Universidade Federal de Minas Gerais
Departamento de Engenharia Elétrica

Programa de Pós-Graduação em Engenharia Elétrica

Performance Evaluation of Stochastic DES Through Analytical Models and Simulation: An Open-Pit Mine Study

Roberto Gomes Ribeiro

Tese de Doutorado

Belo Horizonte
13/07/2018

Universidade Federal de Minas Gerais
Departamento de Engenharia Elétrica

Roberto Gomes Ribeiro

Performance Evaluation of Stochastic DES Through Analytical Models and Simulation: An Open-Pit Mine Study

Trabalho apresentado ao Programa de Pós-Graduação em Engenharia Elétrica do Departamento de Engenharia Elétrica da Universidade Federal de Minas Gerais como requisito parcial para obtenção do grau de Doutor em Engenharia Elétrica.

Orientador: *Prof. Dr. Carlos Andrey Maia*
Co-orientador: *Prof. Dr. Rodney Rezende Saldanha*

Belo Horizonte
13/07/2018

Ribeiro, Roberto Gomes.
R484p Performance evaluation of stochastic des through analytical models and simulation [manuscrito] : an open-pit mine study / Roberto Gomes Ribeiro. - 2018.
xxiv, 121 f., enc.: il.

Orientador: Carlos Andrey Maia.
Coorientador: Rodney Rezende Saldanha.

Tese (doutorado) Universidade Federal de Minas Gerais, Escola de Engenharia.

Anexos: f. 113-121.

Bibliografia: f. 106-112.

1. Engenharia elétrica - Teses. 2. Markov, Processos de - Teses. 3. Mineração a céu aberto - Teses. I. Maia, Carlos Andrey. II. Saldanha, Rodney Rezende. III. Universidade Federal de Minas Gerais. Escola de Engenharia. IV. Título.

CDU: 621.3(043)

"Performance Evaluation of Stochastic DES Through Analytical Models and Simulation: An Open-pit Mine Study"

Roberto Gomes Ribeiro

Tese de Doutorado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Doutor em Engenharia Elétrica.

Aprovada em 13 de julho de 2018.

Por:

Prof. Dr. Carlos Andrey Maia
DEE (UFMG) - Orientador

Prof. Dr. Rodney Rezende Saldanha
DEE (UFMG) - Coorientador

Prof. Dr. Marcone Jamilson Freitas Souza
Departamento de Computação (DECOM) (UFOP)

Prof. Dr. Fernando Luiz Pereira de Oliveira
Instituto de Ciências Exatas e Biológicas (UFOP)

Prof. Dr. Frederico Rodrigues Borges da Cruz
Departamento de Estatística (UFMG)

Dr. Adriano Chaves Lisboa
Gaia (BH-TEC)

Prof. Dr. Vinícius Mariano Gonçalves
DEE (UFMG)

Acknowledgements

I would like to thanks the agencies FAPEMIG, CAPES and CNPq for supporting this thesis. In addition, the Universidade Federal de Minas Gerais - UFMG for providing a high-quality education in all aspects.

INCIPIT VITA NOVA

Whenever I find myself growing grim about the mouth; whenever it is a damp, drizzly November in my soul; whenever I find myself involuntarily pausing before coffin warehouses, and bringing up the rear of every funeral I meet; and especially whenever my hypos get such an upper hand of me, that it requires a strong moral principle to prevent me from deliberately stepping into the street, and methodically knocking people's hats off - then, I account it high time to get to sea as soon as I can.

—HERMAN MELVILLE (Moby Dick)

Resumo

Apesar de o avanço computacional ter viabilizado a otimização de DES (Discrete Event Systems) estocásticos, a criação de metodologias eficientes para este propósito continua sendo um grande desafio. A principal razão é que soluções numéricas para problemas DES estocásticos exigem um considerável custo computacional. Assim, a presente tese discute um problema de escolha de portfólio de projetos não-linear, no qual, estamos interessados em estimar o impacto de cada combinação de projeto através de métodos que lidam com DES estocástico. Particularmente, o sistema em estudo é uma operação de Mineração a céu aberto. Trata-se de um sistema complexo, onde diversos fatores impactam nos resultados finais de produção. Caracterizado pelas incertezas de seus processos, o ambiente de mineração é comumente analisado por meio de simulação de eventos discretos. No entanto, por via de regra, esta ferramenta exige um alto custo computacional. Dentro do contexto de otimização, esta medida é um aspecto crucial, uma vez que limita o número de cenários a serem avaliados. Assim, este trabalho recorre a metodologias analíticas eficientes na análise de problemas desta natureza. Na ocasião, apresentamos métodos de aproximação analítica e uma nova metodologia híbrida que exploram propriedades Markovianas para modelar o sistema em questão. No que diz respeito aos métodos de aproximação analítica, apresentamos abordagens de aproximação por primeiro e segundo momento. Pelos experimentos realizados, não foi possível afirmar que existem evidências de equivalência entre os modelos analíticos desenvolvidos e um modelo de simulação padrão. Porém, os resultados sugerem que as aproximações analíticas realizadas são atraentes e competitivas formas de lidar com a otimização do portfólio de projetos, uma vez que o tempo computacional utilizado para avaliar cada estimativa é extremamente menor que ferramentas de simulação convencionais. Em relação à abordagem híbrida apresentada neste trabalho, foi desenvolvida uma nova metodologia que agrega Algebra Max-Plus com Cadeia de Markov para modelar o mesmo sistema. Apesar de não ser tão eficiente quando as aproximações analíticas, os resultados obtidos sinalizam que a nova metodologia híbrida é também mais rápida que ferramentas convencionais de simulação. Além disso, a análise experimental realizada nos permite dizer que existem evidências de equivalência entre os resultados obtidos utilizando a abordagem híbrida e por uma ferramenta de simulação padrão. Uma vez que esta tese aborda um problema não-linear de escolha de portfólio, também propomos uma estratégia indutiva de linearização. Tal técnica é um mecanismo de geração de colunas, no qual, os termos não-lineares são parcialmente convertidos em novas variáveis de decisão. Basicamente, colunas são geradas heuristicamente com base na dinâmica do sistema e avaliadas por algum método DES estocástico. Temos então um problema linear da mochila em que o número de colunas geradas depende do método DES estocástico utilizado. Os resultados deste estudo indicam que a estratégia de linearização desenvolvida em conjunto com um dos métodos de aproximação analítica consiste em uma metodologia eficiente para tomada de decisão em problemas de portfólio caracterizados por inter-relações entre projetos.

Palavras-chave: Cadeia de Markov; Rede de fila fechada; portfólio de projetos; mineração a céu aberto.

Abstract

Although the increase of computing power over the last years have opened up the possibility of optimizing stochastic DES (Discrete Event System) problem, the development of efficient methodologies for this purpose is still a grand challenge. The main reason is the fact that numerical solutions for stochastic DES problem, generally, require considerable computational effort. This Thesis discusses a nonlinear project portfolio problem which we must estimate the effect of each project combination through stochastic DES methods. Particularity, the system studied represents an open-pit mine appropriate for estimating the iron production index. Once that the time taken to run each estimative is a crucial aspect in optimization context, the purpose of this work is presenting alternative methods which address such stochastic DES system. Hence, we explored Markovian properties to design a load-haulage cycle of an open-pit mine. The Thesis presents analytical approximation methods and a new hybrid methodology. Regarding the analytical methods, we considered a first-moment and a second-moment approximation. Although we did not find evidence of equivalence between the analytical models and a standard simulation model, the results suggest that analytical approximation methods consist of a quite attractive and competitive way to concern with project portfolio optimization once that the computational time taken to run each estimative is remarkably faster than the standard simulation tool. Regarding the new hybrid method, it consists of an alternative methodology also faster than the standard simulation tool, but not so fast as analytical approximation methods. Such methodology aggregates Max-Plus Algebra with Markov Chain for modeling the same system. The experimental analysis conducted showed evidence of equivalence between the results acquired by this hybrid methodology and by the standard simulation tool. Once that this Thesis addresses a nonlinear project portfolio problem, we also proposed an inductive linearization technique. Regarding this technique, it consists of a column generation mechanism, in which, the nonlinear terms are partiality converted into new decision variables. The columns are generated heuristically based on the system dynamic and evaluated by some stochastic DES method. The outcome is a linear Knapsack problem where the number of columns depends on the stochastic DES method used. As a result, we can state that this linearization strategy combined with an analytical approximation method consists of an efficient strategy address decision makers of a project portfolio problem with interrelationships between projects.

Keywords: Markov chain; closed queuing network; project portfolio; open-pit mine.

Contents

List of Figures

List of Tables

Nomenclatures

cdf	Cumulative Distribution Function
DES	Discrete Event System
FIFO	First In First Out
GSMP	Generalized Semi-Markov Process
IS	Infinity Service
iid	Independent and Identically Distributed
KP	Knapsack Problem
LIFO	Last In First Out
LST	Laplace-Stieltjes Trasform
pdf	Probability Density Function
pmf	Probability Mass Function
PS	Processor Sharing
QKP	Quadratic Knapsack Problem
RR	Round Robin
TEG	Timed Event Graph
TOST	Two One-Sided Test
VBA	Visual Basic for Applications

List of Symbols

$\mathbf{X}$	Finite state space of an automata.
$\mathbf{X_m}$	Finite market state space of an automata.
$\mathbf{E}$	Countable set of event.
$f(.)$	State transition function.
$\Gamma(.)$	Feasible event function of a state.
x_0	Initial state of an automata.
x_k	State of an automata at the instant t_k.
x	Current state of an automata.
$\boldsymbol{\chi}$	Tuple which denotes the state of an automata formed by parallel composition.
$\mathbf{V_d}$	Deterministic clock structure.
$\mathbf{v}_\alpha$	Clock sequence of the event α.
$v_{\alpha k}$	Interval between *k-th* and *(k-1)-th* occurrence of event α.
$y_{\alpha k}$	Instant of *k-th* occurrence of event α.
$\mathbf{y}$	Set of current clock values.
$\mathbf{y}'$	Set of clock values after a transition occurrence.
N_α	Current score of event α.
$\mathbf{N}$	Set of current scores of an automata.
$\mathbf{N}'$	Set of scores of an automata updated after a transition occurrence.
$p(x',x,e)$	State transition probability from x to x' conditioned the occurrence of event e.
$p_0(x)$	Probability of an automata starts in state x.
X	Random variable.
$\{X(t)\}$	Stochastic process that represents the random variable X.
$X(t_k)$	Random variable for X at the instant t_k.

t_k	Discrete time.
$\mathbf{x}$	Sequence of state.
$C_X(.)$	Cumulative Distribution Function of the random variable X.
$\mathbf{V_s}$	Stochastic clock structure.
V_α	Random variable which describes the occurrences of event α.
$\{V_\alpha(t)\}$	Lifetime sequence of event α.
$N_\alpha(t_k)$	Score of event α at the instant t_k.
$N_\alpha(t_k, t_{k+1})$	Score of occurrence of event α in the interval (t_k, t_{k+1}).
p_{ij}	State transition probability from state i to state j.
L	Number of state.
$\mathbf{M_p}$	Transition probability matrix.
$\mathbf{M_q}$	Transition rate matrix.
$\pi_i(t_k)$	Probability of a system be in the state i at the instant t_k.
$\boldsymbol{\pi}(t_k)$	Vector composed by the probabilities of each state at the instant t_k.
π_i	Stationary probability of state i. For a queue network, it is denoted by the M-tuple, where M is the number of nodes.
$\boldsymbol{\pi}$	Stationary probability vector.
K	Queue capacity.
q	Number of server in a queue.
q_m	For a queue network, it denotes the number of server of the node m.
$\boldsymbol{\chi}$	Tuple that represents a state in a queue network.
$\pi_m(n)$	Marginal probability of n nodes in queue m.
$\bar{N}_m$	Mean number of nodes in queue m.
$\bar{t}_m$	Mean response time of queue m.
$\bar{t}_m(.)$	Dependent mean response time of queue m.
$G(N)$	Normalization constant.
$g(N,M)$	Representation of normalization constant taken into account subnetworks of the main system.
λ_α	Occurrence rate of event α.
μ_m	Service rate of node m.
μ'	Service rate of node into a phase-type distribution.

$\mu_m(n)$	Dependent service rate of node m with (n clients).
ν_m	Arrival rate of a open queuing network. For a closed queuing network, it denotes the relative arrival rate.
τ	Continuous time variation.
k	Discrete-time variation.
α, β	Events.
ψ_m	*Throughput* of queue m.
κ	Number of phases.
cv	Coefficient of variation.
n_m	Current number of trucks in the process m.
$\lambda_m(n_m)$	Dependent occurrence rate.
$\nu_m(n_m)$	Dependent arrival rate in the process m.
θ_i	Decision variable i (projects i).
$\boldsymbol{\theta}$	Project portfolio.
$J[\theta,\omega]$	Performance index for a sample sequence ω.
$E[J(\theta)]$	Expected value for $J(\theta)$.
$\hat{E}[J(\theta)]$	Estimative for the expected value $E[J(\theta)]$.
$E[Prod]$	Expected production index i.
$E[Prod(\theta_i)]$	Dependent expected production index i.
$E[Prod(\theta)_{first}]_\ell$	Expected iron production index applying the First-moment method.
$E[Prod(\theta)_{second}]_\ell$	Expected iron production index applying the Second-moment method.
$E[Prod(\theta)_{siman}]_\ell$	Expected iron production index applying the Siman model.
$E[Prod(\theta)_{hybrid}]_\ell$	Expected iron production index applying the Hybrid method.
ST_j	Service time of process j.
$ST_j(\boldsymbol{\theta})$	Dependent service time of process j.
$E[ST_j(\boldsymbol{\theta})]$	Expected service time of process j.
$\varphi_{\ldots}$	Term which denotes the interrelationship between projects.
$E\left[\pm J(\varphi_{ij})\right]$	Measure of intensity, direction, and relationship between projects i e j.
Z_i,Z_i'	Random variables.
ω_1	Auxiliary term for experiment 1.

ω_2	Auxiliary term for experiment 1.
ω_3	Auxiliary term for experiment 1.
$H_{0_1}^{+}$,$H_{0_1}^{-}$	Null hypotheses of experiment 1.
$H_{a_1}^{+}$,$H_{a_1}^{-}$	Alternative hypotheses of experiment 1.
$H_{0_2}^{+}$,$H_{0_2}^{-}$	Null hypotheses of experiment 2.
$H_{a_2}^{+}$,$H_{a_2}^{-}$	Alternative hypotheses of experiment 2.
$H_{0_3}^{+}$,$H_{0_3}^{-}$	Null hypotheses of experiment 3.
$H_{a_3}^{+}$,$H_{a_3}^{-}$	Alternative hypotheses of experiment 3.
n_p	Number of projects in a portfolio.
n_v	Number of decision variables in a portfolio problem.
n	Dynamic Number of trucks.
N	Fixed Number of trucks.
C	trucks capacity.
$T_{horizon}$	Time horizon.
$\bar{T}_{cicle}$	Cycle time.
$\varepsilon_{\oplus}$	Neutral element.
A,B and C	Labels for places in TEG.
T_1 and T_2	Sojourn time labels in TEG.
$Y(k)$	Instant of the k-th departure.
$u(k)$	Instant of the k-th occurrence of an event.
$T_{cycleMP}$	Cycle-time vector in the Max-Plus definition of the general DES system.
$E[T_{cycleMP}]$	Expected Cycle-time vector of the general DES system.
$T_{cycleMP_n}$	Cycle-time vector of a sub DES system without decisions running with n trucks.
$E[T_{cycleMP_n}]$	Expected Cycle-time vector of a sub DES system without decisions running with n trucks.
$\rho(n)$	Server utilization rate of the 'dump zone' process of a sub DES system without decisions running with n trucks.
R	Budget available.
$\boldsymbol{\theta_s}$	Project portfolio selected.
$E\left[Port(\boldsymbol{\theta_s})\right]$	Expected value of a portfolio.

$E[Prod(0)]$	Expected iron production index of the scenario without projects application.
$E[G(\theta_i)]$	Increase of total production index provided by project θ_i.
$\mathbf{A}$	Dynamic auxiliary matrix for the new knapsack formulation.
a_{ij}	Elements of matrix $\mathbf{A}$.
$\mathbf{\Theta}$	Set of decision variable.
θ_i	Decision variable i.
w_i	Cost associated with the decision variable i.
$\zeta_{ij}(m)$	Auxiliary function which describes the influence of projects θ_i and θ_j at the process m.
ϕ_{ij}	Term which denotes the intensity of the interrelationship between two projects.

CHAPTER 1: INTRODUCTION

"... huge amounts of available computing resources increase the trend to solve models through simulation and do not encourage researchers to look for tractable analytical solutions."
Raymond A. Marie

It is probable that never in history have been discussed and created computational methods for analysis of stochastic problems as in the last years. The extraordinary advance of computational resources allowed the development of a wide range of tools for this purpose. Many of these tools are intended for dealing with the stochastic behavior of the problems. In the global world of nowadays, it is usual that companies make regular investments in their production operations to increase the competitiveness. However, the stochastic nature of these operations does not guarantee the expected return on the investments. Taken it into account, Companies employ the mentioned tools and make decisions based on probabilistic measures.

Usually, we solve problems of stochastic behavior through simulation. According to Chwif and Medina (2007), the term *'simulation'* must be classified into two categories. The first consist of solutions that don't need computational methods. A good example is the construction of physical prototypes for analyses of real systems. Studies which are conducted by this approach are classified as *'non-computational simulation'*. In the second category, the use of computational tools is indispensable.

Classified as *'computational simulation'*, such category is composed of computational models which depict the dynamic of real systems. The use of simulation represents the possibility of predict behavior. Then, in the decision-making context, the goal is looking into the future and seeing the result of investments before their application. According to Kellner et al (1999), *computational simulation* is an inexpensive way to gain important insights when the costs, risks or logistics of manipulating the real system of interest are prohibitive.

One important subcategory of *computational simulation* is DES (Discrete Event System) simulation. According to Banks (2005), DES simulation is the modeling of systems in which the state variables change only at those discrete points in time at which events occur. This condition allows that the model represented by the simulation to be analyzed numerically.

Banks (2005) also points out that, with the extraordinary advance of computer hardware, DES simulation became one of the most widely used and accepted methodologies in the field of operational research and system analysis. Taken into account the benefits of DES simulation, many commercial software packages were developed. As computer hardware became more powerful, more accurate and faster, these packages do as well. Thereby, the number of companies which employs DES simulation in decision-making mechanisms increased substantially.

On the decision-making context, there is a class of problem known as project portfolio selection. According to Archer and Ghasemzadeh (1999), it is a crucial decision in many companies, which must make informed decisions on investment, where the appropriate distribution of investment is a challenge. There are many factors responsible for this complexity. One is the uncertain inherent to the investments. As explained previously, a standard methodology for dealing with such uncertain is DES simulation. Summarily, we analyze the investments through a DES simulation model which depicts the real system. Hence, we select the investment which best meets the company goals.

Another factor is the interrelationship among projects that compose the project portfolio. It means that it is necessary to consider how a project can affect the others. According to Costa (2011), many studies in the literature point out that this factor constitutes the most critical of all factors involved in the project portfolio problem. One reason is the fact that it is difficult, even impossible, mapping and quantifying all possible interrelationships.

Thus, a proposal for obtaining answers to a project portfolio problem is by optimization methods. When these methods concern DES simulation problem, they can be classified into a methodology known in the literature as *'simulation optimization'* or *'optimization via simulation'*. According to Fu (2002), the assumption in the simulation optimization is that the objective function is not available directly and we must estimate it through simulation. Since that project portfolio selection is a discrete optimization problem in which we analyze the projects through of a DES simulation model, we can classify this problem into a subclass named *'discrete optimization via simulation'*.

According to Nelson (2010), such subclass addresses solving stochastic problems with a countable number of feasible solutions, when the system is complicated enough that the expected value is neither analytically nor numerically tractable but we can estimate it by running simulation. That is the reason for most popular commercial simulation softwares have embedded some optimization mechanism. Fu (2002) summarizes the strategies of search used by these softwares. Generally, these tools have a slow convergence in practice because DES simulation requires considerable computational effort.

Regarding the difficulties in dealing with discrete optimization via simulation problems, this study explores two fields to improve the quality of the solutions. The first field consists of developing an adequate optimization mechanism. In this field, the present Thesis shows a computational method for project portfolio selection in open-pit mining. We can classify such problem as discrete optimization via simulation because the search for optimal solutions happens within a stochastic environment, in which, we must obtain each project profit through of a DES simulation model.

To improve the optimization method efficiency, we explore peculiarities of the problem to limit the number of evaluations. The justification is the fact that it is necessary a considerable computational effort for evaluating a DES simulation model. This situation influences directly the optimization strategy to be chosen. The second field consists of reducing the time taken to run each function evaluation and, consequently, using more adequately the computational resource available.

As mentioned before, DES simulation requires a substantial computational effort. According to Bolch et al (2006), the main drawback of DES simulation is the time taken to run the models for large, realistic systems particularly when results with high accuracy are desired. Since that the stochastic nature of the models does not guarantee exact answers, the obtained results by DES simulation are featured by probabilistic measures such *confidence interval* and *significance level*. These measures are directly related to the computational effort spent on obtaining results.

Once that DES simulation may not be the more convenient way in the broader context of a decision problem, we explore in this Thesis another approach for conducting the stochastic DES systems analysis. Before the creation of the computational resource available nowadays, this class of problems was usually treated analytically. According to Marie (2011), the scientific ambition was limited by computing power. Therefore, it was necessary using imagination to look for approximations in order to reduce the complexity of the problems. With the computational advance, it increases the trend of finding solutions for DES problems through simulation. According to Marie (2011), this situation does not encourage researchers to look for tractable analytical solutions. Marie (2011) also argues that designers prefer to increase the computational resource when the expected performances are not attained rather than looking for other solutions.

Since that this work concerns to 'discrete optimization via simulation' problem in which the computational time is a crucial factor, analytical methodologies that deal with stochastic problems are explored. The goal is to present alternatives methods faster than standard DES simulation tools. It is important to point out that DES simulation is a powerful methodology for dealing with stochastic problems. However, when it is considered a DES simulation model as an objective function of an optimization problem, the time taken to run such model is a crucial aspect. Marie (2011) explains that some mathematical/probabilistic properties can be used to analyze problems of stochastic nature without simulation. Thus, Marie (2011) addresses the way to be followed by this work. It, in turn, is organized by the following seven chapters.

Chapter 2 presents the target stochastic problem. Summarily, it is a 'discrete optimization via simulation' problem in which the decision variables impact into the open-pit mining dynamic. Since that the system is a stochastic DES, this Chapter presents a general description of stochastic problems as well. Principles about automata are presented in Chapter 3. Taken into account that stochastic timed automata are the basis of DES simulation, the goal is to show the connection between simulation and analytical approximation techniques. Like this, to complete the Chapter 3, the fundamental theory of Markov chain is presented as well.

Aiming to extend the mentioned connection, Chapter 4 presents concepts of queue theory as a special case of stochastic automata, as well as its evolution for analysis of queuing networks based on Markovian properties. Although the limitation of Markov Chain, the history shows a scientific effort for expanding the application of these properties, such a way that it can be widely use. Thus, we explore the Markovian concepts in Chapter 5, in which, a load-haulage cycle of an open-pit mine, subject to stochastic nature, is designed as a closed queuing network. As a result, Chapter 5 presents DES Markovian models which employ methods of first-moment and second-moment approximation. Moreover, this Chapter shows the first batch of experiments as well. Regarding this test, an experiment analysis is conducted to compare the obtained results by the mentioned analytical methods and the ones acquired by a standard simulation tool.

Bearing in mind the limitation of DES Markovian models, Chapter 6 presents Stochastic Max-Plus System, which is a particular class of stochastic DES with compact representation, easier of coding in a program and faster of simulating. As a contribution, we expand the modeling capability of this system by aggregating Markov properties into it. Summarily, the methodology consists of a new modeling method faster than standard simulation tools to concern with stochastic systems. The methodology aggregates Max-Plus Algebra with Markov Chain for modeling the same load-haulage cycle. To verify the accuracy of this new methodology, we conducted another batch of experiments in which the goal is to compare the obtained results with the results presented in Chapter 5.

Regarding the decision-making context, an optimization strategy for project portfolio selection is presented in Chapter 7. Substantially, we proposed a new linear formulation that considers the interrelationship between projects. Instead of trying all the possible projects combinations, the proposed strategy searches for identifying the set of projects which produce good feasible solutions based on performance measures from the designed DES model. At last, a conclusion and prospects are given in Chapter 8.

CHAPTER 2: CHARACTERIZATION OF THE PROBLEM

"Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin."
John Von Neumann

2.1 A description of the problem

Usually, stochastic DES models are evaluated by computational simulation tools. Although computer hardware have been becoming more fast and powerful, there is nowadays a scientific effort for developing of efficient and fast simulation models. Such effort is further plausible when we associate simulation with an optimization problem. The aggregation of these two methodologies consists of a *optimization via simulation*. As argued by Fu (2002), the assumption of this aggregation is the fact the objective function estimates are not possible without running simulation models.

Figure 2.1 illustrates an abstraction of the *optimization via simulation* methodology. As it can be seen, an optimization strategy generates a setting of decision variables $\{\theta_1, \theta_2, \theta_3, \ldots\}$ and the result $E[J(\theta)]$, which is the expected value of $J(\theta)$, is obtained through a simulation model. Then, the optimization strategy uses $E[J(\theta)]$ to define a new setting of values. This cycle repeats until the algorithm reaches a computational time or until the optimization strategy finds a local optimum.

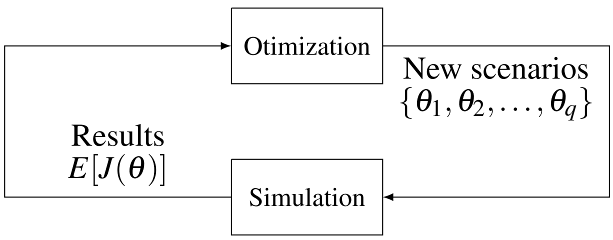


Figure 2.1: Optimization via Simulation

In this context, we consider as a numerical example an open-pit mine with stochastic behavior, in which, we are interested in evaluating the expected value of the iron production index. Summarily, the system is composed of many connected queues characterized by general distribution. Figure 2.2 illustrates a load haulage cycle which is an abstraction of a real open-pit mine.

Fundamentally, a fixed number of trucks transports the extracted material from the load zone to the dump zone. During this operation, queues can be formed in each process. Another feature of the system is the fact that trucks can exit the load haulage cycle because of some operational stop. A stop can be preventive or corrective maintenances, supplies, or even exchange of operators after the working time. After a sojourn time, the stopped truck goes back to the load haulage cycle and, so, keeps its operation.

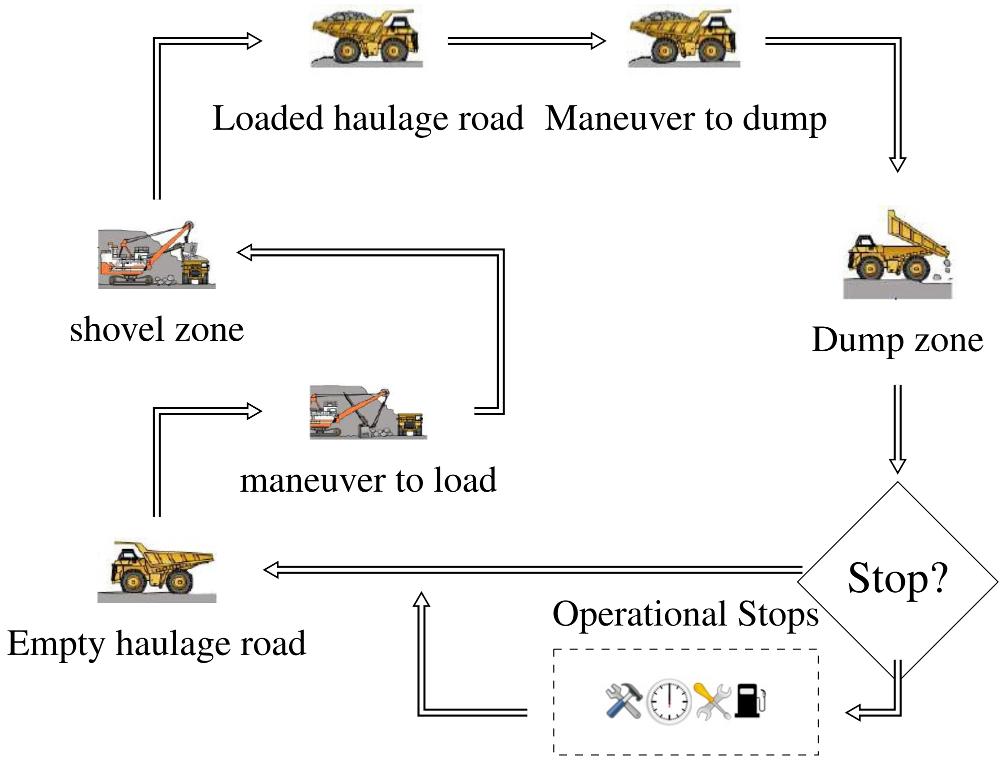


Figure 2.2: Load haulage cycle

As before-mentioned, we are interested in evaluating the expected value of the iron production index. This measure depends on load-haulage cycle dynamic. Then, it is a non-deterministic output of a stochastic DES system. The next section presents a general description of this class of problems.

2.2 A general description of stochastic DES problems

In a stochastic problem, we usually characterize the environment by the uncertainties coming from its activities. In practice, many systems have some uncertain behavior which fit it into this class of problem. Such behavior is related to human decisions, natural phenomenon or non-deterministic fails. To analyze a problem which is within this class, we need historical databases that portray the system dynamics. Therefore, we may describe the processes in terms of random variables.

Since we are dealing with random variables to model uncertainties, a common way of evaluating it is through an expected value. This amount can represent production rates from a specific agent, average times, mean number of visit in a process, etc. Let $J(\theta, \omega_i)$ be a perform index which depends on the parameter θ and the sample sequence ω. The expected value $E[J(\theta)]$ is expressed by the Equation (2.1):

$$E[J(\theta)] = \lim_{k \to \infty} \frac{\sum_{i=1}^{k} J(\theta, \omega_i)}{k}, \tag{2.1}$$

where k denotes the amount of sample. In this case, we assume that the stochastic system goes to the steady state. It means that the stochastic process associated with the system is stationary and ergodic.

For stochastic DES problems, there are two different fields to estimate the expected value $E[J(\theta)]$. The first is through simulation. A common tool address DES simulation is the SIMAN language (Profozich and Sturrock, 1995). More recently, Max-Plus Algebra (Akian et al, 2006) appears as another attractive way for a special class of DES simulation. The second field consists of solutions using analytical models. In this field, we explore Markovian properties, phase-distribution, and concepts of queue theory. We discuss both fields in next sections.

2.3 Approximation by DES simulation

In the analysis through DES simulation, there are computational limits regarding processing and memory capacity. So, it is required a finite number of samples K and an estimator of the expected value $\hat{E}[J(\theta)]$. It means that an approximation for $E[J(\theta)]$ can be measured by the Equation (2.2):

$$\hat{E}[J(\theta)] = \frac{\sum_{i=1}^{K} J(\theta, \omega_i)}{K}. \tag{2.2}$$

Using this approach, we must characterize the desired estimative by its variability. Therefore, it is necessary to use statistical methods to perform the sample size K for some variance and the confidence bounds. The tutorial Guo et al (2013) explains several commonly used methods for sample size determination. In general, a small value for K in a stochastic problem cannot provide a good accuracy while a high K increases the execution time once that each sample evaluation requires a certain computational effort. Considering these circumstances, we must limit the sample size due to the available computational resource.

Another important feature related to computational effort, in a stochastic DES problem analysis, it is the fact that each sample $J(\theta, \omega_i)$ is a random variable. Thus, it is necessary a true random number generator which a computer employs for simulation purposes. According to Cassandras and Lafortune (2009), the termination "random number generator" refers to a mechanism that generates *iid* (independent and identically distributed) samples from a uniform distribution. Usually, this mechanism is a pseudo-random number generator. The reason is the fact that the basis of number generation is a deterministic function. Thus, the result produced has a pseudo-random behavior. In this context, Von Neumann (1951) advises about the misinterpretation of pseudo-random number generator as a truly random generator.

Nowadays, there is a broad scientific interest address this field. The article Wichmann and Hill (2006) synthesizes some pseudo-random number generator and some statistical metrics sufficient to evaluate its quality. Furthermore, Cassandras and Lafortune (2009) points out that the quality of the pseudo-number generator depends on the amount of computational effort we are willing to put in. For this reason, we reinforce that a DES simulation tool may not be suitable for applying in the optimization field.

2.4 Approximation through analytical methods

In general, analytical approximation methods require much less computational effort than simulation. The main reason is a fact that neither a pseudo-random number generator nor a number K of samples are required. Rather of it, we represent the stochastic dynamic of the model by a Markov chain in which a linear equation system describes its behavior. As a result, the solution of this system gives us the stationary probabilities of the Markov chain and, consequently, we can obtain the expected value $E[J(\theta)]$.

Regarding this modeling method, a disadvantage is the fact that the cardinality of the state space can grow drastically, i.e, computing the stationary probabilities can be a hard (even impossible) task in complex systems. However, when we can express a stochastic DES model as a queuing network, very fast numerical techniques have been developed to derive important performance measures without resorting to the underlying state space (Bolch et al, 2006).

In spite of this efficiency, in a stochastic DES problem each process of the system can be characterized by general distributions. This feature implies that there is no analytical solution and an approximation method is required to evaluate an estimative of $E[J(\theta)]$ i.e the value of $\hat{E}[J(\theta)]$. One approach is to apply a first-moment approximation. In this method is considered that the service time of each process is exponentially distributed, even following a general distribution. According to Marie (2011), it is a classic approximation which is done consciously and its consequence depends generally on the modeled context.

Another method of analytical approximation is though phase-type distribution. Regarding this method, each process characterized by a general distribution is substituted by a grid of fictitious process with inter-event times exponentially distributed. According to Cox (1955), we can represent any general distribution having a rational LST (*Laplace-Stieltjes Trasform*) by a sequence of exponentially distributed phases. Altiok (1985) points out that "phase-type distributions are important in queuing theory because their structure can give rise to a Markovian state-description". Since there is an analytical solution for Markovian processes, the Altiok's statement indicates that we can measure the value of $\hat{E}[J(\theta)]$ from phase-type distribution approximation.

As DES simulation, both before-mentioned analytical approaches for conducting DES problem are approximations. However, we can characterize simulation results by statistical measures as variance and confidence bounds. Regarding the analytical approximation, the methods produce exact estimates. In other words, it is not possible to predict the accuracy of the result when $\hat{E}[J(\theta)]$ is obtained using analytical approximation methods.

CHAPTER 3: PRELIMINARY CONCEPTS

"Probability and poetry were unlikely partners
in the creation of a computational tool"
Brian Hayes

3.1 Concepts of Automata

A DES is generally composed of a set of states and possible events which define its evolution. There are many formal ways for modeling the logical behavior of DES. One of this ways is by means of automata. Also known as '*state-finite machine'*, it is an abstract instrument that represents a language with a set of well-defined rules. In addition, it can be classified as deterministic or stochastic, timed or untimed and finite or infinite. The aim is to introduce concepts of stochastic timed automata because it is the base of DES simulation. Moreover, Cassandras and Lafortune (2009) claims that any study of DES must start with a study of automata.

3.1.1 Automata notation and definitions

The first step is to show a formal definition of a deterministic untimed automaton, such presented in Cassandras and Lafortune (2009).

Definition 3.1.1. A deterministic untimed automaton is denoted by the 6-tuple $(\mathbf{X}, \mathbf{E}, f, \Gamma, x_0, \mathbf{X_m})$. where:

- $\mathbf{X}$ is a finite set of states;
- $\mathbf{E}$ is a Countable set of events associated with the automaton F;
- $f : \mathbf{X} \times \mathbf{E} \to \mathbf{X}$ is the state transition function: $f(x_1, \alpha) = x_2$ means that there is a transition labeled by event α ($\alpha \in \mathbf{E}$) from state x_1 to state x_2;
- Γ: $\mathbf{X} \to 2^{\mathbf{E}}$ is a feasible event function; $\Gamma(x)$ denotes the set of all active events when the system is in the state x;
- x_0 is the initial state ;
- $\mathbf{X_m}$ is the set of marked states($\mathbf{X_m} \subseteq \mathbf{X}$).

An automaton can be represented graphically. Figure 3.1 shows a deterministic finite-state automaton composed by two states and two events.

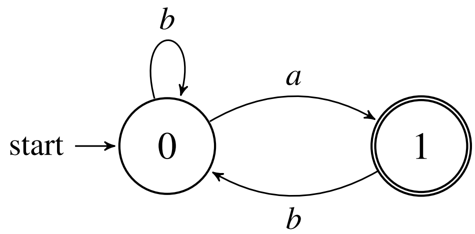


Figure 3.1: Deterministic finite-state automaton with two states

The follow 6-tuple describes a formal definition of this automaton:

- $\mathbf{X} = \{0,1\}$;
- $\mathbf{E} = \{a,b\}$;
- $f : f(0,a) = 1, f(0,b) = 0$ and $f(1,b) = 0$;
- Γ: $\Gamma(0) = \{a,b\}$ and $\Gamma(1) = \{b\}$;
- $x_0 = 0$;
- $\mathbf{X_m} = \{1\}$ (just state 1).

The definition denotes that the system starts in state 0 and can only end in state 1 (marked state). Supposes that the system is in state 0. Thus, both events a and b can occur. The only way to reach state 1 is whether event a occurs at some point. In addition, the system remains in state 0 forever unaffected or it eventually becomes 0 if event b occurs when the system is in state 1.

As mentioned previously, automata are used to represent the logical behavior of DES. Therefore, they can be applied widely. For instance, in the logistic field, an inventory control can be modeled as an automaton. In this application, events can mean the depart and arrival of products, while the states are the stock volume. In manufacture systems, the states of an automaton can denote the possible situations of a production chain. Systems featured by queues are also usually described by automata, in which, the states in this system indicate the number of clients in the queues.

Regarding the last class of the mentioned system, let us supposes a finite queue with one server and three wait modules, such illustrated in Figure 3.2. The system starts with 0 clients and must end in the same situation. Such system can represent a service terminal composed of a unique server and a limit number of 'waiting chair'. Hypothetically, clients only opt to wait whether some spot is available. Another feature is the fact that the service terminal can only finish its activity after attending all clients.

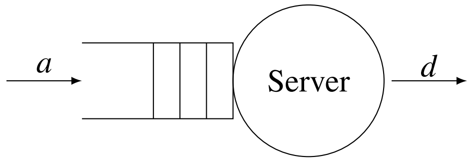


Figure 3.2: Finite queue

The labels a and d denotes in Figure 3.2 the arrival and depart of clients, respectively. Since that the system denotes a finite queue, clients only come in if the queue is not full. It means that the arrival of clients depends on the current number of clients in the system. Regarding the depart, the event d is only feasible whether at least one client is in the system.

The logical behavior of this system can be described by the finite automaton illustrated in Figure 3.3.

