

Homework 2

ID2209

Distributed Artificial Intelligence and Intelligent Agents

Kim Hammar

Due Date: 22 November 2016

Problem Statement

Implementing the FIPA Dutch Auction Interaction Protocol [1] using the JAVA Agent Development Framework (JADE) [2] and laying out a theoretical model of the the game mechanisms involved in dutch auctions between autonomous self-interested agents. The purpose of the homework was to get further practice in developing MultiAgentSystems (M.A.S) and in particular M.A.S that involves *negotiations* between agents.

Main problems and solutions

- *Implementing the FIPA dutch action protocol in JADE*
 - Agent design - Designing and implementing auctioneer and bidder agents.
 - Society design - Implementing the interactions of the protocol as defined by the specification [1].
- *Establish a theoretical model for a specific scenario and apply concepts from game theory*

All possible strategies for the agents were considered to find which strategies are in Nash equilibrium.

Task 1: Implementation

Agent design

Two type agents are used for the dutch auction scenario:

- **ArtistManagerAgent**

The **ArtistManagerAgent** is the *auctioneer* in this scenario; it auctions out artifacts from artists through dutch auctions. When carrying out a dutch auction the auctioneer tries to find the market price for a given good, i.e the auctioneer starts out offering the good at some artificially high price and then continuously lowers the price for each round until some bidder accepts the price or the price reaches the reserved price. The reserved price is the lowest price that the auctioneer is willing to sell the good for, if no bidder accepts that price then the auction is cancelled and the good is not sold.

The **ArtistManagerAgent** is self-interested and wants to sell goods for as high price as possible to increase its personal revenue. For each auction i the **ArtistManagerAgent** will select a strategy s_i consisting of:

I Initial price

II Rate of reduction (i.e how much to lower price from one round to the next one if no buyer was found)

III Reserve price

Frankly, a self-interested auctioneer with infinite time would put the initial price arbitrary high and the rate of reduction arbitrary low since that would guarantee that the auctioneer would receive the highest price possible for a certain good. However if we assume that the auctioneer to some degree values his time, then he would start the auction at an initial price that is higher than what he expect that good to be sold for, but still within a realistic price-range. Further more, the auctioneer would choose a rate of reduction high enough such that participans in the auction that chose not to bid in a previous round might consider to bid in the next round.

The best strategy for the auctioneer is the strategy that optimizes the expected revenue. The expected revenue depends on the type of auction as well as the strategies of the bidders. Dutch auction gives best expected revenue if the agents use *risk-averse* strategies.

The `ArtistManagerAgent` is implemented with a `FSMBehaviour` with all the different states of a dutch auction, i.e: `FIND_BIDDERS_STATE`, `OPEN_AUCTION_STATE`, `SEND_CFP_STATE`, `COLLECT_BIDS_STATE`, `MODIFY_PRICE_STATE`, `SELECT_WINNER_STATE`, `CLOSE_AUCTION_STATE`.

- **CuratorAgent**

The `CuratorAgent` acts as a *bidder* in this scenario, it receives the current price of each round in the dutch auction and can choose to either bid and accept the price or to ignore the price (i.e not bid). The agent is self-interested and wants to obtain artifacts that it is interested in for as low price as possible.

Typically for each dutch auction i a participating bidder would have a personal valuation of the good that is being auctioned and then follow a strategy s_i that decides when to bid and when not to bid. The agent could be *risk-neutral* and bid for the auction at the first round that has an announced price less than or equal to its private valuation. The agent could also be *risk-averse*, if the true value of some good is unknown an agent might be willing to bid higher than its private valuation. Or, the agent could be *risk-inclined* and try to obtain the good for a price lower than its private valuation but with a increased risk of loosing the good to another bidder.

The essential thing here is that the best strategy for a particular `CuratorAgent` depends on what strategies the auctioneer and other bidders choose. For instance if a curator agent knows that the auctioneer has a reserved price far lower than its own valuation of the good and that the other bidders will not bid unless the price reaches some even lower value, then the best strategy for the agent is to be *risk-inclined*. Unless the winner of the auction has complete information about the auctioneer's and the other bidder's strategies then it will always be susceptible to the winner's curse, e.g the agent doesn't know if it should be happy with winning the auction or worried because he might have overvalued the good.

The `CuratorAgent` is implemented with a `ParallelBehaviour` for receiving different messages of the FIPA Dutch Auction Interaction Protocol and taking the appropriate action.

Society design

The implementation follows the specification of the FIPA Dutch Auction Interaction Protocol [1] to the full, it uses exactly the same messages and interactions as the specification suggest. In case of multiple, competing and simultaneous bids the first-come-first-served principle is applied.

Task 2: Game mechanisms

To find a *pure-strategy* nash equilibrium we consider each possible combination of strategies, and for each combination we check whether this combination forms a best response for every agent.

Agents

$$Agents = \{\text{ProfilerAgent (PA)}, \text{ArtistManagerAgent (AMA)}, \text{CuratorAgent (CA)}\}$$

Actions

$$Ac_{AMA} = \{\text{sell-high-quality (HQ)}, \text{sell-low-quality (LQ)}\}$$

$$Ac_{CA} = \{\text{quote-based-on-demand (QD)}, \text{quote-based-on-interest (QI)}\}$$

$$Ac_{PA} = \{\text{buy (B)}, \text{not-buy (NB)}\}$$

Outcomes

3 agents having 2 distinct actions each means $2^3 = 8$ different combinations of actions and outcomes.

$$\Omega = \{\omega_1, \omega_2, \omega_3, \omega_4, \omega_5, \omega_6, \omega_7, \omega_8\}$$

Environment function

The outcomes depend on the *combination* of actions performed by the different agents, which can be modelled with an *environment function* τ .

$$\begin{aligned} \tau(HQ, QD, B) &= \omega_1 & \tau(HQ, QD, NB) &= \omega_2 & \tau(HQ, QI, N) &= \omega_3 \\ \tau(HQ, QI, NB) &= \omega_4 & \tau(LQ, QD, B) &= \omega_5 & \tau(LQ, QD, NB) &= \omega_6 \\ \tau(LQ, QI, B) &= \omega_7 & \tau(LQ, QI, NB) &= \omega_8 \end{aligned}$$

Utility functions

The utility function u_i maps outcomes to utility for Agent i . The utilities are based on the assumptions listed in the assignment.

