Computer Engineering and Intelligent Systems
ISSN 2222-1719 (Paper) ISSN 2222-2863 (Online)
Vol.6, No.5, 2015

A Generalized Trust Calculation Algorithm

Hariyali Vipinchandra Adhyaru
Student
Department Of Computer Engineering, Vadodara Institute Of Engineering and Research, Kotambi
E-Mail: hariyali.ce@gmail.com

Alpa Girishbhai Patel
Student
Department Of Computer Engineering, Vadodara Institute Of Engineering and Research, Kotambi
E-Mail: alpapatel2121@yahoo.co.in

Atul Saurabh
Assistant Professor
Department of Computer Science and Engineering, Babaria Institute of Technology BITS edu campus, Varnama
E-Mail: atul.saurabh@gmail.com

Abstract
Abstract the current world is a digital world. And certainly we are living simultaneously with digital equipment. The digital computer is no exception. Human life, now a days, largely depends upon computer systems. We use computers for complex mathematical calculations, gaming, day to day account keeping, train tickets reservation, flight ticket reservation, war weapon launching, and space science. The list is keeps on going. The computer based systems are invented or discovered to make human life easier. But the journey is not so easy. If any system starts malfunctioning or behaving abnormally then it starts harming to the human race. As for example: If a war weapon controlling system is mal-programmed then it may possible that weapon may be launched if there is no bellicose situation or may be placed on different trajectory. Another example we may suggest from internet hacking. The confidential data may be hacked and can be used for malicious purpose. Both the example suggests one single point. Even though we use computer based system to make our life easier, it is not advisory to trust on the system without any examination. The trust is not a functionality of the system, but is a very important attribute. It is considered as a soft security. The more trusty a system is, the more reliable it will be. And it will be safer to use the system. The calculation of trust is not depending on one parameter. Various parameters need to take into accounts to calculate the trust. In this article we are proposing an algorithm for achieving the same. This algorithm is based on two concrete mathematical tools: The Markov Chain and The Baye's Theorem. The Markov chain or model is used to calculate the trust based on some previous trust factors. The Baye's theorem is used to find the probability of finding the truthfulness of the data.
Keywords: Multi-Agent System, JADE,Markove Chain, Baye's Theorm, Trust Calculation

1. Introduction

A multi-agent system is a distributed system where agents are coordinating and cooperating with each other to achieve some unified goal. Agents are intelligent entity which provides backbone of this kind of system. Normally agent is defined as a software entity which has a set of behaviors activated at different time frame to handle complicated conditions.

Agent plays an important role in a system where smartness is required for achieving multi-threading and multi-programming. Each instance of agent has its own kind of behavior and area of specialization. If it requires handling a situation which does not fall into corresponding area, agent may communicate using standard protocols available to solve the problem efficiently and effectively. Let us take a book trading example. There may be some agents working like a seller. The buyer tries to purchase a book from a seller which possibly not available with that one. If the system is really smart enough, it can figure out, automatically, in background, to which agent the book is available. Both the seller agents may negotiate the price and final price should be sent to the buyer. This will save the time for buyer to search the same book again over the system. A less smart system can at least prepare a list of seller who is having that book within the price restriction imposed by the buyer.

Multi-agent system works well only when the coordinating agents works well. If any agent is ill programmed then the entire system suffers from serious drawbacks. That kind of agents should not be allowed into the system. To work together, agents must have faith in each other. In more specific way, if agents are trustworthy the system will be stable and client of the system is happy with the performance. Being trustworthy means an agent has a firm belief on other agent in the system terms of job completion on certain constraint imposed on the job. If this belief is violated by an agent then other agent in the system should be notified immediately.

Multi-agent system consists of more than one agent and every agent is working independently. Whenever an agent needs any coordination from another agent, it communicates using passing message. When notified, the communicator comes to know the status of the work. The work assigned to be done, only single communication is not required. Rather a series of communication might be possible with more than one agent. Hence multiple agents are responsible for one single job. In this scenario it will be hard to figure out any discrepancies, if arisen, from any ill programmed agent which leads the job to failure. This discussion simply imposes a requirement of a centralized monitoring system through which all the agents in the system should be monitored. This centralized system tests each agent, either by direct communication or by some kind of feedback system or both, and establish trust factor which simply signifies the rank of agents in system. Higher the rank, the more reliable will be the agent.