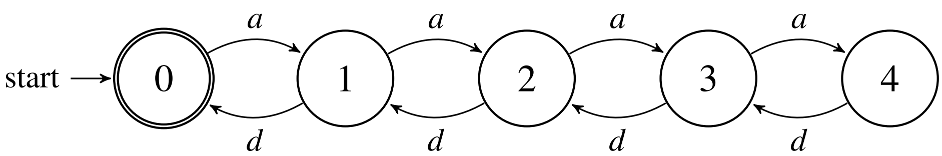


Figure 3.3: Finite queue with buffer capacity 4 described by an automaton

The formal definition for the automaton shown in Figure 3.3 is described by the 6-tuple $(\mathbf{X}, \mathbf{E}, f, \Gamma, x_0, \mathbf{X_m})$, whose:

- $\mathbf{X} = \{0, 1, 2, 3, 4\}$;
- $\mathbf{E} = \{a, d\}$;
- $f : f(0,a) = 1, f(1,a) = 2, f(1,d) = 0, f(2,a) = 3, f(2,d) = 1, f(3,a) = 4, f(3,d) = 2$ and $f(4,d) = 3$;
- Γ: $\Gamma(0) = \{a\}$,$\Gamma(1) = \{a,d\}$, $\Gamma(2) = \{a,d\}$, $\Gamma(3) = \{a,d\}$, $\Gamma(4) = \{d\}$;
- $x_0 = 0$;
- $\mathbf{X_m}$: just state 0.

3.1.2 Parallel composition operador

To model of systems composed of a set of automata that operate concurrently, it is necessary composition operators. One of this operators is named 'parallel composition'. According to Cassandras and Lafortune (2009), "the standard way of building models of entire systems from models of individual system components is parallel composition". Therefore, such operator allows the analysis of complex systems, which are a combination of many processes.

Cassandras and Lafortune (2009) argue that in general systems are formed by components which interact. Thus, the set of events **E** of each component includes private events that pertain to its own internal behavior and common events that are shared with other automata. Common event can only be executed whether all component of the composition execute it simultaneously. Regarding private events, they are not subject to this constraint and can be executed whenever possible. In another words, each component can execute its private events without the communion of the others, while common events can only occur whether all components can execute it.

Let suppose as an example two automaton, $F_1$and F_2. Taken into account the follow definition, $F_1||F_2$ denotes the parallel composition of the mentioned automata.

Definition 3.1.2. The operator $||$ denotes the parallel composition of automata.

The 6-tuple $(\mathbf{X_1} \times \mathbf{X_2}, \mathbf{E_1} \cup \mathbf{E_2}, f, \Gamma_{1||2}, (x_{1_0}, x_{2_0}), \mathbf{X_{T_1}} \times \mathbf{X_{T_2}})$ denotes the result of the parallel composition $F_1||F_2$, where:

- $\mathbf{X_1} \times \mathbf{X_2}$ is the resulting set of states, which is the product of states of automata F_1 and F_2;
- $\mathbf{E_1} \cup \mathbf{E_2}$ is the resulting set of events, which is union between events of automata F_1 and F_2;
- $f : \mathbf{X} \times \mathbf{E} \to \mathbf{X}$ is the transition function;
- $\Gamma_{1||2}, (x_{1_0}, x_{2_0}))$ is the feasible event function;
- (x_{1_0}, x_{2_0}), formed by a composition between the initial states of F_1 and F_2, is the initial state of the resulting automaton;
- $\mathbf{X_{T_1}} \times \mathbf{X_{T_2}}$ is the set of marked states.

Now let suppose a system which is two finite queues in series, such illustrated in Figure 3.4. Each queue has a buffer capacity of 2, one waiting and another being serving. The system is composed of the set of events $\{a, b, d\}$. From Figure 3.4, it is possible to see that a and d are the arrival and depart of clients, while event b is the connection between the queues.

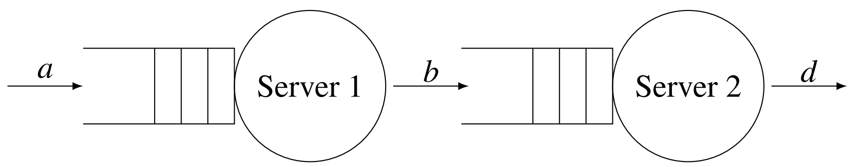
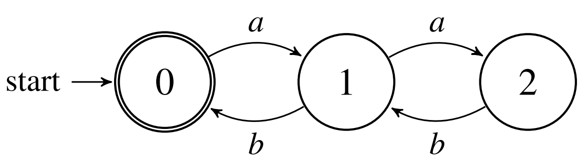
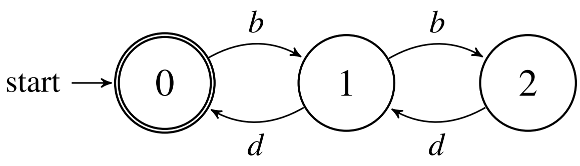


Figure 3.4: System composed by two queues in series

Figure 3.5 shows the individual automata, F_1 and F_2, which represents the queues of system illustrated in Figure 3.4.



(a) Server 1



(b) Server 2

Figure 3.5: Individual representation of the queues through automata

Applying the operation $F_1||F_2$ it possible to obtain the parallel composition, in which, represents the system dynamic. The 6-tuple $(\mathbf{X_1} \times \mathbf{X_2}, \mathbf{E_1} \cup \mathbf{E_2}, f, \Gamma_{1||2}, (x_{1_0}, x_{2_0}), X_{m_1} \times X_{m_2})$ of $F_1||F_2$ is composed by the following values:

- $\mathbf{X_1} \times \mathbf{X_2} = \{(0,0),(1,0),(2,0),(0,1),(1,1),(2,1),(0,2),(1,2),(2,2)\}$
- $\mathbf{E_1} \cup \mathbf{E_2} : \{a,b,d\}$
- $f : f((0,0),a) = (1,0), f((1,0),a) = (2,0), f((2,0),b) = (1,1), f((0,1),d) = (0,0), f((0,1),a) = (1,1), f((1,1),a) = (2,1), f((1,1),b) = (0,2), f((1,1),d) = (1,0), f((2,1),d) = (2,0), f((2,1),b) = (1,2), f((0,2),a) = (1,2), f((0,2),d) = (0,1), f((1,2),a) = (2,2), f((1,2),d) = (1,1)$ and $f((2,2),d) = (2,1)$;
- $\Gamma_{1||2} : \Gamma_{1||2}(0,0) = \{a\}, \Gamma_{1||2}(0,1) = \{a\}, \Gamma_{1||2}(2,0) = \{b\}, \Gamma_{1||2}(0,1) = \{a,d\}, \Gamma_{1||2}(1,1) = \{a,b,d\}, \Gamma_{1||2}(2,1) = \{b,d\}, \Gamma_{1||2}(0,2) = \{a,d\}, \Gamma_{1||2}(1,2) = \{a,d\}$ and $\Gamma_{1||2}(2,2) = \{d\}$;
- (x_{1_0}, x_{2_0}): $(0,0)$;
- $\mathbf{X_{m_1}} \times \mathbf{X_{m_2}}$: $\{(0,0)\}$.

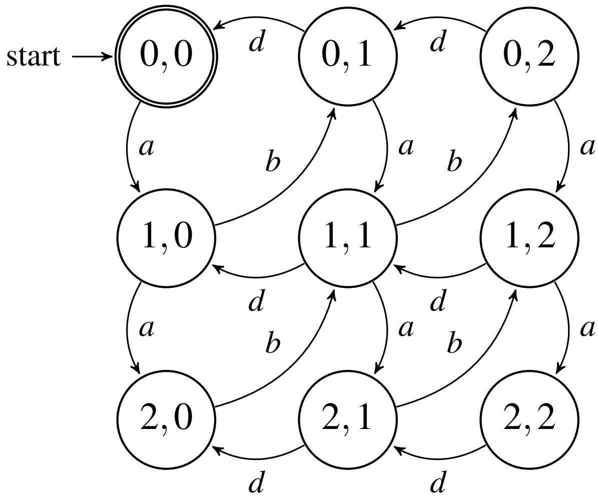


Figure 3.6: Automaton which represents the parallel composition $F_1||F_2$

Therefore, the full system, composed of two queues, can be designed by the automaton illustrated in Figure 3.6.

3.1.3 Timed Automata

As described previously, automata is an abstract instrument used for modeling the logical behavior of a system. However, many DES are endowed with a timing mechanisms. Taken this into account, a clock structure is required to compose the concepts about automata presented above. The fundamental notions about such mechanisms can be seen in details in (Alur and Dill, 1994).

A clock structure is defined by a lifetime sequence. For each state is checked which are the feasible transitions and the instant or occurrence. Thereby, taken into account the concepts of automata presented combined with such clock structure, it is possible to model the logical behavior of timed DES. Cassandras and Lafortune (2009) describe in details the logical dynamic of timed automata. Moreover, they point out that such clock structure is the base of many DES simulation tools. The following is a formal definition of timed automata.

Definition 3.1.3. A timed automata F is composed by 7-tuple $(\mathbf{X}, \mathbf{E}, f, \Gamma, x_0, \mathbf{X_m}, \mathbf{V_d})$. where:

- $\mathbf{X} = \{x_0, x_1, \ldots, x_{k-1}, x_k, x_{k+1}, \ldots\}$ is a set of countable states;
- $\mathbf{E}$ is a finite set of events associated with automata F;
- $f : \mathbf{X} \times \mathbf{E} \rightarrow \mathbf{X}$ is a transition function: $f(x, \alpha) = x'$ indicates that if the automaton is in state x and occur the event α ($\alpha \in \mathbf{E}$), instantaneously occur a transition to state x';
- Γ: $\mathbf{X} \rightarrow 2^{\mathbf{E}}$ is a function of feasible events: $\Gamma(x)$ denotes all actives events assuming that the automaton is in state x;

- x_0: is the initial state;
- $\mathbf{X_m}$: is the marked states ($\mathbf{X_m} \subseteq \mathbf{X}$);
- $\mathbf{V_d}$: Deterministic clock structure;

As it can be seen, this is the same as the untimed automata defined previously with only a few changes. The main difference is the just the timing mechanism. In order to present the key idea of timed automata, such mechanism is described.

3.1.3.1 Clock structure

A clock structure $\mathbf{V_d}$ that composes a timed automaton is defined by Equation (3.1). For each event α, $\mathbf{v}_\alpha$ denotes a lifetime sequence.

$$\mathbf{V_d} = \{\mathbf{v}_\alpha \ : \ \alpha \ \in \ \mathbf{E}\} \tag{3.1}$$

An event α, which compose the set $\mathbf{E}$, is characterized by clock sequence. Thus, the structure $\mathbf{v}_\alpha$ is formed of lifetimes. Equation (3.2) formalizes $\mathbf{v}_\alpha$. The label $v_{\alpha k}$ denotes the interval between the k-th and the $k-1$-th occurrence of event α assuming that α keeps active.

$$\mathbf{v}_\alpha = \{v_{\alpha 0}, v_{\alpha 1}, \ldots, v_{\alpha k-1}, v_{\alpha k}, v_{\alpha k+1}, \ldots\} \ \forall \ \alpha \in \ \mathbf{E} \tag{3.2}$$

Figure 3.7 shows the logical dynamic of an automaton with two timed events, α e β, which are described by the lifetime sequences: $\mathbf{v}_\alpha = \{v_{\alpha 1}, v_{\alpha 2}, \ldots\}$ e $\mathbf{v}_\beta = \{v_{\beta 1}, v_{\beta 2}, \ldots\}$.

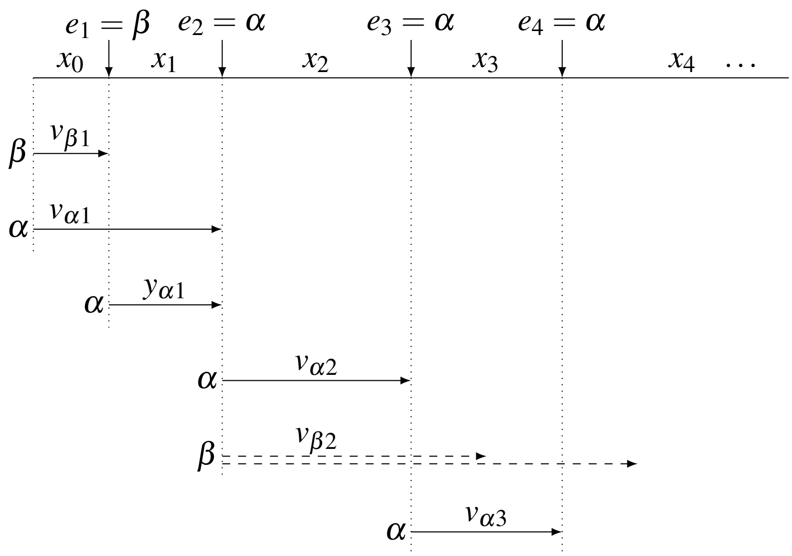


Figure 3.7: Dynamic of a timed automaton with $\mathbf{E} = \{\alpha, \beta\}$

Figure 3.7 shows a sample path in which the set of states is represented by $\{x_0, x_1, x_2, \ldots\}$ where the initial state is x_0. The label e_k represents the event occurred at instant k. From this dynamic it is possible note that α and β are feasible in the initial state ($\Gamma(x_0) = \{\alpha, \beta\}$). Consequently, both events are activated in this state. Once that $v_{\beta 1} < v_{\alpha 1}$, the first transition occurs due to the event β function. It means that $e_1 = \beta$. After this transition, the system change from state x_0 to state x_1 ($f(x_0, \beta) = x_1$). Since that in this new state only α is feasible, the event β is not activated. The lifetime of α is updated to $y_{\alpha 1} = v_{\alpha 1} - v_{\beta 1}$ because such event was activated in state x_0 and keeps feasible after transition to state x_1.

Once that the next event to occur is the one with the smallest clock value, $e_2 = \alpha$. Then, the system changes from state x_1 to state x_2 ($f(x_1, \alpha) = x_2$). The events α e β are feasible in state x_2 ($\Gamma(x_2) = \{\alpha, \beta\}$). Therefore, both events are activated. In this new condition, $v_{\alpha 2} < v_{\beta 2}$. Then, $E_3 = \alpha$. After this transition $f(x_2, \alpha) = x_3$. It means that just α is feasible ($\Gamma(x_3) = \{\alpha\}$) and this event is activated again. Regarding event β, two situations can happen. The first, assuming that $v_{\beta 2} < v_{\alpha 3}$, β is deactivated. The second situation indicates that $v_{\beta 2} > v_{\alpha 3}$. Then, β keeps activated and its occurrence (or not) depends on the next state.

3.1.3.2 Timing rules

Timed automata require a set of rules. The fundamental idea is to guarantee a correct analysis of the system evolution. Essentially, such set is a mechanism for selecting the next event, as following described.

1. Assuming that the system is in state x, the next event is that in the set $\Gamma(x)$ (feasible events in state x) with small clock value.
2. A event α is activated in the follow conditions:
   - α was the last event occurred and it remained feasible after the transition;
   - An event $\beta \neq \alpha$ was the last event occurred. In this instant, α was not feasible. However, the transition reached a state in which α is feasible.
3. An event α is deactivated when a event $\beta \neq \alpha$ occurs causing the transition to a state in which event α is not feasible.

Another principle of the timing rules is the fact that each event $\alpha \in \mathbf{E}$ is associated with a score $N_{\alpha,k}$. Assuming that the system starts at the instant t_0, $N_{\alpha,k}$ denotes the number of times that event α was activated in the interval $[t_0, t_k]$. Thus, taken into account such score and the timing rules presented, it is possible to analyze the system evolution over time. As defined in Equation (3.2), each event has a lifetime sequence. That way, the score of an event α is used as a pointer to α's lifetime sequence. Thus, when α is activated, its score specifies the next lifetime to be assigned to its clock.

3.1.3.3 Timing Dynamic

According to Cassandras and Lafortune (2009), the timing dynamic requires the follow notations:

- Current state k
  - x: current state of the system;
  - e: most recent event occurred;
  - t: time instant of e occurrence;
  - N_α: score of event α at instant t;
  - y_α: clock value of event α at instant t.
- next state $k+1$
  - x': next state;
  - e': next event;
  - t': instant of occurrence of e'
  - N'_α: new score of event α;
  - y'_α: new clock value of event α at instant t'.

Using these notations, the system dynamic is described by the follow steps:

1. Assuming that state x is known, to determine the set of feasible events $\Gamma(x)$.
2. $y^* = \min_{\alpha \in \Gamma(x)} \{y_\alpha\}$
3. $e' = \arg \min_{\alpha \in \Gamma(x)} \{y_\alpha\}$
4. $x' = f(x, e')$
5. $t' = t + y^*$
6. $y'_\alpha = \begin{cases} y_\alpha - y^*, & \text{if } \alpha \neq e' \text{ and } \alpha \in \Gamma(x) \\ v_{\alpha, N_\alpha + 1} & \text{if } \alpha = e' \text{ or } \alpha \notin \Gamma(x) \end{cases} \quad \alpha \in \Gamma(x')$

7. $N'_\alpha = \begin{cases} N_\alpha + 1, & \text{if } \alpha = e' \text{ or } \alpha \notin \Gamma(x) \\ N_\alpha & \text{otherwise} \end{cases} \quad \alpha \in \Gamma(x')$

Assuming that the system starts in state x_0, the initial clock value and score, for each event $\alpha \in \mathbf{E}$, are defined by Equation (3.3) and Equation (3.4).

$$y_\alpha = \begin{cases} v_{\alpha 1} & \text{if } \alpha \in \Gamma(x_0) \\ \text{undefined} & \text{otherwise} \end{cases} \tag{3.3}$$

$$N_\alpha = \begin{cases} 1 & \text{if } \alpha \in \Gamma(x_0) \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

As defined previously, $f(x,e)$ is the transition function of state x, given the occurrence of event e. Taken into account the sequence of steps which describes the system dynamic, it is possible see that the next event depends on the clock values ($y_\alpha \ \forall \ \alpha \in \mathbf{E}$) and the scores ($y_\alpha \ \forall \ \alpha \in \mathbf{E}$). Therefore, for a timed automaton, the transition function between states can be written by Equation (3.5).

$$x' = f^x(x, \mathbf{y}, \mathbf{N}, \mathbf{V_d}) \tag{3.5}$$

The terms $\mathbf{y}$ and $\mathbf{N}$ denote a set composed by the clock values (y_α) and a set formed by the values of score (N_α) at instant t, respectively, for all events $\alpha \in \mathbf{E}$. Let $\mathbf{N}'$ and $\mathbf{y}'$ be the same sets updated after a transition occurrence. Let $\mathbf{N}'$ and $\mathbf{y}'$ can be writen by Equations (3.6) and (3.7), respectively.

$$\mathbf{y}' = \mathbf{f}^{\mathbf{y}}(x, \mathbf{y}, \mathbf{N}, \mathbf{V_d}) \tag{3.6}$$

$$\mathbf{N}' = \mathbf{f}^{\mathbf{N}}(x, \mathbf{y}, \mathbf{N}, \mathbf{V_d}) \tag{3.7}$$

Summarily, a timed automaton is composed by x, $\mathbf{y}$ and $\mathbf{N}$. These terms are the 'minimum and complete' set which represents the system. Thus, Equations (3.5), (3.6) and (3.7) describe its dynamic and define the sequence of events occurrence. Figure 3.8 shows an abstraction of a timed automaton, in which, the input consist of its clock structure $\mathbf{V_d}$. For each transition k, the tuple (e_k, t_k) denotes the event and the instant of occurrence. The output is the sequence of events occurrence obtained from the system dynamic.

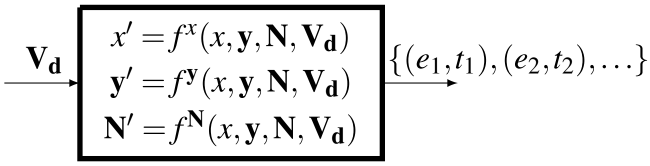


Figure 3.8

To exemplify, let's suppose the queue system illustrated in Figures 3.2 and 3.3. As described previously, such system is composed of $\mathbf{E} = \{a, d\}$. And let's suppose that events a and d are timed, in which, its lifetime sequence are $\mathbf{v_a} = \{1, 3, 1, 2.5, 3, \ldots\}$ and $\mathbf{v_d} = \{2, 7.0, \ldots\}$. The system in this condition can be classified as an timed automaton and represented by the 7-tuple:

- $\mathbf{X} = 0, 1, 2, 3, 4$;
- $\mathbf{E} = \{a, d\}$;
- $f : f(0, a) = 1, f(1, a) = 2, f(1, d) = 0, f(2, a) = 3, f(2, d) = 1, f(3, a) = 4, f(3, d) = 2$ and $f(4, d) = 3$;
- Γ: $\Gamma(0) = \{a\}$,$\Gamma(1) = \{a, d\}$, $\Gamma(2) = \{a, d\}$, $\Gamma(3) = \{a, d\}$, $\Gamma(4) = \{d\}$;
- $x_0 = 0$;
- $\mathbf{X_m} = \{0\}$ (just state 0).
- $\mathbf{V_d} = \{\{1.0, 2.5, 1.5, 1.5, 3.0, \ldots\}, \{1.75, 7.0, \ldots\}\}$

It was added to this queue system a clock structure which characterizes a timed automaton. To simplify, the timed sequence of events is composed only of the first occurrences. Thereby, for this structure, we have the following sequence of events:

$$\{(e_1 = a, t_1 = 1), (e_2 = d, t_2 = 2), (e_3 = a, t_3 = 4), (e_4 = a, t_4 = 5), (e_5 = a, t_5 = 7.5), (e_6 = d, t_6 = 9)\}.$$

Figure 3.9 shows the system dynamic taken into account the first six transitions.

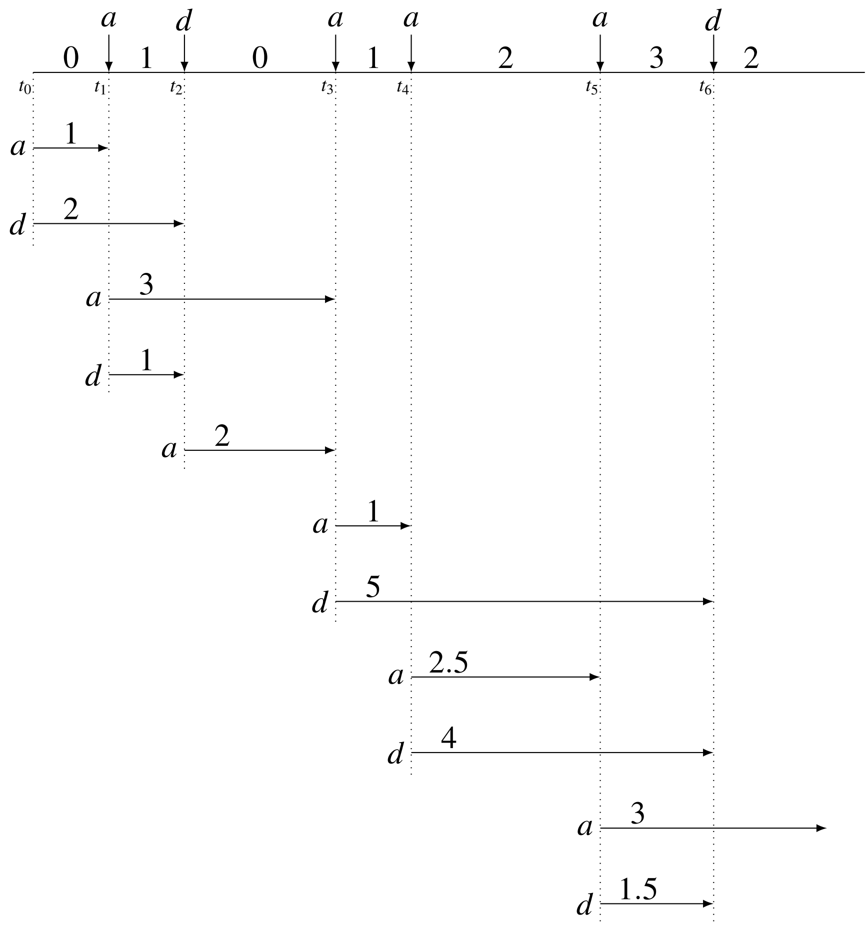


Figure 3.9: Dynamic of queue system illustrated in Figure 3.2

3.1.4 Stochastic timed automata

Many practical system are characterized by uncertain. According to Cassandras and Lafortune (2009), they always operate in environments which are constantly plagued by uncertainty. The authors also argue "this feature is especially true in dealing with DES, which, by their nature, often involve unpredictable human actions and machine failures". Regarding this system, events occur associated with some uncertain. For instance, essentially, a queue system is composed of random events, which are, the arrivals and departs. Each random event follows a probability distribution which describes its behavior. Thus, as can be seen in the following definition, probabilistic properties are considered in a stochastic timed automaton.

Definition 3.1.4. The 6-tuple $(\mathbf{X}, \mathbf{E}, \Gamma, p, p_0, \mathbf{V_s})$ denotes the stochastic timed automaton F, where:

- $\mathbf{X}$ is a finite set of states;
- $\mathbf{E}$ is a finite set of events associated with F;

- Γ: $\mathbf{X} \leftarrow 2^{\mathbf{E}}$ is a function of feasible events. $\Gamma(x)$ denotes all actives events assuming that the automaton is in state x;
- $p(x';x;e)$ is the transition probability from state x to $x^{'}$ given the occurrence of event e, and such that $p(x^{'};x;e) = 0 \; \forall \; e \notin \Gamma(x)$;
- $p_0(x)$ is the probability $P[x_0 = x] \; \forall \; x \in \mathbf{X}$, that is, the a probability of the system starts in state x;
- $\mathbf{V_s}$: is a Stochastic clock structure.

It is possible to see that the clock structure, the transitions, and the initial state are described by random variables. Such condition defines a stochastic behavior for the automaton. Taken into account these descriptions, it possible to model the logical dynamic of queue system which is plagued by uncertainties through stochastic timed automata. However, for conducting this kind of analysis, it is necessary an understanding of stochastic process principles. Bertsekas and Tsitsiklis (2002) present theoretical fundamental of probability and stochastic processes. Some of these concepts are discussed in the sequence.

3.1.4.1 Fundamental concepts of stochastic processes

Let X be a random variable, the stochastic process with describes X is defined by Equation (3.8).

$$\{X(t)\} = [X(t_0), \ldots, X(t_{k-1}), \ldots, X(t_k), \ldots, X(t_{k+1}), \ldots, X(t_n)] \tag{3.8}$$

Where, $X(t_k)$ denotes a random values for X at the instant t_k. For each possible value of x_k, it is defined, from the sequence $\mathbf{x} = [x_0, \ldots, x_{k-1}, x_k, x_{k+1}, \ldots, x_n]$, the joint cdf (cumulative distribution function) $C_X(.)$ is given by Equation (3.9).

$$C_X(x_0, \ldots, x_n; t_0, \ldots, t_n) = P[X(t_0) \leq x_0, \ldots, X(t_n) \leq x_n] \tag{3.9}$$

For the analysis of stochastic sequence it is necessary to specify its joint cdf. According to Cassandras and Lafortune (2009), this is an extremely tedious task in arbitrary stochastic processes. However, many stochastic processes are classified as stationary and/or independent. Moreover, in many cases can be described by Markovian properties. “For many of the stochastic processes which turn out to be useful models of systems we encounter in practice, there are relatively simple means to accomplish this task" (Cassandras and Lafortune, 2009). In the sequence, it is present the mentioned classification of stochastic processes.

Definition 3.1.5. Stationary process - The dynamic behavior of stationary process is time invariant. A stochastic process is stationary if only if satisfies Equation (3.10), where τ denotes a time variation.

$$C_X(x_0,\ldots,x_n;t_0+\tau,\ldots,t_n+\tau) = C_X(x_0,\ldots,x_n;t_0,\ldots,t_n) \;\; \forall\, \tau \in \mathbb{R}. \tag{3.10}$$

Definition 3.1.6. Independent process - A process is defined as independent if all random variable of its sequence are mutually independent. In a process classified as independent, the joint cdf can be described by Equation (3.11).

$$C_X(x_0,\ldots,x_n;t_0,\ldots,t_n) = P[X(t_0) \leq x_0] \times \ldots \times P[X(t_n) \leq x_n] \tag{3.11}$$

Moreover,an independent stochastic process is formed of a sequence, in which, all random variables are drawn from a unique distribution. Thus, in this case, Equation (3.11) can be generalized by $C_X(x_k) = P[X(t_k) \leq x_k]$. It means that the stochastic behavior of the process is captured by a single cdf. Then, the stochastic process is refereed by a sequence of *iid* random variables.

Definition 3.1.7. Markovian process - A Markovian process is formally described by Equation (3.12).

$$P[X(t_k+1) = x_j|X(t_k) = x_i] = P[Z(t_k+1) = x_j|X(t_k) = x_i, X(t_{k-1}) = x_{i-1},\ldots,X(t_0) = x_0] \tag{3.12}$$

Equation (3.12) implies that the transition probability from current state to next state does not depends of the past occurrence. This Memoryless properties indicates that the entire past history is summarized in the current state. Such class of stochastic process is detailed in the next section of this chapter.

3.1.4.2 Stochastic clock structure

As well as in a timed automaton (deterministic), a stochastic clock structure is composed of a lifetime sequence. Let $\{V_{\alpha k}\} = \{V_{\alpha 1}, V_{\alpha 2},\ldots\}$ be the lifetime sequence of event α. for a stochastic clock structure, each $V_{\alpha k}$ is a random variable described by some probability distribution.

A stochastic clock structure $\mathbf{V_s}$ of a stochastic timed automaton is defined by Equation (3.13). Thus, for each event α, there is a random variable V_α.

$$\mathbf{V_s} = \{V_\alpha \ : \ \alpha \ \in \ \mathbf{E}\} \tag{3.13}$$

Let $\{V_{\alpha k}\}$ be the lifetime sequence of α. Assuming that $\{V_{\alpha k}\}$ is composed of clock values *iid*, the random variable V_α is drawn from a single distribution. Equation (3.14) satisfies this condition.

$$V_\alpha(t) = P[V_\alpha \leq t] \tag{3.14}$$

The function $V_\alpha(t)$ in Equation (3.14) represents the lifetime sequence $\{V_{\alpha k}\}$. Since that V_α is the random variable of event α, $P[V_\alpha \leq t]$ is the probability of the instant of occurrence be equal of less than t.

For a clock structure $\mathbf{V_s}$ featured by the set of events $\mathbf{E}$, in which, each $\alpha \in \mathbf{E}$ is described by a lifetime sequence $\{V_{\alpha k}\}$ *iid*, Equation (3.13) can be rewritten by Equation (3.15).

$$\mathbf{V_s} = \{V_\alpha(t) \ : \ \alpha \ \in \ \mathbf{E}\} \tag{3.15}$$

3.1.4.3 *Generalized Semi-Markov Process* - GSMP

GSMP (Generalized Semi-Markov Process) is a stochastic process generated by a stochastic timed automaton. The Markovian property of GSMP is the Memoryless properties described early. "The Markovian aspect of the GSMP stems from the fact that process state behaves like a Markov chain at state transition points"(Cassandras and Lafortune, 2009). However, it depends on the time stay in the state. In this turn, it follows arbitrary distributions. Such condition implies that in a GSMP the transitions between processes occur respecting the principles of a Markovian process. But the time stay in each state is a random variable associated with a generalized probability distribution.

Definition 3.1.8. "A stochastic process is a mathematical model of a probabilistic experiment that evolves in time and generates a sequence of numerical values"(Bertsekas and Tsitsiklis, 2002). Thus, GSMP can be seen as a stochastic process $\{X(t)\}$ generated by a stochastic timed automaton composed by a set of states $\mathbf{X}$.

A stochastic process inherent to a stochastic timed automaton is by definition a GSMP. Taken into account that the concepts of stochastic timed automaton are the base of DES simulation, Cassandras and Lafortune (2009) argue that "understanding the mechanism by which stochastic timed automata generate a GSMP is tantamount to understanding the basic principles of discrete-event computer simulation". Thus, from this analogy, it possible to observe the features of the DES system modeled.

One of this features consist of verifying the more frequent state. Let $\{X(t)\}$ be a GSMP generated by a stochastic timed automaton $(\mathbf{X}, \mathbf{E}, \Gamma, p, p_0, \mathbf{V_s})$, the goal is to compute the probability of a system be in state x at the instant t, $P[X(t) = x]$. Therefore, from a GSMP it is possible to compute such measure.

3.1.4.4 Poisson process seen as a GSMP

Poisson process is a fundamental counting process commonly used to represent arrival that evolves in continuous time. According to Bertsekas and Tsitsiklis (2002), the Poisson process is a continuous-time analog of the Bernoulli process [1] and applies to situations where there is no natural way of dividing time into discrete periods.

In a stochastic timed automaton, $N_\alpha(t)$ denotes the counting value of event α at instant t. Due to the counting nature of the process, $N_\alpha(t_0) \leq N_\alpha(t_1) \leq N_\alpha(t_2) \leq N_\alpha(t_k)$ para $t_0 \leq t_1 \leq t_2 \leq t_k$. Equation (3.16) defines the measure $N_\alpha(t_k, t_{k+1})$ which denotes the number of occurrence of event α into a arbitrary interval $(t_{k+1} - t_k)$.

$$N_\alpha(t_k, t_{k+1}) = N_\alpha(t_{k+1}) - N_\alpha(t_k), \;\; k = 0, 1, 2, \ldots \tag{3.16}$$

Thus, a Poisson process is used to describe how $N_\alpha(t_k)$ evolves, given that α is characterized by the following properties.

Definition 3.1.9. A Poisson process is featured by the properties:

- Time-homogeneity - For the time intervals (t_{k-1}, t_k) e (t_k, t_{k+1}) of same size, the number of occurrence are equally likely. It means that $N_\alpha(t_{k-1}, t_k) = N_\alpha(t_k,\ t_{k+1}) = n$. Such property implies that the occurrence of event α in any time interval of same size follow the same probability law.

- Independence - This property indicates the number of occurrences $N_\alpha(t_k, t_{k+1}) = n$ of event α into the time interval (t_k, t_{k+1}) is independent of the history outside this interval. In other words, the variables $N_\alpha(t_{k-1}, t_k), N_\alpha(t_k, t_{k+1}), N_\alpha(t_{k+1}, t_{k+2}), \ldots$ are mutually independents for any time instant t_k.

- Exponential distribution - By definition, the inter-events times (t_k, t_{k+1}) in a Poisson process are exponentially distributed with rate λ. Thus, $V_\alpha(t) = 1 - \mathrm{e}^{-\lambda t}$.

Bertsekas and Tsitsiklis (2002) and Cassandras and Lafortune (2009) show in details how these properties are analytically developed.

[1] Bernoulli process is a sequence $\{X_1, X_2, X_3, \ldots\}$ of independent Bernoulli random variables, in which, each X_k has a probability p of success and a probability $1 - p$ of fail (Bertsekas and Tsitsiklis, 2002).

Many practical systems of a stochastic nature are characterized by described properties. According to Cassandras and Lafortune (2009), there are several reasons why the Poisson counting process is an essential element in stochastic modeling and analysis. One of this reason is the fact that it is possible to simplify a GSMP, generated by a stochastic timed automaton, in a Markov Chain. Next section of this chapter discusses this subject.

The 6-tuple $(\mathbf{X}, \mathbf{E}, \Gamma, p, p_0, \mathbf{V_s})$ depicts a Poisson process seen as a GSMP, generated by a stochastic timed automaton composed by a unique event, where:

- $\mathbf{X} = \{0, 1, 2, \ldots\}$;
- $\mathbf{E} = \{a\}$;
- Γ: $\Gamma(x) = \{a\} \ \forall \, x \in \mathbf{X}$;
- $p(x+1; x; a) = 1 \ \forall \, x \in \mathbf{X}$;
- $\mathbf{V_s} = \{V_a(t)$ where $V_a(t) = 1 - \mathrm{e}^{-\lambda t}\}$.

Given that $\mathbf{X} = \{0, 1, 2, \ldots\}$ is the set of states, $\{X(t)\}$ is the GSMP generated by the stochastic timed automaton. It can be seen that this representation is composed by a unique event. However, from this analogy, it is possible to infer a superposition of Poisson processes for stochastic timed automata composed of more than one event.

Definition 3.1.10. The superposition of m Poisson processes, for all m mutually independent and with parameters $\lambda_\alpha \ \forall \ \alpha \in \mathbf{E}$, is also a Poisson process with parameters Λ. According to Cassandras and Lafortune (2009), the term Λ must satisfies Equation (3.17).

$$\Lambda = \sum_{\alpha \in \mathbf{E}} \lambda_\alpha \tag{3.17}$$

To exemplify, lets supposes a infinity queue system seen as a stochastic timed automaton and represented by the 6-tuple $(\mathbf{X}, \mathbf{E}, \Gamma, p, p_0, \mathbf{V_s})$, where:

- $\mathbf{X} = \{0, 1, 2, \ldots\}$;
- $\mathbf{E} = \{a, d\}$;
- Γ: $\Gamma(0) = \{a\}$ and $\Gamma(x) = \{a, d\} \ \forall \, x > 0$;
- $p(x+1; x; a) = 1 \ \forall \, x \geq 0$ and $p(x-1, x; d) = 1 \ \forall \, x > 0$;
- $p_0(0) = 1$ e $p_0(x) = 0 \ \forall \, x > 0$;

- $\mathbf{V_s} = \{V_a(t), V_d(t)\}$, where $V_a(t) = 1 - \mathrm{e}^{-\lambda_a t}$ and $V_d(t) = 1 - \mathrm{e}^{-\lambda_d t}$.

From this representation, it is possible to see that the automaton is deterministic in terms of transition because, given the occurrence of an event, the next state is known. Moreover, assuming that the events a and d are independent, the automaton is composed of a Poisson clock structure $\mathbf{V_s}$. Regarding $\mathbf{V_s}$, the lifetime sequence of events are Poisson process with rates λ_a and λ_d. Taken into account the concepts presented in this section, this automaton can by seen as a Markov Chain.

3.2 Concepts of Markov Chain

In 1913, the Russian mathematician A. A. Markov founded a new branch of probability theory by applying mathematics to poetry (Hayes et al, 2013). The proposal consisted of textual analysis of the Alexander Pushkin's novel in verse *Eugene Onegin*. Although his idea has not changed the understanding or appreciation of the poem, the methodology created by Markov extended probability in a new direction. Essentially, Markov proved that the independence of random variables was not a necessary condition for the validity of the weak law of large numbers and the central limit theorem (Basharin et al, 2004).

Nowadays, this theory is used in many scientific fields. The concepts are presented by Cooper (1981), (Bertsekas and Tsitsiklis, 2002) and Cassandras and Lafortune (2009). Even though the apparently limitation of using Markov Chains, a DES model designed through Markovian properties is a powerful tool to analyze stochastic problems. Lee and Lee (2014) applies Markovian properties for modeling stochastic signals of switching patterns. The proposed technique uses a hidden Markov model based on jump linear system (Shi and Li, 2015). Casella et al (2016) propose a computational stochastic model, based on discrete-time Markov process, to simulate the dynamic behavior of residential loads in a power system.

Cui et al (2010) present a survey on the developments and applications of finite Markov Chain in the reliability field. Ruessink (2006) applies Markovian properties to find best-fit parameter values and their uncertainty in nearshore process models. Yu et al (2004) demonstrate the application of Markov chain models to valuate generation assets within deregulated electricity markets. Ye et al (2004) present a cyber-attack detection technique through anomaly-detection, and discusses the robustness of the modeling technique employed. Such technique uses a Markov chain model to represent a profile of event transition in a usual operating condition of a computer and network system.

Yin and Zhao (2008) present a comprehensive model for collaboration trust maintenance in supply chain, integrated adaptive N-stage Markov chain. In this study, the transferring process of collaboration trust is interpreted through Markovian properties. Wang and Li (2011) show the basic principles of Markov Chain theory and its application to market occupation rate and promotion strategy. Hongyan and Zhong (2009) indicate that there is an aging trend in Shanghai. Such conclusion was acquired from a forecast model of Markov chain.

