ANAC 2018 - Agent Description
Report on the Virtual Agent "Emma"
Human-Agent Negotiation Challenge

Sylvain Daronnat
Strathclyde University
sylvain.daronnat@strath.ac.uk

Introduction

The goal of this agent is to maximize its score by using the time limit and the apparent "fairness" of its offers as a leverage over the player's actions. This version of the agent is based on a modified version of the BuildingBehavior, NiceExpression and NiceFreeMessage and Core Java files included in the IAGO framework.

Assumptions

Firstly, we assumed that the agent would not be able to access the player's point table during the negotiation process. Moreover, when asking about the player's preferences, it is highly possible that the opponents may deliberately lie in order to maximize their gains and/or minimize their losses. Consequently, the most reliable way to know about the opponent's preferences is to send or reject offers. As a result, we purposely chose to reply with evasive answers when the players asks about the agent's preferences in order to prevent further counter-productive interactions. We also put an emphasis on the "fairness" of the agent's offers which is an important parameter to take into account during a negotiation process. This is why this agent will always try to convince the opponent that each of its decisions are, in a way, fair and perfectly acceptable for everyone.

Features

Here is a list of the features implemented in the virtual agent.

- **Offers based on the quantity of the items**
  During the first half of a round, the agent will react to offers deemed "unacceptable" by sending counter-offers which apparent fairness are based on the quantity of items available in the current round. Depending on the differences in term of value per items for the player and the agent, the outcome in terms of points can be very different while still distributing the items evenly between both parties.
  Here is the pseudo-code representing the distribution of items based on their quantity:

```
boolean AdvantageAgent = True
Integer AgentRepartition, PlayerRepartition = 0
for Item x in list of Current Items do
    Integer Modulo = Modulo(Quantity of Item x)/2
    if AdvantageAgent = True then
        AgentRepartition = (Quantity of item x / 2) + Modulo
        PlayerRepartition = Quantity of item x - AgentRepartition
        AdvantageAgent = False
    else
        PlayerRepartition = (Quantity of item x / 2) + Modulo
        AgentRepartition = Quantity of item x - AgentRepartition
        AdvantageAgent = True
    end if
end for
{After this initial repartition, if the Agent's score is still lower than the threshold set for the current round, we keep adding the least valuables items to the Agent's offer until we reach it.}
```

- **Offers based on the value of the items**
  During the second half of a round, the agent will issue counter-offers based not on the

quantity of items for the current round, but on their values. Depending on the number of items available in the current round, the offer will be more or less biased in the agent's favor.

Here is the pseudo-code computing an offer based on the values of the items:

```
1:  boolean AdvantageAgent = True
2:  Integer AgentRepartition, PlayerRepartition, CurrentPoints = 0
3:  for Item x in list of Current Items do
4:      while CurrentPoints < TotalPossiblePoints/2 and AgentRepartition of item x < Total Quantity of Item
        x do
5:          CurrentPoints += Points from Item x
6:          AgentRepartition += 1
7:      end while
8:      PlayerRepartition = Total Quantity of Item x - AgentRepartition
9:      CurrentPoints = 0
10: end for
```

- **The Agent's First Offer**
  After a set amount of time following the beginning of a new game and if the opponent is not in the process to make any offer, the agent will issue an initial offer in order to begin the negotiation process. This offer is set to come early and be relatively fair for the first round of the game (in order to gain the trust of the player who is just discovering the negotiation interface) and set to come later and be increasingly more biased in the agent's favor in the following rounds of the experiment (in order to try and use the participant's trust in the system to accept offers as quickly as they may have in the previous rounds).

- **The Agent's Final Offer**
  Near the end of the current round, the agent will issue the "fairest" offer possible based on its current heuristic in an attempt to prevent the round from ending with a BATNA. The offer will be sent with a message underlying the little amount of time remaining.

- **The Agent's Acceptance Model**
  Depending on the current round and the amount of time remaining, the agent's acceptance model will be set to either only accept offers that goes in its favor, fair offers with no clear advantage for neither of the parties or slightly unfair offers biased in the opponent's favor. No matter which round is currently being played, the acceptance model will be set to be less restrictive as times goes by in order to avoid ending in a scenario where everyone loses.

- **Emotions/Questions to the Agent**
  As explained in the "Assumptions" section of the document, we want to prevent the participant from interacting too much with the agent via messages/emotions as it can prove inefficient to settle on a offer quickly. Interactions are still possible but the agent's messages are set to prevent the player from wanting to further interact with the agent. Here is an example of these answers in context:
  Participant: "Do you like X more than Y?"
  Agent: "Yes, I like X, but let's get down to business, send me an offer."

- **Welcome Messages**
  Depending on the previous amount of time required to come to an agreement on the previous round, the agent will display different messages in order to get the player to settle more quickly on an offer at the new round and thus, maximize the amount of points gained by the agent. Here are some examples of these messages:
  After a long round: "Back at it again, hopefully this time we'll settle on something quicker."
  After a short round: "Last round was amazing, let's finish this one as efficiently!"