University of Hradec Králové
Faculty of Informatics and Management
Department of Information Technologies

Some Specific Problems in the Applications of Discrete Event Systems

DOCTORAL THESIS

Author: Ing. Bc. Zuzana Němcová
Supervisor: prof. RNDr. Martin Gavalec, CSc.

Hradec Králové, 2016

To my family

Contents

List of Figures

List of symbols and abbreviations

$\mathbb{N}$	the set of natural numbers
$\mathbb{R}$	the set of real numbers
$\overline{\mathbb{R}}$	set of real numbers extended by $\varepsilon = -\infty$ (in max-plus algebra)
$\mathbb{Z}$	the set of integers
$\wedge$	conjunction
$\vee$	disjunction
$\triangle$	operation of conjunction
∇	operation of disjunction
$\cap$	intersection of sets
$\cup$	union of sets
ε	absorbing value in max-plus algebra
e	neutral element in max-plus algebra
$\oplus$	maximum operation (max-plus algebra)
$\otimes$	addition operation (max-plus algebra)
$\oplus$	maximum operation (max-min algebra)
$\otimes$	minimum operation (max-min algebra)
$\otimes_{\mathrm{d}}$	drastic operation
$\otimes_G$	Gödel operation
$\otimes_L$	Łukasiewicz operation
$\otimes_p$	product operation

A	matrix A
a_{ij}	the (i,j)th entry of A
$a^{\otimes k}$	$a \otimes a \otimes a \cdots \otimes a$ (value a appears k times)
$A^{\otimes k}$	$A \otimes A \otimes A \cdots \otimes A$ (matrix A appears k times)
$A^{(1)}$	A with entries $(a_{ij} - 1)$
x	vector x
$x(t)$	vector x at time t
λ	eigenvalue
$\lambda(A)$	eigenvalue of A
V	set of nodes, $V = \{v_1, v_2, \ldots\}$
E	set of edges $E = \{e_1, e_2, \ldots\}$
$G(A)$	communication graph of A
$G^c(A)$	critical graph of A
$V^c(A)$	nodes belonging to critical graph of A
$E^c(A)$	edges belonging to critical graph of A
$w(\rho)$	weight of the path ρ
$l(\rho)$	length of the path ρ
DES	discrete event system
FCFS	first-come-first-served regime
LCFS	last-come-first-served regime
lcm	least common multiple
SCC*	strongly connected component
SR	set of all strong paths
WR	set of all weak paths

Declaration

I hereby declare that the thesis is my original work and it has been written by me in its entirety. I have acknowledged all sources used and have cited these in the reference section. This thesis has also not been submitted for any degree in any university previously.

Hradec Králové
March, 2016 Zuzana Němcová

Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisor Prof. Martin Gavalec for the continual support of my Ph.D. study and related research, for his patience, motivation, and immense knowledge. His guidance helped me over the entire time of this research and the writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D. study.

Besides my advisor, I would like to thank Prof. Peter Mikulecký for his insightful comments and encouragement.

Also I would like to thank my loving family for supporting me spiritually throughout writing this thesis and my life in general.

Abstract

The present thesis treats the solution of specific problems of discrete event systems arising in the modeling of production systems, traffic or computer networks, and others.

The thesis is focused on two areas of such problems. The first one is the control of the queues in a manufacturing system in order to optimize the total production cost. The service times at particular servers naturally differ in such a system. As a consequence, queues are created—the queues in front of the slower servers can increase, whereas the queues in front of fast servers are decreasing and the server can easily become idle. Both mentioned states are economically undesirable. Two methods of finding the optimal setup of service capacities of the servers in each time period during which the service times stay unchanged are suggested.

The second area of issues covers the description of the steady states of systems performing in discrete time. This description is done using extremal algebras. Transition matrices with fuzzy values are applied to discrete time systems. Various triangular fuzzy t-norms are used, in dependence on the character of the system. The development of the system in time is then described by the power sequence of the transition matrix, computed in the so-called max-t fuzzy algebra. The steady states of the system corresponding to periodic behavior of the power sequence as well as the solution of the eigenproblem are described.

Both the knowledge of the development of the queues and the knowledge of the the steady states in such systems is strategically important for the system manager, who can then flexibly take actions and positively affect the performance of the system.

Abstrakt

Dizertační práce se zabývá řešením specifických problémů, které vznikají při návrhu a modelování systémů diskrétních událostí, např. systémů hromadné obsluhy, počítačových sítí.

Práce se zabývá dvěma oblastmi takových problémů. První oblastí je zkoumání vývoje délek front systému hromadné obsluhy s cílem optimalizovat nastavení systému s ohledem na jeho náklady. Systém hromadné obsluhy se obvykle skládá z několika obslužných míst, kterými musí požadavek projít v daném pořadí. Rychlost obsluhy se může na každém z míst lišit, a proto může dojít v průběhu chodu systému k tvorbě front. Fronty před pomalým obslužným místem mohou růst, zatímco fronty před rychlejšími místy mohou mít klesající charakter. Může se také stát, že obslužné místo zůstane po nějakou dobu nevytížené. Příliš dlouhé fronty, ale i nevytíženost obslužných míst jsou pro výrobní systém nežádoucími stavy. V práci jsou navrženy dvě metody, kterými lze nastavení systému řídit tak, aby k těmto stavům docházelo v co nejmenší možné míře.

Druhou oblastí je popis ustálených stavů systémů pracujících v diskrétním čase. V tomto případě jsou systémy modelovány s využitím tzv. extremálních algeber. Systém je popsán maticí přechodu, jednotlivé jeho stavy pak posloupností vektorů. Vývoj systému v čase zachycuje také posloupnost mocnin matice přechodu. Ustálené stavy je pak možné určit jak z periodického chování těchto posloupností, tak i řešením tzv. vlastního problému. V závislosti na charakteru systému je pro studium ustálených stavů použito různých fuzzy t-norem, posloupnosti vektorů a matic jsou pak počítány v takzvané max-t fuzzy algebře.

Jak znalost vývoje front, tak i znalost potenciálního dosažení ustáleného stavu, jsou pro manažera systému strategicky důležitými informacemi, neboť umožňují odpovídajícím způsobem zareagovat a pozitivně tak ovlivnit výkon tohoto systému.

1. Introduction

1.1 Discrete Event Systems—Motivation

Nowadays, manufacturing technologies play an important role for the development of a country's industrial growth, which dictates the trend of the economy. To find a competitive advantage in a manufacturing environment requires systematic and integrated planning and optimization approaches. In this context, the aim of a manufacturing system is to attain its overall performance by cost reduction as well as by using resources in the development, design, production, delivery and support of the product.

In recent years, industry and also academia have been interested more and more in methods that enable analyzing, modeling, and controlling complex systems, especially discrete event dynamic systems, in order to improve their efficiency, flexibility, and economy.

This thesis introduces the results of an investigation of some issues of discrete event systems using extremal algebras and simulation models as effective tools for studying complex systems.

The simulation of a system in discrete time means modeling the system as an interlaced network consisting of dynamic and static objects. The objects picture the individual entities described by specific characteristics, which influence the passage of the entity through the system. Despite the fact that the simulated time is continuous, the changes of the system state are implemented at discrete moments of time. These systems have in most cases a mass character. Such systems can be seen in real life, for example, a delivery of ordered products, manufacturing products, providing a customer service at the post counter, or a communication network. Naturally, queues occur in these systems; the knowledge of the development of these queues is important for the process and/or also for possible cost optimization. Waiting costs certainly represent a significant part of the total costs. These can be either some kind of penalization for staying in the queue that is inadmissible due to the applied technology, or can be represented by the dissatisfaction of customers standing in a long queue. Therefore the control of the lengths of the queues can lead to considerable savings and thus to an increase in the efficiency of the system.

In the last decades, some special issues of operations research (for example, dynamic programming, path problems, assignment or transportation problems, special scheduling problems, or problems with disjunctive constraints) have been explored by the application of the so-called extremal algebras, which have been widely studied in the literature. In recent years, the importance of these powerful tools has increased. Extremal algebras can help to understand the development of the system dynamics. Many issues in the subject areas of optimization, mathematical physics, geometry, control theory, machine scheduling, discrete event processes, manufacturing systems, telecommunication networks, and traffic control can be solved by using various types of extremal algebras, according to the character of the study. The attractiveness of these tools is grounded in the fact that the classes of the above mentioned nonlinear problems can be studied by applying a linear-algebraic approach.

The state transitions of a discrete event system (abbreviated as DES) are initiated by events (a DES can also be called an event-driven system although this depends on whether the state transitions are synchronized by a clock or occur asynchronously) and these events occur at discrete moments of time. In other words, with each transition of the system, some event can be associated. The event can represent the start/end of some activity, for example, a completion of the product, a customer's arrival, or a machine breakdown. Between these events, time lags of different length can be observed (the lags can be of a deterministic or a stochastic character).

The motivation for studying DES are problems that can arise during the operation of such systems. Lets look more closely at a production system, an assembly line manufacturing some product. The assembly line can consist of several concatenated processing units, the servers. These servers serve the incoming requests and machine the product. The product has to pass through a sequence of servers. The passage of the product through the system depends on the connections of the servers. They can be interconnected in an arbitrary combination of parallel and/or serial connections. Each server offers its service during a given service time, and then the request is transported to the next server. The server can usually start working on a new product if it has finished processing the previous one. The server can serve at most one product in one time instant. It is also assumed that each server starts working as soon as all the needed parts are available. In other words, the server has to wait until the service of the request by the preceding server has ended. Two aspects of such systems can be seen here. Since the service times differ, queues are formed in such a system. The service intensity of a faster server is higher than the service intensity of a slower server and it is clear that the queues in front of a slower server can increase, whereas a faster server can easily become idle. Both states are undesirable, especially from the economic point of view, because they can mean indirect costs. The second aspect is that after the system starts, a transient time period can be seen: after this transient time period, the system can become steady. The parameters of the system can become constant or some differences can appear, but in predictable periods. Whether the system gets into a steady state or not depends on the specific setting of the parameters of the system (with some settings, it can happen that the stability of the system is never achieved).

The state of the system in time t can be described by a state vector, say $x(t)$. The transition matrix, denoted by A, describes the transitions of the system from one state to another. By multiplying the transition matrix and the state vector, the next state of the system, $x(t+1)$, is obtained: it can be written $A \cdot x(t) = x(t+1)$. During the operation of the system, after some time, it can happen that the system reaches some sequence of states repeatedly, $x(k) = x(k+d)$: a period in the system's behavior of length d occurs. The period of the state vector indicates the stable state of the system, the so-called orbit. The stable states of the system can be also investigated by the matrix period, because $A \cdot x(t) = x(t+1)$ and in the next step it is $A \cdot x(t+1) = x(t+2)$, which is in fact $A^2 \cdot x(t) = x(t+2)$, and generally can be written as $A^k \cdot x(t) = x(t+k)$ or by the computation of the eigenvector, which is in fact the eigenvector of A^d where d is the period.

Both the knowledge of the development of the queues and the knowledge of the reaching of the stable states in such systems is strategically important for the system manager because this allows taking action flexibly, and thus positively affects the system.

Discrete event systems can be studied by several methods. Computer simulation is quite widely used to study such systems. One of the disadvantages of simulation is that to create an appropriate model with the corresponding details and functions that operate properly is a demanding task (but of course, not unachievable). However, it is without doubt that the model is in accordance with reality (with a high degree of similarity).

Another way to understand or describe the system is to make a mathematical model of it. Such models also allow the prediction of the system behavior, but in general, the mathematical analysis is more exact and allows an easy derivation of some properties of the system and the deduction of general conclusions. But it depends on the type of the system and also on the purpose of the study. To use a mixture of analytic and simulation methods thus seems to be advantageous.

1.2 Structure of the Thesis

Apart from the introductory chapter, this thesis consists of three main parts:

(1) A general introduction to the scientific topics and explanation of the theoretical fundamentals for the issues addressed in the thesis as well as a review of the research in related areas is given in Chapter 3.

(2) The main results of the research are introduced in Chapters 4 and 5. Chapter 4 concerns the study of the queues in discrete event systems (of closed and opened type) and tries to propose a solution of related issues: the control of the system's performance and cost optimization. Chapter 5 gives an overview of the results from the investigation of the steady states of discrete event systems.

(3) Lastly, the concluding chapters, Chapters 6 and 7, summarize the principal results of this research and outline possible directions for their further development.

2. Objectives of the Thesis

The main objective of this thesis is to solve some specific issues of system theory which arise in the operation of discrete event systems and to develop the tools appropriate for solving these problems.

To achieve the main objective, the following particular objectives have been specified:

(a) to describe the development of the queue lengths in a closed linear queuing system using max-plus algebra,

(b) to describe the development of the queue lengths in an open linear queuing system, and suggest methods which allow controlling the evolution of the system states with respect to the changes in the queue lengths, thus positively affecting the total costs of the system,

(c) to investigate the steady states of the discrete event systems using max-t fuzzy algebras and to contribute to the development of an algebraic tool in this direction,

- to investigate the steady states of systems where a maximal demand for the reliability of the system is required, which means to describe the matrix power periods in max-drast algebra,
- to investigate the states of systems where the consumption of given supplies is considered, which means to give a systematic characterization of the eigenspaces in a max-Łukasiewicz fuzzy algebra.

3. Theoretical Background and Literature Review

This chapter offers an explanation of the motivation and the key terms discussed in Chapters 4 and 5, which introduce the main results of the research; as well as providing a knowledge base and overview of the results in the literature concerning related issues.

3.1 Operations Research

Operations research is the youngest area in economics where mathematics is applied. It can be briefly characterized as a summary of relatively independent research methods using a systems approach to solve economic, organizational, technical, or other complex issues with the assistance of mathematical modeling or computational techniques for finding the optimal decision, i.e., the best decision in terms of the predetermined objectives [Jablonský, 2002].

Operations research can be characterized also as a tool for obtaining the optimal solution for a specific problem with respect to various limitations which affect the running of the system. Performing operations in a system is always dependent on (a) the limitations in disposable resources which are used during these operations, (b) the performance of other operations of the system process, (c) external aspects and other specific factors.

The methods of operations research include a set of instruments for managing complex economic systems at both the microeconomic and macroeconomic level. Operations research provides the opportunity to improve economic decisions, especially in terms of speed and quality, by using quantitative methods. This is the reason why mathematical modeling is an integral part of economic theory and practice. Mathematical modeling is based on the formulation of a simplified model of the real system. Figure 3.1 depicts the simplest modeling process, where something real was modeled with the use of the process of abstraction. The model is in most cases only an approximation of the true behavior of the system in reality, but when modeled accurately, the system and the model can be used interchangeably [Cassandras and Lafortune, 2008].

This approach brings also other benefits, such as,

- structuring the system and specifying all possible variations of the system states (of course, the number of such states can be unlimited),
- in most cases, the costs of experimenting with the model of the system are lower than the costs of experiments with the real system,
- the model of the system can be analyzed in a shorter time,
- the parameters of the system can be easily modified according to the desired experiments.

According to [Jablonský, 2002], the particular stages during problem solving in operations research can be specified as follows:

1. *problem recognition and definition*
   Crucial here is the ability of the executive officers to detect the issues, determine the need to select a particular approach and to put together a team of relevant experts.

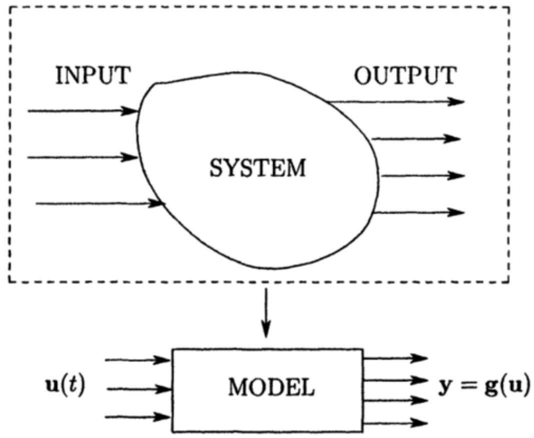


Figure 3.1: Simple modeling process [Cassandras and Lafortune, 2008]

2. *formulation of an economic model of the problem*

   In most cases the real problem is complicated and it is not possible to cover all of its aspects in the system model. Fortunately, this is not necessary. It is sufficient to cover the most substantive elements and relations, in view of the issues, and in accordance with this, to determine the following parts of the economic model:

   - the goal of the analysis determining the target state of the modeled system
   - a description of the processes in the system which affect the goal of the analysis
   - a description of the aspects influencing the processes
   - a description of the relations between the processes, aspects, and the goal of the analysis

3. *formulation of the mathematical model of the problem*

   This stage is about the formalization of the economic model, in other words, transferring the economic model into mathematical language. The mathematical model contains the same elements as the economic model, but it is formulated differently, mostly in linear or nonlinear functions of n variables. The processes are represented by variables, the aspects are represented by linear or nonlinear equations or inequalities, and the relations are represented by the parameters which can not be influenced by the user.

4. *solving the mathematical model*

   This phase has a rather technical character: the process of solving the mathematical model most often uses software tools, such as Matlab or MS Excel.

5. *interpretation of the results and their verification*

   The interpretation of the results, i.e., the explanation of the obtained outcomes, is very important. The meaning of verification lies in the verification of the model's correctness, in other words, in whether important elements in the model have not been forgotten.

6. *implementation of the results*

   The main goal is to implement the results as practices, and thus to improve the functioning of the investigated system.

Various types of issues can be solved by means of operations research and thus the following disciplines, which can be considered as the principal ones, can be distinguished: mathematical programming, multi-criteria decision making, graph theory, invertory management, queuing theory, recovery models, Markovian decision-making processes, game theory, and simulations. [Jablonský, 2002]

Because of the focus of the operations research on various practical applications and its wide scope, it overlaps with other disciplines, notably industrial engineering and operations management (see [Lawrence and Pasternack, 1997, Stevenson, 2014]), and draws on psychology and organization science.

3.1.1 Queuing systems

The knowledge basis and the terminology for this subchapter were borrowed from [Ng and Soong, 2008].

Queuing theory, as a discipline of operations research, defines a set of analytic techniques in the form of closed mathematical formulas to describe the properties of processes with a random demand and supply and a mass character. These processes are characterized by the emergence of *requests, customers* connected with some units that need the *service*: the implementation of some specific operations at a *service facility.* The service facility may contain more than one *server.* If an arriving request finds all of the servers occupied, it can wait in a queue. Such a request will receive service later, according to the service discipline; and after the service it leaves either the server or, if it was the last server in a sequence, it leaves the system. The terms 'customers' and 'servers' take various forms according to the type of the system.

Basically, a queuing system consists of three major components:

1. the input process,
2. the system structure,
3. the output process.

The input process, the arrival process, has three aspects: the size of the arriving population, the arrival patterns, and the behavior of the arriving customers. The size of the arriving population can be considered to be infinite in case the number of potential customers is very large compared to those in the system. An infinite population causes the queueing analysis to be more tractable, and hence allow of simple closed form solutions. On the other hand, assuming a queuing system with a finite customer population is more specific, because the arrival process is affected by the number of customers that are already in the system.

The arrival patterns vary, depending on whether the customers arrive in some regular pattern or totally at random. For customers arriving regularly, the arrival pattern can be easily described by single number, the rate of arrival. If the customers arrive randomly, then it is necessary to fit a statistical distribution to the arrival pattern in order to render the analysis of the system more mathematically feasible. The most common arrival pattern is the Poisson process, whose inter-arrival times are exponentially distributed: it describes very well a completely random arrival pattern and leads to very simple and elegant queuing results. Among other commonly used probability distributions there are (denoted by single letters):

- M: Markovian, implying a Poisson process
- D: Deterministic, with constant inter-arrival times
- E_k: an Erlang distribution of order K for the inter-arrival times
- G: A general probability distribution of inter-arrival times
- GI: A general and independent (inter-arrival times) distribution.

Customers arriving to the system can behave differently according to the conditions during their arrival. It can happen that the system is full (due to the finite waiting queues) or all servers can be busy. It can happen that the system is full and the just arrived customer leaves forever without entering the system: such systems are called blocking systems.

The second component of a queuing system, the system structure, can be characterized by two attributes. The first attribute is the physical number and layout of the servers. The service facility can contain more than one server and these servers can be concatenated in series and/or in paralel. The second attribute is the system capacity, which refers to the maximal number of customers that can be found in the system. In a multi-server queuing system, this is the sum of the maximum sizes of the waiting queues and the number of servers. Naturally a waiting queue can have some maximal capacity, although it is much easier to analyze queuing systems with infinite capacities where if the server is busy, the customer simply joins the waiting queue.

The output process at the last component of a queuing system can be characterized by two aspects of the service behavior that influence the departure process: the queueing (serving) discipline and the service time distribution. The queuing discipline means the way waiting customers are selected for service. In general, five common types can be distinguished:

- First-come-first-served - FCFS
- Last-come-first-served - LCFS
- Priority
- Processor sharing
- Random

The most frequent way is the first one, FCFS. This discipline does not assign priorities: the customers are served in the order of their arrival. The LCFS discipline is commonly found in stack operations, where items are stacked and the available operations are at the top of the stack. The priority discipline suppose a division of the customers into several priority classes. Intuitively, those that are located in the class with higher priority are served before the others. This discipline can be subclassified into the so-called preemptive and the non-preemptive regimes. Processor sharing means that the capacity of the server is divided among all the customers in the waiting queue.

The service time distribution is similar to the arrival patterns. It concerns the amount of service time that is required by customers. If this amount is the same for all, then the service time distribution can be described by a single number. But it can be observed that in general not all customers need the same time of service. Thus there is a probability distribution describing the length of service times the server renders to the customers. Again, the most common distributions can be mentioned:

- M: Markovian (or memoryless), implying an exponential distribution
- D: Deterministic, constant service times
- E_k: Erlang distribution of order K
- G: General distribution.

Many processes in the system interact with each other, and most of the quantities associated with these processes are of a probabilistic nature and evolve in time. The values of such variables can be only expressed through a probability. The primary random variables of a queuing system can be seen in Figure 3.2.

When speaking about a queueing system, the Kendall notation is also worth mentioning. David G. Kendall, a British statistician, devised a shorthand notation that helps to categorize and describe such complex systems in a short form. The Kendall notation is of the form

$$A/B/X/Y/Z$$

where A is the customer arrival pattern (inter-arrival-time distribution); B is a service pattern (service-time distribution); X is the number of parallel servers; Y is the system capacity; and Z is the queuing discipline. The last two parameters, Y, Z, can be omitted if $Y = \infty$ and $Z =$ FCFS.

For example, the notation $D/D/1/\infty/\text{FCFS}$ represents a queuing system where the inter-arrival time of the customers is constant, and so are the service times of

Notation	Description
$N(t)$	The number of customers in the system at time t
$N_q(t)$	The number of customers in the waiting queue at time t
$N_s(t)$	The number of customers in the service facility at time t
N	The average number of customers in the system
N_q	The average number of customers in the waiting queue
N_s	The average number of customers in the service facility
T_k	The time spent in the system by kth customer
W_k	The time spent in the waiting queue by kth customer
x_k	The service time of kth customer
T	The average time spent in the system by a customer
W	The average time spent in the waiting queue by a customer
$\bar{x}$	The average service time
$P_k(t)$	The probability of having k customers in the system at time t
P_k	The stationary probability of having k customers in the system

Figure 3.2: Random variables of a queuing system, [Ng and Soong, 2008]

the servers. The system has only the one parallel server and the capacity of the queue in front of this server is considered to be infinite. The customers entering the system are served according the FCFS discipline. Similar systems are investigated in the practical part of this thesis, with different numbers of servers, more servers concatenated in series, as well as both closed and open types of systems.

A very simple and interesting result should be mentioned in connection with queuing systems: Little's theorem. In the literature, it can also be found under equivalent names, such as Little's law, result, lemma, or formula. The theorem was not proven for a long time: it existed only as an empirical rule. In 1961, Little proved it formally. The theorem refers to the average number of customers in a steady state of the system, denoted by N:

$$N = \lambda T$$

where λ is the average arrival rate of customers entering the system and T is the average time spent by a customer in the system. The attractiveness of this theorem lies in the fact that the average number of customers is not influenced by the inter-arrival-time distribution, the service-time distribution, the layout of the servers, or practically anything else.

