Optimization of Zero-Order Markov Processes with Final Sequence of States

Alexandru Lazari

Moldova State University, MD-2009, Chisinau, Republic of Moldova

*Corresponding Author: lazarialexandru@mail.md

Copyright ©2013 Horizon Research Publishing All rights reserved.

Abstract In this paper the zero-order Markov processes with final sequence of states X and unit transition time are analyzed. The evolution time $T(p)$ of these systems is studied, where p represents the distribution of the states of the system. The problem of minimization the expectation $\mathbb{E}(T(p))$ is considered. This problem is reduced to a geometric program, which is efficiently solved using convex optimization based on interior-point methods. The main idea of the proof is to show that the expression $\mathbb{E}(T(p)) + 1$ is a posynomial function in variables which represent the components of distribution of the states that participate in final sequence of states. For some particular cases the explicit solution is obtained.

Keywords Zero-Order Markov Process, Final Sequence of States, Evolution Time, Geometric Programming, Posynomial Function, Convex Optimization, Interior-Point Methods

1 Introduction

The discrete Markov processes are often used as mathematical models that describe various applied actual problems from many important domains: economy, technique, biology, industry, medicine and others. Based on these stochastic systems, the researchers obtain new numerical methods and algorithms for solving the complex problems of the society.

The Markov stochastic systems were studied in many scientific papers. The main results regarding these systems were described in [7], [8] and [20]. Some applications were presented in [9], [11], [16], [17] and [22].

The stochastic systems with final sequence of states generalize the discrete Markov processes. For these systems the stopping condition is defined, i.e., the system stops when it passes through given final sequence of states. Unlike Markov processes, the evolution time of stochastic systems with final sequence of states is finite and depends on realization of stopping condition. For these systems the problem of determining the main probabilistic characteristics of evolution time of the system is interesting. This problem was studied in [14] and [15],

where polynomial algorithms based on the main properties of homogeneous linear recurrences, generating function and numerical derivation of regular rational fractions were obtained. The main generalizations of these systems were presented in [12], [13] and [15].

In this paper the zero-order Markov processes with final sequence of states and unit transition time are studied. These systems represent a particular case of stochastic systems with final sequence of states. They are also called stochastic systems with final sequence of states and independent states or strong memoryless stochastic systems with final sequence of states, since at every discrete moment of time the state of the system does not depend on previous states.

For these stochastic processes the efficient method for minimizing the expectation of the evolution time is elaborated. This method is based on geometric programming approach, that reduces the problem to the case of convex optimization using interior-point methods.

Geometric programming was introduced in 1967 by Duffin, Peterson, and Zener in [6]. Wilde and Beightler in 1967 and Zener in 1971 contributed with many results (see [24] and [25]) referred of many extensions and sensitivity analysis. A short history of geometric programming was presented by Peterson in [19].

A geometric program represents a type of optimization problem described by objective and constraint functions that have a special form. This form is characterized by following rules: the objective function and left-hand side of inequality constraints need to be posynomials and left-hand side of equality constraints needs to be monomials. Also, in standard form, the geometric program is a minimization problem with right-hand side of all constraints equal to 1 and all inequality constraints containing the " $\leq$ " operator. A good tutorial on geometric programming was presented in [3].

First numerical methods based on solving a sequence of linear programs were elaborated by Avriel et al. [2], Duffin [5] and Rajpogal and Bricker [21]. Nesterov and Nemirovsky in 1994 described the first interior-point method for geometric programs and demonstrated the polynomial time complexity in [18]. Recent numerical approaches were presented by Andersen and Ye [1], Boyd and Vandenberghe [4], Kortanek [10] and Ungureanu [23].

2 Statement of the problem

In this paper a discrete stochastic system L with the set of possible states V , where $|V| = N \leq \infty$, is considered. The state of the system at every discrete moment of time $t = 0, 1, 2, \dots$ is denoted by $v(t) \in V$. The transition time of the system from the state u to another state v at every moment of time t is equal to 1.

On the set V a distribution function $p : V \rightarrow [0, 1]$ is defined, where $p(v)$ represents the probability with which the system L starts its evolution from the state $v \in V$. Also, the transition of the system from arbitrary state $u \in V$ to another state $v \in V$ at every moment of time t is performed with probability $p(v) \in [0, 1]$.

The finishing evolution of the system is conditioned by passing consecutively through fixed sequence of states $X = (x_1, x_2, \dots, x_m) \in V^m$. Let T be the stopping time of the discrete stochastic system L . The value T represents the evolution time of the stochastic system L .

The system L represents a zero-order Markov process with final sequence of states X and distribution of the states p . For this system the moments of positive integer order n of the evolution time are studied in [14] and [15]. In particular case the expectation of evolution time is obtained.

Next, we consider that the distribution p is not fixed. So, we have the zero-order Markov process $L(p)$ with final sequence of states X and distribution of the states p , for every p from the set

$$\mathcal{P} = \{y = (y_1, y_2, \dots, y_N) \in [0, 1]^N \mid \sum_{j=1}^N y_j = 1\}.$$

The problem is to determine the optimal distribution $p^* \in \mathcal{P}$, that minimizes the expectation of the evolution time $T(p)$ of the stochastic system $L(p)$, i.e.

$$T(p^*) = \min_{p \in \mathcal{P}} T(p).$$

In the following chapters we will show how this problem can be solved using geometric programming.

3 Definitions and notations

3.1 Geometric programming

The definition of geometric program requires the definition of monomials and posynomials.

Definition 1. In the context of geometric programming, a monomial is defined as a function $f : \mathbb{R}^s \rightarrow \mathbb{R}$ of the form $f(x_1, x_2, \dots, x_s) = cx_1^{\alpha_1} x_2^{\alpha_2} \dots x_s^{\alpha_s}$, where $c > 0$ and $\alpha_i \in \mathbb{R}$, $i = \overline{1, s}$.

Definition 2. A posynomial is defined as a linear combination of monomials, i.e. represents a function of the form $f(x_1, x_2, \dots, x_s) = \sum_{k=1}^K c_k x_1^{\alpha_{1k}} x_2^{\alpha_{2k}} \dots x_s^{\alpha_{sk}}$, where $c_k > 0$, $k = \overline{1, K}$ and $\alpha_{ik} \in \mathbb{R}$, $i = \overline{1, s}$, $k = \overline{1, K}$.

