An Extension of a Polynomial Time Algorithm for the Calculation of the Limit State Matrix in a Random Graph

Dmitrii Lozovanu
Institute of Mathematics and Computer Science, Academy of Sciences,
Academy str., 5, Chisinau, MD–2028, Moldova
lozovanu@usm.md

Stefan Pickl
Institut für Theoretische Informatik, Mathematik und Operations Research,
Fakultät fur Informatik, Universität der Bundeswehr, München
stefan.pickl@unibw.de

Abstract A characterization of a simple Markov process based on a random graph theoretic structure is introduced. We propose a polynomial time algorithm for the calculation of a limit state matrix. The algorithm is based on two procedures which will be derived in this contribution. They exploit a distinguished decomposition principle of the underlying graph theoretic structure and the special property of an acyclic directed graph.
Keywords :Random Graph, Markov Process, Limit State Matrix, Polynomial Time Algorithm, Decomposition.

1 Introduction

Let a time-discrete system with finite set of states X be given. Assume that the dynamics of the system is described by a simple Markov process with a given matrix $P = (p_{x,y})$, where $p_{x,y} \geq 0$, $\forall x, y \in X$ and $\sum_{y \in X} p_{x,y} = 1$, $\forall x \in X$. The matrix contains the probability for the states' transitions. We consider the problem of determining a matrix $S = (s_{x,y})$, where an arbitrary element $s_{x,y}$ of this matrix represents the probability that the system will occupy the state y after a large number of transitions (when it starts in the state x).

This problem arises as an auxiliary one in many practical and theoretical decision problems [2, 3, 7].

2 Graphical Interpretation and Main Results

In a first step, we introduce a graphical interpretation of the considered Markov process. We apply the random graph $G = (X, E)$ of states [1, 2, 4], where $e = (x, y) \in E$ if the probability $p_{x,y}$ is strictly positive. It is easy to see that the following lemma holds.

Lemma 1 *A simple Markov process is an ergodic process without transient states if and only if the random graph $G = (X, E)$ is strongly connected.*

It is well-known [2, 7] that for the ergodic process all rows of the matrix (of the limit state probabilities) S are the same, i. e. $\pi_y = s_{x,y}$, $\forall x, y \in X$. The vector π with the components π_y for $y \in X$ can be determined by solving the following system of linear equations: $\pi = \pi P$, $\sum\limits_{y \in X} \pi_y = 1$.

If the Markov process is not ergodic then the random graph G contains several strongly connected components $G^1 = (X^1, E^1)$, $G^2 = (X^2, E^2)$, ..., $G^k = (X^k, E^k)$ where $\bigcup\limits_{i=1}^{k} X^i = X$. Additionally, among these components, there are such strongly connected components $G^{i_r} = (X^{i_r}, E^{i_r})$, $r = 1, 2, \ldots, q$ which do not contain a leaving directed edge $e = (x, y)$ where $x \in X^{i_r}$ and $y \in X \setminus X^{i_r}$. We call such components G^{i_r} *deadlock* components in G.

Lemma 2 *If $G^{i_r} = (X^{i_r}, E^{i_r})$ is a deadlock strongly connected component in G then X^{i_r} is an ergodic class (recurrence chain) of the Markov process; if $x \in X \setminus \bigcup\limits_{r=1}^{q} X^{i_r}$ then x is a transient state of the system in the Markov process.*

3 Algorithmic Approach: An Algorithm for the Calculation of the Matrix

Applying this characterization of the Markov process described above and using the results from [2, 4] we can propose an algorithm for the calculation of the matrix of the limit probabilities S. The algorithm consists of two parts. The first part determines the limit probabilities $s_{x,y}$ for $x \in \bigcup\limits_{r=1}^{q} X^{i_r}$ and $y \in X$.

The second procedure calculates the limit probabilities $s_{x,y}$ for $x \in X \setminus \bigcup\limits_{r=1}^{q} X^{i_r}$ and $y \in X$.

Algorithm for the calculation of the matrix of limit probabilities

Procedure 1:

1. For each ergodic class X^{i_r} we solve the system of linear equations:

$$\pi^{i_r} = \pi^{i_r} P^{i_r}, \quad \sum_{y \in X^{i_r}} \pi_y^{i_r} = 1$$

   where P^{i_r} is the the matrix of probability transitions corresponding to the ergodic class X^{i_r}, i.e. P^{i_r} is a submatrix of P, and π^{i_r} is a vector with the components $\pi_y^{i_r}$ for $y \in X^{i_r}$.

