AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE
WYDZIAŁ ELEKTRONIKI, AUTOMATYKI, INFORMATYKI i ELEKTRONIKI

Symulacja dyskretna systemów złożonych
Stock Market Symulator

Autorzy:
Adam Klekowski
Przemysław Ziaja
Dominik Bober

Kraków 2020

Abstract

Następujący dokument jest dokumentacją projektu *Stock Market Symulator*, opicuje pracę wykonaną w formie zalicznienia przedmiotu *Symulacja dyskretna systemów złożonych* prowadzonego przez dr hab. inż. Jarosław Wąs, opiekunem jak również osobą oceniająco projekt jest mgr inż. Robert Lubaś

Spis teści

1 Wprowadzenie

1.1 Opis problemu

Celem projektu jest zasymulowanie zmian cen akcji na giełdzie papierów wartościowych. Chcemy z jak najlepszą precyzją określić wahania na rynku i móc przewidzieć jak inwestorzy zareagują na wprowadzenie niekonewncjonalnych strategii innych nabywców. Jak również zaobserwować jak inne czynniki ekonomiczne mogą ingerować w decyzje graczy na giełdzie. Planujemy zobserwować skuteczność różnych strategi kupowania oraz sprzedarzy na giełdzie, a także ocenić jaka liczba graczy preferujących daną metodę gry na rynku jest optymalna i nie wpływa na sprawność jej działania

Chemy zobserwować zmienność giełdy, założenia są więc takie, by program zapisywał kursy wybranej ilości form, a nsatępnie przedstawiał je w formie pozwalającej na analizę wahań rynku. Tym samym pozwalolić porównywać sytuacje różnych spółek, jak również zestwić wyniki z innymi czynnikami niezależnymi od działąń symulowanych firm. Jak również móc zestawyć wyniki naszych symulacji z rzeczywistymi danymi i móc określić skuteczność naszych algorytmów.

1.2 Potencjalne możliwości rozwiązania

Analizując nasze cele, planowaliśmy stworzyć system posiadający funkcjonalność giełdy, to znaczy potrafiący dopasować oferty kupna do odpowiednich ofert sprzedarzy. Najlepszym rozwiązaniem wydało nam się użycie wątków, tak by każdy agent symulujący inwestora mógł działać samodzielnie i niezależnie. Aby mieć pewność, że każdy kapitalista ma równe szanse, jak róznież abyśmy my mieli jak największą kontrolę nad wykonywanymi akcjami zdecydowaliśmy, że działenia rynku będzą zależne od cykli, imitujących jednostki czasu w których wszystkie informacje na temat giełby byłyby każdorazowo aktualizowane. Model powinien pozwalać na modyfikowanie ilości inwestorów, jak rónież zmiane ich strategi oraz ich parametrów, tak abysmy mogli przenalizować wiele wariantów i możliwości stanu rynku.

2 Przegląd literatury

W tym rozdziale chcielibyśmyy przedstawić prace, które udało nam się znaleść i przeanalizować, opisują podobną problematykę i oferują gotowe rozwiązania oraz implementacje algorytmów.

Jest wiele symulatów opisujących rynek, jednak ich autorzy niechętnie dzielą się ich specyfikacją, ponieważ są to narzędzia wykorzystywanie przez firmy by te mogły optywalizować swoje strategi i unikać błędów grając na giełdzie.

Są jednak prace, których twórcy chętnie działą się ich wynikami działania oraz spostrzereniami. Symulatorem, który był tym do którego działania chcielibyśmy dązyć tworząc nasz program jest *Stockyard* przedstawiony w 2017 roku. Jest to wieloagentowy i wilowątkowy program wykorzystujący rzeczywiście funkcjonujące algorytmy oraz przetwarzający istniejące już dane, takie jak historie kursów i transakcji. Alogrytm opiera się na rozdzielniu symulacji na 3 różne kategorie pracy, są to *Traders/Subsribers* imitujących inwestorów na rynku, którzy wysyłają polecania kupna lub sprzedarzy, *Exchange Agents* którch zadaniem jest wykonywanie tranzacji, ,czyli dopasowywanie BID(polecenie kupna) do ASK (polecenie sprzedarzy) oraz *Kernel* kontorlujący działanie całego programu oraz będący pomostem informacyjnym pomędzy wszystkimi agentami.

