

AN APPLICATION OF GRAPH THEORY IN MARKOV CHAINS RELIABILITY ANALYSIS

Pavel SKALNY

Department of Applied Mathematics, Faculty of Electrical Engineering and Computer Science,
VSB–Technical University of Ostrava, 17. listopadu 15/2172, 708 33 Ostrava-Poruba, Czech Republic

pavel.skalny@vsb.cz

Abstract. *The paper presents reliability analysis which was realized for an industrial company. The aim of the paper is to present the usage of discrete time Markov chains and the flow in network approach. Discrete Markov chains a well-known method of stochastic modelling describes the issue. The method is suitable for many systems occurring in practice where we can easily distinguish various amount of states. Markov chains are used to describe transitions between the states of the process. The industrial process is described as a graph network. The maximal flow in the network corresponds to the production. The Ford-Fulkerson algorithm is used to quantify the production for each state. The combination of both methods are utilized to quantify the expected value of the amount of manufactured products for the given time period.*

Keywords

Discrete time Markov chain, flow in network, reliability.

1. Introduction

The reliability of production plays the fundamental role in an industrial sphere. Nowadays the reliability of industry process is on a high level. It is increased by improving the quality of each component or by redundancy of the production process. Even though it is the top reliability process, there is still a chance that system fails. In our case we analyse the process which has no redundancy. Thus the information about the probability of the systems failures is very valuable.

In the previous work an [1] a reliability analysis of the part of an industry process was realized. In the former application there was analysed a part of the industry process distributed in parallel. The main aim of the cited paper was to estimate the probability, that

the production will be equal or greater than the industrial partner demand. To deal with the task a Monte Carlo simulation of Discrete time Markov chains was used.

The goal of this paper is to present the Discrete time Markov chain analysis realized on the system, which is not distributed in parallel. Since the analysed process is more complicated the graph theory approach seems to be an appropriate tool.

Graph theory has previously been applied to reliability, but for different purposes than we intend. One of example of application of graph theory in reliability is a reliability polynomial [3] or network reliability. In this approach a graph describes a network where each component (edge) has the same probability of fail. The most related work to our paper is probably the research of Christopher Dabrowski [4], [5]. As well as in our paper Dabrowski uses graph theory as a tool for counting with discrete time Markov chains. In contrast to our work the Markov chains described in Dabrowski one are used for different purposes. Dabrowski analyses a large scale grid. His study describe the application for finding states of the grid, which could lead to the system degradation. Dabrowski uses the graph theory to describe transitions of the Markov chain model between the initial state and the absorbing state. We use the graph theory to describe the analysed process on which the discrete time Markov chain is be applied.

2. Markov Chains

Markov chain is a random process with a discrete time set $T \subset \mathbb{N} \cup \{0\}$, which satisfies the so called “Markov property”. The Markov property means that the future evolution of the system depends only on the current state of the system and not on its past history.

$$\begin{aligned} \mathbf{P}\{X_{n+1} = x_{n+1} | X_0 = x_0, \dots, X_n = x_n\} = \\ = \mathbf{P}\{X_{n+1} = x_{n+1} | X_n = x_n\}, \end{aligned} \quad (1)$$

where X_1, \dots, X_n is a sequence of random variables. The index denotes certain time $t \in T$ x_1, \dots, x_n is a sequence of states in time $t \in T$. As a transition probability p_{ij} we regard probability, that the system changes from the state i to the state j :

$$p_{ij} = \mathbf{P}\{X_{n+1} = x_j | X_n = x_i\}. \tag{2}$$

Matrix \mathbf{P} , where p_{ij} is placed in row i and column j , is for all admissible i and j called transition probability matrix:

$$\mathbf{P} = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{pmatrix}. \tag{3}$$

Clearly all elements of the matrix \mathbf{P} satisfy the following property:

$$\forall i \in \{1, 2, \dots, n\} : \sum_{j=1}^n p_{ij} = 1. \tag{4}$$

As a probability vector we will understand a vector \vec{v} where, an amount of its elements v_i are equal to the amount of states in Markov chain and the following equations holds:

$$\forall v_i \in \vec{v} : v_i \leq 1, \text{ and } \sum_{all i} = 1. \tag{5}$$

By $\vec{v}(t)$ we denote vector of probabilities of all the states in time $t \in T$. By $\vec{v}(0)$ we denote an initial probability vector. Usually, all its coordinates are equal to zero except the first, which is equal to 1. The vector $\vec{v}(0)$ denotes the probability in which state the system occurs in time $t = 0$. It is easy to proof that [6]:

$$\vec{v}(t) = \vec{v}(0) \cdot \mathbf{P}^t. \tag{6}$$

As a stationary distribution we will understand a vector \vec{z} which satisfies the property:

$$z = \vec{z}\mathbf{P}. \tag{7}$$

Suppose that the limit $\pi = \lim_{n \rightarrow \infty} \vec{\pi}(n)$ exists, then the probability vector $\vec{\pi}$ is called limiting distribution (limiting vector):

$$\pi = \lim_{n \rightarrow \infty} \vec{\pi}(0)\mathbf{P}^n. \tag{8}$$

In some literature there is written that, the stationary distribution is equal to the limiting distribution. Actually it is true only if the discrete time Markov chain (further as DTMC) satisfies the condition of ergodicity (irreducibility, aperiodicity). To be able to

calculate the limiting distribution π we will describe further properties of DTMC.

A Markov chain j is periodic with period p , if on leaving state j a return is possible only in a number of transitions that is a multiple of the integer $p > 1$. For example Markov chain with following transition probability matrix:

$$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

is periodic with period $p = 2$. The DTMC is said to be irreducible if every state can be reached from every other state.

The probability of ever returning to state j is denoted by f_{jj} and is given by:

$$f_{jj} = \sum_{n=1}^{\infty} f_{jj}^n. \tag{9}$$

If $f_{jj} < 1$, then state j is said to be transient. If $f_{jj} = 1$, the state is recurrent and we can define the mean recurrent time.

As mean recurrent time M_{jj} we will understand:

$$M_{jj} = \sum_{n=1}^{\infty} n f_{jj}^n. \tag{10}$$

If $M_{jj} < \infty$ (or equal to ∞), we say that the state is positive recurrent (or null recurrent). In a finite Markov chain, each state is either positive-recurrent or transient, and furthermore, at least one state must be positive-recurrent [6]. Theorem Let a Markov chain C be irreducible. Then C is positive-recurrent or null recurrent or transient, i.e.:

- all the states are positive-recurrent, or
- all the states are null-recurrent, or
- all the states are transient.

If all the states of DTMC are positive-recurrent and aperiodic, then the probability distribution $\vec{v}(n)$ converges to a limiting distribution π , which is independent of initial distribution $\vec{v}(0)$. The π can be obtained by solving the following equations:

$$\pi = \pi \mathbf{P}, \quad \pi > 0 \text{ and } \sum_{for all i} \pi_i = 1. \tag{11}$$

3. Preliminaries

In this section we will establish fundamental terminology of graph theory which will be used further in this paper.

3.1. Definition of an Oriented Graph

An oriented graph is an ordered pair $G = (V; E)$, where V is the set of vertices and E is the set of edges. $E \subseteq V \times V$.

3.2. Definition of Network

Network is a four-tuple $S = (G; s; t; x)$, where:

- G is an oriented graph,
- vertices $s \in V(G)$, $t \in V(G)$ are the source and sink,
- $x : E(G) \rightarrow \mathbb{R}^+$ is a positive labelling on edges, called edge capacities.

3.3. Definition of Flow in Network

Flow in a network $S = (G; s; t; x)$ is a function $f : E(G) \rightarrow \mathbb{R}_0^+$, where:

- no edge capacity is exceeded:
 $\forall e \in E(G) : 0 \leq f(e) \leq x(e)$,
- the conservation of flow equation holds:
 $\forall v \in V(G), v \neq z, s : \sum_{e \rightarrow v} f(e) = \sum_{e \leftarrow v} f(e)$.

The value of a flow f is $\|f\| = \sum_{e \leftarrow z} f(e) - \sum_{e \rightarrow z} f(e)$.

3.4. Definition of Cut

Cut in the network $S = (G; s; t; x)$ is such a set of edges C , $C \subseteq E(G)$, such that in the factor of graph G , $G - C$, no oriented path remains. As minimal cut we will understand the cut where each proper subset of C is not a cut in the network.