This paper presents an elegant algorithm to compute trust factor of each agent in the system. Also this paper deals with a procedure to compute trust factor.

2. Introduction to Markov Chains and Bayes' Theorem

2.1 Probability

Probability is a branch of mathematics that deals with calculating the likelihood of a given event's occurrence, which is expressed as a number between 0 and 1.

Mathematically probability can be expressed as:

Properties of probability= Number of events can occur/ Total no of possible outcome

1. The sum of probabilities of an event and its complementary is
P(A)+P(A')=1

2. The probability of impossible event is zero.
P({Ø})=0

3. The probability of union of two events is the sum of their probabilities minus their interaction.
P(A\B)=P(A)+P(B)-P(A\ B)

4. If an event is subset of another events then its probability is less than or equal to it.
If A<B then P(A) /P(B)

2.2 Bayes' Theorem

Reverend Thomas Bayes' proved most important theory in statistics: Let T denotes “theory” and D denote “Data”. Then probability of theorem being true, given that the data has been observed is -

P(T| D)=P(D|T)P(T)/P(D)

Where,

P(D)=P(D| T) P(T)+P(D|T')P(T')

Where, T' being the event that the theory is false.

2.3 Random Variable

A Random Variable is a function or mapping f: E ->R from event space to real number. In other words, a random variable is a way to associate an event with a number.

1. Let an experiment of tossing coin is made. The event space is {HT}
   - Let the function is defined as f(H)=3 and F(T)=2
   - So, the event of coming up H is associated with 3 and that of T is associated with 2. So, f:E->R

Is a random variable.

2.4 Stochastic Process

A Stochastic Process (SP) is nothing but a collection of random variables to express the evolution of any system. The evolution of any system must have a start point and final point. The start point is always known. But the final point depends upon various conditions that a system met during its evolution. Again, the system does not reach final point immediately after start point. There may have been many intermediate points. By introducing intermediate points, a system is allowed to take any random path to reach the final point. Also there may be more than one final point in which system can stay.

The Stochastic Process normally depicted as:

2.5 Markov Chain

As told earlier, Markov Chain is a Stochastic Process in which future point (either intermediate point or final point) depends only on the current intermediate point. Markov property can also be called as memory less property as the state of the system at future time t(n+1) is decided by the system state at the current time t_{n} and does not depend on the state at earlier time instance t1,t2,t3…,tn.

- In general term, the distribution of where I go next depends on where I am now not on where I have been.

Markov chains are the combinations of probabilities and matrix operations. Markov chains models a process that proceeds in steps (time, sequence, trial, etc.); like a series of probability trees. This model can be in one “state” in each step. When the next step occurs, the process can be in the same state or move to another state. These movements are defined by probabilities. One can always find the probabilities of being in any given state many steps into the process.

The markov chain can be demonstrated using a transition diagram. The state of the diagram represents the intermediate point of the stochastic process. The advancement from the one state to another depends only upon the current state and not on the previous state of the system.

The above transition diagram represents an evolution of the system where S1, S2, and S3 are states of the system. The arrow represents the transition from one state to another state. The weight of the transition arrow represents the chance or the probability for the transition.

If the system has initial configuration as:
That means the probability that the system is in state S_{1}

$$\begin{bmatrix} A \\ B \\ 1-(a+b) \end{bmatrix}$$

We can predict the future state of the system as:
This matrix multiplication represents the next configuration of the state. The row of resultant column matrix represents the next probability of the corresponding state.

3. Literature Survey and Related Work

3.1 On How Agents Make Friends: Mechanism for Trust Acquisition

Babak Esfandairi and Sanjay Chandrashekharan had referred simple mechanisms for trust acquisition and propogation. Authors had studied one to one trust acquisition mechanisms. Authors had given propagation model which is uses a directed graph methods to calculate trust for human agents. A problem with their work is that it does not make a distinction between distrust and lack of knowledge about trust.

