Exercise 6 (Team/Ind Exercise, 30 Points Total) **Leigh Tesfatsion**
DUE: Tuesday, April 14, 11:00am **Econ 308, Spring 2009**

**** LATE ASSIGNMENTS WILL NOT BE ACCEPTED - NO EXCEPTIONS!**

Learning Vs. Zero-Intelligence Trading in Double-Auction Markets (JAVA VERSION — CONTINUATION OF EXERCISE 5)

Basic References:

1 * Ian Guffy and Nanpeng Yu, **DA-MAOS (Java/RepastJ): A Double Auction Multi-Agent Open-Source Market Experimentation Platform** www.econ.iastate.edu/tesfatsi/demos/DAMAOS/DAMAOSHomePage.htm

2 ** L. Tesfatsion, **"Learning Algorithms: Illustrative Examples"**, (Syllabus III.B) http://www.econ.iastate.edu/classes/econ308/tesfatsion/LearnAlgorithms.LT.pdf

3 ** L. Tesfatsion, **"Notes on Learning"**(Syllabus III.B) http://www.econ.iastate.edu/classes/econ308/tesfatsion/LEARNING.Econ308.pdf

4 ** L. Tesfatsion, **"Market Basics for Price-Setting Agents** (Syllbus II) http://www.econ.iastate.edu/classes/econ308/tesfatsion/MBasics.SlidesIncluded.pdf

5 ** L. Tesfatsion, **"Experimental Design: Basic Concepts and Terminology"** (Syllabus I.A) http://www.econ.iastate.edu/classes/econ308/tesfatsion/expdesign.pdf

EXERCISE OVERVIEW:

In the previously assigned Exercise 5 (Java Version), you developed/used a ZI Trading Demo to run illustrative ZI trading experiments either from scratch or using Ref.[1].

Building on this prior work, Exercise 6 asks you to explore whether the introduction of learning capabilities for some or all of the traders in your ZI trading environment has any substantial effects on market efficiency and market advantage.

Updated team/individal assignments for Exercise 6 are posted at the Exercise Information Site linked to the course homepage. Here is the direct link:

www.econ.iastate.edu/classes/econ308/tesfatsion/ExTeams.Econ308.2009.htm

EXERCISE DETAILS:

Part A (4 Points): Constructive Understanding of Individual Trader Learning Capabilities

For the purposes of this exercise, assume that any trader with learning capabilities uses reactive reinforcement learning in the form of **Modified Roth-Erev (MRE) Reinforcement Learning** (see Refs.[2-3]).

Using simple flow diagrams, mathematical equations, and careful verbal explanations (NOT CODE), describe carefully – step by step – how any trader (either a buyer or a seller) could be modeled as a trader that LEARNS OVER TIME how to choose its bid price (if a buyer) or its ask price (if a seller) by means of MRE reinforcement learning in the double auction environment you developed/used for Exercise 5.

Part B (8 Points): Incorporation of Learning Traders Into a ZI Double Auction

Carry out either ONE of the following two options.

OPTION 1: Develop Your Own Java Learner Trading Demo

- Let NS and NB denote, respectively, the total number of seller traders and the total number of buyer traders in the Java ZI Trading Demo you developed for Exercise 5.
- MODIFY the code of this Java ZI Trading Demo so that the user can designate any number of seller traders (from 0 to NS) and buyer traders (from 0 to NB) to be "learning traders."
- Using your analysis in Part A, MODIFY the code of this Java ZI Trading Demo so that any trader designated to be a "learning trader" choses its bid prices (if a buyer) or ask prices (if a seller) over time in accordance with MRE reinforcement learning.
- Be sure to include in your modified code a "stopping rule" that stops a simulation run after a certain number M of time steps (or "ticks").
- After debugging your modified code and ensuring it runs and compiles properly, **print out a copy of this modified code and turn it in as your answer for Part B of Exercise 6.**

OPTION 2: Verify the DA-MAOS Code for MRE Learning Traders

For the purposes of this exercise, the Market Type for the DA-MAOS demo (Ref.[1)] should be set as a "Discriminatory-Price K-Double Auction," meaning that the price for any matched buyer-seller pair is set between their bid price and ask price. As explained on the DA-MAOS demo homepage, the specification of "K" determines exactly where the price is set between the bid price and ask price.