Posynomials are closed under addition, multiplication, and nonnegative scaling. Posynomials are not the same as polynomials, since:

- a polynomial's exponents must be non-negative integers, but a posynomial's exponents can be arbitrary real numbers;
- a polynomial's coefficients can be arbitrary real numbers, but a posynomial's coefficients must be positive real numbers.

Using these notions, we can define a geometric program in the following way.

Definition 3. A geometric program is an optimization problem of the form:

$$f_0(x_1, x_2, \dots, x_s) \rightarrow \min,$$

subject to

$$\begin{cases} f_i(x_1, x_2, \dots, x_s) \leq 1, & i = \overline{1, r} \\ h_j(x_1, x_2, \dots, x_s) = 1, & j = \overline{1, l} \\ x_k > 0, & k = \overline{1, s}, \end{cases}$$

where $f_i(x_1, x_2, \dots, x_s)$, $i = \overline{0, r}$, are posynomials and $h_j(x_1, x_2, \dots, x_s)$, $j = \overline{1, l}$, are monomials.

In order to efficiently solve a geometric program we need to convert it to a convex optimization problem. Efficient solution methods for general convex optimization problems were developed by Boyd and Vandenberghe in [4]. The conversion is based on a logarithmic change of variables $y_k = \ln x_k$, $k = \overline{1, s}$ and a logarithmic transformation of the objective and constraint functions. The obtained convex optimization problem has the form:

$$\ln f_0(e^{y_1}, e^{y_2}, \dots, e^{y_s}) \rightarrow \min,$$

subject to

$$\begin{cases} \ln f_i(e^{y_1}, e^{y_2}, \dots, e^{y_s}) \leq 0, & i = \overline{1, r} \\ \ln h_j(e^{y_1}, e^{y_2}, \dots, e^{y_s}) = 0, & j = \overline{1, l}. \end{cases}$$

Unlike the original problem, the obtained problem looks more complicated, but it is convex and can be solved very efficiently using standard interior-point methods (see [3] and [4]). Also, the interior-point methods for solving geometric programs are very robust, require no starting point or other parameters and always find the globally optimal solution of the problem.

3.2 Homogeneous linear recurrences

Next, we remind some definitions and notations from [12], [14] and [15] regarding homogeneous linear recurrences. We consider a subfield K of the field $\mathbb{C}$.

Definition 4. The sequence $a = \{a_n\}_{n=0}^{\infty} \subseteq \mathbb{C}$ is called non-degenerated homogeneous linear m -recurrent sequence on the set K if there exists the vector $q = (q_k)_{k=0}^{m-1} \in K^m$ such that $q_{m-1} \neq 0$ and $a_n = \sum_{k=0}^{m-1} q_k a_{n-1-k}$, for all $n \geq m$.

Definition 5. The vector q from previous definition represents the generating vector and the vector $I_m^{[a]} = (a_n)_{n=0}^{m-1}$ is called initial state of the sequence a .

We introduced the following notations:

- $Rol[K][m]$ is the set of non-degenerated homogeneous linear m -recurrent sequences on the set K ;
- $G[K][m](a)$ represents the set of generating vectors of length m of the sequence $a \in Rol[K][m]$.

Definition 6. The function $G^{[a]}(z) = \sum_{n=0}^{\infty} a_n z^n$ is called generating function of the sequence $a = (a_n)_{n=0}^{\infty} \subseteq \mathbb{C}$ and the function $G_t^{[a]}(z) = \sum_{n=0}^{t-1} a_n z^n$ is called partial generating function of order t of the sequence a .

Definition 7. Let $a \in Rol[K][m]$ and $q \in G[K][m](a)$. For the sequence a we will consider the unitary characteristic polynomial $H_m^{[q]}(z) = 1 - zG_m^{[q]}(z)$. For an arbitrary $\alpha \in K^*$ the polynomial $H_{m,\alpha}^{[q]}(z) = \alpha H_m^{[q]}(z)$ is called characteristic polynomial of the sequence a .

Also, the following notation is introduced:

- $H[K][m](a)$ is the set of characteristic polynomials of degree m of the sequence $a \in Rol[K][m]$.

The next theorem presents the formula for the generating function.

Theorem 1. If $a \in Rol[K][m]$ and $q \in G[K][m](a)$, then $G^{[a]}(z) = \frac{G_m^{[a]}(z) - \sum_{k=0}^{m-1} q_k z^{k+1} G_{m-1-k}^{[a]}(z)}{H_m^{[q]}(z)}$.

4 Preliminary results

4.1 Distribution of evolution time

The zero-order Markov processes with final sequence of states were described in Section 2 and the evolution time $T(p)$ of the system was defined. Let us consider the distribution $a = (\mathbb{P}(T(p) = n))_{n=0}^{\infty}$.

In [14] and [15], the properties of sequence a were studied. We obtained that $a \in Rol[\mathbb{R}][m]$ for $p(x_1) \neq 1$ and $p(x_j) \neq 0$, $j = \overline{1, m}$. Also, we obtained formulas for the initial state $I_m^{[a]}$ and the minimal generating vector $q \in G[\mathbb{R}][m](a)$. The established algorithm was described in the following way.

Algorithm 1.

Input: $X = (x_j)_{j=1}^m \in V^m$, $\pi_j = p(x_j)$, $j = \overline{1, m}$.

Output: $I_m^{[a]}$, $q \in G[\mathbb{R}][m](a)$.

1. The values $w_k = \prod_{j=1}^k \pi_j$, $k = \overline{1, m}$, are determined.
2. The initial state $I_m^{[a]} = (0, 0, \dots, 0, w_m)$ and the values $u_{1,0} = 1 - \pi_1$ and $v_{1,0} = 1$ are calculated.
3. For each $s = \overline{2, m}$ the following steps are executed:
 - (a) The values

$$t(s) = \min(\{t \in \{2, 3, \dots, s+1\} \mid x_{t-1+j} = x_j, j = \overline{1, s+1-t}\}),$$

are obtained.

