Full-text Available Online at
www.ajol.info and
www.bioline.org.br/ja

Vol. 20 (4) 1127-1135

# Portfolio Allocation under the Vendor Managed Inventory: A Markov Decision Process

***[1]EZUGWU, VO; IGBINOSUN, LI**

*Department of Mathematics and Statistics*
*University of Uyo, Akwa-Ibom State, Nigeria*
*1. ezugwuvitus@gmail.com 2 .luckyigbinosun@uniuyo.edu.ng*

**ABSTRACT:** Markov decision processes have been applied in solving a wide range of optimization problems over the years. This study provides a review of Markov decision processes and investigates its suitability for solutions to portfolio allocation problems under vendor managed inventory in an uncertain market environment. The problem was formulated in the frame work of Markov decision process and a value iteration algorithm was implemented to obtain the expected reward and the optimal policy that maps an action to a given state. Two challenges were examined –the uncertainty about the value of the item which follows a stochastic model and the small state/action spaces that can be solved via value iteration. It was observed that the optimal policy is expected to always short the stock when in state 0 because of its large return. However, while the return is not as large as in state 0, the probability of staying in state 2 is high enough that the vendor should long the stock because he expects high reward for several periods. We also obtained the expected reward for each state every ten iterations using a discount factor of $\lambda = 0.95$. In spite of the small state/action spaces, the vendor is able to optimize its reward by the use of Markov decision process. ©JASEM

**http://dx.doi.org/10.4314/jasem.v20i4.29**

***Keywords:*** Portfolio Allocation, Vendor Managed Inventory, Markov Decision Process, Value Iteration, Expected Reward, Optimal Policy.

Decision making plays a very important role on individual, organizational, societal and governmental levels. In this study, the decision maker (vendor), after considering all surrounding circumstances, has to go through the mental process before an action is taken among several alternatives. The kind of decision taken by the vendor today affects its future either positively or negatively. The fundamental decision faced by the vendor is how to optimally allocate its funds at each decision epoch during a time horizon on an uncertain market environment in order to optimize its reward. From control point of view, the vendor as the controller must optimize its reward function at each decision epoch by selecting appropriate action(s) from its action space. The optimal policy that maps action to a given state was also studied. The main objective of the study is to apply Markov decision process to portfolio allocation problem under vendor managed inventory environment in order to obtain the expected reward for each decision and the optimal policy that maps an action to a given state.

Inventory management is very important in most companies as well as commercial sectors because it helps the company or the vendor to respond quickly to customers' demands, which is an important element in competitive markets. An inventory is a collection of people, equipment, and procedure that function to keep account of the quality of items in inventory and determine which item to purchase or what quantity to produce. This study considers portfolio allocation problem under vendor managed inventory system. Vendor Managed Inventory (VMI) is a partnership between a supplier and a customer where the supplying organization makes inventory replenishment decisions on behalf of the customer,(Chukwu and Echo,2009). Traditionally, investment is the current commitment of resources in order to achieve later benefits. These benefits are obtained under portfolio management which is a decision process of dividing the total investment funds among some major asset classes such as equities, bonds, goods etc (Haley, 2009). Portfolio allocation is how an investor allocates his funds among a set of investments to maximize return while simultaneously minimizing risk ( David, 2008). Portfolio allocation is also the investment of liquid capital to various trading opportunities like goods, stocks, foreign exchange and others. A portfolio is constructed with the aim of achieving a maximum expected return for a given risk and time horizon. Portfolio allocation problem is a problem which has generated a great deal of research since it was first formally defined by Harry Markowitz in 1952, where he used mean variance(MV) optimization model as a breakthrough achievement in modern portfolio theory ( David, 2008). However, there is a major drawback, the Markowitz model calculates the covariance between each pair of securities and because the covariance between any pair of investments must be calculated to run the Markowitz model, this results in a large number of calculations. The size of the covariance matrix coupled with the fact that

*Corresponding author email: ezugwuvitus@gmail.com*

Markowitz model is formulated as a quadratic program (much less efficient than linear program) means that the model becomes infeasible very quickly as the number of investments increases. Because of covariance problem, it turns out that Markowitz model has not been much used in practice since its publication in 1952. For this reason, other models have been developed.