The history shows us an evaluation of Markov Chains applications, especially in the queue context. According to Bolch et al (2006), Markov process constitutes the fundamental theory underlying the concepts of queuing systems. In a load-haulage system, each process can be mapped, in principle, as unique Markov chain. Due to Cassandras and Lafortune (2009), queuing systems can be seen as stochastic timed automata. Before we present the connection between this class of automata and Markov chain, we show some fundamental definitions.

3.2.1 Markov chain definitions

According to Bolch et al (2006), Markov processes provide very flexible, powerful, and efficient means for the description and analysis of dynamic system properties. Besides, they constitute the fundamental theory underlying the concept of queue systems. Essentially, queue systems can be designed as a Markov process and, hence, it can be mathematically evaluated. Bolch et al (2006) point out that fundamental properties of queue systems are commonly proved in terms of the underlying Markov process. Therefore, many performance measures can be derived. Bertsekas and Tsitsiklis (2002) presents concepts of discrete-time Markov chain. Fundamentally, a stochastic process $\{X(t_k)\}$ classified as Markovian is also defined as a discrete-time Markov chain whether its state space is discrete.

Cassandras and Lafortune (2009) argue that the main feature of a discrete Markov chain is that its stochastic behavior is described by transition probabilities of the form. Equation (3.18) describes $p_{ij}(t_{k+1})$, in which, it is the transition probability from state i to state j at instant t_k.

$$p_{ij}(t_{k+1}) = P[X(t_k+1) = j|X(t_k) = i] \tag{3.18}$$

As it can seen in sequence, from this measure, it is possible to determine the probability of being at any state at any time instant.

3.2.2 Homogeneous Markov Chains

In a Homogeneous Markov Chains, the transition probability $p_{ij}(t_{k+1})$ is time invariant. It means that, for any time instant t_k, $p_{ij}(t_{k+1})$ is independent of k. Therefore, Equations (3.18) can be simplified as (3.19) fos this kind of Markov chain.

$$p_{ij} = P[X' = j|X = i], \tag{3.19}$$

where X and X' denote the current and the next state. The label p_{ij} represents the transition probability from state i to state j.

Taken this into account, a Markov chain, in general, can be specified as a set of states $\mathbf{X}$ and a set of transitions probabilities. Equation (3.20) shows a transition probability matrix of a Markov chain with L states.

$$\mathbf{M_p} = \begin{bmatrix} p_{00} & p_{011} & p_{02} & \cdots & p_{0L} \\ p_{10} & & & \cdots & p_{1L} \\ \vdots & & & & \vdots \\ & & & & \\ p_{L0} & & & & p_{LL} \end{bmatrix} \tag{3.20}$$

Moreover, the transitions probabilities to each state of the system must satisfies Equation (3.21).

$$\sum_{j \in \mathbf{X}} p_{ij} = 1 \; \forall \; i \in \mathbf{X} \tag{3.21}$$

3.2.3 Chapman-Kolmogorov equation

One of the goals of a Markov chain is to be able to analyze the probability of the system being in any state at a given instant. According to Gardiner (1985), this analysis can be done, in a simple manner, by the Chapman-Kolmogorov. Let $\pi_i(t_k)$ be the probability of the system being in state i at instant t_k. In a finite Markov chain of size L, $\boldsymbol{\pi}(t_k)$ denotes a vector composed of each $\pi_i(t_k) \; \forall \; i \in \mathbf{X}$, such described by Equation (3.22).

$$\boldsymbol{\pi}(t_k) = [\pi_{i=1}(t_k), \ldots, \pi_{i=L}(t_k)] \;\; i \in \mathbf{X} \tag{3.22}$$

Regarding this notation, Equation (3.23) is a matrix representation for the Chapman-Kolmogorov equation.

$$\boldsymbol{\pi}(t_k + k) = \boldsymbol{\pi}(t_k) \cdot \mathbf{M_p}^k \tag{3.23}$$

Thus, $\boldsymbol{\pi}(t_k + k)$ is the state probability vector after k discrete-time instants. Assuming that just one event occurs in each instant, k can be understood as the number of transitions.

3.2.4 Example- Application of Chapman-Kolmogorov equation

To exemplify, lets consider a dynamic system described by an automaton. Figure 3.10 illustrates this automaton. Given that, at the discrete-time instant t_k, the system being in state 1. We are interested in computing the probability of the system being in state 2 after $\kappa = 3$ transitions. That is, $\pi_2(t_k + 3)$.

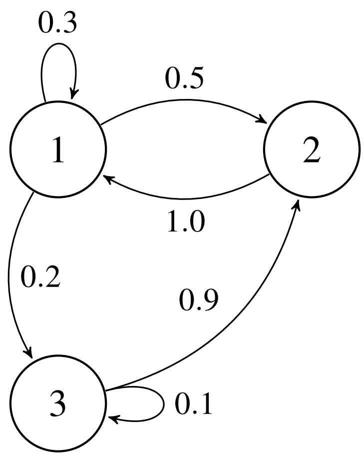


Figure 3.10: State transition diagram

The computation of $\pi_2(t_k + 3)$ without the use of Chapman-Kolmogorov equation consists of analyzing all possibilities to reach state 2 from state 1 after $\kappa = 3$ transitions. Let $\{X(t_k)\} = \{X_1, X_2, X_3\}$ be a sequence of states after the first $\kappa = 3$ transitions and let X_0 be the current state at discrete-time instant t_k. Thus, $\pi_2(t_k + 3)$ is obtained by the follow way:

$$\begin{aligned}
\pi_2(t_k+3) \;=\; & P[X_1 = 1|X_0 = 1] \times P[X_2 = 1|X_1 = 1] \times P[X_3 = 2|X_2 = 1] + \\
& P[X_1 = 2|X_0 = 1] \times P[X_2 = 1|X_1 = 2] \times P[X_3 = 2|X_2 = 1] + \\
& P[X_1 = 1|X_0 = 1] \times P[X_2 = 3|X_1 = 1] \times P[X_3 = 2|X_2 = 3] + \\
& P[X_1 = 3|X_0 = 1] \times P[X_2 = 3|X_1 = 3] \times P[X_3 = 2|X_2 = 3]
\end{aligned}$$

Then,

$$\begin{aligned}
\pi_2(t_k+3) &= 0.3 \times 0.3 \times 0.5 + 0.5 \times 1.0 \times 0.5 + 0.3 \times 0.2 \times 0.9 + 0.2 \times 0.1 \times 0.9 \\
&= 0.367
\end{aligned}$$

The same result can be computed, in a simple manner, using the Chapman-Kolmogorov equation. Let $\mathbf{M_p}$ be the transition matrix of the same automaton:

$$\mathbf{M_p} = \begin{bmatrix} 0.3 & 0.2 & 0.5 \\ 1.0 & 0 & 0 \\ 0 & 0.9 & 0.1 \end{bmatrix}$$

Thus, since that the 1 is the current state at instant t_k, $\boldsymbol{\pi}(t_k) = [1,0,0]$. Taken into account the probability notation shown, $\boldsymbol{\pi}(t_k+3) = [\pi_1(t_k+3), \pi_2(t_k+3)\pi_3(t_k+3)]$. Applying Equation (3.23), it is obtained:

$$\boldsymbol{\pi}(t_k+3) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0.3 & 0.2 & 0.5 \\ 1.0 & 0 & 0 \\ 0 & 0.9 & 0.1 \end{bmatrix}^3 = \begin{bmatrix} 0.507 & 0.367 & 0.126 \end{bmatrix}$$

Once that the second position of the vector $\boldsymbol{\pi}(t_k+3)$ denotes $\pi_2(t_k+3)$, the answer is $\pi_2(t_k+3) = 0.367$.

3.2.5 Stationary probability

As seen early, from the Chapman-Kolmogorov equation, it is easy to compute the state probability in any discrete-time instant. However, for a special class, it possible to get a complete analysis of systems represented by Markov chain. According to Cassandras and Lafortune (2009), a system modeled as a Markov chain is allowed to operate for a sufficiently long period of time so that the state probabilities can reach some fixed values which no longer vary with time. Therefore, it is important to take into account the ‘steady-state convergence theorem presented by Bertsekas and Tsitsiklis (2002).

Theorem 1. *Steady-State Convergence Theorem - Consider a Markov chain with a single recurrent class, which is aperiodic*[2]*, Equation (3.24) is valid.*

$$\pi_i = \lim_{\kappa \to \infty} \pi_i(\kappa)\ \forall\ i \in \mathbf{X} \tag{3.24}$$

The label π_i denotes the stationary state od state i. The steady-state probabilities π_i sum to 1 and form a probability distribution on the state space, called the stationary distribution of the chain (Bertsekas and Tsitsiklis, 2002). Assuming a Markov chain featured by definitions of steady-state convergence theorem. Then, when the discrete-time instant $\kappa \to \infty$, it is reached the stationary state. In this condition, the state probabilities of the system are independent of κ.

[2]Definitions of an aperiodic Markov chain with a single recurrent class are given in (Bertsekas and Tsitsiklis, 2002)

Equation (3.25) represents the Chapman-Komogorov equation for stationary distribution of the chain.

$$\boldsymbol{\pi} = \boldsymbol{\pi} \cdot \mathbf{M_p} \tag{3.25}$$

where $\boldsymbol{\pi}$ denotes the probability vector composed if $\pi_i \; \forall \; x_i \in \mathbf{X}$, such show in Equation (3.26).

$$\boldsymbol{\pi} = [\pi_{i=1}, \pi_{i=2}, \ldots] \;\; i \in \mathbf{X} \tag{3.26}$$

The values of $\boldsymbol{\pi}$ must satisfies the linear Equation (3.25) and the Equation (3.27).

$$\sum_{i \in \mathbf{X}} \pi_i = 1 \tag{3.27}$$

3.2.6 Example - Computing the stationary probability

Lets suppose the same automaton shown in Figure 3.10. Regarding the Markovian properties presented by Bertsekas and Tsitsiklis (2002), this automaton is a aperiodic Markov chain with a single recurrent class. Therefore, the steady-state convergence theorem is valid.

For this example, the stationary distribution can be obtained by Equation (3.25). Thus:

$$\begin{bmatrix} \pi_1 & \pi_2 & \pi_3 \end{bmatrix} = \begin{bmatrix} \pi_1 & \pi_2 & \pi_3 \end{bmatrix} \cdot \begin{bmatrix} 0.3 & 0.2 & 0.5 \\ 1.0 & 0 & 0 \\ 0 & 0.9 & 0.1 \end{bmatrix}$$

Taken into account Equation (3.27), the stationary probabilities π_1, π_2 and π_3 are determined by solving the linear system:

$$\begin{aligned} \pi_1 &= 0.3\pi_1 + \pi_2 \\ \pi_2 &= 0.2\pi_1 + 0.9\pi_3 \\ \pi_3 &= 0.5\pi_1 + 0.1\pi_3 \\ 1 &= \pi_1 + \pi_2 + \pi_3 \end{aligned}$$

Thus, the stationary distribution of the chain is $\boldsymbol{\pi} = [\pi_1 = 0.443, \pi_2 = 0.310, \pi_3 = 0.246]$.

3.2.7 Stochastic timed Automata seen as Markov chain

As shown in the previously section, a stochastic timed automaton composed by a Poisson clock structure can be simplified in a Markov chain. In this case, we are dealing with a continuous-time case. According to Cassandras and Lafortune (2009), we define a chain to be homogeneous if all transition functions are independent of the absolute time instant. For such system, the transition between states is also described in terms of rates. From the Poisson process definition, it is necessary that the inter-event times are exponentially distributed. The physical interpretation of this definition is detailed by Cassandras and Lafortune (2009).

Accordingly, Cassandras and Lafortune (2009) argue that the transition probability Matrix $\mathbf{M_p}$ of a Markov chain has the same information of a transition rate matrix. Therefore, for a continuous Markov chain, the stationary distribution of the chain can be obtained using the transition rate matrix $\mathbf{M_q}$. As a result, a Markov chain which represents a stochastic timed automaton is composed by $\mathbf{M_q}$, besides the set of states $\mathbf{X}$. Equation (3.28) defines $\mathbf{M_q}$.

$$\mathbf{M_q} = \begin{bmatrix} q_{00} & q_{011} & q_{02} & \cdots & q_{0L} \\ q_{10} & & & \cdots & q_{1L} \\ \vdots & & & & \vdots \\ & & & & \\ q_{L0} & & & & q_{LL} \end{bmatrix} \tag{3.28}$$

Where L denotes the number of states of states and q_{ij} the *instantaneous* transition rate from state i to state j. Equation (3.29) describes q_{ij}.

$$q_{ij} = \begin{cases} \lambda_\alpha & \text{if } i \neq j \\ -\sum\limits_{\substack{i \neq j \\ j=0}}^{L} q_{ij} & \text{otherwise } (i = j) \end{cases} \tag{3.29}$$

The term λ_α represents the occurrence rate of an event α, described by Poisson rules, which generates the transition from state i to state j. Let's consider a small variation $\Delta\tau$ in the Markov chain dynamic. From the physical interpretation, the main diagonal ($i = j$) terms of the matrix $\mathbf{M_q}$ denote the total event rate that characterizes the states. Otherwise, the off-diagonal terms ($i \neq j$) indicate the instantaneous rate between states.

For this kind of analysis, it is necessary to consider the steady-state convergence theorem. Regarding a continuous Markov chain, the Chapman-Kolmogorov Equation (3.22) must be interpreted by Equation (3.30).

$$\frac{d\boldsymbol{\pi}(t)}{dt} = \boldsymbol{\pi}(t) \cdot \mathbf{M_q} \tag{3.30}$$

Then, taken into account the mentioned theorem and the fact that the transitions between states are independent of the absolute time instant, for $t \to \infty$, the derivative $d\boldsymbol{\pi}(t)/dt = 0$. Thereby, if Equation (3.24) is verified, the stationary probability vector $\boldsymbol{\pi}$ must satisfy Equation (3.31) and Equation (3.27) (Cassandras and Lafortune, 2009).

$$\boldsymbol{\pi} \cdot \mathbf{M_q} = 0 \tag{3.31}$$

3.2.8 Example - Computing the stationary probability of a stochastic timed automata seen as Markov chain

Lets suppose the stochastic timed automaton described by the 6-tuple $(\mathbf{X}, \mathbf{E}, \Gamma, p, p_0, \mathbf{V_s})$. For simplicity, it was defined that, given the occurrence of any event, the next state is known. The term $p(.)$ represents such condition.

- $\mathbf{X} = \{1,2,3\}$;
- $\mathbf{E} = \{a,b,c\}$;
- Γ: $\Gamma(1) = \{a\}$, $\Gamma(2) = \{b\}$ e $\Gamma(3) = \{a,c\}$;
- $p(2;1;a) = 1$, $p(3;2;b) = 1$, $p(3;2;c) = 1$, $p(2;3;a) = 1$ and 0 otherwise;
- $p_0(1) = 1$, $p_0(2) = 0$ and $p_0(3) = 0$;
- $\mathbf{V_s} = \{V_a(t), V_b(t), V_c(t)\}$, where $V_a(t) = 1 - \mathrm{e}^{-\lambda_a t}$, $V_b(t) = 1 - \mathrm{e}^{-\lambda_b t}$ e $V_c(t) = 1 - \mathrm{e}^{-\lambda_c t}$.

This automaton is composed of a Poisson clock structure, in which, the events a, b and c follow exponential distribution with rates λ_a λ_b and λ_c, respectively.

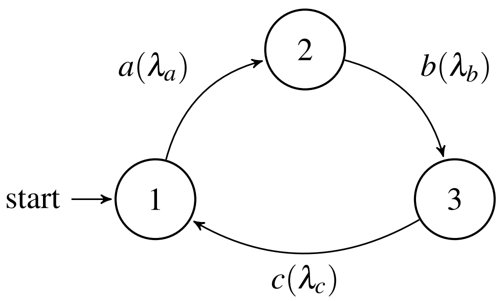


Figure 3.11: Stochastic timed automaton with 3 states

This automaton generates the Markov chain illustrated in Figure 3.12.

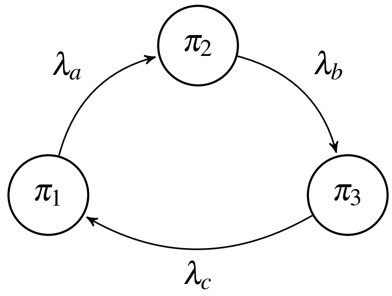


Figure 3.12: Transition diagram of the automaton shown in Figure 3.11.

Taken into account Equation (3.29), the transition matrix $\mathbf{M_q}$ complies with Equation (3.32).

$$\mathbf{M_q} = \begin{bmatrix} -\lambda_a & \lambda_a & 0 \\ 0 & -\lambda_b & \lambda_b \\ \lambda_c & 0 & -\lambda_c \end{bmatrix} \tag{3.32}$$

Let's suppose that the values of λ_a λ_b and λ_c be, respectively 0.6, 0.7, e 0.4. Then, Matrix $\mathbf{M_q}$ is formed by:

$$\mathbf{M_q} = \begin{bmatrix} -0.6 & 0.6 & 0 \\ 0 & -0.7 & 0.7 \\ 0.4 & 0 & -0.4 \end{bmatrix}$$

Using $\mathbf{M_q}$ in Equation (3.31):

$$\begin{bmatrix} \pi_1 & \pi_2 & \pi_3 \end{bmatrix} \times \begin{bmatrix} -0.6 & 0.6 & 0 \\ 0 & -0.7 & 0.7 \\ 0.4 & 0 & -0.4 \end{bmatrix} = 0$$

Including the equality Equation (3.27) in the system, the stationary distribution of this Markov chain is the solution of the linear system:

$$\begin{aligned} 0.4\pi_3 - 0.6\pi_1 &= 0 \\ 0.6\pi_1 - 0.7\pi_2 &= 0 \\ 0.7\pi_2 - 0.4\pi_3 &= 0 \\ \pi_1 + \pi_2 + \pi_3 &= 1 \end{aligned}$$

As a result, $\boldsymbol{\pi} = [\pi_1 = 0.298, \pi_2 = 0.255, \pi_3 = 0.447]$.

3.2.9 Superposition of automata seen as Markov chain

A full DES model can be seen as the parallel composition of many stochastic timed automata. For this system, a clock structure $\mathbf{V_s}$ describes how the DES model evolves in reason of occurrence of events in the time. Assuming that all automata can be described by Markovian properties, the full DES model can be simplified in a Markov chain.

In this case, the stationary probability of each state must be represented by the M-tuple $\pi(n_0,\ldots,n_M)$, in which, M denotes the number of automaton that composes the system. Thus, the stationary probabilities are determined by solving the linear system (3.31).

The drawback of this representation the fact that the cardinality of state space grows drastically. Consequently, to compute the stationary distribution can be a hard task, even impossible. Despite this limitation, the history shows us a scientific effort to extend the application of Markovian properties, especially in the queue context. According to Bolch et al (2006), many numerical solution methods have been developed to get important performance measures without resorting to the underlying state space. Thus, the next chapter presents concepts and the mechanisms used to derive such measures.

CHAPTER 4: QUEUING NETWORK

"One of the least understood classes of operations problems is that concerned with the design, loading, and, especially, the scheduling of discrete, statistically varying flows through complex networks."
James R. Jackson

This Chapter presents essential knowledge of queuing network. The aim is to establish a solid connection with analytical approximation methods that deal with stochastic DES. As discussed in Chapter 2, the target problem is a cyclic production system of stochastic nature. Therefore, we are interested in looking for tractable analytical solutions for such problem.

4.1 Queuing theory

According to Cooper (1981), queuing theory is a field that concerns the construction and analysis of mathematical models kind systems, in which, provide service to customers whose arrival times and service requirements are random. The goal of presenting concepts about this field is to explain how queuing theory can be used within the scope of the research presented in this thesis. Then, from this concepts, to obtain useful performance measures which describes the stochastic DES in study.

From a DES view, a general queue system is composed of buffers and servers interconnected, being the complete system represented by a parallel composition of automata. As an example, the more simple queuing system is composed of a single queue with a single server. Clients arrive and depart due to events occurrences. In case of the system is busy and occurs a newly arrived, the client is buffered and waits for its turn. Figure 4.1 depicts a simple queue witha single server. The events, a and d denote arrivals and departs of clients, respectively.

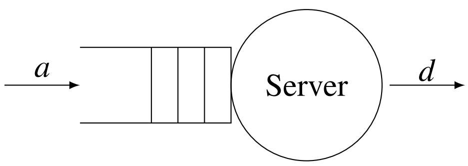


Figure 4.1: Simple queue system

There are many features to describe a queue system. Such characteristics generally are the queue discipline, the number of servers and the probability distribution of inter-event times. In sequence, it is presented a usual notation, well-known as *Kendall's notation*, for describes queue systems.

4.1.1 Kendall's notation

The Kendall's notation is usually used to describes a queue or, even systems composed of many queues. It has the follow general notation:

$$A/B/q - \text{queue discipline}$$

where A is the distribution which describes the inter-arrival times of clients, B indicates the distribution of the service time and q the number of servers ($q > 0$). Following, it is listed the symbols usually used for representing A and B:

- M: Exponential distribution (Markovian process);
- E_k: Erlang distribution with k phases;
- H_k: Hyperexponential distribution with k phases;
- C_k: Cox distribution with k phases;
- D: Deterministic distribution;
- G: General distribution;
- G_I: General distribution with independent inter-arrival times.

The fourth parameter of Kendall's notation (queue discipline) determines the service strategy adopted in the queue. When the buffer is not empty at the instant that a busy server becomes available, such strategy defines the buffer position that will be attended. Some normally used queue discipline are:

- FIFO (*First In First Out*): Queue discipline in which the clients are served in the order of their arrival (by default, it is assumed FIFO in the Kendall's notation);
- LIFO (*Last In First Out*): Queue discipline in which the clients are served in the inverse order of their arrival;
- RR (*Round Robin*): Queue discipline in which the time is divided into slices of specified length. If the client who is in the server does not finish before such a slice of time, it is preempted and returned to the last position of the buffer. This action is repeated until the preempted client finish his task;

- PS (*Processor Sharing*): This queue discipline is similar to RR. However, it is characterized by dynamic time intervals. In other words, the slices of execution times are small at the begin. For each client that shares the server, this time interval increases accordingly with the number of times that he is preempted;

- IS (*Infinity Service*): This queue discipline indicates that no queue is formed. It means that clients are processed at the instant that they arrive.

The Kendall's notation can be extended of many ways, even to represents single queues of queueing networks. One of this extensions consists of single queue with finite buffer, such presented in follow:

$$A/B/q/K - \text{disciplina da fila}$$

The parameter K denotes the queue capacity, taking into account the number of clients waiting and the client on the server. For this kind of queue, new arrivals are lost if the number of clients in the system is K.

4.1.2 Markovian queue system

To demonstrate the principles of the analytical approximations used in this work, three kinds of Markovian queues are presented. The first, such demonstrated in the following notation, consist of the more simple representation:

$$M/M/1 - \text{FIFO}$$

Regarding this notation, it is a queue with a single server. Moreover, the sequence of occurrences of the arrival and depart events are Poisson processes. Such queue can be represented by the stochastic timed automaton $(\mathbf{X}, \mathbf{E}, \Gamma, p, p_0, \mathbf{V_s})$, where:

- $\mathbf{X} = \{0, 1, 2, \ldots\}$;

- $\mathbf{E} = \{a, d\}$;

- Γ: $\Gamma(0) = \{a\}$ and $\Gamma(x) = \{a, d\}\ \forall\, x > 0$;

- $p(x+1; x; a) = 1\ \forall\, x \geq 0$ and $p(x-1, x; d) = 1\ \forall\, x > 0$;

- $p_0(0) = 1$ and $p_0(x) = 0\ \forall\, x > 0$;

- $\mathbf{V_s} = \{V_a(t), V_d(t)\}$, where $V_a(t) = 1 - \mathrm{e}^{-\lambda t}$ and $V_d(t) = 1 - \mathrm{e}^{-\mu t}$.

This stochastic timed automaton is composed of a Poisson clock structure $\mathbf{V_s}$, in which, the sequence of occurrences of events a and d are Poisson process with rates λ and μ, respectively. Regarding the definitions presented in the previous chapter, the mentioned stochastic timed automaton generates the Markov chain shown in Figure 4.2.

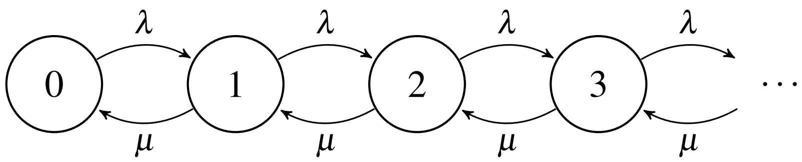


Figure 4.2: Transition diagram of a $M/M/1-$FIFO queue

Figure 4.2 illustrates a Markov chain with infinite number of states. Let's consider the *Birth-Death*[1] theory, so, the system reaches the stationary state if $\lambda < \mu$. The measure 'stationary probability' of each state is determined by Equation (4.1). The deduction of this equation can be seen in (Cooper, 1981).

$$\pi_n = \begin{cases} 1-\frac{\lambda}{\mu} & \text{if } n=0 \\ (1-\frac{\lambda}{\mu})\left(\frac{\lambda}{\mu}\right)^n & \text{otherwise } (n>0) \end{cases} \tag{4.1}$$

For a finite Markovian system with FIFO queue discipline, it is considered the following notation.

$$M/M/1/K - \text{FIFO}$$

Figure 4.3 shows a $M/M/1/K-$FIFO. For this kind of queue, the stationary probabilities of states are determined by Equation (4.2). The deduction of this equation can also be seen in (Cooper, 1981).

$$\pi_n = \begin{cases} \left(\frac{1-\frac{\lambda}{\mu}}{1-\frac{\lambda}{\mu}^{K+1}}\right)\left(\frac{\lambda}{\mu}\right) & \text{if } 0 \le n \le K \\ 0 & \text{otherwise } (n>K) \end{cases} \tag{4.2}$$

[1]Birth-Death process: For a homogeneous and aperiodic Markov chain with a unique recurrence class, the changes of states only occur between neighboring states. In others words, if the state at instant t_k is $X_k = 1$, then the state at instant t_{k+1} must be $X_{k+1} = (i+1)$ or $X_{k+1} = (i-1)$.

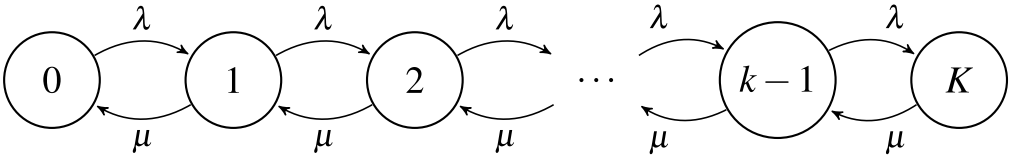


Figure 4.3: Transition diagram of a $M/M/1/K-$FIFO queue

At last, for a infinity Markovian system with IS queue discipline is described by the following notation.

$$M/M/1-\text{IS}$$

This kind of queue can be seen as a queue with infinity servers. In this case, no client is buffered. A $M/M/1-$IS queue, also known in the literature as $M/M/\infty$, indicates that the clients that arrive are attended simultaneously. Thus, the depart rates are dependents of number of the clients. Figure 4.4 illustrates the transition diagram of a $M/M/1-$IS, where, λ and μ are the arrival and service rates, respectively.

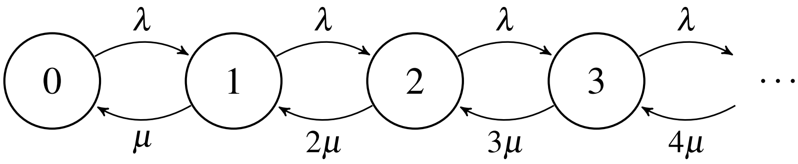


Figure 4.4: Transition diagram of a $M/M/1-$IS queue

For a $M/M/1-$IS queue, the stationary probabilities of states is determined by Equation (4.3). The deduction of this equation can also be seen in (Cooper, 1981).

$$\pi_n = e^{-\frac{\lambda}{\mu}} \frac{\frac{\lambda}{\mu}}{n!} \quad n \geq 0 \tag{4.3}$$

Cooper (1981) and Cassandras and Lafortune (2009) discuss others Markovian queue notations.

4.2 Open queuing network

A queuing network consists of many connected stations. Also known as nodes, such stations are processes that execute determined tasks. In this circumstances, clients are transfered between stations which compose the queuing network. The Figure 3.4, shown in Chapter 3, is a good example of a minimum queuing network. In such figure, the depart of server 1 is the arrival of server 2.

A queuing network is defined as 'open' when clients can incoming and outcoming the system. This way, the number of current clients in the network is unlimited. Figure 4.5 shows an open queuing network composed of two nodes. The terms μ_1 and μ_2 denote the service rates of each node, while v_0 indicates the arrival rate of clients in the network. As it can be seen, the labels v_1 and v_2 represent the arrival rate in the queue of the network.

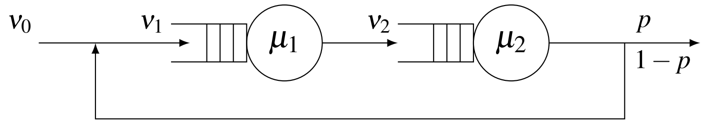


Figure 4.5: Open queuing network

For the analysis of queuing network, it is required to consider an important classical theorem. Known as Burke's theorem (Burke, 1956), it gives us the possibility of treating each node individually in a queuing network.

Theorem 2. *Burke's theorem: For Markovian queues $M/M/1$, $M/M/q$ or $M/M/\infty$, in which, the inter-arrival time is a Poisson process with rate λ and the service time is exponentially distributed with rate μ, the depart of the queue is a Poisson process with parameter λ if $\lambda < \mu$.*

Let's considered a queuing network fully Markovian, in which, all the queues that compose the system are Markovian (Markovian queue network). Then, taking into account the Burke's theorem, the performance analysis of each queue must be done in terms of the arrival rate, the service rate and the number of servers. Equation (4.4) represents a flow balance equation which determines the arrival rates. The term p_{jm} represents the transition probability from process j to process m.

$$v_m = v_0 + \sum_{j=1}^{M} v_j p_{jm} \quad \forall\, m = 1, \ldots, M \tag{4.4}$$

As an example, let's suppose that the queue the network illustrated in Figure 4.5 are $M/M/1$. Therefore, μ_1 and μ_2 are the rates that describe exponential distributions and v_0 is the rate of an arrival event described by a Poisson process. In this context, the flow balance Equation (4.4) can be applied. Thus, the following system of equations is enough for computing the arrival rate of each queue, assuming that v_0 is known.

$$\begin{aligned} v_1 &= v_0 + (1-p)v_2 \\ v_2 &= v_1 \end{aligned}$$

Another important theorem for analysis of queuing network was provided by Jackson

(1957). Known as 'product-form', Jackson's theorem is a remarkable statement that a queuing network constructed under Markovian conditions can be treated individually.

As defined in the early section, π_n denotes the stationary probability of state n, in which, n indicates the number of clients in any queue. For a Markovian queuing network, the set of states is formed by the parallel composition of all queues of the network. Let $\boldsymbol{\chi} = \{X_1, X_2, \ldots, X_M\}$ be the tuple that represents a state generated from the parallel composition operation, $\pi(n_1, \ldots, n_M)$ denotes its stationary probability (Equation (4.5)). For each X_m, it is attributed a value n_m ($n_m \geq 0$) that indicates the number of clients in the queue m.

$$\pi(n_1, \ldots, n_M) = P[X_1 = n_1, \ldots, X_M = n_M] \tag{4.5}$$

Theorem 3. *Jackson's theorem: Assuming that, in a Markovian queuing network with M queues, $\lambda_m < \mu_m \; \forall \; 1 \leq m \leq M$:*

$$\pi(n_1, \ldots, n_M) = \pi_{n_1} \pi_{n_2} \ldots \pi_{n_M} \tag{4.6}$$

The Jackson's theorem shows us that the stationary probability of a state, for a Markovian queuing network, can be expressed as the product of the stationary probability of the individual queues. These, in turn, are seen as isolated systems. In this case, the types of queues that describes the nodes are fundamental for formulating an expression that allows us to calculate the stationary probability in a queuing network. As detailed in the previous section, for queues $M/M/1-FIFO$, $M/M/1/K-FIFO$ and $M/M/1-IS$, Equations (4.1), (4.2) and (4.3) must be considered.

4.2.1 Performance measures

From the stationary probability, it possible to determine many performance measures of a queuing network. Some of these measures are presented in the sequence. The formulation to compute such measures is detailed, as well. According to Bolch et al (2006), the most important performance measures are:

4.2.1.1 Marginal probability

For a queuing network, $\pi_m(n)$ denotes the marginal probability of contain exactly n clients in node m. This measure can be computed by Equation (4.7).

$$\pi_m(n) = \sum_{n_1 + \cdots + n_m = n} \pi(n_1, \ldots, n_m) \tag{4.7}$$

Regarding the marginal probability, it possible to see that $\pi_m(n)$ denotes the summation of the stationary probabilities of states with n clients in node m. Moreover, this performance measure must be normalized due to Equation (4.8).

$$\sum_{n=0}^{N} \pi_i(n) = 1 \;\; \forall\, 1 \leq i \leq M \tag{4.8}$$

4.2.1.2 Throughput

The throughput, denoted by ψ_m, indicates the rate that clients outcoming of node m. Generally, it can be determined by Equation (4.9).

$$\psi_m = \sum_{n=1}^{\infty} \pi_m(n)\mu_m(n) \tag{4.9}$$

Whose $\mu_m(n)$ denotes the dependent service rate of process m. This rate depends of the number of clients n in the mentioned process. For instance, let m be a process with q_m servers. If that each server operates with rate service μ_m, then $\mu_m(n) = \min(n, q_m)\mu_m$.

4.2.1.3 Mean number of clients

The mean number of clients in the process m can be computed by Equation (4.10).

$$\bar{N}_m = \sum_{n=1}^{\infty} n\pi_m(n) \;\; \forall\, 1 \leq m \leq M \tag{4.10}$$

4.2.1.4 Mean response time

The mean response time $\bar{T}_m$ is the mean time that clients spend in process m. Based on Little law (Little, 1961), Equation (4.11) shows how this performance measure is determined.

$$\bar{T}_m = \frac{\bar{N}_m}{\psi_m} \tag{4.11}$$

4.3 Closed Queuing network

Closed queuing networks consists of cyclic models with a finite and fixed number of clients. In the queuing theory context, it means that no clients can be entering or be exiting the cycle. According to Cassandras and Lafortune (2009), a closed queuing network is represented by Equation (4.12) and (4.13).

The term M denotes the number of nodes which composes the queue system and p_{mj} indicates the transition probability from node m to node j. It is possible observing that in the Equation (4.12) neither external arrivals nor departs of clients occur. The finite and fixed population is also guaranteed by Equation (4.13), in which, N denotes the number of clients that run in the network and X_m the number of clients in node m ($1 \leq m \leq M$). In other words, X_m is the queue size of node m.

$$\sum_{j=1}^{M} p_{mj} = 1 \quad \forall\, m = 1, \ldots, M \tag{4.12}$$

$$\sum_{m=1}^{M} X_m = N \tag{4.13}$$

For this kind of queuing system, the cardinality of the state space (L) is finite as well. The term L is determined by the binomial coefficient shown in Equation (4.14).

$$L = \binom{M+N-1}{M-1} \tag{4.14}$$

According to Gordon and Newell (1967), $\pi(n_1, \ldots, n_M)$ can be obtained by Equation (4.15),

$$\pi(n_1, \ldots, n_M) = \frac{1}{G(N)} \prod_{m=1}^{M} (\rho_m)^{n_m}, \tag{4.15}$$

where ρ_m denotes the traffic intensity of process m and $G(N)$ a normalization constant. This constant depends on the total number of clients.

The traffic intensity ρ_m of the process m is determined by the ratio expressed by Equation (4.16).

$$\rho_m = \frac{\nu_m}{\mu_m} \quad \forall\, m = 1, \ldots, M \tag{4.16}$$

Where ν_m and μ_m denote the relative arrival rate and the depart rate, respectively.

Once that there are no external arrivals in a closed queuing network, the set of flow balance Equations expressed in Equation (4.4) is simplified in Equation (4.17).

$$v_m = \sum_{j=1}^{M} v_j p_{jm} \quad \forall\, m = 1, \ldots, M \tag{4.17}$$

Taken into account that, for closed queuing networks, there are only $(M-1)$ independent linear equations, infinite solutions can be found. However, any solution can be considered. In this analysis, it is only necessary a value for each v_m that satisfies the M balance Equations (4.17).

The normalization constant $G(N)$ is defined such a way that the summation of stationary equations must be 1, as presented in Equation (4.18). It means that, from this definition, it is possible to determine $G(N)$. Consequently, the stationary probability of each state.

$$\frac{1}{G(N)} \sum_{n \in S(N,M)} \prod_{m=1}^{M} (\rho_m)^{n_m} = 1 \tag{4.18}$$

The term $S(N,M)$ in the Equation (4.18) denotes the set defined by Equation (4.19).

$$S(N,M) = \{(n_1, n_2, \ldots, n_M) | \sum_{m=1}^{M} n_m = N \;\&\; n_1 \geq 0 \;\forall\, m\} \tag{4.19}$$

4.3.1 A recursive mechanism for computing the normalization constant

As presented in Equation (4.18), the calculus of $G(N)$ requires the summation of L stationary probability states . The cardinality of L, in turn, has a factorial growth. Taken this into account, to determine $G(N)$ can be an arduous, even impossible, task for problems with a huge number of states. Regarding such complexity, Buzen (1973) presents a recursive mechanism for computing $G(N)$. The basic principle of this technique is obtaining recursively the values for $G(1)$, $G(2)$, ... and $G(N)$.

According to Buzen (1973), the normalization constant $G(N)$ can be determined from a step function, as presented by Equation (4.20).

$$g(n,m) = \begin{cases} 1, & \text{if } n = 0 \;\; \forall\, 1 \leq m \leq M \\ (\rho_m)^n, & \text{if } m = 1 \;\; \forall\, 0 \leq n \leq N \\ g(n, m-1) + \rho_m . g(n-1, m), & \text{otherwise} \end{cases} \tag{4.20}$$

The term n and m of the mentioned step function denote the number of clients and of nodes, respectively. The term $g(n,m)$ denotes the normalization constant of all possible sub closed queuing network, in which, $0 \leq n \leq N$ and $1 \leq m \leq M$. Therefore, $g(N,M)$ represents $G(N)$.

Table 4.1 presents the iterative dynamic of the recursive algorithm developed by Buzen (1973). Regarding this mechanism, the normalization constant $g(N,M)$ is determined with complexity $O(N.M)$. This solution assumes that the process times are independent of the number of clients present at a given time. However, Buzen (1973) extends his method for dealing with problems in which such condition is not satisfied. For this problems, the complexity is not $O(N.M)$, but, it still so less costly than the standard formulation (Equation (4.18)). A new derivative of this recursive mechanism is presented in the following section.

	$m=1$	$m=2$	$\ldots$	m	$\ldots$	$m=M$
$n=0$	$(\rho_1)^0$	1	$\ldots$	1	$\ldots$	1
$n=1$	$(\rho_1)^1$					
$n=2$	$(\rho_1)^2$					
$n=3$	$(\rho_1)^3$					
$\vdots$	$\vdots$				$g(n-1,m)$	
					$\downarrow .b_m$	
n	$(\rho_1)^n$		$g(n,m-1)$	$\rightarrow$	$g(n,m)$	
$\vdots$	$\vdots$					
$n=N$	$(\rho_1)^N$					$g(N,M)$

Table 4.1: Iterative dynamic of the recursive algorithm (Buzen, 1973)

4.3.2 Example 1

Let's suppose a closed queuing network composed of 4 nodes, such illustrated in Figure 4.6. All the queues are classified as $M/M/1-\text{FIFO}$. The term μ_m denotes the service rate of node m. The number of clients (N) which run in the system and the services rate must be defined. For this example, $N=2$, $\mu_1=2$, $\mu_2=1$, $\mu_3=0,5$ and $\mu_4=1$.