(b) The parameters

$$v_{s,k} = 0, k = \overline{0, t(s)-3}; v_{s,t(s)-2} = \frac{w_{t(s)-1}}{\pi_1};$$

$$v_{s,k} = -\frac{w_{t(s)-1}}{\pi_1} \sum_{j=k-t(s)+2}^{s-t(s)+1} u_{j,k-t(s)+1},$$

$$k = \overline{t(s)-1, s-1}, \text{ are calculated.}$$

(c) The quantities

$$u_{s,k} = \pi_1(v_{s-1,k} - v_{s,k}), k = \overline{0, s-2};$$

$$u_{s,s-1} = -\pi_1 v_{s,s-1}, \text{ are determined.}$$

4. The components $q_0 = 1 - \pi_1 v_{m,0}$, $q_k = -\pi_1 v_{m,k}$, $k = \overline{1, m-1}$, of generating vector $q \in G[\mathbb{R}][m](a)$, are calculated.

In the case when exists $j \in \{1, 2, \dots, m\}$ such that $p(x_j) = 0$, the evolution time is infinite. Also, the case $p(x_k) = 1$, $k = \overline{1, m}$, is trivial. In this case the evolution time is equal to $m-1$, i.e., $\mathbb{P}(T(p) = m-1) = 1$ and $\mathbb{P}(T(p) = n) = 0$, for all $n \in \mathbb{N} \setminus \{m-1\}$.

4.2 Simplified version of Algorithm 1

Next, we eliminate the parameters

$$u_{sk}, s = \overline{1, m}, k = \overline{0, s-1},$$

from Algorithm 1. The following theorem holds.

Theorem 2. If one assumes the notations from Section 4.1, then the formula

$$v_{sk} = \begin{cases} 0, & 0 \leq k \leq t(s)-3 \\ \frac{w_{t(s)-1}}{\pi_1}, & k = t(s)-2 \\ -\frac{w_{t(s)-1}}{\pi_1}(1 - \pi_1), & t(s-t(s)+1) = 2 \\ & \text{and } k = t(s)-1 \\ -\frac{w_{t(s)-1}}{\pi_1}, & t(s-t(s)+1) \geq 3 \\ & \text{and } k = t(s)-1 \\ w_{t(s)-1} v_{s-t(s)+1, k-t(s)+1}, & t(s) \leq k \leq s-1 \end{cases}$$

is true.

Proof. The particular case $k \leq t(s)-2$ is trivial from Algorithm 1. Next, we consider the case $k \geq t(s)$. We have:

$$\begin{aligned} v_{sk} &= -\frac{w_{t(s)-1}}{\pi_1} \sum_{j=k-t(s)+2}^{s-t(s)+1} u_{j,k-t(s)+1} \\ &= -\frac{w_{t(s)-1}}{\pi_1} \left(u_{k-t(s)+2, k-t(s)+1} + \sum_{j=k-t(s)+3}^{s-t(s)+1} u_{j,k-t(s)+1} \right) \\ &= -\frac{w_{t(s)-1}}{\pi_1} \left(-\pi_1 v_{k-t(s)+2, k-t(s)+1} + \sum_{j=k-t(s)+3}^{s-t(s)+1} \pi_1 (v_{j-1, k-t(s)+1} - v_{j, k-t(s)+1}) \right) \\ &= -\frac{w_{t(s)-1}}{\pi_1} \left(-\pi_1 v_{k-t(s)+2, k-t(s)+1} + \pi_1 (v_{k-t(s)+2, k-t(s)+1} - v_{s-t(s)+1, k-t(s)+1}) \right) \\ &= w_{t(s)-1} v_{s-t(s)+1, k-t(s)+1}. \end{aligned}$$

In the case when $k = t(s) - 1$, we obtain

$$\begin{aligned} v_{sk} &= -\frac{w_{t(s)-1}}{\pi_1} \sum_{j=1}^{s-t(s)+1} u_{j0} \\ &= -\frac{w_{t(s)-1}}{\pi_1} \left(u_{1,0} + \sum_{j=2}^{s-t(s)+1} u_{j0} \right) \\ &= -\frac{w_{t(s)-1}}{\pi_1} \left(1 - \pi_1 + \sum_{j=2}^{s-t(s)+1} \pi_1 (v_{j-1,0} - v_{j0}) \right) \\ &= -\frac{w_{t(s)-1}}{\pi_1} (1 - \pi_1 + \pi_1 (v_{1,0} - v_{s-t(s)+1,0})) \\ &= -\frac{w_{t(s)-1}}{\pi_1} (1 - \pi_1 v_{s-t(s)+1,0}) . \end{aligned}$$

If $t(s - t(s) + 1) = 2$, then

$$v_{s-t(s)+1,0} = \frac{w_{t(s-t(s)+1)-1}}{\pi_1} = \frac{w_1}{\pi_1} = 1,$$

that implies the formula

$$v_{s,t(s)-1} = -\frac{w_{t(s)-1}}{\pi_1} (1 - \pi_1) .$$

But, if $t(s - t(s) + 1) \geq 3$, then $v_{s-t(s)+1,0} = 0$ and we have the relation

$$v_{s,t(s)-1} = -\frac{w_{t(s)-1}}{\pi_1} .$$

In this way we obtained the assertion. $\square$

Using Theorem 2, Algorithm 1 can be rewritten in the following form.

Algorithm 2.

Input: $X = (x_j)_{j=1}^m \in V^m$, $\pi_j = p(x_j)$, $j = \overline{1, m}$.

Output: $I_m^{[a]}$, $q \in G[\mathbb{R}][m](a)$.

1. The values $w_k = \prod_{j=1}^k \pi_j$, $k = \overline{1, m}$, are determined.
2. The initial state $I_m^{[a]} = (0, 0, \dots, 0, w_m)$ is obtained.
3. For each $s = \overline{1, m}$ the following steps are executed:

(a) The values

$$t(s) = \min(\{t \in \{2, 3, \dots, s+1\} \mid x_{t-1+j} = x_j, j = \overline{1, s+1-t}\}),$$

are calculated.

(b) Using Theorem 2, the parameters $v_{s,k}$, $k = \overline{0, s-1}$, are obtained.

4. The components $q_0 = 1 - \pi_1 v_{m,0}$, $q_k = -\pi_1 v_{m,k}$, $k = \overline{1, m-1}$, of generating vector $q \in G[\mathbb{R}][m](a)$, are calculated.

4.3 Expectation of evolution time

The following theorem holds.

Theorem 3. *The expectation of the evolution time $T(p)$ of zero-order Markov process $L(p)$ can be determined using the formula*

$$\mathbb{E}(T(p)) = -1 + (m + w_m^{-1}) + \frac{1}{w_m} \sum_{k=0}^{m-1} (k+1) z_{mk},$$

where $z_{sk} = -\pi_1 v_{sk}$, $s = \overline{1, m}$, $k = \overline{0, s-1}$.

