

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].
- *Constructing a theoretical model of the game mechanisms of dutch auctions and evaluating it*

TODO

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

		Agent i	
		D	C
Agent j	D	(4, 4)	(4, 1)
	C	(1, 4)	(1, 1)

Conclusions

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].