4. Problem Formulation

The research presented here, was motivated by the practical problem. Analysed company was asked to quantify the probability that production fails was. Knowledge of risk, that the order won't be delivered in time, is important for the partner's firm to establish sufficient goods supplies. In the previous application see [1] a reliability analysis of the part of an industry process was realized. For each machine we could distinguish two modes – in order '1' or in fail '0'. Thus the whole system could occur in one of the 2^n states where n is an amount of machines of analysed industry process. Since machines were organized in parallel it was easy to calculate a whole production of each state. The

production of the certain state was calculated as a sum of production of functional states. More complicated situation occurs when the system is not connected in parallel. The aim of this paper is to present a calculation of a maximal production of each of 2^n states by usage a graph flow in network theory. In the following application we will simplify the process that, even at the beginning the gods can be reached from the begin to end of the process. We assume that for each state the process will produce a maximum possible production w .

5. Application

In this section we will present an example how to use a flow in network theory in a reliability analysis. First, we will define states of the industry process and calculate a maximum production of each state by using the Ford Fulkerson algorithm. At the end we will demonstrate on the certain data how to calculate an expected value of production for a given time t .

The analyzed industry process consists of six machines. For each machine we distinguish 2 different states one - work, zero - in fail. Thus the industrial process consists of 2^6 different states. State denotes an ordered six -tuple of ones and zeroes.

Let us describe the industry process of a firm with an oriented network. Every machine is represented as a vertex. The begin and the end of the processes represented by a source and sink of the network. The begin of a process consists of acceptance of gods and division between machines of the process. The end of a process consists of product inspection and expediting to the customers. Oriented edges describes the direction of the production process. Labelling of vertices represents the maximum amount of gods processed by the certain machine. To be able to work with labelled edges each vertex V_1, V_2, \dots, V_6 from the structure of the process (Fig. 1) is replaced by two vertices connected by the edge labelled by the same value as former vertex.

For our purposes all edges except newly created edges (with original labelling of vertices) incident with source and sink are labelled by ∞ . For each of 2^6 state we will find the maximum flow in a network. For each state the edges incident with the vertices representing the machines occurred in failure will be removed from the network. To calculate the maximal production of each state we will find a max flow in network. In our application we will assume that the maximal flow is achieved for any possible state. Without this simplification, there would be nearly impossible to estimate the production of certain state.

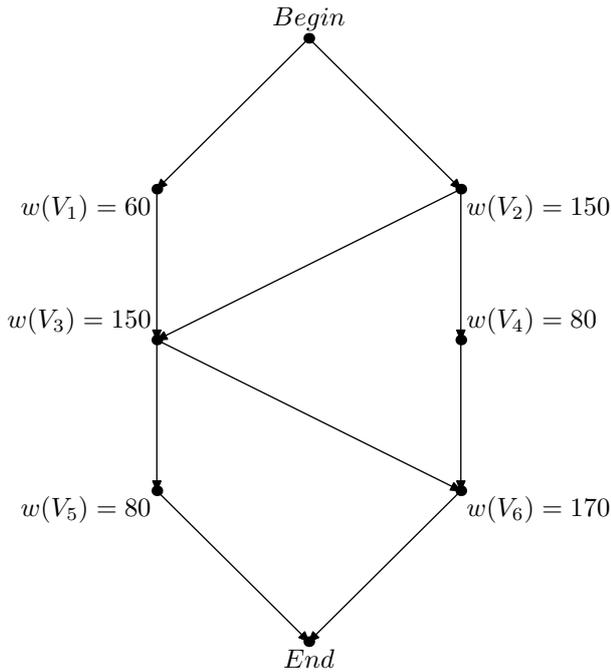


Fig. 1: Structure of process.

To find a maximal flow we will use a well known Ford-Fulkerson algorithm. For searching the graph edges a "Breadth-first" search was used. In the Tab. 1 there is a demonstration of several states and their production w_i .

To be able, to compute with DTMC, we need to calculate elements of transition probability matrix. To calculate the transition probability matrix \mathbf{P} we need to calculate probabilities of failure p_f , and probability of repair p_r for each machine. To estimate probability p_f that the system fails during one hour we calculated the expected value as an average length of period fail Q :

$$p_f = \frac{1}{Q}. \tag{12}$$

Using the maximum likelihood method we estimated the probability p_r which says that a machine will be repaired within one hour:

$$p_r = \frac{\Delta V}{V}, \tag{13}$$

where V is an amount of all repairs that were realized and ΔV is an amount of all repairs that lasted less than one hour.