3.2 Discrete Event Systems

The methods of operations research can be applied to various system models and these models can be also classified according to various criteria.

Although the concept *system* is understood intuitively, we can begin with its definition, which can be found in the literature:

- "System is a regularly interacting or interdependent group of items forming a unified whole." (*Merriam-Webster Dictionary*)
- "System is an aggregation or assemblage of things so combined by nature or man as to form an integral or complex whole." (*Encyclopedia Americana*)
- "System is a combination of components that act together to perform a function not possible with any of the individual parts." (*IEEE Standard Dictionary of Electrical and Electronic Terms*)

It is worth emphasizing that a system contains its components and is closely connected with its function as a result of the components' interactions. When modeling a system, one is not limited to systems with physical objects, for example, manufacturing systems or computer networks. Systems studying economic mechanisms or human behavior can also be modeled.

The system approach to modeling distinguishes itself from the more traditional analytic approach by emphasizing the interactions and connectedness of its components. In other words, every cause is indirectly connected with its consequence but also with the other causes in the system. The system's behavior is the way the system reacts to the inputs coming into it. [Bureš, 2011]

There can be found various classifications of systems on a general level in the literature. Let's mention some of them [Bureš, 2011]:

- according to its relation to reality: Real or Abstract

  In real systems, all system components are real objects whereas in abstract systems, the components have a non-material character.

- according to its origin: Natural or Artificial

  Artificial systems are created as a result of people's conscious initiative.

- according to its relation to its environment: Open or Closed

  In open systems we are interested also in capturing the relations between the system and its environment. If not, it is considered a closed system.

A system can also be described in different ways, depending on the goal of the system analysis. A system can be described: verbally, by a list of its basic components; with the use of block schemes; using methods from graph theory; or by a matrix.

The systems working in discrete time, the so-called discrete event systems (DES), are the subject of an emerging discipline in systems and control theory. The basic fundamentals can be found in [Cassandras and Lafortune, 2008, Zeigler et al., 2000]. These systems are usually man-made and have a complex hierarchical structure. Such systems can be met with in real life, for example, in the production of a product on an assembly line, in patients passing through the sections of a polyclinic, in processing information with the assistance of computational techniques, in the transit of vehicles through a road network, in serving customers in a department store, or in a washing line.

According to [Cassandras and Lafortune, 2008], discrete event systems can also be classified into the following categories:

- according to their behavior in time: **Static or Dynamic**

  The states of a static system do not change in time, whereas the states of a dynamic one change over time. It can be said that a static system is memoryless, but a dynamic system is a memory system. Or, it can be said that for a static system it holds that the output is independent of the past values of the input. On the other hand, the output of a dynamic system depends on the past values of the input. Thus, determining the output of a static system requires no memory of the input history, which is not the case for a dynamic system.

- according to the type of the system's quantities: **Continuous or Discrete**

  In continuous systems, the states changes continuously in all components, the state space is a continuum consisting of all n-dimensional vectors of real numbers. In discrete systems, the state space is a discrete set and the state variables are only permitted to move at discrete points in time from one discrete state to another. It is often simpler to visualize the dynamic behavior of discrete state systems. However, to formally express the state equations and solve them may be considerably more complex than for a continuous system.

- according to the character of its mathematical description: **Linear or Nonlinear**

  In linear systems, all elements of the mathematical description of the system have the character of a linear operation; it satisfies superposition and homogenate principles. The principle of superposition is described by the following property: if a stimulus S_1 produces a response R_1, and the stimulus S_2 produces a response R_2, then the superposition of the two stimuli, $(S_1 + S_2)$, produces the superposition of the responses $(R_1 + R_2)$. Observe that if $S_1 = S_2$, doubling the input leads to doubling the output. If at least one operation has a nonlinear character, the system is considered to be non-linear.

- according to the type of input–output relationship: **Time varying or Time invariant**

  The system is said to be time invariant if the following property holds: if an input to the system at time t, $u(t)$, results in the output $y(t)$, then the input $u(t+k)$ results in the output $y(t+k)$ for any k. On the other hand, the output of a time-varying system is not always the same when the same input is applied.

- according to the ability to predict the future behavior of the system: **Deterministic or Stochastic**

  A system is stochastic if at least one of its input variables has a random character. Otherwise, the system is said to be deterministic. The state of a stochastic system is defined by a random process, whose behavior can be described only probabilistically. The state at time t is a random vector and it is only its probability distribution function that can be evaluated.

- according to the possibility of using feedback: **Open-loop or Closed-loop**

  In open-loop control, the input remains fixed regardless of the effect that it has on the observed output: the effect can be either good or bad. For closed-loop systems, the input depends on the effect it causes on the output.

What does the word *event* in *discrete event system* mean? The states of a discrete system can be described by a discrete set, and the state transitions are observed at discrete points in time. If these points are connected with some events, then the system is called a *discrete event system*. In other words, the event occurs instantaneously and causes the transition of the system from one state to another (such a system is also called "event driven"). [Cassandras and Lafortune, 2008]

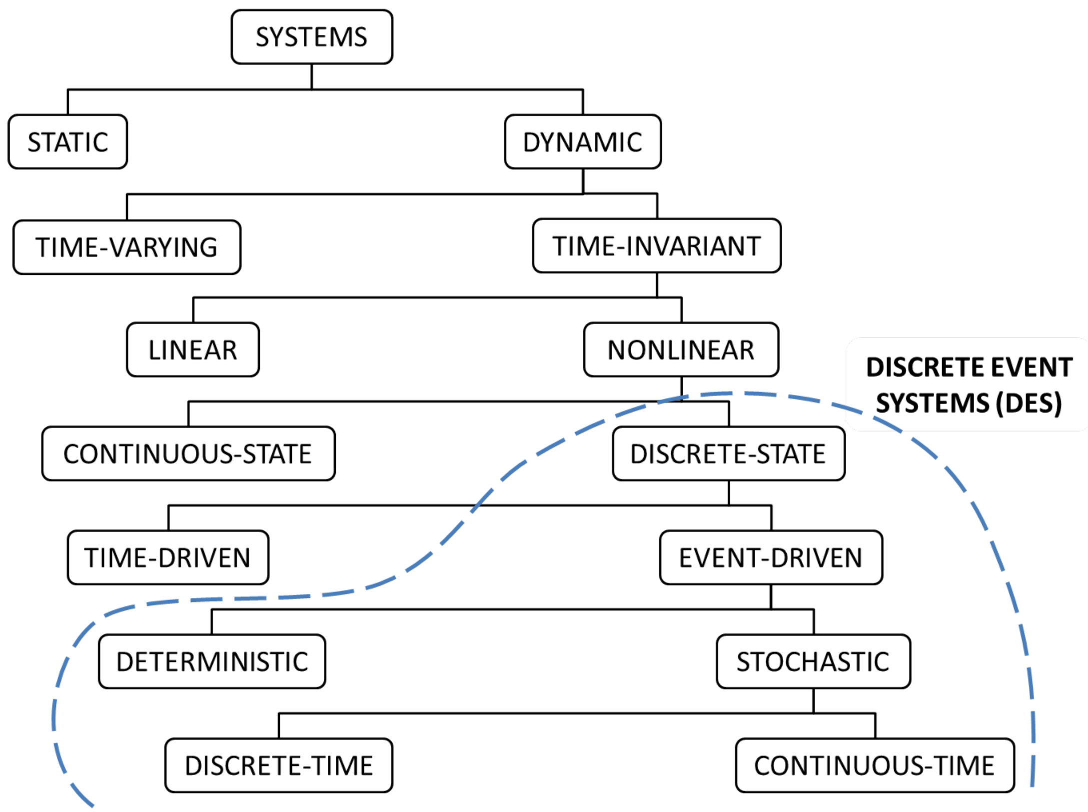


Figure 3.3: The major categories of system classification, [Cassandras and Lafortune, 2008]

On the other hand, the so-called *discrete time systems* can be distinguished, where the states generally change with the passage of time (because the passage of time is the cause of the change, the system is also called "time driven"). In other words, with every "clock tick" the state is expected to change. To complete the idea, continuous state systems are considered as time-driven by their very nature. But a discrete state system can be of either type, depending on whether the state transitions are synchronized by a clock or occur asynchronously. The major categories to classify systems are shown in Figure 3.3. [Cassandras and Lafortune, 2008]

3.2.1 Graph Theory and Matrix Approach

The word “graph” has two meanings in the language of mathematics: the first is a plot representing the values of a function, and the second, used in the following, is a collection of points and lines representing some structure.

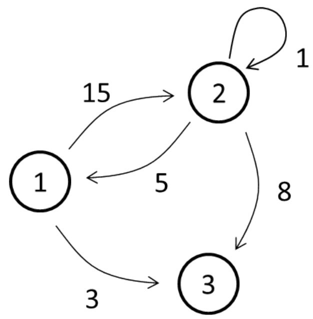
Graph theory is a logical and systematic approach and is very well reviewed in [Bondy and Murty, 1976, Tutte, 1984, Gross and Yellen, 2004]. Various kinds of systems and problems in numerous fields of science and technology can be analyzed by means of graph models. If the system is complex, then the graph is complex too and thus a visual analysis is impracticable. For these cases, the system can be studied with the use of the matrix representation of the graph and analyzed with computer support.

Let us denote the graph by $G = (V, E)$, where $V = \{v_1, v_2, \ldots\}$ is the set of nodes (or vertices) and $E = \{e_1, e_2, \ldots\}$ is the set of edges between the vertices. The so-called end vertices of an edge e_k are the vertices v_i and v_j associated with this edge: $e_k = (v_i, v_j)$. The edges in the graph can be directed and even weighted (see Fig. 3.4). A graph with directed edges is often called a digraph.

The weights in a graph can have various meanings, depending on the purpose of the study of the system. For example, it can be the cost (or a balance of given resources) of using the path from node i to j, or the time to get from node i to j, or the relative importance of the node i compared to that of node j (then the directed edge is drawn from node i to j).

To any $n \times n$ matrix A a graph can be associated, called the communication graph of A, denoted by $G(A)$. Representations by graphs and matrices of systems are used throughout the present thesis.

A description of the basic terms from graph theory used in this thesis follows [Heidergott et al., 2006].



$$A = \begin{pmatrix} 0 & 15 & 3 \\ 5 & 1 & 8 \\ 0 & 0 & 0 \end{pmatrix}$$

Figure 3.4: Weighted digraph: Vertices are represented by circles, edges by segments. On the right, there is the corresponding matrix.

Loop is an edge of the form (v_i, v_i).

Walk is a finite sequence of the form $(v_{i1}, e_{j1}, v_{i2}, \ldots, e_{jk}, v_{ik})$. The walk starts at a vertex and contains neighbour elements. The length of a walk is equal to k, the number of edges.

Trail is a walk where any edge is traversed at most once.

Path is a trail where any vertex is visited at most once (except the initial and terminal ones).

Weight of the path will be denoted by $w(p)$ and is defined by the sum of the weights of all edges constituting the path.

Length of the path will be denoted by $l(p)$ and is the number of edges belonging to the path.

Circuit, cycle is a closed path, i.e., the initial and terminal vertices are the same.

Tree is a graph that does not contain any cycles.

Null graph is a graph where V and E are empty sets, in other words, it is a graph with no vertices.

Trivial graph is represented by one single isolated vertex.

Complete graph contains every possible edge between all its vertices.

Connected graph is a graph where all vertices are connected to each other. A trivial graph is connected by convention.

Strongly connected graph is a graph where all nodes are reachable from all other nodes in the graph, i.e., there exist a path between every pair of nodes. Notice that for directed graphs, there can exist an edge between two nodes, but the reachability need not be fulfilled for both directions (then the graph is not considered to be strongly connected).

Irreducible matrix is a matrix of a strongly connected graph. If the matrix is not irreducible, it is called *reducible*.

Component of a graph is a subgraph (not a null graph) whose vertices are a subset of the vertex set and edges are a subset of the edges set.

Graph theory is a powerful tool for the description and investigation of a system structure, and provides many methods that can be found in the literature. To mention a few: solving the Hamiltonian path problem, the shortest path problem, finding the minimum spanning tree, the traveling salesman problem, and others.

3.3 Extremal Algebras

Extremal algebras are used mainly for describing and studying systems working in discrete time. During the operation of such a system, a stable state can arise. The system is described by a transition matrix and the eigenvectors of such a matrix represent the stable states of the system.

With the use of extremal algebras, we are able to get various answers about system characteristics. Some examples of these questions follow.

- What is the maximum speed at which the system could run?
- What is the latest time to start so that the production meets the delivery dates?
- If all machines start working at time zero, at what time will each machine finish work on the project?
- Can the system repeatedly reach some sequence of states?
- How great a delay and to which events can be tolerated without prejudice to the calculated finishing time of the project?
- How to choose a sequence of nodes when travelling through some network, so that we visit as many nodes as possible before the given supplies are exhausted?

The term *extremal algebra* was used for the first time by [Vorobjov, 1967]. This concept occurs under different names in the literature, for example, tropical algebra, max-algebra, or fuzzy algebra. It is a special algebraic structure equipped with two operations $\oplus$ and $\otimes$. The operation $\oplus$ is not addition as in usual algebra, but is one of the operations of extremum: maximum or minimum. The operation of $\otimes$ is a chosen group or semigroup operation so that these two operations can be commutative and associative. The distributive law holds, and there exist neutral elements with respect to both operations [Zimmermann, 2003]. For example, if one chooses $\oplus = \max$ and $\otimes = \min$, one speaks of a max-min algebra; if one chooses $\oplus = \max$ and $\otimes = +$, one speaks of max-plus algebra.

The operations $\oplus$, $\otimes$ are then extended in a natural way to Cartesian products of vectors and matrices. The detailed mathematical investigation at a general level can be found in [Cuninghame-Green, 1979].

In the following, an overview of the main results in extremal algebras is given. Max-plus algebra is introduced in more detail with those basic notions and definitions that are crucial for understanding the context for the following parts of this thesis.

3.3.1 Applications of Extremal Algebras

The max-plus algebra is a modern modeling and solution tool for many economic problems that arise from manufacturing, transportation, the allocation of resources, and information processing technology. It has been studied from the early 1960's. Perhaps the first paper was by Cuninghame-Green [Cuninghame-Green, 1960], which was followed by [Cuninghame-Green, 1962, Cuninghame-Green, 1979, Cuninghame-Green, 1991].

The main advantage of using max-plus algebra for the description of discrete event systems lies in linearizing the model (using conventional linear algebra to model a DES leads to a non-linear description), see [Cuninghame-Green, 1979, Heidergott et al., 2006]. An interesting aspect from the point of queueing theory is that max-plus linear queueing networks allow fork and join nodes. This has resulted in successful applications of max-plus algebra in many areas, including manufacturing system modeling [Ren et al., 2007, Seleim and ElMaraghy, 2014], multi-criteria decision making using the method AHP [Gursoy et al., 2013], scheduling [Bouquard et al., 2006], performance evaluation [Cohen et al., 1985, Reddy et al., 2009], performance optimization [Febbraro et al., 1994], and control [Schutter and van den Boom, 2001, Houssin et al., 2007].

In the last decades, intensive research has been conducted on applying deterministic max-plus algebra to the design of timetables for transportation networks, especially for railways. These networks operate according to a timetable where the technical running times of the trains on the tracks, the routes of the trains, and information about the planned connections between trains are given. The aim is to design a stable timetable with the (minimal) asymptotic period length so that all train events (arrivals, departures, passages at stations) can occur at regular intervals.

In [Goverde, 2007] the analysis of the Dutch national railway network can be found. In a given timetable, one single delay can cause an avalanche of delays of connected lines. This is the main issue for timetable planners to solve: should the train wait on a connecting train which has been delayed? The paper describes a sensitivity analysis of timetables with regard to robustness to delays, based on a description of a railway timetable in max-plus algebra. The system's stability is also investigated, i.e., the self-regulatory behavior of the railway system to return to a steady state of the railway timetable after disruptions. An application to the Metro-bus public transport system in Mexico City can be found in [Konigsberg, 2008]. Other studies of transportation networks can be found in [Braker, 1993, Heidergott and de Vries, 2001, Goverde et al., 2011].

Also worth mentioning is the computer application PETER (Performance Evaluation of Timed Events in Railways) [Goverde and Odijk, 2002], based on efficient numerical algorithms. The application enables real-time analysis of large regular timetables and was developed as a decision support tool for timetable planners, so that they can improve the efficiency of timetables and the management of delays.

Max-plus algebra is also used in the field of biology. The dynamics of the translation of the cellular protein production, in which ribosomes move in one direction along the mRNA strand, is described by using a time event graph in [Brackley et al., 2012]. The use of max-plus algebra in dynamic programming in biology can be found in [Comet, 2003].

The description of a DES using one given transition matrix describing the system's characteristics might not be convenient in all cases. The entries of the matrix can represent, for example, the time variable that can vary from case to case, which is the basic idea of interval matrices. The system is then described by two matrices: the first specifies the lower bound and the second specifies the upper bound of specific entries. The study of interval matrices brings a new perspective on the system and from the theoretical point of view, it offers new challenges in discovering connections. Important contributions in this area include the study of its eigenvectors [Cechlárová, 2005], eigenspaces [Cechlárová, 2001, Cechlárová and Cuninghame-Green, 2002, Gavalec et al., 2011], matrix robustness [Myšková and Plavka, 2013, Myšková and Plavka, 2014, Plavka, 2014], and solvability of equations [Myšková, 2012, Myšková, 2016, Rohn, 1989].

The periodic behavior of a DES indicates its steady state. The steady states of the system can be found and described in max-plus algebra by finding the periods of the powers of the matrix [Butkovič and Cuninghame-Green, 2007, Gavalec, 2000b, Sergeev, 2009, Molnárová, 2005] or by finding the (derived) vector periods (i.e., the system states) generated by the matrix. Vector periods are also known in the literature as "orbits."

No less important are the research results about other extremal algebras. Vector periods in max-min algebra were studied in [Gavalec, 2000a, Semančíková, 2007, Semančíková, 2006]. Matrix periods in max-min and max-prod algebras have been studied in [Liu et al., 2016]. The characterization of the interval eigenvectors in max-min algebra is given in [Gavalec and Plavka, 2010, Gavalec et al., 2014].

3.3.2 Max-plus Algebra

The basic definitions and notations can be found, for example, in [Heidergott et al., 2006]. The max-plus algebra is denoted by $\mathcal{R}_{\max} = (\overline{\mathbb{R}}, \oplus, \otimes, \varepsilon, e)$, where $\overline{\mathbb{R}}$ is the set of real numbers extended by the infinite value $\varepsilon = -\infty$ and the zero element $e = 0$. The binary operations $\oplus$, $\otimes$ are defined on $\overline{\mathbb{R}}$ by

$$a \oplus b = \max(a, b), \tag{3.3.1}$$

$$a \otimes b = a + b. \tag{3.3.2}$$

These operations are extended to matrices and vectors in a formal way. The operation $\otimes$ has priority over the operation $\oplus$, for example,

$$2 \otimes 1 \oplus 4 = (2 \otimes 1) \oplus 4 = 4.$$

These operations can be extended to the infinite value ε: $a \oplus \varepsilon = \varepsilon \oplus a = a$ and $a \otimes \varepsilon = \varepsilon \otimes a = \varepsilon$. The powers in max-plus algebra are defined by

$$x^{\otimes n} = x \otimes x \otimes \cdots \otimes x = x + x + \cdots + x = n \times x. \tag{3.3.3}$$

A list of the algebraic properties of max-plus algebras follows. [Heidergott et al., 2006]:

- Associativity

$$\forall x, y, z \in \mathcal{R}_{\max} : x \oplus (y \oplus z) = (x \oplus y) \oplus z$$

$$\forall x, y, z \in \mathcal{R}_{\max} : x \otimes (y \otimes z) = (x \otimes y) \otimes z$$

- Commutativity

$$\forall x, y \in \mathcal{R}_{\max} : x \oplus y = y \oplus x \quad \text{and} \quad x \otimes y = y \otimes x$$

- Distributivity of $\otimes$ over $\oplus$

$$\forall x, y, z \in \mathcal{R}_{\max} : x \otimes (y \oplus z) = (x \otimes y) \oplus (x \otimes z)$$

- Existence of zero element

$$\forall x \in \mathcal{R}_{\max} : x \oplus \varepsilon = \varepsilon \oplus x = x$$

- Existence of unit element

$$\forall x \in \mathcal{R}_{\max} : x \otimes e = e \otimes x = x$$

- The zero is absorbing for $\otimes$

$$\forall x \in \mathcal{R}_{\max} : x \otimes \varepsilon = \varepsilon \otimes x = \varepsilon$$

- Idempotency of $\oplus$

$$\forall x \in \mathcal{R}_{\max} : x \oplus x = x$$

To demonstrate the practical usefulness of extremal algebras, the following motivating example with the use of a max-plus algebra will be given. A similar example is described in [Heidergott et al., 2006]. For instance, a simple railroad between two cities can be considered (see Fig. 3.5). The cities, say S_1 and S_2, are connected by two tracks. The train running from S_1 to S_2 has a travel time of 5 time units, but in the opposite direction, the journey from S_2 to S_1 takes 6 time units. It can be easily seen that these two tracks form a circuit. Travelling from S_1, visiting S_2, and coming back to S_1 then takes 11 time units. There exist two more tracks in the system. These tracks connect the suburbs of each of the cities with its main station. The trip along the suburbs of the cities S_1, S_2 takes 2, respectively 3 time units. These times can be

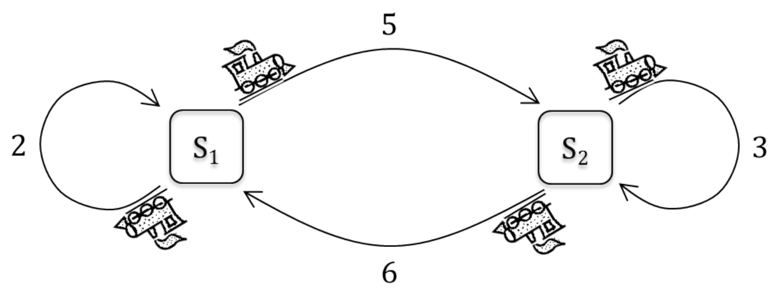


Figure 3.5: The railway network, adapted from [Heidergott et al., 2006]

rewritten into a matrix A where the entry a_{ij} denotes the travel time from the city S_j to S_i.

$$A = \begin{pmatrix} 2 & 6 \\ 5 & 3 \end{pmatrix} \tag{3.3.4}$$

There are four trains in the system: one train at each track. The task is to design a timetable for which: the travel times of the trains are given, the frequency of the trains is as high as possible, the frequency must be the same along all four tracks, yielding a timetable with regular departure times, the trains arriving into the station should wait for each other (due to the changeover of the passengers) and each train departs as soon as possible. The departure time of two trains at the station S_1 will be denoted by x_1. The departure time for these two trains are the same because of the changeover of the passengers between the trains. Similarly, the departure time for two trains at the station S_2 will be denoted by x_2. Then these two departure times can be written as a vector $x \in \mathbb{R}^2$. The first departure time will be given by $x(0)$. The $(k+1)$st departure times, the vector $x(k+1)$, are given by the vector $x(k-1)$:

$$\begin{aligned} x_1(k+1) &>= x_1(k) + a_{11} + \delta \\ x_1(k+1) &>= x_2(k) + a_{12} + \delta \end{aligned} \tag{3.3.5}$$

where δ denotes the time reserved for the change of the passengers from one train to the other. Or, the parameter δ can be considered as a part of the travel time. Then the above idea simplifies to

$$x_1(k+1) = \max(x_1(k) + 2, x_2(k) + 6). \tag{3.3.6}$$

Similarly, for the second departure time,

$$x_2(k+1) = \max(x_1(k) + 5, x_2(k) + 3). \tag{3.3.7}$$

This equation can be rewritten more compactly as

$$x_i(k+1) = \max_{j=1,2,\ldots,n}(a_{ij} + x_{ij}(k)), \qquad i = 1, 2, \ldots, n. \tag{3.3.8}$$

The operators of classical algebra can be replaced by the operators of max-plus algebra: the operator of maximization will be replaced by addition, $\bigoplus$ (pronounced in this case as "big o plus"), and the operator of addition will be replaced by the operator of multiplication, $\otimes$ (pronounced as "o times"). The (3.3.8) can be then rewritten as

$$x_i(k+1) = \bigoplus_{j=1}^{n}(a_{ij} \otimes x_{ij}(k)), \qquad i = 1, 2, \ldots, n, \tag{3.3.9}$$

and the same equation can be rewritten in vector form, which indicates that this process is not described in conventional linear algebra, max-plus algebra is used:

$$x(k+1) = A \otimes x(k). \tag{3.3.10}$$

For $k = 1$ we get $x(1) = A \otimes x(0)$, and for $k = 2$ we obtain $x(2) = A \otimes x(1) = A \otimes (A \otimes x(0)) = (A \otimes A) \otimes x(0) = A^{\otimes 2} \otimes x(0)$. The symbol $\otimes$ in the exponent means the power of matrix A in max-plus algebra. This idea can be also generalized by the formula $x(k) = A^{\otimes k} \otimes x(0)$.