Konno and Yamazaki, (1980), developed a model that uses mean absolute deviation (MAD) rather than the mean variance as a measure of risk. This model does not measure how pair of securities is related. This enables the problem to be formulated as a linear program. For this reason, the MAD model is much easier in computational sense. After this model, Markov Decision Process came into place. The problem that is inherent in any portfolio optimization model is the uncertainty in the forecasted marked data. Invariably, any potential investor will make predictions about future market when deciding how to allocate his funds, not doing so would be foolish. In this study, we used the optimization technique under uncertainty which is Markov decision process. A Markov decision process is a representation of dynamic program. An MDP is represented by the state, the decision set, which is made up of a finite set of allowable decisions, the transition probabilities, and the expected reward. This technique is limited by the fact that there can only be a finite number of elements in the decision set and the state space, as opposed to stochastic programming in which uncertainty is represented by the probability distribution(David, 2008).

Many related research works have been carried out. Dror and Ball (1987) considered the application of integrated inventory and transportation problem. They investigated the problem of distributing heating oil among customers using a fleet of vehicles. Their objective was to minimize the annual delivery stock-out costs using both deterministic and stochastic demands. The allocation of human and physical resources over time as a fundamental problem that is central to management science was carried out by Warren, et al (2003). They reviewed a mathematical model of dynamic resource allocation that is motivated by problems in transportation and logistics using an algorithm developed by Warren. They showed how problems in freight transportation can be solved through dynamic programming to select a policy that maximizes the expected reward over the time horizon. Transaction costs and resampling are two important issues that need great attention in every portfolio investment planning, Dror and Trudean , 1996; Christophette et al (2004) considered a risky asset whose instantaneous rate of return takes two different values and changes from one to the other one at random times which are neither known or directly observable. They studied the optimal strategy of traders who, in the presence of cost transaction, invest on this risky asset, or on a non-risky asset according to their belief on the current state of the instantaneous rate of return and finally applied dynamic programming. In Application of Markov Decision Process to a Simplified Model of Robot Fire Fighter studied by Kwame (2009), he provided a review of Markov decision process and investigated their suitability for the problem of designing autonomous intelligent agent for forest fire fighting. He formulated the problem in the frame work of Markov decision process and implemented a fast value iteration algorithm to obtain the optimal policy. Arseal (2009) studied the Graphic Processing Unit (GPU)-Bases Markov Decision Process. He used Markov decision process to provide a mathematical frame work for modeling decision making in situation where outcomes are partly random and partly under the control of the decision maker and finally applied value iteration to obtain the optimal policy. Md.Noor and John (2010) studied stochastic investment decision with dynamic programming. In their research, proper investment decision making is key to success for every investor in their efforts to keep pace with the competitive business environment. The mitigation of exposure to risk plays a vital role, since investors are now directly exposed to the uncertain decision environment. They opined that the expected reward on investment of a decision often carries high degree of uncertainty and their objective was to formulate a dynamic programming model for the investment incorporating the uncertainty in a probabilistic manner in order to find a policy that maximizes the expected gain. Kobbane et al (2012) discussed the approach of using MDPs for dynamically optimizing the network operations to fit the physical conditions. They observed that the MDP model allows a balanced design of different objectives, for example, minimizing energy consumption and maximizing sensing coverage. Mohammad, et al (2015) applied Markov decision process in wireless sensor network. They opined that wireless sensor networks (WSNs) operate as stochastic system because of randomness in the monitored environments. For more service time and low maintenance cost, WSNs require adaptive and robust methods to address data exchange, topology formulation, resource and power optimization, sensing coverage and object detection, and security challenges, Dimitrios, (2013) studied portfolio selection with multiple risky assets, linear transaction costs, and a risk measure in a multi-period

setting and formulated the multi-period portfolio selection problem as a dynamic program. To solve the problem, he constructed approximate dynamic programming (ADP) algorithms which included Conditional-Value-at-Risk (CVaR) as a measure of risk for different separable functional approximations of the value functions.

