A Negotiation Meta Strategy Combining Trade-off and Concession Moves

Team GiT GuD
Ashley Dyer, Aaron Post, Paul Quint
Automated Negotiation

Main key for autonomous agent interaction

**Negotiation Protocols**: set of rules that control interactions

Who can participate, negotiation states, events that change state, actions, etc.

**Agents’ Decision Making Models**: how agents act in line with negotiation protocol in order to achieve their objectives

**Negotiation Objects**: range of issues over which an agreement is made
Automated Negotiation

Negotiation Process: distributed search through a space of potential agreements

Proposal: a solution to the negotiation problem
Purpose of Study

To combine two existing negotiation models to improve the negotiation process, and increase agents’ utilities gained from these new offers.

Combining:

negoEngine

- based on concessions

  - Trade-off Strategy

    - Multiple decision variables are traded-off against one another

  - Modify trade-off algorithm to improve performance
Negotiation Strategies

Negotiation is modeled as a thread of exchange between $i$ agents (typically 2).

Each entry to the thread is an offered contract and the final entry is ACCEPT if the last offer was accepted or REJECT if the negotiations ran out of time.

Each offer is an assignment of values to each of $n$ decision variables denoted $D$.

Each agent has a value function for each decision variable, $V_j^i : D_j \rightarrow [0, 1]$.

Each agent weights the value of each variable of a contract so the total value function is in $[0,1]$.

$$V^i(x) = \sum_{j=1}^{n} w_j^i V_j^i(x_j)$$
Two Existing Strategies

negoEngine

and

Trade-off
negoEngine

Uses a linear combination of known tactics that may shift as negotiation proceeds.

Behavior Dependent: Changes overall tactics to imitate the opponent using Tit-for-Tat type strategies.

Time dependent: Concedes value as time passes to ensure an ACCEPT before the negotiation deadline.
Trade-off

“The main idea of this tactic is to find a proposal with the same utility [for me] as the previous one [I] offered, but expected to be more acceptable for [my] opponent”

\[
\text{trade-off } f^a(x, y) = \arg \max_{z \in iso^a(\theta)} Sim(z, y)
\]

Iterate through decision variables of `x`, changing each one to make it more similar to `y` without changing the value to agent `a` (don’t give up aspirations)
Smart Trade-Off

How do we choose which variables to change first when making a trade-off? (This is significant in outcome)

Ans: Try to guess which variables are important to opponent by examining the variability between offers within a window of size $m$.

Why does this work?

Ans: I won’t budge on what’s important to me.

It’s smarter to change opponent’s less important variables first.
Why a Meta Strategy?

Both strategies individually have disadvantages:

negoEngine concedes value at every offer

Trade-Off doesn’t know how to concede value intelligently

Answer: To get the best of both strategies, we combine them
Meta Strategy

1. Exploit current aspiration level using smart trade-off as much as possible the opponent has offered something less optimal to me than it’s previous offer (deadlock)

2. When deadlock is detected, use the negoEngine tactic for a round to make an offer that reduces my aspiration level

3. Repeat until ACCEPT or the deadline is reached

Time Complexity: $O([\text{number of decision variables}] \times [\text{variability window size}])$
Negotiation Experiment

An experiment was setup to test the negotiating effectiveness of the meta strategy.

For this negotiation process one agent makes an offer and the opponent responds with a counter offer. The negotiation deadline is set to a maximum 40 offers per agent.

Five different negotiating strategies will be tested. Each pair of negotiation agents will be ran 100 times to find the average utility achieved.
Negotiation Strategies Tested

NegoTO: uses the meta strategy defined earlier. Employs the trade-off strategy until a deadlock is found, then makes an offer using the negoEngine.

Alternate: alternates between both negoEngine and trade-off strategies

Random: randomly selects a strategy between negoEngine and trade-off

TO: uses the trade-off tactic while the utility of the offer received is higher than the previous, if not it lowers aspiration level by .05

Nego: uses the negoEngine strategy
Measured Results

Utility Product: Product of the resulting utility of both agents after an agreement is reached

Utility Difference: Difference between the resulting utility of the agreement

Both of these factors are important for negotiations. For most negotiations the utility product should be high and utility difference should be low so both parties get a good deal of similar value. If the negotiations are a competition then a higher utility difference is desired to achieve a larger win.
Example Negotiation

