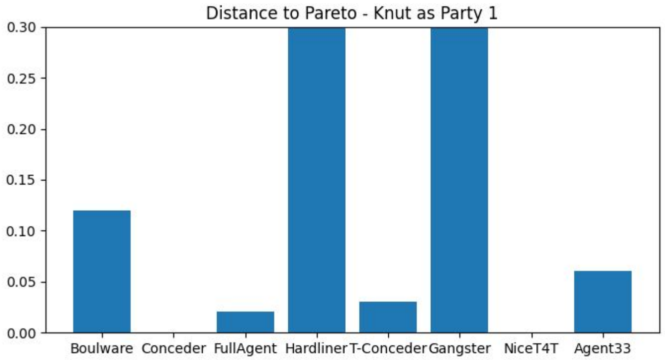
$$\bar{\Delta} = \frac{(u_1 - u_0) + (u_2 - u_1) + (u_3 - u_2) + \cdots + (u_n - u_{n-1})}{n-1}$$

$$\bar{\Delta} = \frac{-u_0 + u_n}{n-1}$$

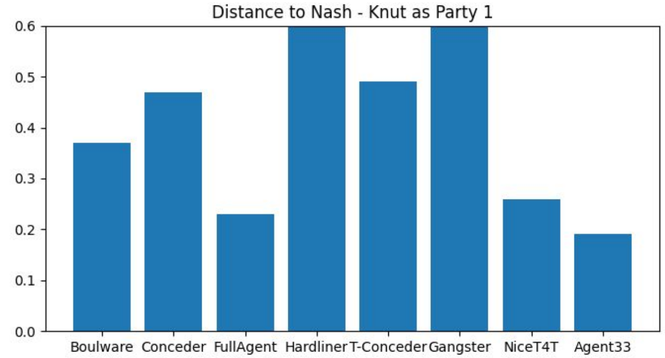
```java
private Bid generateAveragedTitForTatBid()
{
        Bid bid;
        double time_end = timeline.getTime();

        /*
         * If we are early in the negotiation I expect to see smaller changes, and thus uses a larger window
         * If we are closer to the end of the negotiation I expect greater changes, and thus uses a smaller window
        */

        double time_start = time_end - 0.15;
        if (AVERAGE_STOP_TIME - time_end < 0.05) {
                time_start = time_end - 0.075;
        }


        BidHistory bids_received_within_time = new BidHistory();
        bids_received_within_time = bids_received.filterBetweenTime(time_start, time_end).sortToTime();

        Bid first_bid = bids_received_within_time.getFirstBidDetails().getBid();
        Bid last_bid = bids_received_within_time.getLastBidDetails().getBid();

        double first_bid_util = getUtility(first_bid);
        double last_bid_util = getUtility(last_bid);

        double max_utility = getUtility(generateMaximumUtilityBid());
        double min_utility = getUtility(generateMinimumUtilityBid());

        double avg_utility_change = (last_bid_util - first_bid_util)/(bids_received_within_time.getHistory().size()-1);

        /*Not lower than minimum*/
        double utility_start = Math.max(getUtility(my_prev_bid)-(avg_utility_change*1.05), min_utility);

        /*Not higher than maximum*/
        double utility_end = Math.min(getUtility(my_prev_bid)-avg_utility_change*0.95, max_utility);


        //Make sure that there is at least a gap of 0.04 between the lower and upper bound
        double gap = utility_end - utility_start;
        if (gap <= 0.04) {
                utility_start = utility_start + gap - 0.04;
        }

        bid = getBidsOfUtility(utility_start, utility_end, first_bid.getDomain());

        return bid;
}
```



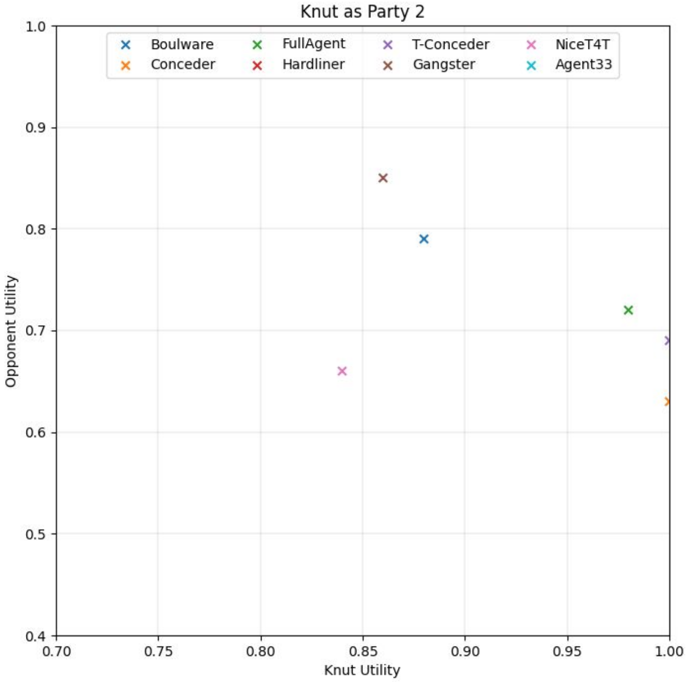
Knut as Party 1

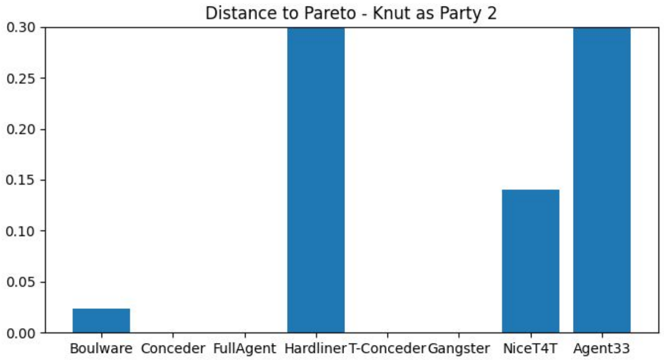


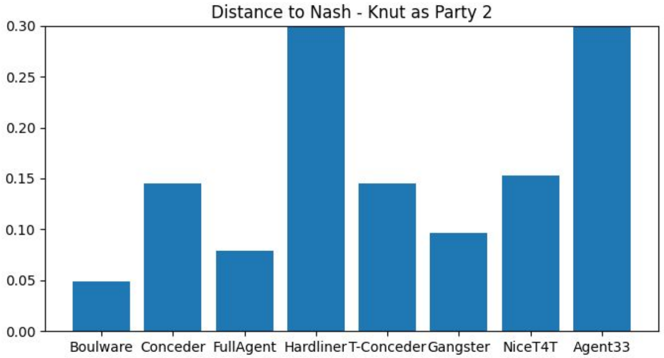
Distance to Pareto - Knut as Party 1



Distance to Nash - Knut as Party 1


Knut as Party 2
Boulware
Conceder
FullAgent
Hardliner
T-Conceder
Gangster
NiceT4T
Agent33
1.0
0.9
0.8
0.7
0.6
0.5
0.4
Opponent Utility
0.70
0.75
0.80
0.85
0.90
0.95
1.00
Knut Utility


Distance to Pareto - Knut as Party 2
0.30
0.25
0.20
0.15
0.10
0.05
0.00
Boulware
Conceder
FullAgent
Hardliner
T-Conceder
Gangster
NiceT4T
Agent33


Distance to Nash - Knut as Party 2
0.30
0.25
0.20
0.15
0.10
0.05
0.00
Boulware
Conceder
FullAgent
Hardliner
T-Conceder
Gangster
NiceT4T
Agent33