Remark 3.3.1. The communication graph, $G(A)$, and the powers of the matrix, $A^{\otimes k}$, are closely related in max-plus algebra. It can be shown that the element $[A^{\otimes k}]_{ji}$ is the maximal weight of the path from i to j of length k [Heidergott et al., 2006].

This is a simple example of system equations that are not linear in conventional algebra, but are linear in max-plus algebra. The reader can easily see the connection with the initial departure time: the kth departure time can be simply computed by using the given initial departure time and the power of the transition matrix, which can be calculated directly. For example, the departure times for the initial departure time $x_1(0) = x_2(0) = 0$ are

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 6 \\ 5 \end{pmatrix}, \begin{pmatrix} 11 \\ 11 \end{pmatrix}, \begin{pmatrix} 17 \\ 16 \end{pmatrix}, \begin{pmatrix} 22 \\ 22 \end{pmatrix} \dots. \tag{3.3.11}$$

Moreover, if the initial departure time is an eigenvector (the eigenvectors and eigenvalues will be briefly introduced in the following chapter), the obtained timetable will be regular (unlike the timetable (3.3.11)). In other words, the eigenvector represents a realizable initial timetable allowing the periods with minimal cycle time.

In reality, delays can appear in such a system. And once the timetable is given, one can be interested in its sensitivity to possible disruptions of the schedule. An algorithm to compute the propagation of initial delays over a periodic railway timetable is presented in [Goverde, 2010].

The Eigenproblem in Max-plus Algebra

Analogously to its use in linear algebra, an important role in extremal algebras is played by independence of vectors, regularity of matrices, solvability and unique solvability of systems of linear equations, and finding eigenvectors and eigenvalues. With the efficient algorithms for finding the eigenvalues and eigenvectors of a square matrix presented in [Butkovič, 2010], it is possible to obtain these values in $O(n^3)$ time. An overview of results in this area is given also in [Zimmermann, 1976, Zimmermann, 1981].

The eigenproblem for a given square matrix A consists in finding a non-trivial solution $x \neq (\varepsilon, \ldots, \varepsilon)$ to the equation

$$A \otimes x = \lambda \otimes x \tag{3.3.12}$$

where λ is the eigenvalue and x is the eigenvector of A. In queuing systems, the eigenvector indicates how the system behaves if the time evolution has settled down. The eigenvalue indicates the difference between two following states (vectors) in the settled time evolution. [Heidergott et al., 2006]

Note that a square matrix can have more than one eigenvalue. The eigenvectors are not necessarily unique, because if x is an eigenvector, then so is $c \otimes x$, where c is any finite number. The set of all eigenvectors associated with an eigenvalue is called its eigenspace (a vector space) [Heidergott et al., 2006].

Because the inverse operation to the operation of maximum does not exist, it was necessary to develop methods that help solve equations and inequalities, the eigenvector problem, and so on.

The maximum cycle mean plays a key role in max-plus algebra. The following celebrated theorem was first published and proved by Cuninghame-Green [Cuninghame-Green, 1962]:

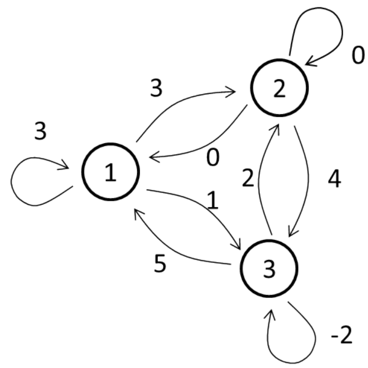
Theorem 3.3.1. *Let $A \in R(n, n)$ be an irreducible matrix. Then this matrix has one and only one eigenvalue, $\lambda(A)$, which is equal to the maximum cycle mean of the associated digraph A, i.e.,*

$$\lambda(A) = \max \left\{ \frac{w(\rho)}{l(\rho)}; \rho \textit{ is a cycle in } G(A) \right.$$

where for a cycle $\rho = (i_1, i_2, \ldots, i_k)$ its weight is given by $w(\rho) = a_{i_1,i_2} \otimes a_{i_2,i_3} \otimes \cdots \otimes a_{i_k,i_1}$ and its length by $l(\rho) = k$.

Example 3.3.1. *Compute The Maximum cycle mean for a given A, see Figure 3.6.*

The cycle means of the elementary cycles of length 1 are $0, 3, -2$; of length 2, they are $32, 3, 3$, and for length 3, they are $4, 1$. The maximal cycle mean, λ, is then equal



$$A = \begin{pmatrix} 3 & 3 & 1 \\ 0 & 0 & 4 \\ 5 & 2 & -2 \end{pmatrix}$$

Figure 3.6: Weighted digraph and corresponding matrix.

to 4*. The eigenvalue stays unchanged if the maximization is taken over all cycles (not only the elementary ones).*

According to [Heidergott et al., 2006], a cycle p in $G(A)$ whose average weight is maximal is called *critical*. The *critical graph* of A, denoted by $G^c(A) = (V^c(A), E^c(A))$, is the graph consisting only of those nodes and edges that belong to a critical cycle. The nodes are sometimes referred to as *critical nodes*. The critical graph for the previous example (Figure 3.6) contains the vertices $1, 2, 3$ and the cycle between them, see Figure 3.7. Note that the critical graph of an irreducible matrix need not be strongly connected.

To compute the maximal cycle mean from its definition is quite easy for small matrices, but for complex systems it is difficult, because it can be very complicated to go through all the elementary cycles. And this was the stimulus for searching for some algorithm. According to [Butkovič, 2010], one of the first methods was Vorobyov's $O(n^4)$ formula, which was followed by Lawlers's $O(n^3 \log n)$ and Karps's algorithm (also presented in [Butkovič, 2010]). Karps's algorithm finds the lambda

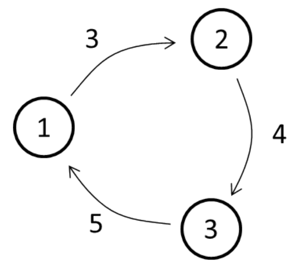


Figure 3.7: The critical graph of A.

for the matrix $n \times n$ in $O(n|E|)$ time where E is the set of edges of $G(A)$.

Max-plus Model of the Queuing Network and its Optimization

Queuing network modeling has been investigated by many researchers. A general overview of modeling the production and transfer lines using queuing networks can be found in [Papadopoulos and Heavey, 1996]. The stochastic approach in queuing theory was introduced in [Cinlar, 2013, Buzacott and Shanthikumar, 1993, Tempelmeier, 2003]; [Le Boudec and Thiran, 2001] provides a deep insight into flow problems in networking using a deterministic approach.

Heidergott's paper [Heidergott, 1998] was a strong motivation to follow his study of queuing systems. Using stochastic methods, he studied closed and open tandem systems (i.e. single server queues connected in a series) with several modifications. The description of max-plus transition matrices is given and shown in examples. In chapter 4.1, a description of the deterministic types of closed queuing systems (with the use of max-plus algebra) is given and supplemented with the computation of the queue lengths. The following chapter, chapter 4.2, is an extension of the research to the queue lengths for open linear systems. In that chapter, methods for the optimization of the system's costs are suggested.

The stochastic approach to the optimal control of a system is also used in [Miller, 2009]. The optimization goal there is to minimize the cost function, which takes into account the average queue length (related with the average time of service, or/and the price of rejected requests).

In [Hao and Yifei, 2011] can be found a simulation of the queuing processes in a bank together with suggestions for optimization .

The optimization of a fuzzy finite capacity queuing model in which the arrival pattern and the service pattern follow an exponential distribution under an uncertain parameter is presented in [Pardo and de la Fuente, 2008]. The optimization is based on the selection of the optimal number of servers with the goal of providing a high degree of satisfaction to customers of the system. The optimization process is solved through Markov chains with fuzzy states.

Studying queuing models using fuzzy environments leads to a wider knowledge and more informative and accurate results about the behavior of the system. That is why fuzzy models can have a broader range of applications.

3.3.3 Max-t Fuzzy Algebras

The discrete time systems studied with the use of max-t fuzzy algebras can be sometimes intuitively called fuzzy systems. But this can be confusing in some ways because the whole study of the system in this way is not about studying "classical" fuzzy systems, which means constructing the function of membership, fuzzification, inference and defuzzification. The main focus lies in the description of the development of the system's states in time. Both the states described by vectors and the transition matrix have fuzzy numbers as entries. And this may be the reason why these systems are sometimes called fuzzy. Naturally, the terminology is continually developing. For the transition from one state to another, various operations for the conjunction of the sets are used. And these operations are also much used for studying "classical" fuzzy systems.

Fuzzy logic appeared as a necessary extension of Zadeh's theory of fuzzy sets [Zadeh, 1965]. The development of fuzzy logic over the last two decades has been driven by the applications in artificial intelligence and decision making processes.

The standard membership degrees of fuzzy sets (values from the unit interval), can be understood in a natural way as the truth degrees of an infinite valued logic, with $[0, 1]$ as its truth degree set.

According to [Iliadis et al., 2008], fuzzy algebra claims that each object, say x, of the universe, denoted by X, belongs to a fuzzy set A with respect to a characteristic real number $\mu_A(x)$ called the degree of membership. In other words, it is the truth degree of the formula $x \in A$.

With this understanding, the intersection of the fuzzy sets A and B, i.e., $A \cap B$, is $\mu_{A\cap B}(x)$, and this corresponds to the truth degree of the statement $x \in A$ AND $x \in B$. The word AND has the meaning of a conjunction connective. It can be claimed that the theory of crisp sets becomes presented in the language of classical logic. Already the first paper of Zadeh [Zadeh, 1965] offers different proposals for understanding the above mentioned operation of conjunction, AND, as taking the minimum (or as taking a usual, algebraic product). Similarly, for the operation of union $A \cup B$, Zadeh suggests taking the maximum of the membership degrees in A, B (or their algebraic sum). These relationships have been widely used in theoretical considerations about the foundations of fuzzy set theory. In particular, in the last decade, there has been initiated an enormous development in the field of fuzzy logic in the narrow sense. [Gottwald, 2005]

Max-t fuzzy algebra is defined over the interval $[0, 1]$, and, as the name suggests, has two binary operations: the operation of maximum (denoted by $\oplus$) and one of the t-operations, (denoted by $\otimes$) are used. These operations are extended to vectors and matrices in a formal way. According to the chosen t-operation, we can then talk, for example, about max-Łukasiewicz, max-min, max-prod or max-drast fuzzy algebra.

Triangular norms

Triangular norms (briefly t-norms) were first introduced by Karl Menger in 1942 in his paper [Menger, 1942]. The main idea was to construct metric spaces where probability distributions (instead of numbers) describe the distance between two elements of the space. [Klement et al., 2000]

Schweizer and Sklar in [Schweizer et al., 1960] integrated t-norms into the context of probabilistic metric spaces. These t-norms are the operations used to interpret the conjunction in fuzzy logics and the intersection of fuzzy sets [Klement et al., 2004]. These functions have applications in many areas, such as decision making processes, statistics, game theory, information and data fusion, probability theory, and risk management. The t-norms, together with t-conorms, play a key role in fuzzy set theory. Interesting results with practical impact using t-norms in the calculations can be found for example in [Chaira, 2015, Chen and Du, 2015, Liang et al., 2014].

The t-norm is an operation of conjunction, denoted by $\triangle$, satisfying [Jantzen, 2007]:

$$\begin{aligned} 1\triangle x &= x \\ x\triangle y &= y\triangle x \\ x\triangle(y\triangle z) &= (x\triangle y)\triangle z \\ \text{If } w &\leq x \text{ and } y \leq z \text{ then } w\triangle y \leq x\triangle z. \end{aligned}$$

In other words, the operation is associative, commutative, nondecreasing, and has 1 as a neutral element.

Remark 3.3.2. In fact, any operation that agrees with the truth table for two-valued conjunction is a candidate for a t-norm and that is why the definition of fuzzy logic operations may vary from application to application. [Jantzen, 2007]

Although there exist many families of t-norms (for example, the Aczel–Alsina, the

Jane Doe1, Hamacher, Dombi, Yager, Einstein product, for an overview see [Gottwald, 2001, Klement et al., 2000]), four main t-norms will be described in the following text: Łukasiewicz, Gödel, Product and Drastic.

How to choose a suitable t-norm is a good question. The answer is not obvious. Each t-norm offers a good approach and sees things from a different perspective. In general, the t-norms can not be linearly ordered. According to [Iliadis et al., 2008], some t-norms can be characterized as optimistic, with others as pessimistic. Others assign a case a high degree of membership when one or more parameters have extreme values.

To fulfill domain-context demands, various applications require various t-norms, and sometimes it is necessary to create a new one. In [Pedrycz and Gomide, 2007], the following three basic methods of forming a new t-norm are described in detail:

- construction of a new t-norm on the basis of some given t-norms with the use of a monotonic transformation,
- the use of the addition and multiplication of real numbers together with a function of one variable, called additive and multiplicative generators,
- forming a new t-norm from a family of given t-norms based on the concept of the "ordinal sum."

Łukasiewicz norm

The main motivation for the development of logic was to create a logical system that is free of some of the paradoxes in two-valued Boolean logic. The revolutionary step was made by Jan Łukasiewicz with his introduction of three-valued logic, and later on also multi-valued logic [Łukasiewicz, 1930, Łukasiewicz, 1970].

Both Łukasiewicz implication and negation were designed by Jan Łukasiewicz in 1930. A complete description of the derivation of Łukasiewicz conjunction can be also found in [Gottwald, 2005].

As mentioned above, the Łukasiewicz logic has the two following primitive connectives, implication and negation [Gottwald, 2005]:

$$x \rightarrow_L y = \min(1, 1 - x + y) \tag{3.3.13}$$

$$\neg x = 1 - x \tag{3.3.14}$$

From these primitive properties, further connectives are defined as syntactic abbreviations:

$$\varphi \,\&\, \psi = \neg(\varphi \rightarrow_L \neg\psi), \qquad \varphi \veebar \psi = \neg\varphi \rightarrow_L \psi \tag{3.3.15}$$

With the above defined properties, one gets a conjunction and a strong disjunction with truth degree functions that are usually called Łukasiewicz arithmetical conjunction and Łukasiewicz arithmetical disjunction:

$$x \,\&\, y = \max\{x + y - 1, 0\}, \qquad x \veebar y = \min\{u + v, 1\}. \tag{3.3.16}$$

Then, in the case of conjunction,

$$x \otimes_L y = \max\{x + y - 1, 0\}. \tag{3.3.17}$$

Łukasiewicz implication has applications in many fields. Together with the negation of Łukasiewicz implication, it was used by Mills in [Mills, 1993] to design sensors for small robots. Łukasiewicz implication is also used in [Kosko, 1986] as a measure of the entropy of the fuzziness of messages, to develop fuzzy information theory. The use of the fuzzy conjunction in digital hardware implementation is formulated in [Rudas et al., 2008].

The fuzzy extension of description logics using Łukasiewicz logic (the formalism for the representation of structured knowledge) is described in [Bobillo and Straccia, 2011]. Description logic is frequently used in the design of ontologies, which have been successfully used as part of expert and multiagent systems, as a knowledge base in robotics, as well as the core element in the Semantic Web (which aims at converting the current Web into a "Web of data" by defining the meaning of information).

Łukasiewicz arithmetical conjunction can also be used in many types of situations in real life. The fact that the number 1 is subtracted and the maximum with zero is taken (in case that $x+y<1$) leads to the observation that the count of the operation is the remainder, some part that is over the unit. Following this idea, the conjunction can be used to compute the amount of money that should be paid off for a phone bill, where x can be the price for the total SMSs, and y can represent the price for the calls in total. The number 1 here is replaced by the amount of the lump sum. There are other similar situations, for example, data backup on a server, the maximal capacity of a pond, overspending of project funds, or savings of partners with a common bill.

Gödel norm

This is the simplest norm, in the literature known also as the "minimum norm" or the "strongest norm"; the conjunction is defined as the minimum of the entries: of the truth degrees of the constituents [Gottwald, 2001].

$$x \otimes_G y = min(x, y) \tag{3.3.18}$$

In contrast to Łukasiewicz logic (which might be considered as a logic of absolute or metric comparison), Gödel logic is a logic with a relative comparison.

Product norm

The definition of the product norm follows, a complete description of this norm can be found in [Hájek et al., 1996].

$$x \otimes_p y = x \cdot y \tag{3.3.19}$$

In [Perovic et al., 2011] there can be found the application and comparison of the Gödel and Product norms in a classification problem—the ordering of some finite set of evaluations according to certain criteria. The product norm corresponds here to the stochastic or probability independence of elementary events (propositional letters), while the Gödel norm represents another kind of extreme situation: the logical dependence of propositional letters.

Drastic norm

The drastic triangular norm (in the literature there can be also found the term "weakest norm" or "drastic product") is the basic example of a non-divisible t-norm on any partially ordered set, see [Casasnovas and Mayor, 2008]. The drastic triangular norm (the operation of $\otimes_d$) is defined as follows:

$$x \otimes_{\mathrm{d}} y = \begin{cases} \min(x, y) & \text{if } \max(x, y) = 1 \\ 0 & \text{if } \max(x, y) < 1 \end{cases} \tag{3.3.20}$$

The drastic norm is "drastic" in the sense that when there is some possibility of using, for example, two weighted paths, say path x and path y, they can be used if and only if one of them is equal to 1. Otherwise such a concatenated path can not be used. Moreover, if the walk consists of more than two paths, then at most one path can be weighted by a value less than 1. The interpretation of the drastic norm then reflects a situation when extreme demands are imposed on the reliability of a system: the walk is reliable if there is no more than one unsure segment.

The matrix powers in max-drast algebra behave differently than those in max-min algebra. However, the properties of the matrix periods in both cases show some similarity.

The differences between the four basic t-norms can be seen in Figure 3.8. Although the figure depicts the intersection of the fuzzy sets in the classical conception, the plots show the difference between the operations clearly. The contour plots for the basic t-norms can be seen in Figure 3.9 (the contour plot shows us the sets where the function in question has constant values). From the figure it can also be seen that the Gödel, Product and Łukasiewicz norms are continuous whereas the drastic norm is not. According to [Pedrycz and Gomide, 2007], the continuity of the operations on fuzzy sets is a highly desirable feature in many applications.

The norms can be also compared, for details see [Klement et al., 2000], since they are just functions. The comparison is done pointwise. If for two t-norms, say $\triangle_1$ and $\triangle_2$, it holds that

$$x \triangle_1 y \leq x \triangle_2 y$$

for all $x, y \in [0, 1]$, then $\triangle_1$ is weaker than $\triangle_2$ (or, equivalently, it can be said, $\triangle_2$ is stronger than $\triangle_1$), and thus one can write $\triangle_1 \leq \triangle_2$. This yields the following order of the four basic t-norms:

$$\text{Drastic} < \text{Łukasiewicz} < \text{Product} < \text{Gödel}.$$

It can also be observed from the definitions of the operations for the t-norms that by using the product or Łukasiewicz norm, new elements (values) are generated, whereas for the Gödel and drastic norm this is not true. The drastic norm is also characterized by the fact that small changes in the inputs produce big changes in the output. This results from the character of the operations.

The characteristic of idempotency can be formulated as follows: the value x is idempotent if $x \triangle x = x$. Note that the values 0 and 1 are idempotent elements for any t-norm. The t-norm is Archimedean (i.e., for any two elements a and b from the set there exists $n \in \mathbb{N}$ such that $n \cdot a > b$) if and only if its only idempotent elements are the trivial ones 0 and 1. [Casasnovas and Mayor, 2008]

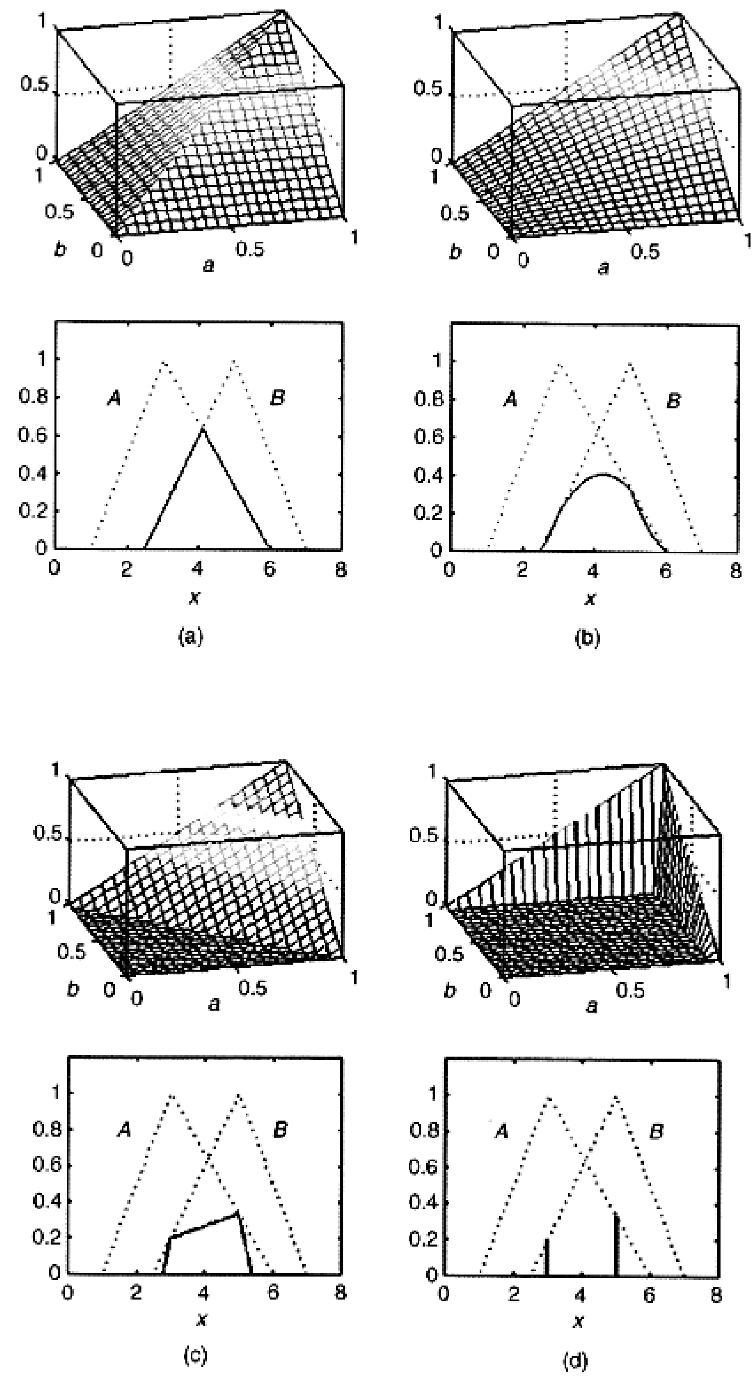


Figure 3.8: T-norms used in the realization of the intersection of fuzzy sets A and B: (a) Gödel, (b) Product, (c) Łukasiewicz, (d) Drastic. [Pedrycz and Gomide, 2007]

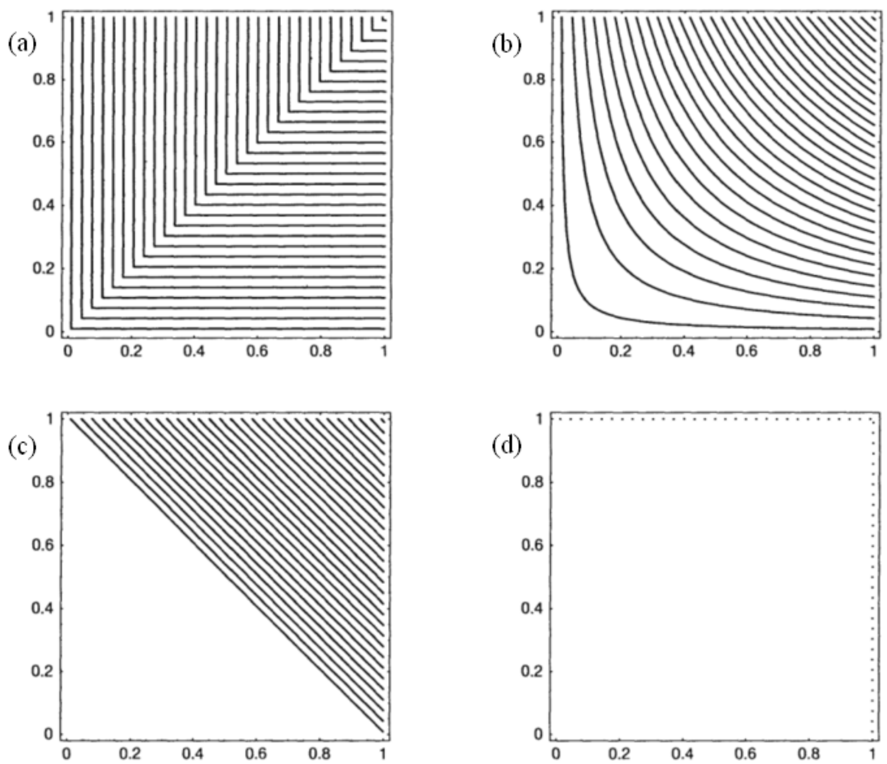


Figure 3.9: Contour plot of the t-norms: (a) Gödel, (b) Product, (c) Łukasiewicz, (d) Drastic. [Klement et al., 2000]

Triangular conorms

A brief characterization of the *triangular conorms* (t-conorms for short) will be given next.