Inną pracą który oferującą obiecujące wyniki jest *Bristol Stock Exchange(BSE)* stworzony w 2012 roku. Napisany w Pytnonie 2.8, program wykorzystuje Limit Order Book(LOB). Algorytmy wykorzystane przy implementacji używają historycznch danych funkcjonowania rynku, dokładniej analizując przyszłe działanai kapitałowców. System wykorzystuje listy specjalnych struktur BID'ów i ASK'ów każdego z podrynków, by jak najbardziej zoptymalizować działanie tranzakcji. Model pozawala na nieograniczanie iplementowanie i dodawanie nowych agentów posiadającyhc odmienne strategie oraz definiujące ich parametry. Algorytm jest w urzyciu praktycznie od swojego powstania i do dzisiaj jest wyokorzystywany do przewidywania akcji na rynku przez wiele firm.

Przebadalismy również inneg prace takie jak Agent-based model stowrzony przez E. Panayi, M. Hermana i A. Wetherilt, który wykorzystuje dane z *Chi-X ex-*

change. Strategię G. C. Silaghi i V Robu która operuje na historycznych danych z *NASDAQ*. Ablo wciąrz wspieranego symulatora *OUCH*, który jest wykorzystywany również przy implementacji wyrzej wymienionego *Stockyard*.

3 Proponowany model

3.1 Cele modelu

Celem, który chcieliśmy osiągnąc jest jak najwierniesze oddanie natury giełdy. Oczekujemy, że nasz model, w przypadku gdy otrzymuje rzeczywiste dane był w stanie reagować naturalnie i adekwatnie do rzeczywistości. Tak by takie przyszłe wydarzenia jak tendencje wzrostowe czy spadki na rynku, mogły być przez nasz symulator przewidziane i odnotowane.

3.2 Założenia algorytmów

Nasz model zkłada poniższe założenia:

1. Giełda funkcjonuje cyklicznie - co pewien wycinek czasu wykonywane są wszystkie transakcje, odtotowywane są kursy i wysłane są aktualne dane do inwestorów.
2. Przewidujemy zachowania niewielkiej liczby firm (2–3).
3. Każdy trader ma stale aktualne informacje, a to czy podejmie decyzje w danej iteracji zależy od jego parametrów i strategii.
4. Dzielimy kapitłowców na kilka kategorii - niektorzy agenci wykorzystują specjalne algorytmy i stosują nauczanie maszynowe, natomiast inne posiadają proste instrukcje, niewymagające obszernej analizy wcześniejszych kursów na giełdzie, tak by móc zasymulować różne typy osób na grających na giełdzie.
5. Na podstawie wszystkich dokonanych tranzakcji w trwającej iteracji wyliczny jest kurs akcji i odnotowywyany w rejesrze zamówień.
6. Każdy trader jest oddzielnym wątkiem.
7. Istnieje proces głowny *Kernel*, który wykonuję główną oś programu i czówa nad poprawnym funkcjonowaniem agentów.

3.3 Diagramy cykli działań

Diagram opisujący działanie programu uwzględniający działania Kernela oraz traderów:

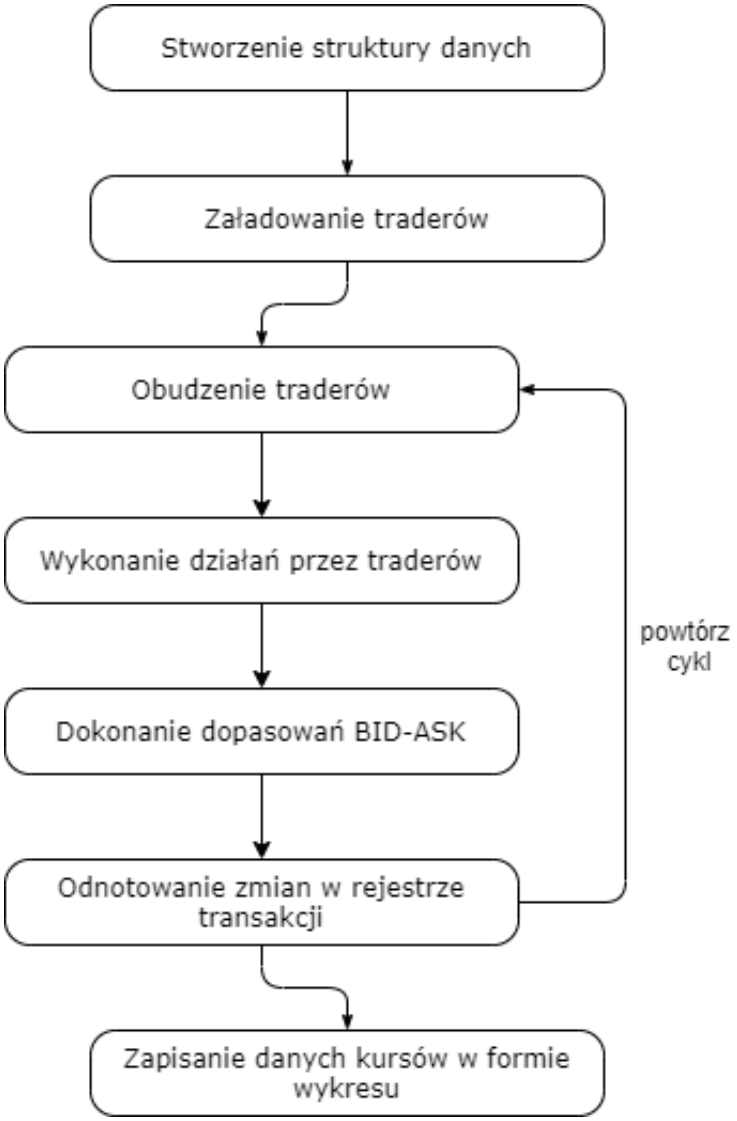

Figure 1: Diagram działania symulatora

3.4 Stosowane podejście w nawiązaniu do opisu literaturowego.

Po wnikliwej analizie literatury, zdecydowaliśmy, że nasz symulator będzie implementować wiele rodziajów agentów, wykorzystujących różne typy strategii, tak jak to jest to opisane w większości wcześniej wymienionych pozycji. Agenci, którzy wykorzystują sztuczną inteligencję, uczą się na rzeczywistych danych udostępnionych przez giełdy, jest to najczęściej stosowana strategia, którą stosuję większość autorów prac, które udało nam się rozpatrzeć. Wproawdziliśmy pewną modyfikację, aby uniknąć podobnych przewidywań każdego agenta podzieliliśmy zbiór danych giełdowych na mniejsze elementy, aby każdy agent uczył się na innych danych. Kernel naszego programu wykorzystuje listy struktur w trakcie dopasowywania BID'ów i ASK'ów, podobne rozwiązanie stosowane jest w symulatorze BSE.

3.5 Diagram przypadków użycia

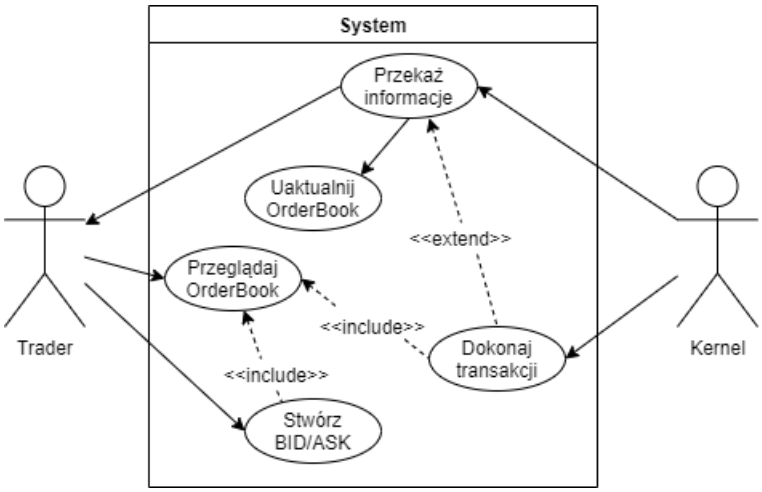

Figure 2: Diagram przypadków użycia

4 Symulacja zjawiska - implementacja

4.1 Wybór technologii

Zdecydowaliśmy się napisać nas program w języku Python, ponieważ jest to język, który dobrze spisuje się wtego typu symulacjach. Możemy dość swobodnie manipulować danymi wejściowymi a także zmieniać parametry i funkcjonalności już zaimplementowanych klas. Mogliśmy w krótkim czasie i przy niewielkim nakładzie storzyć działający prototyp i testować jego działanie. Pozwoliło nam to abyśmy od samego początku tworzenia projektu testować i sprawdzać nasze koncepty. W łatwy sposób mogliśmy również modyfikować i a także nanosić korekty do już napisanego kodu. Naszym założeniem było by program był wielo-agentowy oraz wielo-wątkowy, dlatego właśnie Python wydał nam sie najlepszym wyborem, posiada on biblioteki do tworzenia i obsługi wątków na takim poziomie, który nas satysfakcjonował. Jednak najważniejszym i decydującym czynnikiem była implementacja sztucznej inteligencji.