The calculated probabilities for given machines V_1, V_2, \dots, V_6 are presented in the Tab. 2. Since there are 2^6 different states the probability matrix will consist of 2^{6+6} elements. With presented probabilities p_f, p_r we can calculate the probability transition matrix. For example $p_{11} = \prod_{i=1}^6 p_{fi}$ and p_{89} probability that the

system changes from the state 8 to the state 9 in the Eq. 14, see the Tab. 1:

$$p_{89} = (1 - p_{r1}) \cdot p_{r2} \cdot p_{f3} \cdot (1 - p_{f4}) \cdot (1 - p_{f5}) \cdot (1 - p_{f6}) = 0.64 \cdot 0.36 \cdot 0.005 \cdot 0.997 \cdot 0.996 \cdot 0.997 = 0.0011. \tag{14}$$

Tab. 1: Production of states.

state	V_1	V_2	V_3	V_4	V_5	V_6	prod. w
1	1	1	1	1	1	1	210
2	0	1	1	1	1	1	150
3	1	0	1	1	1	1	60
4	1	1	0	1	1	1	80
⋮	⋮	⋮	⋮	⋮	⋮	⋮	
8	0	0	1	1	1	1	0
9	0	1	0	1	1	1	80
⋮	⋮	⋮	⋮	⋮	⋮	⋮	
23	0	0	0	1	1	1	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	
64	0	0	0	0	0	0	0

Each element of transition probability matrix is a result of multiplication of non-zero probabilities. Since the probability matrix consists of non-zero elements (some of them re close to zero), every state can be reached from every other in time $t = 1$ the analyzed DTMC is irreducible and aperiodic. Because of the DTMC is finite and irreducible it is also positive-recurrent. Then we can use the Eq. 6 to calculate the limiting distribution π . The Eq. 11 were solved numerically by using the "backslash"(implemented solver of linear equations) from Matlab program. In Tab. 4 there are presented few Elements of the 1st row of transition probability matrix and limiting distribution π .

After calculating the transition probability matrix \mathbf{P} and the production of each state we can quantify the reliability of the industry process. To describe the reliability of the process we will calculate the expected value of production. The expected value of production W within time T is a sum of all expected values for each time step $t \in \{1, 2, \dots, T\}$. In our case, the expected value $E(W)(t)$ of production for the certain time t is equal to:

$$E(W)(t) = \sum_{i=1}^t \vec{v}(0)\mathbf{P}^t\vec{W}, \tag{15}$$

where \vec{W} is a vector of productions w of all 2^6 states. The results for calculating an expected value of production for time step 1 (time t) are presented in the Tab. 3 (maximal production for one time step is 170). According to the Tab. 3 we can conclude, that the expected value of production for the certain time quickly converges to the limit expected value ($\pi \cdot w$).

Tab. 2: Input probabilities.

prob.	V_1	V_2	V_3	V_4	V_5	V_6
p_f	0.100	0.006	0.005	0.003	0.004	0.003
p_r	0.360	0.360	0.400	0.375	0.240	0.190

6. Complexity of the Algorithm

Tab. 3: Calculation of expected value.

time	expected value	relative expected value
1	201.8095	0.9610
2	197.2069	0.9390
3	194.5720	0.9265
4	193.0340	0.9192
5	192.1160	0.9148
6	191.5550	0.9121
7	191.2030	0.9105
8	190.9750	0.9094
∞	190.4157	0.9067

Tab. 4: First row of transition probability matrix, limiting distribution π .

$p_{1,1}$	0.8810	π_1	0.7300
$p_{1,2}$	0.0980	π_2	0.2030
$p_{1,3}$	0.0050	π_3	0.0120
$p_{1,4}$	0.0040	π_4	0.0090
$p_{1,5}$	0.0030	π_5	0.0050
$p_{1,6}$	0.0040	π_6	0.0120
$p_{1,7}$	0.0010	π_7	0.0110
$p_{1,8}$	$5.1905e^{-5}$	π_8	0.0030
\vdots	\vdots	\vdots	\vdots
$p_{1,64}$	$1.0800e^{-13}$	π_{64}	$8.8980e^{-11}$

In this section we aim to estimate the complexity of the Ford-Fulkerson (F-F) algorithm for calculating the production \vec{W} of all states. As a $T(n)$ we will denote the worst-case time complexity of an algorithm with the input of size n . The complexity of F-F algorithm is $T(n) = O(n^3)$. In our case the size of input n is equal to the number of vertices $|V|$ of the network. The algorithm should run up to $2^{|V|}$. Thus $T(n) < O\left(\sum_{i=1}^{|V|} \binom{|V|}{i} (i)^3\right)$. The given formula is an upper bound of complexity of the algorithm for searching the vector of production \vec{W} in a network. To be able to estimate a complexity of the whole application we have to add a complexity of matrix exponentiation which is less than $T(n) < (2^{|V|})^3 t$ and multiplication of first row of transition probability matrix by the vector \vec{W} which is equal to $2^{|V|}$. The last form is not very significant In comparison with complexity of matrix exponentiation and calculating the vector \vec{W} of production.