A t-conorm is an operation of disjunction, the dual operation to that of a t-norm, on the unit interval, denoted by ∇, and satisfying [Jantzen, 2007]

$$\begin{aligned} 0\nabla x &= x \\ x\nabla y &= y\nabla x \\ x\nabla(y\nabla z) &= (x\nabla y)\nabla z \\ \text{If } w &\leq x \text{ and } y \leq z \text{ then } w\nabla y \leq x\nabla z. \end{aligned}$$

A t-conorm can be generated from a t-norm by

$$x\nabla y = 1 - (x+1)\triangle(y+1) \tag{3.3.21}$$

Among the basic t-conorms there are [Klement et al., 2000]

$$\begin{aligned} \text{Maximum:} \quad & x \oplus_G y = \max(x, y) \\ \text{Probabilistic sum:} \quad & x \oplus_p y = x + y - x \cdot y \\ \text{Łukasiewicz sum:} \quad & x \oplus_L y = \min(x + y, 1) \\ \text{Drastic sum:} \quad & x \oplus_{\mathrm{d}} y = \begin{cases} 1 & \text{if } \min(x, y) > 0 \\ \max(x, y) & \text{if } \min(x, y) = 0 \end{cases} \end{aligned}$$

It can be easily seen from the definition of a conorm that the difference between norms and conorms lies in their boundary conditions. The name, t-conorm, comes from the fact that in the unit interval, the value x acts in a similar way as its complement $1 - x$. [Klement et al., 2000]

Similarly to triangular norms, the conorms can be visualized, see Figures 3.10 for 3D and 3.11 for contour plots.

Remark 3.3.3. Since both the operations for t-norms and t-conorms were introduced as operations with two arguments, they can be, of course, extended to n-ary operations in the usual way by induction.

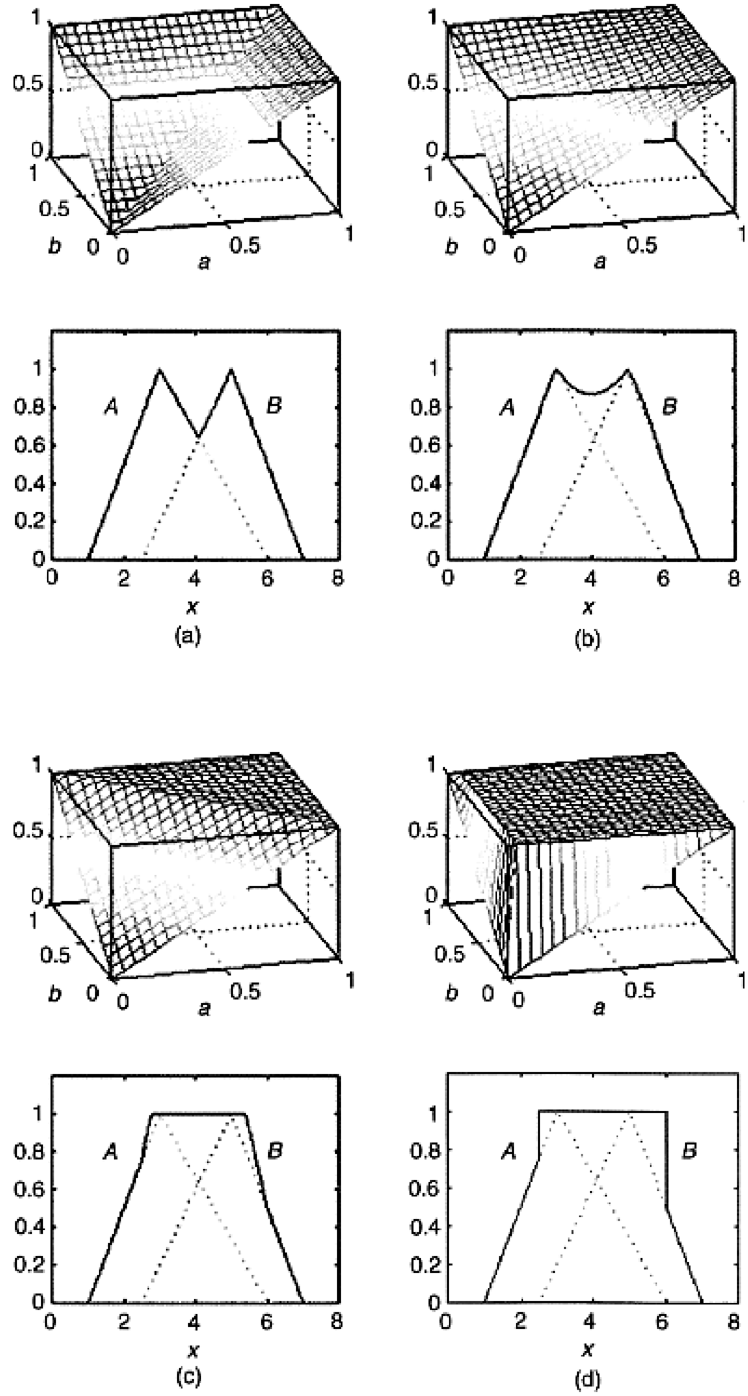


Figure 3.10: T-conorms used in the realization of the intersection of fuzzy sets A and B: ((a) Maximum, (b) Product, (c) Łukasiewicz, (d) Drastic. [Pedrycz and Gomide, 2007]

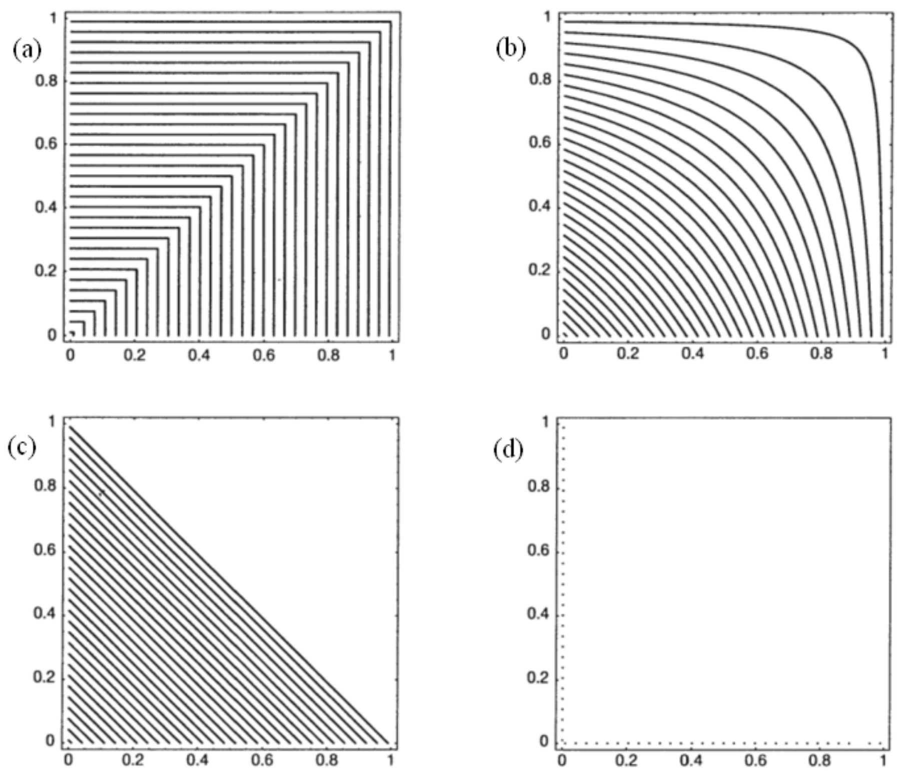


Figure 3.11: Contour plot of the t-conorms: (a) Maximum, (b) Product, (c) Łukasiewicz, (d) Drastic. [Klement et al., 2000]

4. Queuing Systems

The research results from the investigation of economic systems working in discrete time can be divided into two parts. The first part deals with the optimization of the performance of the queuing systems presented in the further subchapters.

4.1 Closed Linear Queuing Systems

The research results about closed systems will be introduced and summarized in this chapter. The results have been already published, cf. [Gavalec and Němcová, 2010, Němcová, 2011a, Němcová, 2011b]. Similar systems were investigated using stochastic methods by Heidergott [Heidergott, 1998, Heidergott, 2000]. The considered queueing systems lead to a special type of max-plus matrices, the so-called two-diagonal matrices.

The queuing system considered in this chapter consists of n servers which form a closed circle. Each server can serve only one customer at a time, and every customer has to pass all servers continuously. In other words, they are circulating in the system (see Fig. 4.1). In a simplified way, at the beginning there is one customer in each queue, thus there are n customers in the system.

Systems of this kind occur in manufacturing systems where several successive operations must be repeated many times, or without repeated operations but with the outgoing items immediately being replaced by fresh ones. These are the systems with limited capacity where customers are passing through all stations and after

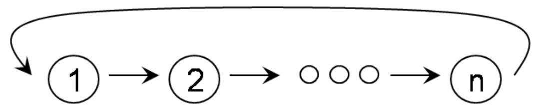


Figure 4.1: A closed queuing system

completion of the service at the nth server, the customer leaves the network and at the same time another customer enters the system replacing the one leaving. Another area of possible applications is transport problems with circulating vehicles, for example, in public transport (busses passing through all the stations).

In this chapter a fixed natural number n with notation $N = \{1, 2, \ldots, n\}$ is used. The set of all square matrices (vectors) of dimension n over $\overline{\mathbb{R}}$ will be denoted by $\overline{\mathbb{R}}(n, n)$ ($\overline{\mathbb{R}}(n)$). A discrete event system with n components can be described by the transition matrix $A \in \overline{\mathbb{R}}(n, n)$ and by the sequence of states $x(k) \in \overline{\mathbb{R}}(n), k = 1, 2, \ldots$. Let $\sigma_i(k)$ be the kth service time at queue i and let $x_i(k)$ be the time of completion of the kth service at node i. The state vector $x(k) = (x_1(k), x_2(k), ..., x_n(k))$ describes the time evolution of the system:

$$x(k+1) = A \otimes x(k) \tag{4.1.1}$$

The transition matrix of this system is of a special type. It can be called a *two-diagonal* matrix

$$A = \begin{pmatrix} \sigma_1 & \varepsilon & \cdots & \varepsilon & \sigma_1 \\ \sigma_2 & \sigma_2 & \varepsilon & \cdots & \varepsilon \\ & & \ddots & & \vdots \\ \cdots & \varepsilon & \sigma_{n-1} & \sigma_{n-1} & \varepsilon \\ \cdots & \cdots & \varepsilon & \sigma_n & \sigma_n \end{pmatrix} \tag{4.1.2}$$

This specific form of the matrix (see also the communication graph for $n = 5$, Fig. 4.2) follows from the fact that a service node, say M, cannot serve the next customer until both this customer has already been served at the previous node $(M-1)$ and the

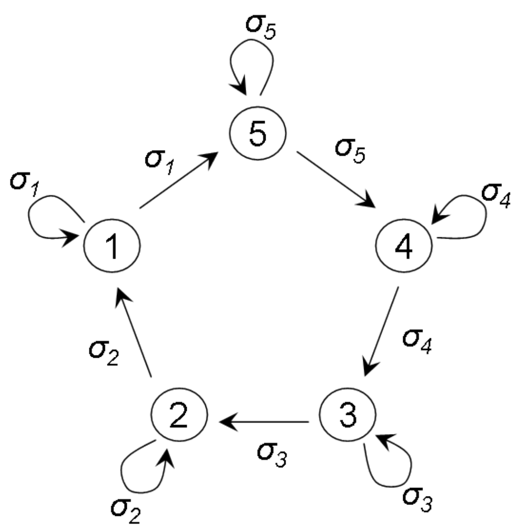


Figure 4.2: The communication graph of A for $n = 5$

service of the current customer at node M has been completed. In the following, the eigenvalue, its eigenspace (the set of all its eigenvectors), and its dimension will be described. Also the lengths of the queues in the system will be computed. The maximal average weight of the cycles in the communication graph of A is denoted by $\lambda = \lambda(A)$ and $A_\lambda = A \otimes (\lambda^{-1})$. The notation A_λ^+ is used for the sum of the formal powers of A_λ, as follows: $A_\lambda^+ = \bigoplus_{r=0}^{n-1} A_\lambda^{(r)}$. The columns of A_λ^+ with the value 0 on the diagonal are called the fundamental eigenvectors of A.

4.1.1 Eigenvalues of Two-Diagonal Matrices

Theorem 4.1.1. *The eigenvalue $\lambda(A)$ of a two-diagonal matrix A is equal to* $\max\{\sigma_i, i \in N\}$.

Proof. Write $\sigma^* = \max\{\sigma_i, i \in N\}$. It will be shown first that $\lambda \leq \sigma^*$. The eigenvalue λ is the maximal average weight of cycles in the communication graph of A, consider a cycle $c = (i_0, i_1, i_2, \ldots, i_k)$ with $\overline{w}(c) = \lambda$,

$$c = (i_0, i_1)(i_1, i_2)(i_2, i_3)\ldots(i_{k-1}, i_k) \ , \tag{4.1.3}$$

$$w(c) = a_{i_0 i_1} + a_{i_1 i_2} + \ldots + a_{i_{k-1} i_k} \ . \tag{4.1.4}$$

As $a_{i_{s-1} i_s} \leq \sigma^*$ for all $s = 1, 2, ..., k$, one has

$$\lambda = \overline{w}(c) = \frac{w(c)}{k} = \frac{\sum_{s=1}^{k} a_{i_{s-1} i_s}}{k} \leq \frac{k\sigma^*}{k} = \sigma^* \ . \tag{4.1.5}$$

It was shown that $\lambda \leq \sigma^*$.

On the other hand, $\max(\sigma_i) = \sigma_{i^*}$ for some $i^* \in N$. The average weight of the cycle $c^* = (i^*, i^*)$ whose only arc is the loop from i^* to i^* is

$$\overline{w}(c^*) = \frac{a_{i^* i^*}}{1} = \frac{\sigma_{i^*}}{1} = \sigma^* \ . \tag{4.1.6}$$

It follows that $\lambda \geq \sigma^*$ and, as a consequence, $\lambda = \sigma^*$. □

4.1.2 Eigenvector of Two-Diagonal Matrices

Theorem 4.1.2. *The number of fundamental eigenvectors of a two-diagonal matrix A is equal to the number of occurrences of the maximal value σ^* on the diagonal of A.*

Proof. The number of the fundamental eigenvectors is equal to the number of columns with the value 0 on the diagonal in A_λ^+. The matrix A_λ^+ has the value 0 exactly in those places where A has its maximal values because the loop (i, i) has the maximal weight of all paths from the node i to i. □

The elements of A_λ are denoted by a'_{ij}, $i, j \in N$ and the elements of A_λ^+ by a_{ij}^+. The columns of A_λ^+ are described by the following theorem.

Theorem 4.1.3. *Let A be a two-diagonal matrix and let $k \in N$. Then*

$$a^+_{kk} = \sigma_k - \lambda \ , \tag{4.1.7}$$

$$a^+_{k+m\,k} = \sum_{i=1}^{m} \sigma_{k+i} - m\lambda \ . \tag{4.1.8}$$

Proof. It can be easily seen from the communication graph of A that the path $p = (k+m, k+m-1, \ldots, k)$ has the maximal weight of all paths from the node $k+m$ to node k. Hence the maximal weight of the path from $k+m$ to k is $w(p) = \sum_{i=1}^{m}(\sigma_{k+i} - \lambda)$, which is the value of $a^+_{k+m\,k}$. □

Remark 4.1.1. The first two diagonals in A^+_λ are the same as in A_λ, because the values in A_λ are negative and by squaring this matrix one get smaller values than the former ones on the diagonals. Thus,

$$a^+_{i\,i} = a'_{i\,i} \ , \tag{4.1.9}$$

$$a^+_{i+1\,i} = a'_{i+1\,i} \ . \tag{4.1.10}$$

Subsequent diagonals in A^+_λ are taken from the powers of A_λ

$$a^+{}_{i+l\,i} = a'^{(l)}_{i+l\,i} \qquad l = 2, 3, \ldots \tag{4.1.11}$$

Remark 4.1.2. It follows from Theorem 4.1.3 that the eigenvectors of a two-diagonal matrix can be computed directly from A by using the formula describing the kth column of A^+_λ. The eigenvectors correspond to the columns in A with the maximal value on the diagonal, i.e., with the value zero at the same place in A^+_λ. The computation can be done recursively:

$$a^+_{kk} = \sigma_k - \lambda \ , \tag{4.1.12}$$

$$a^+_{k+m+1\,k} = a^+_{k+m\,k} + \sigma_{k+m+1} - \lambda \ . \tag{4.1.13}$$

for every $m = 1, 2, \ldots, n-1$ (the indices are computed modulo n).

The dimension of the eigenspace of A can be found directly from A.

Theorem 4.1.4. *The dimension of the eigenspace of a two-diagonal matrix A is equal to 1 if $\sigma_i = \sigma_j$ for all $i, j \in N$, and it is equal to the number of occurrences of the maximal element on the main diagonal, otherwise.*

Proof. If $\sigma_i = \sigma_j$ for all $i, j \in N$, then clearly $a^+_{ij} = 0$ for every $i, j \in N$. In this case all columns of A^+_λ are fundamental eigenvectors, and they are all equal to the same zero vector. Hence the dimension of the eigenspace is 1.

It can be shown that in any other case, every pair of fundamental eigenvectors is independent. On the contrary, assume that the kth and the lth column of A^+_λ are two fundamental eigenvectors. Assume that $k < l$, i.e., $l = k + m$, $k = l + (n - m)$ for some $m \in N$, $0 < m < n$.

If the kth and lth columns are dependent, then there is a real number c such that $a^+_{rk} = c \otimes a^+_{rl}$ for every $r \in N$. As the columns are fundamental eigenvectors, they both have the value 0 on the main diagonal, i.e., $a^+_{kk} = 0$ and $a^+_{ll} = 0$. As all inputs of A^+_λ are ≤ 0, we get $c = 0$ and $a^+_{lk} = a^+_{kl} = 0$.

The equality $a^+_{lk} = a^+_{k+m\,k} = \sum_{i=1}^{m} \sigma_{k+i} - m\lambda = 0$ implies that $\sigma_{k+1} = \sigma_{k+2} = \cdots = \sigma_l = \lambda$, and the equality $a^+_{kl} = a^+_{l+(n-m)\,l} = \sum_{i=1}^{n-m} \sigma_{l+i} - (n-m)\lambda = 0$ implies that $\sigma_{l+1} = \sigma_{l+2} = \cdots = \sigma_k = \lambda$. Hence $\sigma_i = \sigma_j$ for all $i, j \in N$, which completes the proof. □

4.1.3 The Lengths of the Queues

The eigenvectors and the eigenvalue alone do not describe the lengths of the queues in the system. According to the assumptions, there are n circulating tokens (representing customers) in the considered type of closed system. The queues in the stabilized state are naturally created in front of the slowest servers in the system, and there are no queues in front of the faster servers. The queue lengths are completely described below in the case of one slowest server in the system (the case with more than one slowest server is analogous).

For convenience, assume without loss of generality that the slowest server is placed at position n and the servers at positions $1, 2, \dots n-1$ are faster. In other words, assume $\lambda = \sigma_n$ and $\lambda > \sigma_i$ for $i < n$. The maximal value λ is the eigenvalue of the two-diagonal matrix describing the system and it is the period of the stabilized state. In the stable state, tokens are leaving the slowest server n at intervals of length λ. Every token then circulates, visiting the remaining servers, until it eventually reaches the slowest server and enters the queue at position n. During the working of the system, the length of this queue changes, but the change is never larger than 1.

The basic length of the queue is denoted by z and the extended length of the queue by $z+1$, the time during which the extended length is preserved as m and the rest of the time (when the queue has the basic length) as $\lambda - m$.

Theorem 4.1.5. *In the system described by a two-diagonal matrix A with a single maximal value, λ, the basic length, z, and the time, m, during which the queue has its extended length can be computed by the following formula, where k denotes the eigenvector column. In the formula below, z is the integer part of the quotient $|a^+_{k-1\,k}|/\lambda$ and $0 \leq m < \lambda$ is the remainder.*

$$|a^+_{k-1\,k}| = z\lambda + m \quad . \tag{4.1.14}$$

Proof. The tokens are circulating in the system with period of length equal to λ. The absolute value $|a^+_{k-1\,k}|$ represents the minimal time in which a token gets from node $k-1$ to node k. If this value is divided by λ, then the integer part, z, of the quotient indicates the minimal number of tokens coming from node $k-1$ and waiting in the queue before they enter node k. In the special case when the division gives no remainder, the length of the queue is permanently equal to z. In the general case, the remainder m is positive, and it is equal to the time between the arrival of a further token joining the queue, and the token at the first position leaving the queue and entering the service at k. During this time the length of the queue is $z+1$, otherwise the length is z. $\square$

Remark 4.1.3. If a two-diagonal matrix A has the maximal value λ at several positions on the main diagonal, there is more than one queue in the system. The lengths of the queues can be computed similarly to the above. However, the situation with more than one queue is more complicated, as the lengths substantially depend on the positions of the maximal value.