## MATERIALS AND METHODS

*Methodology*: In this section, the method applied is Markov Decision Process which an extension of Markov Chain. The Markov decision process (MDP) frame work developed to investigate a solution to a portfolio allocation problem is given by

$$r_i(e_{i,t}, \omega_{i,t}, a_{i,t}) = (\omega_{i,t} + a_{i,t})\hat{\alpha}_{i,t} \; - \; C(a_{i,t}) \quad \ldots\ldots\ldots\ldots (1)$$

The allocation has two challenges: (i) the uncertainty about the value of item that changes with the expected return and follows a stochastic model. (ii) The state/action space which is small and can be handled by value iteration method.

Definition of Notations

M =Number of states in the state space

$\hat{\alpha}_i$ =An expected return

S = Set of all possible combinations of economic states and weights

W =Set of weights

$r_t(S,a)$ = Reward accrued between time $t$ and $t+1$ for a given state and action

$e_{i,t}$ = The economic state of each item $i$ at time $t$

A = Set of all possible actions

$A_S$ = The set of actions available at each state

$S_t$ = The state of the process at time $t$

$a_t$ = Action taken at time $t$

$p_{ij}$ = The probability of moving from state $i$ at time $t$ to state $j$ at time $t+1$ for a Markov process

$p(S_t, a_t, S_{t+1})$ = The probability of transitioning from state $S_t$ at time $t$ to another state $S_{t+1}$ at time $t+1$ for a given action $a_t$ for a Markov decision process

$C(a_t)$ = The transaction cost function for a given action.

$\lambda$ = Discount factor

$c$ = The transaction cost constant

The objective is to maximize the expected reward of a portfolio of items over a finite time horizon, each with expected return. To find a policy that optimally chooses an action, we assume

We have a fixed capital and a fixed universe of items to deal with.

We have an expected return for each item which changes each time period following a well-defined Markov process ie $p[X_n = j / X_{n-1} = i] = p_{ij}$ (2)

Here, we define each item as following a Markov model that transitions from period to period and each state of the Markov model has an expected return $\hat{\alpha}_i$ associated with it.

*States*: The state is the economic state of each item. The economic state in the study is the economic value (price) of the item. The set of $M$ states is defined as $E = \{1,2,...,M\}$.

*Transition*: Each time period, the item transits to either the same state or a new state. The transition probability matrix is given as,

**Table 1**: The transition probability matrix for the Markov model

$$p = \begin{array}{c} \\ 1 \\ 2 \\ \cdot \\ \cdot \\ \cdot \\ M \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & . & . & , \ldots & M \end{array} \\ \begin{bmatrix} p_{11} & p_{12} & \cdot & \cdot & \cdot & p_{1M} \\ p_{21} & p_{22} & \cdot & \cdot & \cdot & p_{2M} \\ \cdot & & & & & \cdot \\ \cdot & & & & & \cdot \\ \cdot & & & & & \cdot \\ p_{M1} & p_{M2} & , & \cdot & \cdot & p_{MM} \end{bmatrix} \end{array}$$

In the study, the states of the Markov model are $0,1,$ and $2$. That is $M = 3$,

Where: $0 =$ Bear Market

$1 =$ Recession

$2 =$ Bull Market

*Markov Decision Process*; This comprises of four major elements; states, actions, Markov transition probabilities and reward. In most cases, a fifth element, decision epoch is added to the model. At each time step, the process is in state $s_t$ and the decision maker (vendor) may choose any action $a_t$

available in state $s_t$. The process responds at the next time step by randomly moving into a new state $s_{t+1}$ and giving the decision maker a corresponding reward $r_t(S,a)$. Markov Decision Process is an extension of Markov chain; the difference is the addition of action and reward. An MDP takes the Markov state for each item with associated expected return and assigns weight describing how much of the capital to invest in that item. Each state of the MDP contains the economic weight of the item and current weight invested. The actions allow us to modify the weights of the item from period to period. The rewards also specify how much expected return the item generates in its current state.