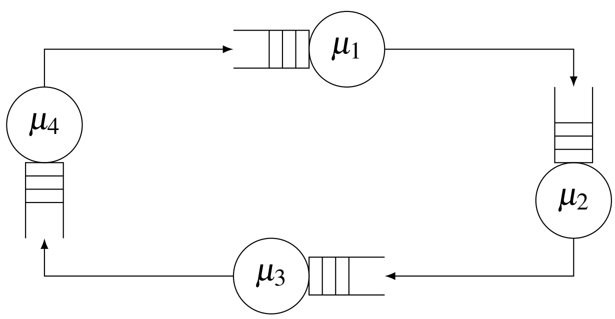


Figure 4.6: Example 1

For this example, it is applied the strategy described in this section for computing the stationary probability of each state. And, then, to compute the performance measures: marginal probability, throughput, the mean number of clients and the mean response time. As described previously, the first step is to find a relative arrival rate which satisfies the flow balance equations. For this example, the relatives arrival rates v_m for all nodes m must satisfy Equation (4.17). Then, applying Equation (4.16), we obtained the traffic intensity of each process. As a result, $\rho_1 = 0.5$, $\rho_2 = 1$, $\rho_3 = 2$ e $\rho_4 = 1$.

From the step function shown in Equation (4.20), it is computed the normalization constant $G(N)$. Table 4.2 demonstrates the iterative dynamic for computing such constant. As a result, for $N = 2$ and $M = 4$, $G(N) = 13,25$.

	$m = 1$	$m = 2$	$m = 3$	$M = 4$
$n = 0$	1.0	1.0	1.0	1.0
$n = 1$	0.5	1.5	3.5	4.5
$N = 2$	0.5^2	1.75	8.75	13.25

Table 4.2: Iterative dynamic for computing $g(N,M)$ (Example 1)

From the normalization constant $G(N)$ and the traffic intensity measured ($\rho_m \ \forall\ m = 1,\dots,4$), the stationary probability of each state is obtained by Equation (4.15). Figure 4.7 shows the transition diagram which describes the dynamic of this system, which, in turn, composed of $L = 10$ states. Table 4.3 presents such states and their stationary probabilities.

$\chi = \{n_1,\dots,n_M\}$	$\pi(n_1,\dots,n_M)$
0,0,0,2	0.075
0,0,1,1	0.151
0,0,2,0	0.302
0,1,0,1	0.075
0,1,1,0	0.151
0,2,0,0	0.075
1,0,0,1	0.038
1,0,1,0	0.075
1,1,0,0	0.038
2,0,0,0	0.019
Soma	1

Table 4.3: Stationary probabilities (Example 1)

From the stationary probabilities presented, it is possible to compute the marginal probabilities. Table 4.4 presents the values for this performance measure.

	$m = 1$	$m = 2$	$m = 3$	$M = 4$
$n = 0$	0.83	0.66	0.32	0.66
$n = 1$	0.15	0.26	0.38	0.26
$N = 2$	0.02	0.08	0.30	0.08

Table 4.4: Marginal probabilities $\pi_m(n)$

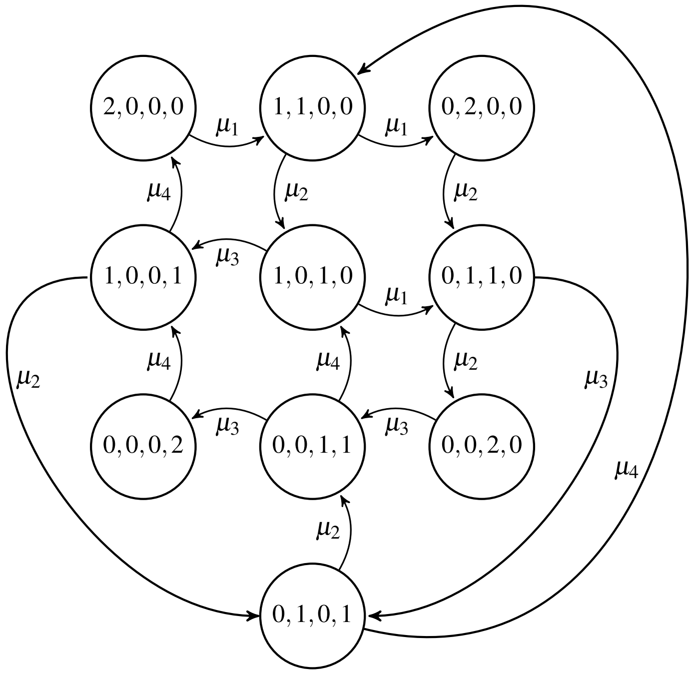


Figure 4.7: Transition diagram for the Markovian model with 2 clients

From Equation (4.9) it is computed the throughput of each state. Table 4.5 shows the results for ψ_m.

	$m=1$	$m=2$	$m=3$	$M=4$
ψ_m	0.34	0.68	1.36	0.68

Table 4.5: Throughput ψ_m

The mean number of clients of each state is obtained by Equation (4.10). Table 4.6 presents the value of $\bar{K_m}$.

	$m=1$	$m=2$	$m=3$	$M=4$
$\bar{K_m}$	0.19	0.42	0.98	0.42

Table 4.6: Mean number of clients $\bar{K_m}$ of each m process

At last, Table 4.15 shows the mean response time of each process m. Such performance measure is determined by Equation (4.11).

	$m=1$	$m=2$	$m=3$	$M=4$
$\bar{T_m}$	0.56	0.61	0.72	0.61

Table 4.7: Mean response time $\bar{T_m}$ of each process m

4.4 Analysis of Close Queuing network with different disciplines

The strategy presented in previously is applicable just in closed queuing networks which all the nodes are $M/M/1-$FIFO queues. However, many problems are composed of nodes described by different disciplines. Taken it into account, it is presented an adaptation of the previous strategy. This method extends the applicability such a way that it is possible dealing with closed queue networks composed of nodes featured by different disciplines. In this study, it is considered $M/M/q-$ FIFO and $M/M/1-$ IS queues.

Gordon and Newell (1967) argues that the stationary probability can be described in terms of the queue discipline and the number of servers. That is, the depart rate of a process can be dependent of the number of clients. Bolch et al (2006) describe the equations that make possible this new approach. Regarding such equations, Equation (4.21) represents a generalized way to determine the stationary probability, while Equation (4.22) represents the computation of the normalization constant.

$$\pi(n_1,\dots,n_M) = \frac{1}{G(N)}\prod_{m=1}^{M}\beta_m(n_m) \tag{4.21}$$

$$G(N) = \sum_{n\in S(N,M)}\prod_{i=m}^{M}\beta_m(n_m) \tag{4.22}$$

It is possible to see that both Equation (4.21) and Equation (4.22) are written in terms of $\beta_m(n_m)$. It is an auxiliary step function which represents the dependent behavior of node m with n_m clients. Such behavior depends essentially on the queue discipline and the number of servers in each node m. In this study is considered the method presented by Kappas and Yegulalp (1991). The authors use Jackson's product form to derive an expression which deals with closed queuing networks formed by different kinds of queues.

Regarding the study proposed by Kappas and Yegulalp (1991), the authors apply the queue theory in truck-shovel system of open-pit mining. Moreover, it is considered the queues disciplines $M/M/1-$FIFO and $M/M/1-$IS. According to Kappas and Yegulalp (1991) the auxiliary term $\beta_i(n_i)$ is determined by the step function (4.23).

$$\beta_m(n_m) = \begin{cases} 1, & \text{if } n_m = 0 \\ \frac{(\rho_m)^{n_m}}{\prod_{k=1}^{n_m} b_m(k)}, & \text{otherwise} \end{cases} \tag{4.23}$$

where $b_m(.)$ denotes the step function (4.24). The element $b_m(.)$ is described by the number of servers (q_m) in node m and queue discipline of the same node.

$$b_m(k) = \begin{cases} 0, & \text{if } k = 0 \\ \min(k, q_m), & \text{if } 0 < k \leq N \quad \& \quad \text{FIFO} \\ k, & \text{if } 0 < k \leq N \quad \& \quad \text{IS} \end{cases} \tag{4.24}$$

4.4.1 Computing the normalization constant for system with different disciplines

As mentioned early, Buzen (1973) also presents an extension of his method which deals with systems with dependent rates. Such extension is adequate for the kind of system discussed in the current section. Regarding this method, the same approach for deducing an expression for the normalization constant $G(N)$ presented in the early section can be applied.

Such defined, $g(n,m)$ denotes the normalization constant of each sub closed queuing network (n,m). Thus, Equation (4.22) can be rewritten as Equation (4.25).

$$g(n,m) = \sum_{n \in S(n,m)} \prod_{i=1}^{m} \beta_i(n_i) \tag{4.25}$$

The Equation (4.26), for $m > 1$, points out how the constant $g(n,m)$ can be determined recursively from the methodology shown by Buzen (1973).

$$\begin{aligned} g(n,m) &= \sum_{k=0}^{n} \left[\sum_{\substack{n \in S(n,m) \\ \& \ n_m = k}} \prod_{i=1}^{m} \beta_i(n_i) \right] \\ &= \sum_{k=0}^{n} \beta_m(k) \left[\sum_{n \in S(n-k,m-1)} \prod_{i=1}^{m-1} \beta_i(n_i) \right] \\ &= \sum_{k=0}^{n} \beta_m(k) g(n-k,m-1) \end{aligned} \tag{4.26}$$

From Equations (4.25) and (4.26), it is derived the step function (4.27). Such function defines $g(n,m)$ for the kind of closed queuing system discussed in this section.

$$g(n,m) = \begin{cases} 1, & \text{se } n = 0 \; \forall \; 1 \leq m \leq M \\ \beta_m(n), & \text{se } m = 1 \; \forall \; \leq n \leq N \\ \sum_{k=0}^{n} \beta_m(k) g(n-k, m-1), & \text{caso contrário} \end{cases} \tag{4.27}$$

It can be seen that the normalization constant $g(N,M)$ is obtained with complexity $O(2.N.M)$.

4.4.2 Example 2

Let's suppose a Markovian cyclic system with 4 processes, in which, each one has a single server. Figure 4.8 illustrates the system. The processes 1 and 3 have the services rates μ_1 and μ_3 respectively. In addition, processes 1 and 3 are featured by FIFO queue discipline. There is no queue in the other two processes. In other words, processes 2 and 4 are characterized by IS queue discipline.

Let's consider the same values used in example 1. So, $N = 2$, $\mu_1 = 1$, $\mu_2 = 1$, $\mu_3 = 0.5$ and $\mu_4 = 1$. The strategy detailed in this section is used to compute the stationary probability of each process. Besides, the performance measures: throughput, mean number of clients and the mean response time.

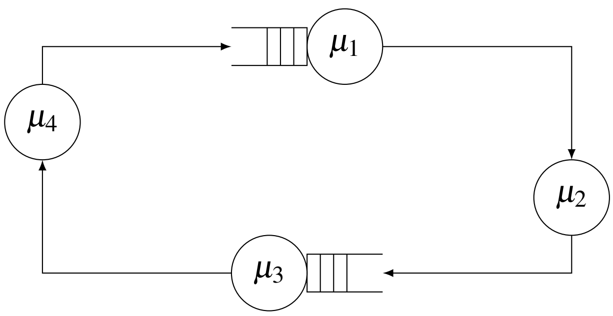


Figure 4.8: Example 2

The first step is to find a relative arrival rate which satisfies the flow balance equations. As well as in example 1, $v_m = 1 \; \forall \; m$ satisfy Equation (4.17). From Equation (4.16), it is obtained the traffic intensity of each process. As a result, $\rho_1 = 0.5$, $\rho_2 = 1$, $\rho_3 = 2$ e $\rho_4 = 1$.

Table 4.8 shows the values obtained from Equation (4.24):

where k, from 0 to N, denotes the number of clients and m the process. Using Equation (4.23), the values for each combination $k \times m$ is computed. Table 4.9 shows the results of $\beta_m(k)$.

	$m=1$	$m=2$	$m=3$	$m=4$
$k=0$	0	0	0	0
$k=1$	1	1	1	1
$k=2$	1	2	1	2

Table 4.8: Auxiliary function $b_m(k)$

	$m=1$	$m=2$	$m=3$	$m=4$
$n=0$	1.00	1.00	1.00	1.00
$n=1$	0.50	1.00	2.00	1.00
$n=2$	0.25	0.50	4.00	0.50

Table 4.9: Auxiliary function $\beta_m(k)$

Applying the values of Table 4.9 in the Equation (4.25) it possible to compute the normalization constant $G(N)$. Table 4.10 demonstrates the iterative dynamic for computing such constant. As a result, for $N=2$ and $M=4$, $G(N)=12,25$.

	$m=1$	$m=2$	$m=3$	$m=4$
$n=0$	1,00	1,00	1,00	1,00
$n=1$	0,50	1,50	3,50	4,50
$n=2$	0,25	1,25	8,25	12,25

Table 4.10: Iterative dynamic for computing $g(N,M)$ (Example 2)

Considering the normalization constant $G(N)$ and the values of $\beta_m(k)$ shown in Table 4.9 it is possible to obtain, by Equation (4.21), the stationary probability of each process.

Table 4.11 shows the states $(n_1,\ldots,n_M)$ and their stationary probability $\pi(n_1,\ldots,n_M)$.

$\chi=\{n_1,\ldots,n_M\}$	$\pi(n_1,\ldots,n_M)$
0,0,0,2	0.041
0,0,1,1	0.163
0,0,2,0	0.327
0,1,0,1	0.082
0,1,1,0	0.163
0,2,0,0	0.041
1,0,0,1	0.041
1,0,1,0	0.082
1,1,0,0	0.041
2,0,0,0	0.020
Soma	1

Table 4.11: Stationary probabilities (Example 2)

Considering the values of Table 4.11, the marginal probability is obtained using Equation (4.7). Thus, Table 4.12 presents the results for this performance measure.

	$m=1$	$m=2$	$m=3$	$m=4$
$n=0$	0.82	0.67	0.27	0.67
$n=1$	0.16	0.29	0.41	0.29
$n=2$	0.02	0.04	0.33	0.04

Table 4.12: Marginal probability $\pi_m(n)$

Applying Equation (4.9), the throughput of each state is computed, as shown in Table 4.13.

	$m=1$	$m=2$	$m=3$	$m=4$
ψ_m	0.37	0.65	1.47	0.65

Table 4.13: Throughput ψ_m

Table 4.14 shows the mean number of clients of each process. Such performance measure is determined by Equation (4.10).

	$m=1$	$m=2$	$m=3$	$m=4$
$\bar{K}_m$	0.20	0.37	1.06	0.37

Table 4.14: Mean number of clients $\bar{K}_m$ of each process m

At last, Table 4.15 presents the mean response time of each process. This performance measure is obtained by Equation (4.11).

	$m=1$	$m=2$	$m=3$	$m=4$
$\bar{T}_m$	0.56	0.56	0.72	0.56

Table 4.15: Mean response time $\bar{T}_m$ of each process m

4.5 Convolution algorithm

The techniques presented previously consist of efficient methods for computing the stationary probability and, consequently, the desired performance measures. However, besides the complexity $O(2.N.M)$, or even $O(N.M)$, to determine the normalization constant $G(N)$, the complexity for computing the stationary probability of all states is factorial, such as Equation (4.14). Therefore, the number of states L in a Markov chain of a complex system can become very huge and it cannot be possible to compute this measure for all states.

Given this condition, a technique based on convolution is detailed in this section. The core of this new strategy is computing directly the performance measure 'marginal probability' applying the same analogy presented by Buzen (1973) for determining the normalization constant $G(N)$. According to Bolch et al (2006), the convolution algorithm was one of the first efficient methods to analysis closed queuing network. The name 'convolution algorithm' comes from the convolution of two pmfs (*probability mass functions*). Let A, B and C be vectors of size $N+1$, the convolution $C = A \circledast B$ ($\circledast$: convolution operator) is defined by Equation (4.28).

$$C(n) = \sum_{k=0}^{n} A(k).B(n-k), \ \ \forall\, n = 0, \ldots, N \tag{4.28}$$

To exemplify, we show that the Equation deduced in (4.26), for $m > 1$, is the convolution between the functions $\beta_m(.)$ and $g(.,m-1)$. Thus, $g(n,m)$ can be described in terms of the convolution operator ($\circledast$), such as shown by Equation (4.29).

$$g(.,m) = \beta_m(.) \circledast g(.,m-1) \ \ \forall\, 1 < m \leq M \tag{4.29}$$

As mentioned, the number of states in a Markov chain of a complex system can become very huge. Therefore, it cannot be possible to compute all $\pi(n_1, \ldots, n_M)$. However, the convolution algorithm is sufficient to obtain the marginal probability directly, as demonstrated by Bolch et al (2006). That means, without explicitly to compute $\pi(n_1, \ldots, n_M)$. In general terms, this technique consist of substituting the Equation (4.21) in Equation (4.7), as shown in Equation (4.30).

$$\begin{aligned}
\pi_m(n) &= \sum_{\substack{n \in S(n,m) \\ \&\ n_m = n}} \frac{1}{G(N)} \prod_{j=1}^{M} \beta_j(n_j) \\
&= \sum_{\substack{n \in S(n,m) \\ \&\ n_m = n}} \frac{\beta_m}{G(N)} \prod_{\substack{j=1 \\ \& j \neq m}}^{M} \beta_j(n_j) \\
&= \frac{\beta_m(n)}{G(N)} \boxed{\sum_{\substack{n \in S(n,m) \\ \&\ n_m = n}} \prod_{\substack{j=1 \\ \&\ j \neq m}}^{M} \beta_j(n_j)} \\
&= \frac{\beta_m(n)}{G(N)} G_M^{(m)}(N-n)
\end{aligned} \tag{4.30}$$

The dashed area must be interpreted as a normalization constant of a closed queuing network without the node m and with n clients less. Then, analogously, the term $G_M^{(m)}(n)$ can be described by Equation (4.31).

$$G_M^{(m)}(n) = \sum_{\substack{n \in S(n,m) \\ \& \ n_m = N-n}} \prod_{\substack{j=1 \\ \& \ j \neq m}}^{M} \beta_j(n_j) \tag{4.31}$$

Balbo et al (1977) presents an efficient numerical method for computing $G_M^m(n)$. This method considers all n at the interval $\{0,\ldots,N\}$. From Equation (4.8), it possible to see that the summation of marginal probabilities of each process m must be 1. Therefore, such condition can be described by Equation (4.32).

$$\sum_{n=1}^{N} \pi_m(n) = \sum_{n=1}^{N} \frac{\beta_m(n)}{G_m(N)} G_M^m(N-n) = 1 \ \ \forall\, 1 \leq m \leq \ M \tag{4.32}$$

Considering Equation (4.32), it possible to derive a formulation for the normalization constant $G_M^{(m)}(n)$:

$$G(n) = \sum_{k=0}^{n} \beta_m(k) G_M^{(m)}(n-k)$$
$$G(n) = \beta_m(0) G_M^{(m)}(n) + \sum_{k=1}^{n} \beta_m(k) G_M^{(m)}(n-k)$$

In accordance with Equation (4.23), $\beta_m(0) = 0\ \forall\ 1 \leq m \leq M$. Thus, $G_M^{(m)}(n)\ \forall\ 1 \leq m \leq M$ can be determined by Equation (4.33).

$$G_M^{(m)}(n) = G(N) + \sum_{n=1}^{N} \beta_m(n) G_M^{(m)}(N-n) \ \ \forall\, 0 \leq n \leq N \tag{4.33}$$

With the initial condition, $G_M^{(m)}(0)$, given by Equation (4.34).

$$G_M^{(m)}(0) = 1 \ \ \forall\, 1 \leq m \leq M \tag{4.34}$$

In conclusion, the computation of the normalization constant $G_M^{(m)}(N-n)$, for all process m, can be summarized by the recursively step function (4.35).

$$G_M^{(m)}(k) = \begin{cases} 1, & \text{if } k = 0 \\ G(N) - \sum_{k=1}^{N} \beta_m(n) G_M^{(m)}(N-k), & \text{otherwise} \end{cases} \tag{4.35}$$

4.5.1 Example 3

Let's consider the same Markovian cyclic system of example 2, with $N = 2$ clients and $M = 4$ processes. The goal is applying the convolution algorithm described in this section to compute the marginal probabilities and, then, to compare with the results obtained in example 2.

From the values of Table 4.10, the normalization constant $G(N) = 12,25$. Using the $\beta_m(k)$ values of Table 4.9 in Equation (4.33), $G_M^{(m)}(n)$ for each combination n and m is obtained. Table 4.16 presents the values of $G_M^{(m)}(n)$.

	$m = 1$	$m = 2$	$m = 3$	$m = 4$
$n = 0$	1.00	1.00	1.00	1.00
$n = 1$	4.00	3.50	2.50	3.50
$n = 2$	10.00	8.25	3.25	8.25

Table 4.16: Normalization constant $G_M^{(m)}(n)$

As described in the sequence, the marginal probabilities are determined by Equation (4.30).

For $m = 4$ and changing the number of clients from 0 to 2:

$$\begin{aligned}
\pi_4(0) &= \tfrac{\beta_4(0)}{G(2)} G_M^{(4)}(2-0) &= \tfrac{1}{12.25} \times 8.25 &= 0.67 \\
\pi_4(1) &= \tfrac{\beta_4(1)}{G(2)} G_M^{(4)}(2-1) &= \tfrac{1}{12.25} \times 3.50 &= 0.29 \\
\pi_4(2) &= \tfrac{\beta_4(2)}{G(2)} G_M^{(4)}(2-2) &= \tfrac{0,5}{12.25} \times 1.00 &= 0.04
\end{aligned}$$

For $m = 3$ and changing the number of clients from 0 to 2:

$$\begin{aligned}
\pi_3(0) &= \tfrac{\beta_3(0)}{G(2)} G_M^{(3)}(2-0) &= \tfrac{1}{12,25} \times 3.25 &= 0.27 \\
\pi_3(1) &= \tfrac{\beta_3(1)}{G(2)} G_M^{(3)}(2-1) &= \tfrac{2}{12,25} \times 2.50 &= 0.41 \\
\pi_3(2) &= \tfrac{\beta_3(2)}{G(2)} G_M^{(3)}(2-2) &= \tfrac{4}{12,25} \times 1.00 &= 0.33
\end{aligned}$$

For $m = 2$ and changing the number of clients from 0 to 2:

$$\begin{aligned}
\pi_2(0) &= \tfrac{\beta_2(0)}{G(2)} G_M^{(2)}(2-0) &= \tfrac{1}{12,25} \times 8.25 &= 0.67 \\
\pi_2(1) &= \tfrac{\beta_2(1)}{G(2)} G_M^{(2)}(2-1) &= \tfrac{1}{12,25} \times 3.50 &= 0.29 \\
\pi_2(2) &= \tfrac{\beta_2(2)}{G(2)} G_M^{(2)}(2-2) &= \tfrac{0,5}{12,25} \times 1.00 &= 0.04
\end{aligned}$$

For $m = 1$ and changing the number of clients from 0 to 2:

$$\begin{aligned}
\pi_1(0) &= \tfrac{\beta_1(0)}{G(2)} G_M^{(1)}(2-0) &= \tfrac{1}{12.25} \times 10 &= 0.82 \\
\pi_1(1) &= \tfrac{\beta_1(1)}{G(2)} G_M^{(1)}(2-1) &= \tfrac{0,5}{12.25} \times 4 &= 0.16 \\
\pi_1(2) &= \tfrac{\beta_1(2)}{G(2)} G_M^{(1)}(2-2) &= \tfrac{0,25}{12.25} \times 1.00 &= 0.02
\end{aligned}$$

It possible to observe that the marginal probabilities are equivalent to the values obtained in example 2. Consequently, the other performance measures as well.

4.6 Phase type distribution - Marie's method

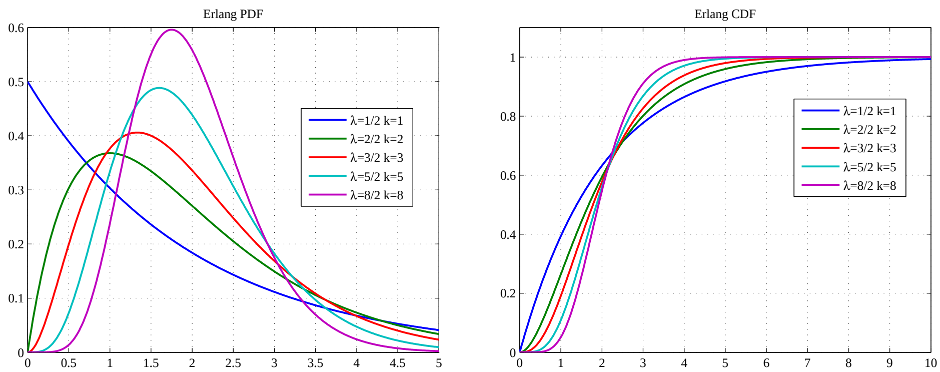
The previous section presented an efficient method that concern closed queuing networks. Summarily, the stochastic cyclic system is seen as a cyclic Markovian system and, then, desired performance measures are determined analytically. According to Marie (2011), the main grievance done to the use of Markovian models is that the time duration of process is not really exponentially distributed in real systems.

However, the author points out that phase-type distribution can reproduce as close as necessary the general distributions of the real system. The *Pollaczek-Khinchin formula* (Pollaczek, 1930) shown that some performance measures of the $M/G/1$ queue just depends on first and second moments. This argument suggests that the appropriate phase-type distribution for each queue m can be chosen according to the expected service time μ and the measure of dispersion cv. Anyway, in many practical applications we do not need the complete pdf (probability density function), since the first and the second moments (mean and variance) are sufficient to give a very good approximation for the distribution.

According to Altiok (1985), phase-type distribution is important in the queue theory because its structure can generate a Markovian model and, consequently, performance measures can be computed by analytical techniques. This category of distributions were first introduced in the scientific field by Erlang (1917). Known as Erlang distribution, it was developed for telephone networks analysis and, since that, it has a range of applications which concerns stochastic process. In general, this distribution represents the sum of *iid* random variables which are exponentially distributed.

To exemplify, Figures 4.9(a) and 4.9(a) show pdf and cdf of the Erlang distribution, where λ and κ denote the rate and the number of phases, respectively. Since that expected value of the Erlang distribution is κ/λ, the first-moment of the five functions illustrated are equals. What distinguishes one of the other is the coefficient of variation, which is defined by $cv = \sqrt{1/\kappa}$.

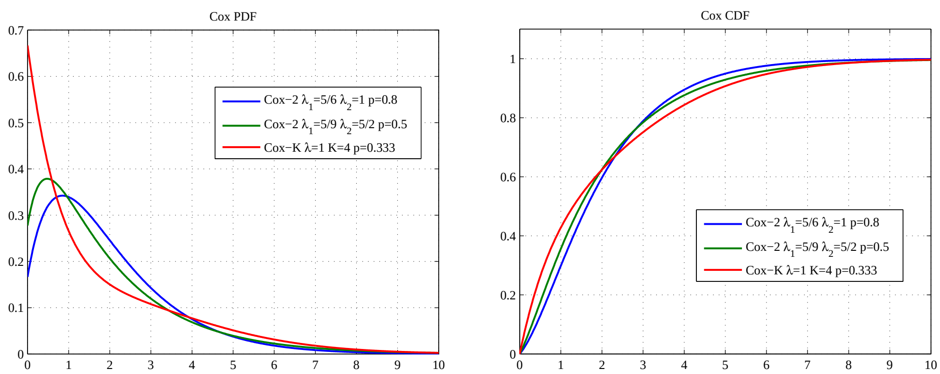
Decades later Cox (1955) argued that any distribution which has a rational Laplace-



(a) pdf (b) cdf

Figure 4.9: Erlang Distribution

Stieltjes transform can be represented by a sequence of exponential distribution in phase. From the Cox's study, many types of research were conducted with the goal of expanding the applicability of phase-type distribution. To exemplify, Figures 4.10(a) and 4.10(b) show the pdf and cdf of three configuration of cox distributions, respectively.



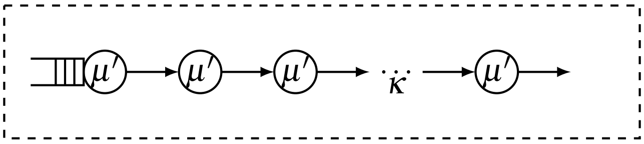
(a) pdf (b) cdf

Figure 4.10: Cox Distribution

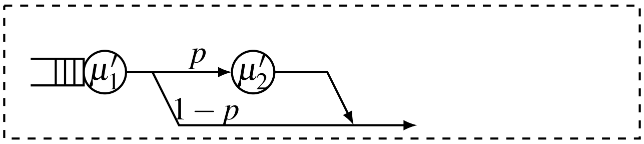
An interesting observation is that all pdf profiles are quite different for the presented distributions. In this case, we are trying to approximate all the moments of the distribution, which is clearly an exigent and uncertain procedure which relies upon the quantity and quality of the collected data. However, all cdf's are approximated by the phase-type distribution. Since essentially, the simulation uses an inversion method based on the cdf (Cassandras and Lafortune, 2009). This reinforces the fact that phase-type distribution is a good approximation in practice.

The Figure 4.11 shows a symbolical representation of a Queue modeled by Phase-type distributions. Regarding it, Figure 4.11(a) represents an Erlang distribution characterized by κ exponential phases with service rate μ'. In turn,Figure 4.11(b) illustrated a Cox-2 distribution, which has 2 exponential phases characterized by distinguished services rates, μ_1' and μ_2'. From phase 1 the transition to phase 2 occurs with probability p, while the absorbing state is reached with probability $1-p$. In other words, $1-p$ is the probability of exiting the process without pass through phase 2. This branching is a feature of any Cox distribution.

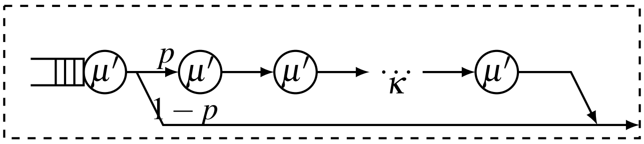
At last, 4.11(c) shows a Cox-k distribution with κ exponential phases of equal service rate μ' connected in series such as the Erlang distribution. Furthermore, there is a probability of reached the absorbing state after the phase 1. We can see from the Figures 4.10(a) and 4.10(b), that although pdf's are substantially different (since we are trying to approximate derivatives) the cdf's are closer to each other.



(a) Erlang



(b) Cox-2



(c) Cox-k

Figure 4.11: Symbolical representation of a Queue modeled by Phase-type distribution

The motivation for using phase-type distribution is the fact that it is possible to represent any distribution as close as necessary. Hence, it is important to answer the question: *'How close it should be*'. The accuracy of an approximation through Phase-type distribution is related to the number of phases and, indeed, this approximation may have a significant computational cost. Usually, an intellectual endeavor is spent to find an accurate pdf approximation in the simulation context. However, it is known that the mechanism which generates pseudo-random numbers commonly considers the cdf instead. Understood in this way, phase-type distribution can be explored to find a cdf estimation, which is not so costly as the pdf estimation.

In this context, a convenient approximation through phase-type distribution was presented by Marie (1979). The author presents an approximate solution for the asymptotic behavior of relatively general queuing networks. This solution is obtained from queuing network decomposition, where the flow among subnetworks is inferred and, hence allowing the estimate of the performance measures. One application of Marie's method is presented by Ekren et al (2013). The authors present an autonomous vehicle storage and retrieval system modeled as a 'semiclosed' queuing network. It is a system characterized by non-exponential inter-events times. Therefore, one of the steps consists in applying the method proposed by Marie.

As mentioned, the phase-type distribution is chosen subject to a given cv. Lagershausen (2012) argues that the first criterion for selecting the phase-type distribution is the proper fit of μ_m and cv_m for all process m. According to Bolch et al (2006), depending on the value of the cv^2, different phase-type models are used. If $cv^2 \cong 1/\kappa \in \mathbb{N}^+$, then an Erlang distribution is selected. For $cv^2 \geq 0.5$ the Cox-2 distribution is used and, lastly, Cox-k distribution is applied otherwise. The parameters of these three phase-type distribution are obtained according

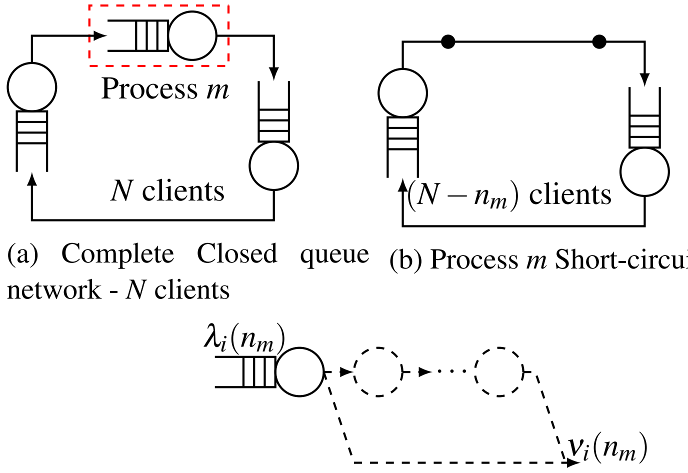
Erlang($1/\mu'$,κ)	$\frac{1}{cv^2}$ $\mu_1' = \frac{\mu}{\kappa}$
Cox-2($1/\mu_1'$,$1/\mu_2'$,p)	$\mu_1' = 2\mu$ $\mu_2' = \mu\frac{1}{cv^2}$ $p = 1 - \frac{1}{2cv^2}$
Cox-k($1/\mu'$,κ,p)	$k = \frac{1}{cv^2}$ $\mu' = [\kappa - p(k-1)]\mu$ $p = 1 - \frac{2\kappa cv^2 + (\kappa-2) - \sqrt{\kappa^2+4-4\kappa cv^2}}{2(\kappa-1)(cv^2+1)}$

Table 4.17: Equations to compute the parameters of the phase-type distributions: Erlang, Cox-2 and Cox-k

The present thesis considers an iterative method proposed by Marie (1979) as the second-moment approximation. According to Lagershausen (2012), it is a very precise approximation method for the performance evaluation of general closed queuing networks. In Marie's method, processes characterized by general distributions are converted into a load-dependent exponential process. For each process m of a closed queuing network with M processes, the load-dependent arrival and depart rates, $\lambda_m(n_m)$ and the $\nu_m(n_m)$ must be computed for $0 \leq n_m \leq N$. For this purpose, the complement network $(M-m)$ is aggregated according to the Norton's Theorem and, hence, $\lambda_m(n_m)$ can be computed according to Equation (4.36).

$$\lambda_m(n_m) = \lambda_c^{(m)}(N - n_m) \tag{4.36}$$

Taken to account that the number of clients circulating in the system is fixed and equal N, $\lambda_c^{(m)}(N-n_m)$ must be seen as the throughput of the subnetwork with process m short-circuited and n_m clients less. For all processes m and $0 \leq n_m \leq N$, $\lambda_c^{(m)}(N-n_m)$ can be computed using any product form algorithm such as convolution method presented in previous section. Once that $\lambda_m(n_m)$ is known, it can be applied for the isolated analysis of the single process m of a closed queuing network.



(a) Complete Closed queue network - N clients

(b) Process m Short-circuited

(c) Isolated analysis of process m

Figure 4.12: Closed queuing network with 3 processes ($M = 3$)

Figure 4.12 shows a symbolical representation of this important step of the Marie's method. In turn, Figure 4.12(a) illustrates a closed queuing network with 3 process, while Figure 4.12(b) shows the same network with process m short-circuited and with $N - n_m$ less. The process m modeled by phase-type distribution is presented in Figure 4.12(c).

Regarding this representation, Marie and Stewart (1977) show that this isolate analysis can be understood such as a Birth-Death process, where $\lambda_m(n_m)$ and $\nu_m(n_m)$ denote the Birth and Death rates, respectively. This assumption holds that the equilibrium shows in Equation (4.37) cannot be violated. This equation indicates the probability of a client leaves process m in state n_m is equal to the probability of a client arrives at the same process when this state is $(n_m - 1)$.

$$\nu_m(n_m)\pi_m(n_m) = \lambda_m(n_m - 1)\pi_m(n_m - 1) \tag{4.37}$$

From Equation (4.37) and the features of the phase-type distribution which describes the process m, such as shown in Table 4.17, Marie (1980) derived expression to determine $\nu_m(n_m)$ for each isolated process m for all n_m. Hence, $\pi_m(n_m)$ can be computed from the equation (4.38). Consequently, the required performance measures can be determined, such as shown in Equations (4.10), (4.9) and (4.11).

$$\pi_m(n_m) = \begin{cases} \frac{1}{1+\sum_{j=1}^{N}\prod_{n_m=0}^{j-1}\frac{\lambda_m(n_m)}{\nu_m(n_m+1)}}, & \text{if } n_m = 0 \\ \pi_m(0)\prod_{j=0}^{n_m-1}\frac{\lambda_m(n_m)}{\nu_m(n_m+1)}, & \text{otherwise,} \end{cases} \tag{4.38}$$

In conclusion, the Marie's method can be summarized in four steps, like as presented by Bolch et al (2006). The first step consists of constructing an auxiliary network with the same topology neglecting the coefficient of variation and initializing the load-dependent service rates with the given service rates μ_m. In the second step a product form algorithm, such as the convolution method presented in the previous section, is used to compute $\lambda_m(n_m)$. Thereby, the third step employs $\lambda_m(n_m)$ in the isolated analysis of each process m to compute ν_m. $\pi_m(n_m)$ are then obtained using Equation (4.38). The fourth step consists of setting the service rates according to each $\nu_m(n_m)$ and, hence, step two is applied again to update $\lambda_m(n_m)$. This procedure repeats until two stop conditions are fulfilled. The first condition denotes that the sum of the mean number of clients in each process must be equal N with a suitable tolerance ε, while the second indicates that the conditional throughputs of each process must be consistent with the topology of the network.

CHAPTER 5: OPEN-PIT MINE AS A QUEUING SYSTEM

"One of the most important aims of science is to try and explain what happens in the world around us. Sometimes we seek explanations for practical ends."
Samir Okasha

Banks (2000) argues that one of the main advantages of using simulation tools is that, once we have developed a valid simulation model, we can explore new policies, operating procedures, or methods without the expense and disruption of experimenting with the real system. Such argumentation can be also considered in the analytical approximation context, assuming that we have a valid queuing model. The question at this point is *What is a valid queuing model?*. It is important reinforce that both DES simulation and DES through Markovian properties are approximation methods. However, the obtained results using DES simulation are featured by measures of dispersion such variance and confidence bounds, while the ones acquired through analytical methods are exact.

Taken this into account, this Chapter, through the quantitative method[1], presents a methodology that concerns stochastic DES system through Markovian properties discussed in the previous Chapters. Thus, we explore Markovian concepts to design a load-haulage cycle of an open-pit mine, subject to stochastic nature, as a closed queuing network. As a result, the Chapter presents DES Markovian models which employ methods of first-moment and second-moment approximation. The goal is to compare the results by the mentioned analytical methods and the results acquired by a standard simulation tool. Like this, the methodology presented has an inductive argument. According to Okasha (2016), although this argument does not guarantee the certainty of the result, it has the purpose of widening the scope of knowledge and, in addition, nonetheless seems like a perfectly sensible way of forming beliefs.

5.1 DES Simulation

As discussed previously, a process through which a system model is evaluated numerically whose the data from the process are used to estimate many interest measures is known as simulation (Cassandras and Lafortune, 2009). In this section, we discuss a standard DES

[1] According to Marconi and Lakatos (2006), the quantitative data analysis is described through observable data, indicators, and trends.

simulation tool named as SIMAN language (Profozich and Sturrock, 1995), which is the basis of the Arena Rockwell software (Vamanan et al, 2004).