4.1.4 Three-Diagonal Matrices

There are two types of three-diagonal matrices, depending on the positions of the other two diagonals. Both types are shown in (4.1.15). Type 1 contains one diagonal above the main diagonal and the second one under the main diagonal. In the second type, both diagonals are placed under the main diagonal (or above the main diagonal, in which case the situation is analogous, only the arcs in the communication graph of matrix have the inverse direction).

$$\begin{pmatrix} \sigma_1 & \sigma_1 & \varepsilon & \cdots & \varepsilon & \sigma_1 \\ \sigma_2 & \sigma_2 & \sigma_2 & \varepsilon & \cdots & \varepsilon \\ \vdots & & \ddots & & & \vdots \\ \vdots & & & \ddots & & \vdots \\ \varepsilon & \cdots & \varepsilon & \sigma_{n-1} & \sigma_{n-1} & \sigma_{n-1} \\ \sigma_n & \varepsilon & \cdots & \varepsilon & \sigma_n & \sigma_n \end{pmatrix} \quad \begin{pmatrix} \sigma_1 & \varepsilon & \cdots & \varepsilon & \sigma_1 & \sigma_1 \\ \sigma_2 & \sigma_2 & \varepsilon & \cdots & \varepsilon & \sigma_2 \\ \vdots & & \ddots & & & \vdots \\ \vdots & & & \ddots & & \vdots \\ \cdots & \varepsilon & \sigma_{n-1} & \sigma_{n-1} & \sigma_{n-1} & \varepsilon \\ \varepsilon & \cdots & \varepsilon & \sigma_n & \sigma_n & \sigma_n \end{pmatrix} \tag{4.1.15}$$

4.1.5 Eigenvalues of Three-Diagonal Matrices

The eigenvalues of three-diagonal matrices of both types can be computed in the same way as in Theorem 4.1.1. In other words, if A is three-diagonal, then $\lambda(A)$ is the maximal input in A.

4.1.6 Eigenvectors of Three-Diagonal Matrices

For the first type of three-diagonal matrix A, the columns of the corresponding matrix A^+_λ and, therefore also the eigenvectors of A, can be computed by the formula presented in the following theorem.

Theorem 4.1.6. *Let A be a three-diagonal matrix of type 1 and let $k \in N$. Then*

$$a^+_{k+m\,k} = \max\left(\sum_{i=1}^{m} \sigma_{k+i} - m\lambda, \sum_{i=1}^{n-m} \sigma_{k-i} - (n-m)\lambda\right) \quad . \tag{4.1.16}$$

Proof. It can be easily seen from the communication graph of A that there are two possible paths from node $k+m$ to node m. The first path is $p = (k+m, k+m-1, \dots, k)$ and the second one is $q = (k+m, k+m+1, \dots, k)$. As the elements of A^+_λ represent the maximal path weights, the element $a^+_{k+m\,k}$ is not less than the maximal weight of the paths p, q. On the other hand, any other path from $k+m$ to m is an extension of p or q. Hence its weight cannot be larger than $\max(w(p), w(q))$ because all elements in A^+_λ are ≤ 0. $\square$

The eigenvectors (or columns of A^+_λ) of a three-diagonal matrix of the second type cannot be expressed by a single formula. The computation in the special case when $\sigma_1 \leq \sigma_2 \leq \cdots \leq \sigma_n$ will be described next.

The largest value is equal to the eigenvalue $\sigma_n = \lambda$, and every kth column of matrix A^+_λ with diagonal value $a^+_{k\,k} = 0$ is an eigenvector of A. All the columns of A^+_λ are described in Theorem 4.1.7 below.

Theorem 4.1.7. *Let A be a three-diagonal matrix of type 2 and let $\sigma_1 \leq \sigma_2 \leq \cdots \leq \sigma_n$. Then the following formulas can be used for the computation of the kth column of A^+_λ (the maximal weights of paths from any other node to node k). If the path leads from node $k+s$ to k for $0 \leq s \leq n-k$ (in other words, the path does not cross node n), then the values $a^+_{k+s\,k}$ are computed by*

$$
\begin{aligned}
a^+_{k\,k} &= \sigma_k - \lambda \ , && (4.1.17)\\
a^+_{k+2m\,k} &= \sum_{l=1}^{m} \sigma_{k+2l} - m \cdot \lambda \ , && (4.1.18)\\
a^+_{k+2m+1\,k} &= \sum_{l=1}^{m} \sigma_{k+2l} + \sigma_{k+2m+1} - (m+1) \cdot \lambda \ . && (4.1.19)
\end{aligned}
$$

If the path, say c, crosses node n, then the path must be divided into two paths, say p and q, where p ends in node n and q begins there. Then the above formulas can be used separately on p and q, and the maximal weight of c is $w(p) + w(q)$.

Proof. There are two possible paths from node $i+2$ to node i: $p = (i+2, i)$ and $q = (i+2, i+1, i)$. The weights of the paths are $w(q) = (\sigma_{i+2} - \lambda) + (\sigma_{i+1} - \lambda) \leq (\sigma_{i+2} - \lambda) = w(p)$. Then it follows that $w(p)$ is the maximal weight of the path. So, one "long step" is more advantageous than two "short steps." If node n is crossed, which means that $i+1 = n$, then $w(p) = w(q)$.

There are also two paths from node $i+3$ to node i: $p = (i+3, i+2, i)$ and $q = (i+3, i+1, i)$. If node n were not crossed, the weights of the paths are $w(q) = (\sigma_{i+3} - \lambda) + (\sigma_{i+1} - \lambda) \leq (\sigma_{i+3} - \lambda) + (\sigma_{i+2} - \lambda) = w(p)$. As the value σ with higher index is more advantageous than a σ with lower index (because of increasing σ_i), it is clear that p is the most advantageous path.

In general, if n is not crossed by the considered path, then for the maximal weight of the path from node $k+2m$ to node k, only long steps should be used, and from node $k+2m+1$ to node k there has to be used just one short step right at the beginning of the path. The formulas above are valid if the sequence $\sigma_k, k \in N$ increases.

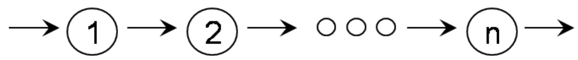


Figure 4.3: The open queuing system

On the other hand, if the considered path from $k+s$ to k crosses n, then $\ldots \leq \sigma_{n-2} \leq \sigma_{n-1} \leq \sigma_n > \sigma_1 \leq \sigma_2 \leq \ldots$. For the fulfillment of the monotonicity condition, the path can be divided into two subpaths p and q with increasing values σ_i. The subpath p begins in the start node $k+s$ and ends in node n, while the subpath q starts in node n and ends in the end node k of the first path. Then the weight of the considered path is the sum of $w(p)$ and $w(q)$. □

Remark 4.1.4. It can be easily seen that the computation of the eigenvectors is a special case of Theorem 4.1.7 for all k with $\sigma_k = \sigma_n = \lambda$.

4.2 Open Linear Queuing Systems

This chapter summarizes already published results in the study of open systems, which can be found in [Cimler and Němcová, 2012, Gavalec and Němcová, 2011, Gavalec and Němcová, 2014, Němcová, 2014]. The sections proposing the optimization methods (Chapters 4.2.5 and 4.2.6) will be published either in the journal *Economics and Management* or some similar journal.

Similarly to the closed type of system, also in the open queuing system (see Figure 4.3) the requests come in order to be served at a sequence of service places. The intensity of arrival of the requests to the system can naturally vary in time. The service times of the servers at the service places can also differ. Depending on the ratios between these quantities, queues can appear, and, on the other hand, some of the servers in the system can become idle. Sometimes the occurrence of queues and/or idle servers is undesirable because this can pose additional indirect costs. This situation can result in searching for better possible system settings.

The arrangement of the service places is important as well: the service places can be concatenated either in series and/or in parallel. Queuing systems can have various structures, from those with the simplest layout (for example the cash desk at a gasoline station) to those with a complex organization (for example, assembly lines).

Particular characteristics describing the behavior of the system can be computed during the system analysis, depending on the user's intentions. These characteristics can be divided into several categories [Baum and Breuer, 2005]:

- The time characteristics of the requests;
- Characteristics related to the number of requests;
- Probabilistic characteristics;
- Cost characteristics.

Assessing these quantities is important especially for system design or modernization. It helps to forecast the corresponding extent of the system so that its performance is optimal.

4.2.1 Types of Queues

Generally, the queues in front of the service places behave differently for different system settings. Some of them remain stable, some of them increase or decrease. In the so-called transient time, it can happen that the queue is temporarily increasing or decreasing. This depends not only on the service intensity at the previous service place but also on the situation at all preceding service places. After the transient time, in which the queues can possibly change the sign of their growth, the steady state is considered. Then the queues do not change their character, i.e., each queue either remains stable, or constantly increases or decreases (except in the situation where the queue becomes empty).

In the steady state, four types of queues are distinguished:

- constant queues
- decreasing queues
- increasing queues
- empty queues.

An empty queue emerges as a result of a decreasing queue, or it is a special case of the constant queue.

The vector of service times is denoted by $\mathbf{z} = (z_0, z_1, \ldots, z_n)$, i.e., the service time at the server i is z_i. Note that the index i takes the values $i = 0, 1, 2, \ldots n$. The length of the queue in front of server i at time t is denoted by $l_i(t)$, and is contained in the vector of waiting requests $\mathbf{l}(t) = (l_0(t), l_1(t), \ldots, l_n(t))$.

The following theorem describes various situations that may occur after changing the service intensity in the ith service place. Some of them can be seen in Fig. 4.4. The number in the circle denotes the sequence of service places, the letter l indicates the queue, the arrows its tendency. The upward continuous arrow means that the queue will increase in the long term, the downward continuous arrow indicates that the queue will decrease in the long term. Similarly, the dashed arrows represent the short term tendencies of the queues: their temporary growth and decay.

Theorem 4.2.1. *The queue in front of the ith server in the system remains constant in the stable state if* $z_i = \max(z_0, ..., z_{i-1})$.

The queue in front of the ith server in the system will decrease in the stable state if $z_i < \max(z_0, ..., z_{i-1})$.

The queue in front of the ith server in the system will increase in the stable state if $z_i > \max(z_0, ..., z_{i-1})$.

The queue in front of the ith server in the system will temporarily increase if $z_{i-k} > z_{i-1} < z_i$.

Proof. If $z_i = \max(z_0, z_1, ..., z_{i-1})$, then clearly the intensity of the arrivals of requests at the queue in front of server i (queue i) is the same as the service intensity of server i. In other words, the server serves the requests at the same speed as the requests come at the queue in front of the server: the length of the queue in a stable state does not change.

If $z_i < \max(z_0, ..., z_{i-1})$, then the service intensity of server i is higher than the intensity of the arrivals of requests at queue i: the queue will decrease in a stable state.

If $z_i > \max(z_0, ..., z_{i-1})$, the service intensity of server i is lower than the intensity of arrivals of the requests at queue i. Thus the queues will continually grow just in front of those servers which precede the faster servers (permanently growing queues are created in front of the servers which are slower than all servers before them).

If $z_{i-k} > z_{i-1} < z_i$, the ith queue will temporarily increase because the queue in front of the $(i-1)$th (faster) server overflows to the ith queue. But in the long term the queue will decrease because the intensity of arrivals at queue i will be lower, thanks to server z_{i-k}, than the service intensity of server i. □

The rate of change of the length of the queue at each server (except the first one) can be calculated via the following theorem.

Theorem 4.2.2. *Let R be the rate of change of the length of the queue at server i. The rate of change in the stable state is*

$$R = \frac{z_i - z_{i-1}}{z_{i-1} z_i} \tag{4.2.1}$$

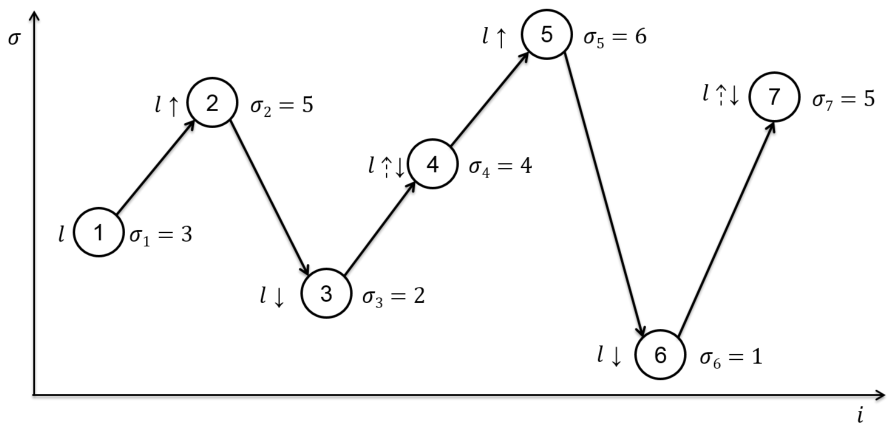


Figure 4.4: Changes in the lengths of the queues

Proof. Denote by z_i and z_{i-1} the service times at servers (i) and $(i-1)$. Denote by $1/z_i$ and $1/z_{i-1}$ the service intensities of the servers. Then it is easily seen that the queue in front of server (i) enlarges by $1/z_{i-1}$ and lessens by $1/z_i$. Then

$$\frac{1}{z_{i-1}} - \frac{1}{z_i} = \frac{z_i - z_{i-1}}{z_{i-1} z_i}.$$

□

The length of the queue at server i changes, averaging over $z_{i-1} \cdot z_i$ time units, by about $z_i - z_{i-1}$ requests. If $z_i - z_{i-1} > 0$, the type of the queue is increasing. But if $z_i - z_{i-1} < 0$, the type of the queue is decreasing.

The time period during which the service times are constant will be referred to below as a stage. The transition from one stage to another can be have two reasons: internal or external (also described below). An external reason means that the intensity of arrivals, z_1, has changed. An internal reason means that the service intensity, z_i for $i > 1$, has changed by managerial decision as a response to changes in the queue lengths.

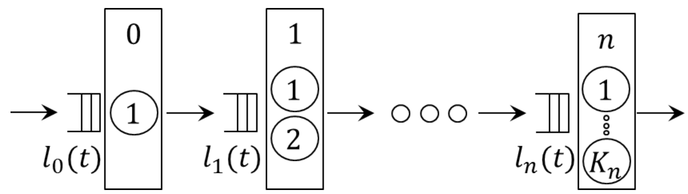


Figure 4.5: An open linear system

4.2.2 The Considered System

Consider an open linear queuing system (in the literature, this is also known as a tandem queuing network, see [Lawrence and Pasternack, 1997]) with $n+1$ service places, see Figure 4.5. The incoming requests have to pass through the series of all the service places and then leave the system. There can be arbitrarily long queues in front of each service place before the system starts. The system is studied in the so-called stages: a stage is the time period during which the system settings remains unchanged. The time variable during a stage takes the values $t = 1, 2, ..., T$, where T is the length of the stage. The manager of the system has the following information about the system's settings:

- z_i, the basic service time of the server(s) at service place i,
- K_i, the number of identical servers at the service place i,
- l_i, the length of the queue at the service place i.

The service intensity at a particular service place is then $\sigma_i = K_i/z_i$. The intensity of arrivals is represented by the service intensity of the entrance server (indexed by $i = 0$). This server can be considered as a gatekeeper, which controls the access of the requests. It is assumed that the queue in front of the entrance server is long enough; the gatekeeper admits the requests to the system with a constant speed.

The service times of the service places can be changed. The cause of this change can be either internal or external. An internal reason means that the manager of the system has decided to change some of the service times, quantities that can be directly influenced (the change in the value z_i, where $i > 0$). An external reason represents a change in the intensity of the arrivals of the requests coming into the system (the change in z_0, respectively K_0, in the model). It is important to note that the possibility of an unexpected change in the intensity of arrivals as well as the need to cope with the different types of queues can foil the effort to set the service intensities to the intensity of arrivals at the start of the system.

4.2.3 Production Costs

The aim of the optimization is to adapt service times to the arrival intensities and to control the lengths of the queues as well as the number of empty queues to minimize the total cost of the system.

The optimization is made in each stage separately, with respect to the actual system settings and the function of the total production cost of the stage. The function includes the components that constitute important factors for the system's performance.

The manager of the system can then significantly affect the overall production costs of the system by making the decision to change various service intensities.

The production cost P consists of the work costs W, the costs of inaction on the part of some of the servers I, the costs resulting from long queues Q, and the costs of changing the number of servers in some of the service places C, i.e.,

$$P = W + I + Q + C \tag{4.2.2}$$

Among the basic system parameters also belong the constituent costs at particular service places, W_i, I_i, Q_i, C_i, and the maximum tolerated (not yet penalized) queue lengths M_i.

The work costs occur whether the request is being served or not. It is proportionally dependent on the number of the servers in the service place i, K_i. These are the costs necessary to ensure the functioning of the servers, for example, electricity costs or wages.

The idle costs are the indirect costs that are spent when the server is not working. These costs are also proportionally dependent on the number of inactive servers in the service place i.

Queues that are too long can be considered as a loss of profit, and the same for the next sort of costs, queuing costs. This reflects the reality: the longer the queue, the higher the dissatisfaction of the customers. The costs resulting from excessively long queues are directly proportional to the degree to which they exceed the given maximum tolerated queue length M_i.

The costs of change are one-shot costs representing the amount of money expended for a factual change in the system settings. These change costs are added to the total costs of the stage that has been ended by the change.

The possible reactions of the manager of the system depend on the ratios of the constituent costs. If the queuing costs are several times higher than the idle costs (which means that staying in a queue that is considered to be long is undesirable, for example, because the surface of the varnished product will become dry and that makes further processing impossible: the products are spoiled), then there will be a tendency to maintain the lengths of the queues under the threshold level rather than take some action towards eliminating servers that are idle.

For simplicity, it is assumed that any change in the service times is performed exactly at one server by one time unit, and all more complex changes are performed as a series of such simple changes. With respect to the computational complexity, the change costs are considered to be constant.

Denote by $l_i(t)$ the length of the queue in front of service place i. The total production cost of a stage is then computed by the following formula.

$$P(T,j) = \sum_{t=1}^{T} \left(\sum_{i=1}^{n} W_i K_i + \sum_{\substack{i=1 \\ l_i(t) < \frac{K_i}{z_i}}}^{n} I_i \left(\frac{K_i}{z_i} - l_i(t) \right) + \sum_{\substack{i=1 \\ l_i(t) > M_i}}^{n} Q_i \left(l_i(t) - M_i \right) \right) + C_j \tag{4.2.3}$$

4.2.4 The Types of Systems

The deterministic concept of a system can be considered as a special case of the stochastic approach. With the use of this approach it is possible to compute the parameters of the system, and thus it is also possible to predict the system's future development.

Both the service times and the request flow can be of two types: continuous or discrete. Continuous service time means that the manager of the system can change the setting of the system by accelerating or decelerating the service times of the servers so that the condition $\sigma_i > 0$ is fulfilled. Considering the discrete service times approach, it is possible to add or take away another identical server to some service place in order that $K_i \geq 1$. A continuous request flow means that there is no need to wait until the whole request has been served: the served part of the request can leave the service place and fall in another queue. If the request flow is discrete, the request can fall into the next queue as soon as it is completed.

Thus four types of system can be distinguished:

1. a system with continuous service time and continuous request flow,
2. a system with continuous service time and discrete request flow,
3. a system with discrete service time and continuous request flow,
4. a system with discrete service time and discrete request flow.

System with continuous service times and continuous request flow

An example of such a system is processing the letters at a post office, where the requests are represented by packages of letters and at every time unit someone carries away the already processed letters from each worker. In case of need, workers can slow down or speed up the pace of work.

For this type of system it is supposed that there is one server at each specific service place, $K_i = 1$. Then $z_i = 1/\sigma_i$ in this type of system. The given service time σ_i can be changed within certain limits. In other words, the server can be accelerated or decelerated. Only a deceleration where $\sigma_i > 0$ can be taken into account: the server is still working (it is not possible to put the server out of service).

Because the served part of the request can fall into the next queue, the quantity of the already served part of the request can be computed.

Henceforth, the following notation will be used: the served quantity of request in service place i at time t is denoted by $f_i(t)$, and the number of requests present at the ith service place is denoted by m_i, $m_i \geq 0$. The remaining part of the request that is being served at service place i is denoted by s_i, $0 \leq s_i \leq 1$. The notation s_i will be used in Theorem 4.2.4 (for the systems with continuous service time with one server at each service place).

Theorem 4.2.3. *Consider a queuing system with continuous service times and continuous request flow. Then the parameters $f_i(t)$, $m_i(t)$ and $l_i(t)$ are computed as follows.*

$$f_i(t) = \min(m_i(t-1), z_i) \tag{4.2.4}$$

$$m_i(t) = m_i(t-1) - f_i(t) + f_{i-1}(t-1) \tag{4.2.5}$$

$$l_i(t) = m_i(t) - 1 \tag{4.2.6}$$

Proof. Formulas (4.2.4), (4.2.5) and (4.2.6) follow directly from the definition. □

Remark 4.2.1. The served quantity is described as a minimum of a number of requests in time $t-1$ and service intensity of ith server. Due to the continuity of the flow of requests, it can happen that the capacity of the server is not fully used, i.e., $0 < m_i(t-1) < z_i$. It can be seen that m_i, the number of requests, need not be an integer.

Remark 4.2.2. Having $l_i(t) \geq 0$ indicates that there is a queue, whereas $-1 < l_i(t) < 0$ indicates that the server is not fully used. But $l_i(t) = -1$ means that the server is not working.

A system with continuous service times and discrete request flow

An example of such a system is an assembly line manufacturing spare parts for cars. The spare part can move ahead to the next service place after it has been fully completed. The speed of the machines can be adjusted.

Similarly to the previous type of system, there is one server at each service place i in this type and the manager can change the service times of the servers. The request flow is discrete: the request remains in the service until it has been completed.

Theorem 4.2.4. *Consider a system with continuous service times and discrete request flow. The amount of requests served at service place i at time t is*

$$f_i(t) = \min(s_i(t-1), z_i) \tag{4.2.7}$$

The number of requests present at the service place i at time t, $m_i(t)$, and the value s_i are computed by the following recursion rules:

If $(s_{i-1}(t) = 1$ *or* $s_{i-1}(t) = 0)$ *and* $s_{i-1}(t-1) > 0$

If $(s_i(t) = 1$ *or* $s_i(t) = 0)$ *and* $s_i(t-1) > 0$

$m_i(t) = m_i(t-1);$ *else* $m_i(t) = m_i(t-1) + 1;$ *endif;*

else

If $(s_i(t) = 1$ *or* $s_i(t) = 0)$ *and* $s_i(t-1) > 0$

$m_i(t) = m_i(t-1) - 1;$ *else* $m_i(t) = m_i(t-1);$ *endif; endif.*

If $m_i(t) > 0$

If $s_i(t-1) = 0$

$s_i(t) := 1;$ *else* $s_i(t) := s_i(t-1) - z_i;$ *endif;*

else

$$s_i(t) := 0; \textit{ endif.} \tag{4.2.8}$$

Proof. These recursion rules follow from the properties of s_i and m_i. □

Theorem 4.2.5. *The length of the queue is*

$$l_i(t) = m_i(t) - 1 \tag{4.2.9}$$

Proof. The length of the queue is dependent on the number of working servers and there is only one server in each service place in this type of system. □

A system with discrete service times and continuous request flow

An example of such a system is similar to the first example of a system (with continuous service time and continuous request flow). The only difference is that in case of need, the manager can add or recall another clerk to some service place.

In the case of discrete service times, the manager can change the number of servers at the ith service place, K_i, by adding or taking away some additional server(s), preserving the condition $K_i \geq 1$. All the servers at service place i have the same service time σ_i.

The remaining part of the requests that are being served at service place i by the kth server is denoted by s_{i_k}, $0 \leq s_{i_k} \leq 1$. The notation s_{i_k} will be used in Theorems 4.2.6 and 4.2.7 (for the systems with continuous service time with K_i servers at each service place).