*Decision Epoch*: We refer as decision epoch the set of times at which decisions are made, ie $t = 1,2,3,...,T$. At each decision epoch, the vendor observes the state $s_t$, chooses an action $a_t$ and receives a reward $r_t(S,a)$ which is a function of state and action of that decision epoch.

State Space: The state space $S$ of the MDP consists of all possible combination of economic state $e_{i,t}$, and weights $\omega_{i,t}$ for all the items. There are $W$ discrete set of weights. If there are $N$ item at time $t$, the state $s_t$ of the MDP is the economic state of each item $i$ at time $t$, $e_{i,t}$ and the proportion of money $\omega_{i,t}$ invested in the item. That is $s_t = \{(e_{1,t}, e_{2,t}, ..., e_{N,t}), (\omega_{i,t}, \omega_{2,t}, ..., \omega_{N,t})\}$. Since there are $N$ items, each item takes $M$ economic states and have $W$ weights assigned to the item, then the cardinality of the state space for one item is $MW$ and the cardinality of the state space for $N$ items is $N^{MW}$, so $S = \bigcup_{t \in T} s_t$.

*Action Space*: The action space $A$ consists of all possible actions the vendor can take. At each decision epoch, the vendor takes an action. Since we have $N$ items and $W$ different weights, the action space at any given time for one item is $W$ and $N^W$ for $N$ items. In each state $s_t \in S$, the decision maker, based on what he observes in the state, chooses an action $a_t$ from the set of all allowable action in that state $A_s$, then $A = \bigcup_{s \in S} A_s$. The action space for this study is $A = \{-2, -1, 0, 1, 2\}$. These actions are based on the amount invested, Md. Noor et al (2010), where

1. $-2 =$ Invest the capital on a risk free item
2. $-1 =$ Short the stock
3. $0 =$ Invest nothing in the stock and everything in cash
4. $1 =$ Long stock
5. $2 =$ Invest the capital and any excess amount above the working capital in the item.

At each decision epoch, the MDP transits to the same state or a new state depending on the transitions of the Markov model for each item and action taken.

*Reward*: When an action $a_t \in A_s$ is taken by the vendor in state $s_t$ at decision epoch $t$, the vendor receives a reward $r_t(S,a)$ which is a function of the state and action taken. Transaction costs for this study includes fixed cost and variable costs such as stock out cost, holding cost and transportation cost. $C(a_t)$ is the transaction cost function based on the action $a_t$ he takes and is defined as

$$C(a_t) = c\sum_{i=1}^{N} |a_{i,t}|; 0 \geq c \leq 1 \quad (3)$$

Therefore, the reward for item $i$ currently in state $(e_{i,t}, \omega_{i,t})$ when an action is taken is defined as

$$r_t = (e_{i,t}, \varpi_{i,t}, a_{i,t}) = (\omega_{i,t} + a_{i,t})\hat{\alpha}_i - C(a_{i,t}).$$

*Policy*: A policy is a function that maps an action to every state. A policy is optimal if it generates at least as much as total reward as all other possible policies. Using MDP, the value iteration algorithm method suggested by Puterman (1994) was used to find a stationary $\varepsilon-$optimal policy. A policy is optimal if the decision rule employed is invariant over time.
The algorithm is as follows

1. Select $v^0$, specify $\varepsilon \succ 0$, and set $n = 0$
2. For each $s \in S$, compute $v^{n+1}(s)$ by

$$\upsilon^{n+1}(s)=\max_{a\in A_s}\left\{r(S,a)+\sum_{j\in S}\lambda P[j/s,a]\upsilon^{n}(j)\right\}$$ ..... (4)

3. If $\left|\upsilon^{n+1}-\upsilon^{n}\right|\prec\frac{\varepsilon(1-\lambda)}{2\lambda}$

   Go to step $4$, otherwise increment $n$ by one and return to step $2$

4. For each $s\in S$, choose

$$d_{\varepsilon}(s)\in\arg\max_{a\in A_s}\left\{r(s,a)+\sum_{j}\lambda P[j/s,a]\upsilon^{n+1}(j)\right\}$$

and then stop.

