Journal of Applied Mathematics and Physics, 2023, 11, 670-678
https://www.scirp.org/journal/jamp
ISSN Online: 2327-4379
ISSN Print: 2327-4352

Management Tools for Random Demand Inventory and Supply

Jeremie Ndikumagenge, Jean Pierre Ntayagabiri, Vercus Ntirandekura

Center for Research in Infrastructure, Environment and Technologies "CRIET", University of Burundi, Bujumbura, Burundi
Email: jeremie.ndikumagenge@ub.edu.bi, jpntayaga2@gmail.com, ntirandekuraczi@gmail.com

How to cite this paper: Ndikumagenge, J., Ntayagabiri, J.P. and Ntirandekura, V. (2023) Management Tools for Random Demand Inventory and Supply. *Journal of Applied Mathematics and Physics*, **11**, 670-678.
https://doi.org/10.4236/jamp.2023.113044

Received: December 24, 2022
Accepted: March 12, 2023
Published: March 15, 2023

Abstract

A mathematical management model's added value is obtained only after the design and implementation of a user-friendly operating and usage tool. Following work on developing an automated inventory management system and/or supplies, a dynamic model for the rational management of product stocks was established. Its implementation aims to limit or eliminate overstocking and/or stock depletion. The orderable quantity prediction tool based on a settable and preset time period demonstrates the added value of incorporating probabilistic mathematical principles into supply management processes. In this context, this article discusses aspects of the design and implementation of random demand management algorithms based on Markov chains. The goal is to forecast the state or behavior of goods marketing company's product stocks and to develop a user supply management interface. The latter's functional application will ultimately demonstrate the accuracy of the model. This paper also looks at how to use Markov chains to predict the reliability of any technical device, as well as how to implement an automated system with the desired technical specifications.

Keywords

Model, Algorithm, Markov Chain, Transition Matrix, Steady State, State Space

1. Introduction

The exchange of goods and services is one of the activities that materialize human interactions. This quasi-routine activity involves physical entities or people, as well as legal entities such as businesses and corporations.

The volume of trade over the last ten years shows that the number of goods and services traded has steadily increased throughout history [1]. The size and number of exchanges are accompanied, among other things, by 1) difficulties in

managing and forecasting the quantities to be ordered in order to obtain supplies, build up reserves, and ensure that stock does not run out or overflow; and 2) customer behavior uncertainty manifested by ignorance of the quantities and frequency with which they can buy over a given period of time. These last activities, carried out solely by human intuition on a daily basis, increase the likely margins of error in estimating salable and/or flow out quantities.

Marketing storable goods are activities whose realization recalls random events and/or processes that can be executed over time. As a result, Markov chains accurately model this type of process. They are distinguished by the fact that the probabilities implying how the process will evolve in the future are determined solely by the current state of the process and are thus independent of past events [2]. A stochastic process is a random phenomenon that evolves over time.

The Markov chain in stochastic analysis specifies a system of transitions of an entity from one state to another. A Markov chain's states are divided into two categories: transient states, which are only visited an almost finite number of times (p.s.), and recurrent states, which are visited an almost infinite number of times (p.s.), as well as all other states in the same recurrence class. A recurrent irreducible Markov chain and the process's marginal law converge to either the unique P-invariant probability measure or the null vector or null recurrence. This theory is particularly applicable to random markets and queuing models.

The need to ensure the reliability of a system arose around the turn of the twentieth century, peaking during World War II. Currently, the study of dependability is evolving into its own discipline. Economic issues, for example, necessitate a certain level of inherent dependability. The ability to predict system reliability is critical for ensuring the safety of braking systems, nuclear systems, computer systems, and other systems. In short, reliability analysis is a critical step in any study of a system's dependability [3].

It is critical to characterize a system's behavior throughout its life cycle. The virtual impossibility of repairing certain equipment (satellites), the cost of system failures, spare parts inventories, and so on demonstrates the importance of understanding the reliability of the systems used. Because failures generally occur at random, it is logical to use probability calculations to strengthen their reliability through prediction. The reliability of a device, defined as its probability of functioning correctly for a given duration, is equivalent to determining the likelihood that no failure will occur during this duration [4].

