Introduction

This lab will introduce you to TradingPlatform (TP), our course’s software for simulating trading environments. Using TP, the course TAs have developed a few simple trading games (prediction markets, ad exchanges, wireless spectrum auctions), for which you will build trading agents, over the course of the semester. This document describes the architecture, key data structures, and core functionality of TP. You will want to refer to this document throughout this lab, and at other points later in the course.

Note: TP is written in Java. If you aren’t comfortable programming in Java, find a lab partner who is. Ask the TAs for help finding someone if necessary.

Agent Infrastructure

Once again, the TAs have programmed the games, and you will be programming the agents. To simplify your life, we provide you with a starting point for this lab, Lab3Agent. Copy the source code from…

Start by extending Lab3Agent and naming your new class after you and your partner in this lab. You should see that Lab3Agent implements the Agent interface, which only requires onTradeRequest(TradeRequest tradeRequest). This method will be called by the server whenever there is an opportunity to trade in an auction.

Second-Price Sealed Bid Auction

Fill out onTradeRequest to bid optimally in a second-price sealed bid auction. You will want to construct a SimpleBidBundle with SimpleSealed as the BundleType. When you are ready to bid, inform the server using market.bid(Bid bid).

Wait for everyone to finish, and we’ll compete!

First-Price Sealed Bid Auction

Now modify your implementation to bid (in equilibrium or not) in a first-price sealed bid auction with uniform valuations. We’ll do another competition!
Ascending, Open Outcry Auction

Wrap your code in an if clause, checking to see if the BundleType is SimpleSealed. Create a second if clause, checking for SimpleOutcry. You will fill in this clause with code that bids in an ascending, open outcry auction.

In a basic ascending, open outcry auction, the auctioneer announces a standing high bid, and then asks whether anyone is willing to pay one increment above this high bid. This process repeats until no one is willing to pay any more that the high bid. Once bidding stops, a winner is selected from among those bidders who were willing to pay the high bid. (Ties are broken randomly.)

Note that during each round of bidding, each agent will receive multiple identical TradeRequest messages before bidding closes (just in case some get lost!).

We'll now compete in this setting for a few rounds!

Ascending, Open Outcry Auction with Activity Rule

An activity rule defines what participating agents must do to remain active in an auction. For example, agents might be required to bid in all early rounds of ascending auction if they want to bid in later rounds. Modify your code (if necessary) to follow this activity rule. In other words, make sure your agent responds to every TradeRequest (unless, of course, it decides to drop out of the auction); otherwise, the server will reject all of its future Bids.

Modify your code to compete in this setting!

Ascending, Open Outcry with Activity Rule & No Jump Bidding

Ascending auctions without jump bidding define an increment, which guarantees that the high bid increases in a consistent fashion. In these auctions, your bids are replies to binary queries: i.e., “Yes, I want to bid above the standing high bid,” or “No, I don’t.”

Jump bidding is when you bid more than what is required in order to become the high bidder. Modify your agent so that it occasionally places jump bids. The activity rule is still in effect though, so be sure to respond to each TradeRequest (unless you are ready to drop out).

Modify your code to compete in this setting!

Unknown Valuations

Finally, let’s modify the game so that you are unsure of your valuation. Thus far, valuations have been drawn uniformly at random in the range [0, 100]. This time, your valuation is drawn from a normal distribution, whose mean is drawn uniformly at random in the range [0, 100], and whose variance is 10. Assuming this valuation structure, modify your agent to bid in an ascending, open outcry auction without an activity rule, but with jump bidding.

When you are ready, we will run one final competition as a Monte Carlo simulation.