$\upsilon^{n+1}$ is found by iterating equation (4) until some convergence measure is obtained. That is until the difference between $\upsilon^{n+1}$ and $\upsilon^{n}$ becomes smaller than some threshold. The fundamental idea used in this approach is to compute the value of each state and then use the value to select an optimal action in each state.

*A Case Study*: Here, we apply value iteration algorithm on the developed MDP model to obtain the expected reward for each state and also find the $\varepsilon-$ optimal policy for the MDP. We consider one item and allow a few weight

We consider a vendor located in Ogbete main market, Enugu, Enugu State of Nigeria selling foodstuff (Rice and Beans). He buys from the distributor in lorry loads every month and sells to customers in bags. At the end of each month, he takes decision on how to re-invest his funds based on the prevailing market price and the expectation of future market price. According to the vendor, there are periods the prices of the items rise, fall or remain stable.
In the study, we consider one item (Rice). There are situations where all the actions are not considered because of the prevailing market. The actions; short the stock, invest nothing and everything in cash, long the stock are the actions most frequently considered and taken by the vendor. While actions; invest the capital on a risk free item, invest the whole capital plus any excess amount above the working capital are considered and taken in rare occasions.

The table below shows the summary of information and data recorded by vendor on the movement of the item (Rice) from one state to another based on the price of rice per bag for four years $(2012-2015)$.

**Table 1**: Data – The movement of the item from one state to another

| | 0 | 1 | 2 | $n_i$ |
|---|---|---|---|---|
| 0 | 5 | 4 | 1 | 10 |
| 1 | 2 | 12 | 6 | 20 |
| 2 | 1 | 1 | 18 | 20 |

Where state $0=$ Bear market $(\prec N7{,}500)$

$1=$ Recession $(N7{,}500-7{,}800)$ inclusive

$2=$ Bull market $(\succ N7{,}800)$ per a bag.

$5$ is the number of time the price remained in state $0$ (Bear market), while $4$ is the number of time the price moved from state $0$ to state $1$ and so on.

**Table 2**: The expected return for each state of the Markov chain

| **State** | **Expected return** $\hat{\alpha}_i$ |
|---|---|
| **0** | -0.01 |
| **1** | 0.0001 |
| **2** | 0.005 |

The expected return for each state was obtained by taking the average of his profit on those months he had bear market, recession and bull market respectively. It is expressed as the proportion of money invested. The negative sign shows that on average he was at loss.

## RESULTS AND DISCUSSIONS

Using multinomial distribution, the transition probabilities are estimated as $p_{ij}=\frac{n_{ij}}{n_{i.}}$, where $n_{ij}$ is the number of time the price moved from state $i$ to state $j$ and $n_{i.}$ is the number of time the price is in state $i$. From table $1$, the transition probability matrix is as given in table 3 below.

**Table 3**: Transition probability matrix for the Markov model

| | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 0.5 | 0.4 | 0.1 |
| 1 | 0.1 | 0.6 | 0.3 |
| 2 | 0.05 | 0.05 | 0.9 |

Recall that states $0,1,2$ are the states of Markov chain.

Each state of MDP contains the Markov model state and the weight assigned to it. Now, we have three states, three weights and one item, the cardinality of the state space for one item is $MW = 3x3 = 9$. Therefore, the entire MDP state space is written as $S = \{(0,-1),(1,-1),(2,-1),(0,0),(1,0),(2,0),(0,1)(1,1),(2,1)\}$, where the first number is the Markov state and second the weight of the item. We note that there is a limited set of actions for each state. This is because the vendor does not consider all the actions at the same time due to the prevailing market. The set of actions he considers depending on the prevailing market price at each decision are $(0.1,2),(-2,-1,0),and(-1,0,1)$. If the current weight on the item is $-1$ with the signal in state $i$, the available set of actions is $A_{i,-1} = \{0,1,2\}$ ie $A_{(0,-1),(1,-1),(2,-1)} = \{0,1,2\}$. If the current weight is $0$, the available set of actions is $A_{i,0} = \{-1,0,1\}$, ie $A_{(0,0),(1,0),(2,0)} = \{-1,0,1\}$ and so on. Since we consider one item, we then build up the transition probability matrix for the MDP for each action taken which incorporates weights from the transition probability matrix for the markov model. We then place these probabilities (table 3) in the proper cells since the weight do not affect the transition probabilities.