2. Materials, Tools, Equipment and Methods

Random events or processes of product supply management necessitate the use of appropriate materials and methods. As stated in the introduction, this work makes use of the tools provided by Markov chains and its eventual applications.

2.1. Material

Because we are dealing with processes for managing a system with discrete ran-

dom events, the implementation of such a system is centered on the use of adjacency matrices and state transition matrices, as well as the design of process-specific execution algorithms. As stated in the introduction, this work makes use of the capabilities provided by Markov chains as materials and equipment.

2.1.1. Discrete-Time Markov Chains

Let $\{X_n, n \geq 0\}$ be a sequence of random variables with values in the set of states E assumed to be equal to *N*. We say that this sequence is a Markov chain if $\forall n \geq 0$ and any sequence $(i_0, i_1, \cdots, i_{n-1}, i, j)$, we have:

$$\mathbb{P}\left(\overbrace{X_{n+1}=j}^{\text{Future}} \mid \overbrace{X_n=i_n, X_{n-1}=i_{n-1}, \cdots, X_0=i_0}^{\text{Present and pass time}}\right) = \mathbb{P}\left(\underbrace{X_{n+1}=j}_{\text{Future}} \mid \underbrace{X_n=i}_{\text{Present}}\right) \quad (1)$$

probability distributions characterize this chain:

1) One-dimensional distribution:

$$P_i(n) = P\{X_n = i\}, \forall n, i \in S \quad (2)$$

2) The conditional probability distribution (also known as the transition probability distribution):

$$P_{i,j}(n) = P\{X_{n+1} = j \mid X_n = i\}, \forall i, j, n \in S \quad (3)$$

Consider $(X_n)_{n \in N}$ to be a Markov chain with values in *S*.

The Markov chain is said to be homogeneous if the probability of passing to state *j* at time $n+1$ knowing that we were in state *i* at time *n* does not depend on *n*:

$$P(X_{n+1} = j \mid X_n = i) = P(i, j) \text{ for all } n \in N. \quad (4)$$

where are, the transition probabilities of the Markov chain are denoted by the numbers $P_{(i,j)}$, $(i, j) \in S^2$.

2.1.2. Probabilities and Transition Matrix

The probability that $X(n)$ is in state $i \in S$ is one of the state probabilities $\pi_i(n) = P(X(n) = i)$, the sum of which is obviously unity:

$$\sum_{i \in S} \pi_i(n) = 1 \quad (5)$$

while $p_{i,j}(n) = P(X(n+1) = j \mid X(n) = i)$ are the probabilities of one-step transitions from state *i* to state *j* at time *n*, satisfying the relation:

$$\sum_{j \in S} p_{ij}(n) = 1 \quad (6)$$

These transition probabilities are generally written in the form of a matrix called the Transition Matrix or Stochastic Matrix:

$$P = \begin{pmatrix} p_{00} & p_{01} & \cdots & p_{0i} & \cdots \\ p_{10} & p_{11} & \cdots & p_{1i} & \cdots \\ \vdots & \vdots & \ddots & \vdots & \cdots \\ p_{i0} & p_{i1} & \cdots & p_{ii} & \cdots \\ \vdots & \vdots & \ddots & \vdots & \ddots \end{pmatrix} \quad (7)$$

whose sum of the elements in the same row equals one.

The probabilities of two-step transitions from state i to state j are calculated as follows:

$$\begin{aligned} p_{ij}^2 &= P\left(X\left(n+2\right)=j \mid X\left(n\right)=i\right) \\ &= \sum_{k\in S} P\left(X\left(n+2\right)=j \mid X\left(n+1\right)=k, X\left(n\right)=i\right) P\left(X\left(n+1\right)=k \mid X\left(n\right)=i\right) \\ &= \sum_{k\in S} P\left(X\left(n+2\right)=j \mid X\left(n+1\right)=k\right) P\left(X\left(n+1\right)=k \mid X\left(n\right)=i\right) \\ &= \sum_{k\in S} p_{kj}\, p_{ik} \end{aligned} \tag{8}$$

which is written as a matrix:

$$P^{(2)} = PP = P^2 \tag{9}$$

We can use the Chapman-Kolmogorov equations to obtain the m-step transition probabilities from state i to state j:

$$\begin{aligned} p_{ij}^{(m)} &= P\left(X\left(n+m\right)=j \mid X\left(n\right)=i\right) \\ &= \sum_{k\in S} P\left(X\left(n+m\right)=j \mid X\left(n+m-l\right)=k, X\left(n\right)=i\right) P\left(X\left(n+m-l\right)=k \mid X\left(n\right)=i\right) \\ &= \sum_{k\in S} P\left(X\left(n+m\right)=j \mid X\left(n+m-l\right)=k\right) P\left(X\left(n+m-l\right)=k \mid X\left(n\right)=i\right) \\ &= \sum_{k\in S} p_{kj}^{(l)}\, p_{ik}^{(m-l)} \end{aligned} \tag{10}$$

for all $i \in S$; $j \in S$; $l = 1, 2, \cdots, m$; $m = l+1, l+2, \cdots$.

These equations show that if you go from state i to state j in m steps, you will arrive at state k after exactly l (l being less than m) states. Thus, $p_{kj}^{(l)}\, p_{ik}^{(m-l)}$ is only the conditional probability that, given a starting point of state j, the process proceeds to state k after l steps, then to state i after $m - l$ steps. As a result, the sum of these conditional probabilities over all possible k must equal $p_{ij}^{(m)}$. The special case of $l = 1$ yields Equation (10):

$$p_{ij}^{(m)} = \sum_{k\in S} p_{kj}\, p_{ik}^{(m-1)} \tag{11}$$

or in a matrix form

$$P^m = P^{(m-1)}P = \cdots = P^m \tag{12}$$

2.2. Methods

Inventory and supply management is an important aspect of a business's operations. When it is effective, it reduces overstocking and increases the overall profitability of the company. Every procurement manager is faced with the daily task of making the best use of their products storage, namely inventory and supply storage management. At daily basis, he should dynamically modify inventories without exposing the company to risk of stock-out [5]. Overstocks result in additional costs due to storage fees on the one hand, while stocks out result in a revenue loss due to the client's selflessness in the company on the other. The Markov's chain addresses the inventory and supply process with threshold limits using probabilistic laws and principles.

Thus, a careful examination of the events that occur during the marketing of storable goods (stock) reveals that these processes can be correctly represented

or modeled by Markov chains. The temporary and temporal storage of these goods can be studied using Markov chain concepts. A Markov chain is a random process $\left(X_n\right)_{n\in N}$, the transitions of which are determined by a stochastic matrix $P\left(X_n, X_{n+1}, X_{n+2}, \cdots\right)$. These processes demonstrate the Markov property, namely that observed from a (stopping) time T, $\left(X_{T+n}\right)_{n\in N}$ depends solely on X_T and is, once again, a Markov chain. Here are some major algorithms related to Markov chain mathematical tool. Figures 1-4 show the implemented Markov chain algorithms, while Figure 5 shows the transition probability. The all algorithms presented in this sub-subsection are all written in the form of program in high level language Java.

2.2.1. Algorithm for the Markov Chain Test

The Markov chain testing algorithm is depicted in the Java source code in Figure 1. The final one ensures that the modeled processes exactly correspond to the Markov chain.

2.2.2. Algorithm for Calculating Matrices' Products

As shown in Figure 2, one of the main methods in state transition matrices is the matrices' products algorithm.

2.2.3. Chapman-Kolmogorov Equation Solution Algorithm

The algorithm for solving the Chapman-Kolmogorov equations computes the transition probabilities, which are then used to determine the stationary probability state. Figure 3 shows the algorithm as depicted in the java source code.

2.2.4. Algorithm for Constructing State Transition Matrices

The Transition matrix allows us to determine the stock state at any given time. Figure 4 depicts the algorithm in pseudo-code form using the Java programming language paradigm.

