RivAgent: An agent submitted to the ANAC 2025 ANL league

Jumpei Kawahara
Tokyo University of Agriculture and Technology
kawahara@katfuji.lab.tuat.ac.jp
Japan

June 14, 2025

1 Introduction

In the ANL 2025 competition, agents are required to coordinate multiple sub-negotiations simultaneously while maximizing the overall utility from the perspective of a central coordinator. This structure introduces a unique challenge compared to typical bilateral negotiation settings, as agents must not only evaluate individual offers but also consider the interdependencies among multiple ongoing negotiations.

The agent developed for this competition, named **RivAgent**, aims to effectively coordinate these sub-negotiations through a combination of expected utility estimation and time-dependent concession strategies. One of the key goals of this agent is to adapt its offer generation and acceptance behavior dynamically based on the negotiation progress and the opponent's behavior patterns.

In this report, I detail the architecture and implementation of the RivAgent, focusing on its core strategies: coordination strategy, time-based concession control, bidding logic, and acceptance policy.

2 The Strategy of RivAgent

This section describes the strategy of RivAgent.

2.1 Valuable Settings

This subsection defines the notation and basic settings used in the negotiation strategy, including the structure of the negotiation and relevant variables.

- i: Sub negotiation index (n: Total number of sub negotiations)
- t: step in arbitrary sub negotiation (T: Total number of steps)
- $a \in A$: Arbitrary outcome that accepting offer
- e: Outcome that ending negotiation
- $o \in O$: Arbitrary outcome (m: Total number of outcomes)

2.2 Coordination Strategy

This subsection defines coordination strategies used during negotiation.

2.2.1 OAP: Opponent's Acceptance Probability

Opponent's Acceptance Probability (OAP) estimates the probability that the opponent will accept an offer. It is used in the calculation of ECU.

Algorithm 1 Calculate OAP

Require: Now is i-th sub negotiation
Require: Buffer$(\rho) = \{\rho^1, .., \rho^\beta\}$
Collect Opponent Offer History $\overline{X} = \{\overline{x}_0, .., \overline{x}_{T_{end}}\}$
if $t > 2m$ and $\frac{t}{T} \geq 0.25$ **then** ▷ Calculate ρ
    Calculate $\rho = \frac{\text{CountUnique}(\overline{X})}{m}$
    Append ρ to Buffer(ρ)
end if
if $\beta = 0$ **then** ▷ Calculate OAP
    $OAP = 0.5$
else
    Calculate $\rho' = \sum_j^\beta 0.55^{\beta - j} \rho^j$
    Calculate $OAP = 0.45 + 0.1\rho'$
end if

2.2.2 ECU: Expected Center Utility

Expected Center Utility (ECU) represents the expected utility of a particular outcome from the perspective of the center, taking into account future transitions. It is used in both the bidding and acceptance strategies.

Algorithm 2 Build ECU Tree

Require: Now is i-th sub negotiation
Require: Already calculated OAP
for j: n to i **do**
    for all $o^j \in O^j$ **do** ▷ Calculate ECU using child's ECU
        if $j = n$ **then**
            $ECU^n(o) = \text{CenterUtility}(\{o^1, .., o^{n-1}, o\})$
        else
            $ECU^j(o) = p_1 ECU^{j+1}(a_1) + ... + p_m ECU^{j+1}(a_m) + qECU^{j+1}(e)$
            if o is END NEGOTIATION **then**
                $ECU^j(o) \leftarrow 0.9ECU^j(o)$
            end if
        end if
    end for
    if $j > i$ **then** ▷ Sort ECUs
        $\{ECU^j(o_1), .., ECU^{j+1}(o_m)\} \leftarrow \text{SortByDescend}(\{ECU^j(o_1), .., ECU^j(o_m)\})$
        Remove $ECU^j(o)$ if $ECU^j(o) < ECU^j(e)$
        $p_{rest} = 1.0$ ▷ Calculate Transition Probability
        for k: 1 to $m - 1$ **do**
            $p_k \leftarrow p_{rest} \cdot OAP$
            $p_{rest} \leftarrow p_{rest} - p_k$
        end for
        $q \leftarrow p_{rest}$
    end if
end for

2.3 Time-Based Strategy

This subsection presents a time-based strategy that dynamically adjusts offer thresholds depending on the progress of negotiation and the variety of the opponent's past proposals. The level of concession is controlled based on these factors.

Algorithm 3 Time-Based Strategy

Require: Now is i-th sub negotiation
Require: Already calculated $\{ECU^i(a_1), .., ECU^i(a_{m-1}), ECU^i(e)\}$
Require: $ECU_{max} = ECU^i(a_1)$, $ECU_{min} = ECU^i(e)$
for t: 0 to $T - 1$(Not accept and not end negotiation) **do**
    if $t < 5$ **then** ▷ Calculate α
        $\alpha = 1.7$
    else
        $\nu = \text{CountUnique}(\overline{x_{t-5}}, .., \overline{x_{t-1}})$
        if $\nu = 1$ **then**
            $\alpha = 1.3$
        else:
            $\alpha = 1.7$
        end if
    end if
    $TH_{min} = min(ECU_{min} + (ECU_{max} - ECU_{min})(1 - \alpha^{\frac{t}{T}}), 0.5ECU^i_{max})$ ▷ Calculate threshold range
    $TH_{max} = max(th_{min} + 0.1(ECU^i_{max} - ECU^i_{min}), ECU^i_{max})$
    Doing some process
end for

2.4 Bidding Strategy

This subsection describes the bidding strategy used to select an appropriate offer from a set of candidates that fall within a given threshold range. The selection is based on ECU values and the diversity of issue values.