**Table 4**: Transition probability matrix for MDP for action $-2$

| | 0,-1 | 1,-1 | 2,-1 |
|---|---|---|---|
| 0,1 | 0 − 5 | 0.4 | 0.1 |
| 1,1 | 0.1 | 0.6 | 0.3 |
| 2,1 | 0.05 | 0.05 | 0.9 |

For action $-2$, we observe that action $-2$ is available in $A_{i,0} = \{-2,-1,0\}$, that is in states $(0,1),(1,1),(2,1)$ and states $(0,-1),(1,-1),(2,-1)$ are possible future states for transition. Similarly, the other transition probability matrices for each action are obtained in the same manner.

**Table 5**: Transition probability matrix for action $-1$

| | | | | 0,-1 | 1,-1 | 2,-1 |
|---|---|---|---|---|---|---|
| | 0,0 | 1,0 | 2,0 | | | |
| 0,0 | 0.5 | 0.4 | 0.1 | 0 | 0 | 0 |
| 1,0 | 0.1 | 0.6 | 0.3 | 0 | 0 | 0 |
| 2,0 | 0.05 | 0.05 | 0.9 | 0 | 0 | 0 |
| 0,1 | 0 | 0 | 0 | 0.5 | 0.4 | 0.1 |
| 1,1 | 0 | 0 | 0 | 0.1 | 0.6 | 0.3 |
| 2,1 | 0 | 0 | 0 | 0.05 | 0.05 | 0.9 |

**Table 6**: Transition probability matrix for action 0

| | 0,-1 | 1,-1 | 2,-1 | 0,0 | 1,0 | 2,0 | 0,1 | 1,1 | 2,1 |
|---|---|---|---|---|---|---|---|---|---|
| 0,−1 | 0.5 | 0.4 | 0.1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1,−1 | 0.1 | 0.6 | 0.3 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2,−1 | 0.05 | 0.05 | 0.9 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0,0 | 0 | 0 | 0 | 0.5 | 0.4 | 0.1 | 0 | 0 | 0 |
| 1,0 | 0 | 0 | 0 | 0.1 | 0.6 | 0.3 | 0 | 0 | 0 |
| 2,0 | 0 | 0 | 0 | 0.05 | 0.05 | 0.9 | 0 | 0 | 0 |
| 0,1 | 0 | 0 | 0 | 0 | 0 | 0 | 0.5 | 0.4 | 0.1 |
| 1,1 | 0 | 0 | 0 | 0 | 0 | 0 | 0.1 | 0.6 | 0.3 |
| 2,1 | 0 | 0 | 0 | 0 | 0 | 0 | 0.05 | 0.05 | 0.9 |

**Table 7**: Transition probability matrix for action $1$

| | 0,0 | 1,0 | 2,0 | 0,1 | 1,1 | 2,1 |
|---|---|---|---|---|---|---|
| 0,−1 | 0.5 | 0.4 | 0.1 | 0 | 0 | 0 |
| 1,−1 | 0.1 | 0.6 | 0.3 | 0 | 0 | 0 |
| 2,−1 | 0.05 | 0.05 | 0.9 | 0 | 0 | 0 |
| 0,0 | 0 | 0 | 0 | 0.5 | 0.4 | 0.1 |
| 1,0 | 0 | 0 | 0 | 0.1 | 0.6 | 0.3 |
| 2,0 | 0 | 0 | 0 | 0.05 | 0.05 | 0.9 |

**Table 8:** transition probability matrix for action $2$

| | 0, 1 | 1,1 | 2,1 |
|---|---|---|---|
| 0,−1 | 0.5 | 0.4 | 0.1 |
| 1,−1 | 0.1 | 0.6 | 0.3 |
| 2,−1 | 0.05 | 0.05 | 0.9 |

We apply the value iteration algorithm on equation (1) using the transition probabilities for each action .

chosen and the expected return for each Markov state.