The complexity of calculating the production \vec{W} will also depend on the amount of vertices of the network and on the character of the network. Another way how to decrease the amount of computation is to use the minimal cut. Let S_f fail state be a state with zero production w equal to zero. Let C denotes the set of vertices of the state S_f . Clearly the edges incident with C form the cut of the network. Any other state where the vertices C are in failure has the zero production.

For example in the Tab. 1 we can see that the production of the 8th (0,0,1,1,1,1) state is zero. Thus any state (for example the 23rd state), where the first and the second vertex is in fail has also the zero production. For those states, there is no need to use the Ford-Fulkerson algorithm to calculate the maximal flow. For the large time t there would be more suitable to use limit probability distribution than to exponentiate the matrix.

A problem could occur with saving the transition probability matrix in the computer memory. An amount of elements of the matrix grow exponentially (2^{2n}) with an amount of elements of analysed systems. The problem could be partially solved by rounding small values to zero and save the matrix as a sparse

matrix. In this case there should be more complicated to proof the convergence of the DTMC. Another way how to work with large matrices is to divided the DTMC into more separate systems and calculate with their transition probability matrix separately. Other possibility how reduce the transition probability matrix is to shorter the time step and define only one possible change within the time step. It has the effect of simplifying the transition matrix.

7. Conclusion

In the paper we have presented the application of graph flow in the network in the reliability analysis. The main worth of this work is an innovative usage of discrete time Markov chains and the flow in network theory in reliability analysis. The DTMC we used to describe the changes of the analysed process. For calculating for larger time step t , there is possible to use the limiting distribution instead of exponentiate the probability matrix. The Ford-Fulkerson algorithm was used to compute the industry production of analysed system. The main disadvantage of the presented approach is a high computational complexity. In further research there is possible to modify the Ford-Fulkerson algorithm to be more suitable for our purposes.

Acknowledgment

This work was supported by the Grant of SGS No. SP2013/116, VSB–Technical University of Ostrava, Czech Republic.

References

- [1] SKALNY, P. and B. KRAJC. Discrete-Time Markov Chains in Reliability Analysis-Case Study. In: *International Joint Conference CISIS'12-ICEUTE'12-SOCO'12 Special Sessions*. Ostrava: Springer Berlin Heidelberg, 2012, pp. 421–427. ISBN 978-3-642-33017-9. DOI: 10.1007/978-3-642-33018-6_43.
- [2] WEST, D. B. *Introduction to graph theory*. N.J.: Prentice Hall, 2001. ISBN 01-301-4400-2.
- [3] PAGE, L. B. and J. E. PERRY. Reliability polynomials and link importance in networks. *IEEE Transactions on Reliability*. 1994, vol. 43, iss. 1, pp. 51–58. ISSN 0018–9529. DOI: 10.1109/24.285108.
- [4] DABROWSKI, Ch., F. HUNT and Katherine MORRISON. *Improving the Efficiency of Markov Chain Analysis of Complex Distributed Systems*. National Institute of Standards and Technology. Gaithersburg MD, 2010. Available at: http://www.nist.gov/customcf/get_pdf.cfm?pub_id=906149.
- [5] DABROWSKI, Ch. and F. HUNT. Using Markov Chain and Graph Theory Concepts to Analyze Behavior in Complex Distributed Systems. In: *The 23rd European Modeling and Simulation Symposium*. Rome: Curran Associates, Inc., 2011. Available at: <http://www.nist.gov/itl/antd/upload/DabrowskiHunt-Paper129-updated.pdf>.
- [6] Stewart, W. *Introduction to the Numerical Solution of Markov Chains*. Princeton University Press, 1994. ISBN 0-691-03699-3.
- [7] CORMEN, T. H. *Introduction to algorithms*. Cambridge: MIT Press, 2001. ISBN 02-620-3293-7.

About Authors

Pavel SKALNY was born in Ostrava. He received his M.Sc. from Mathematics and History in Silesian University in Opava in 2009. His research interests include Stochastic modelling in reliability and fracture mechanics modelling.