3.2 Bayesian Network-Based Trust Model

Yao Wang and Julita Vassileva proposed Bayesian network based trust model. Authors had considered situation where frequent interactions are there between two agents. They have presented a flexible method for presenting differentiated trust.

3.3 FIRE - An Integrated Trust and Reputation Model for Open Multi-Agent Systems

T. Dong Huynh and Nicholas R. Jennings and Nigel R. Shadbolt had presented novel model named FIRE which is an integrated trust and reputation model for open multi-agent systems. Authors claimed that FIRE can be easily adapted to multi domains because of its modularized design and parameterized configuration.

3.4 Modeling Trust in Multi-Agent Systems

Eli Stickgold, Sam Mahoney, Jonathan Pfautz , Joseph Campolongo and Erik Thomsen had proposed graph based approach for estimating trust for each agent in multi agent systems. In order to achieve above task authors have used Katz Centrality Matric as a measure of trust. The authors claimed that the developed algorithm does not require any hard facts and calculates trust based on relative information.

3.5 SecuredTrust: A Dynamic Trust Computation Model for Secured Communication in Multi-Agent Systems
Anupam Das and M. Mahfuzul Islam presented a novel and dynamic trust computation model called SecuredTrust for evaluating agents in multi-agent environments. Here authors had analyzed parameters related to the evaluation of trust and then proposed comprehensive quantitative model for calculating trust. They also proposed load balancing algorithm based on analyzed parameters.

3.6 Trust Decision Making in Multi Agent System
Chris Burnett, Timothy J Norman and Katia Sycara proposed their system considering risk, uncertainty and high dynamicity. Authors had found a new approach to select trustworthy partner. They have given five delegation strategies for trust evaluation which are Simple Delegation, Delegation with Monitoring, Delegate without Monitoring, Delegate with Reputational Incentive and Abstain from Delegation.

4. Proposed Algorithm

After going through some research papers, we are proposing a new general algorithm for calculating trust for any particular system.

Algorithm 1: Direct trust Calculation Algorithm

T →The initial testing phase counter
$\mathbf{\Gamma_i}$ →The trust value of the system $\mathbf{\Omega_i}$
Θ [T] ← {**0, 0, 0, 0**} Initial output array
numop($\mathbf{\Lambda_i}$) →Assign a number to the the output Λ_i
ε→define a range for which failover can be accepted
ξ→threshold value to which failover can be avoided
(ω, Λ)→task and known output pair
π→Penalty for lying
ν →Reward for telling truth
AssignTask(**Ω, ω**) →Assign task ω to System **Ω**. This method returns the output
Y supplied by the system **Ω**

FOR i←1 to |**T**|
begin
Θ[i] ←AssignTask(Ω_i , ω_i)
end
count←0
FOR i ←1 to |**T**|
begin
IF |numop(Θ[i]) - numop(Λ_i)| ≥ε
count ++
end
IF count ≥ξ
π←count *π
$\Gamma_i \leftarrow \Gamma_i + (\|T\| - count) * \nu - \pi$
End

4.1 Calculating the Initial Trust Value
Initially the initiator will assign some work to the system with supplied weight and will also calculate the chance of failure. Like this, he will setup the initial trust value for each and every system (the testing stage). Let us suppose we have three suppliers S1, S2, S3.

Algorithm 2: Feedback Based Trust Calculation Algorithm

T →The initial testing phase counter
$\mathbf{\Gamma_i}$ →The trust value of the system $\mathbf{\Omega_i}$
Θ[T] ←{0,0,0,. . .,0} Initial output array
numop($\mathbf{\Lambda_i}$) →Assign a number to the the output Λ_i
ε→define a range for which failover can be accepted
ξ→threshold value to which failover can be avoided
(ω, Λ)→task and known output pair
π→Penalty for lieing

ν →Reward for telling truth
ReceiveFeedback (**Ω_i ,ω_i ,χ_i ,Λ_i ,Ω_j**) → receiving a feedback about a system
Ω_i from system Ω_j by assigning a task ω_i whose desired outcome is χ_i
and received output is Λ_i
S_{suff} → Sufficient number of individual evidence
Σ_{Si} ← ReceiveFeedback((**Ω_i ,ω_i ,χ_i ,Λ_i ,Ω_j**))