ArtistManagerAgent

$$\begin{aligned} u_{AMA}(\omega_1) &= 1 & u_{AMA}(\omega_2) &= -2 & u_{AMA}(\omega_3) &= 1 & u_{AMA}(\omega_4) &= -2 \\ u_{AMA}(\omega_5) &= 2 & u_{AMA}(\omega_6) &= -1 & u_{AMA}(\omega_7) &= 2 & u_{AMA}(\omega_8) &= -1 \end{aligned}$$

Preference ordering of ArtistManagerAgent for the different outcomes:

$$\{\omega_5, \omega_7\} \prec_{AMA} \{\omega_1, \omega_3\} \prec_{AMA} \{\omega_6, \omega_8\} \prec_{AMA} \{\omega_2, \omega_4\}$$

CuratorAgent

$$\begin{aligned} u_{CA}(\omega_1) &= 1 & u_{CA}(\omega_2) &= 1 & u_{CA}(\omega_3) &= 1 & u_{CA}(\omega_4) &= 1 \\ u_{CA}(\omega_5) &= 1 & u_{CA}(\omega_6) &= 1 & u_{CA}(\omega_7) &= 1 & u_{CA}(\omega_8) &= 1 \end{aligned}$$

ProfilerAgent

$$\begin{aligned} u_{PA}(\omega_1) &= 2 & u_{PA}(\omega_2) &= 0 & u_{PA}(\omega_3) &= 2 & u_{PA}(\omega_4) &= 0 \\ u_{PA}(\omega_5) &= 1 & u_{PA}(\omega_6) &= 0 & u_{PA}(\omega_7) &= 1 & u_{PA}(\omega_8) &= 0 \end{aligned}$$

Preference ordering of ProfilerAgent for the different outcomes:

$$\{\omega_1, \omega_3\} \prec_{PA} \{\omega_5, \omega_7\} \prec_{PA} \{\omega_2, \omega_4, \omega_6, \omega_8\}$$

Since the CuratorAgent cannot affect the outcome and has a role of a simple intermediate point in the buying and selling between ArtistManagerAgent and ProfilerAgent, we can neglect it in the analysis of strategies and nash equilibriums.

There exists a dominant strategy for the ProfilerAgent, which is to always buy, however there exist no dominant strategy for the ArtistManagerAgent, the best strategy for the ArtistManagerAgent depends on the strategy chosen by the ProfilerAgent. The payoff matrix between the agents looks like the following.

		ArtistManagerAgent	
		HQ	LQ
ProfilerAgent	B	(2, 1)	(1, 2)
	NB	(0, -2)	(0, -1)

As the matrix shows, there is a single nash equilibrium, namely the actions: (B, LQ) . Assuming that ProfilerAgent chooses to buy the good the best strategy for ArtistManagerAgent is to produce/sell a low quality product and likewise assuming that the ArtistManagerAgent chooses to produce/sell a low-quality product the best strategy for the ProfilerAgent is to buy, i.e. neither agent has any incentive to deviate from the equilibrium. Note that the nash equilibrium in this case is also Pareto efficient.

The strategy for finding the nash equilibrium can be done in steps, since there exists a dominant strategy for ProfilerAgent we can exclude all outcomes that are dominated, since in either of the dominated cases the ProfilerAgent would always prefer to choose another strategy, no-matter what strategy ArtistManagerAgent chooses:

		ArtistManagerAgent	
		HQ	LQ
ProfilerAgent	B	(2, 1)	(1, 2)
	NB	(0, -2)	(0, -1)

Now there are only two combination of strategies to analyze, and we then see that if the ProfilerAgent chooses the strategy to buy then the dominant strategy for ArtistManagerAgent is to produce a low-quality product, so then we get:

		ArtistManagerAgent	
		HQ	LQ
ProfilerAgent	B	(2, 1)	(1, 2)
	NB	(0, -2)	(0, -1)

Conclusions

Auctions are mechanisms to reach agreement on the issue of allocating resources between entities and can be used just as well for agents as for humans. Dutch auctions are a special type of

auction that is *open-cry descending*, which requires minimal communication between bidder and auctioneer agents to reach agreement on allocating resources.

In a M.A.S with self-interested agents that are operating in the same environment, finding the *best decision* for each agent in many cases resembles studies from game theory. Concepts like dominant strategies, Pareto efficiency and Nash equilibrium can often be applied when analyzing interactions between self-interested agents in a M.A.S.

Attachments

Documented source code can be found in the attached zipfile. See README.MD in the root directory for instruction on how to execute and build the program.

References

- [1] Foundation for Intelligent Physical Agents. Fipa dutch auction interaction protocol specification. <http://www.fipa.org/specs/fipa00032/XC00032F.pdf>, 2001. [Online; accessed 18-Nov-2016].
- [2] Telecom Italia. Java agent development framework. <http://jade.tilab.com/>, 2016. [Online; accessed 11-Nov-2016].