Let the discount factor $\lambda = 0.95$ and set $\varepsilon = 0.01.$ the cost function is given as $C(a_t = 0.003 a_t)$. The software $C^{++}$ was used to run the algorithm and the result of the iteration is as shown in table below. We obtain the convergence of expected reward for each state every $10$ iteration

**Table 9:** Convergence of the expected reward for each state every $10$ iterations

| N/State | (0,-1) | (0,0) | (0,1) | (1,-1) | (1,0) | (1,1) | (2,-1) | (2,0) | (2,1) | Epsilon optimal $(\varepsilon)$ |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| **1** | 0.010 | 0.007 | 0.004 | 0 | 0 | 0 | -0.001 | 0.002 | 0.005 | 0.010 |
| **10** | 0.035 | 0.032 | 0.029 | 0.022 | 0.024 | 0.025 | 0.029 | 0.032 | 0.035 | 0.003 |
| **20** | 0.055 | 0.052 | 0.049 | 0.041 | 0.043 | 0.044 | 0.049 | 0.052 | 0.055 | 0.002 |
| **30** | 0.066 | 0.063 | 0.060 | 0.052 | 0.055 | 0.056 | 0.060 | 0.063 | 0.066 | 0.00091 |
| **40** | 0.073 | 0.070 | 0.067 | 0.059 | 0.061 | 0.063 | 0.067 | 0.070 | 0.073 | 0.00054 |
| **50** | 0.077 | 0.074 | 0.071 | 0.064 | 0.066 | 0.067 | 0.072 | 0.075 | 0.078 | 0.00032 |

We also obtain the optimal policy for each state, by choosing the action that maximizes the expected reward as given from value iteration as in table below. That is the policy mapping an action to each state using step $4$ of the algorithm.

**Table 10**: Action mapped to each state using step $4$ of the algorithm

| State | (0,-1) | (0,0) | (0,1) | (1,-1) | (1,0) | (1,1) | (2,-1) | (2,0) | (2,1) |
|---|---|---|---|---|---|---|---|---|---|
| **Action** | 0 | -1 | -2 | 0 | 0 | 0 | 2 | 1 | 0 |

In table 3,it is observed that the system has a very high probability $(0.9)$ of staying in state $2$ (Bull market) and a very low probabilities (0.05) and (0.05) to move from states $2$ to $0$ and $1$ respectively. In table 2, the 0ptimal policy is predictable. When the item is in state $0$ which predicts a strong negative return, we always perform the action that gives us a short position to capture the negative expected return. Similarly, when the item is in state $2$, we long the stock to capture the strong positive return.

By inspecting the expected reward of every state in table $9$, you would expect the optimal policy to always short the stock when in state $0$ because of its large return. However, while the reward is not as large as state $0$, the probability $(0.9)$ of staying in state $2$ is so high enough that we should long the stock because we expect high reward for several periods. The $\varepsilon-$optimal policy is as given in the last column of table $9$. Table $10$ shows the optimal action

mapped to each state using step 4 of the algorithm to obtain the expected reward.

*Conclusion:* In the study, we applied Markov decision process to a portfolio allocation problem under vendor managed inventory in an uncertain market environment. The objective is to obtain the optimal policy that maps an action to a given state in order to maximize the expected reward. The prices of the items are uncertain and change at any time. The vendor has to choose actions based on the price ( the prevailing market price and the expected future price) to maximize its expected reward. The problem was formulated in the frame work of Markov decision process and value iteration algorithm was adopted to obtain the optimal policy. In the case study, the optimal is expected to always short the stock when in state 0 because of its large return. While the return is not as large as in state 0, the probability of staying in state 2 is so high enough that the vendor should long the stock for he expects high returns for several periods. We were also able to obtain the convergence of the expected reward for each state every 10 iterations as shown in table 9. Table 10 shows the best action for each state.

