TheMemorizer: A Memory-Enhanced Sequential Multi-Deal Negotiation Agent

Abstract

TheMemorizer is an adaptive negotiation agent designed for the ANL 2025 challenge of sequential multi-deal negotiations. The agent employs a sophisticated memory system to track negotiation history, statistical decision-making processes, and role-adaptive strategies for both center and edge agent positions. Key innovations include rejection-aware bidding, progress-sensitive acceptance thresholds, and computational optimization through adaptive outcome space exploration.

1. Introduction

Sequential multi-deal negotiations present unique challenges where agents must balance immediate gains against future opportunities while managing constraints across multiple interconnected negotiations. TheMemorizer addresses these challenges through three core mechanisms: (1) comprehensive memory tracking of negotiation patterns, (2) statistical acceptance criteria that adapt to negotiation progress, and (3) role-specific coordination strategies that optimize for different agent positions in the network.

2. Coordination: Multi-Deal Management Strategy

TheMemorizer implements a dual-architecture approach to handle multi-deal coordination, employing different adapters based on computational feasibility and agent role.

2.1 Adapter Architecture

The agent utilizes two distinct adapters:

- **McufAdapter**: Activated when the complete outcome space can be computed ($\leq 10^4$ cases), enabling optimal decision-making through exhaustive analysis
- **MockAdapter**: Used for large outcome spaces or complex scenarios, employing sampling-based heuristics

2.2 Role-Specific Coordination

Center Agent Coordination: The center agent maintains an agreements list tracking all previous negotiation outcomes and considers the impact of current decisions on future negotiations. When evaluating outcomes, it constructs test contexts including previous agreements and explores potential future combinations using sampled outcomes from subsequent negotiations.

Edge Agent Coordination: Edge agents focus on maximizing individual negotiation utility while incorporating leverage factors based on their position in the negotiation sequence (leverage = negotiation_index + 1), allowing later negotiations to command better terms.

2.3 Computational Optimization

For scenarios where complete enumeration is possible, TheMemorizer employs the McufAdapter to:

- Pre-compute all possible outcome combinations
- Determine if current utility can be improved (can_improve_state())
- Make optimal decisions based on complete information

When computational limits are exceeded, the system gracefully degrades to sampling-based approaches with a maximum of 1,000 sampled outcomes per negotiation.

3. Bidding Strategy: Memory-Enhanced Proposal Generation

TheMemorizer's bidding strategy combines utility maximization with learned negotiation patterns through comprehensive memory tracking.

3.1 Rejection Memory System

The agent maintains detailed rejection statistics in rejection_counts dictionary, tracking:

- i_rejected: Number of times the agent rejected opponent offers
- opponent_rejected: Number of times opponents rejected agent offers

This information feeds into utility calculations as small adjustments (±0.000002) to break ties and favor outcomes with better negotiation dynamics.

3.2 Utility Calculation Framework

Edge Agent Bidding:

utility = base_utility + (0.000002 × i_rejected) - (0.000002 × opponent_rejected)

Center Agent Bidding: The agent constructs test contexts incorporating:

- Previous agreements from completed negotiations
- Current proposal being evaluated
- Projected outcomes for remaining negotiations (initially set to None)

The utility calculation includes progress-based opponent rejection penalties:

utility = base_utility - (0.05 × progress × opponent_rejected × 10^(-(leverage-1)))

3.3 Outcome Space Management

For large outcome spaces (>1,000 outcomes), TheMemorizer implements intelligent sampling, sorting outcomes by utility and retaining the top 1,000 candidates. This ensures computational efficiency while maintaining solution quality.

4. Acceptance Strategy: Statistical Threshold Adaptation

TheMemorizer employs a sophisticated statistical acceptance mechanism that adapts to negotiation progress, role, and outcome distribution characteristics.

4.1 Statistical Acceptance Framework

The core acceptance decision uses a z-score approach:

accept if: offer_utility > mean_utility + (z × std_utility)

Where z is dynamically calculated as:

base_z = 3 × (1 - progress) × agent_type_factor
z = base_z / (1 + 5 × std_ratio)

4.2 Adaptive Parameters

Progress Sensitivity: The base threshold decreases linearly with negotiation progress, becoming more accepting as deadlines approach.

Role Adaptation:

- Edge agents: agent_type_factor = 1.0
- Center agents: agent_type_factor = 1.2 (more selective due to coordination requirements)

Variance Adjustment: The algorithm reduces acceptance thresholds when outcome utilities have high variance (std_ratio), preventing overly rigid behavior in diverse outcome spaces.

4.3 Backup Acceptance Mechanism

TheMemorizer includes a safety mechanism accepting offers when offer_utility / best_utility < 0.15, preventing deadlocks in scenarios where the statistical threshold might be too restrictive.

4.4 MCUF Integration

The max center utility function have some unique properties compared to the other center utility functions. The utility of the center is determined only by the best deal he closed, while the other deals still improve the utility of the edge agents. The consequences of the current bid can be calculated, as the current bid doesn't affect the future negotiations utility function. We utilize those properties in the McufAdapter to measure the utility of each bid and reject all bid that does not improve the utility function of the center agent. Due to this mechanism the utility of the center agent can only improve between negotiations, while preventing the edge agents any utility 'for free'.

5. Implementation Features

5.1 Debugging and Monitoring

The agent includes comprehensive logging through the my_print() method, tracking proposal decisions, acceptance rationales, and statistical parameters for analysis and debugging.