2.2.5. Function for Calculating Transition Probability

The transition probability function shows the probability transition from one state to another at a given time. Figure 5 depicts the function used to calculate transition probabilities during the prediction process.

```
public class markov {
    static boolean checkMarkov(double m[][])
    {
        for(int i=0;i<m.length;i++)
        {
            double sum=0;
            for(int j=0;j<m[i].length;j++)
                sum=sum+m[i][j];
                if(sum!=1)
                    return false;
        }
        return false;
    }
}
```

Figure 1. Algorithm for the Markov chain test.

```
public class MatrixMultiplication {

   public static void main(String[] args) {
       Scanner s = new Scanner(System.in);
       System.out.print("Enter number of rows in A :");
       int rowsinA = s.nextInt();
       System.out.print("Enter number of columns in A/rows in B :");
       int columnsinA = s.nextInt();
       System.out.print("Enter number of columns in B :");
       int columnsinB = s.nextInt();
       int[][] a = new int[rowsinA][columnsinB];
       int[][] b = new int[columnsinA][columnsinB];
       System.out.println("Enter matrix A :");
       for (int i = 0; i < a.length; i++) {
           for (int j = 0; j < a[0].length; j++) {
               a[i][j] = s.nextInt();
           }
       }

       System.out.println("Enter matrix B :");
       for (int i = 0; i < b.length; i++) {
           for (int j = 0; j < b[0].length; j++) {
               b[i][j] = s.nextInt();
           }
       }

       int[][] c = multiply(a, b);
       System.out.println("Product of A and B is :");

    for(int i=0;i<c.length;i++)
    {
        for (int j = 0; j < c[0].length; j++)
        {
            System.out.println(c[i][j]+ "");
     }
       System.out.println();
   }
 }

 public static int[][] multiply(int[][] a, int [][] b){
     int rowsinA = a.length;
     int columnsinA = a[0].length;
     int columnsinB = b[0].length;
     int [][] c = new int[rowsinA][columnsinB];
     for (int i = 0; i < rowsinA; i++) {
         for (int j = 0; j < columnsinB; j++) {
             for (int k = 0; k < columnsinA; k++) {
                 c[i][j] = c[i][j]+a[i][k]*b[k][j];
             }
         }
         System.out.println();
     }
     return c;

   }
 }
```

Figure 2. Algorithm for calculating matrices' products [6].

```
public class GFG {
    static double [][] multiply(double [][]G, double [][]B,int N){
        double[][]C= new double [N][N];
        for(int i=0;i<N;++i)
            for(int j=0;j<N;++j)
                for(int k=0;k<N;++k)
                    C[i][j]+=G[i][k]*B[k][j];
        return C;
    }
    static double[][] matrix_power(double[][]M,int p,int n){
        double[][]G=new double[n][n];
        for(int i=0;i<n;++i)
            G[i][i]=1;
        while(p>0)
        {
            if(p%2==1)
                G=multiply(G,M,n);
            M=multiply(M,M,n);
            p/=2;
        }
        return G;
    }
}
```

Figure 3. Chapman-Kolmogorov equation solution algorithm.

```
public class CMTD {
    private int k;
    private double p,q;
    private double[][]P;
    public CMTD (int k, double p, double q, int choix){
        this.p=p;
        this.q=q;
        this.k=k;
        if(choix==1)
            P=matrice();
        else if (choix==2)
            P=matriceInfinie();
        else
            System.out.println("Probleme");
    }
    public double [][] get_matrice(){
        return P;
    }
    public static void affichage (double[][] matrice){
        for(int i=0;i<matrice.length;i++){
            for(int j=0;j<matrice.length;j++)
                System.out.println(matrice[i][j]);
            System.out.println("");
        }
    }

    public double[][]matrice(){
        double[][]P=new double[k+1][k+1];
        for(int i=0;i<=k;i++){
            for(int j=0;j<=k;j++){
                if(i==j && i==k){
                    P[i][j]=1-q;
                }
                else if(i==j && i==0){
                    P[i][j]=1-p;
                }
                else if(i==j){
                    P[i][j]=1-p-q;
                }
                else if (i>0 && j==i-1){
                    P[i][j]=q;
                }
                else if (i<k && j==i+1){
                    P[i][j]=p;
                }
                else{
                    P[i][j]=0;
                }
            }
        }
        return P;
    }
}
```

Figure 4. Algorithm for constructing state transition matrices.

```
static double findProbability(double[][]M,int N,int F,int S,int T)
{
    double[][]MT=matrix_power(M,T,N);
    return MT[F][S];
}
```

Figure 5. Function for calculating transition probability.