If $\pi_y^{i_r}$ are known then $s_{x,y}$ for $x \in X^{i_r}$ and $y \in X$ can be calculated as follows:

Set $s_{x,y} = \pi_y^{i_r}$, $\forall x, y \in X^{i_r}$, $r = 1, 2, \ldots q$ and $s_{x,y} = 0, \forall x \in X^{i_r}, \forall y \in X \setminus X^{i_r}$, $r = 1, 2, \ldots q$.

Procedure 2:

1. We construct an auxiliary acyclic directed graph $GA = (XA, EA)$ which is obtained from the graph $G = (X, E)$ by using the following transformations:

   We contract each set of vertices X^{i_r} into one vertex z^{i_r} where X^{i_r} is a set of vertices of strongly connected deadlock components $G^{i_r} = (X^{i_r}, E^{i_r})$. If the obtained graph contains parallel directed edges $e^1 = (x, z)$, $e^2 = (x, z)$, $\ldots$, $e^l = (x, z)$ with the corresponding probabilities $p_{x,z}^1$, $p_{x,z}^2$, $\ldots$, $p_{x,z}^l$ then we change them by one directed edge $e = (x, z)$ with the probability $p'_{x,z} = \sum_{i=1}^{l} p_{x,z}^i$; after this transformation to each vertex z_r^i we put in correspondence an directed edge of the form $e = (z^r, z^r)$ with the probability $p'_{z^r,z^r} = 1$.

2. We fix the directed graph $GA = (XA, EA)$ obtained by the construction principle from step 1 where $XA = (X \setminus (\bigcup_{r=1}^{q} X^{i_r})) \cup Z^q$, $Z^q = \{z^1, z^2, \ldots, z^q\}$. In addition we fix the new probability matrix $P' = (p'_{x,y})$ which correspond to this random graph GA.

3. For each $x \in XA$ and every $z^i \in Z^q$ we find the probability $\pi'_x(z^i)$ of the system transaction from the state x to the state z^i. The probabilities $\pi'_x(z^i)$ can be found by solving the following systems of linear equations:

$$P'\pi'(z^1) = \pi'(z^1), \quad \pi'_{z^1}(z^1) = 1, \pi'_{z^2}(z^2) = 0, \ldots, \pi'_{z^p}(z^q) = 0;$$
$$P'\pi'(z^2) = \pi'(z^2), \quad \pi'_{z^1}(z^1) = 0, \pi'_{z^2}(z^2) = 1, \ldots, \pi'_{z^p}(z^q) = 0;$$
$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$
$$P'\pi'(z^q) = \pi'(z^q), \quad \pi'_{z^1}(z^1) = 0, \pi'_{z^2}(z^2) = 0, \ldots, \pi'_{z^p}(z^q) = 1;$$

   where $\pi'(z^i)$ is the vector with components $\pi'_x(z^i)$ for $x \in XA$. So, each vector $\pi'_x(z^i)$ gives probabilities of system transactions from states $x \in XA$ to the ergodic class X^i.

4. We put $s_{x,y} = 0$ for every $x, y \in X \setminus (\bigcup_{r=1}^{q} X^{i_r})$ and $s_{x,y} = \pi'_x(z^r)\pi_y^{i_r}$ for every $x \in X \setminus (\bigcup_{r=1}^{q} X^{i_r})$ and every $y \in X^{i_r}$, $X^{i_r} \subset X$, $r = 1, 2, \ldots, q$.

The algorithm described above represents a modification of the algorithm proposed in [6]:

The algorithm from [6] works on initial graphs and do not use the contraction operation.

