SCMLOneShot: Patient Agent

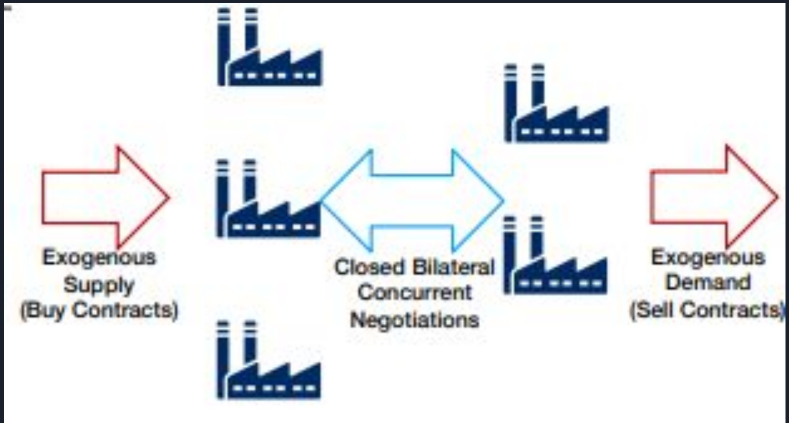
Chris Mascioli and Amy Greenwald

Contents

- Overview of the Game
- Other Agents
- PatientAgent Design
- Results

OneShot Description

- The game consists of 2 layers of agents
  - The first layer is given a set of exogenous buy contracts
  - The second layer is given a set of exogenous sell contracts
- Each agent in one layer has a negotiation thread with all agents in the other layer
- The negotiation issues are:
  - Quantity
  - Price
- The agents pay penalties when they are unable to reach agreement with other agents
- Goal of the game is to accrue the greatest profit at the end



OneShot World - Respond/Propose Order

- At the beginning of each day one layer is selected to make the first proposals (at random, but not uniformly)
- The layer that does not make the first proposal gets to make the last response

Time Step	Layer 0	Layer 1
0	Propose	Respond/Propose
1	Respond/Propose	Respond/Propose
2	Respond/Propose	Respond/Propose
...	...	...
19	Repond/Propose	Respond

OneShot World - Agent Placement

- The world is populated by a number of player agents and some environmental agents
- After adding one of each player agent to the world, the world is then populated by three additional environmental agents provided by the competition
  - SingleAgreementAspirationAgent
  - GreedyOneShotAgent
  - GreedySyncAgent
- Due to how the negotiations are configured, it is advantageous to be more towards the top in each layer (as you'll get offers sooner), but PatientAgent did not exploit this

Contents

- Overview of the Game
- Other Agents
- PatientAgent Design
- Results

AdaptiveAgent (AA) and LearningAgent (LA)

- Two agents given in the tutorial
- All the best-performing OneShot agents from last year built off them
- Both agents share an acceptance strategy
  - If q_offer > q_needed, reject offer
  - Otherwise if u(p_offer) ≥ u(p_next_offer), accept
  - Otherwise reject offer
- Both agents also share an offer strategy
  - q_offer = q_needed
  - p_offer = (p_best - p_worst)*f(t) + p_worst
    - f(t) is some decreasing function of time that goes from 1 to 0
- Agents differ in how they set p_worst
- Performance not significantly different (although LA has a slight edge most of the time)

AdaptiveAgent and LearningAgent

- The agents perform very well
- It’s hard to find something that consistently beats them
- They amplify the effect of having bad agents in your layer, as one agent from your layer giving a bad price locks you in forever
- An agent could, at the first step of the negotiation, offer every agent in the other layer the best price for them and thus force all further trading to be at that price
  - Such an agent doesn’t perform very well in testing, but the threat of such a strategy shuts down a number of other possible strategies

Contents

Motivating Observations

- Trading is significantly better than not trading, even at the worst price
  - u(0, 0) << u(p_worst, q_exog) < u(p_best, q_exog)
- AA/LA don't make trades until very late (in worlds with only LA/AA, there are never any successful contracts until the last step of the negotiation) so early negotiation steps can be ignored
- Strategy as buyer vs seller doesn't really matter, but strategy as first-to-offer vs last-to-offer is significant
- Starting balances are very high so there's no risk of bankruptcy

PatientAgent - First-to-Propose

- When PatientAgent is first to propose, it inherits from AdaptiveAgent and plays exactly the same as it does
- We chose to inherit from AdaptiveAgent instead of LearningAgent so that its negotiations when it was using the actual PatientAgent strategy (not just the inherited AA strategy) would not impact future negotiations

PatientAgent - Second-to-Propose

- Acceptance Strategy
    - Before the final time step
        - If p_offer == p_best, we accept
        - Otherwise, reject
    - At final time step
        - “Patient look-ahead” (Baarslag, 2021)
            - Waits for all offers and then picks the bundle with the highest utility
            - Ties broken by the package offered by the group with the lowest total balance
- Offer Strategy
    - q_offer - current q needed
    - p_offer - p_best