Proof. From Theorem 1 and Algorithm 2, we obtain the formula for the generating function:

$$G^{[a]}(z) = \frac{w_m z^{m-1}}{H_m^{[q]}(z)}.$$

Since $G^{[a]}(1) = \sum_{n=0}^{\infty} \mathbb{P}(T(p) = n) = 1$, we have the relation $H_m^{[q]}(1) = w_m$.

The derivative of generating function is

$$\frac{\partial G^{[a]}(z)}{\partial z} = \frac{(m-1)w_m z^{m-2} H_m^{[q]}(z) - w_m z^{m-1} \frac{\partial H_m^{[q]}(z)}{\partial z}}{(H_m^{[q]}(z))^2},$$

which implies

$$\begin{aligned} \mathbb{E}(T(p)) &= \left. \frac{\partial G^{[a]}(z)}{\partial z} \right|_{z=1} = (m-1) - \frac{1}{w_m} \left. \frac{\partial H_m^{[q]}(z)}{\partial z} \right|_{z=1} \\ &= (m-1) - \frac{1}{w_m} \left. \frac{\partial \left(1 - \sum_{k=0}^{m-1} q_k z^{k+1} \right)}{\partial z} \right|_{z=1} \\ &= (m-1) + \frac{1}{w_m} \sum_{k=0}^{m-1} (k+1) q_k . \end{aligned}$$

Since from Algorithm 2 we have $q_0 = 1 - \pi_1 v_{m,0}$ and $q_k = -\pi_1 v_{m,k}$, $k = \overline{1, m-1}$, we obtain

$$\begin{aligned} \mathbb{E}(T(p)) &= (m-1) + \frac{1}{w_m} \left(1 + \sum_{k=0}^{m-1} (k+1) (-\pi_1 v_{m,k}) \right) \\ &= -1 + (m + w_m^{-1}) + \frac{1}{w_m} \sum_{k=0}^{m-1} (k+1) z_{mk} . \end{aligned}$$

The proof is complete. $\square$

5 Main results

5.1 General case of the problem

In this subsection we present the main results. We prove that the problem of optimization the expectation of evolution time can be reduced to a geometric program.

Lemma 1. *The sequence $(t(s))_{s=1}^m$ represents a monotonically increasing sequence that verifies the relations $t(1) = 2$ and $2 \leq t(s) \leq s+1$, $s = \overline{1, m}$.*

Proof. The relations $t(1) = 2$ and $2 \leq t(s) \leq s + 1$, $s = \overline{1, m}$, are easily obtained from definition of the coefficients $t(s)$, $s = \overline{1, m}$. Also, we have

$$(x_{t(s)}, x_{t(s)+1}, \dots, x_s) = (x_1, x_2, \dots, x_{s+1-t(s)})$$

and

$$(x_{t(s+1)}, x_{t(s+1)+1}, \dots, x_{s+1}) = (x_1, x_2, \dots, x_{s+2-t(s+1)}),$$

for all $1 \leq s < m$. The last relation implies

$$(x_{t(s+1)}, x_{t(s+1)+1}, \dots, x_s) = (x_1, x_2, \dots, x_{s+1-t(s+1)}),$$

for all $1 \leq s < m$. Since the value $t = t(s)$ represents the least positive integer number that verifies the relation

$$(x_t, x_{t+1}, \dots, x_s) = (x_1, x_2, \dots, x_{s+1-t}),$$

we have $t(s) \leq t(s+1)$, for all $1 \leq s < m$. So, the sequence $(t(s))_{s=1}^m$ represents a monotonically increasing sequence. $\square$

Lemma 2. *If $t(s) = 2$ then $w_s = \pi_1^s$.*

Proof. Let be $t(s) = 2$. Then, from the definition, we have $x_1 = x_2 = \dots = x_s$, that implies $\pi_1 = \pi_2 = \dots = \pi_s$ and, finally, we obtain $w_s = \pi_1^s$. $\square$

Lemma 3. *If $t(s) = 2$ and $k \geq 1$, then $z_{s,k} = \pi_1^k(1 - \pi_1)$.*

Proof. Let be $t(s) = 2$ and $k \geq 2$. Then,

$$z_{s,k} = w_{t(s)-1} z_{s-t(s)+1, k-t(s)+1} = \pi_1 z_{s-1, k-1}.$$

Applying Lemma 1, we have $2 \leq t(s-1) \leq t(s) = 2$, that implies $t(s-1) = 2$, i.e., if $k-1 \geq 2$, thus,

$$z_{s,k} = \pi_1 z_{s-1, k-1} = \pi_1^2 z_{s-2, k-2}.$$

Repeating the calculus for $(k-1)$ times, we obtain

$$z_{s,k} = \pi_1^{k-1} z_{s-(k-1), 1}.$$

Since $t(s - (k-1)) = 2$, we have

$$z_{s,k} = \pi_1^{k-1} z_{s-(k-1), 1} = \pi_1^k(1 - \pi_1)$$

and the assumption is proved. We mention that the proof is also valid for $k = 1$. $\square$

Lemma 4. *The relation*

$$w_s = w_{t(s)-1} w_{s+1-t(s)}, \quad s = \overline{1, m},$$

holds.

Proof. From the definition of overlapping levels $t(s)$, $s = \overline{1, m}$, we have

$$(x_1, x_2, \dots, x_{s+1-t(s)}) = (x_{t(s)}, x_{t(s)+1}, \dots, x_s),$$

that implies

$$(\pi_1, \pi_2, \dots, \pi_{s+1-t(s)}) = (\pi_{t(s)}, \pi_{t(s)+1}, \dots, \pi_s)$$

for every integer number s from the interval $[1, m]$.