5.2 Memory Efficiency

TheMemorizer optimizes memory usage by:

- Lazy computation of outcome spaces
- Efficient trace processing that only updates new offers
- Sample caching to avoid redundant calculations

6. Conclusion

TheMemorizer represents a comprehensive approach to sequential multi-deal negotiation, successfully integrating memory-based learning, statistical decision-making, and adaptive coordination strategies. The agent's dual-adapter architecture ensures both optimal performance when computationally feasible and robust heuristic behavior in complex scenarios. The sophisticated acceptance mechanism balances exploitation of known good outcomes with exploration necessitated by time constraints and uncertainty.

The agent's design addresses the core challenges of ANL 2025: managing sequential dependencies, adapting to different roles, and making principled decisions under uncertainty while maintaining computational efficiency across diverse scenario complexities.

Note: TheMemorizer demonstrates significant innovation in memory-enhanced negotiation through its rejection tracking system and statistical acceptance criteria, providing a strong foundation for sequential multi-deal negotiation scenarios.

7. Performance Analysis and Lessons Learned

7.1 Tournament Results

TheMemorizer finished in 20th place (last) out of 20 competing agents in the ANL 2025 tournament with a final score of 5,584. This performance was significantly below expectations, with the winning agent (Agent 20725) achieving a score of 17,293—more than three times higher than TheMemorizer's result.

7.2 Performance Analysis

7.2.1 Critical Design Flaws

Backup Acceptance Logic Error: A critical bug was identified in the backup acceptance mechanism. The condition `(offer_utility / best_utility) < 0.15` was intended to prevent deadlocks but actually caused the agent to accept offers that were significantly worse than optimal outcomes (less than 15% of the best possible utility). This logic should have been reversed to `> 0.85` to accept good offers.

Statistical Acceptance Over-Complexity: The sophisticated z-score based acceptance mechanism, while theoretically sound, may have been too complex for the tournament scenarios. The dynamic adjustment of acceptance thresholds based on variance (`z = base_z / (1 + 5 × std_ratio)`) could have led to accepting suboptimal offers in high-variance scenarios or being too restrictive in low-variance situations.

7.2.2 Strategy Misalignments

Memory System Overhead vs. Benefit: The extensive rejection tracking system provided minimal utility improvements (±0.000002 adjustments) while adding computational complexity. The small magnitude of these adjustments suggests they had negligible impact on actual performance while potentially introducing noise in decision-making.

Role Adaptation Ineffectiveness: The leverage-based coordination strategy for edge agents and the penalty system for center agents may have been poorly calibrated for the tournament scenarios. The agent's attempt to be more selective as a center agent (agent_type_factor = 1.2) might have resulted in missing beneficial agreements.

MCUF Adapter Rigidity: While the McufAdapter was designed to ensure monotonic utility improvement, it may have been overly conservative, rejecting offers that could have led to better sequential outcomes even if they didn't immediately improve current utility.

7.2.3 Computational Strategy Issues

Sampling Strategy Limitations: When outcome spaces exceeded 1,000 options, the agent's sampling approach of retaining only the top utility outcomes may have missed strategically important mid-range options that could have been more acceptable to opponents.

Adapter Architecture Overhead: The dual-adapter system, while conceptually elegant, may have introduced unnecessary complexity without sufficient performance benefits, particularly the threshold-based switching between McufAdapter and MockAdapter.

7.3 Key Lessons Learned

7.3.1 Simplicity vs. Sophistication

The tournament results demonstrate that sophisticated theoretical approaches don't automatically translate to practical performance. Simpler, more robust strategies often outperform complex systems with multiple interdependent components that can fail in unexpected ways.

7.3.2 Testing and Validation

The poor performance highlights the critical importance of comprehensive testing across diverse scenarios. The backup acceptance mechanism bug suggests insufficient validation of edge cases and boundary conditions.

7.3.3 Calibration Challenges

Fine-tuning parameters like acceptance thresholds, leverage factors, and rejection penalties requires extensive empirical testing. The small magnitude adjustments in TheMemorizer's memory system suggest these parameters were not properly calibrated for real tournament conditions.

7.4 Recommendations for Future Development

1. **Simplified Decision Logic**: Replace complex statistical acceptance mechanisms with simpler, more robust threshold-based approaches
2. **Extensive Testing Protocol**: Implement comprehensive testing across varied scenarios before deployment
3. **Parameter Sensitivity Analysis**: Conduct thorough analysis of parameter impacts on performance
4. **Fallback Strategy Validation**: Ensure backup mechanisms improve rather than degrade performance
5. **Performance Monitoring**: Implement real-time performance tracking to identify issues during competition

7.5 Conclusion

TheMemorizer's performance in ANL 2025 serves as a valuable reminder that theoretical sophistication must be balanced with practical robustness. While the agent incorporated several innovative concepts—memory-enhanced bidding, statistical acceptance criteria, and adaptive coordination—the implementation suffered from critical flaws that overshadowed these innovations. The experience provides important insights for future agent development, emphasizing the need for simpler, more reliable strategies backed by extensive empirical validation.

The poor ranking, while disappointing, offers crucial learning opportunities for advancing negotiation agent design. Future iterations should prioritize reliability and empirical validation over theoretical complexity, ensuring that sophisticated strategies are properly implemented and thoroughly tested before deployment.