Do tego celu wykorzystaliśmy bibliotekę Tensorflow w wersji 2. Jest w znacznym stopniu uproszczona w porównaniu do swojej poprzedniczki. Ponadto posiada akcjeleracje GPU i jest wydawana przez Google, który także udostępnia platformę Google Colab, na której prowadziliśmy obliczenia. Do tworzenia predykcji kursów giełdowych akcji wykorzystaliśmy sieć opartą o komórki LSTM która pobiera dane o 3 poprzednich kursach ceny akcji oraz baryłki ropy naftowej. Ze względu na wysoki koszt obliczeniowy sieć nie jest głęboka.

4.2 Diagramy statyczne UML

W poniższym diagramie nie uwzględniliśmy klas pochodnych, w rzeczywistym modelu istnieją klasy dziedziczące po `OrderBook` i `Trader`, rozszerzają one funkcjonalności swoich rodziców i pozwalają na wprowadzenie bardziej złożonych algorytmów.

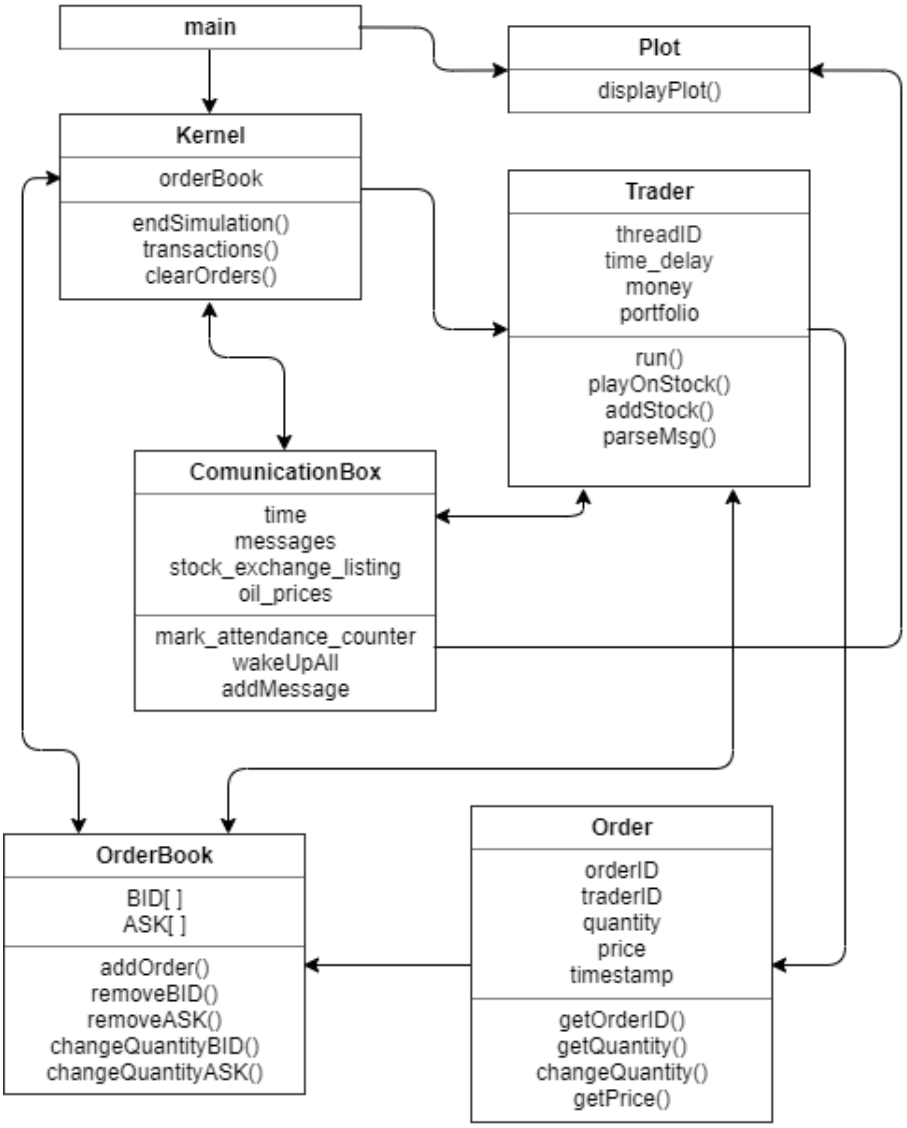

Figure 3: Diagram statyczny przedstawiający zaimplementowane klasy

4.3 Załączniki

Wszystkie pliki w wersji wykonywalnej można znaleść w naszym repozytorium pod tym adresem:

https://github.com/AdamKlekowski/stock-market-simulator.git

5 Wyniki symulacji

5.1 Uzyskane wyniki ilościowe i jakościowe

Po przeprowadzeniu w pętli 175 różnych symulacji dla różnych punktów początkowych oraz różnych stosunkach rodzajów traderów uzyskaliśmy dane do analizy statystycznej. Dla każdej symulacji nie uzyskaliśmy wartości znacznie odbiegających od rzeczywistych danych.

5.2 Analiza danych

Otrzymane dane poddaliśmy analizie używając 3 miar:

- korelacji
- średnigo błądu bezwzględnego
- średnieggo błądu kwadratowego

Analiza korelacji nie pokazała dużych zależności pomiędzy otrzymanymi danymi, a rzeczywistymi. Dla skrejnie dużych lub małych ilości inteligentnych traderów otrzymaliśmy histogramy o bardzo nieregularnym kształcie, natomiast w średnich zakresach histogram korelacji od ilości inteligentnych traderów był płaski. Dzięki analizie tej miary udało nam się ustalić, że duża ilość inteligentnych traderów wypłaszcza wykres, natomiast duża ilość traderów losowych dodaje zakłócenia przypominające te rzeczywiste.

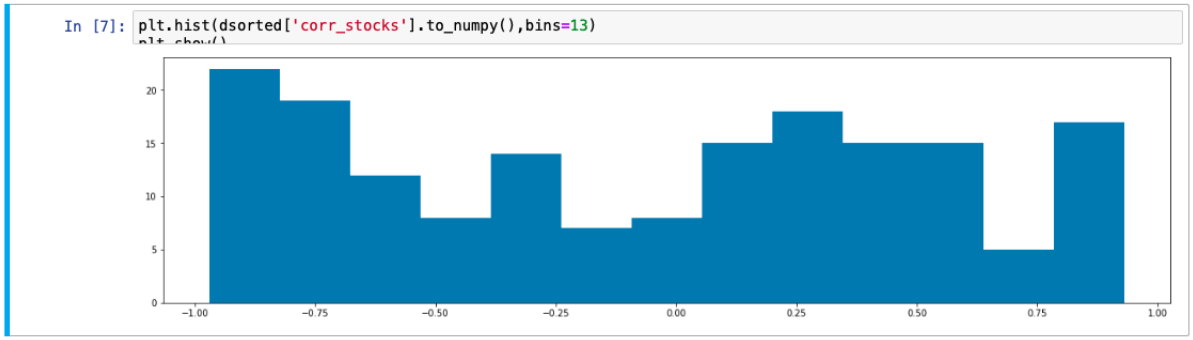

Figure 4: Histogram korelacji notowań rzeczywistych i symulowanych

Powyższa miara okazała się jednak niewystarczająca. Analiza średniego błędu bezwzględnego pokazała niewielkie średnie odchylnie od danych rzeczywistych. Miara ta jest bardzo łatwa do interpretacji i w większości przypadków po analizie histogramu tej miary dla różnych symulacji błąd nie był większy niż 6 dolarów.

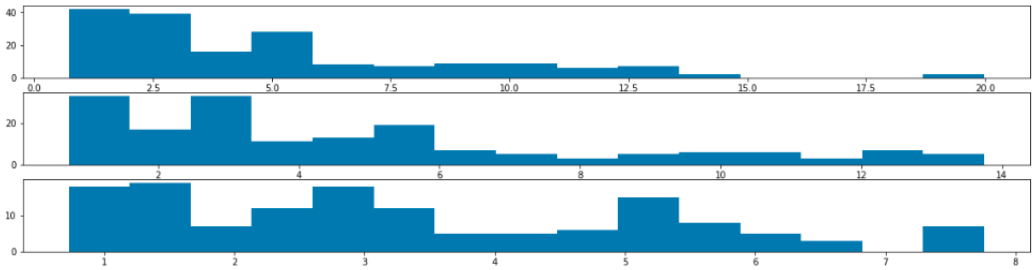

Figure 5: Histogramy wartości średniego błędu absolutnego (pierwszy od góry prezentuje całe dane, kolejne zawężają przedział)

Dodatkowo analiza średniego błędu kwadratowego również pokazała, że duża ilość symulacji posiada błąd o wartości nie większej niż 20 dolarów. Jest to świetna miara do porównania modeli pod względem dużych odchyleń jednak jej wartość jest trudna do interpretacji.