Then, we obtain

$$\begin{aligned} w_{t(s)-1} &= \prod_{j=1}^{t(s)-1} \pi_j = \frac{w_s}{\prod_{j=t(s)}^s \pi_j} \\ &= \frac{w_s}{\prod_{j=1}^{s+1-t(s)} \pi_j} = \frac{w_s}{w_{s+1-t(s)}}, \quad s = \overline{1, m}, \end{aligned}$$

which represents the assertion of the lemma. $\square$

Lemma 5. *For each $k = \overline{0, m-1}$ there exist the positive integer values m^* and k^* that verify the relations*

$$\begin{cases} z_{mk} = \frac{w_m}{w_{m^*}} z_{m^*k^*} \\ m^* - k^* = m - k \\ 0 \leq k^* < t(m^*) \end{cases}$$

Proof. This Lemma is obtained from Theorem 2 and Lemma 4. Applying Theorem 2, we obtain that there exists the index l such that

$$z_{mk} = w_{t(m)-1} w_{t(m-t(m)+1)-1} \dots w_{t(l)-1} z_{m^*k^*}.$$

Next, using Lemma 4, we have

$$\begin{aligned} z_{mk} &= \frac{w_m}{w_{m+1-t(m)}} \cdot \frac{w_{m+1-t(m)}}{w_{(m-t(m)+1)+1-t(m-t(m)+1)}} \cdot \dots \cdot \frac{w_l}{w_{m^*}} z_{m^*k^*} = \frac{w_m}{w_{m^*}} z_{m^*k^*}. \end{aligned}$$

Also, from Theorem 2, it is easy to see that the difference between indexes is not changed at every step of recurrence, i.e., the relation $m^* - k^* = m - k$ holds. The condition $0 \leq k^* < t(m^*)$ represents the stopping rule of the recurrent formula from Theorem 2. $\square$

Theorem 4. *The expression $\mathbb{E}(T(p)) + 1$ represents a posynomial in the variables $\pi_1, \pi_2, \dots, \pi_m$.*

Proof. From Theorem 2 and Theorem 3 we have

$$z_{s,k} = \begin{cases} 0, & 0 \leq k \leq t(s) - 3 \\ -w_{t(s)-1}, & k = t(s) - 2 \\ w_{t(s)-1}(1 - \pi_1), & t(s - t(s) + 1) = 2 \\ & \text{and } k = t(s) - 1 \\ w_{t(s)-1}, & t(s - t(s) + 1) \geq 3 \\ & \text{and } k = t(s) - 1 \\ w_{t(s)-1} z_{s-t(s)+1, k-t(s)+1}, & t(s) \leq k \leq s - 1. \end{cases}$$

Let be $z_{mk} = \frac{w_m}{w_{m^*}} z_{m^*k^*}$, where the indexes m^* and k^* were defined in Lemma 5.

The following scenarios are possible:

- The case $k^* \leq t(m^*) - 3$.

In this case we have $z_{mk} = \frac{w_m}{w_{m^*}} z_{m^*k^*} = 0$.

- The case $k^* = t(m^*) - 2$, $t(m^*) \leq m^*$ and $t(m^* - t(m^*) + 1) \geq 3$.

In this case we have $k^* + 1 = t(m^*) - 1$, that implies $z_{m^*k^*} = -w_{t(m^*)-1}$ and $z_{m^*,k^*+1} = w_{t(m^*)-1}$. We obtain

$$\begin{aligned} f_{mk} &\stackrel{def}{=} (k+1)z_{mk} + (k+2)z_{m,k+1} \\ &= \frac{w_m}{w_{m^*}} ((k+1)z_{m^*k^*} + (k+2)z_{m^*,k^*+1}) \\ &= \frac{w_m}{w_{m^*}} ((k+1)(-w_{t(m^*)-1}) + (k+2)w_{t(m^*)-1}) \\ &= \frac{w_m}{w_{m^*}} w_{t(m^*)-1} = \frac{w_m}{w_{m^*+1-t(m^*)}}. \end{aligned}$$

It is easy to see that this expression represents a posynomial in the variables $\pi_1, \pi_2, \dots, \pi_m$.

- The case $k^* = t(m^*) - 2$, $t(m^*) \leq m^*$ and $t(m^*) - t(m^*) + 1 = 2$.
In this case we have $k^* + 1 = t(m^*) - 1$, that implies $z_{m^*k^*} = -w_{t(m^*)-1}$ and

$$z_{m^*,k^*+1} = w_{t(m^*)-1}(1 - \pi_1).$$

Let be $r \geq 2$. Using Lemma 3, we obtain:

$$\begin{aligned} z_{m^*,k^*+r} &= w_{t(m^*)-1} z_{m^*-t(m^*)+1,k^*+r-t(m^*)+1} \\ &= w_{t(m^*)-1} z_{m^*-t(m^*)+1,r-1} \\ &= w_{t(m^*)-1} \pi_1^{r-1} (1 - \pi_1). \end{aligned}$$

Next, applying Lemma 2, Lemma 4 and Lemma 5, we have the relation

$$\begin{aligned} g_{mk} &\stackrel{\text{def}}{=} \sum_{j=k}^{m-1} (j+1) z_{mj} = \sum_{j=k}^{m-1} (j+1) z_{m,k+(j-k)} \\ &= \frac{w_m}{w_{m^*}} \sum_{j=k}^{m-1} (j+1) z_{m^*,k^*+(j-k)} \\ &= \frac{w_m}{w_{m^*}} \left[(k+1)(-w_{t(m^*)-1}) \right. \\ &\quad \left. + \sum_{j=k+1}^{m-1} (j+1) w_{t(m^*)-1} \pi_1^{j-k-1} (1 - \pi_1) \right] \\ &= \frac{w_m}{w_{m^*+1-t(m^*)}} [-(k+1) \\ &\quad + \sum_{j=0}^{m-k-2} (j+k+2) \pi_1^j (1 - \pi_1)] \\ &= \frac{w_m}{\pi_1^{m^*-t(m^*)+1}} (1 - m \pi_1^{m-k-1}) \\ &= \frac{w_m}{\pi_1^{(m^*-k^*)+(k^*-t(m^*)+2)-1}} (1 - m \pi_1^{m-k-1}) \\ &= \frac{w_m}{\pi_1^{m-k-1}} (1 - m \pi_1^{m-k-1}) \\ &= \frac{w_m}{\pi_1^{m-k-1}} - m w_m, \end{aligned}$$

that implies the formula

$$m w_m + g_{mk} = \frac{w_m}{\pi_1^{m-k-1}}.$$

In this way we obtain that the expression $m w_m + g_{mk}$ represents a posynomial in the variables $\pi_1, \pi_2, \dots, \pi_m$.

- The case $k^* = m^* - 1$ and $t(m^*) = m^* + 1$.
In this case we have

$$\begin{aligned} z_{mk} &= z_{m,m-1} = \frac{w_m}{w_{m^*}} z_{m^*,m^*-1} = \\ &= \frac{w_m}{w_{m^*}} (-w_{t(m^*)-1}) = \frac{w_m}{w_{m^*}} (-w_{m^*}) = -w_m, \end{aligned}$$

that implies

$$m w_m + m z_{m,k} = m w_m - m w_m = 0.$$

- The case $k^* = t(m^*) - 1$.
Since $k^* - 1 = t(m^*) - 2$, this case is contained in previous studied scenarios.