We conclude that Markov decision process is a good model for solving portfolio allocation problem under vendor managed inventory in an uncertain market environment despite the small state/action spaces. One could apply other optimization models to take care of situations where the state/action spaces are large or considered to be infinite. This research work can be extended to two item case where each item follows a different Markov process. With more than one item in the system, we must consider the relationship between the items, defined by conditional probabilities. Study should be made when the items are independent and when they are correlated. Each item has a different transition probability matrix, which is used to build up the transition probability matrix for the Markov Decision Process for a given action when the items are independent or correlated.

## REFERENCES

Arsael, P.J. (2009). GPU-Based Markov Decision Process Solver. An unpublished M.Sc Thesis, Reykjavik University – School of Computer Science.

Christophette, B.S; Rajna, G.B; Benoite, D.S; Denis, T.E, (2004). Viscosity Solutions to Optimal Portfolio Allocation Problem in Models with Random Time Changes Transaction Costs. ( National Centre of Competence in Research, Financial Valuation and Risk Management). Working Paper No. 552.

Chukwu, W.I.E;Encho, L.T. (2009), Vendor Managed Inventory. Journal of Statistical Association, Book of Abstract.

David, R. (2008), A Robust Optimization Approach to the Portfolio Selection Problem Using Mean Absolute Deviation Model. Bachelor of Applied Science Thesis: Department of Industrial Engineering, University of Toronto.

Dimitrios, K (2013). Stochastic Dynamic Programming Methods for the Portfolio Selection Problem..An unpublished ph.D Thesis, Department of Management, London School of Economics.

Dror, M; Ball, M. (1987), Inventory/Routing; Reduction from Annual to a Short Period Problem. Naval Research Logistics Quarterly..34: 891-905

Dror, M;Trudean, P. (1996), Cash Flow Optimization in Delivery Scheduling. .European Journal of Operation Research. 88:504-515.

Haleh, V. (2009), Optimization Portfolio Selection. A ph.D Dissertation, Graduate School-New Brunswick Rutgers, University of New Jersey.

Kobbane, A; Koulali, M; Tembine, H; Kofbi, M; Ben-othman, J. (2012), Dynamic Power Control with Energy Constraints for Multimedia Wireless Network. In Proceedings of the IEEE International Conference on Communications. IEEE 2012: 518-522.

Konno, H; Yamazaki, H, (1991). Mean Absolute Deviation Portfolio Optimization Model and its Applications to Tokyo Stock Management. Management Science 37.5: 519 – 531.

Kwame, O.H , (2009). Application of Markov Decision Processes to a Simplified Model of Robotic Firefighter.PGD Thesis. African Institute of Mathematical Sciences (AIMS ).

Markowitz Harry, M.(1952).Portfolio Selection. Journal of Finance, 7 No 1, 77 – 91.

Md. Noor-E-Alam; John, D, (2010). Stochastic Investment Decision Making with Dynamic Programming. (Proceedings of the 2010

International Conference on Industrial Engineering and Operations Management, Dhaka, Bangladesh, January 9 -10, 2010).

Mohammad, A.A; Thai Hoang, D; Disit, N; Hwee-Pinch, T; Shaowei, L., (2015). Markov Decision Process with Application in Wireless Sensor Network; A Survey. G-xiv; 1501.00644VI[CS.NI], 4th January, 2015.

Puterman, M.L., (1994). Markov Decision Process: Discrete Stochastic Dynamic Programming. John Wiley & Sons, New York.

Warren, B; Van Roy, B, (2003). Approximate Dynamic Programming for High-Dimensional Dynamic Resource Allocation Problem. In Handbook of Learning and Approximate Dynamic Programming. 261- 279