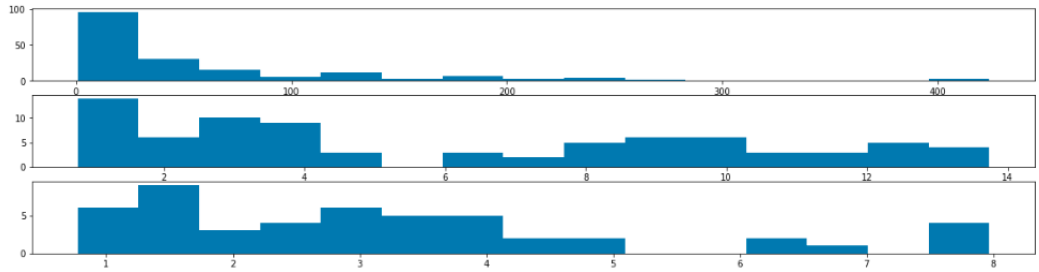

Figure 6: Histogramy wartości średniego błędu kwadratowego (pierwszy od góry prezentuje całe dane, kolejne zawężają przedział)

Z powyższych miar udało się także uzyskać zależność pomiędzy kalibracją modelu (wartości części współczynników w traderach były ustalone ręcznie) a ceną

ropy. W zbiorze symulacji wyraźnie widać wzrost obu błędów w chwilii spadku ceny ropy i ustabilizowania się niej na niższym poziomie.

5.3 Porównanie wyników z danymi rzeczywistymi

Symulacje z 1. części danych (przed spadkiem ropy) są bardzo podobne do danych rzeczywistych. Dodatkowym atutem symualcji jest to, że tylko w kilku przypadkach przewidziała zawyżoną cenę akcji na końcu symulacji, co może być przydatną pomocą przy inwestowaniu. Jednakże każdy okres w czasie wymaga skalibrowania względem cen ropy, na których model będzie przewidywał.

6 Wnioski

6.1 Wnioski

Zastosowaliśmy różne metody podejścia do problemu symulacji, przygotowane rozwiązania w części odwzorowują stan rzeczywisty zachowania giełdy papierów wartościowych. Głowne założenia projektu zostały osiągnęte, nie mniej jednak jakość wyników można poprawić wprowadzjąc modyfikacje wymienione w punkcie future works.

6.2 Wyzwania i trudności

Największym wyzwaniem było opracowanie takich strategi i założeń rynku by te jak najdokładniej owzorowywały zmienność giełdy i decyzje podejmowane przez inwestorów. Naszym założeniem od samego początku projektu było to by nasze obseracje można było bez trudności porównać z innymi rzeczywistymi danymi i móc stwierdzić, że nasz rynek funkcjonował w sposób naturalny i jak najlepiej odwzorowywał zachowania na jego prawdziwym odpowiedniku. Możemy stwierdzić że ten cel został przez nas osiągnięty i analizując wykresy, które drukuje symulator, można o nich powiedzieć, że wyglądają rzetelnie i mogą zostać pomylone z już istniejącymi.

6.3 Future Works

W dalszych etapach rozwoju należałoby skupić się na rozbudowaniu złożoności traderów i rozwinięciu strategii przez nich stosowanych. Aby jeszcze bardziej odwzorować naturę rynku, odpowiedniym byłoby poszerzenie i zwiększenie złożoności czynników neizwiązanych z częścią rynku, ale wpływającyhc na jego zachowanie. Dobrym kierunkiem byłoby również poszerzenie bazy agentów o posiadających inne metody postępowania oraz bazujących na innych danych wejściowych.

Bibliografia

[1] Efstathios Panayi, Mark Harman, Anne Wetherilt *Agent-based modelling of stock markets using existing order book data.*

[2] Gheorghe Cosmin Silaghi, Valentin Robu *An agent strategy for automated stock market trading combining price and order book information.*

[3] Dave Cliff *BSE: A MINIMAL SIMULATION OF A LIMIT-ORDER-BOOK STOCK EXCHANGE.*

[4] The NASDAQ Group *O*U*C*H Version 4.2.*

[5] Jianling Wang, Vivek George, Tucker Balch, Maria Hybinette *STOCK-YARD: A DISCRETE EVENT-BASED STOCK MARKET EXCHANGE SIMULATOR.*