Consequently, we obtain that the expression $m w_m + \sum_{k=0}^{m-1} (k+1) z_{mk}$ represents a posynomial in the variables $\pi_1, \pi_2, \dots, \pi_m$. Applying Theorem 3, we have

$$\mathbb{E}(T(p)) + 1 = \frac{1 + \left[m w_m + \sum_{k=0}^{m-1} (k+1) z_{mk} \right]}{w_m}.$$

From the properties of posynomials, we obtain that the expression $\mathbb{E}(T(p)) + 1$ represents a posynomial in the variables $\pi_1, \pi_2, \dots, \pi_m$. $\square$

Theorem 4 is very important. It shows how the problem described in Section 2 can be reduced to the geometric program

$$\mathbb{E}(T(p)) + 1 \rightarrow \min,$$

subject to

$$\begin{cases} \sum_{x \in Y} p(x) \leq 1 \\ p(x) > 0, \forall x \in Y, \end{cases}$$

where $Y = \{x_1, x_2, \dots, x_m\}$. If $\pi^* = (\pi^*(x))_{x \in Y}$ represents the optimal solution of this geometric program, then $p^* = (p^*(x))_{x \in V}$ represents the optimal solution of the initial problem, where

$$\begin{cases} p^*(x) = \pi^*(x), & x \in Y \\ p^*(x) = 0, & x \in V \setminus Y. \end{cases}$$

In obtained geometric program we actually have $\sum_{x \in Y} p(x) = 1$. Indeed:

- The expression $\sum_{x \in Y} p(x)$ represents a posynomial, but is not a monomial in variables $p(x), x \in Y$.
- These constraints are equivalent for the optimal solution, since $p^*(x) = 0, \forall x \in V \setminus Y$ and $\sum_{x \in V} p^*(x) = 1$.

We have to mention that the relation $p^*(x) = 0, \forall x \in V \setminus Y$, is true from the statement of the initial problem. Indeed, if we have $p^*(x^*) > 0$ for at least one state $x^* \in V \setminus Y$, then the states that are components of final sequence of states have a lower probability of realization and, in consequence, the evolution time is greater and the solution is not optimal.

5.2 Optimal solutions for particular cases

In this subsection some particular cases of the problem are presented. The explicit optimal solutions are obtained. The following theorems hold.

Theorem 5. If $t(m) = 2$, then the optimal solution is $p^* = (p^*(x))_{x \in V}$, where $p^*(x_1) = 1$ and $p^*(y) = 0$, for all $y \in V \setminus \{x_1\}$, and the minimal value of the expectation of evolution time is $\mathbb{E}(T(p^*)) = m - 1$.

Proof. Let be $t(m) = 2$. We have

$$x_1 = x_2 = \dots = x_m$$

and the stochastic system $L(p)$ represents a stochastic system with final critical state x_1 . We obtain that

$$Y = \{x_1, x_2, \dots, x_m\} = \{x_1\}.$$

So, the optimal solution is $p^* = (p^*(x))_{x \in V}$, where $p^*(x_1) = 1$ and $p^*(y) = 0$, for all $y \in V \setminus \{x_1\}$. It is easy to observe that this fact implies the relation $\mathbb{E}(T(p^*)) = m - 1$. $\square$

Theorem 6. *If $t(m) = m + 1$, then the components $p^*(y)$, $y \in V$, of the optimal solution p^* are direct proportionally with the multiplicities $m(y)$, $y \in V$, of the respective states in final sequence of states X and the minimal value of the expectation of evolution time is*

$$\mathbb{E}(T(p^*)) = -1 + \prod_{y \in Y} \left(\frac{m}{m(y)} \right)^{m(y)}.$$

Proof. Let be $t(m) = m + 1$. We have

$$z_{mk} = \begin{cases} 0, & 0 \leq k \leq m - 2 \\ -w_m, & k = m - 1 \end{cases}.$$

From Theorem 3, the formula for the expectation of evolution time is $\mathbb{E}(T(p)) = -1 + w_m^{-1}$. So, we have the optimization problem

$$\mathbb{E}(T(p)) + 1 = w_m^{-1} \rightarrow \min,$$

subject to $\sum_{x \in Y} p(x) = 1$ and $p(x) > 0$, $\forall x \in Y$,

which is equivalent with the optimization problem

$$w_m \rightarrow \max,$$

subject to $\sum_{x \in Y} p(x) = 1$ and $p(x) > 0$, $\forall x \in Y$.

For each state $x \in Y$ we denote by $m(x)$ its multiplicity in final sequence of states X . We obtain the optimization problem

$$f((p(x))_{x \in Y}) = \prod_{x \in Y} (p(x))^{m(x)} \rightarrow \max,$$

subject to $\sum_{x \in Y} p(x) = 1$ and $p(x) > 0$, $\forall x \in Y$.

Next, we apply the classical optimization method involving the partial derivatives of the objective function. We chose a state $y^* \in Y$ and we denote by Y^* the set $Y \setminus \{y^*\}$. We have $p(y^*) = 1 - \sum_{x \in Y^*} p(x)$, that implies

$$f((p(x))_{x \in Y^*}) = (p(y^*))^{m(y^*)} \prod_{x \in Y^*} (p(x))^{m(x)}.$$

For each $y \in Y^*$, we obtain

$$\frac{\partial f}{\partial p(y)} = \prod_{x \in Y^* \setminus \{y\}} (p(x))^{m(x)} \cdot \frac{\partial h(y, y^*)}{\partial p(y)},$$

where $h(y, y^*) = (p(y))^{m(y)} (p(y^*))^{m(y^*)}$. We have

$$\begin{aligned} \frac{\partial h(y, y^*)}{\partial p(y)} &= m(y) (p(y))^{m(y)-1} (p(y^*))^{m(y^*)} \\ &\quad - m(y^*) (p(y))^{m(y)} (p(y^*))^{m(y^*)-1} \\ &= (p(y))^{m(y)-1} (p(y^*))^{m(y^*)-1} \\ &\quad \cdot (m(y)p(y^*) - m(y^*)p(y)). \end{aligned}$$