- Let NS and NB denote, respectively, the total number of seller traders and the total number of buyer traders in the DA-MAOS demo (Ref. [1]).
- VERIFY whether or not the DA-MAOS demo [1] permits the user to designate any number of seller traders (from 0 to NS) and buyer traders (from 0 to NB) to be learning traders. If not, modify the code to permit this.
- Using your analysis in Part A, VERIFY whether or not the DA-MAOS demo [1] correctly codes for MRE reinforcement learning when "ModRothErev" is selected to be the learning method of a trader. If not, modify the code accordingly.
- VERIFY whether or not the DA-MAOS demo [1] has a "stopping rule" that permits a user to stop a simulation run after a certain number M of time steps (or "ticks"). If not, modify the code to permit this.
- Ensure that the (modified) DA-MAOS code runs and compiles properly.
- If you have NOT modified the code, your answer for Part B should simply be a short description of your verification efforts.
- If you have MODIFIED the code, your answer for Part B should consist of a short description of your verification efforts **together with print-outs of the modified parts of the code with modified parts clearly indicated.**

IMPORTANT NOTE: Hereafter, the demo resulting from either Option 1 or Option 2 in Part B will be referred to as the **Java Learner Trading Demo**.

Part C (4 Points): Develop an Experimental Design

Part C asks you to specify numerical parameter values for an experimental design to be conducted with your Java Learner Trading Demo from Part B.

As clarified in Part D, below, the purpose of this experimental design will be to experimentally test the effects of learning capabilities on Market Efficiency and Market Advantage. For Part C you are only asked to DEVELOP the experimental design, not to run any actual experiments.

The treatment factors to be considered in this experimental design are the exact numbers of seller traders and buyer traders who have learning capabilities. More precisely, the following four *Learning Treatments (LTs)* will be considered:

Treatment 1: ALL TRADERS are ZI-C traders (i.e., zero-intelligent traders constrained only by budget considerations).

Treatment 2: Only ONE SELLER has learning capabilities and ALL OTHER TRADERS are ZI-C traders.

Treatment 3: ALL SELLER TRADERS have learning capabilities while ALL BUYER TRADERS are ZI-C traders.

Treatment 4 ALL TRADERS (BOTH BUYERS AND SELLERS) have learning capabilities.

Complete this experimental design by specifying and reporting specific numerical values for all maintained parameter values, as follows:

- Select and report specific numerical values for the number NS of seller traders and the number NB of buyer traders to be maintained across all experimental runs.
- Select and report specific numerical values for all other required parameter settings in your Java Learner Trading Demo to be maintained across all experimental runs. Be sure to include among your specified parameter values an integer value $M \geq 1000$ for the maximum number of time steps (or "ticks") permitted for each simulation run. (The ideal is to have each run reach a point where all trade ceases.)
- Specify and report specific numerical values for all required parameter settings for the MRE reinforcement learning algorithm(s) used in Treatments 1 through 4. (You might want to have every learning trader have the same MRE parameter values, or you might want to permit different learning traders to learn using different MRE parameter values.)
- These specific numerical values should be selected to give you an economically interesting (non-trivial) trading environment.

Part D (8 Points): Carry out your experimental design.

Using your Java Learner Trading Demo from Part B, implement your experimental design in Part C as follows:

1. For each of the four treatments of your experimental design in Part C, conduct N experimental runs ($N \geq 10$) of your Java Learner Trading Demo using N distinct initial seed values for the pseudo-random number generator.
2. For each of the four treatments of your experimental design in Part C, and for each run $n = 1, ..., N$ conducted for this treatment, report:

   (a) the specific treatment number;

   (b) the pseudo-random number seed value (the identifier for the run);

(c) the Market Efficiency attained by the Mth time step for this run;

(d) the Market Advantage (positive or negative) attained by each buyer and seller by the Mth time step for this run.

3. For each of the four treatments of your experimental design in Part C, display the histogram for the Market Efficiency outcomes attained across the N runs in the Mth time step.

4. For each of the four treatments of your experimental design in Part C, report the sample mean and sample standard deviation ("N definition") for the Market Efficiency outcomes attained across the N runs in the Mth time step.

5. For each of the four treatments of your experimental design in Part C, report and display the Market Advantage outcomes attained by buyers and sellers across the N runs in the Mth time step in some interesting and informative manner of your own choosing.

Important Note: For Part D, be sure to use the precise measures for "Market efficiency" and "Market Advantage" introduced and explained in Ref. [4]. Also, be sure to use the precise measures for "sample mean," "sample standard deviation" (N definition), and "histogram" as provided in Ref. [5].

Part E (6 Points): Analyze Your Findings

As best you can, provide an explanation and interpretation for the experimental findings you reported in Part D. In particular, in what sense (if any) do these findings support the conjecture that the degree of Market Efficiency and Market Advantage attained for the double-auction market modeled in your demo does NOT depend on the learning capabilities specified for the participant traders? Explain carefully.