In the case when the subgraph $G' = (X \setminus (\bigcup_{r=1}^{q} X^{i_r}), E')$ of G generated by the set of vertices $X \setminus \bigcup_{r=1}^{q} X^{i_r}$ has a structure of an acyclic graph then the Procedure 2 in the algorithm can be exchanged by the following procedure:

Procedure 2′:

1. We make step 1 of Procedure 2 of the algorithm and determine the auxiliary directed graph $GA = (XA, EA)$. Then for every directed edge $e = (z^r, z^r)$ in GA we set $p'_{z^r,z^r} = 0$.
2. We fix the directed graph $GA = (XA, EA)$ obtained according to the construction from step 1, where $XA = (X \setminus (\bigcup_{p=1}^{q} X^{i_p})) \cup Z^r$, Z=$\{z^1, z^2, \ldots, z^p\}$. Then we change the probabilities $p'_{x,y}$ of edges $e = (x, z^p) \in EA$ as follows:

   For every vertex $x \in XA \setminus Z^p$ we find directed edges $e^1 = (x, z^1)$, $e^2 = (x, z^2)$, $\ldots$, $e^r = (x, z^q)$ with associated probabilities p_{x,z^1}, p_{x,z^2}, $\ldots$, p_{x,z^q} and determine the value $Q(x) = \sum_{p=1}^{q)} p_{x,z^i}$; then change p'_{x,y^i} by p''_{x,z^i}, where

$$p''_{x,y^i} = \frac{1}{Q(x)} p'_{x,y^i}, \quad i = 1, 2, \ldots, q.$$

   After that we obtain a new matrix of probability transitions $P'' = (p''_{x,y})$ for GA.

3. For each $x \in XA$ we calculate $P_x(z^i, t)$ by using the following formula

$$P_x(z^i, t+1) = \sum_{y \in XA} P_y(z^i, t) p'_{x,y}, \quad t = 0, 1, 2, \ldots, |XA|,$$

where $P_{z^i}(z^i, 0) = 1, \forall z^i \in Z^q$; then we calculate

$$P_x(z^i) = \sum_{t=1}^{|XA|} P_x(z^i, t), \ \forall x \in X \setminus (\bigcup_{p=1}^{q} X^{i_p}), \ \forall z^i \in Z^q.$$

Here the values $P_x(z^i, t)$ express the probability that the system will occupy the state z^i after t transactions when it start transactions in x; the values $P_x(z^i)$ represents the limiting probability from the state x $X \setminus (\bigcup_{p=1}^{q} X^{i_p})$ to the state $z^i \in Z^q$.

4. We put $s_{x,y} = 0$ for every $x, y \in X \setminus (\bigcup_{p=1}^{q} X^{i_p})$ and $s_{x,y} = P_x(z^r)\pi_y^{i_r}$ for every $x \in X \setminus (\bigcup_{p=1}^{q} X^{i_p})$ and $y \in X^{i_r}$, $r = 1, 2, \ldots q$.

The following theorem holds.

Theorem 1 *The algorithm calculates correctly the matrix of limit probabilities S. The running time of the algorithm is $O(|X|^3)$.*

The proof of the theorem follows from [6].

4 Conclusion

This contribution deals with a special characterization of a simple Markov process based on a random graph theoretic structure. The authors derived a special polynomial time algorithm for the calculation of a limit state matrix. The algorithm is based on two procedures which are elaborated in this contribution. They exploit a distinguished decomposition principle of the underlying graph theoretic structure and the special property of an acyclic directed graph.

This is a new results which extends approaches from [5] and [6]. In [6] the algoritm works on initial graphs and do not use this special contraction operation.

5 Acknowledgements

The authors would like to thank the unknown referees and especially Daniel Dubois for their critical remarks and suggestions which helped to improve the quality of the contribution.

References

[1] Bollobas B., Random Graphs, Second Edition, Cambridge University Press, (2001)

[2] Howard R.A., Dynamic Programming and Markov Processes. Wiley (1960)

[3] Lozovanu D., Pickl S., Optimization and Multiobjective Control of Time-Discrete Systems. Springer Verlag (2009)

[4] Lozovanu D., Pickl S., Dynamic Programming Algorithms for Solving Stochastic Discrete Control Problems. Bulletin of the Academy of Science of RM, Ser. Mathem. 3 (2009)

[5] Lozovanu, D., Pickl, S. Algorithms for Solving Stochastic Discrete Control Problems with Finite Time Horizon. Proceedings of CAO09 IFAC Workshop on Control Applications of Optimisation (2009).

[6] Lozovanu D., Pickl., A Dynamic Programming Approach for Finite Markov Processes and Polynomial Time Algorithms for the Calculation of the Limit State Matrix in Markov Chains. CEJOR, 2010 (Accepted for publication).

[7] Puterman M., Markov Decision Processes. Wiley (1993)