So, the system

$$\frac{\partial f}{\partial p(y)} = 0, \quad \forall y \in Y^*,$$

is equivalent with the system

$$m(y)p(y^*) - m(y^*)p(y) = 0, \quad \forall y \in Y^*,$$

which implies the relation

$$p(y) = \frac{m(y)}{m(y^*)} p(y^*), \quad \forall y \in Y^*.$$

We obtain the formula

$$1 = \sum_{y \in Y} \frac{m(y)}{m(y^*)} p(y^*) = \frac{p(y^*)}{m(y^*)} \sum_{y \in Y} m(y) = \frac{mp(y^*)}{m(y^*)},$$

from which we have $p(y^*) = \frac{m(y^*)}{m}$. Next, we obtain the components of the optimal solution of the problem:

$$p^*(y) = p(y) = \frac{m(y)}{m}, \quad \forall y \in Y.$$

For $\mathbb{E}(T(p^*))$, we have the formula

$$\mathbb{E}(T(p^*)) = -1 + w_m^{-1} = -1 + \prod_{y \in Y} \left(\frac{m}{m(y)} \right)^{m(y)},$$

which represents the assertion of the theorem. $\square$

6 Practical implementation

6.1 Expectation of the evolution time

For implementing the established method, we used the Wolfram Mathematica[®] package.

Initially we need to set the input parameters of the method: the sequence *Var* of variables that represents the distribution of the states from final sequence of states, the sequence *Xind* of indexes (in sequence *Var*) of the variables from final sequence of states and, optionally, the approximation *eps* of the zero value in numerical calculus for selecting real solutions when is applied the classical optimization method. For example, if the final sequence of states of the stochastic system is $X = (x_1, x_1, x_2, x_2, x_1, x_1)$ and the approximation of the zero value is $\varepsilon = 10^{-4}$, then we need to run the following instructions in Wolfram Mathematica[®]:

```
Clear[p1, p2]; Var = {p1, p2};
XInd = {1, 1, 2, 2, 1, 1}; eps = 10^(-4);
```

Next, in order to obtain the expectation of the evolution time based on the established method, we need to run the following sequence of instructions:

```
m = Length[XInd];
X = Table[Var[[XInd[[i]]]], {i, 1, m}];
w = Table[1, {i, 1, m}];
For[i = 1, i <= m, i++,
  For[j = 1, j <= i, j++, w[[i]] = w[[i]] * X[[j]]];
t = Table[s + 1, s, 1, m];
For[s = 2, s <= m, s++,
```

```

For[t[[s]] = 2, t[[s]] <= s, t[[s]] ++,
  For[j = 1, j <= s + 1 - t[[s]], j ++,
    If[XInd[t[[s]] - 1 + j] != XInd[j], Break[]];
    If[j > s + 1 - t[[s]], Break[]];
z = Function[{s, k},
  If[k <= t[[s]] - 3, 0,
  If[k == t[[s]] - 2, -w[t[[s]] - 1],
  If[k == t[[s]] - 1,
    If[t[[s] - t[[s]] + 1] == 2,
      w[t[[s]] - 1] * (1 - X[[1]]),
      w[t[[s]] - 1]],
  w[t[[s]] - 1] * z[s - t[[s]] + 1, k - t[[s]] + 1]]];
"The expectation function is : "
DG1 = FullSimplify[-1 + m + (1/w[[m]]) *
  (1 + Sum[(k + 1) * z[m, k], {k, 0, m - 1}])]

```

6.2 Optimal distribution of the evolution time

Having the formula obtained in previous subsection, we can apply the geometric programming approach or other methods for optimizing the evolution time of zero-order Markov process.

Next, we present two alternative versions using Wolfram Mathematica® package. These versions are easier to implement than geometric programming, but they are less effective, since Wolfram Mathematica® package uses general methods for solving nonlinear optimal problems. Additionally, we mention that the second version works without warnings and errors only when the system of nonlinear equations containing partial derivatives of the expectation of evolution time has a finite non-empty set of admissible solutions.

First implementation is based on integrated *Minimize* function. For the example analyzed in previous subsection, we need to run the following instructions:

```

Clear[p1, p2]; Minimize[{DG1,
  {p1 >= 0, p2 >= 0, p1 + p2 <= 1}}, {p1, p2}]/N

```

The second implementation is based on the classical optimization method. First we need to express an arbitrary variable by others from the relation $\sum_{p \in Var} p = 1$.

For the studied example, we have the instruction:

```
p1 = 1 - p2;
```

Next, we need to run the following instructions:

```

XVar = Variables[DG1]; mVar = Length[XVar];
y = Simplify[
  Table[D[DG1, XVar[[i]]] == 0, {i, 1, mVar}]];
r = N[Solve[y, XVar]];
rCond = Function[ri,
  For[i = 1, i <= Length[ri], i ++,
    If[Im[ri[[i, 2]]] >= eps || Re[ri[[i, 2]]] <= 0 ||
      Re[ri[[i, 2]]] >= 1, Break[]];
  i > mVar];
critPoints = Select[r, rCond];
critPointsRe = Table[Table[
  critPoints[[i]][[j]][[1]] -> Re[critPoints[[i]][[j]][[2]],
  {j, 1, Length[critPoints[[i]]]}],
  {i, 1, Length[critPoints]}];
critPointsValRe = DG1/.critPointsRe;
critPointsMap = Table[
  {critPointsRe[[i]], critPointsValRe[[i]]},

```

```

  {i, 1, Length[critPoints]}];
min = Min[critPointsValRe];
minCond = Function[ri, ri[[2]] == min];
"The optimal solution is : "
solFin = Select[critPointsMap, minCond][[1]]

```

For presenting the optimal value of the variable $p1$, we need to add the following instruction:

```
"p1 = " <> ToString[p1/.solFin][[1]]
```

6.3 A two-dimensional numerical example

For the example considered in Section 6.1, by running the programs from Section 6.1 and 6.2, we obtain

$$\mathbb{E}(T(p)) = -1 + p_1^{-2} + p_1^{-1} + p_1^{-4} p_2^{-2},$$

the optimal distribution $p^* = (0.673694, 0.326306)$ and the optimal evolution time $\mathbb{E}(T(p^*)) = 48.2808$. The results obtained by using both versions from Section 6.2 are the same.