Theorem 4.2.6. *Consider a queuing system with discrete service times and continuous request flow. Then the parameters s_{i_k}, $f_i(t)$, $m_i(t)$ and $l_i(t)$ are computed according to the following formulas.*

$$\begin{aligned}
&\textit{If } m_i(t) > 0\\
&\quad \textit{For } k = 1 : K_i\\
&\qquad \textit{If } s_{i_k}(t-1) > 0\\
&\qquad\quad s_{i_k}(t) := s_i(t-1) - z_i; \quad \textit{endif;}\\
&\qquad \textit{If } s_{i_k}(t-1) \leq 0 \textit{ and } l_i > 0\\
&\qquad\quad s_{i_k}(t) := 1;\\
&\qquad\quad l_i(t) := l_i(t-1) - 1; \quad \textit{endif;} \quad \textit{endfor;}\\
&\textit{endif.}
\end{aligned} \tag{4.2.10}$$

$$f_i(t) = \sum_k \min(s_{i_k}(t-1), z_i) \tag{4.2.11}$$

$$m_i(t) = m_i(t-1) - f_i(t) + f_{i-1}(t-1) \tag{4.2.12}$$

$$l_i(t) = m_i(t) - \sum(s_{i_k}) \tag{4.2.13}$$

Proof. The value s_{i_k} is changed according to the service phase. The quantity of requests served, $f_i(t)$, is the minimum of the remaining part of the request and the service intensity because due to the continuity of the flow of requests, it can happen that $s_{i_k}(t-1) < z_i$ (the server is not fully used). The value $m_i(t)$ is computed with respect to the incoming and outgoing served parts of the requests. The length of the queue is the difference between the number of the present requests and the sum of the remaining parts at all servers in the service place. □

System with discrete service times and discrete request flow

An example of such a system is similar to the assembly line manufacturing spare parts, except that changing the settings of the system is different. To change the speed of the process in a service place, another identical machine is added or taken away.

Similarly to the previous type of system, the manager can change the number of servers in the service place i, however the request flow is discrete.

The remaining part of the request, s_{i_k}, is computed by rules 4.2.10.

Theorem 4.2.7. *Consider a system with discrete service times and discrete request flow. The values m_i, l_i are computed by the following rules:*

For $k = 1 : K_{i-1}$

If $(s_{i-1_k}(t) = 1$ *or* $s_{i-1_k}(t) = 0)$ *and* $s_{i-1_k}(t-1) > 0$

$x := x + 1;$ *endif; endfor;*

For $k = 1 : K_i$

If $(s_{i_k}(t) = 1$ *or* $s_{i_k}(t) = 0)$ *and* $s_{i_k}(t-1) > 0$

$y := y + 1;$ *endif; endfor;*

$$m_i(t) = m_i(t-1) + x - y. \tag{4.2.14}$$

For $k = 1 : K_i$

If $s_{i_k}(t) > 0$ *or* $s_{i_k}(t-1) > 0$

$o := o + 1;$ *endif; endfor;*

$$l_i(t) = m_i(t) - o. \tag{4.2.15}$$

Proof. The value $m_i(t)$ is computed with respect to incoming requests from the previous $(i-1)$ service place (variable x) and the completed requests in service place i (variable y). The length of the queue is calculated by subtracting the number of working servers (variable o) from the number of the requests present at the service place. □

4.2.5 Method Based on the Previous State Evaluation

There are two kinds of optimization. The first is the selection of the scheduling or control strategy and the second is in the determination of the system parameters. This and the following subchapter deal with the second type, the reduction of the costs of the system. For the examples and calculations, the system with continuous service times and continuous request flow is considered. The model of the system was designed using the language VBA and the tool MS Excel.

The first proposed method employs the Markovian property, i.e., the future development of the system's state depends on its current state. The past history has been completely summarized in the current state and the system has no memory, so that it is not known how the current state was reached.

The method uses the evaluation of the system states. The evaluation criteria take into account only the present state and the previous one (but not the sequence of preceding events).

The suggestions for the changes in system settings are given every time unit (the so-called turn), so there is no need to penalize a change of the system settings via the addition of change costs (in the calculation of the function of total production cost), hence $C_j = 0$. Also due to the continuity of the changes, each stage is of length $T = 1$.

Remark 4.2.3. Note that the length of the stage is adjusted to 1, therefore the turn and the stage are equivalent concepts for this method.

The following sequence of steps is performed at the end of each stage for each service place:

1. Compute the characteristics, i.e., the length of the queue $l_i(t)$, queue tendency $d_i(t)$, particular costs for the turn, total costs for the turn;

2. Weight the urgency of the change, the weights are denoted by $G_i(t)$;

3. Suggest strategies;

4. Choose the appropriate action;

5. Apply the changes.

The length of the ith queue is computed according to the type of the system: it depends on the above mentioned character of the requests. In general, the queue length at time (t) is equal to the queue length at time $(t-1)$ less the quantity of requests that have left the queue, plus the quantity of requests that have fallen into this queue.

The queue tendency, $d_i(t)$, reflects how the actual queue will change with respect to the current settings; it is dependent on $l_i(t)$, $l_{i-1}(t)$ and also on $\sigma_i(t)$ and $\sigma_{i-1}(t)$. Its value expresses the increase or decrease in the actual queue and also gives information about the intensity of this variation. If $d_i(t) > 0$, the queue is increasing, for $d_i(t) < 0$, it is decreasing, and $d_i(t) = 0$ means that the length of the queue will not change in the next turn. The computation of this parameter is again dependent on the character of the requests.

The value of the particular costs for a turn (considering idle and queuing costs) will be positive if either $l_i(t) < K_i(t)/z_i$ in case of idle costs, or $l_i(t) - M_i > 0$ in case of queuing costs. The total costs for the turn are then computed as the sum of the particular costs.

After the computation of the above mentioned basic characteristics has been made, the particular situations at the service places are weighted. The weights are expressed by a multiple of Q_i,I_i according to the importance of the situation. If the queue tends to grow and the limit M_i will be exceeded in u turns (the value depends on the need to provide prompt reactions upon changes in the system), then the weight $G_i(t) = Q_i \cdot x'$, where $x' \in \langle 0, 1\rangle$ is a coefficient expressing the urgency of the reaction. If the queue tends to grow and the limit M_i is already exceeded, then the weight is intensified

	$l_i(t) = 0$	$0 < l_i(t) < M_i$	$l_i(t) > M_i$
$d_i(t) > 0$	$Q_i \cdot x'$ do nothing	$Q_i \cdot x''$ accelerate	$Q_i \cdot (l_i(t) - M_i) + Q_i$ accelerate
$d_i(t) < 0$	$I_i \cdot y'$ acc. preceding	$I_i \cdot y''$ decelerate	$Q_i \cdot (l_i(t) - M_i)$ accelerate
$d_i(t) = 0$	$I_i \cdot z'$ acc. preceding	0 do nothing	$Q_i \cdot (l_i(t) - M_i)$ accelerate

Figure 4.6: Weights and strategies

by an addend: $G_i(t) = Q_i \cdot (l_i(t) - M_i) + Q_i$. Formulas for the weights of a queue that is tending to fall are constructed similarly. An overview of the formulas for the computation of $G_i(t)$ and suggested strategies (do nothing, accelerate z_i, accelerate the preceding (z_{i-1}) or decelerate) for different combinations of the values $l_i(t)$ and $d_i(t)$ are shown in Figure 4.6.

Remark 4.2.4. The coefficients in Table 4.6 should satisfy $x' < x''$ and $y' > y''$ (according to the urgency of the situations).

With the knowledge of the particular weights $G_i(t)$, it is possible to decide which of the situations is the most urgent: it is the situation with the highest weight, and the suggested strategy can then be implemented according to this comparison.

Example 4.2.1. *How the evaluation method works is illustrated in the following figures. A system with continuous service times and request flows is considered. During the computation of the particular and total costs, the work costs can be omitted, because these costs are proportional to the number of servers at the service place and for this type of system there is exactly one server at each service place. The initial parameters are set to* $z_i(0) = \{3, 5, 4, 3, 6, 6, 7, 6, 4, 4\}$, $l_i(0) = \{2, 5, 6, 7, 6, 6, 8, 7, 6, 6\}$, $M_i = 3$, $Q_i = I_i = 10$.

Figure 4.7 shows the 38*th step of the optimization and the values of the computed*

38	$z_i(38)$	$l_i(38)$	queue tendency	Idle costs	Queuing costs	Total costs for turn	$G_i(38)$	Stategy	Total	
1	3	2,							Total cumulative costs	10 893,5
2	2	5,65	-0,1667	0,	26,5	26,5	26,5	accelerate		
3	1	4,6	-0,5	0,	16,	16,	16	accelerate	Total for turn	127,6667
4	0,5	5,25	-1,	0,	22,5	22,5	22,5	accelerate		
5	2	6,55	1,5	0,	35,5	35,5	45,5	accelerate		
6	2	3,	0,	0,	0,	0,	0	do nothing		
7	1	3,6333	-0,5	0,	6,3333	6,3333	6,3333	accelerate		
8	1	2,6667	0,	0,	0,	0,	0	do nothing		
9	0,5	2,9	-1,	0,	0,	0,	7,5	deccelerate		
10	0,5	5,0833	0,	0,	20,8333	20,8333	20,8333	accelerate		

Figure 4.7: Cost optimization: stage 38

parameters. One of the queues tends to grow (the queue in front of the fifth server), some of them tend to fall, and some remain stable (the queues in front of servers $6, 8, 10$*). It can be observed that the greatest weight is* $G_5(38) = 45.5$*. This means that the service time at the fifth service place will be accelerated in the next turn because the maximum tolerable limit of the queue length (*$M_5 = 3$*) has already been exceeded (*$l_5(38) = 6.55$*) and the queue is tending to grow.*

The development of the service times during the optimization for $t = \{0, 1, \ldots, 21\}$ *and also the reaction of the queue lengths in time connected with costs are shown in Figure 4.8. The starting system settings can be seen in the column with index* 0*.*

Figure 4.9 depicts the evolution of the costs for a turn and the total costs (the total costs are connected to the secondary axis in the graph) in comparison to the evolution of the system costs of an unoptimized system. At the start, the costs are nearly the same, and then the optimization takes effect. It can be seen that after 67 *turns, the system reaches a state where the queues are of acceptable length (not too long and/or not empty) and thus do not create any undesirable costs.*

z_i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
2	5	5	5	5	5	5	5	5	5	5	5	5	5	5	4	3	3	3	3	3	3	3	3	3	3
3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
5	6	6	6	6	6	6	5	4	3	3	3	3	3	3	3	3	3	3	3	3	3	2	2	2	2
6	6	6	6	6	6	6	6	6	6	6	6	5	4	3	3	3	3	3	3	3	3	3	2	2	2
7	7	6	5	5	4	4	4	4	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2	1
8	6	6	6	5	5	4	4	4	4	4	3	3	3	3	3	3	3	2	2	2	2	2	2	2	2
9	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	3	3	2	2	2	2	2	2	2
10	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	3	2	2	2	2	2
$l_i(t)$																									
1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	5	5,13	5,27	5,4	5,53	5,67	5,8	5,93	6,07	6,2	6,33	6,47	6,6	6,73	6,82	6,82	6,82	6,82	6,82	6,82	6,82	6,82	6,82	6,82	6,82
3	6	5,95	5,9	5,85	5,8	5,75	5,7	5,65	5,6	5,55	5,5	5,45	5,4	5,35	5,35	5,43	5,52	5,6	5,68	5,77	5,85	5,93	6,02	6,1	6,18
4	7	6,92	6,83	6,75	6,67	6,58	6,5	6,42	6,33	6,25	6,17	6,08	6	5,92	5,83	5,75	5,67	5,58	5,5	5,42	5,33	5,25	5,17	5,08	5
5	6	6,17	6,33	6,5	6,67	6,83	6,97	7,05	7,05	7,05	7,05	7,05	7,05	7,05	7,05	7,05	7,05	7,05	7,05	7,05	7,05	6,88	6,72	6,55	6,38
6	6	6	6	6	6	6	6,03	6,12	6,28	6,45	6,62	6,75	6,83	6,83	6,83	6,83	6,83	6,83	6,83	6,83	6,83	7	7	7	7
7	8	8	7,97	7,93	7,85	7,77	7,68	7,6	7,52	7,35	7,18	7,05	6,97	6,97	6,97	6,97	6,97	6,97	6,97	6,97	6,97	6,97	7,13	7,13	6,63
8	7	7	7,03	7,03	7,08	7,08	7,08	7,08	7,08	7,17	7,17	7,17	7,17	7,17	7,17	7,17	7,17	7	6,83	6,67	6,5	6,33	6,17	6,17	6,67
9	6	5,92	5,83	5,78	5,73	5,73	5,73	5,73	5,73	5,73	5,82	5,9	5,98	6,07	6,15	6,23	6,23	6,4	6,4	6,4	6,4	6,4	6,4	6,4	6,4
10	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6,08	6,17	6,42	6,58	6,58	6,58	6,58	6,58	6,58
costs for turn																									
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	20	21,3	22,7	24	25,3	26,7	28	29,3	30,7	32	33,3	34,7	36	37,3	38,2	38,2	38,2	38,2	38,2	38,2	38,2	38,2	38,2	38,2	38,2
3	30	29,5	29	28,5	28	27,5	27	26,5	26	25,5	25	24,5	24	23,5	23,5	24,3	25,2	26	26,8	27,7	28,5	29,3	30,2	31	31,8
4	40	39,2	38,3	37,5	36,7	35,8	35	34,2	33,3	32,5	31,7	30,8	30	29,2	28,3	27,5	26,7	25,8	25	24,2	23,3	22,5	21,7	20,8	20
5	30	31,7	33,3	35	36,7	38,3	39,7	40,5	40,5	40,5	40,5	40,5	40,5	40,5	40,5	40,5	40,5	40,5	40,5	40,5	40,5	38,8	37,2	35,5	33,8
6	30	30	30	30	30	30	30,3	31,2	32,8	34,5	36,2	37,5	38,3	38,3	38,3	38,3	38,3	38,3	38,3	38,3	38,3	40	40	40	40
7	50	50	49,7	49,3	48,5	47,7	46,8	46	45,2	43,5	41,8	40,5	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	39,7	41,3	41,3	36,3
8	40	40	40,3	40,3	40,8	40,8	40,8	40,8	40,8	41,7	41,7	41,7	41,7	41,7	41,7	41,7	41,7	40	38,3	36,7	35	33,3	31,7	31,7	36,7
9	30	29,2	28,3	27,8	27,3	27,3	27,3	27,3	27,3	27,3	28,2	29	29,8	30,7	31,5	32,3	32,3	34	34	34	34	34	34	34	34
10	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30	30,8	31,7	34,2	35,8	35,8	35,8	35,8	35,8	35,8

Figure 4.8: The development of service times, queue length, and costs

4.2.6 Method Based on the Simulation of Future States

The system can be affected by the manager by changing the service intensity at some service place. The main reason is to reduce total costs and optimize the manufacturing process. The decision is divided into two parts. First, it must be decided at which moment the change in the service intensity will be done in order to minimize the average production costs. Second, it has to be decided which of the service intensities will be changed. If the queue at service place i is too long and is still increasing, then the service intensity z_i should be accelerated. This can be done, for example, by adding a parallel service machine at position i. On the other hand, if the server(s) in the service place i is (are) idle, then z_i can be diminished.

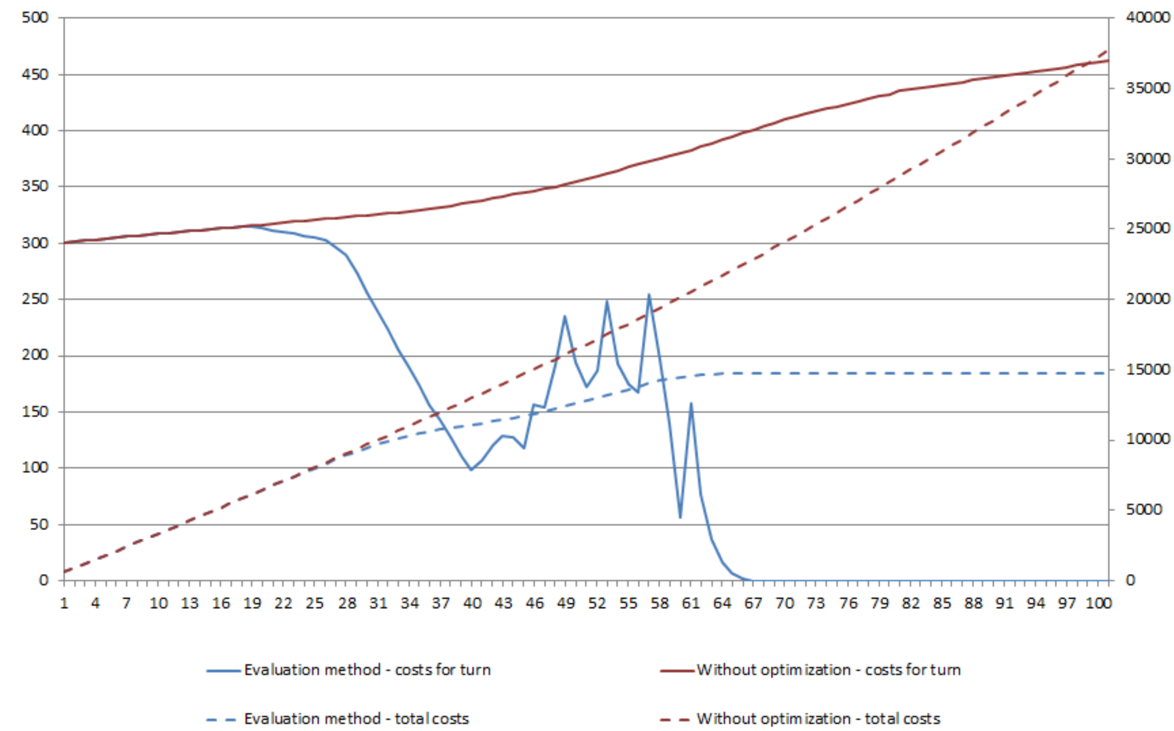


Figure 4.9: The evolution of costs

The second method is based on the simulation of the system's future development for several possible settings and on the computation of the average production costs for each of these simulations. Unlike the previous method, the suggestion to change the system settings is not given every time unit (for some types of systems it is undesirable to continually change the system settings), thus the length of the stage is $T \geq 1$. The intention is to let the system perform for some time period and influence it only in time when necessary, not too soon but also not too late, therefore also $C_j \geq 0$. For computational simplicity, the change is assumed to be provided at exactly one service place at one time instant.

Assuming that the stage will end at $T = t$, the average production costs during

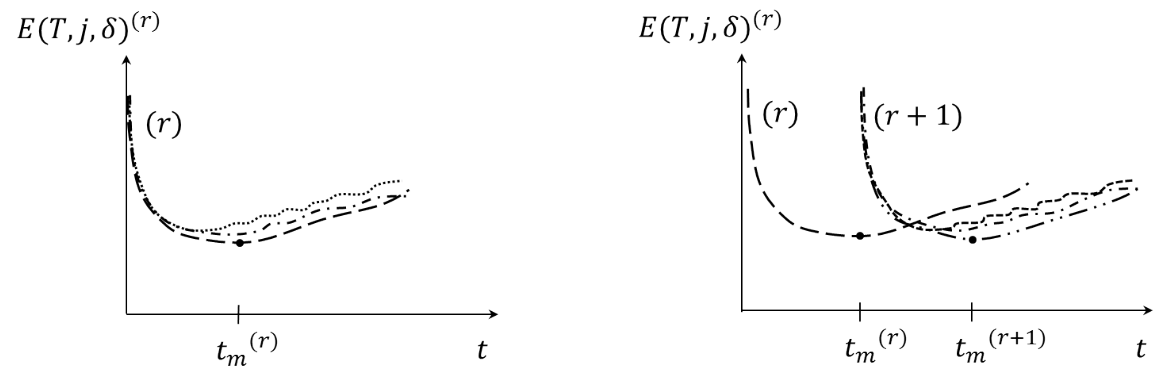


Figure 4.10: The decision process

the stage are computed according to the following formula:

$$E(T, j, \delta) = \frac{P(T, j)}{T}. \tag{4.2.16}$$

The parameter j indicates the service place where the change in the system settings is made, δ expresses the intensity of this modification (for example, $j = 5$ and $\delta = 3$ means that the change is considered at the fifth service place and in dependence on the type of service times described above, either the service time of the server is decelerated by $+3$ time units or the number of servers is increased by $+3$ units).

At the beginning of each stage, the rth set of functions $E(T, j, \delta)^{(r)} = \frac{P(T,j)}{T}$ is computed (in other words, the index r indicates the number of the stage). This set represents the evolution of the system's average production costs for all possible intended settings. The optimal setting is the one that corresponds to the function of the set which contains the global minimum of the rth set. This minimum is very important, because the time when this function reaches its minimum is the time convenient for the next change of the system settings (because from this moment on, the average costs are increasing) and the question, What will the change be like? will be answered by choosing the right function from the next, $(r+1)$th, set of functions.

The process is illustrated by Fig. 4.10. The curves of the graphs are not smooth.

This is a result of the variation in the development of the queues. The functions marked as (r) (the left part of the figure) represent the rth stage and were computed by calculating the function of average costs if the stage ends aat time T for each possible change. The result of the first step of the process is the time $t_m^{(r)}$ (global minimum of the set of functions), which answers the question When to make the change? because the continued work of the system will lead to the growth of the average production costs.

Remark 4.2.5. For the first stage it holds that the $E(T, j, \delta)^{(1)}$ is the one and only one because it was assumed the change costs are constant.

Example 4.2.2. *The method is again illustrated on an example of a system with both the service times and request flows continuous. The initial parameters are set to* $\sigma_i(0) = \{3, 5, 4, 3, 6, 6, 7, 6, 4, 4\}$*,* $l_i(0) = \{2, 5, 6, 7, 6, 6, 8, 7, 6, 6\}$*,* $M_i = 3$*,* $I_i = Q_i = 10$*,* $C_i = 30$*. For simplicity, the changes in service times are considered to be unit changes (*$\delta = \pm 1$*).*

The time evolution of the costs for 100 *turns are depicted in Figure 4.11. The graph shows the costs for the turn and the total costs in comparison to the evolution of costs of the system without optimization (total costs are connected to the secondary axis in the graph).*

The graph also shows the average production costs if the stage ends at T *(purple one with marks). This function consists of parts of the particular curves related to the chosen settings during the optimization process. During the first stage, the function reaches its minimum after* 28 *turns. At* $t = 29$*, the next set of functions was computed and one of the settings was chosen. The new stage began. The part of the curve related to the chosen setting corresponds to the part of the purple curve for* $29 < t < 39$*. Again, for* $t = 40$ *the next set of functions was computed.*

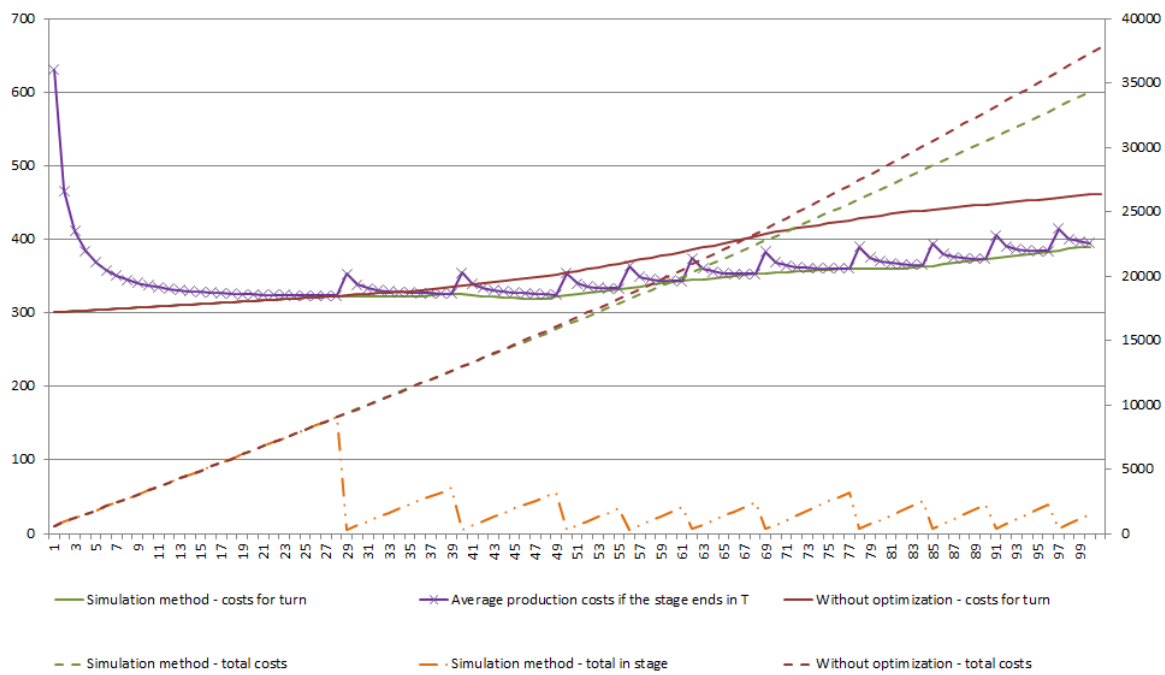


Figure 4.11: The evolution of costs

5. Steady States of a System

The second part of the main results of the thesis are the description of the steady states of a complex system using extremal algebras.