If $\|\Sigma_{Si}\| \geq S_{suff}$
begin
count ←0
for i ←0 to $\|\Sigma_{Si}\|$
If $\|numop(\Sigma_{Si} .\Lambda_i) - numop(\Sigma_{Si} .\chi_i)\| \geq \varepsilon$
count ++
If count $\geq \xi$
$\pi \leftarrow count * \pi$
$\sigma \leftarrow \Gamma_{\Omega i} * \pi$
$\Gamma_{\Omega i} \leftarrow \Gamma_{\Omega i} + (\|\Sigma_{Si}\| - count) * \nu - \sigma$
End
End

Now if at any point of time system S3 say complains that the work get fail due to some other reason like delivery system, then we will use Bayes' theorem to calculate the weight to truth. And the trust will be increased or decreased accordingly.

4.3 Future Calculation

Now if a buyer is using any system say S2 then the initiator can give the probabilistic view of the chance of failure using the Markov Chain as we have the previous data on which we can predict the future. Now if the buyer reports the failure result, we again calculate the truth weight of the buyer using Bayes Theorem and increase or decrease the trust factor accordingly.

4.4 Calculating the False Alarm

If any system reports that it is recovered from the previous illness, the initiator will first calculate Probability of Truth using Markov Chain. If it is above the threshold value, the initiator will put the system in testing stage. Otherwise the trust will go down to zero.

Algorithm 2: Future Trust Calculation

MarkovMatrix [2][2] →Initial matrix to calculate next trust value
CurrentTrust[2] →Current trust matrix
S_{suff}→Sufficient number of individual evidence
BuildMarkovMatrix(MarkovMatrix) →This function generates a markov matrix
IF $\|S_{suff}\|$ is large
BuildMarkovMatrix(MarkovMatrix)
CurrentTrust[1,2]=CurrentTrust[2] * MarkovMatrix[2][2]

Algorithm 2: Updating Trust Value On Claim

Ċ→Claim
∏→Output
AcceptClaim(Ω_i, Ω_j , ω, Λ , X) →The system Ω_i claims that Ω_j is responsible
for the output Λ of task ω of which desired output was X.
Γ→Trust value
π →Penalty for lying
υ →Reward for telling truth
numop(Λ)→Converting output to number
B(T/F) →The probability that the claim is true/false when incident has occured.
Ċ→AcceptClaim(Ωi, Ωj , ω, Λ , X)
∏ → assignTask(Ωj , ω)
IF | numop(X) - numop(Λ)| ≥ numop(∏)

$\Gamma_{Ċi\,\Omega i} \rightarrow \Gamma_{Ċi\,\Omega i} + (\upsilon * B(T))$
else
$\Gamma_{Ċi\,\Omega i} \rightarrow \Gamma_{Ċi\,\Omega i} - (\pi * B(F))$

5. Implementation

The algorithm is implemented on JADE framework. JADE is a platform providing Agent–Oriented programming (AOP) in JAVA. AOP essentially models an application as a collection of components called agents that are characterized by, among other things, autonomy, proactively and an ability to communicate. The architectural model of an agent-oriented application is peer to peer, as any agent is able to initiate communication with any other agent or be the subject of an incoming communication at any time.

A scenario of book trading is taken as an example. There are various agents participating in this simulation. The buyer agent (see Figure-1) is a general buyer which purchase book. There are sellers which are providing books. One agent called TrustCalculatorAgent is computing trust value of seller agents based on various parameters passed to it at its setup time. The fig:[Figure-2] shows the result of query for trust about a particular seller agent.

6. Result and Analysis

The Figure-3, Figure-4 and Figure-5 show the plotting of trust value calculated by TrustCalculator agent for various seller agents. The plot shows important result. As the price range increases the trust goes down. Also as the price varies agents are clustered into three main categories, the high, the mid and the low trusted agent. This calculation is based on price only. But there may be various parameter on which the trust value may go high or low.