Shown are graphs of sample negotiations

X axis: perceived seller utility

Y axis: perceived buyer utility

Both parties make concessions with their offers in order to eventually agree upon an outcome between their initial utility level
Results

| agent_i | $V^a(x)$ | $V^i(x)$ | * | $|$ |
|---------|----------|----------|---|----|
| Negoto  | 0.611    | 0.572    | 0.350 | 0.039 |
| Random  | 0.649    | 0.515    | 0.333 | 0.135 |
| Alternate | 0.634    | 0.514    | 0.326 | 0.120 |
| TO      | 0.734    | 0.490    | 0.360 | 0.244 |
| Nego    | 0.742    | 0.303    | 0.224 | 0.439 |

| agent_i | $V^a(x)$ | $V^i(x)$ | * | $|$ |
|---------|----------|----------|---|----|
| Negoto  | 0.652    | 0.592    | 0.332 | 0.039 |
| Random  | 0.592    | 0.553    | 0.327 | 0.039 |
| Alternate | 0.608    | 0.543    | 0.330 | 0.065 |
| TO      | 0.658    | 0.512    | 0.337 | 0.146 |
| Nego    | 0.630    | 0.399    | 0.252 | 0.231 |

| agent_i | $V^a(x)$ | $V^i(x)$ | * | $|$ |
|---------|----------|----------|---|----|
| Negoto  | 0.543    | 0.613    | 0.333 | 0.039 |
| Random  | 0.576    | 0.558    | 0.321 | 0.018 |
| Alternate | 0.550    | 0.574    | 0.316 | 0.024 |
| TO      | 0.598    | 0.562    | 0.336 | 0.036 |
| Nego    | 0.637    | 0.407    | 0.259 | 0.230 |

| agent_i | $V^a(x)$ | $V^i(x)$ | * | $|$ |
|---------|----------|----------|---|----|
| Negoto  | 0.437    | 0.776    | 0.339 | 0.339 |
| Random  | 0.562    | 0.606    | 0.340 | 0.044 |
| Alternate | 0.503    | 0.638    | 0.321 | 0.135 |
| TO      | 0.636    | 0.565    | 0.360 | 0.071 |
| Nego    | 0.579    | 0.453    | 0.262 | 0.127 |

| agent_i | $V^a(x)$ | $V^i(x)$ | * | $|$ |
|---------|----------|----------|---|----|
| Negoto  | 0.341    | 0.728    | 0.248 | 0.387 |
| Random  | 0.483    | 0.591    | 0.286 | 0.109 |
| Alternate | 0.423    | 0.605    | 0.256 | 0.183 |
| TO      | 0.484    | 0.600    | 0.290 | 0.115 |
| Nego    | 0.506    | 0.494    | 0.250 | 0.011 |

Key:
- $agent_i$ indicating opposing agent
- $V^a(x)$ denoting agent utility
- $V^i(x)$ denoting opponent utility
- * denoting utility
- $|$ product of utility
- $|$ difference of utility
Experiment Analysis

The NegoTO tactic is clearly the strongest, winning all matchups with $V^a(x) > V^i(x)$ where $V^a(x)$ is NegoTO and $V^i(x)$ is all other tactics. In general the utility product is higher and utility difference is lower when one of the agents is using NegoTO as opposed to other matchups. This shows that the NegoTO strategy advances negotiations and achieves fair agreements for both agents.

Both the Alternate and TO strategy obtain high personal utility and would do well in a competition, but have higher utility differences.

Nego consistently does the worst because it concedes to quickly reach a deal
Extending to $n$ agents

Three aspects to consider:

- How $n-1$ proposals from $n-1$ agents might affect behavior of concession tactics
- How to extend trade-off algorithm to handle $n-1$ proposals
- Need to define a new negotiation protocol for $n$ agents
Extending to $n$ agents - Concession Tactics

Some tactics are not influenced by proposals received during negotiation

Ex: time-dependent and resource-dependent tactics

Behavior-dependent tactics: different family of tactics that depend on proposals from other agents

Generate new proposals based on how other agents behave

Try to model opponent’s behavior
Extending to $n$ agents - Trade-off Algorithm

$$x' = \arg\max_{x_i \in \{x_1, \ldots, x_p\}} \min\left\{\sum_{j=1}^{n-1} (1 - Sim(x_i, y_j))^2 \right\}$$
Extending to $n$ agents - Negotiation Protocol

If an agent’s proposal to the rest of the $n-1$ agents is accepted, the process finishes.

Otherwise, the same or another agent creates a new offer.

If no offer is reached by the deadline ($t_{\text{max}}$) the process also finishes.
Conclusion

The negoEngine’s consideration of time and behavior coupled with the trade-off algorithm’s consideration of all possible offers at a utility level make up the meta strategy devised.

This meta strategy outperformed the other strategies tested in all cases.

Using the NegoTO tactic improved negotiations utility product and reduced utility difference, giving a higher utility level to both sides and making fairer deals.
Questions