5.1 Steady States of a DES in Max-drast Algebra

In this chapter, a summary of the research results about the stable states of a DES using max-drast algebra will be given. The results of this research have been already published in [Gavalec and Němcová, 2012]. The drastic triangular fuzzy norm is used, the basic terms were explained in Chap. 3.3.3.

Let n be a fixed natural number with notation $N = \{1, 2, \ldots, n\}$. Let $A \in \overleftarrow{I}(n, n)$ be a fixed matrix, the entries of A are denoted by $a_{ij} \in \overleftarrow{I}$, $i, j \in N$.

If A is interpreted as the reliability matrix of the transitions in a discrete time fuzzy system, then the drastic norm in the computation of the matrix powers reflects extreme demands on the reliability of the system. This follows from the computation of the powers of A. More precisely, a zero entry $a_{ij} = 0$ of A means that the reliability of the transition from state i to state j in the system is zero. If $a_{ij} = 1$, then the transition between states i and j is reliable. On the other hand, if a_{ij} lies between zero and one, i.e., $a_{ij} = u$, $u \in (0, 1)$, then some uncertainty in the transition expressed by the value u must be considered. The smaller this value, the higher the degree of uncertainty.

The idea of reliability of an edge (a one-step transition) can be extended to the reliability of a path in the characteristic weighted digraph $G(A)$ of A (a several-step transition). The nodes of $G(A)$ correspond to the states of the system and are denoted by numbers in N, while an edge from node i to node j has the weight $w(i,j) = a_{ij}$, for all $i,j \in N$. Getting from state i to state j reliably means that all necessary steps are reliable (weighted by 1). If there is one uncertain step (having a positive weight smaller than 1) in the sequence of steps, then the sequence is considered as uncertain. Extreme demands on the reliability of a system imply that any sequence containing two or more uncertain steps is considered as inadmissible.

Formally, a path is called *strong* if all edges of the path are reliable (weighted by 1). The path is called *weak* if and only if exactly one edge of the path is weighted by $u \in (0,1)$ and the weight of the remaining edges is 1. Finally, the path is called *inadmissible* if more than one edge in the path is weighted less than 1. In other words, there is more than one uncertainty on the path from node i to j, or if all edges in the path have zero weight. The set of all strong (weak) paths is denoted by SR (WR). The notation $SR(i,j,r)$ ($WR(i,j,r)$) stands for the set of all paths in SR (WR) from i to j of length r.

5.1.1 Matrix Powers in Max-drast Algebra

Definition 5.1.1. Let $i,j \in N$ and let $p = (i_0, i_1, \ldots, i_r)$, with $i_0 = i, i_r = j$ be a path in $G(A)$. The weight of path p in max-drast algebra is set to the drastic fuzzy product

$$w_d(p) := \bigotimes_d \left\{ a_{i_{s-1}i_s};\ s = 1,2,\ldots r \right\} \ . \tag{5.1.1}$$

Denote the entries of the rth power, A^r, of A by a_{ij}^r. The interpretation of this value is given in the following proposition.

Proposition 5.1.1. *Let* $A \in \overleftarrow{I}(n,n)$, $i,j \in N$, *and let* r *be a natural number. Then the entries of* A^r *have values given by*

$$a_{ij}^r = \bigoplus \{ w_d(p);\ p \text{ is a path in } G(A) \text{ from node } i \text{ to node } j \text{ of length } r \} \ . \tag{5.1.2}$$

Proof. The assertion is proved easily by induction on r, using Definition 5.1.1 and matrix multiplication in max-drast algebra. □

A more specific formula for the values of elements in rth matrix power in max-drast algebra is presented in the next proposition.

Proposition 5.1.2. *Let* $A \in \overleftarrow{I}(n,n)$, $i,j \in N$ *and let* r *be a natural number. Then the entries of* A^r *are described by*

$$a_{ij}^r = \bigoplus \{ w_d(p);\ p \in SR(i,j,r) \cup WR(i,j,r) \} \ . \tag{5.1.3}$$

Proof. The assertion follows from Definition 5.1.1 and Proposition 5.1.1. Clearly, by the definition of the operation $\otimes_{\mathrm{d}}$, the weight of every inadmissible path p is zero. If the set in Equation (5.1.3) is empty, then the result of the operation $\bigoplus$ is equal to zero, which corresponds to the fact that there are neither strong nor weak paths from i to j in this case, while the inadmissible paths have zero weights. □

Proposition 5.1.3.

$$w_d(p) > 0 \quad \text{if and only if} \quad p \in SR(i,j,r) \cup WR(i,j,r) \tag{5.1.4}$$

$$a_{i,j}^r > 0 \quad \text{if and only if} \quad \exists p \in SR(i,j,r) \cup WR(i,j,r) \tag{5.1.5}$$

Proof. The assertion follows directly from the previous considerations. □

The last proposition in this section describes the conditions under which the weight of a path in max-drast algebra is equal to 1 or to some value u with $0 < u < 1$.

Proposition 5.1.4. *For the weight of a path p between nodes i and j of length r the following assertions hold:*

$$w_d(p) = 1 \quad \text{if and only if} \quad p \in SR(i, j, r) \tag{5.1.6}$$

$$0 < w_d(p) < 1 \quad \text{if and only if} \quad p \in WR(i, j, r) \tag{5.1.7}$$

Proof. This proposition results from the definition of strong and weak paths. The weights of all edges in a strong path are 1, hence the weight of such a path is 1. Similarly, the weight of a weak path (with exactly one weak edge and all the others strong) is the least element $u \in (0, 1)$. $\square$

In the next section, the subdigraphs of $G(A)$ are considered, and referred to as *threshold digraphs*, which will be denoted by $G(A, h)$, where $h \in \langle 0, 1\rangle$ is a threshold level. In the threshold digraph $G(A, h)$, such edges are only included if their weight is equal to or greater than h. The strongly connected components of $G(A, h)$ will be also considered. By a standard definition, a strongly connected component K is a subset of nodes in the digraph such that any two nodes i, j are contained in a common cycle.

A component K is called non-trivial if there is at least one cycle of positive length in K. The set of all non-trivial strongly connected components of the digraph $G(A, h)$ will be denoted by $\mathrm{SCC}^*(G(A, h))$. For $K \in \mathrm{SCC}^*(G(A, h))$, the component period $\mathrm{per}(K)$ is defined as the greatest common divisor of the lengths of all cycles in K.

5.1.2 Computation of the Matrix Period

Max-min and max-drast algebra differ in their interpretation. A max-min matrix describes the flow capacities, while a max-drast matrix concerns the reliability of the transitions in the system.

Similarly, as in max-min algebra, by max-drast operations no new elements (except 0) are created. As a consequence, the matrices in the power sequence of A in max-drast algebra only contain the entries from A.

The above mentioned reliability of the path is represented by powers of A. The paths of lengths that are even much longer than the number of edges in the digraph of A can be investigated. This means computing high powers of A. As a consequence of the repetition of elements and the possibility of creation only the zero element in the matrix, sooner or later there must occur a period in the power sequence of A, $\mathrm{per}(A)$.

The computation of the length of a matrix period in max-min algebra has been described in [Gavalec, 1997]. The matrix period in max-min algebra is computed by the following formula.

$$\mathrm{per}_{\min}(A) = \mathrm{lcm}\big\{\mathrm{per}(K);\ K \in \mathrm{SCC}^*(G(A,h)), h \in \langle 0,1\rangle\big\}\ . \tag{5.1.8}$$

In the computation of the matrix period in max-drast algebra, only the level $h = 1$ must be considered.

Theorem 5.1.5. *Let* $A \in \overleftarrow{I}(n,n)$. *Then*

$$\mathrm{per}_{\mathrm{drast}}(A) = \mathrm{lcm}\big\{\mathrm{per}(K);\ K \in \mathrm{SCC}^*(G(A,1))\big\}\ . \tag{5.1.9}$$

Proof. Write $d^* = \mathrm{lcm}\,\mathrm{per}(K)$, where $K \in \mathrm{SCC}^*(G(A,1))$. Also denote the period of matrix A in max-drast algebra by $d_A = \mathrm{per}_{\mathrm{drast}}(A)$: this is the least d such that $\exists R, \forall r > R : A^r = A^{r+d}$. Determine the period of the elements of the matrix powers as $d_{i,j} = \mathrm{per}_{\mathrm{drast}}(A_{i,j})$. This is also the least d such that $\exists R, \forall r > R : a^r_{i,j} = a^{r+d}_{i,j}$. Write $d_A = \mathrm{lcm}\{d_{ij}, i,j \in N\}$. In this notation the assertion of the theorem can be simply written as $d_A = d^*$.

The proof will be done in two steps. In step (i) it will be proved that $d_A | d^*$, and in step (ii) that $d^* | d_A$.

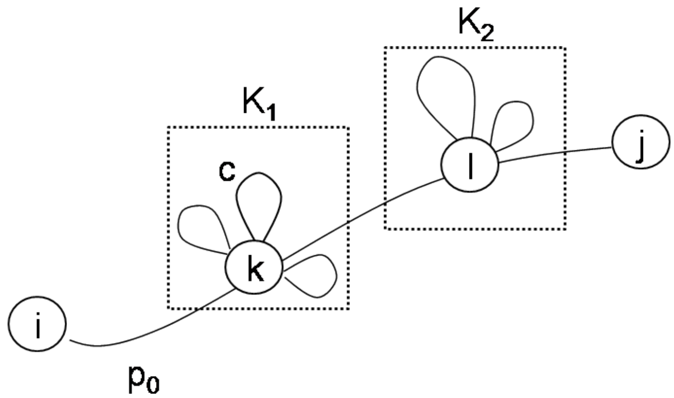


Figure 5.1: The path p_0

(i) We have that $\operatorname{lcm} d_{i,j}|d^*$ if and only if $(\forall i,j) : d_{i,j}|d^*$, it is clear that if the least common multiple (e.g., $\operatorname{lcm} = a \cdot b$) divides some number, then also each specific component of the multiple (e.g., a) divides the number. Consider non-trivial $K \in \mathrm{SCC}^*(G(A,1))$ and $p_0 \in SR(i,j,r) \cup WR(i,j,r)$, a path of length r between i and j which goes through the strongly connected components K (see Fig. 5.1).

Then $a^r_{i,j}(p_0)$ can be found such that it is equal to some $a^{r+d}_{i,j}(p_0)$. Then the weight of such a path is $w(p_0 + C) = w(p_0) \otimes_d w(C) = w(p_0)$, where C is a certain combination of strong cycles in the component $C \subseteq K$. Because $d_{i,j}(p_0)$ is a period dependent on the greatest common divisor of K, $d_{i,j}(p_0)|\operatorname{per}(K)|d^*$. The elements of the rth power of A are then $a^r_{i,j} = \max\{a^r_{i,j}(p_0)\}$. If $d_{i,j}(p_0)|d^*$, then also $d_{i,j}|d^*$. Write $d^*_{i,j} = \operatorname{lcm} d_{i,j}(p_0)$. Let d be such that $\forall p_0 : a^r_{i,j}(p_0) = a^{r+d}_{i,j}(p_0)$, then also $a^r_{i,j}(p_0) = a^{r+d^*_{i,j}}_{i,j}(p_0)$ and furthermore, $a^r_{i,j} = a^{r+d^*_{i,j}}_{i,j}$. As was proved, $d_{i,j}|d^*$. $d^*_{i,j} = \operatorname{lcm} d_{i,j}(p_0)$ and thus $d^*_{i,j}|d^*$. This results in $d_{i,j}|d^*_{i,j}|d^*$.

(ii) It has to be proved that $d^*|d_A$, i.e., $\operatorname{lcm}\big\{\operatorname{per}(K), K \in \mathrm{SCC}^*(G(A,1))\big\}|d_A$. Let $K \in \mathrm{SCC}^*(G(A,1))$, $i \in [K]$. It is claimed that $\operatorname{per}(K)|d_{i,i}$, which implies $\operatorname{per}(K)|d_A$, i.e., $d^*|d_A$.

Proof of the claim: Let d be such that $a^r_{i,i} = a^{r+d}_{i,i}$ for all $r > R$. Then take an

r such that $\mathrm{per}(K)|r$. Then also $\mathrm{per}(K)|r+d$ and it is clear that also $\mathrm{per}(K)|d$. As $d_{i,i}$ is such a d that $a^r_{i,i} = a^{r+d}_{i,i}$, then $\mathrm{per}(K)|d_{i,i}$. Now $d_A = \mathrm{lcm}\, d_{i,i}$, hence $\mathrm{per}(K)|d_{i,i}|d_A$. □

From the computation of the drast operation it can be seen that zero is the result of the case where $\max(x, y) < 1$. This result is thus insensitive to the variability of the elements x and y, which are less than one.

Theorem 5.1.6. *Let $A \in \overleftarrow{I}(n, n)$. If all elements of A, denoted by $u \in (0, 1)$, are replaced by one constant value $c \in (0, 1)$, then the period of matrix power in max-drast will not be changed.*

Proof. The reliability of a path of r steps from node i to node j, $a^r_{i,j}$, is the maximum of the paths weighted by the drast operation. Then $a^r_{i,j}$ is equal to the value of the most reliable path of length r. For the computation of d, the period of the element $a_{i,j}$, consider the $p_0 \in \mathit{WR}(i, j, r) \cup \mathit{SR}(i, j, r)$ which goes through the strongly connected components K (see Fig. 5.1). If $p_0 \in \mathit{WR}(i, j, r)$, then the weight of the path is changed from u to c without a change in the period of the element. On the other hand, if the path $p_0 \in \mathit{SR}(i, j, r)$, then two possibilities can arise. The first is that for the computation of the element $a^{r+1}_{i,j}$ a strong cycle is used, in which case $a^r_{i,j} = a^{r+1}_{i,j} = 1$. The second is that for the computation of the element $a^{r+1}_{i,j}$ a weak cycle is used, in which case $a^{r+1}_{i,j} = c$. In either case, at least one of the elements $a^{r+l}_{i,j}, l < d$ will be equal to 1, therefore the period can not be changed. □

Theorem 5.1.7. *Let $A \in \overleftarrow{I}(n, n)$. The matrix power periods in max-min and max-drast are equal if and only if the edges on level $h \in u$ do not create a new component C such that*

$$\mathrm{lcm}\big\{\mathrm{per}(K);\ K \in \mathrm{SCC}^*(G(A, 1))\big\}|\,\mathrm{per}(C)\ . \tag{5.1.10}$$

Proof. This results from the computation of the matrix powers periods in both algebras. In max-drast algebra only the level $h = 1$ must be considered, while in max-min algebra, periods of all non-trivial strongly connected components K at all levels can influence the matrix power period by the final computation of the least common multiple of $\mathrm{per}(K)$ through all levels. It is clear that by an increase of the quantity of cycles in some component K_i, the greatest common divisor of this cycles can decrease. The values of the periods differ only when at some level $h \neq 1$ there appears a K_i whose $\mathrm{per}(K_i)$ is not the divisor of a matrix period in max-drast. Therefore this $\mathrm{per}(K_i)$ must be included into the computation of the least common multiple and will change the matrix power period in max-min. □

Theorem 5.1.8. *Let* $A \in \overleftarrow{I}(n,n)$. *Then*

$$\mathrm{per}_{\mathrm{drast}}(A) | \, \mathrm{per}_{\mathrm{min}}(A) \ . \tag{5.1.11}$$

Proof. The calculation of the matrix period in both algebras differs in the final computation of the least common multiple. Hence the period in max-drast is a divisor of the period in max-min. □

Theorem 5.1.7 shows the case when the period in max-min equals that in max-drast. The following example (see A below and the corresponding threshold graphs in Fig. 5.2) represents a case when the periods computed in the two algebras differ.

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0,8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0,8 \\ 0 & 0 & 0 & 0 & 0 & 0,8 & 0 & 0 \end{pmatrix} \tag{5.1.12}$$

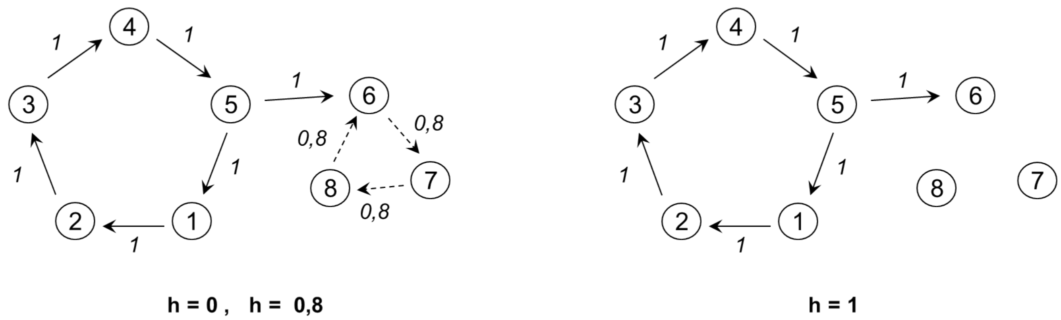


Figure 5.2: Threshold graphs

The computation of the period in max-min algebra covers all threshold levels $h = \{0,8;1\}$. At level $h = 0,8$ there can be found two non-trivial strongly connected components $K_1 = \{1,2,3,4,5\}$ and $K_2 = \{6,7,8\}$. The period of the first component, K_1, is equal to the greatest common divisor of the cycles in this component. Because there is only one such cycle, $\mathrm{per}(K_1) = 5$. Similarly, $\mathrm{per}(K_2) = 3$. At level $h = 1$ there is only one strongly connected component $K_3 = \{1,2,3,4,5\}$, $\mathrm{per}(K_3) = 5$. Then the matrix power period in max-min algebra is $\mathrm{per}_{\min}(A) = \mathrm{lcm}\{\mathrm{per}(K_1), \mathrm{per}(K_2), \mathrm{per}(K_3)\} = \mathrm{lcm}\{5,3,5\} = 15$.

In max-drast algebra only $h = 1$ must be considered: there is exactly one non-trivial strongly connected component at this level, $K_4 = \{1,2,3,4,5\}$. The period of this component is also the greatest common divisor of the cycles in the component: $\mathrm{per}(K_4) = 5$, so $\mathrm{per}_{\mathrm{drast}}(A) = \mathrm{lcm}\{\mathrm{per}(K_4)\} = \mathrm{lcm}\{5\} = 5$.

5.2 Steady States of a DES in Max-Łukasiewicz Algebra

This chapter summarizes a part of the results published in the journal *Fuzzy Sets and Systems*; for a full overview see [Gavalec et al., 2015].

Moreover, this chapter introduces a systematic characterization of the eigenspaces in a max-Łukasiewicz fuzzy algebra. The paper describing this idea, [Gavalec and Němcová, 2016], has also been sent to the journal *Fuzzy Sets and Systems* and is currently in the review process.

The study of discrete event systems with the use of extremal algebras is inspired by recent successes in max-plus (the so-called tropical) algebra. The operation of maximum together with the Łukasiewicz operation, defined in Chapter 3.3.3, forms the so-called max-Łukasiewicz algebra. A max-Łukasiewicz semiring is defined over the interval $[0, 1]$. The operations of addition and multiplication are extended to the vectors and matrices in a formal way. A certain similarity between max-Łukasiewicz and tropical algebras can be observed. The following will show that it is possible to translate problems between these two algebras.

The state of a system at time t can be described by a state vector, say $x(t)$. The transition matrix, denoted by A, describes the transitions of the system from one state to another. By the multiplication of the transition matrix with the state vector, the next state of the system, $x(t+1)$, is obtained; it can be written $A \otimes_L x(t) = x(t+1)$. During the operation of the system, after some time, it can happen that the system reaches a steady state. In a max-Łukasiewicz fuzzy algebra, the state vectors of the steady states correspond to the eigenvectors of the transition matrix A.

The investigation of the steady states of the system leads to the study of the set of all basic eigenvectors (connected to some specific eigenvalue): the eigenspace. Unlike a max-plus algebra, in a max-Łukasiewicz algebra, there need not be only one eigenvalue–eigenspace pair. In connection with the graph representation, see Fig. 5.3, the Łukasiewicz norm can represent, for example, the account balance of some agent travelling in a network. Suppose that the agent is given one unit of money before entering the system. In the left part of the figure there can be seen the quantities c_{ij}, the travel costs of moving from one node to another. The right part of the figure

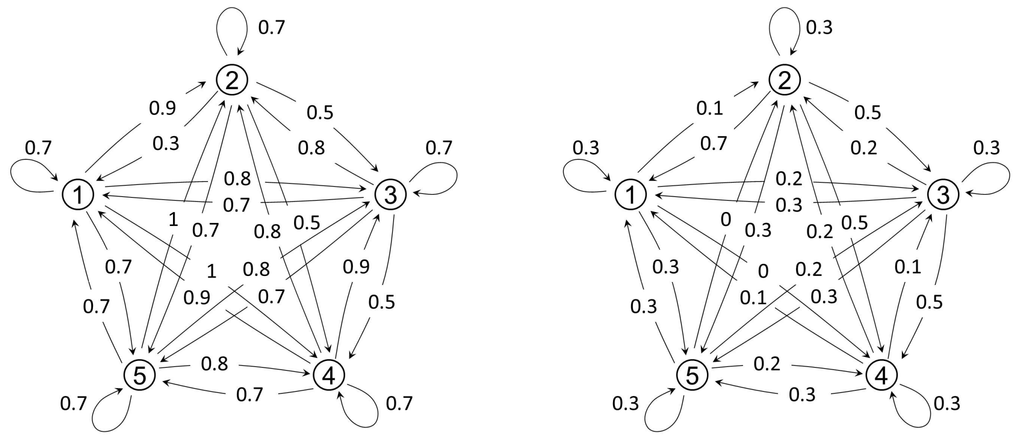


Figure 5.3: An example of an aggressive network: values of travel costs (left) and account balance after one step (right)

depicts the quantity $a_{ij} = 1 - c_{ij}$: the money that will be left in the bank account after moving from node i to node j. If the agent decides to move from node 3 to node 2, the left part of the figure shows that the cost of that journey is 0.8 unit. The right part shows that the account balance will be 0.2 unit. Moreover, when the agent decides to move from node 3 through 2 to 1, then the mechanism of the Łukasiewicz norm shows more clearly. The account balance (of course, computed from the right part of the figure) is now $\max(0.2+0.7-1; 0) = 0$, the agent has drawn all the money from the account. There is no other possible move in the network, the game is over. The impulse to study the eigenproblem in max-Łukasiewicz algebra $A \otimes_L x = \lambda \otimes_L x$ comes from the need to control the dynamics of the agent's funds. In other words, to know precisely when the game is over.