3. Results and Discussion

3.1. Results

3.1.1. Algorithm for the Markov Chain Test

The Markov chain testing algorithm ensures that the processes being modeled exactly correspond to the Markov chain and thus have Markov properties.

3.1.2. Chapman-Kolmogorov Equation Solution Algorithm

The algorithm for solving the Chapman-Kolmogorov equations allows for the determination or calculation of transition probabilities, including those in the steady state.

3.1.3. Algorithm for Constructing State Transition Matrices

The transition matrix construction algorithm enables the creation of a "mirror" of all possible states for a system. The algorithm for calculating the transition probability, on the other hand, displays the state of a system at a given instant over a specified time period.

3.2. Discussions

Similar to the supply management processes and tools that can be built using Markov chains, guaranteeing the reliability of a technical system, particularly an artificial one, over a given period is equivalent to calculating that no failure will occur during this period. As a result, the theory of reliability, whose goal is to investigate the ability of technical devices to perform a required function under given conditions, during a given time, and for a given period, necessitates a mathematical modeling tool analogous to Markov chains. Satisfying the standards of reliability of technical systems through the application of probability theory of random phenomena appears to be a broad axis of application for the model designed and algorithms developed in this work.

In this era of increased commercial competition, decision-making by trial and error and/or improvised approximation based on natural human thought has only resulted in certain bankruptcy for the institution concerned. Thus, the algorithms developed in this article, when applied to automated inventory management models, provide numerous benefits, including reduced product storage costs and reduced shortfalls due to stock depletion.

This is demonstrated, for example, by the scientific contribution and applicability of the algorithm for solving the Chapman-Kolmogorov equations, the algorithm for constructing the transition matrix and the algorithm for calculating the probability of state transition.

4. Conclusions

The tools for implementing a model, no matter how well designed, are critical in ultimately constructing a management system with the desired technical specifications and obtaining the expected results.

The computer application of the discrete-time Markov chain model allows a company to manage the supply, storage, and sale processes of countable products without shortages and/or overstocks. Such a working method increases the system's dependability.

Using an illustrative example, this work demonstrated that the predictability of the reliability of a system for managing supplies, stocks, and sales of a company selling storable goods allows for the limitation, if not elimination, of the phenomena of stock depletion and/or stock overflow, thanks to a simple and ergonomic management interface in which the algorithms developed will be implemented. These results show that the management and decision-making support model accurately predicts the quantity in stock as well as the quantity to be ordered, avoiding the phenomenon of storage products exhaustion or overflow.

As a result, this tool ensures that additional costs due to additional warehousing costs and potential losses due to out-of-stock situations are avoided. Finally, the implementation of simple and ergonomic management interface algorithms simplifies product storage monitoring procedures and makes managers' daily tasks easier.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Santhi, P.K. (2019) Markov Decision Process in Supply. Madurai Kamaraj University, Tamil Nadu.

[2] Hillier, F.S. (1967) Introduction to Operations Research. Holden-Day, San Francisco.

[3] Deflem, Y. and Van Nieuwenhuyse, I. (2011) A Discrete Time Markov Chain Model for a Periodic Inventory System with One-Way Substitution. https://doi.org/10.2139/ssrn.1868085

[4] Agbam, A.S. (2020) Application of Markov Chain (CM) Model to the Stochastic Forecasting of Stocks Prices in Nigeria: The Case of Dangote Cement. *International Journal of Applied Science and Mathematical Theory*, **6**, 20.

[5] Onyesolu, M., Abara, J., Chukwuneke, C. and Asogwa, D. (2018) Modeling a Dynamic Supply Chain Management System for an Utility Company in Nigeria. *Journal of Software Engineering and Applications*, **11**, 275-284. https://doi.org/10.4236/jsea.2018.116017

[6] Cormen, T.H., Leiserson, C.E., Rivest, R.L. and Stein, C. (2010) Algorithmique. Dunod, Paris.