This language is a common way to model stochastic DES, in which the latest implementation consists of the very flexible simulation environment known as Arena (Pegden and Davis, 1992). This software is a very attractive tool that supports all basic steps of a simulation study. In addition, it is very simple to use. Nowadays there is a wide range of DES problems solved through this tool, mainly solutions that concern open-pit mine problems. Tan et al (2012) present an optimization methodology for optimizing the transportation cost in an open-pit copper mining. In this study, to estimate the maximum mine capacity, the authors used Arena software for design the open-pit copper mining and optimize the number of trucks.

Many others studies related to open-pit mine have been conducted using the Arena software. For instance, Torkamani and Askari-Nasab (2012) implement a DES simulation model using this software to analyze the truck-shovel haulage system in open-pit mining linked to the short-term plans. According to the authors, this strategy guarantees that the operational plans will follow the strategic plan in place. Another application that concerns stochastic DES simulation using Arena Software in the open-pit mining context is presented by Tan and Takakuwa (2016). In order to find an effective truck dispatching system, the authors propose a practical simulation model using Arena Software and VBA(Visual Basic for Applications) language.

5.2 Load-haulage cycle

A fundamental goal of any mining project is maximizing the ore extraction at low cost, i.e., the goal is maximizing the iron production index. At the operational level, the aim is ensuring the effective use of resources. Torkamani and Askari-Nasab (2012) state that mining equipment is one of the highest capital expenditure in a mine. Considering this aspect, these authors proposed a methodology based on simulation to analyze the haulage system in open-pit mines. According to Torkamani and Askari-Nasab (2012), an efficient truck-shovel system can reduce the costs of haulage, operation, and maintenance. Therefore, this analysis is extremely important for efficient use of those resources (truck and shovel) and it can contribute to improving the iron production index.

As mentioned previously, Torkamani and Askari-Nasab (2012) implements a DES simulation model in Arena (Rockwell Softwares) in order to simulate the load-haulage system in an iron ore open-pit mine. In this study, each simulation scenario consists of a distinct combination of a number of trucks and shovels. The goal is to maximize the iron production index at lowest possible operating cost. As the use of simulation requires high computational effort, sometimes it is not possible to try all feasible scenarios. The strategy taken by Torkamani and

Askari-Nasab (2012) then considers proper indicators to chose the scenarios to be evaluated.

Those indicators were based on utilization rates of trucks and shovels. At each step (new scenario), the utilization rates of trucks and shovels were observed and the number of each one could be increased if the rates were high. Using this strategy, we can get a DES simulation in which it is not interesting to add equipment because this action does not increase the ore production.

Ercelebi and Bascetin (2009) propose another strategy for allocation of trucks in mines. In this work, the authors applied closed queuing theory. This method divides the typical shovel-truck-haulage system into four steps:

1. Shovel (Loading the trucks);
2. Loaded Haulage road (traveling loaded);
3. Dump (emptying the trucks);
4. Empty Haulage road (traveling empty).

Basically, each truck goes to load the site and awaits until the loading process is completed. Following, the trucks go to the dump site and dumps the iron into a crusher.

The measure $\bar{T}_{cycle}$ denotes the *mean cycle time*. In other words, $\bar{T}_{cycle}$ is the mean time that a truck completes one load haulage cycle, as shown in Figure 2.2. Such measure depends on the dynamic of the system and must be estimated by some stochastic DES strategy. In a standard load-haulage system, which is composed of only the four steps mentioned, $\bar{T}_{cycle}$ is obtained by Eq. (5.1),

$$\bar{T}_{cycle} = \bar{T}_s + \bar{T}_{lh} + \bar{T}_d + \bar{T}_{eh} \tag{5.1}$$

where $\bar{T}_s$ and $\bar{T}_d$ are the mean response time at the process Shovel and Dump, respectively, while $\bar{T}_{lh}$ denotes the mean response time to transport loaded and $\bar{T}_{eh}$ represents the mean response time to transport empty.

According to Ercelebi and Bascetin (2009), in a load-haulage system, an estimative for the iron production index $E[Prod]$ over a given time period of interest can be computed by Equation (5.2).

$$E[Prod] \approx N \cdot C \cdot \frac{T_{horizon}}{\bar{T}_{cycle}} \tag{5.2}$$

where N, C and $T_{horizon}$ denote the number of trucks, the truck's capacity and the time horizon, respectively.

5.3 DES model for a load-haulage cycle through Markovian properties

This section describes a load-haulage cycle through Markovian properties. It is considered first-moment and second-moment approximation methods. Both models are based on the knowledge discussed in the early Chapter. The first-moment approximation assumes that the random variables which classify the load-haulage cycle follow the exponential distribution, even though it's not true in the real system. The second-moment approximation considers the coefficient of variation and defines the same random variables in terms of phase-type distribution.

The load-haulage cycle considered in this study is composed of 10 processes in which 4 are operational stops, such as shown in Table 5.1. Each one of then is featured by random variables which follow general distributions. Moreover, the processes are characterized by a kind of queue discipline. An especial attention is required for the queue discipline of the process 3 and 6. Usually, in a load-haulage cycle, the 'resource' 'road' can be shared, at the same time, by all trucks that compose the cycle. However, passing through is prohibited. In this study, the roads processes were divided into small slots of roads to guarantee such premise. Thus, let Q_j be the number of slots of process j ($j \in (3,6)$), Q_j-FIFO is defined as a queue discipline that represents this behavior.

Table 5.1: Processes of a Load-haulage cycle

Id	Process	Expected service time	Coefficient of variation	Queue Discipline	Prob. of Occourrence
ST_1	Maneuver to load	$E[ST_1(\boldsymbol{\theta})]$	$0 < cv_1 \leq 1$	FIFO	-
ST_2	Shovel site	$E[ST_2(\boldsymbol{\theta})]$	$0 < cv_2 \leq 1$	FIFO	-
ST_3	Loaded Haul. road	$E[ST_3(\boldsymbol{\theta})]$	$0 < cv_3 \leq 1$	Q_3-FIFO	-
ST_4	Maneuver to Dump	$E[ST_4(\boldsymbol{\theta})]$	$0 < cv_4 \leq 1$	FIFO	-
ST_5	Dump site	$E[ST_5(\boldsymbol{\theta})]$	$0 < cv_5 \leq 1$	FIFO	-
ST_6	Empty Haul. road	$E[ST_6(\boldsymbol{\theta})]$	$0 < cv_6 \leq 1$	Q_6-FIFO	-
ST_7	Preventive maintenance	$E[ST_7(\boldsymbol{\theta})]$	$0 < cv_7 \leq 1$	FIFO	p_1
ST_8	Corrective maintenance	$E[ST_8(\boldsymbol{\theta})]$	$0 < cv_8 \leq 1$	FIFO	p_2
ST_9	Supply	$E[ST_9(\boldsymbol{\theta})]$	$0 < cv_9 \leq 1$	FIFO	p_3
ST_{10}	Shift change	$E[ST_{10}(\boldsymbol{\theta})]$	$0 < cv_{10} \leq 1$	FIFO	p_4

Once that the processes are featured by general distributions, each one of them was written in terms of the expected time service $E[ST_j(\boldsymbol{\theta})]$ and the coefficient of variation cv_j, for $1 \leq j \leq 10$. The term $\boldsymbol{\theta}$ indicates that each $E[ST_j(\boldsymbol{\theta})]$ depends on the initiative set $\{\theta_1, \theta_2, \ldots, \theta_q\}$. In addition, the processes ST_7, ST_8 , ST_9 and ST_{10} are operational stops associated with the probability of occurrence p_1, p_2, p_3 and p_4, respectively.

5.3.1 First-moment approximation

Since that our goal is to compute $E[Prod]$, this section present an analytical approximation strategy to compute $\bar{T_{cycle}}$. It consists of a first moment approximation because the load-haulage cycle discussed is modeled as a closed queuing network with all service time exponentially distributed, while it is not true in the real model. In other words, we ignored the coefficient of variation of the distributions which characterized each processes of the system. The reason for that is to see load-haulage cycle as a Markovian model and, then, to make applicable the product form method presented previously.

Figure 5.1 illustrates the load-haulage cycle modeled as a closed queue network. A similar model was presented in Ribeiro et al (2016). The main difference among these models are the road processes 3 and 6. While Ribeiro et al (2016) classifies the disciplines of such nodes as IS (Infinity Service), this study rate them as Q_j-FIFO. Such was mentioned early, the reason for that is the fact that passing through is prohibited.

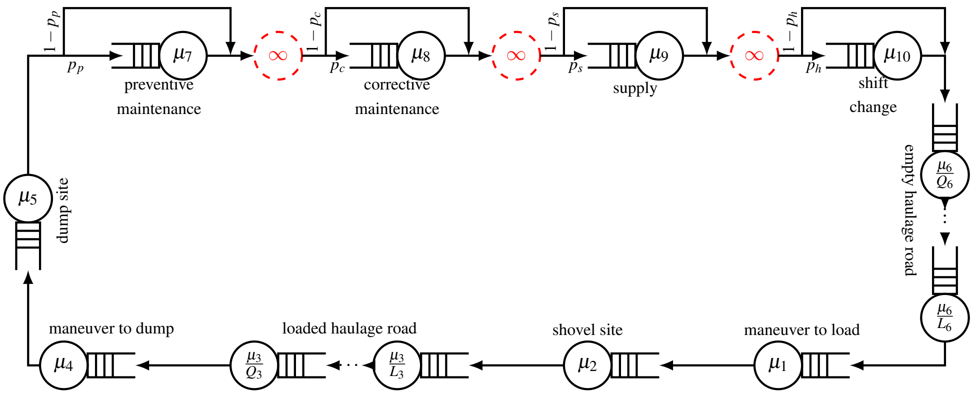


Figure 5.1: Closed queuing network using first moment approximation

As it can see in Figure 5.1, three fictitious nodes were added. Designed by dashed lines, such nodes are important because establish transient decisions between operational stop nodes. Once that the service rate of each fictitious node is infinity, they do not change the final result.

To compute the mean cycle time it was derived an expression from the Equation (5.1), such is presented in Equation (5.3):

$$\bar{T}_{cycle} = \sum_{m=1}^{6} \bar{T}_m + p_p\bar{T}_7 + p_c\bar{T}_8 + p_s\bar{T}_9 + p_h\bar{T}_{10}. \tag{5.3}$$

Since that the occurrence of each operational stop is connected with a transient probability, only a proportion of the mean time spent in each operational stops was added in the total cycle time.

At last, the convolution technique presented in the previous Chapter can be applied for computing the mean response time of each node, accordingly with Equation (5.3). Assuming that the truck's capacity, the number of trucks and the time horizon are known, the iron production index can be evaluated from Equation (5.2).

5.3.2 Second-moment approximation

Different from the modeling presented previously, the coefficients of variation of the distributions which characterize the processes are now considered. Figure 5.2 shows the load-haulage investigated in this paper designed as a closed queuing network. As it can see, the only difference between such model and the model illustrated in Figure 5.1 is the fact that processes with FIFO discipline are represented by a grid of nodes instead of a unique node exponentially distributed.

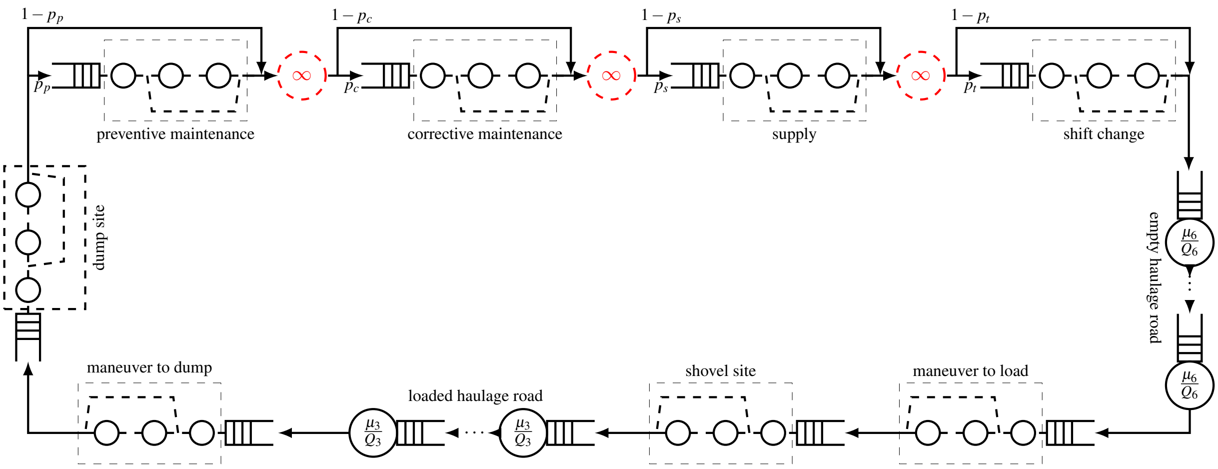


Figure 5.2: Closed queuing network using second moment approximation

Each grid of nodes represents a phase-type distribution, such was explained in section 4.6. Thereby, the Marie's method can be applied to compute the marginal probabilities and, consequently, the mean response time of each process. Since that such performance measure is known, Equation (5.3) can be used for evaluating the mean response time. Finally, presuming that the truck's capacity, the number of trucks and the time horizon are known, the iron production index can be computed from Equation (5.2).

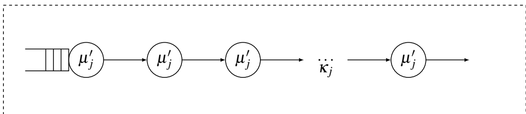
5.4 First batch of the computational experiment

As already mentioned, the motivation for this study is a simulation optimization problem, in which, the objective function is a stochastic model. The decision variables, $\{\theta_1, \theta_2, \ldots, \theta_q\}$, are initiatives that impact on the service times of model's processes. Since that there are many possible combinations for $\{\theta_1, \theta_2, \ldots, \theta_q\}$ and it can not be possible to evaluate all possibilities, it is important developing an efficient simulation method to increase the amount of combination evaluated.

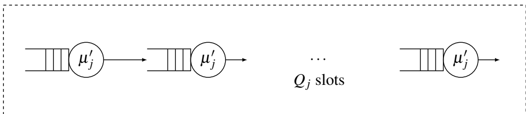
In this section, we wish to make inference about the difference between the results obtained from the analytical approximation methods and the results acquired from a model developed from a well-known simulation tool. Since that the expected iron production index, $E[Prod]$, depends on the decision variables $\{\theta_1, \theta_2, \ldots, \theta_q\}$, it is defined such measure in terms of $\boldsymbol{\theta}$. Then, $E[Prod] = E[Prod(\boldsymbol{\theta})]$. Regarding this analysis, $E[Prod(\boldsymbol{\theta})]$ is evaluated for different scenarios by the mentioned methods.

The goal is to provide evidence of equivalence between them. For this analysis, it was considered a load haulage cycle which depicts a mining front of real open-pit mine. The system is composed of 10 processes in which 4 are operational stops, such as shown in Table 5.1. Once that the processes are featured by general distributions, each one of them was written in terms of the expected time service $E[ST_j(\boldsymbol{\theta})]$ and the coefficient of variation cv_j, for $1 \leq j \leq 10$. The term $\boldsymbol{\theta}$ indicates that each $E[ST_j(\boldsymbol{\theta})]$ depends on the initiative set $\{\theta_1, \theta_2, \ldots, \theta_q\}$. In addition, the processes ST_7, ST_8 , ST_9 and ST_{10} are operational stops associated with the probability of occurrence p_1, p_2, p_3 and p_4, respectively.

With the exception of ST_3 and ST_6, the processes are described by Erlang distribution with $\kappa_j \approx 1/cv_j^2$ and $\mu_j' = \kappa_j / E[ST_j(\boldsymbol{\theta})]$. The parameters κ_j and μ_j' are the number of phases and the service rate of the Erlang distribution that describes the process j ($\forall\ j\ \&\ j \neq (3,6)$). This representation is showed in Figure 5.3(a).



(a) Erlang distribution for $\forall\ j\ \&\ j \neq (3,6)$.



(b) Sequence of Exponential distribution for $j = (3,6)$.

Figure 5.3: Processes ST_j representation

As explained early, the processes ST_3 and ST_6 are depicted by a sequence of Q_3 and Q_6 slots to avoid passing through. Each slot is a random variable. For simplicity, it was considered that these variables follow exponential distributions with rates $\mu_3' = Q_3/E[ST_3(\boldsymbol{\theta})]$ and $\mu_6' = Q_6/E[ST_6(\boldsymbol{\theta})]$. The number of slots Q_3 and Q_6 are approx. $1/cv_3^2$ and $1/cv_6^2$, respectively. Figure 5.3(b) illustrates this representation.

Regarding the experiment analysis, a sampling based on Latin Hypercube Sampling (Iman, 2008) was used to generate a meaningful sample set. Thereby, it was considered as parameters the number of trucks, the expected service time, the coefficient of variation and the probabilities of occurrence, such as shown in Table 5.1. It was generated 30 samples in which the minimum and maximum values for the parameters $E[ST_j(\boldsymbol{\theta})]$ $(\forall j)$ are 10(min) and 30(min), respectively, while the minimum and maximum for the number of trucks are 1 and 20. Moreover, the generated values for the coefficient of variation cv_j $(\forall j)$ must be greater then 0 and less or equal 1, while the probabilities of occurrence $(p_1,\ p_2,\ p_3,p_4)$ must be greater then 0 and less then 0.5. Table A.1(Appendix A) shows the sample generated.

Since that the goal is to compare the analytical approximations methods with a standard simulation tool, the Arena Rock tool is used to generates a SIMAN model which represents the load haulage cycle described. This thesis does not discuss in details such model because its understanding is relatively simple. Essentially, the processes are representing in the mentioned software as Time-Delay-Release blocks. In addition, the road processes (3 and 6) are depicted by loop structures which represent the Q-FIFO discipline. The scenarios presented in Table A.1 were run by the first and second moment approximation methods and by the SIMAN model.

It was considered the Caterpillar 793F trucks with the nominal payload capacity of 226.8 tons (approx.). For the analytical approximation methods, first end second moment, $E[Prod(\boldsymbol{\theta})]$ can be estimated by Equation (5.2) for a well-known time horizon, while the simulation model the interest measure must be estimated by the average of many runs. Then, each scenario has been run 6 times. Moreover, for all methods, it was considered a time period of interest of 365 days $24/7$. Table 5.2 shows the $E[Prod(\boldsymbol{\theta})]$ obtained by the three methods and the time taken to run each scenario.

It was considered two experiment analysis. The first consists of comparing the SIMAN simulation model with the first-moment approximation method, while the second consists of comparing the same simulation model with the second-moment approximation method. For this purpose, the experiment analysis was conducted applying the equivalence test based on TOST (Two One-Sided Test) presented by Walker and Nowacki (2011). The equivalence between the analytical approximation methods and the SIMAN simulation model can be established at the 2.5% significance level if a 95%-confidence interval for the fraction between the methods is contained within the interval from $1-\delta^*$ to $1+\delta^*$. It is accepted as a tolerance deviation δ^* a value no greater than 2% of the expected iron production index.

Let $E[Prod(\theta)_{first}]_\ell$, $E[Prod(\theta)_{second}]_\ell$ and $E[Prod(\theta)_{siman}]_\ell$ be the obtained expected iron production index applying the first-moment, second-moment and SIMAN models over the scenario ℓ, respectively. Then, the terms $\bar{\omega}_1$ and $\bar{\omega}_2$ denote the fractions described by Equation (5.4) and (5.5), respectively.

$$\bar{\omega}_1 = \frac{E[Prod(\theta)_{first}]_\ell}{E[Prod(\theta)_{Siman}]_\ell} \quad 1 \leq \ell \leq 30 \tag{5.4}$$

Table 5.2: Estimated production: Analytical Models x SIMAN

ℓ	$E[Prod(\theta)]$ $\times 10^5$ tons			Exec. time (min)		
	First	Second	SIMAN	First	Second	SIMAN
1	26.63	28.87	32.01	0.0332	0.0776	8.7114
2	36.86	38.96	38.74	0.0077	0.0251	7.8631
3	30.53	32.57	36.78	0.0080	0.0206	18.3722
4	44.27	47.33	49.60	0.0029	0.0241	20.9080
5	38.08	41.62	39.69	0.0007	0.0287	7.3490
6	36.75	38.95	42.53	0.0005	0.0114	7.7666
7	35.31	37.16	38.04	0.0036	0.0242	21.6881
8	40.12	44.10	40.46	0.0010	0.0405	8.8044
9	33.01	33.54	34.07	0.0003	0.0067	5.8298
10	37.33	40.33	38.87	0.0007	0.0217	7.1016
11	41.98	45.43	42.62	0.0005	0.0159	6.9217
12	39.70	40.08	40.18	0.0007	0.0203	7.6925
13	28.37	30.79	33.03	0.0003	0.0074	6.6321
14	37.04	43.21	41.87	0.0022	0.0852	17.0353
15	35.51	37.24	37.66	0.0011	0.0172	10.9911
16	41.09	47.16	46.45	0.0005	0.0331	8.2866
17	18.82	19.53	19.79	0.0003	0.0039	5.2890
18	15.94	16.19	16.18	0.0005	0.0036	5.8430
19	33.94	36.13	37.80	0.0004	0.0101	6.5653
20	38.89	41.97	42.96	0.0035	0.0156	25.5643
21	27.22	28.64	30.13	0.0007	0.0046	8.2612
22	37.39	39.96	38.01	0.0004	0.0076	7.2791
23	56.14	55.64	59.09	0.0017	0.0371	16.4587
24	37.98	42.83	41.38	0.0007	0.0274	7.4277
25	41.65	44.22	43.73	0.0005	0.0182	8.0147
26	36.91	40.59	40.68	0.0010	0.0150	14.4363
27	29.98	31.95	33.30	0.0003	0.0031	6.9262
28	41.67	44.57	42.72	0.0005	0.0189	6.1761
29	39.55	43.36	39.56	0.0006	0.0233	6.9866
30	14.73	15.03	15.09	0.0006	0.0027	6.3721

$$\bar{\omega}_2 = \frac{E[Prod(\theta)_{second}]_\ell}{E[Prod(\theta)_{Siman}]_\ell} \quad 1 \leq \ell \leq 30 \tag{5.5}$$

Thus, the TOST which compare the first-moment and the SIMAN model can be expressed by the hypotheses:

$$\begin{cases} H_{0_1}^{+}: & \bar{\omega}_1 \geq 1+\delta^* \\ H_{a_1}^{+}: & \bar{\omega}_1 < 1+\delta^* \end{cases} \qquad \begin{cases} H_{0_1}^{-}: & \bar{\omega}_1 \leq 1-\delta^* \\ H_{a_1}^{-}: & \bar{\omega}_1 > 1-\delta^* \end{cases}$$

while the TOST which compare the Second-moment and the SIMAN model can be expressed by the hypotheses:

$$\begin{cases} H_{0_2}^{+}: & \bar{\omega}_2 \geq 1+\delta^* \\ H_{a_2}^{+}: & \bar{\omega}_2 < 1+\delta^* \end{cases} \qquad \begin{cases} H_{0_2}^{-}: & \bar{\omega}_2 \leq 1-\delta^* \\ H_{a_2}^{-}: & \bar{\omega}_2 > 1-\delta^* \end{cases}$$

Regarding these test, the statistical software R (R Core Team, 2013) was used. Table 5.3 summarizes the results obtained.

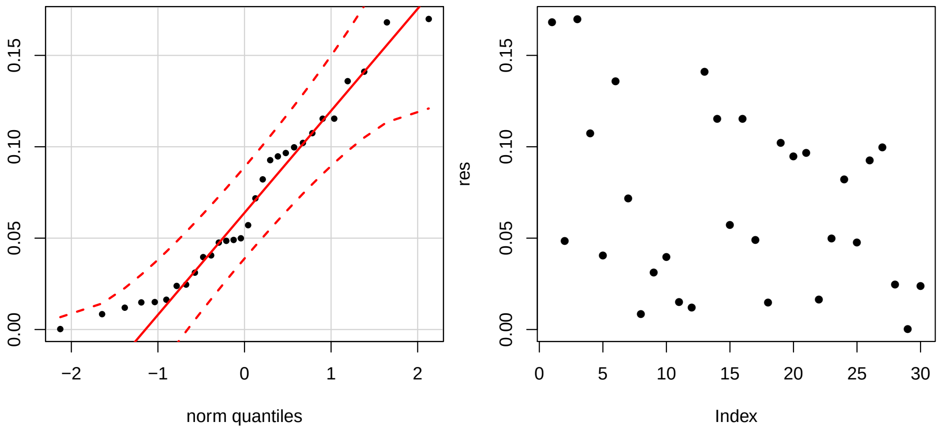
Table 5.3: TOST - Analytical Models x SIMAN

TOST Par.	$\delta^* = 2\%$ Sig. level of 2.5% 95%-confidence interval
First-moment x SIMAN	
$H_{0_1}^{+}$	$t = -10.01$ $DF = 29$ P-value=3.22×10^{-11} Null hypothesis $H_{0_1}^{+}$ can be rejected
$H_{0_1}^{-}$	$t = 5.51$ $DF = 29$ P-value=1 Null hypothesis $H_{0_1}^{-}$ cannot be rejected
Second-moment x SIMAN	
$H_{0_2}^{+}$	$t = -2.67$ $DF = 29$ P-value=6.13×10^{-3} Null hypothesis $H_{0_2}^{+}$ can be rejected
$H_{0_2}^{-}$	$t = 1.52$ $DF = 29$ P-value=6.98×10^{-2} Null hypothesis $H_{0_2}^{-}$ cannot be rejected

The normality residual assumption for both tests was validated from the Shapiro-Wilk method. In addition, the independence residual assumption was checked by the Durbin-Watson. Figures 5.4(a) and 5.5(a) show the normality of the residuals of $\bar{\omega}_1$ and $\bar{\omega}_2$, respectively, while Figures 5.4(b) and 5.5(b) illustrate the independence of residuals of the same terms.

Applying the TOST test, it was observed that the p-values for the null hypotheses $H_{0_1}^{+}$ and $H_{0_2}^{+}$ cannot be rejected. As a result, the equivalence between both approximation method, first-moment and second-moment can not be declared, within the equivalence margin of $\delta^* = 2\%$, with a significance level of 2.5%. These results can be seen in Figure 5.6.

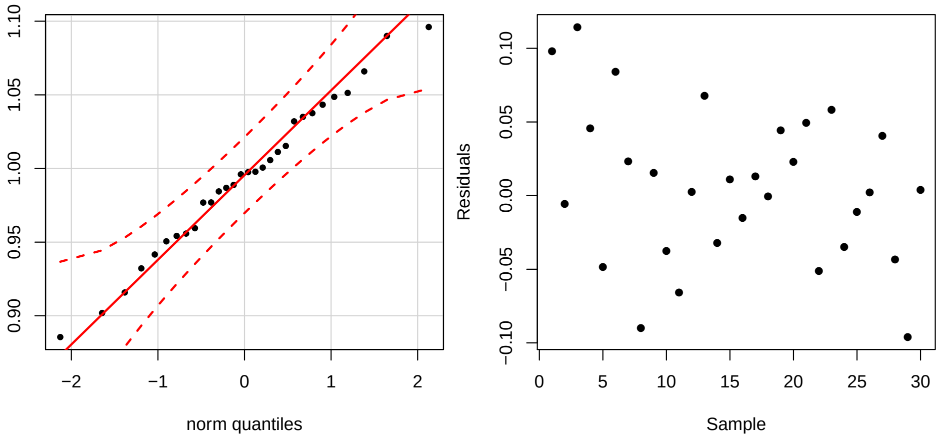
Although we cannot affirm the equivalence between the models, the analytical approximation methods are interesting alternatives to evaluate stochastic DES, especially when the model is associated with an optimization problem. Table 5.2 also shows the computational times expend in each scenario. Assuming that simulation model represent accurately the real model, it is necessary to answer if such model can be substituted by an analytical approximation model. From optimization point view, it is strongly appreciated this substitution once the time



(a) Normality of the residuals: W = 0.93834 and p-value = 0.08207

(b) Independence of the residuals: DW = 1.6255 and p-value = 0.1054

Figure 5.4: Assumptions of the hypothesis test - First-Moment and SIMAN ($\bar{\omega}_1$)



(a) Normality of the residuals: W = 0.98825 and p-value = 0.9792

(b) Independence of the residuals: DW = 2.0981 and p-value = 0.5284

Figure 5.5: Assumptions of the hypothesis test - Second-Moment and SIMAN ($\bar{\omega}_2$)

taken to run each scenario is considerably faster than the computational time spent using simulation models. It means that feasible space of solutions can be explored more broadly. We can see that the First-moment method is around ten thousand times faster than the SIMAN model, while the Second-moment method is almost one thousand times faster than the same simulation model.

The analysis was also conducted changing the number of trucks n from 1 to 20 and

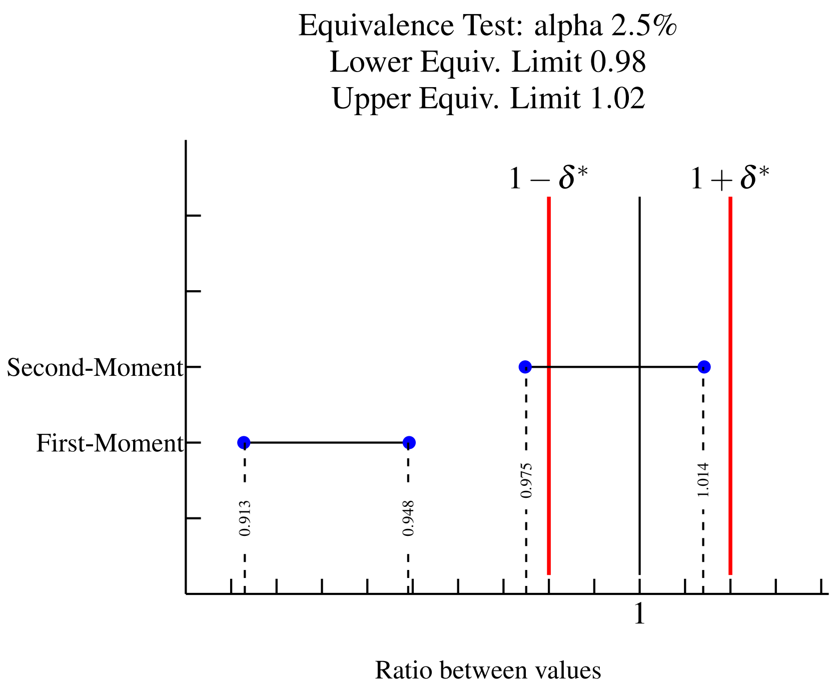


Figure 5.6: Equivalence testing 2.5%-significance level and 95%-confidence interval

applying in the scenarios presented in table A.1. Figures B.1(a), B.2(a), B.3(a), B.4(a), B.5(a), B.6(a), B.7(a) and B.8(b) show, for each scenario ($1 \leq \ell \leq 30$), the increasing of *total production index* in terms of the number of trucks. Figures B.1(b), B.2(b), B.3(b), B.4(b), B.5(b), B.6(b), B.7(b) and B.8(b) depict, for same scenarios, the computational time taken to run. Such figures also present the results obtained by a Hybrid method. Next chapter presents this new method. Although there are differences between the methods, we can see in the Figures (Appendix B) that in most cases the analytical and the simulation methods follow the same ordinal pattern. Since that the motivation of this thesis is a project portfolio problem, we understand that the analytical approximation methods can be used to rank the portfolio projects.

CHAPTER 6: AGGREGATING MAX-PLUS ALGEBRA & MARKOV CHAIN

"Sometimes one should do a completely wild experiment, like blowing the trumpet to the tulips every morning for a month. Probably nothing would happen, but what if it did?"
George H. Darwin

This chapter presents a new simulation methodology, proposed in (Ribeiro et al, 2018), for analysis of DES that aggregates Max-Plus Algebra and Markov Chain faster than standard simulation tools extending the computational modeling power of the stochastic Max-Plus system. The same open-pit mine with stochastic behavior is considered as a numerical example in which we are interested in evaluating the expected value of the iron production index. As mentioned in Chapter 2(Section 2.1), system is composed of many connected queues characterized by general distribution. Figure 2.2 illustrates an abstraction of such system.

The motivation for developing this hybrid model is the fact that open-pit mines usually are systems subject to synchronization, delay and decision phenomena (as the stop decision shown in Figure 2.2). According to Baccelli et al (1992), Max-plus DES can be characterized as the class of DES in which only synchronization and no concurrency or choice occur. Therefore, the methodology presented in this Chapter consists of dividing the system in different modes of operation without decision phenomena. This strategy allows the analysis of each mode by Max-Plus Algebra. Furthermore, the probability of the system operates in a specific mode is obtained from a Markov Chain.

Such methodology is detailed in Section 6.2. Previously, Section 6.1 describes concepts of Max-Plus Algebra. Section 6.3 presents a second batch of experiment analysis where the same TOST of previous test is applied. Regarding this test, $E[Prod(\theta)]$ is evaluated for different scenarios by the hybrid method and by the same standard simulation tool (SIMAN).

6.1 Fundamentals of Max-Plus Algebra

This section present concepts of an alternative methodology that concern with simulation modeling which usually is faster than standard simulations tools. The Max-Plus Algebra is a tool used for modeling systems that are subject to synchronization and delay phenomena Dias et al (2016). Although not so widespread, this algebra consists of a powerful tool to understand the dynamic of several DES system.

Fundamental notions about Max-Plus algebra are presented by Baccelli et al (1992). Essentially, Max-Plus systems are described by dioid structure. According to Baccelli et al (1992) “dioids are structures that lie somewhere between conventional linear algebra and semi-lattices endowed with an internal operation”. Therefore, a Max-Plus algebra is defined by a set $\mathbb{R}_{\max} \cup \{-\infty\}$ together with the two internal operations $\oplus$ and $\otimes$, which denote the max value and the sum of values, respectively. According to Baccelli et al (1992), the operation $\oplus$ is associative, commutative and idempotent and it contains a neutral element $\varepsilon_{\oplus} = -\infty$. While the operation $\otimes$ is associative (but not necessarily commutative) and distributive at left and at right with respect to $\oplus$ and it contains a neutral element $\varepsilon_{\otimes} = 0$.

For modeling a DES simulation system through Max-Plus algebra it is important to consider the use of TEG (Timed Event Graphs). According to Baccelli et al (1992), TEG are Timed Petri Nets in which all places have single upstream and single downstream transitions and are used to model DES systems featured by synchronization and delay phenomena. Due to Dias et al (2016), TEG together with Max-Plus algebra is used for modeling several systems which are commonly addressed by the theory of DES. In this combination, the Max-Plus algebra must be understood as the mathematical structure of the TEG. This structure consists of a set recurrence equations, which are linear in the Max-Plus algebra and offers a strong analogy with conventional linear dynamic systems Dias et al (2016).

A concept about TEG is a fact that it is a bipartite and oriented graph. The fundamental goal is that object flow, named tokens, depends on the event occurrences. Basically, a TEG is described by a place/transitions system (Baccelli et al, 1992), where the number of tokens in a place must be represented by $\mathbb{N}^{+}$. Each place is connected by incoming and outcoming arcs which describe the token flow.

The transition reflects an action of the system and the token’s positions can change according to its occurrence. Further, each transition is associated with a firing function, which is the rule that defines the sojourn time among the enabling transition and its fire. A disabled transition becomes enabled when there are enough tokens (at least one) in each place connected with it by incoming arcs. Fundamentally, there are two kinds of fire function. The first (▬) denotes that a transition fires immediately after being enabled, while the second (▭) denotes that it is necessary a sojourn time before its fire. The figure 6.1 shown a simple TEG with three places and two transitions, where places are representing by cycles and transitions by bars (▬ or ▭). The black cycles indicate the number of tokens in each place.

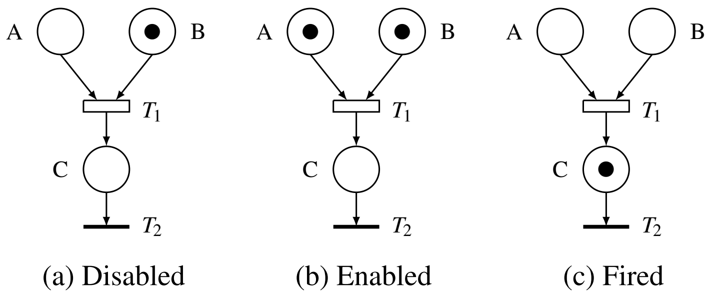


Figure 6.1: Simple TEG.

As we can see in Figure 6.1(a), there are not tokens in place A. This situation indicates that the transition T_1 is disabled. Since exist tokens in place A and B, Figure 6.1(b), T_1 becomes enabled. After the sojourn time T_1 is fired and, as shown in Figure 6.1(c), a token is added in the place C enabling the transition T_2.

An example of DES system modeled with TEG is shown in figure 6.2. This figure represents a simple queue system composed by one server and a set of events $\{a,d\}$, where a and d denote the arrival and depart of clients, respectively. In this figure, the place Q denotes the queue, while S represents the server. The place A indicates whether the server is idle or busy. This condition depends on the number of tokens in this place. For a queue with just one server, this number must be 0 or 1.

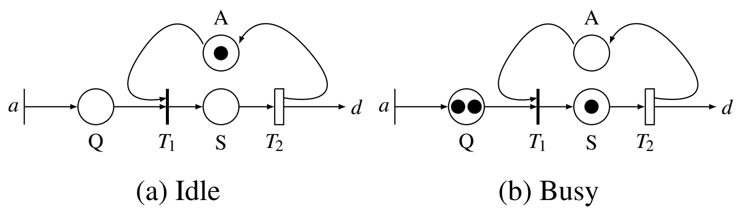


Figure 6.2: Simple Queue with one server

In figure 6.2(a), the server S is idle and there are no clients in the queue Q because the number of tokens in these places is zero. In the figure 6.2(b), there is one token in place S and zero tokens in place A. This situation indicates that the server S is busy. In addition, two clients are waiting in the queue Q.

The event a is spontaneous and depends on an external occurrence. When this event happens, a token is placed in Q and the transition T_1 is enabled (if its condition is disabled) only if there is a token in place A. Otherwise, its condition keeps disabled. Since T_1 is instantaneous, its fire occurs exactly when its condition changes from disabled to enabled. This action removes one token of the places that enabled its fired (Q and A). The event d is associated with the transition T_2. It means that a depart occurs when T_2 is fired. This transition is enabled whether there is a token in place S and, after a sojourn time, a token is removed from S and a token is placed back in A.

As it was mentioned, the figure 6.2 shown a simple queue with one server modeled using TEG. From this model, it is possible to infer equations that describe the queue's dynamic. Let $T_i(k)$ be the instant of the *k-th* fired of the transition T_i $(i = \{1,2\})$ and $u(k)$ the instant of *k-th* occurrence of the event a. Then, $T_1(k)$ can be described by $T_1(k) = \max(u(k), T_2(k-1))$. Let z be the sojourn time of T_2. Then, $T_2(k)$ can be described by $T_2(k) = T_1(k) + z$. Taken into account these statements, the behavior of the queue can be depicted by the system of equations (6.1):

$$\begin{aligned} T_1(k) &= \max(u(k), T_2(k-1)) \\ T_2(k) &= T_1(k) + z \\ Y(k) &= T_2(k) \end{aligned} \quad \leftrightarrow \quad \begin{aligned} T_1(k) &= u(k) \oplus T_2(k-1) \\ T_2(k) &= T_1(k) \otimes z \\ Y(k) &= T_2(k) \end{aligned} \tag{6.1}$$

where $Y(k)$ denotes the instant of the *k-th* departure. As we can see, the right-hand side of the Equation (6.1) represents the model described by Max-Plus algebra. It is a relevant representation once that allows describing the temporal dynamic of a non-linear system in a linear way. Consequently, important measures can be easily computed from this system of equations.