7. Conclusion and Future Work

7.1 Conclusion

By understanding existing work related to trust evaluation, we found some drawbacks which are listed in part 3.1 above. To overcome these problems I propose a new algorithm for calculating trust of a particular system. My proposed algorithm is parameter independent as it uses weight of a parameter instead of parameter, so it makes my system dynamic. Typical mathematics used in proposed algorithm is: Probability approach, Bayes theorem and Markov chain model.

7.2 Future Work

Future work of this report is to implement Trust function, implementation of Markov chain model, implementation of Bayes theorem, and implementation of proposed algorithm. In the implementation of proposed algorithm, number of parameter and weight of those parameters will be taken as input and trust value is generated as output of the algorithm.

References

1. http://en.wikipedia.org/wiki/Multi-agent_system, last access on 27th April, 2015.
2.https://www.google.co.in/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&ved=0CCYQFjAA&url=http%3A%2F%2Fwww.lsi.upc.edu%2F~ia%2Fmluck%2F1_rm.ppt&ei=ZcJsUvaTNZHirAe_xIH4CA&usg=AFQjCNFEacH2dSQv5TZrPASwmId1YzMNKg&sig2=a2JnlGJAOr0grxstpWYpRg&bvm=bv.55123115,d.bmk, last access on 27th April, 2015.
3. http://en.wikipedia.org/wiki/Procedural_reasoning_system, last access on 1st January, 2015.
4. http://www.slideshare.net/easss2012/trust-and-reputation-in-multiagent-systems, last access on 27th October, 2015.
5. Babak Esfandiari, Sanjay Chandrasekharan, "On How Agents Make Friends: Mechanisms for Trust Acquisition", Department of Systems and Computer Engineering, Carleton University, Ottawa, Ontario, Canada.
6. Yao Wang, Julita Vassileva, "Bayesian network-based trust model", in Proceeding of IEEE/WIC International Conference on WebIntelligence (WI), Halifax, Canada, October 2003, pp. 372–378.
7. T. Dong Huynh and Nicholas R. Jennings and Nigel R. Shadbolt, "FIRE: An Integrated Trust and Reputation Model for Open Multi-Agent Systems" in Proceedings of the 7th International Workshop on Trust an Agent Societies, 2004, pp 65-74.
8. Eli Stickgold ,Sam Mahoney , Jonathan Pfautz , Joseph Campolongo , Erik Thomsen , "Modeling trust in Multi Agent System", in Preceeding of the 22ndAnnual Conference on Behavior Representation in Modeling and Simulation , Ottawa, Canada: BRIMS Society, 2013.
9. Anupam Das and M. Mahfuzul Islam, "Secured Trust: A Dynamic Trust Computation Model for Secured Communication in Multi – Agent Systems" IEEE Transactions on Dependable and Secure Computing vol.9,

no.2, year 2012.
10. Chris Burnett, Timothy J. Norman, Katia Sycara, “Trust Decision-Making in Multi-Agent Systems”, Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence , 2011.