PatientAgent - Opponent Modeling

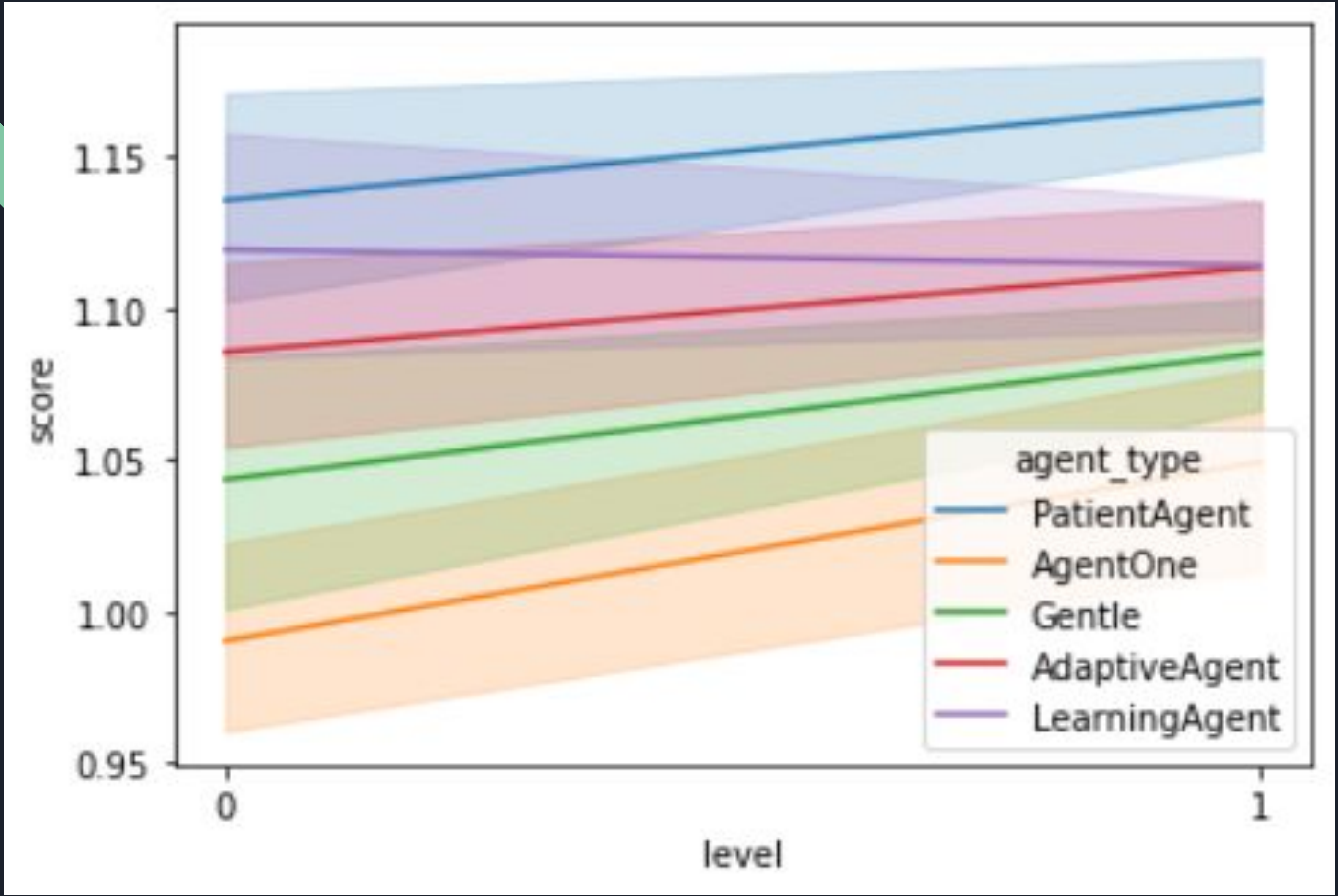
- The aspiration parameters from the family of AdaptiveAgent and LearningAgent are not easily learnable
- PatientAgent builds a simple behavioral model to find SingleAgreementAgents and offers them small quantities at a good price to eliminate them from the simulation for the day to prevent them from trading favorably with a competitor

PateintAgent - Future Improvements

- If other agents are using the WAIT response, PatientAgent sometimes ends up out of sync so does not receive all its offers correctly
- Improve opponent modeling to detect the other built-in agents (GreedyOneShotAgent and GreedySyncAgent)

Contents

- Overview of the Game
- Other Agents
- PatientAgent Design
- Results



score
1.15
1.10
1.05
1.00
0.95
0
1
level
agent_type
PatientAgent
AgentOne
Gentle
AdaptiveAgent
LearningAgent

Acknowledgments

- Amy Greenwald, Brown
- Yasser Mohammad, NEC

Contact

For any questions/discussion, feel free to reach out to me at christopher_mascioli@brown.edu

References

- Tim Baarslag, Tijmen Elfrink, Thimjo Ko ¸ca, and Faria Nassiri Mofakham. Bargaining Chips: Coordinating One-to-Many Concurrent Composite Negotiations. page 8, 2021.