This representation can be extended to model several complex DES problem. Xie (2014) discuss how manufacturing system can be modeled as a Max-Plus algebra. Some applications in the same context are summarized by Heidergott et al (2014). The study presented by Dias et al (2016) deals with synchronization of events modeled by TEG in a manufacturing system. According to the authors, the proposed method facilitates the determination of the best production rate for the system studied and provides a better quality final product by synchronizing the run times.

In the transportation field, Goverde et al (1999) present a mathematical modeling framework based on Max-Plus Algebra for setting a railway system. According to the authors, essential dynamic characteristics can be quantified such as minimum cycle time and critical circuits. Airulla et al (2016) consider traffic light system optimization within two nearby intersections in order to come up with the most minimum traffic queue. In order to be able to predict the queue length and to determine the duration of traffic lights, the authors modeled the vehicle flow using Max-Plus algebra.

6.2 Hybrid model which aggregates Max-Plus algebra and Markov chain

This section shows how a load haulage cycle, as depicted in Figure 2.2, can be modeled using a hybrid mechanism which aggregates Max-Plus Algebra and Markov Chain, extending the computational power of the stochastic Max-Plus systems. As mentioned, Max-Plus Algebra does not deal with the decision phenomena which are, in the system studied, the possibilities of stopping. Such decisions remove trucks of the load haulage cycle and, after a sojourn time, placed they back. Taken it into account, we can argue that the system switches the number of trucks in operation during the time horizon. Figure 6.3 depicts the load haulage cycle without operational stops.

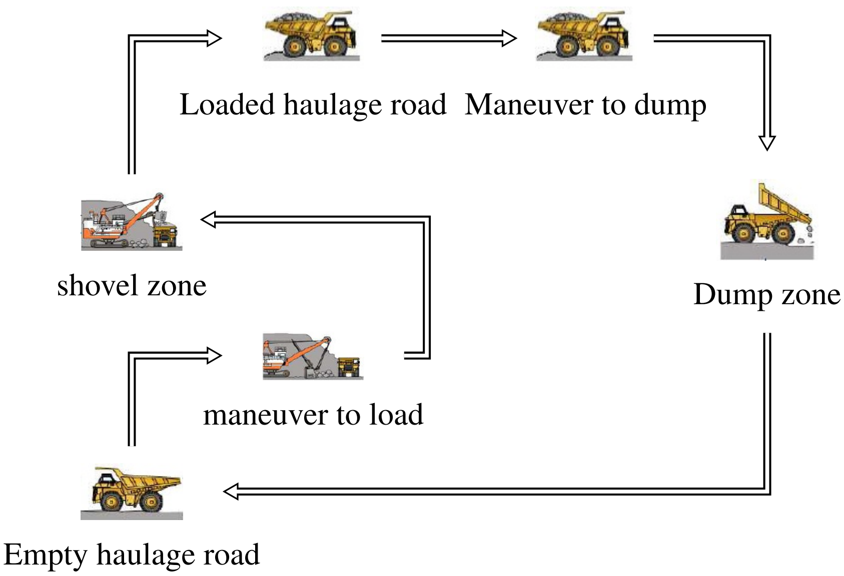


Figure 6.3: Load haulage cycle without decision phenomena

Since that such cycle is just subjected to synchronization and delay phenomena, Max-Plus algebra can be applied to model its dynamic. Another point is that the number of trucks in operation change randomly during a known $T_{horizon}$. Let N be the fleet's size. The symbol $\nearrow$ indicates that a truck exits load haulage-cycle because of some operational stop occurrence ('$-$' one less truck), while the symbol $\swarrow$ indicates that a truck ended the operational stop service and returned to the load-haulage cycle ('$+$' one more truck). Figure 6.4 illustrates this behavior in which the number of trucks switches over the time horizon due to decision phenomena.

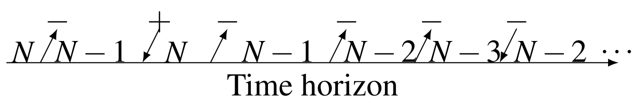


Figure 6.4: Switching of state at the time horizon

We are interested in evaluating the expected value of the iron production index, $E[Prod(\theta)]$. Since that methodology consists of aggregating Max-Plus Algebra and Markov Chain, the result of this aggregation is a DES simulation model. Therefore, different of the analytical approximation model, the interest measure must be estimated by the average of many runs.

Let $T_{cycleMP}$ be the Max-Plus definition *cycle-time vector* of the DES system. According to Heidergott et al (2014), this definition indicates that $T_{cycleMP} = \lim_{k\to\infty} T[k]/k$. For a model with a fixed number of trucks, N, the expected value of $T_{cycleMP}$ can be understood by Equation (6.2).

$$E[T_{cycleMP}] = \frac{\bar{T_{cycle}}}{N} \tag{6.2}$$

Then, assuming that the process associated with the system is stationary and ergodic, $E[Prod(\theta)]$ can be computed in terms of the *cycle-time vector*. Therefore, Equation (5.2) can be rewritten by Equation (6.3) if we assume that $E(1/X) \approx 1/E(X)$.

$$E[Prod] \approx C \cdot \frac{T_{horizon}}{E[T_{cycleMP}]} \tag{6.3}$$

Using Equation (6.3), $E[Prod(\theta)]$ can be computed even when the number of trucks change dynamically. However, for dynamical-structure, the computation of $T_{cycleMP}$ is very costly, even impossible. Taken this into account, the interest measure $E[J(\theta)]$ is obtained by the weighted sum of the production of each possible number of trucks.

Since that N is the fleet's size of the complete system, such shown in Figure 2.2, the number of trucks running around the load haulage cycle depicted in Figure 6.3 can alternate between 0 to N. Let n be this number. Then, $N-n$ is the number of stopped trucks. Let $P(X=n)$ be the probability of the mentioned load haulage cycle operates with n trucks. So, Equation (6.3) can be approximation by Equation (6.4).

$$E[J(\theta)] \approx \sum_{n=1}^{N} P(X=n) \cdot C \cdot \frac{T_{horizon}}{E[T_{cycleMP_n}]} \tag{6.4}$$

The measure $T_{cycleMP_n}$ denotes the *cycle-time vector* of a scenario with a fixed number of trucks, n, running in a system without operational stops. The challenge is to compute $T_{cycleMP_n}$ and $P(X=n)$ for $1 \leq n \leq N$. Thus, Max-Plus Algebra is applied to compute $T_{cycleMP_n}$, while $P(X=n)$ is obtained from a Markov Chain.

There are two aspects behind this approximation. The first assumes that $E[1/X] = 1/E[X]$ while, in general, it is not true. However, $E[1/X] \approx 1/E[X]$ if the coefficient of variation of the stochastic variable is small. The value of such deviation measure can be reduced increasing the number of runs or the size of the simulation (period of interest). The second is related to the operational stops processes. Since that $P(X=n)$ is estimated from Markovian properties, we assume that the service time in these processes are exponentially distributed while it is not necessarily true.

6.2.1 Computing $T_{cycleMP_n}$ for $1 \leq n \leq N$

As it can be seen in Figure 6.3, the load haulage cycle without operational stops is composed of 6 processes. Essentially, the n trucks run around the cycle passing through of all the processes. Each process is a queue system with one server in which the service time follow a general distribution. With exception of the 'load haulage road' and 'empty haulage road', the queue processes are featured by FIFO (First In First Out) discipline.

For the others two processes, an especial attention is required. Usually, the resource 'road' can be shared, at the same time, by all trucks that compose the load-haulage cycle. However, passing through is prohibited. In this study, roads processes were divided into small slots of roads to guarantee such behavior. So, let Q_j be the number of slots and j the index of the road process, Q_j-FIFO was defined as a queue discipline of the queue processes 'load haulage road' and 'empty haulage road'.

Let Z_i be the random variable that describes the service time of the process i. Moreover, let Z'_{3_k} and Z'_{6_k} be the random variable which describes the q slot of the 'road' processes 'load haulage road' and 'empty haulage road', respectively. Considering these definitions, the load haulage cycle without operational stops was modeled with TEG, such as shown in Figure 6.5.

The sojourn time of each fire function (▭) is defined by the random variable which describes the service time of the process. Let the highlighted place in Figure 6.5 (blue place) be the start point of the load haulage cycle and n, number of truck, be the amount of tokens in TEG model. The variable $Y_i(k)$ denotes the instant of the k-th departure in the processes i. Since that processes 3 and 6 are divided in Q slots, this variable is represented by $Y_{i_q}(k)$ where q is the q-th slot.

Taken into account the logical concepts presented in section 6.1, the TEG model that represents the dynamic of the system must follow the Max-Plus Algebra presented in Algorithm 1. The input parameters are the number of trucks (tokens) and an established time horizon. As it can see, the evaluation of $Y_i(k)$ occurs in terms of the random variables Z_i for all i. This

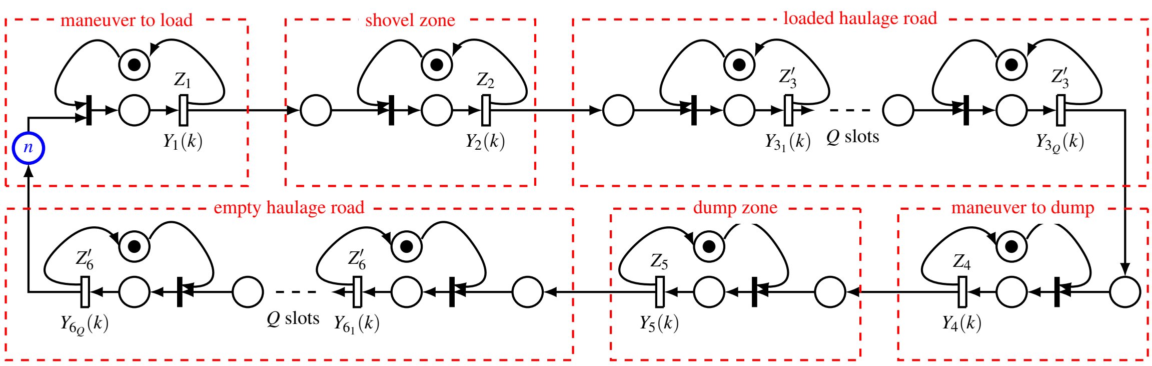


Figure 6.5: Max-Plus DES Model without operational stops

procedure is interrupted when the simulation reaches $T_{horizon}$. At the end, the algorithm returns an array **Y** that contains all $Y_i(k)$.

Algorithm 1: Computing T_{cycle} using Max-Plus algebra

Input : $n, T_{horizon}$
Output : **Y**

$Y_1(1) = Z_1$
$Y_2(1) = Y_1(1) \otimes Z_2$
$Y_{3_1}(1) = Y_2(1) \otimes Z_{3_1}$
for $i = 2, 3 \ldots, k_3$ **do**
    $Y_{3_i}(1) = Y_{3_{i-1}}(1) \otimes Z_{3_i}$
end for
$Y_4(1) = Y_{3_{k_3}}(1) \otimes Z_4$
$Y_5(1) = Y_4(1) \otimes Z_5$
$Y_{6_1}(1) = Y_5(1) \otimes Z_{6_1}$
for $i = 2, 3 \ldots, k_6$ **do**
    $Y_{6_i}(1) = Y_{6_{i-1}}(1) \otimes Z_{6_i}$
end for
while $Y_6(k) \leq T_{horizon}$ **do**
    if $k \leq n$ **then**
        $Y_1(k) = Y_2(k-1)$
    end if
    else
        $Y_1(k) = Y_6(k-n) \otimes Z_1 \oplus Y_2(k-1)$
    end if
    $Y_2(k) = Y_1(k) \otimes Z_2 \oplus Y_3(k-1)$
    $Y_{3_1}(k) = Y_2(k) \otimes Z_{3_1} \oplus Y_{3_2}(k-1)$
    for $i = 2, 3 \ldots, k_3$ **do**
        $Y_{3_i}(k) = Y_{3_{i-1}}(k) \otimes Z_{3_i} \oplus Y_{3_{(i+1)}}(k-1)$
    end for
    $Y_4(k) = Y_{3_{k_3}}(k) \otimes Z_4 \oplus Y_5(k-1)$
    $Y_5(k) = Y_4(k) \otimes Z_5 \oplus Y_6(k-1)$
    $Y_{6_1}(k) = Y_5(k) \otimes Z_{6_1} \oplus Y_{6_2}(k-1)$
    for $i = 2, 3 \ldots, k_6$ **do**
        $Y_{6_i}(k) = Y_{6_{i-1}}(k) \otimes Z_{6_i} \oplus Y_{6_{(i+1)}}(k-1)$
    end for
    $k = k + 1$
end while

Since that $Y_6(.)$ denotes the end of the cycle, the measure of interest T'_{cycle_n} can be easily computed from Equation. (6.5).

$$T_{cycleMP_n} \approx \frac{\sum_{i=n}^{k}(Y_{6_Q}(i) - Y_{6_Q}(i-n))}{n(k-n)} \tag{6.5}$$

Many others performance measures can be obtained from $\mathbf{Y}$. One of this measures is the depart rate of a process. We are interested in computing the rate that trucks let the 'dump zone' process. As it can be seen in the full system depicted in Figure 2.2, the stop decisions occur when a truck let such process. Then, this performance measure is used for computing $P(X = n)$.

Let $\lambda(n)$ be the depart rate of 'dump zone' process when the load haulage cycle without operational stops is running with n trucks. This measure can be estimated from Equation (6.6):

$$\lambda(n) = \mu_5 \rho(n) \tag{6.6}$$

where μ_5 and $\rho(n)$ denotes the rate services and the server utilization of the 'dump zone' process, respectively. Such equation is demonstrated by Bolch et al (2006).

Once that Z_5 is the random variable which describes the 'dump zone' process, we assume that $\mu_5 = 1/E[Z_5]$, where $E[Z_5]$ is the expected value, and $\{z_{5_1}, \cdots, z_{5_k}, z_{5_{k+1}}, \ldots\}$ the clock sequence of Z_5. Regarding $\rho(n)$, which is the fraction of time that the mentioned process is busy, we can compute this variable using $Y_5(.)$ and the mentioned clock sequence.

Since that $Y_5(k+1)$ and $Y_5(k)$ are the instant of $(k+1)$-*th* and k-*th* token exit the 'dump zone', respectively, the time that this process was empty is $\max(Y_5(k+1) - Y_5(k) - z_{5_k}, 0)$. For instance, suppose that $(k+1)$-*th* arrival occurs when the server is executing k-*th*. Then, after a time period of z_{5_k}, the server will immediately start to execute the next token and the empty time between $(k+1)$-*th* and k-*th* arrivals is zero. However, whether the server is executing the k-*th* token and ends before the $(k+1)$-*th* arrival, the server will be empty for time period which is the difference $(Y_5(k+1) - Y_5(k) - z_{5_k})$. Taken it into account, $\rho(n)$ can be obtained by Equation (6.7).

$$\rho(n) = 1 - \frac{\sum_{k=1}^{K-1} \max(Y_5(k+1) - Y_5(k) - z_{5_k}, 0)}{Y_6(K)} \tag{6.7}$$

In Eq (6.7), K denotes the last token and the summation represents the total empty time. From Algorithm 1, it can be seen that the simulation stops when $Y_6(.)$ reaches a well-known time horizon. Therefore, the fraction of the time that the 'dump zone' process is empty can be estimated and, consequently, $\rho(n)$ and $\lambda(n)$.

6.2.2 Computing $P(X = n)$ for $1 \leq n \leq N$

Regarding the computation of variable $P(X = n)$, the concepts presented in Section 3.2 are used. The main propose consists of understanding the operational stop processes from Markovian properties. As already stated, a load haulage cycle can be composed of one or more operational stops. These processes are queues described by general distributions and associated with probabilities of occurrence. Assuming that the service time of each stopping follows an exponential distribution and its probability of occurrence is constant, the dynamic of the stops can be represented a Markov Chain.

Let Z_{sj} be the random variable which describes the service time of the operational stop j and p_{sj} be its probability of occurrence. Assuming that Z_{sj} follows an exponential distribution with expected value $E[Z_{sj}]$ for all j, the service rate of j is $\mu_{sj} = 1/E[Z_{sj}]$.

To demonstrate how the number of stopped trucks can be mapped by Markov Chain, let's we first consider a load haulage cycle with just one operational stop. Figure 6.6 depicts a graphical representation of this chain.

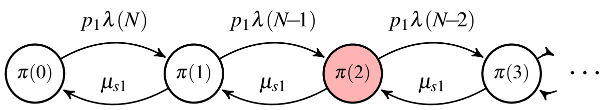


Figure 6.6: Markov chain of one operational stops

Since that n denotes the number of trucks running and N the fleet' size, the number of stopped trucks is $N - n$. Moreover, the variable π_{N-n} denotes the stationary probability of the state $N - n$. For instance, the highlighted node in Figure 6.6 indicates the state in which 2 trucks are stopped.

The measure $\lambda(n)$ for $1 \leq n \leq N$ can be computed applying Equation (6.6). Then, the stationary probability of all feasible state $(\{\pi_1, \cdots, \pi_N\})$ can be computed from the Chapman-Kolmogorov equation. Once that n is the number of trucks running, $P(X = n) = \pi_{N-n}$ for a system with just one operational stop.

For systems with more than one operational stop, the stationary probabilities can also be computed by the Chapman-Kolmogorov equation. However, the Markov Chain must be depicted by the parallel composition of the queues which represent the stop processes.

The stationary probability must be represented by the M-tuple $\pi(y_{s1}, \cdots, y_{sj}, \cdots, y_{sM})$, where y_{sj} denotes the number of trucks in the operational stop j. In this case, $P(X = n)$ must be estimated by the summation of the stationary probability of states that comply with: $n = N - y_{s1} - y_{s2} - \cdots - y_{sM}$. For instance, Figure 6.7 shows a Markov Chain that maps the number of stopped trucks in a load haulage cycle with two operational stops. The highlighted nodes in this figure denote states with 2 stopped trucks. Then, $P(X = N - 2)$ can be computed from the summation of their stationary probabilities.

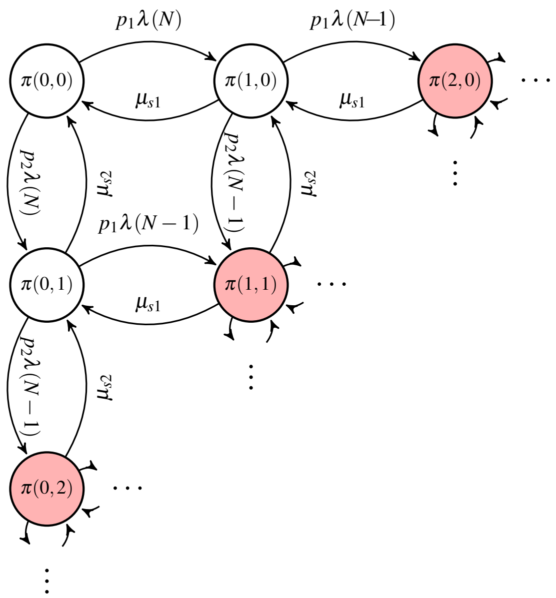


Figure 6.7: Markov chain of two operational stops

Let **S** be the set of feasible states of a Markov Chain which represents M operational stops. Then, the computation of $P(X = n)$ for system with M operational stops can be obtained from Equation (6.8).

$$P(X = n) = \sum_{\substack{\pi(.)\in \mathbf{S} \\ N-n=\sum y_{sj}}} \pi(y_{s1}, \cdots, y_{sM}) \tag{6.8}$$

The drawback for computing $P(X = n)$ is that the number of states in a Markov chain of a system with high value for N or M can become very large. However, using a previous knowledge of the problem, it can be possible to limit the number of stop trucks. For instance, for a system with N trucks, the probability of a situation with more then one-third of the fleet stopped is practically 0. Therefore, it possible to limit drastically the number of states. Another possibility is applying efficient numerical solution methods for large sparse linear equation systems, such presented in Mehmood and Crowcroft (2005).

6.3 Second batch of the computational experiment

The second batch of the computational experiment was conducted following the same analysis of Section 5.4. Therefore, the scenarios presented in Table A.1 were run by the hybrid method and by the SIMAN model. The goal is to provide evidence of equivalence between them. It was considered the Caterpillar 793F trucks with the nominal payload capacity of 226.8 tons (approx.). Each scenario has been run 6 times for a time period of interest of 365 days 24/7. Table 6.1 shows the $E[Prod(\theta)]$ obtained by the two methods. Moreover, the time taken to run each scenario is presented as well.

Table 6.1: Estimated production: Hybrid x SIMAN

ℓ	$E[Prod(\theta)]$ $\times 10^5$ tons		Exec. time (min)	
	Hybrid	SIMAN	Hybrid	SIMAN
1	32.06	32.01	1.88	8.711
2	38.68	38.74	2.90	7.863
3	36.09	36.78	7.41	18.372
4	49.49	49.60	8.35	20.908
5	39.44	39.69	2.85	7.349
6	41.94	42.53	2.92	7.767
7	38.52	38.04	11.22	21.688
8	40.97	40.46	3.55	8.804
9	33.83	34.07	2.41	5.830
10	39.30	38.87	3.02	7.102
11	42.13	42.62	2.78	6.922
12	40.06	40.18	2.92	7.692
13	32.96	33.03	2.88	6.632
14	41.94	41.87	7.61	17.035
15	37.18	37.66	4.73	10.991
16	46.43	46.45	2.66	8.287
17	19.81	19.79	0.62	5.289
18	16.05	16.18	0.93	5.843
19	38.14	37.80	2.29	6.565
20	42.78	42.96	10.46	25.564
21	29.61	30.13	2.00	8.261
22	38.78	38.01	2.42	7.279
23	59.44	59.09	4.87	16.459
24	41.46	41.38	2.30	7.428
25	43.88	43.73	2.42	8.015
26	40.69	40.68	4.64	14.436
27	33.52	33.30	0.77	6.926
28	42.81	42.72	2.07	6.176
29	39.43	39.56	2.33	6.987
30	15.08	15.09	1.06	6.372

The experiment analysis was also conducted applying the equivalence test based on

TOST (Two One-Sided Test) presented by Walker and Nowacki (2011). The equivalence between the hybrid and the SIMAN method can be established at the 2.5% significance level if a 95%-confidence interval for the fraction between the two methods is contained within the interval from $1-\delta^*$ to $1+\delta^*$. It is accepted as a tolerance deviation δ^* a value no greater than 2% of the expected iron production index.

Let $E[Prod(\theta)_h]_\ell$ and $E[Prod(\theta)_s]_\ell$ be the obtained expected *iron production index* applying the hybrid and the SIMAN method on the scenario ℓ. And let, $\bar{\omega}_3$ be the mean value of the fractions described by Equation (6.9)

$$\bar{\omega}_3 = \frac{E[Prod(\theta)_{hybrid}]_\ell}{E[Prod(\theta)_{Siman}]_\ell} \quad 1 \leq \ell \leq 30 \tag{6.9}$$

Therefore, the TOST which compare the Hybrid method and the SIMAN model can be expressed by the hypotheses:

$$\begin{cases} H_{0_3}^{+}: & \bar{\omega}_3 \geq 1+\delta^* \\ H_{a_3}^{+}: & \bar{\omega}_3 < 1+\delta^* \end{cases} \qquad \begin{cases} H_{0_3}^{-}: & \bar{\omega}_3 \leq 1-\delta^* \\ H_{a_3}^{-}: & \bar{\omega}_3 > 1-\delta^* \end{cases}$$

Regarding these test, the statistical software R (R Core Team, 2013) was used. Table 6.2 summarizes the results obtained.

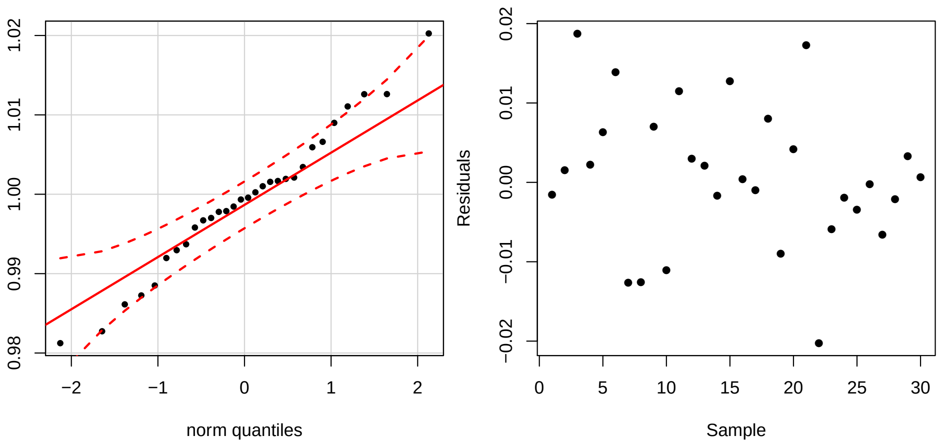
Table 6.2: TOST - Hybrid x SIMAN

TOST Par.	$\delta^* = 2\%$ Sig. level of 2.5% 95%-confidence interval
Hybrid x SIMAN	
$H_{0_3}^{+}$	$t = -12.608$ $DF = 29$ P-value=1.35×10^{-13} Null hypothesis $H_{0_1}^{+}$ can be rejected
$H_{0_3}^{-}$	$t = 11.68$ $DF = 29$ P-value=8.71×10^{-13} Null hypothesis $H_{0_1}^{-}$ can be rejected

The normality residual assumption for the test was validated from the Shapiro-Wilk method. In addition, the independence residual assumption was checked by the Durbin-Watson. Figures 6.8(a) and 6.8(b) show the normality and the independence of the residuals of $\bar{\omega}_3$.

Applying the TOST test, it was observed that the p-values for both hypotheses are less than the significance level 2.5%. Therefore, the null hypotheses $H_{0_3}^{+}$ and $H_{0_3}^{-}$ were rejected. As a result, the equivalence between the Hybrid and the SIMAN method can be declared, within the equivalence margin of $\delta^* = 2\%$, with a significance level of 2.5%. This result can be seen in Figure 6.9.

In conclusion, the aggregation of Max-Plus Algebra and Markov Chain consist of an efficient alternative to evaluate the expected value for the iron production index, $E[Prod(\theta)]$.



(a) Normality of the residuals: W = 0.98235 and p-value = 0.884

(b) Independence of the residuals: DW = 2.3231 and p-value = 0.7583

Figure 6.8: Assumptions of the hypothesis test - Hybrid method and SIMAN ($\bar{\omega}_3$)

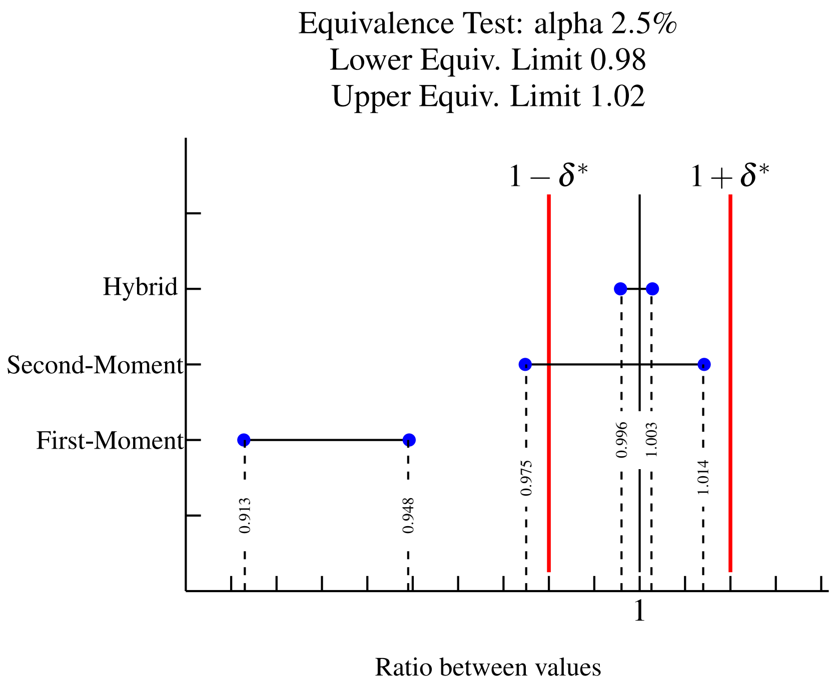


Figure 6.9: Equivalence testing 2.5%-significance level and 95%-confidence interval

As could be seen, there is an equivalence between the system modeled by the hybrid method presented and by a standard simulation tool. Moreover, this new method is almost 3 times faster. From optimization point view, it is very important because, using the hybrid model, it is possible to explore the space of feasible solutions more widely. However, the hybrid model is not so fast when compared with the analytical approximation methods.

CHAPTER 7: SIMULATION OPTIMIZATION STRATEGY

"During the 1950s, I decided, as did many others, that many practical problems were beyond analytic solution and that simulation techniques were required."
Harry Markowitz

The quote above (Lindbeck, 2001) is very plausible and is one affirmation that many engineers usually accept. However, the previous chapters showed a connection between DES simulation and analytical approximation solutions. This thesis addresses discrete optimization via simulation. As mentioned, simulation requires a substantial computational effort. Bolch et al (2006) point out that the main disadvantage of DES simulation is the time taken to run the models for large, realistic systems particularly when results with high accuracy are desired. Taken this into account, we observed that many optimization simulation practical problems are beyond simulation solution and that analytical techniques are required.

Regarding the difficulties in dealing with discrete optimization via simulation problems, this Chapter presents a computational method for project portfolio selection in open-pit mining. Such problem is classified as discrete optimization via simulation because discrete optimization is applied in a stochastic environment, in which, each function evaluation is obtained through a DES stochastic model. Essentially, the optimization method explores peculiarities of the problem and limits drastically the number of evaluations. Before we describe such optimization method, we presented some concepts about project portfolio problems.

7.1 Project portfolio selection

There are many factors that influence the project portfolio selection. Thus, such task is considered by many organizations a critical process. These factors include the uncertain nature and the interrelationship between projects. The first factor addresses the success or failure of the selected portfolio. In other words, assuming risk aversion, it is necessary to consider the chances of expected return not be reached (Santos et al, 2014). The second factor addresses the possible interrelationship between projects. The simplest case consists of determining whether a project will be applied or not. However, for huge problems, it is necessary to consider how a project can influence the others. According to Costa et al (2010), some studies in the literature points out that the interrelationship between projects constitutes the most critical of all factors involved in the project portfolio selection.

Into the project portfolio management context there is no simple way for selecting the projects to compose the portfolio, especially when we must consider the risk associated with each portfolio. Thus, a coherent and powerful set of tools that can be used to help it assess the challenges and risks is required, enabling the analysis of diverse perspectives and guiding the selection of projects that are complementary and should form the portfolio logically (Pennypacker and Dye, 2002).

Due to the challenge of portfolio management problem, many methods have been developed. Souza et al (2012) summarize some of these methods and present a practical model. The authors explore a traditional method of investment analysis and add qualitative attributes to the analysis. In this method, the authors use linear programming and multi-criteria analysis for project portfolio selection. Taken into account the attributes economic values, potential environmental and social impacts, the method was validated applying in a projects portfolio problem.

Cadenas et al (2012) propose a *fuzzy* model for project portfolio selection problems. Such model considers the uncertainties of the investor's preferences and the risk assumed. Moreover, this model considers an exact methodology for smaller problems and a hybrid heuristic for medium and large problems. The author's proposal is to provide an efficient portfolio according to Markowitz theory (Markowitz, 1952), showing a trade-off between risk and expected return. Huang (2012) presents a new type of variable to reflect subjective estimates of the expected return. An index of risk is assigned to each portfolio without historical data. In the mentioned work, the risk is defined based on expert estimates. Also based on Markowitz theory, Costa et al (2010) presents a proposal to support the prioritization of software portfolio of projects.

A methodology for risk management for portfolio analysis of purchase and energy sales is presented in Santos et al (2014). Combining multiobjective optimization techniques and concepts related to risk in the energy sector, the authors propose a metric for measuring the risk of loss in a generating energy company. An adaptation of the ellipsoid method for deterministic multi-objective optimization, where the objectives are maximizing the return and minimizing the risk was taken.

Ranking the best projects can be arduous for large problems since it is necessary to measure all the relationships between assets. Souza et al (2012) explains that when it comes to portfolio definition, few studies that consider the search for an optimal portfolio has been applied in practice and discussed in theory. Other points that contribute to the complexity of the problem are described by Costa et al (2010). One of this points is the fact that the evaluated of the expected return of several variables can turn into an intractable problem. Moreover, the problem of ranking projects in a portfolio has exponential complexity, a common characteristic of combinatorial problems.

7.2 Project portfolio definition

Let $E[Port(\boldsymbol{\theta_s})]$ be the expected value of a portfolio. According to the Markowitz (1952), $E[Port(\boldsymbol{\theta_s})]$ can be computed by Equation (7.1),

$$E\left[Port(\boldsymbol{\theta_s})\right] = \sum_{i=1}^{n_p} \theta_i E\left[J(\theta_i)\right], \tag{7.1}$$

where $\boldsymbol{\theta_s}$ is the portfolio selected, n_p is the number of candidate projects (size of $\boldsymbol{\theta}$) and $E\left[J(\theta_i)\right]$ the expected value of the project θ_i. For a project portfolio problem, the decision consists of applying or not a project. Therefore, there is no weighting. It means that each term θ_i is a binary decision variable.

In a project portfolio definition, $E\left[Port(\boldsymbol{\theta_s})\right]$ represents the sum of the expected returns of each project selected. Since that $\boldsymbol{\theta}$ is the project portfolio, $\boldsymbol{\theta_s} \in \boldsymbol{\theta}$. For a project portfolio with interrelationships between projects, it is necessary to include terms in the formulation which represent how intense are such interrelationships. When only pairs of projects are considered in the project portfolio problem, we have a quadratic formulation to quantify the expected return. For more complex functions, composed of interrelationships between all n_p projects, the formulation corresponds to a n_p-*th* degree polynomial.

Equation (7.2) represents a quadratic function for $E\left[Port(\boldsymbol{\theta_s})\right]$. In this case, the parameter $E\left[\pm J(\varphi_{ij})\right]$ is a measure of intensity and direction of the relationship between projects i and j. The same equation can be extended to higher degree polynomials.

$$E\left[Port(\boldsymbol{\theta_s})\right] = \sum_{i=1}^{n_p} \theta_i E\left[J(\theta_i)\right] + \sum_{j=1}^{n_p} \sum_{i=1}^{n_p} \theta_j \theta_i E\left[\pm J(\varphi_{ij})\right] \tag{7.2}$$

Since that the project portfolio selection is associated with a budget constraint, the project portfolio problem can be classified as a binary backpack problem (Knapsack Problem-KP), which is a classic optimization problem.

7.3 Seeing the problem as a binary knapsack problem

The knapsack problem (KP) is a classic programming model which is applied in many segments. One of this application is the study of capital investment. Figuratively, we can describe this problem as filling a backpack without exceeding a volume limit. The decision consists of placing in the backpack products that maximize (maximization problem) a specified value, respecting its capacity. In this thesis, the products are the projects at the portfolio and the capacity is the available budget. Equation (7.3) is a maximization formulation which represents a binary knapsack problem.

$$\begin{aligned} \underset{\theta_i \in \boldsymbol{\theta}}{\text{Maximize}} \quad & \sum_{i=1}^{n_p} E\left[J(\theta_i)\right] \theta_i \\ \text{subject to} \quad & \sum_{i=1}^{n_p} w_i \theta_i \leq R \\ & \theta_i \in \{0,1\} \end{aligned} \tag{7.3}$$

The term R symbolizes the knapsack capacity and w_i denotes weight associated with the variable θ_i. For this thesis, R is the budget available and w_i is the cost of the project θ_i. The goal is to maximize the summation of the expected values. The first constraint limits the sum of cost to R. The second indicates that the variable $\theta_i \in \boldsymbol{\theta}$ are binary. This kind of problem is known in the art state as a *Knapsack Problem* $0-1$ (KP $0-1$).

For a quadratic knapsack problem $0-1$ (QKP $0-1$), we must consider the interrelationships between two products. Thereby, the objective function becomes quadratic, where a value can be added or subtracted according to this interrelationship. Equation (7.4) represents a QKP $0-1$.

$$\begin{aligned} \underset{\theta_i, \theta_j \in \boldsymbol{\theta}}{\text{Maximize}} \quad & \sum_{i=1}^{n_p} E\left[J(\theta_i)\right] \theta_i + \sum_{i=1}^{n_p} \sum_{j=1}^{n_p} E\left[\pm J(\varphi_{ij})\right] \theta_i \theta_j \\ \text{subject to} \quad & \sum_{i=1}^{n_p} w_i \theta_i \leq R \\ & \theta_i \in \{0,1\} \end{aligned} \tag{7.4}$$

As we can see, the quadratic formulation considers the measure of intensity, direction and relationship $E\left[\pm J(\varphi_{ij})\right]$, which indicates the value to be added or subtracted if projects i and j are included in the same solution. Some other classifications of backpack problem are described by Goldbarg and Luna (2005). In project portfolio context, Ding and Cao (2008) extend the classic the backpack problem and build mathematical models for different optimization kinds of portfolios.

This thesis addresses a project portfolio which we must consider the interrelationships between all the projects. Thus, the problem is binary knapsack problem with an objective function of order n_p. Equation (7.5) presents a generic formulation for a n_p knapsack problem

(n_p-KP).

$$\begin{aligned}
\underset{\theta_1,\theta_2,\ldots,\theta_{n_p}\in\boldsymbol{\theta}}{\text{Maximize}} \quad & \sum_{i_1=1}^{n_p} E\left[J(\theta_{i_1})\right]\theta_{i_1} + \sum_{i_1=1}^{n_p}\sum_{i_2=1}^{n_p} E\left[\pm J(\varphi_{i_1 i_2})\right]\theta_{i_1}\theta_{i_2} + \\
& \sum_{i_1=1}^{n_p}\sum_{i_2=1}^{n_p}\sum_{i_3=1}^{n_p} E\left[\pm J(\varphi_{i_1 i_2 i_3})\right]\theta_{i_1}\theta_{i_2}\theta_{i_3} + \ldots + \\
& \sum_{i_1=1}^{n_p}\sum_{i_2=1}^{n_p}\cdots\sum_{i_{n_p-1}=1}^{n_p}\sum_{i_{n_p}=1}^{n_p} E\left[\pm J(\varphi_{i_1\ldots i_{n_p}})\right]\theta_1\theta_2\ldots\theta_{n_p} \\
\text{subject to} \quad & \sum_{i=1}^{n_p} w_i\theta_i \leq R \\
& \theta_i \in \{0,1\}
\end{aligned} \tag{7.5}$$

Where the terms $\varphi_{i_1 i_2}$ and $\varphi_{i_1 i_2 i_3}$ denote the interrelationship between two and three projects, respectively. The objective function must be extended until the summation which denotes the interrelationship between all the projects. Then, the term $\varphi_{(\cdot)}$ must be extended, as well. Hence, $\varphi_{i_1\ldots i_{n_p}}$ represents the interrelationship between all n_p projects.

A nonlinear Knapsack problem can be rewritten as a linear Knapsack problem. The linearization of a QKP was first introduced by Adams and Sherali (1986). In general, a linearization consists of adding new variables which represent the quadratic terms. Moreover, new constraints must be added in the formulation. There are many techniques of linearization address quadratic problems. One example is presented by Oral and Kettani (1992). Regarding this technique, the authors applying linearization of a binary-quadratic formulation with n variables. The quadratic terms have seen new variables. In addition, n constraints are added to the formulation.