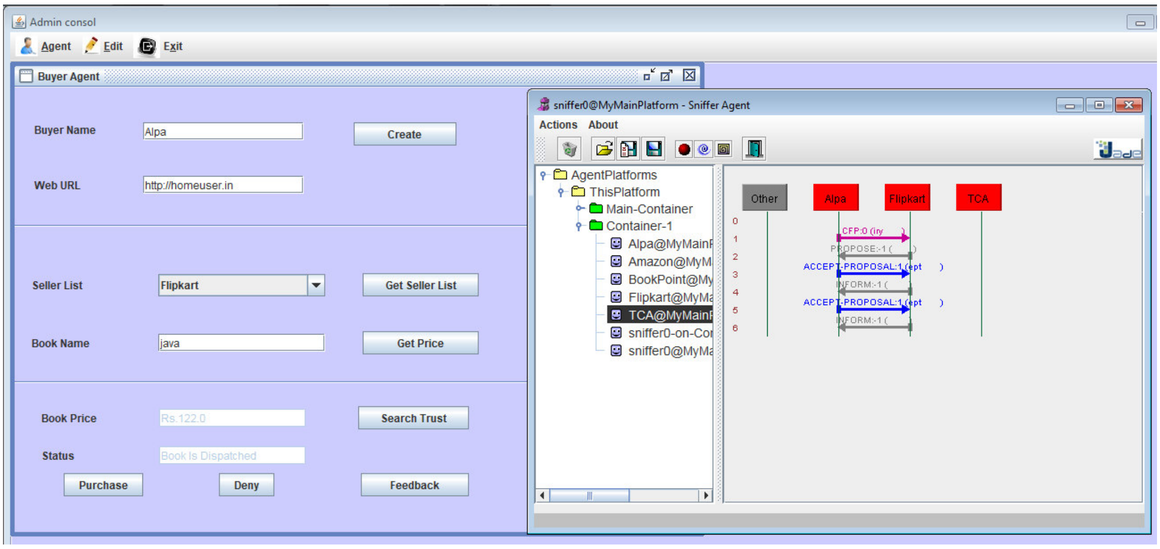

Figure 1. Buyer Agent Purchase A Book

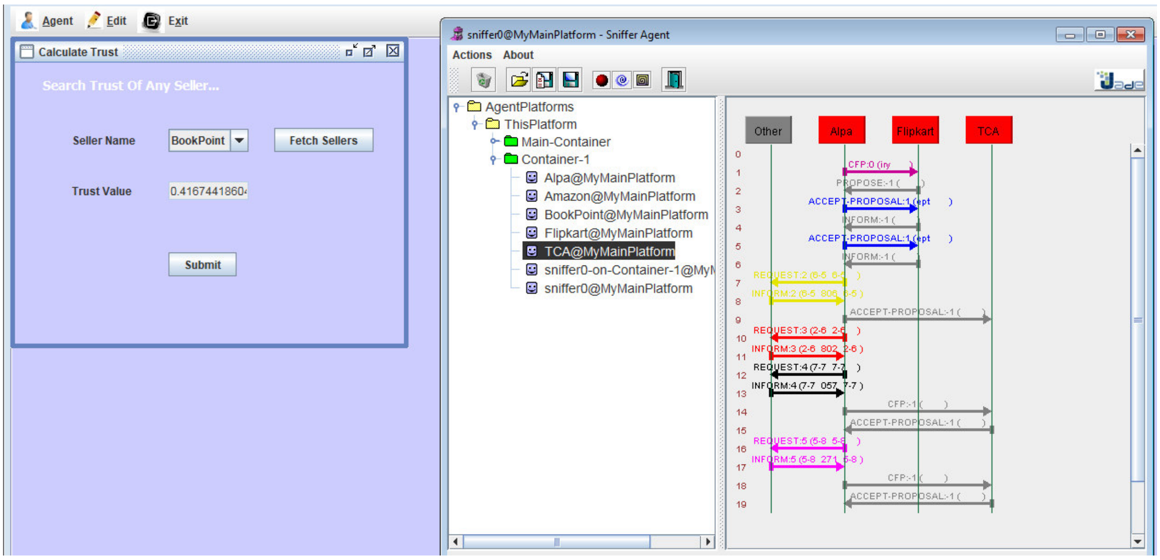

Figure 2 . Trust Calculation Agent In JADE Environment

www.iiste.org

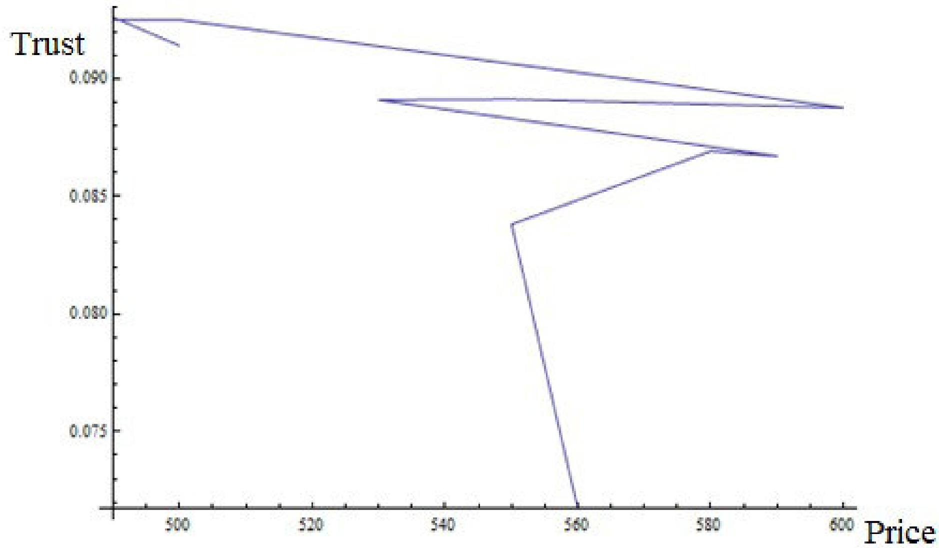