Algorithm 4 Bidding Strategy

Require: Now is i-th sub negotiation, and current step is t
Require: Already calculated $\{ECU^i(a_1), .., ECU^i(a_{m-1}), ECU^i(e)\}$
Require: Already calculated TH_{min}, TH_{max}
Require: L: Total number of issues
Select offers $\{x_1, .., x_{m'}\}$ (Arbitrary selected offer satisfy $TH_{min} \leq ECU^i(x) \leq TH_{max}$)
if $m' = 1$ **then**
    Proposal offer $\hat{x} = x_1$
else
    for l: 1 to L **do** ▷ Calculate weight
        $w_l = \text{Normalize}(1 - \frac{\text{NumbefOfExistValues}(l)}{\text{TotalNumberOfValues}(l)})$
    end for
    for k: 1 to m' **do** ▷ Calculate preference
        $\text{Preference}(x_k) = \sum_l^L w_l Count(x[l])$
    end for
    Selected index $\hat{k} = \underset{k}{\text{argmax}}\ \text{Preference}(x_k)$ ▷ Select offer
    Proposal offer $\hat{x} = x_{\hat{k}}$
end if

2.5 Acceptance Strategy

This subsection explains the acceptance strategy for incoming offers. The decision to accept or reject is based on a comparison between the ECU of the opponent's proposal and a predefined threshold.

Algorithm 5 Acceptance Strategy

Require: Now is i-th sub negotiation, and current step is t
Require: Already calculated $\{ECU^i(a_1), .., ECU^i(a_{m-1}), ECU^i(e)\}$
Require: Already calculated TH_{min}
Require: $\overline{x}$: Offer proposed by opponent
if $m - 1 = 0$ **then**
    End negotiation
else
    if $ECU^i(\overline{x}) > TH_{min}$ **then**
        Accept offer $\overline{x}$
    else
        Reject offer $\overline{x}$
    end if
end if

3 Evaluation

This section presents the performance of RivAgent. Tables 1, 2, and 3 show a performance comparison between the proposed agent and several baseline agents in the AMR 2025 simulation environment. RivAgent achieved the highest overall performance, particularly in the Dinners scenario. Additionally, it recorded the highest mean performance in both the Target Quantity and Job Hunt scenarios. However, its minimum and first quartile (Q1) scores were lower compared to those of Boulware2025 and Linear2025.

Table 1: Dinners Score (1000 tournaments)

Agent		Min	Q1	Mean	Q3	Max
RivAgent	final scores	**1.050**	**7.018**	**12.045**	**16.002**	**47.600**
	weighted average	**0.053**	**0.368**	**0.796**	**1.340**	**2.140**
Boulware2025	final scores	0.630	6.420	10.589	13.000	42.000
	weighted average	0.035	0.333	0.690	0.917	1.900
Linear2025	final scores	0.840	6.420	10.599	13.000	42.000
	weighted average	0.038	0.317	0.700	0.943	1.900
Conceder2025	final scores	0.840	6.210	10.268	12.465	42.000
	weighted average	0.042	0.317	0.691	0.953	1.900
Random2025	final scores	0.390	3.477	5.118	6.608	14.000
	weighted average	0.018	0.180	0.318	0.444	0.700

Table 2: Target Quantity Score (1000 tournaments)

Agent		Min	Q1	Mean	Q3	Max
RivAgent	final scores	8.150	11.750	**19.169**	**26.262**	**58.000**
	weighted average	0.272	0.392	**1.184**	**2.122**	**2.500**
Boulware2025	final scores	**8.550**	**12.600**	18.719	23.300	53.200
	weighted average	**0.285**	**0.420**	1.136	1.828	2.382
Linear2025	final scores	6.500	11.700	17.201	22.000	47.200
	weighted average	0.217	0.390	1.036	1.740	2.377
Conceder2025	final scores	7.600	11.050	16.247	20.625	46.400
	weighted average	0.253	0.368	0.968	1.603	2.377
Random2025	final scores	4.000	5.450	9.479	12.050	25.600
	weighted average	0.133	0.182	0.603	1.000	1.140

Table 3: job hunt Score (1000 tournaments)

Agent		Min	Q1	Mean	Q3	Max
RivAgent	final scores	7.271	12.442	**81.447**	**120.859**	**336.000**
	weighted average	0.242	0.415	**6.059**	**11.600**	**11.700**
Boulware2025	final scores	8.608	**12.835**	78.455	120.731	336.000
	weighted average	0.287	**0.428**	5.868	11.629	11.700
Linear2025	final scores	**8.738**	12.183	72.858	120.578	336.000
	weighted average	**0.291**	0.406	5.670	11.601	11.700
Conceder2025	final scores	4.333	9.893	72.899	118.643	336.000
	weighted average	0.144	0.330	5.706	11.491	11.700
Random2025	final scores	3.583	5.719	52.931	87.562	256.800
	weighted average	0.119	0.191	4.015	8.234	9.567

4 Conclusions

Through my participation in ANL 2025, I was able to develop a high-performance agent. In particular, in the Dinners scenario, the agent achieved the highest scores across all metrics compared to other agents. In the other scenarios as well, the agent recorded the highest mean, Q3, and maximum scores among all competitors. This strong performance was largely due to the use of Expected Center Utility (ECU) as a guiding principle for both bidding and acceptance strategies, enabling the agent to make globally optimized decisions instead of relying on local utility evaluations.

However, in the Target Quantity and Job Hunt scenarios, the agent's minimum and Q1 scores were lower than those of Boulware2025 and Linear2025. I believe this issue stems from insufficient parameter tuning. For future competitions, I plan to focus more on parameter optimization and resubmit an improved version of the agent.