A Knapsack problem is a combinatorial optimization problem. Then it must be classified due to its complexity. According to Papadimitriou and Steiglitz (1998), P is a class of problems which polynomial time solution. In other words, they are relatively simple problems from the computational point of view, in which, there are efficient algorithms for solving. Some classes of combinatorial problems with polynomial solutions such as *Graph connectedness*, *Path in a digraph* and *Minimum Spanning Tree* are detailed by Papadimitriou and Steiglitz (1998). On the other hand, there are problems without a solution in polynomial time. Such problems belong to a class named *NP-hard*. According to Pisinger (1995), it is weakly probable that some polynomial algorithm is created for solving *NP-hard* problems.

This thesis concerns a non-linear Knapsack problem with interrelationships between the variables, such shown by Equation (7.5). Since that even linear KP$(0-1)$ are featured as *NP-hard*, the project portfolio problem that we are dealing is a *NP-hard* problem as well.

Although this complexity, Pisinger (1995) points out that the global solution of many Knapsack problems of high dimension can be found in a fraction of seconds. This author details some exact techniques for this class of problem. Nowadays, there are some computational tools which apply these techniques efficiently. One of this tools is the software Gurobi Optimization (2016).

The following section shows a new methodology which concerns the project portfolio selection discussed in this thesis. Essentially, such section proposes a new linear formulation that considers the interrelationship between projects. As we can see in the Equation (7.4), we are dealing with a complex combinatorial problem. Then, it can be hard, even impossible, to measure all interrelationship. Taken it into account, instead of trying all the possible projects combinations, the proposed strategy searches for identifying the set of projects which produces good feasible solutions based on performance measures from the designed DES model.

7.4 New method for project portfolio selection

This work was motivated by a study that concerns with project portfolio selection in an open-pit mine environment. Regarding this problem, each project aims to change the service time of one specific process. For instance, a project that improves the roads will reduce the dislocation times among the squares of the load-haulage cycle. The goal is to select a set of projects which maximizes the expected iron production for a known time horizon. However, the impact of a project application in this indicator must be evaluated from some stochastic DES model.

Let $E[Prod(0)]$ be the expected iron production index of the scenario without projects application and, let $E[Prod(\theta_i)]$ be the expected iron production index of a scenario composed only by project θ_i. Thus, the increase of iron production $E[G(\theta_i)]$ provided by project θ_i is obtained by Equation (7.6).

$$E[G(\theta_i)] = E[Prod(\theta_i)] - E[Prod(0)] \ \ \forall \, i. \tag{7.6}$$

At this point, we must remember that $\boldsymbol{\theta}$ is the set of all the projects θ_i $(1 \leq i \leq n_p)$ and w_i is the cost of project θ_i $(\theta_i \in \boldsymbol{\theta})$. The goal consists of maximizes the summation of the gains $E[G(\theta_i)]$ respecting an established budget. It is a nonlinear binary problem with possible interrelationship among all projects. In this method, the project portfolio is formulated as binary knapsack problem where the nonlinear term, which denotes the interrelationships between projects, is converted into new projects. Essentially, each decision variable consists of an individual project or a combination of projects. Hence, the cost w_i must be understood, for the combination of projects, as the summation of the individual costs. To exemplify, Figure 7.1 illustrates project portfolio composed by three projects, θ_1, θ_2 and θ_3. The intersection areas denote the interrelationships between them. That are φ_{12}, φ_{13}, φ_{23} and φ_{123}.

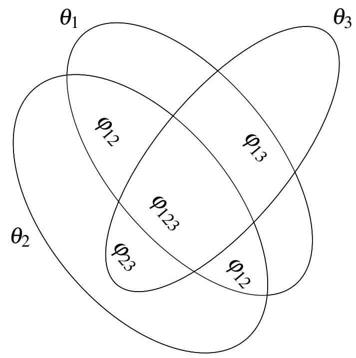
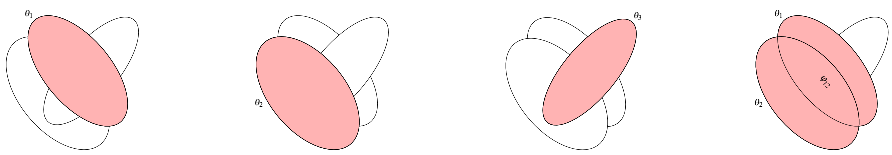
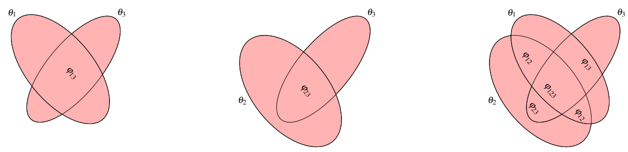


Figure 7.1: Interrelationship in a Project portfolio composed by θ_1, θ_2 and θ_3.

Due to the open-pit mine features, there are interrelationships among projects. It means that a combination of two or more projects must result in different gain compared with the linear solution. As mentioned, this method addresses a linearization strategy which considers the combination of projects as new decision variables. Figure 7.2 shows project portfolio of Figure 7.1 such a way that each possible combination is a project as well. For this example, the result is a linear optimization problem with 7 decision variables. Figures 7.2(a), 7.2(b) and 7.2(c) illustrate the individual projects, θ_1, θ_2 and θ_3, respectively. Figure 7.2(d), 7.2(e) and 7.2(f) show the quadratic combinations between the projects, while Figure 7.2(g) a combination of all the projects.



(a) θ_1 (b) θ_2 (c) θ_3 (d) θ_4-Combination



(e) θ_5-Combination (f) θ_6-Combination (g) θ_7-Combination

Figure 7.2: Seeing the project portfolio of Figure 7.1 such a way that each possible combination is also a project.

The interrelationship between projects constitutes one of the most critical of all factors involved in the project portfolio problem. The problem of ranking projects in a portfolio has exponential complexity, a common characteristic of combinatorial problems. For the optimization point view, the number of feasible combinations in the worse case is 2^{n_p}. Taken this into account, for huge problems, it is necessary to select the combinations which deserve be converted into new decision variables. Thus, the method presented in this section consists of a column generation strategy, in which the columns are generated heuristically based on the features of the problem. In addition, the set of constraint is updated to avoid that projects appear redundantly in the same solution. The final result is a linear KP $(0-1)$ of size $n_v \geq n_p$, where n_v is the number of decision variables.

Let $\mathbf{\Theta}$ be the set of decision variable which represents the projects. The term $E\left[G(\theta_i)\right]$ is the expected gain of project θ_i, such shown by Equation (7.6). It is important to point out that θ_i can be a individually project or a combination of projects, in which, $[Prod(\theta_i)]$ must be estimated by some stochastic DES strategy. Equation (7.7) denotes the new general formulation for the problem.

$$
\begin{array}{lllllll}
\underset{\theta_i \in \mathbf{\Theta}}{\text{Maximize}} & E\left[G(\theta_1)\right]\theta_1 + & E\left[G(\theta_2)\right]\theta_2 + & E\left[G(\theta_3)\right]\theta_3 + & \cdots & + & E\left[G(\theta_{n_v})\right]\theta_{n_v} \\
\text{subject to} & w_1\theta_1 + & w_2\theta_2 + & w_3\theta_3 + & \cdots & + & w_{n_v}\theta_{n_v} \leq R \\
 & a_{11}\theta_1 + & a_{12}\theta_2 + & a_{13}\theta_3 + & \cdots & + & a_{1n_v}\theta_{n_v} \leq 1 \\
 & a_{21}\theta_1 + & a_{22}\theta_2 + & a_{23}\theta_3 + & \cdots & + & a_{2n_v}\theta_{n_v} \leq 1 \\
 & \vdots & & & & & \\
 & a_{n_v1}\theta_1 + & a_{n_v2}\theta_2 + & a_{n_v3}\theta_3 + & \cdots & + & a_{n_vn_v}\theta_{n_v} \leq 1 \\
 & \theta_i \in \{0,1\} & & & & &
\end{array}
\tag{7.7}
$$

As we can see, the goal consists of maximizing the some of the gains provided by the projects. The first constraint indicates that the summation of the costs cannot exceed the available budget, R. Since that a variable θ_i can be a combination of projects, the dashed constraints consist of avoiding that projects appear redundantly in the same solution. The term a_{ij} $(1 \leq i \leq n_v$ and $1 \leq j \leq n_v)$ is a binary value which must be defined due to the column generation rules. The last constraint denotes that the decision variables $\theta_i \in \mathbf{\Theta}$ must be binary. Regarding the terms a_{ij}, let $\mathbf{A}$ be a matrix which denotes all values of a_{ij}. Then, $\mathbf{A}$ must be created due to the following rules:

1. To compute using $E\left[G(\theta_i)\right]$ for all individually projects. It means from 1 to n_p. Then, initially, $n_v = n_p$.

2. To set the individual costs w_i for i from 1 to n_p.

3. To create a 2-dimension identity matrix of size n_p:

$$\mathbf{A} = \underbrace{\begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & & & \ddots & \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix}}_{n_p} \left.\vphantom{\begin{pmatrix} 1\\0\\0\\\vdots\\0 \end{pmatrix}}\right\} n_p$$

4. To check the most relevant combination between the projects (Next section explains a sensibility analysis to identify the most relevant combination).

5. To create a new project which represents the most relevant combination identified at the previous step.

6. To update n_v, $n_v = n_v + 1$. Now we have a new project.

7. To define the cost of this new project (Summation of the costs) and to perform the $E\left[G(\theta_{n_v+1})\right]$.

8. To add a new column in the knapsack formulation. That is, $E\left[G(\theta_{n_v})\right]\theta_{n_v}$, w_{n_v} and $\mathbf{A}$:

$$\mathbf{A} = \underbrace{\begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & a_{1n_v} \\ 0 & 1 & 0 & \cdots & 0 & a_{2n_v} \\ 0 & 0 & 1 & \cdots & 0 & a_{3n_v} \\ \vdots & & & \ddots & & \\ 0 & 0 & 0 & \cdots & 1 & a_{n_pn_v} \\ a_{n_v1} & a_{n_v2} & a_{n_v3} & \cdots & a_{n_vn_v} & 0 \end{pmatrix}}_{n_v} \left.\vphantom{\begin{pmatrix} 1\\0\\0\\\vdots\\0\\a \end{pmatrix}}\right\} n_v$$

Since that this new column is a variable which represents a combination of problems, then each term a_{in_v} $(1 \leq i \leq n_v)$ is:

$$a_{in_v} = \begin{cases} 1, & \text{if the individual project } i \text{ is composing the new combination.} \\ 0, & \text{otherwise} \end{cases}$$

Moreover:

$$a_{n_vi} = \begin{cases} 1, & \text{if the project } i \text{ is one of the two projects that compose the new combination.} \\ 0, & \text{otherwise} \end{cases}$$

9. Go back step 4.

To exemplify, let's consider a project portfolio with three projects illustrated in Figure 7.2. Then, $n_p = 3$. Let's suppose that we know the interrelationships and the costs w_1, w_2, w_3 and w_4. The initial decision variables are θ_1, θ_2 and θ_3. Performing the expected gain of these projects used a stochastic DES method we can obtain $E[G(\theta_1)]$, $E[G(\theta_2)]$ and $E[G(\theta_3)]$. Then, the initial formulation for this example is described by:

$$
\begin{array}{lrrrl}
\underset{\theta_i \in \boldsymbol{\Theta}}{\text{Maximize}} & E[G(\theta_1)]\,\theta_1 + & E[G(\theta_2)]\,\theta_2 + & E[G(\theta_3)]\,\theta_3 & \\
\text{subject to} & w_1\theta_1 + & w_2\theta_2 + & w_3\theta_3 \leq & R \\
& \theta_1 + & 0 + & 0 \leq & 1 \\
& 0 + & \theta_2 + & 0 \leq & 1 \\
& 0 + & 0 + & \theta_3 \leq & 1 \\
& \theta_i \in \{0,1\} & & &
\end{array}
$$

Let's suppose that we found evidence of interrelationship between projects 1 and 2. Then, a new variable θ_4 which represents a combination (1 and 2) is created, the cost $w_4 = w_1 + w_2$ is computed and $E[G(\theta_4)]$ is evaluated by a stochastic DES method. Following the previous mentioned rules, a new column is added in the formulation and the set of constraints is updated. Then, the formulation is now described by:

$$
\begin{array}{lrrrrl}
\underset{\theta_i \in \boldsymbol{\Theta}}{\text{Maximize}} & E[G(\theta_1)]\,\theta_1 + & E[G(\theta_2)]\,\theta_2 + & E[G(\theta_3)]\,\theta_3 + & E[G(\theta_4)]\,\theta_4 & \\
\text{subject to} & w_1\theta_1 + & w_2\theta_2 + & w_3\theta_3 + & w_4\theta_4 \leq & R \\
& \theta_1 + & 0 + & 0 + & \theta_4 \leq & 1 \\
& 0 + & \theta_2 + & 0 + & \theta_4 \leq & 1 \\
& 0 + & 0 + & \theta_3 + & 0 \leq & 1 \\
& \theta_1 + & \theta_2 + & 0 + & 0 \leq & 1 \\
& \theta_i \in \{0,1\} & & & &
\end{array}
$$

Let's suppose now that we found evidence of interrelationship between projects 3 and 4. Then, a new variable θ_5 which represents a combination (3 and 4) is created, the cost $w_5 = w_3 + w_4$ is computed and $E[G(\theta_5)]$ is evaluated. Then, a new column is added in the formulation and the set of constraints is updated. The formulation is now described by:

$$
\begin{array}{llllll}
\underset{\theta_i \in \boldsymbol{\Theta}}{\text{Maximize}} & E[G(\theta_1)]\,\theta_1 + & E[G(\theta_2)]\,\theta_2 + & E[G(\theta_3)]\,\theta_3 + & E[G(\theta_4)]\,\theta_4 + & E[G(\theta_5)]\,\theta_5 \\
\text{subject to} & w_1\theta_1 + & w_2\theta_2 + & w_3\theta_3 + & w_4\theta_4 + & w_5\theta_5 \leq R \\
 & \theta_1 + & 0 + & 0 + & \theta_4 + & \theta_5 \leq 1 \\
 & 0 + & \theta_2 + & 0 + & \theta_4 + & \theta_5 \leq 1 \\
 & 0 + & 0 + & \theta_3 + & 0 + & \theta_5 \leq 1 \\
 & \theta_1 + & \theta_2 + & 0 + & 0 + & 0 \leq 1 \\
 & 0 + & 0 + & \theta_3 + & \theta_4 + & 0 \leq 1 \\
 & \theta_i \in \{0,1\} & & & &
\end{array}
$$

The method must evolve until a limit number of columns or until to find all feasible combinations. In the end, we will have a linear KP $(0-1)$. Since it may be unfeasible to evaluate all project combinations, a challenge is to find how intense are the interrelationships between the projects which compose the combinations. The projects aim to reduce the service time in a specific node. Since that the iron production index depends on the meantime taken in each node of the cycle, the measure *mean response time* of each node is helpful to define those who deserve to be evaluated. A disadvantage of analyzing the projects only individually is the possibility of increasing this performance measure of others nodes. Consequently, neither total cycle time nor total mine production index will be changed. Taken this into account, a sensibility analysis must be conducted to identify relevant combinations.

7.5 Sensibility analysis of interrelationships between projects

The philosophy behind this sensibility analysis is the fact that the *mean response time* indicates the plant bottlenecks, which must be fixed by new projects. Let's consider that two or more projects have a small expected gain. Using conventional optimization methods they would hardly be included in the final optimal portfolio. However, the combination of them can generate a good expected gain when evaluated jointly. Considering these circumstances, the analysis of the problem requires finding which combinations are relevant. For instance, a project that produces a time reduction in the dumping process and increases *mean response time* in the shovel site can produce a significant gain whether combined with another project that provides an improvement in the second mentioned node.

A reason for such behavior is the fact that, generally, each project impacts in a specific process of the system decreasing the value of the performance measure $\bar{t}_m$ of some process $(1 \leq m \leq M)$. As a consequence, they can increase the value of this measure in another process.

Let $\bar{t}_m(\theta_i)$ be the dependent *mean response time*. It means, the *mean response time* is subject to the application of project θ_i. Let $\bar{t}_m(0)$ be the same dependent measure without project application. Thus, the difference $\bar{t}_m(\theta_i) - \bar{t}_m(0)$ indicates how much project θ_i impact at the *mean response time* of process m. From such difference, we can empirically to estimate the interrelationship between projects. Let the term $\zeta_{ij}(m)$ be an auxiliary function which describes the influence of projects θ_i and θ_j at the process m. Equation (7.8) computes the value of each $\zeta_{ij}(m)$.

$$\zeta_{ij}(m) = \begin{cases} \bar{t}_m(\theta_i) - \bar{t}_m(\theta_j), & \text{if } \bar{t}_m(\theta_i) - \bar{t}_m(0) > 0 \text{ and } \bar{t}_m(\theta_j) - \bar{t}_m(0) < 0 \\ 0, & \text{otherwise} \end{cases} \tag{7.8}$$

The concepts behind Equation (7.8) is the fact that $\bar{t}_m(\theta_i) - \bar{t}_m(0)$ indicates if the application of project θ_i increases the *mean response time* in process m, while $\bar{t}_m(\theta_j) - \bar{t}_m(0)$ shows whether the application of project θ_j decreases such measure in the same process. Assuming that both assumptions are true, the value of $\zeta_{ij}(m)$ is the difference $\bar{t}_m(\theta_i) - \bar{t}_m(\theta_j)$. If some one of the assumptions is false, $\zeta_{ij}(m)$ is 0. Now, let ϕ_{ij} be the interrelationship between the projects θ_i and θ_j, for $(\theta_i, \theta_j) \in \Theta$, such presented by Equation (7.7). Thus, ϕ_{ij} is denoted by Equation (7.9).

$$\phi_{ij} = \sum_{m=1}^{M} \zeta_{ij}(m) + \sum_{m=1}^{M} \zeta_{ji}(m) \tag{7.9}$$

Whence, we can argue that ϕ_{ij} implies how much strong is the relationship between the projects θ_i and θ_j. As a result, for each iteration of the methodology presented in the previous section (step 4), we must compute ϕ_{ij} for all combination between projects to check the most relevant.

7.6 Numerical example

To exemplify the efficiency of this optimization strategy, we demonstrated the result obtained by Ribeiro et al (2016). In the mentioned paper, it was available for numerical tests a project portfolio with 15 projects limited to a budget R of 15pu (per unit). Applying a first-moment approximation method, we have as an example the scenario presented in Table 7.1.

As we can see, there are four operational stops in this model. The transient probabilities were computed using the occurrence period of each operational stop of the open-pit mining front. Therefore, the estimated values are: $p_1 = 0.0015$, $p_2 = 0.0127$, $p_3 = 0.0255$ and $p_4 =$

Table 7.1: First moment approximation

Process	Distribution	Expected service time	μ_m	Discipline
Maneuver to load	Triangular	2.266	$\frac{1}{2.266}$	FIFO
Shovel site	Tria (comp.)	3.227	$\frac{1}{3.227}$	FIFO
Loaded Haulage road	Inv. Gaussian	8.333	$\frac{1}{8.333}$	IS
Maneuver to Dump	Triangular.	1.100	$\frac{1}{1.100}$	FIFO
Dump site	Triangular	2.133	$\frac{1}{2.133}$	FIFO
Empty Haulage road	Inv. Gaussian	6.944	$\frac{1}{6.944}$	IS
Preventive maintenance	Triangular	150	$\frac{1}{150}$	FIFO
Corrective maintenance	Triangular	1120	$\frac{1}{1120}$	FIFO
Supply	Gaussian	150	$\frac{1}{150}$	FIFO
Shift Change	Gaussian	16	$\frac{1}{16}$	FIFO

0.1232. Table 7.2 presents the projects which impacts the load-haulage cycle. Moreover, such table shows the individual cost of projects and the process affected.

Table 7.2: Project portfolio

Proj	Description	Cost (pu)	Impact	Where?
θ_1	Bilateral charging	0.5	-15%	process 1
θ_2	Roads improvement	1.5	-20%	process 3,6
θ_3	Dead load reduction	1	5%	par. C
θ_4	Excavation	2	-18%	process 2
θ_5	1 Truck acquisition	5	$N+1$	par. N
θ_6	2 Truck acquisition	10	$N+2$	par. N
θ_7	Rolling A-Frame	1	-17%	process 8
θ_8	Forklif acquisition	0.7	-15%	process 8
θ_9	Preventive kits	0.65	-14%	process 8
θ_{10}	Supply improvement	2	-35%	process 9
θ_{11}	Dump site improvement	2	-25%	process 4
θ_{12}	Shift change improvement	1.5	-25%	process 10
θ_{13}	Backlog's reduction	5	-30%	process 7
θ_{14}	DMT reduction	2.5	-24%	process 5
θ_{15}	Load site improvement	3	-25%	process 1

This problem was formulated according to Equation (7.7) with a limit of 200 'relevant combinations'. The 'Gurobi Optimizer' software was used and the best project portfolio solution was found. Lastly, the optimal solution was converted in a set of isolate project (such as Table 7.2) and the expected gain of this solution was obtained using the queuing network model presented in this paper. As a result, the best portfolio was $\{1,0,0,0,0,1,1,1,1,0,1,0,0,0,0\}$, which provides a production increase of 1049.09 tonnes.

We also solved this example by exhaustion which all feasible solutions evaluated. We observed that the best solution found using the new strategy is the global optimal for this example. Moreover, the solution is almost two times better than the solution obtained ignoring the interrelationship between projects. We can argue that this new strategy is monotonically nondecreasing (for the maximization problem). The reason is the fact that the beginner formulation considering only the individual projects and, hence, iteratively, the mechanism adds new projects into the formulation following the intensity of the interrelationships. Therefore, we can argue that, if we compute by some DES stochastic method the optimal solutions expressed by Equation (7.7) at the $k+1$-*th* and k-*th* iterations , the result obtained at the $k+1$-*th* iteration is better or at least equal the result acquired at the k-*th* iteration.

As we can see, the quality of the solution can depend on the number of iterations once that in each iteration a column which represents a new project (decision variable) is added. Since that we must compute the expected gain of each project by a stochastic DES method and the fact that the computational time is limited, the number of new columns depends on the time taken to run each expected value. Taken this into account, we understand that analytical approximation methods, especially the first-moment approximation, are very attractive and competitive way to be applied in this optimization problem. Although we did not find evidence of equivalences between the stochastic DES methods, the analytical and the simulation methods follow the same ordinal pattern. Hence, analytical approximation methods can be used to evaluate the expected gain of the projects and then to rank the portfolio projects.

CHAPTER 8: CONCLUSIONS AND FUTURE WORKS

"When everything is easy one quickly gets stupid."
Maxim Gorky

This Thesis concerned a nonlinear project portfolio optimization where we must compute the effect of each project combination through a stochastic DES method. The model represents an open-pit mine system appropriate for estimating the iron production index. Then, the study was conducted comparing different tools address stochastic DES systems. We designed a load-haulage cycle of an open-pit mine though DES simulation and analytical approximation methods. In effect, we would like pointing out the drawbacks and advantages of each model and, hence, suggesting the more appropriate method for the mentioned optimization problem.

Regarding the simulation tools, a standard way to simulate stochastic DES is through of SIMAN platform. The main advantage of SIMAN modeling is the fact that it supports all essential steps of a simulation study and it is easy to model. The drawback is the fact that this modeling method needs an expensive computational effort. Therefore, we can state the application of SIMAN modeling in optimization context is not appropriated. About analytical approximations, these modeling methods were extremely efficient. Mainly if we compare with the time taken to run simulation models. This matter is quite remarkable for the optimization context since that the computational time is a crucial aspect and, for the project portfolio problem presented, such time limits the number of interrelationships evaluated. In summary, we recall that stochastic DES simulation and analytical models are approaches to deal with a complex stochastic problem. Both approaches concern to approximation.

However, for our study, the results showed that analytical approximation models are attractive and competitive ways for drastically reducing the computation time. Whence, it can be used instead of simulation models. Although we cannot declare the equivalence between the models, the perform results indicate that analytical approximation methods are appealing alternatives to evaluate the expected gain of the projects once that analytical and simulation methods follow the same ordinal pattern, with small errors. Since the motivation is a project portfolio problem, we understand that the analytical approximation methods can be applied to rank the portfolio projects.

From the optimization point of view, we strongly appreciate the application of analytical approximation once that the time taken to run each scenario is considerably faster than the computational time spent using simulation methods. It means that the optimization strategy can explore the feasible space of solutions more broadly. We can see that the first-moment approximation method is around ten thousand times faster than the SIMAN model, while the second-moment approximation method is almost one thousand times faster than the same simulation model.

Besides the analytical approximation models suggested, more recently we proposed a new hybrid method that appeared as an alternative way, which can consider very general distributions, faster than the standard simulation but slower that analytical model, which can easily implement more complex distribution. Such method aggregate Max-Plus Algebra and Markov Chain extending the model power capability of stochastic Max-plus systems. The main inspiration for this proposal was considered the same load-haulage cycle. It is a system subjected to synchronization, delay and decision phenomena.

Since that Max-Plus Algebra does not deal directly with decisions, we divided the system into different modes of operations without this phenomena. Moreover, we created a Markov Chain to compute the probabilities of the system operates in each mode. As a result, we observed that the aggregation of Max-Plus Algebra and Markov Chain consists of an efficient alternative to perform stochastic DES systems. Despite the novelty of this kind of model, simulation results show that there is an equivalence between the system modeled by the hybrid method presented and by the standard simulation tool. Moreover, the hybrid method is almost three times faster.

Taken this into account, we can state that the hybrid methodology presented can be a good alternative for depicting stochastic problem once that the system behavior is understood efficiently from Max-Plus Algebra and Markovian properties. In spite of this advance, it is necessary to avoid the drastically growth of the state space cardinality of the Markov Chain. Although it can be done using previous knowledge of the problem, for future works we recommend the adaptation and utilization of numerical solutions which deal with large sparse linear equation systems.

Another point is the fact that it was considered a stochastic DES problem in which the number of trucks does not change too often. Thus, we also suggest for future works a broad investigation which deals with more general cases, including systems with dynamical-structure which changing from one mode of operation to other too often. We believe that this future investigation can extend the application of the hybrid methodology, in which, it will be possible to treat more general stochastic problems.

Finally, this Thesis showed different ways of modeling a load-haulage cycle. Since that those open-pit mining systems are composed of many load-haulage cycles, for future works we indicate the application of the modeling methods presented to design a complete open-pit mine as well. Such initiative can be useful for determining the appropriate number of trucks in each load-haulage cycle of an open-pit mine.

Bibliography

Adams WP, Sherali HD (1986) A tight linearization and an algorithm for zero-one quadratic programming problems. Management Science 32(10):1274–1290 94

Airulla DG, Zaky M, Joelianto E, Sutarto HY (2016) Design and simulation of traffic light control system at two intersections using max-plus model predictive control. Int Journal Intelligent Eng and Sys (2016) 79

Akian M, Bapat R, Gaubert S (2006) Max-plus algebra. Handbook of linear algebra (discrete mathematics and its applications) 39:10–14 7

Altiok T (1985) On the phase-type approximations of general distributions. IIE Transactions 17(2):110–116, DOI 10.1080/07408178508975280 9, 58

Alur R, Dill DL (1994) A theory of timed automata. Theoretical Computer Science 126(2):183–235, DOI 10.1016/0304-3975(94)90010-8 15

Archer NP, Ghasemzadeh F (1999) An integrated framework for project portfolio selection. International Journal of Project Management 17(4):207–216 2

Baccelli F, Cohen G, Olsder GJ, Quadrat JP (1992) Synchronization and linearity, vol 2. Wiley New York 76, 77

Balbo, Bruell G, Schwetman SC, D H (1977) Customer Classes and Closed Network Models-A Solution Technique. IFIP Congress pp 559–564 56

Banks J (2000) Introduction to simulation. In: Simulation Conference, 2000. Proceedings. Winter, IEEE, vol 1, pp 9–16, DOI 10.1109/WSC.2000.899690 64

Banks J (2005) Discrete-Event System Simulation. Pearson 1

Basharin GP, Langville AN, Naumov VA (2004) The life and work of aa markov. Linear Algebra and its Applications 386:3–26, DOI 10.1016/j.laa.2003.12.041 27

Bertsekas DP, Tsitsiklis JN (2002) Introduction to probability, vol 1. Athena Scientific Belmont, MA 22, 24, 25, 27, 28, 31, 32

Bolch G, Greiner S, de Meer H, Trivedi KS (2006) Queueing Networks and Markov Chains: Modeling and Performance evaluation with Computer Science applications. John Wiley & Sons 3, 8, 28, 36, 43, 50, 55, 61, 63, 84, 90

Burke PJ (1956) The output of a queuing system. Operations research 4(6):699–704, DOI 10.1287/opre.4.6.699 42

Buzen JP (1973) Computational Algorithms for Closed Queueing Networks with Exponential Servers. Com of ACM 16, DOI 10.1145/362342.362345 xvii, 46, 47, 51, 54

Cadenas JM, Carrillo JV, Garrido MC, Ivorra C, Liern V (2012) Exact and heuristic procedures for solving the fuzzy portfolio selection problem. Fuzzy optimization and Decision making 11(1):29–46 91

Casella I, Sanches B, Sguarezi Filho A, Capovilla C (2016) A dynamic residential load model based on a non-homogeneous poisson process. Journal of Control, Automation and Electrical Systems 27(6):670–679 27

Cassandras CG, Lafortune S (2009) Introduction to discrete event systems. Springer Science & Business Media 8, 10, 13, 15, 18, 21, 22, 24, 25, 26, 27, 28, 31, 33, 34, 41, 45, 59, 64

Chwif L, Medina AC (2007) Modelagem e Simulação de Eventos Discretos: Teoria e Aplicações, 2nd edn 1

Cooper RB (1981) Introduction to queueing theory. North Holland 27, 37, 40, 41

Costa HR (2011) Apoio à seleção de portfólio de projetos de software baseado na moderna teoria do portfólio. PhD thesis, Universidade Federal do Rio de Janeiro - UFRJ 2

Costa HR, de Oliveira Barros M, Rocha AR (2010) Software project portfolio selection a modern portfolio theory based technique. In: SEKE, pp 387–392 90, 91

Cox DR (1955) A use of complex probabilities in the theory of stochastic processes. In: Mathematical Proceedings of the Cambridge Philosophical Society, Cambridge Univ Press, vol 51, pp 313–319, DOI 10.1017/S0305004100030231 9, 58

Cui L, Xu Y, Zhao X (2010) Developments and Applications of the Finite Markov Chain Imbedding Approach in Reliability. IEEE Transactions on Reliability 59(4):685–690 27

Dias J, Maia C, Lucena Jr V (2016) Synchronising operations on productive systems modelled by timed event graphs. International Journal of Production Research 54(15):4403–4417, DOI 10.1080/00207543.2015.1041573 76, 77, 79

Ding W, Cao R (2008) Methods for selecting the optimal portfolio of projects. In: Service Operations and Logistics, and Informatics, 2008. IEEE/SOLI 2008. IEEE International Conference on, IEEE, vol 2, pp 2617–2622 93

Ekren BY, Heragu SS, Krishnamurthy A, Malmborg CJ (2013) An approximate solution for semi-open queueing network model of an autonomous vehicle storage and retrieval system. IEEE Transactions on Automation Science and Engineering 10(1):205–215, DOI 10.1109/TASE.2012.2200676 60

Ercelebi S, Bascetin A (2009) Optimization of shovel-truck system for surface mining. Journal of The Southern African Institute of Mining and Metallurgy 109(7):433–439 66

Erlang AK (1917) Solution of some problems in the theory of probabilities of significance in automatic telephone exchanges. Elektrotkeknikeren 13:5–13 58

Fu MC (2002) Optimization for Simulation: Theory vs. Pratice. INFORMS Journal on Computing 14:192–215 2, 5

Gardiner CW (1985) Handbook of stochastic methods. Advances in Mathematics 55(1):101, DOI 10.1016/0001-8708(85)90015-5 29

Goldbarg MC, Luna HPL (2005) Otimização combinatória e programação linear: modelos e algoritmos. Elsevier 93

Gordon WJ, Newell RR (1967) Closed Queueing Systems with Exponential Servers. Oper Research 15:254–265, DOI 10.1287/opre.15.2.254 45, 50

Goverde RM, Bovy PH, Olsder GJ (1999) The max-plus algebra approach to transportation problems. In: World Transport Research: Selected Proceedings of the 8th World Conference on Transport Research, Volume 3 79

Guo H, Pohl E, Gerokostopoulos A (2013) Determining the right sample size for your test: theory and application. In: 2013 Reliability and Maintainability Symposium, pp 1–9, DOI 10.1109/TR.2012.2230493 7

Gurobi Optimization I (2016) Gurobi optimizer reference manual. URL http://www.gurobi.com 95

Hayes B, et al (2013) First links in the markov chain. American Scientist 101(2):252, DOI 10.1511/2013.101.92 27

Heidergott B, Olsder GJ, Van Der Woude J (2014) Max Plus at work: modeling and analysis of synchronized systems: a course on Max-Plus algebra and its applications. Princeton University Press 79, 80

Hongyan L, Zhong W (2009) The Application of Markov Chain into the Forecast for Population Age Structure in Shanghai. In: IEEE Computational Intelligence and Software Engineering, vol 694, pp 5–7 28

Huang X (2012) A risk index model for portfolio selection with returns subject to experts' estimations. Fuzzy Optimization and Decision Making 11(4):451–463 91

Iman RL (2008) Latin hypercube sampling. Encyclopedia of quantitative risk analysis and assessment 70

Jackson JR (1957) Networks of waiting lines. Operations research 5(4):518–521, DOI 10.1287/opre.5.4.518 42

Kappas G, Yegulalp TM (1991) An application of closed queueing networks theory in truck-shovel systems. International Journal of Surface Mining, Reclamation and Environment 50

Kellner MI, Madachy RJ, Raffo DM (1999) Software process simulation modeling: Why? What? How? The Journal of Systems and Software pp 91–105 1

Lagershausen S (2012) Performance analysis of closed queueing networks, vol 663. Springer Science & Business Media, DOI 10.1007/978-3-642-32214-3 61

Lee H, Lee JH (2014) An auto-framing method for stochastic process signal by using a hidden markov model based approach. International Journal of Control, Automation and Systems 12(2):251–258 27

Lindbeck A (2001) The sveriges riksbank (bank of sweden) prize in economic sciences in memory of alfred nobel 1969–2000. In: The Nobel Prize: The First 100 Years, World Scientific, pp 197–219 90

Little JD (1961) A proof for the queuing formula: L= λ w. Operations research 9(3):383–387 44

Marconi MA, Lakatos EM (2006) Fundamentos de metodologia científica, 6th edn 64

Marie R (1980) Calculating equilibrium probabilities for λ (n)/c k/1/n queues. ACM Sigmetrics Performance Evaluation Review 9(2):117–125, DOI 10.1145/800199.806155 62

Marie RA (1979) An approximate analytical method for general queueing networks. IEEE Transactions on Software Engineering (5):530–538, DOI 10.1109/TSE.1979.234214 60, 61

Marie RA (2011) Disappointments and Delights, Fears and Hopes induced by a few decades in Performance Evaluation. In: Perf. Eval. of Comp. and Communication Systems. Milestones and Future Challenges, Springer, pp 1–9, DOI 10.1007/978-3-642-25575-5 3, 8, 58

Marie RA, Stewart WJ (1977) A hybrid iterative-numerical method for the solution of a general queueing network. In: Proceedings of the Third International Symposium on Measuring, Modelling and Evaluating Computer Systems, North-Holland Publishing Co., pp 173–188 62

Markowitz H (1952) Portfolio selection. The journal of finance 7(1):77–91 91, 92

Mehmood R, Crowcroft J (2005) Parallel iterative solution method for large sparse linear equation systems. Tech. rep., Uni. of Cambridge, Computer Lab 86

Nelson BL (2010) Optimization via Simulation Over Discrete Decision Variables. Tutorials in Operations Research - INFORMS pp 193–207, DOI 10.1287/educ.1100.0069 2

Okasha S (2016) Philosophy of Science: Very Short Introduction. Oxford University Press 64

Oral M, Kettani O (1992) Reformulating nonlinear combinatorial optimization problems for higher computational efficiency. European Journal of Operational Research 58(2):236–249 94

Papadimitriou CH, Steiglitz K (1998) Combinatorial Optimization: Algorithms and Complexity. Courier Corporation 94

Pegden CD, Davis DA (1992) Arena: a siman/cinema-based hierarchical modeling system. In: Proceedings of the 24th conference on Winter simulation, ACM, pp 390–399, DOI 10.1145/167293.167390 65

Pennypacker JS, Dye LD (2002) Project portfolio management and managing multiple projects: two sides of the same coin. Managing multiple projects pp 1–10 91

Pisinger D (1995) Algorithms for Knapsack Problems. PhD thesis, University of Copenhagen 94, 95

Pollaczek F (1930) Über eine aufgabe der wahrscheinlichkeitstheorie. i. Mathematische Zeitschrift 32(1):64–100, DOI 10.1007/BF01194620 58

Profozich DM, Sturrock DT (1995) Introduction to siman/cinema. In: Proceedings of the 27th conference on Winter simulation, IEEE Computer Society, pp 515–518, DOI 10.1109/WSC.1995.478784 7, 65

R Core Team (2013) R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, URL http://www.R-project.org/ 73, 88

Ribeiro GR, Saldanha RR, Maia CA (2018) Analysis of Decision Stochastic Discrete-Event Systems Aggregating Max-Plus Algebra and Markov Chain. Journal of Control, Automation and Electrical Systems pp 1–10, DOI 10.1007/s40313-018-0394-7 76

Ribeiro RG, Saldanha RR, A MC (2016) Modeling and portfolio optimization of stochastic discreteevent system through markovian approximation: An open-pit mine study. In: EUROSIM Congress on Modelling and Simulation, Oulu, Finland 68, 101

Ruessink BG (2006) Application of Markov Chain simulation for model calibration. In: IEEE International Joint Conference on Neural Networks, 2, pp 4318–4325 27

Santos FFG, Vieira DAG, Saldanha RR, Lisboa AC, Lobato MVdC (2014) Seasonal energy trading portfolio based on multiobjective optimisation. International Journal of Logistics Systems and Management 17(2):180–199 90, 91

Shi P, Li F (2015) A survey on markovian jump systems: modeling and design. International Journal of Control, Automation and Systems 13(1):1–16 27

Souza JS, Neto FJK, Anzanello MJ, Filomena TP (2012) A non-traditional capital investment criteria-based method to optimize a portfolio of investments. International Journal of Industrial Engineering 19(4) 91

Tan Y, Takakuwa S (2016) A practical simulation approach for an effective truck dispatching system of open pit mines using vba. In: Winter Simulation Conference (WSC), 2016, IEEE, pp 2394–2405, DOI 10.1109/WSC.2016.7822279 65

Tan Y, Miwa K, Chinbat U, Takakuwa S (2012) Operations modeling and analysis of open pit copper mining using gps tracking data. In: Proceedings of the Winter Simulation Conference, Winter Simulation Conference, p 117, DOI 10.1109/WSC.2012.6465062 65

Torkamani E, Askari-Nasab H (2012) Verifying short-term production schedules using truck-shovel simulation. Relatório técnico, Mining Optimization Laboratory (MOL) 65

Vamanan M, Wang Q, Batta R, Szczerba RJ (2004) Integration of cots software products arena & cplex for an inventory/logistics problem. Computers & Operations Research 31(4):533–547, DOI 10.1016/S0305-0548(03)00010-8 65

Von Neumann J (1951) 13. various techniques used in connection with random digits. Appl Math Ser 12(36-38):3 8

Walker E, Nowacki AS (2011) Understanding equivalence and noninferiority testing. Journal of general internal medicine 26(2):192–196 71, 88