Figure 3. Best Agent Trust Value Graph

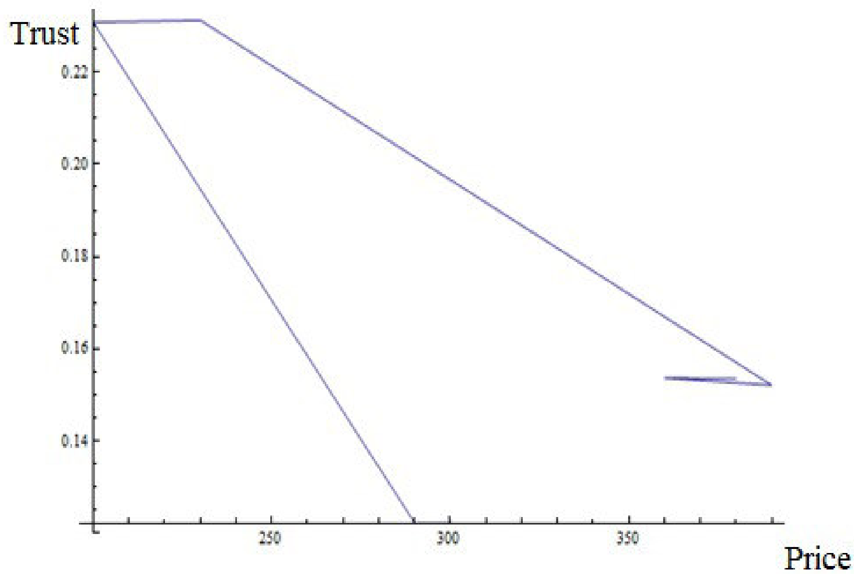

Figure 4. Moderate Agent Trust Value Graph

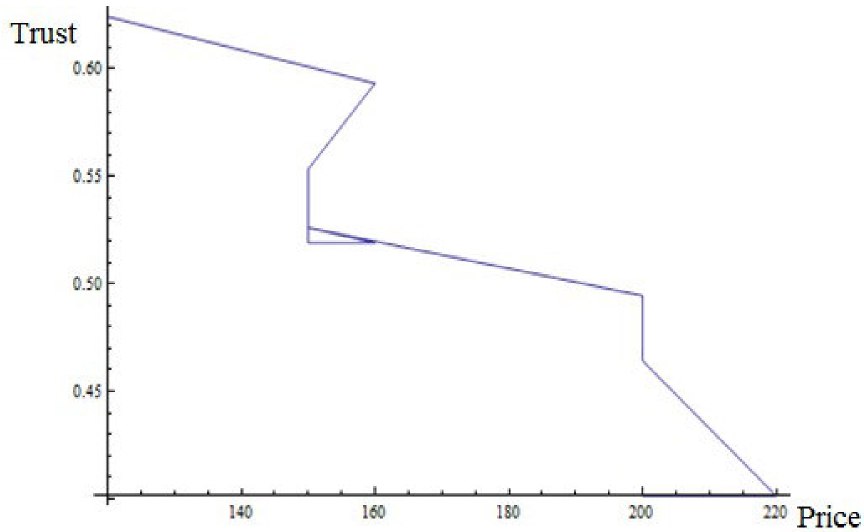

Figure 5. Worst Agent Trust Value Graph