The graph of the function $\mathbb{E}(T(p))$ of the variable $p2$ is obtained by running the following instruction:

```

Plot[{DG1, solFin[[2]]}, {p2, 0, 1},
  AxesOrigin -> {0, 0},
  AxesLabel -> {"p2", "E(T(p))"},
  Epilog -> {PointSize[0.02],
  Point[{solFin[[1]][[1]][[2]], solFin[[2]]}]}]

```

Figure 1 represents this graph and the tangent line $\mathbb{E}(T(p)) = 48.2808$ in the point $(0.326306, 48.2808)$.

Figure 1. The graph of the two-dimensional example.

6.4 A three-dimensional numerical example

We consider the zero-order Markov process with final sequence of states $X = \{x_1, x_2, x_3, x_1, x_2, x_3\}$. By running the programs from Section 6.1 and Section 6.2, we obtain

$$\mathbb{E}(T(p)) = -1 + (p_1 p_2 p_3)^{-1} + (p_1 p_2 p_3)^{-2},$$

the optimal distribution $p^* = (1/3, 1/3, 1/3)$ and the optimal evolution time $\mathbb{E}(T(p^*)) = 755$.

The graph of the function $\mathbb{E}(T(p))$ of the variables $p2$ and $p3$ is obtained by running the following instruction:

```

Plot3D[{DG1, solFin[[2]]}, {p2, eps, 1 - eps},
  {p3, eps, 1 - p2 - eps}, AxesOrigin -> {0, 0, 0},
  AxesLabel -> {"p2", "p3", "E(T(p))"},
  PlotRange -> {{0, 1}, {0, 1}, {0, 5000}}]

```

Figure 2 represents this graph together with the tangent plane $\mathbb{E}(T(p)) = 755$ in the optimal point $(1/3, 1/3, 755)$.

Figure 2. The graph of the three-dimensional example.

7 Conclusions

In this paper the following results were established:

- analysis of the zero-order Markov processes with final sequence of states and unit transition time;
- optimization of the existing algorithm for determining the distribution of evolution time;
- optimization of the expectation of evolution time by applying geometric programming approach;
- implementation of the obtained methods by using Wolfram Mathematica[®] package;
- verification of the obtained scientific results by testing the implemented versions for various numerical examples.

These results can be used in the theory of Markov processes and its applications.

Acknowledgements

This material is supported by 13.08.819.05F Independent Project for Young Researchers "Numerical Methods for Solving Variational and Stochastic Optimal Control Problems". The author thanks the scientific researchers who contributed with their suggestions and comments regarding this paper.

REFERENCES

- [1] E. Andersen, Y. Ye. A computational study of the homogeneous algorithm for large-scale convex optimization, *Comput. Optim. Appl.*, No. 10(3), 243-269, 1998.
- [2] M. Avriel, R. Dembo, U. Passy. Solution of generalized geometric programs, *Int. J. Numer. Methods Eng.*, No. 9(1), 149-168, 1975.
- [3] S. Boyd, S. Kim, L. Vandenberghe, A. Hassibi. A tutorial on geometric programming, *Optim. Eng., Educational Section*, Springer, No. 8, 67-127, 2007.
- [4] S. Boyd, L. Vandenberghe. *Convex optimization*, Cambridge University Press, Cambridge, 2004.
- [5] R. Duffin. Linearizing geometric programs. *SIAM Rev.*, No. 12(2), 668-675, 1970.
- [6] R. Duffin, E. Peterson, C. Zener. *Geometric programming-theory and application*, Wiley, New York, 1967.
- [7] R. A. Howard. *Dynamic Probabilistic Systems*, Wiley, 1972.
- [8] R. A. Howard. *Dynamic Programming and Markov Processes*, Wiley, 1960.
- [9] Q. Hu, W. Yue. *Markov Decision Processes with Their Applications*, Springer-Verlag, 2008.
- [10] K. Kortanek, X. Xu, Y. Ye. An infeasible interior-point algorithm for solving primal and dual geometric programs, *Math. Program.*, No. 76(1), 155-181, 1996.
- [11] A. Lazari. Algorithms for Determining the Transient and Differential Matrices in Finite Markov Processes, *Bulletin of the Academy of Science of RM*, No. 2(63), 84-99, 2010.
- [12] A. Lazari. Compositions of stochastic systems with final sequence states and interdependent transitions, *Annals of the University of Bucharest (mathematical series)*, Vol. LXII, No. 4, 289-303, 2013.
- [13] A. Lazari. Evolution time of composed stochastic systems with final sequence states and independent transitions, *Annals of the Alexandru Ioan Cuza University - Mathematics*, Iasi, România, accepted.
- [14] A. Lazari. Probabilistic characteristics of the evolution time of discrete random systems (in romanian), *Studia Universitatis, CEP USM*, No. 2(22), 5-16, 2009.
- [15] A. Lazari. Numerical methods for solving deterministic and stochastic problems in dynamic decision systems (in romanian), PhD thesis in mathematics and physics, Chişinău, 2011.
- [16] D. Lozovanu, A. Lazari. An Approach for Determining the Matrix of Limiting State Probabilities in Discrete Markov Processes, *Bulletin of the Academy of Science of RM*, No. 1(62), 77-91, 2010.
- [17] D. Lozovanu, S. Pickl. Algorithms for Determining the State-Time Probabilities and the Limit Matrix in Markov Chains, *Bulletin of the Academy of Science of RM*, No. 1(65), 66-82, 2011.
- [18] Y. Nesterov, A. Nemirovsky. Interior-point polynomial methods in convex programming, *Studies in applied mathematics*, SIAM, Philadelphia, Vol. 13, 1994.
- [19] E. Peterson. The origins of geometric programming, *Ann. Oper. Res.*, No. 105(1-4), 15-19, 2001.
- [20] M. Puterman. *Markov Decision Processes*, Wiley, 1993.
- [21] J. Rajpogal, D. Bricker. Posynomial geometric programming as a special case of semi-infinite linear programming, *J. Optim. Theory Appl.*, No. 66(3), 444-475, 1990.
- [22] W. J. Stewart. *Probability, Markov Chains, Queues and Simulation*, Princeton University Press, 2009.
- [23] V. Ungureanu. *Mathematical Programming*. Chisinau, USM, 2001.
- [24] D. Wilde, C. Beightler. *Foundations of optimization*. Prentice Hall, Englewood, Cliffs, 1967.
- [25] C. Zener. *Eng design by geometric programming*. Wiley, New York, 1971.