Wang Z, Li S (2011) Some Application of Markov chain to Market Occupation Rate and Promotion Strategy. In: IEEE Fourth International Conference on Information and Computing, DOI 10.1109/ICIC.2011.109 27

Wichmann BA, Hill I (2006) Generating good pseudo-random numbers. Computational Statistics & Data Analysis 51(3):1614–1622, DOI 10.1016/j.csda.2006.05.019 8

Xie X (2014) Formal methods in manufacturing. CRC Press 79

Ye N, Member S, Zhang Y, Borror CM (2004) Robustness of the Markov-Chain Model for Cyber-Attack Detection US Air Force Office of Scientific Research. IEEE Transactions on Reliability 53(1):116–123 27

Yin M, Zhao SZ (2008) An adaptive n-stage markov chain combined model for collaboration trust maintenance in supply chain. In: Wireless Communications, Networking and Mobile Computing, 2008. WiCOM'08. 4th International Conference on, IEEE, pp 1–4 27

Yu W, Meniber S, Sheblt GB, Matos MA (2004) Application of Markov Chain Models for Short Term Generation Assets Valuation. In: IEEE International Conference on Probabilistic Methods Applied to Power Systems, pp 343–348 27

Sample randomly generated

Table A.1: Sample (from 1 to 30)

N trucks	$E[ST_j(\boldsymbol{\theta})]$ for $1 \leq j \leq 10$ (minutes) Min=10 Max=30 cv_j for $1 \leq j \leq 10$ Min=0.01 Max=1										p_1	p_2	p_3	p_4
5	24.39	29.12	24.77	16.24	13.47	22.51	29.64	27.30	21.01	12.32	0.41	0.11	0.34	0.20
	0.36	0.13	0.33	0.41	0.80	0.43	0.83	0.21	0.88	0.80				
17	24.77	17.36	26.68	25.25	28.16	19.65	14.40	11.80	23.93	22.68	0.42	0.46	0.08	0.03
	0.99	0.38	0.86	0.92	0.20	0.35	0.20	0.38	0.13	0.24				
6	29.88	20.37	27.73	14.03	15.25	15.77	10.50	29.93	19.85	20.64	0.36	0.24	0.42	0.43
	0.29	0.51	0.14	0.28	0.45	0.27	0.39	0.65	0.55	0.76				
11	20.18	13.16	22.18	21.80	18.78	20.27	15.17	22.66	12.65	25.28	0.19	0.31	0.41	0.38
	0.46	0.77	0.17	0.52	0.83	0.21	0.49	0.44	0.67	0.70				
18	17.37	29.45	15.47	22.69	26.35	15.20	13.42	14.21	26.15	25.78	0.09	0.19	0.31	0.36
	0.89	0.48	0.79	0.80	0.69	0.61	0.82	0.53	0.23	0.60				
8	27.20	10.63	16.95	18.94	18.25	18.96	21.87	25.20	17.61	29.38	0.44	0.33	0.11	0.23
	0.26	0.63	0.50	0.27	0.78	0.53	0.71	0.34	0.44	0.97				
13	26.48	22.37	29.15	11.82	23.20	24.75	17.62	24.55	21.73	27.23	0.46	0.48	0.10	0.11
	0.17	0.16	0.94	1.00	0.93	0.11	0.79	0.17	0.98	0.38				
20	16.55	19.41	14.42	29.10	21.79	13.90	16.81	23.31	13.27	28.67	0.47	0.21	0.25	0.24
	0.40	0.81	0.39	0.64	0.38	0.43	0.58	0.24	0.25	0.16				
8	14.45	12.27	29.51	14.76	12.18	27.51	25.32	18.16	20.21	19.29	0.31	0.36	0.21	0.20
	0.33	0.66	0.94	0.22	0.96	0.99	0.57	0.63	0.59	0.55				
17	13.00	25.62	11.65	29.69	26.87	10.97	15.67	17.34	16.52	13.94	0.22	0.07	0.16	0.31
	0.77	0.76	0.44	0.65	0.76	0.55	0.87	0.79	0.32	0.74				
15	11.35	21.01	16.56	10.77	27.63	14.12	12.68	21.47	23.00	11.03	0.21	0.14	0.30	0.35
	0.56	0.71	0.55	0.33	0.90	0.70	0.33	0.71	0.17	0.64				
19	29.30	21.66	19.46	18.20	11.82	23.56	28.63	14.97	15.97	18.63	0.03	0.43	0.47	0.08
	0.41	0.88	0.91	0.36	0.22	0.40	0.54	0.77	0.13	0.65				
7	21.53	15.59	22.69	22.27	28.79	28.94	26.14	15.84	17.22	24.29	0.05	0.18	0.49	0.29
	0.22	0.58	0.63	0.50	0.15	0.85	0.30	0.27	0.40	0.34				
14	10.56	28.08	24.25	26.02	25.61	24.40	24.24	13.37	22.60	17.97	0.33	0.16	0.28	0.47
	0.12	0.11	0.34	0.21	0.29	0.15	0.26	0.14	0.61	0.28				
14	28.04	16.91	17.47	25.59	24.60	26.75	26.72	10.21	15.01	11.54	0.05	0.40	0.19	0.44
	0.83	0.56	0.21	0.57	0.42	0.90	0.73	0.36	0.70	0.88				
12	14.87	25.31	20.38	23.83	16.92	17.21	18.60	12.03	18.11	26.50	0.24	0.39	0.33	0.01
	0.76	0.21	0.42	0.15	0.58	0.71	0.96	0.75	0.29	0.57				
3	27.95	16.51	18.40	20.51	21.16	29.42	20.37	23.43	28.03	23.57	0.08	0.08	0.39	0.04
	0.95	0.68	0.98	0.10	0.55	0.76	0.93	0.59	0.80	0.83				
2	11.04	27.40	21.02	13.61	15.58	26.53	12.32	26.50	19.07	14.91	0.01	0.26	0.23	0.50
	0.51	0.93	0.27	0.75	0.87	0.79	0.35	0.96	0.35	0.37				
9	25.92	23.33	28.66	17.98	17.70	21.95	23.05	12.89	29.40	14.15	0.28	0.28	0.03	0.41
	0.43	0.35	0.79	0.87	0.71	0.93	0.24	0.99	0.57	0.52				
6	18.70	11.96	12.70	12.32	23.45	11.41	28.70	19.63	10.79	22.19	0.11	0.49	0.25	0.12
	0.65	0.25	0.10	0.78	0.66	0.47	0.90	0.85	0.75	0.92				
4	13.36	19.10	10.81	21.13	19.45	25.93	23.82	25.63	25.60	28.47	0.26	0.01	0.14	0.46
	0.14	0.45	0.52	0.89	0.47	0.25	0.12	0.90	0.42	0.26				
9	15.82	18.66	26.29	26.79	16.17	22.75	22.02	17.31	27.22	19.65	0.38	0.37	0.05	0.06
	0.55	0.95	0.75	0.95	0.27	0.52	0.45	0.28	0.51	0.16				
14	19.52	13.74	13.65	10.02	12.73	21.04	11.29	28.88	24.38	21.72	0.27	0.43	0.38	0.33
	0.73	0.86	0.71	0.40	0.13	0.18	0.15	0.41	0.95	0.43				
12	22.31	24.15	15.12	28.42	10.70	17.91	19.45	27.42	10.16	17.23	0.14	0.12	0.13	0.13
	0.20	0.29	0.58	0.44	0.18	0.78	0.62	0.11	0.79	0.48				
16	23.13	26.38	23.43	19.99	14.20	28.29	11.48	11.09	13.69	28.00	0.34	0.05	0.18	0.15
	0.81	0.53	0.67	0.71	1.00	0.59	0.41	0.92	0.67	0.69				
10	12.22	23.24	18.73	24.41	25.16	18.41	27.41	20.97	27.95	10.08	0.39	0.03	0.46	0.28
	0.93	0.98	0.23	0.48	0.50	0.34	0.18	0.46	0.85	0.44				
4	18.60	15.11	21.52	13.01	22.04	12.48	21.23	28.29	14.46	15.41	0.17	0.04	0.01	0.37
	0.64	0.32	0.69	0.84	0.34	0.65	0.49	0.49	0.48	0.11				
17	23.63	11.08	12.64	27.60	10.36	12.94	25.48	18.80	11.60	13.01	0.15	0.29	0.02	0.26
	0.60	0.25	0.85	0.67	0.56	0.83	0.67	0.56	0.85	1.00				
19	20.78	14.58	10.06	17.29	29.63	10.02	19.24	20.40	25.11	16.56	0.13	0.33	0.44	0.09
	0.70	0.42	0.48	0.60	0.63	0.95	0.69	0.83	0.21	0.19				
2	16.70	26.81	25.44	15.37	20.19	16.05	16.05	16.61	29.27	21.25	0.49	0.22	0.37	0.17
	0.88	0.83	0.30	0.18	0.34	0.30	0.98	0.68	0.91	0.89				

General results

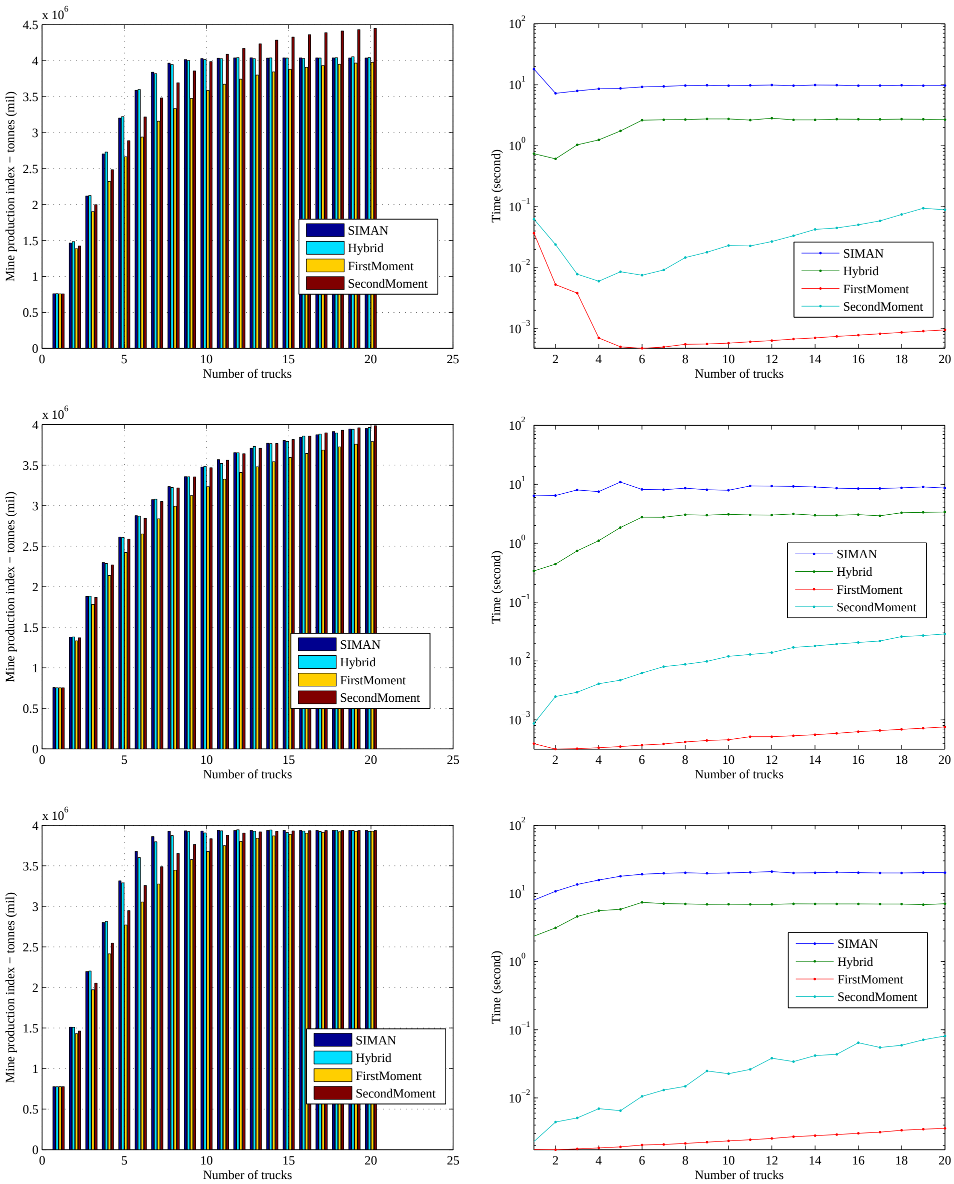


Figure B.1: For the sample from 1 to 3, the graphics in the left show $E[Prod]$ alternating the number of trucks, while the graphics in the right depict the time taken to run.

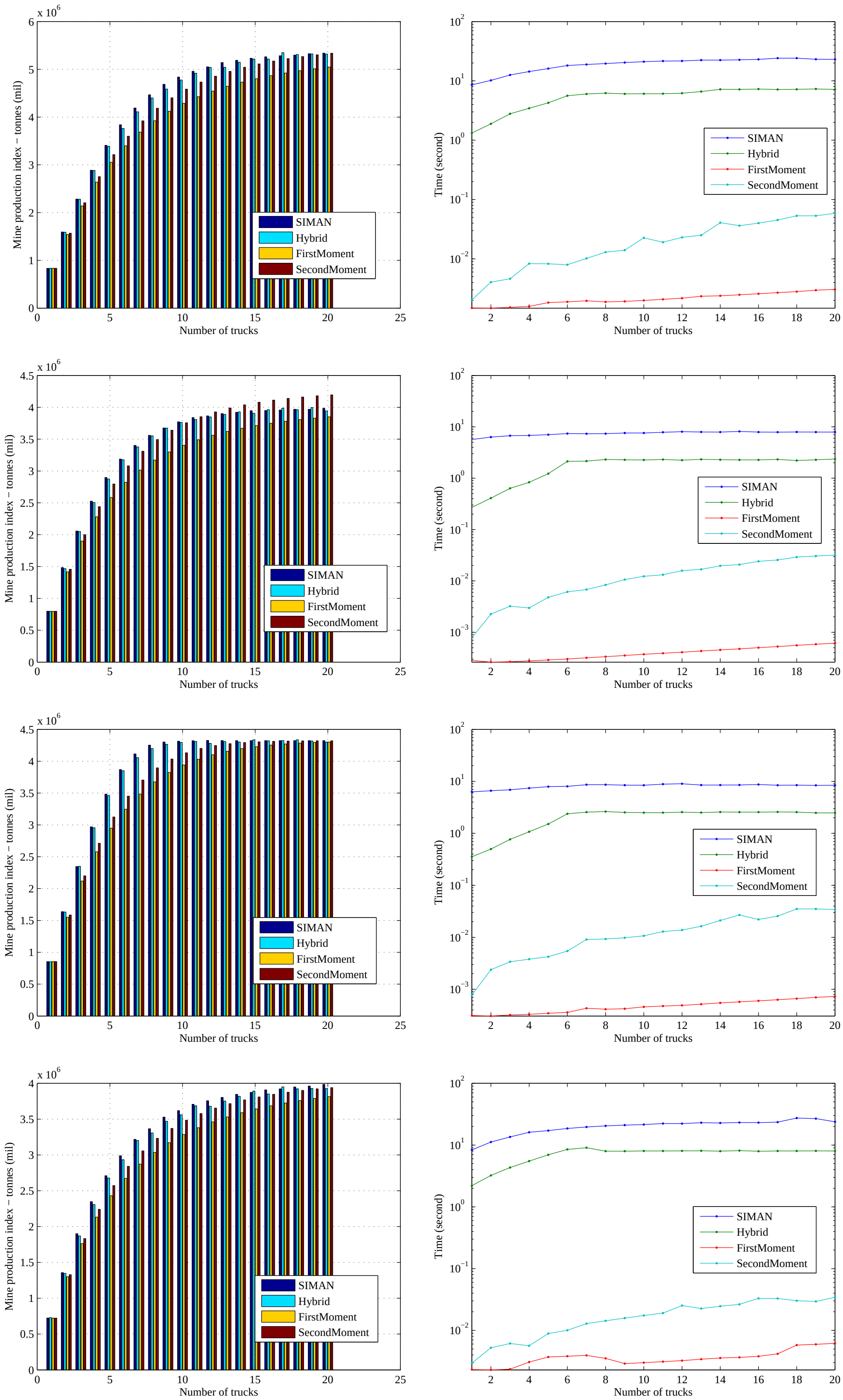


Figure B.2: For the sample from 4 to 7, the graphics in the left show $E[Prod]$ alternating the number of trucks, while the graphics in the right depict the time taken to run.

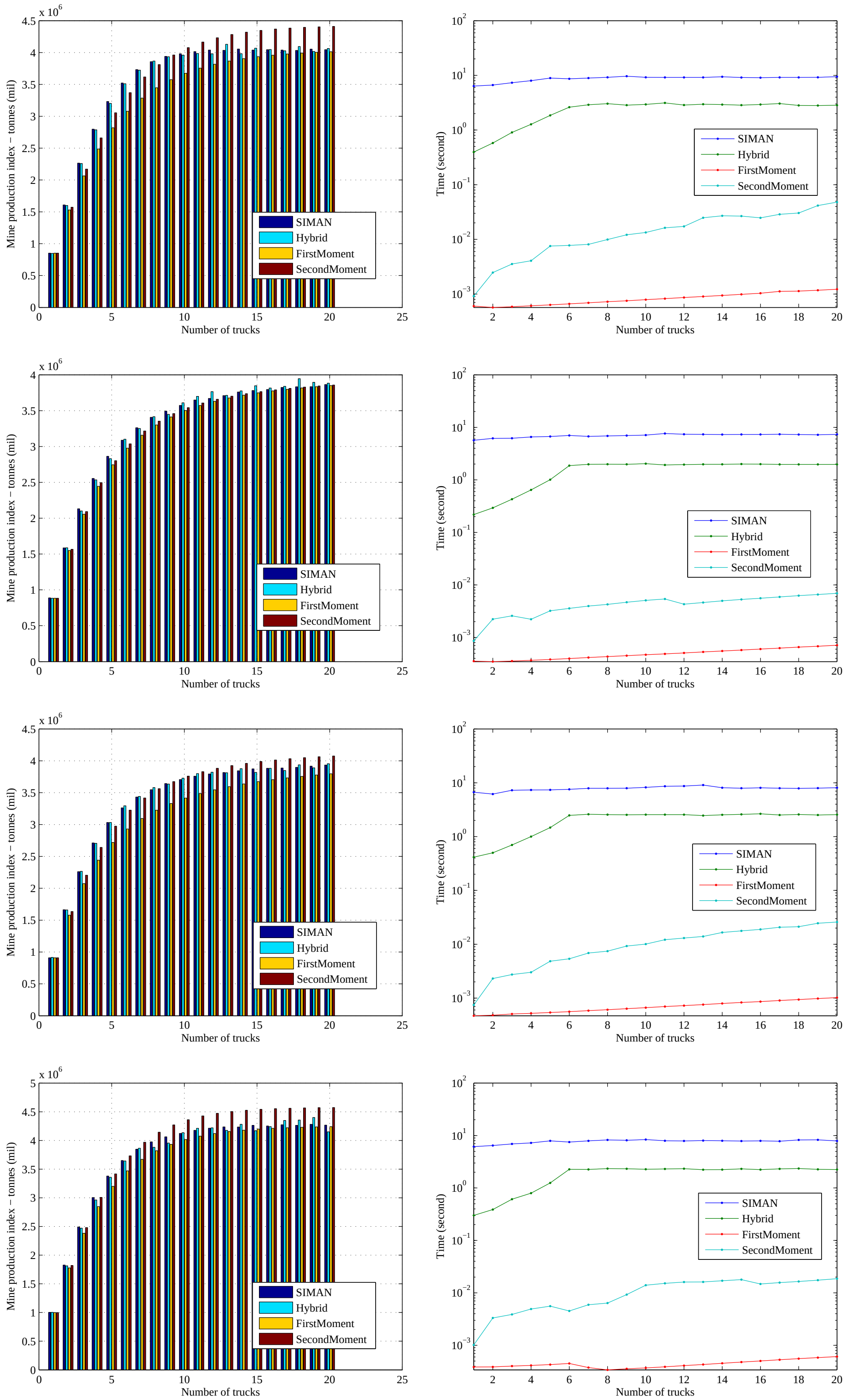


Figure B.3: For the sample from 8 to 11, the graphics in the left show $E[Prod]$ alternating the number of trucks, while the graphics in the right depict the time taken to run.

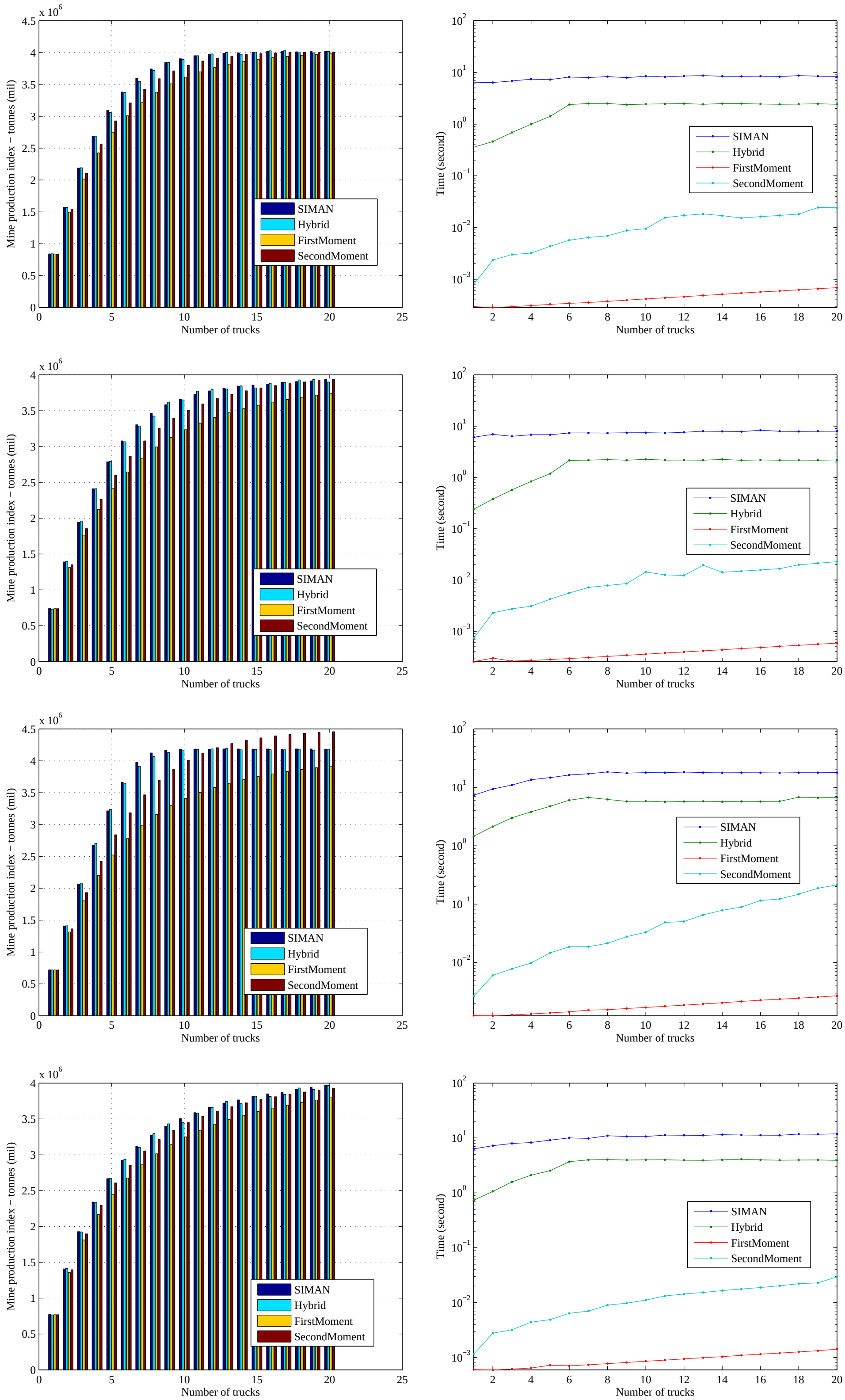


Figure B.4: For the sample from 12 to 15, the graphics in the left show $E[Prod]$ alternating the number of trucks, while the graphics in the right depict the time taken to run.

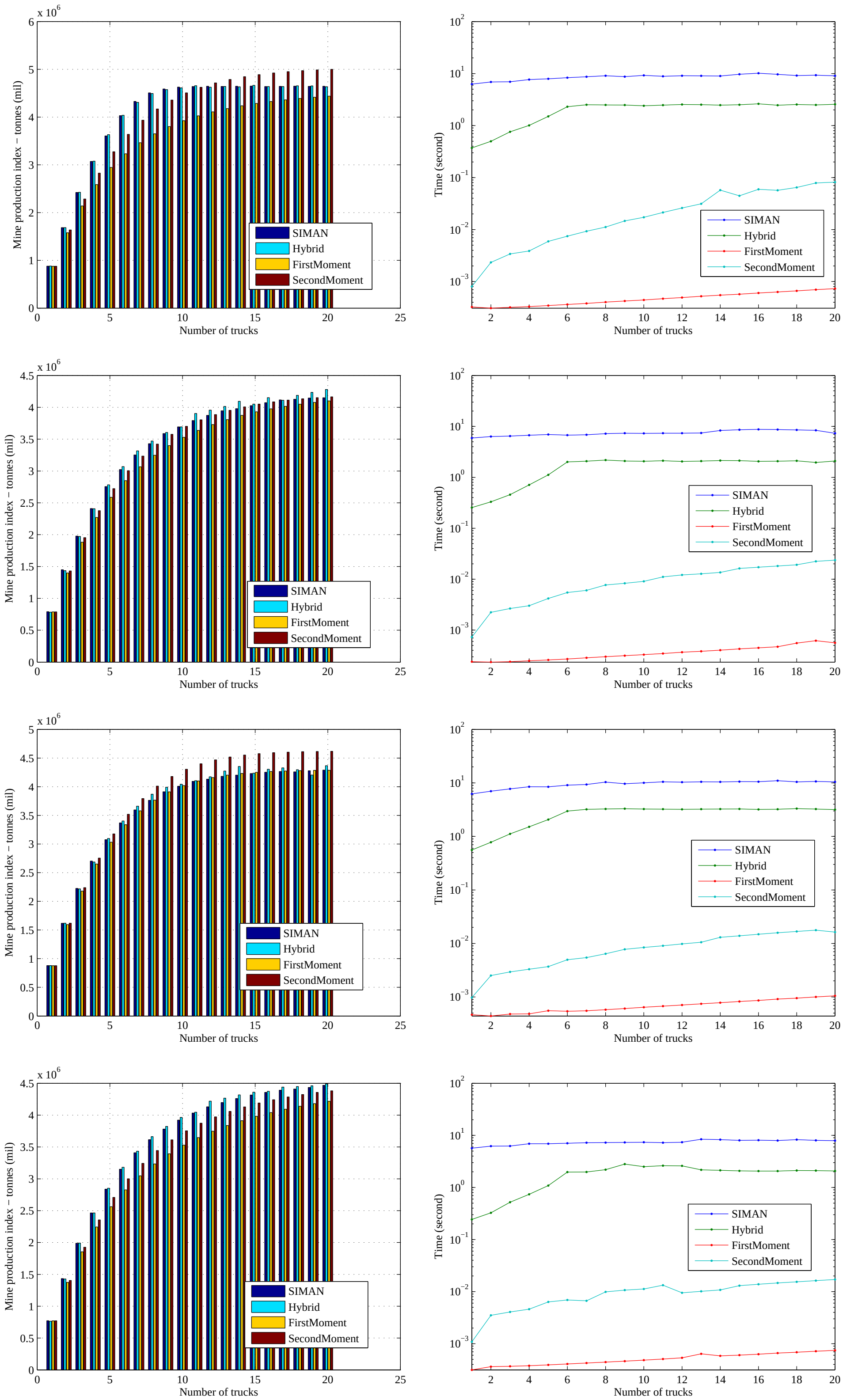


Figure B.5: For the sample from 16 to 19, the graphics in the left show $E[Prod]$ alternating the number of trucks, while the graphics in the right depict the time taken to run.

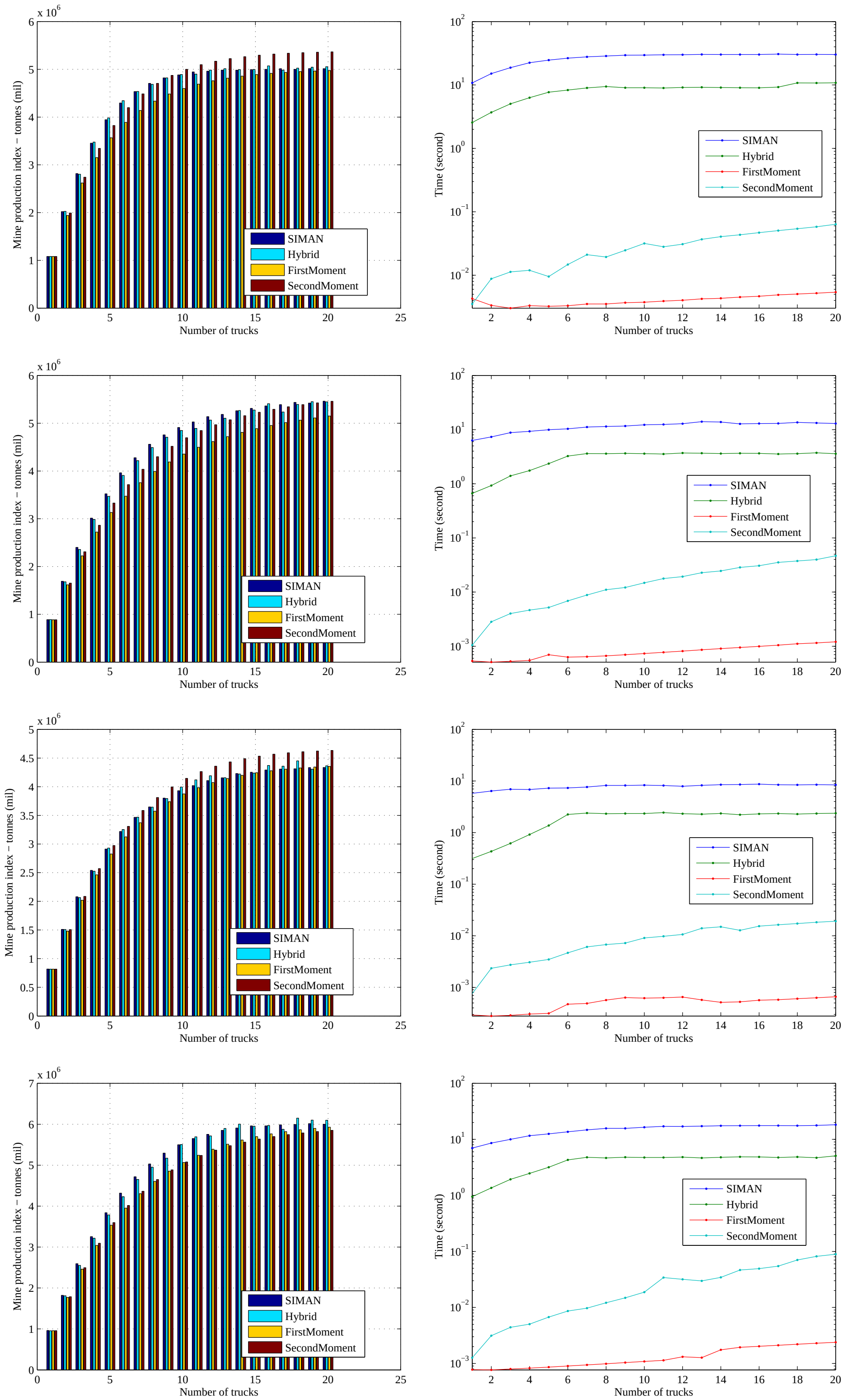


Figure B.6: For the sample from 20 to 23, the graphics in the left show $E[Prod]$ alternating the number of trucks, while the graphics in the right depict the time taken to run.

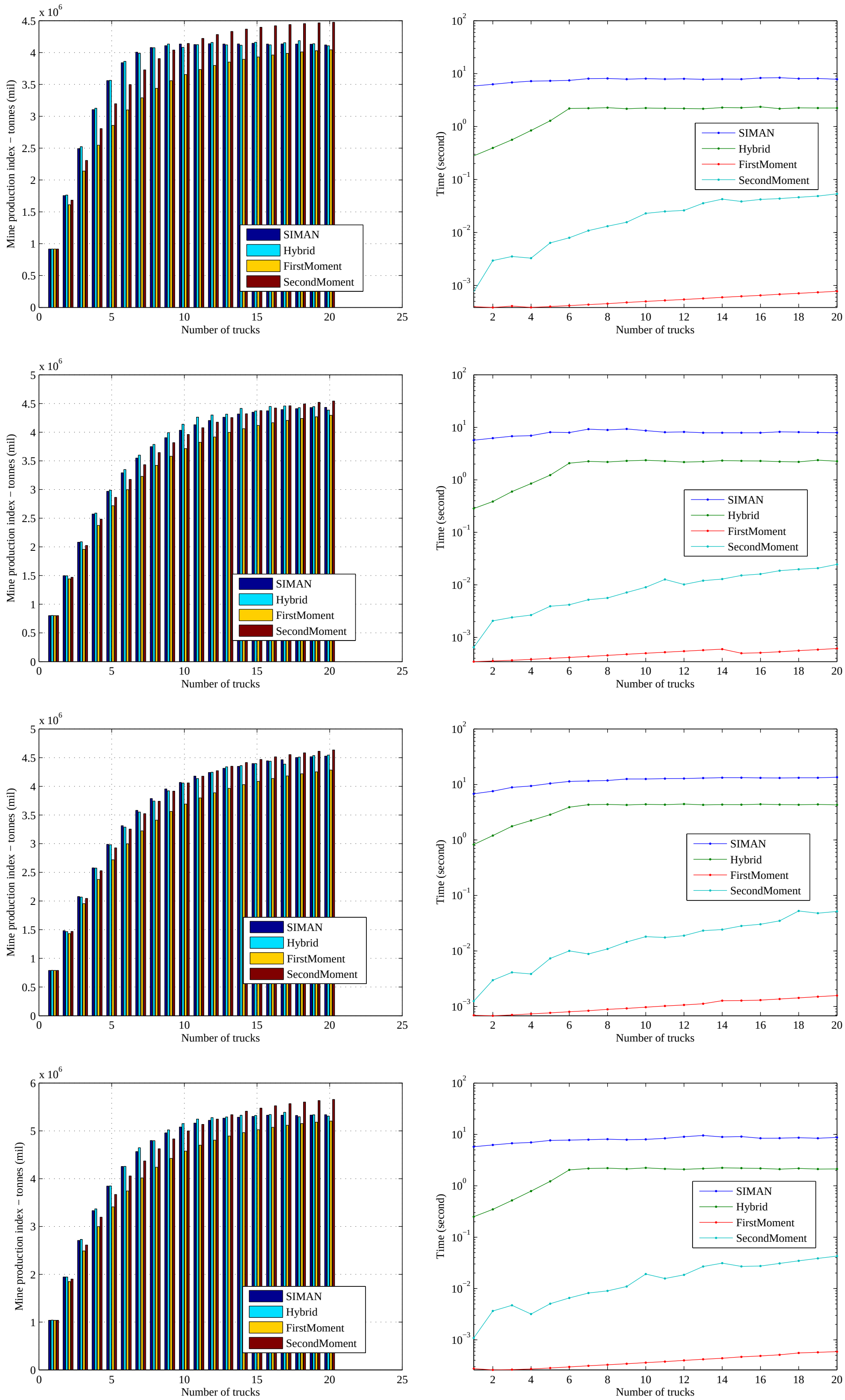


Figure B.7: For the sample from 24 to 27, the graphics in the left show $E[Prod]$ alternating the number of trucks, while the graphics in the right depict the time taken to run.

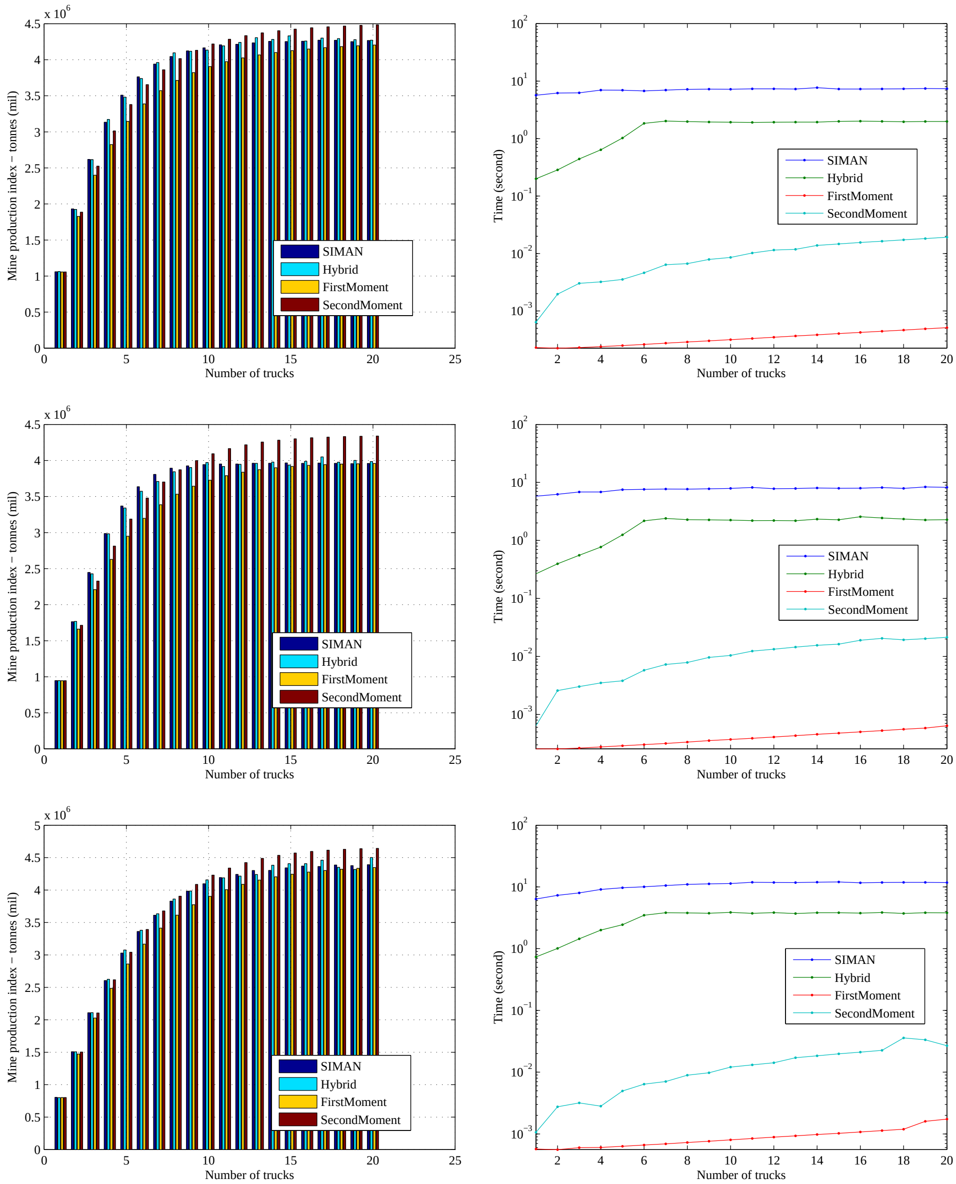


Figure B.8: For the sample from 28 to 30, the graphics in the left show $E[Prod]$ alternating the number of trucks, while the graphics in the right depict the time taken to run.