As has been mentioned above, max-Łukasiewicz linear algebra is closely related to tropical linear algebra. A key observation relating Łukasiewicz linear algebra with tropical linear algebra is that the product of the given matrix A and a vector x in Łukasiewicz algebra can be rewritten by using the operation of addition and the operation of maximum in a following way:

$$A \otimes_L x = A^{(1)} \otimes x \oplus \mathbf{0}, \tag{5.2.1}$$

where $A^{(1)}$ denotes the matrix with entries $(a_{ij} - 1)$, and $\mathbf{0}$ is the vector with entries which all equal to 0. Note that the entries of $A^{(1)}$ belong to $[-1, 0]$, while the entries of x are required to be in $[0, 1]^n$.

The max-Łukasiewicz eigenproblem can then be rewritten as

$$A^{(1)} \otimes x \oplus \mathbf{0} = (\lambda - 1) \otimes x \oplus \mathbf{0}, \quad 0 \leq x_i \leq 1. \tag{5.2.2}$$

Some examples with illustrative graphs and pictures are given for two- and three-dimensional matrices. Generating a graphical representation for higher dimensions does not, unfortunately, result in anything clearly arranged. For two (three) dimensions, the eigenspace can be clearly depicted without the need to have an interactive environment as would be the case for four or more dimensions. In those cases, the computation of the characteristics is more complex, but analogous, and will also be outlined in the further text.

5.2.1 Eigenspaces of a Two-Dimensional Matrix

Let us consider a two-dimensional matrix A, a vector x, and an eigenvalue λ. The eigenproblem in the max-Łukasiewicz algebra can be written as

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \otimes_L \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \lambda \otimes_L \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \tag{5.2.3}$$

and converted to tropical algebra

$$\begin{pmatrix} a_{11} - 1 & a_{12} - 1 \\ a_{21} - 1 & a_{22} - 1 \end{pmatrix} \otimes \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \oplus \begin{pmatrix} 0 \\ 0 \end{pmatrix} = (\lambda - 1) \quad \otimes \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \oplus \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

To solve the eigenproblem, two equations can be identified and solved. The first equation is

$$(a_{11} + x_1 - 1) \ \vee \ 0 \ \vee \ (a_{12} + x_2 - 1) \ \vee \ 0 = (\lambda + x_1 - 1) \ \vee \ 0. \tag{5.2.4}$$

To simplify this equation and prepare it for further use, the element $(1 - \lambda)$ can be added to both sides of the equation (with the effect of separating the variables on the right side of the equation):

$$(a_{11} + x_1 - \lambda) \ \vee \ (1 - \lambda) \ \vee \ (a_{12} + x_2 - \lambda) \ \vee \ (1 - \lambda) = x_1 \ \vee \ (1 - \lambda). \tag{5.2.5}$$

Proposition 5.2.1. *For $x_1 \in (1 - \lambda, 1\rangle$, Equation (5.2.5) holds if and only if either*

$$a_{11} = \lambda \ \wedge \ a_{12} - \lambda + x_2 \leq x_1 \tag{5.2.6}$$

or

$$a_{11} \leq \lambda \ \wedge \ a_{12} - \lambda + x_2 = x_1. \tag{5.2.7}$$

Proof. It is easily seen that if $(1-\lambda) < x_1 \leq 1$, then $(a_{11}+x_1-\lambda) \ \vee \ (1-\lambda) \ \vee \ (a_{12}+ x_2 - \lambda) \ \vee \ (1 - \lambda) = x_1$. This equation is satisfied if and only if $(a_{11} + x_1 - \lambda) = x_1$ and the other terms of the equation are less than or equal to x_1, or $(a_{12}+x_2-\lambda) = x_1$ and the other terms are less than or equal to x_1. □

Proposition 5.2.2. *For $x_1 \in \langle 0, 1 - \lambda\rangle$, Equation (5.2.5) holds if and only if*

$$x_1 \leq 1 - a_{11} \ \wedge \ x_2 \leq 1 - a_{12}. \tag{5.2.8}$$

Proof. If $0 < x_1 \leq (1-\lambda)$, then $(a_{11}+x_1-\lambda) \vee (1-\lambda) \vee (a_{12}+x_2-\lambda) \vee (1-\lambda) = (1-\lambda)$ which, in turn, is true if and only if both terms $(a_{11} + x_1 - \lambda)$ and $(a_{12} + x_2 - \lambda)$ are less than or equal to $(1 - \lambda)$. □

Similarly, the second equation of the eigenproblem is

$$(a_{21} + x_1 - \lambda) \ \vee \ (1 - \lambda) \ \vee \ (a_{22} + x_2 - \lambda) \ \vee \ (1 - \lambda) = x_2 \ \vee \ (1 - \lambda). \tag{5.2.9}$$

Proposition 5.2.3. *For $x_2 \in (1-\lambda, 1\rangle$, Equation (5.2.9) holds if and only if either*

$$a_{22} = \lambda \ \wedge \ a_{21} - \lambda + x_1 \leq x_2 \tag{5.2.10}$$

or

$$a_{22} \leq \lambda \ \wedge \ a_{21} - \lambda + x_1 = x_2. \tag{5.2.11}$$

Proof. Again, if $(1-\lambda) < x_2 \leq 1$, Equation (5.2.9) is true if and only if $(a_{21}+x_1-\lambda) = x_2$ and the other terms of the equation are less than or equal to x_2, or $(a_{22}+x_2-\lambda) = x_2$ and the other terms are less than or equal to x_2. □

Proposition 5.2.4. *For $x_2 \in \langle 0, 1-\lambda\rangle$, Equation (5.2.5) holds if and only if*

$$x_2 \leq 1 - a_{22} \ \wedge \ x_1 \leq 1 - a_{21}. \tag{5.2.12}$$

Proof. If $0 < x_2 \leq (1-\lambda)$, then both $(a_{21} + x_1 - \lambda)$ and $(a_{22} + x_2 - \lambda)$ have to be less than or equal to $(1-\lambda)$. □

The solution set of (5.2.3) is tropically convex. It can be said that it is the tropical convex hull defined by a finite number of points. That is, if the set contains x and y, then the tropical segment joining x and y also belongs to the set. Examples of tropical segments between three points are shown in the left part of Figure 5.4. A tropical segment is a combination of vertical, horizontal, and diagonal segments (for details, see [Nitica and Singer, 2007]). The right part of the figure depicts the tropical convex hull defined by three points.

Now the solution set for the first equation can be pictured, (5.2.5), using some of the above written conditions. Let's say for the entries $a_{11} < a_{12} < \lambda$ the conditions (5.2.7) and (5.2.8) have to be considered. Figure 5.5 shows the solution set for the selected entries. If $x_1 > (1-\lambda)$, then $a_{12} - \lambda + x_2 = x_1$, because $a_{12} - \lambda < 0$, and so the solution of the equation is the straight segment above the diagonal. For the cases where $x_1 \leq (1-\lambda)$, the condition (5.2.8) is considered. The solution set of the first equation is then the union of the solutions of the two above mentioned domains.

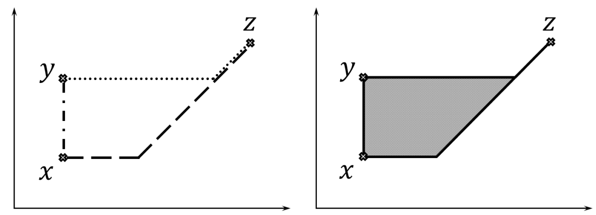


Figure 5.4: Tropical segments and tropical convex hull for the case of three points x, y, z.

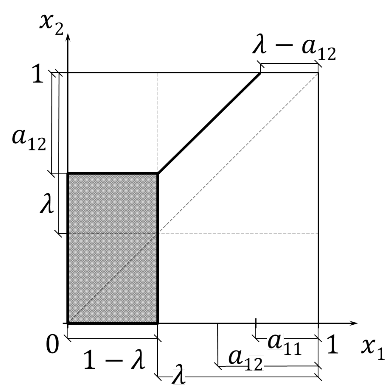


Figure 5.5: The solution set of the first equation in case $a_{11} < a_{12} < \lambda$

Along the same lines, it is possible to construct the solution set for other possible positions of the parameter λ and the variables from the equation. Although there are six possibilities for placing λ, three special types of shapes of solution sets for each of equations can be observed. These types depend on the relative positions of λ and the diagonal entries of the matrix. For the two-diagonal example, the entries of the first column of A have an influence on the “height” of the solution set, whereas the entries of the second column influence the “width” of the solution set.

Figure 5.6 shows for each equation three types of solution sets. The numbers in the notation indicate whether it is the solution set for the first or the second equation, and the Greek letter indicates the type. The types are generally influenced by the position of λ and the diagonal entry of the matrix. For the ith equation, the solution set is of type

- α if and only if $a_{ii} < \lambda$
- β if and only if $a_{ii} = \lambda$
- γ if and only if $a_{ii} > \lambda$

Note that the values of the particular entries are denoted in the picture by a complement to the number one, for example, $\overline{a_{ij}} = 1 - a_{ij}$. Possible changes in the shape of a particular solution set, in dependence on the size of a_{ij}, are outlined by the values $\overline{a_{ij}^*}$ and $\overline{a_{ij}'}$.

The solution sets, denoted by 1α and 2α, are similar to each other in terms of their shapes: a rectangle connected to a segment of solutions in the upper right corner. In the case of type 1α, the parameter $1 - a_{12}$ influences the height of the rectangle and the value $1-\lambda$ influences its width. The limits for the case 2α are symmetric: $1-a_{21}$ influences the width and $1 - \lambda$ influences the height of the solution set.

In the case of type β, the width of the solution set 1β in the picture is the maximal possible, that is, x_1 can be any value from $\langle 0, 1\rangle$. The height is influenced by the value $1 - a_{12}$ and the point of intersection with λ. Symmetrically, for 2β one has that $x_2 \in \langle 0, 1\rangle$ and the width of the solution set is influenced by $1 - a_{21}$ and the intersection point with λ.

It can be seen from the figure that the third type, γ, is the simplest one. The value $1 - a_{12}$ influences the height and the value $1 - a_{11}$ influences the width of the solution set 1γ. Similarly for the type 2γ, the value $1 - a_{22}$ influences the height and the value $1 - a_{21}$ influences the width of the solution set.

The solution set of the eigenproblem is then the intersection of the solution sets of the particular equations. Two numerical examples with graphical representations of the solution sets follow.

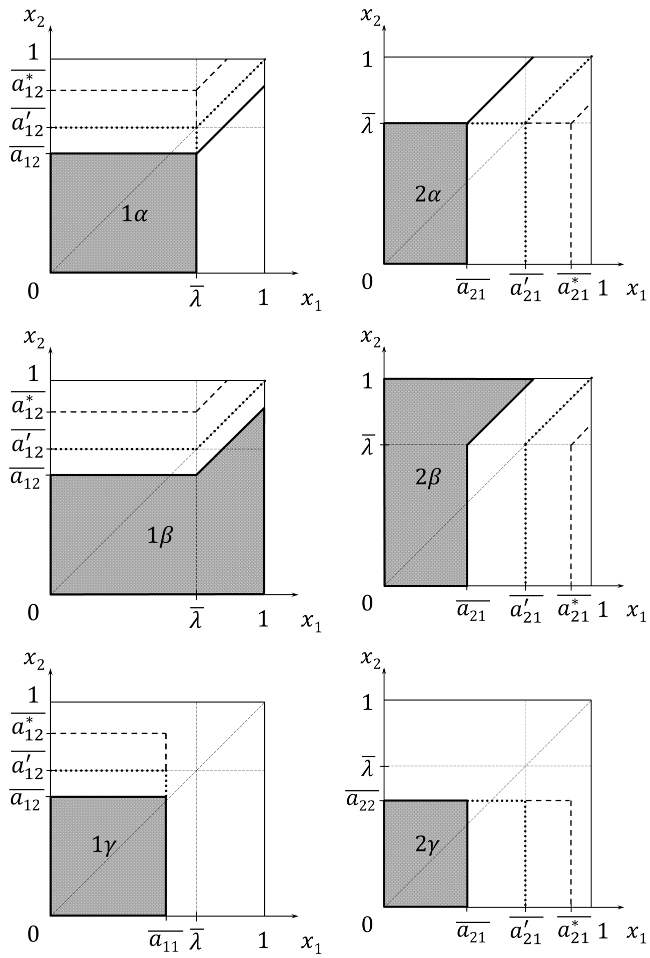


Figure 5.6: Types of solution sets for particular equations

Example 5.2.1. Let us consider two matrices, A and B, and for each, consider $\lambda = 0.4$.

$$A = \begin{pmatrix} 0.3 & 0.8 \\ 0.2 & 0.4 \end{pmatrix}; B = \begin{pmatrix} 0.4 & 0.5 \\ 0.2 & 0.4 \end{pmatrix}$$

To find the eigenspace associated with the given lambda, first the position of λ and the diagonal entries of the matrices should be determined. For A, one has $a_{11} < \lambda = a_{22}$. This means that the eigenspace is the intersection of the sets 1α and 2β. The fact that

$a_{12} - \lambda > \lambda - a_{21}$ causes the segment of solutions to be shortened (the left part of Figure 5.7).

The diagonal entries of B are $a_{11} = a_{22} = \lambda$ and this leads to the conclusion that the eigenspace of B is the intersection of the sets 1β and 2β. Note that the presence of a stripe of solutions (the right part of Figure 5.7) is caused by the distances of the elements a_{12} and a_{21} from λ, specifically by the inequality $a_{12} - \lambda < \lambda - a_{21}$.

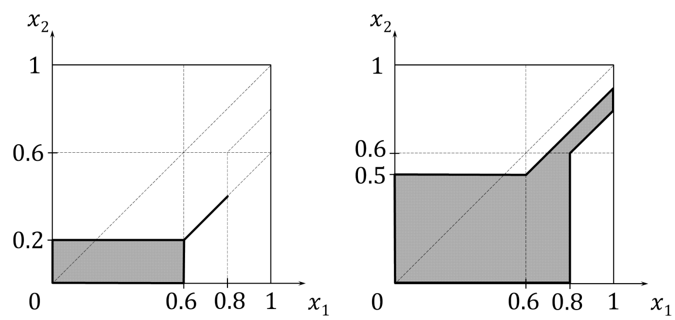


Figure 5.7: Solution sets for A (on the left) and B (on the right)

It can be seen from the previous figures that the value $1 - \lambda$ divides the coordinate system into four parts. The eigenvectors can be described by means of the partition of the indices of the vector components (introduced in [Gavalec et al., 2015]) according to their value. For $x_i \leq 1 - \lambda$, the index $i \in L$, and for $x_i > 1 - \lambda$, include the index $i \in K$. Such a partition is called a (K, L) partition. It is a partition of $\{1, \ldots, n\}$, that is, the subsets $K, L \subseteq \{1, \ldots, n\}$ such that $K \cup L = \{1, \ldots, n\}$ and $K \cap L = \emptyset$, see Figure 5.8. The letters in the circle indicate the membership of i in the set according to the above written conditions; each of the four parts of the whole set is then described by a certain combination of the subsets K and L. The Łukasiewicz eigenvectors satisfying these conditions are called (K, L)-Łukasiewicz eigenvectors. On the other hand, if for some partition, the (K, L)-Łukasiewicz eigenvectors exist, then such a partition is called a "secure partition." For the given eigenspace in Figure 5.8, $L = \{1, 2\}, K = \emptyset$, $L = \{2\}, K = \{1\}$, and $L = \emptyset, K = \{1, 2\}$ are secure.

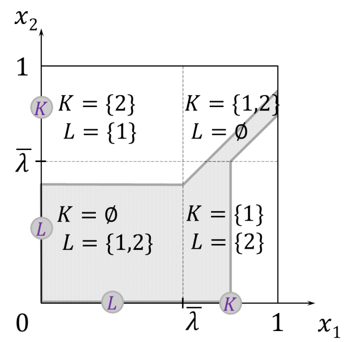


Figure 5.8: (K, L) partitions for two dimensions

5.2.2 Eigenspaces of Matrices of Higher Dimensions

A complex algorithm for the computation of the eigenspace using the (K, L) partition is given in [Gavalec et al., 2015]. Solving the eigenproblem is divided into several steps: finding the secure partitions, and computing the generators of the tropical convex hull separately for each secure partition. The final solution set of the eigenproblem is then the union of the constituent solutions.

To give an example for higher dimensions it is necessary to define the concept of "security."

Three types of "security" can be distinguished, [Gavalec et al., 2015]:

- for $\lambda > 0$, a weighted digraph is called λ-*secure* if for each node of the graph and each walk P issuing from that node, $-\lambda + w(P) \leq 0$.

- a partition is called *secure* if every walk in the weighted digraph of $A^{(\lambda)}$ that starts at a node that belongs to L and has all other nodes in K, has a non-positive weight. The partition $([n], \emptyset)$ is secure if the corresponding directed graph of $A^{(\lambda)}$ is λ-secure, and that the partition $(\emptyset, [n])$ is secure. The notation $A^{(\lambda)}$ stands for the matrix with entries equal to $a_{ij} - \lambda$.

- A node i of a weighted digraph is called secure if the weight of every walk starting at i is nonpositive.

The computation of the eigenspaces for matrices of higher dimensions is shown for the following three-dimensional example.

Example 5.2.2. Consider the max-Łukasiewicz eigenproblem for $\lambda = 0,7$ and

$$A = \begin{pmatrix} 0.6 & 0.8 & 0.4 \\ 0.1 & 0.5 & 0.4 \\ 0.9 & 0.6 & 0.5 \end{pmatrix}.$$

Then the matrix $A^{(\lambda)}$ can be expressed and the corresponding digraph constructed:

$$A^{(\lambda)} = \begin{pmatrix} -0.1 & 0.1 & -0.3 \\ -0.6 & -0.2 & -0.3 \\ 0.2 & -0.1 & -0.2 \end{pmatrix}$$

First, let us determine which of the partitions are secure according to the definition

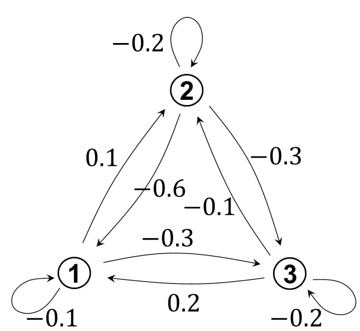


Figure 5.9: Weighted directed graph of $A^{(\lambda)}$

of security for this simple three-dimensional example. It is easily seen that there are two positive paths in the directed graph of $A^{(\lambda)}$, see Figure 5.9. Let's start with the partition $L = \{2\}, K = \{1, 3\}$. This partition is secure because it can be verified that every walk that starts at node 2 and has all other nodes in K (the walk can then continue only between nodes 1 and 3) has nonpositive weight.

To get another secure partition, a node in K that is secure in K (according to the definition of the security of nodes) has to be found. This node, say k, can then be shifted and the partition $(K-k, L+k)$ is also a secure partition.

The node 3 is not secure in K, because the walk $p=(3,1)$ has positive weight $w(p)=0.2$. The node 1 is secure in K because every walk realized from this node (that has all other nodes in K) has nonpositive weight. This node can be added to L, thus the next secure partition $L=\{1,2\}, K=\{3\}$ is obtained. According to the definition, the partitions $L=\{1,2,3\}, K=\emptyset$ and $L=\emptyset, K=\{1,2,3\}$ are also secure.

To picture the (K, L) partitions let's look at Figure 5.10. The graphical representation is a 3D-cube divided (similarly to the two-dimensional example) by the value $\overline{\lambda}=1-\lambda$ into eight blocks. The solution set of the eigenproblem is the union of the tropical convex hulls for each of the secure partitions. In this example, it is the union of the solution sets of the four blocks in the cube represented by the partitions

$$L=\emptyset, K=\{1,2,3\}$$
$$L=\{2\}, K=\{1,3\}$$
$$L=\{1,2\}, K=\{3\}$$
$$L=\{1,2,3\}, K=\emptyset.$$

We will focus on the partition with the least L, i.e., $L=\emptyset, K=\{1,2,3\}$. This partition is graphically represented by the highlighted cube at the upper right corner in Figure 5.10 (all of the vector indices belong to K, that is, the vector components are greater than or equal to $1-\lambda$). Using Theorem 3.2. and Corollary 3.4. of [Gavalec et al., 2015], we compute the generators of the tropical convex hull for this partition. We obtain two vectors, $u=(0.4\ 0.3\ 0.6)$ and $w^{(1)}=(0.8\ 0.7\ 1)$. These vectors are depicted in Figure 5.12. To highlight the generators in the pictures, these points are marked with small squares. The point $w^{(1)}$ lies on the back side of the cube,

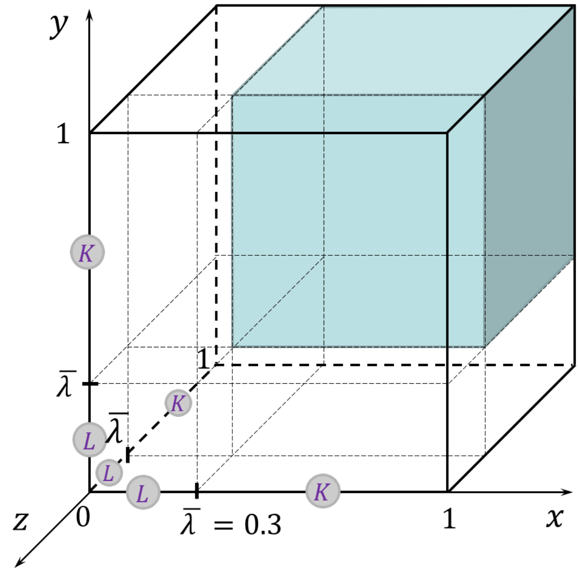


Figure 5.10: (K, L) partitions for three dimensions (upper part)

whereas u lies in its bottom. These two points are connected by a tropical segment. The difference between each pair of the generator's coordinates is 0.4. Therefore, the tropical segment is the segment parallel to the main diagonal of the universal set (every coordinate decreases at the same rate, notice the similarity to the graphical representation of the segment for two dimensions).

Similarly, the generators for the remaining three secure partitions highlighted in Figure 5.11 can be computed. The solution set for the partition $L = \{2\}, K = \{1, 3\}$ is the tropical convex hull of the vectors $u = (0.3\ 0\ 0.5)$, $v^{(2)} = (0.4\ 0.3\ 0.6)$, $w^{(1)} = (0.4\ 0\ 0.6)$ depicted in Figure 5.13. The block is located under the highlighted cube from Figure 5.10 and one can observe that $v^{(2)}$ is a joint vector for these neighboring blocks. The tropical segment between the points u and $v^{(2)}$ is not as simple as the segment for the partition with the least L. At first, all three components of the coordinates can decrease by the value 0.1, until the break point $(0.3\ 0.2\ 0.5)$ is reached. In the pictures, the break points are marked with small circles. This part of

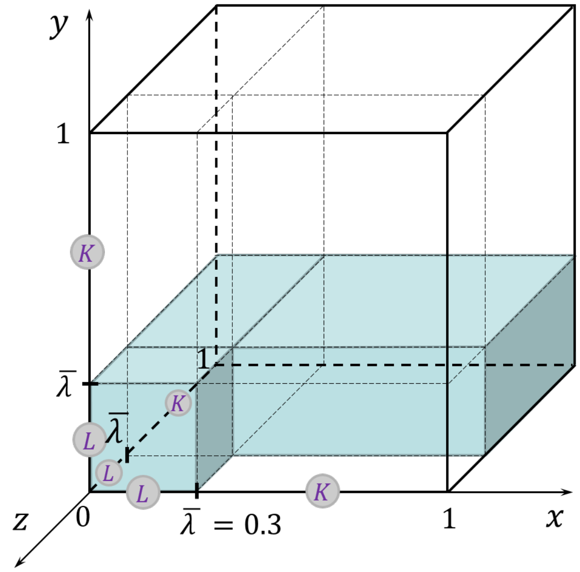


Figure 5.11: (K, L) partitions for three dimensions (lower part)

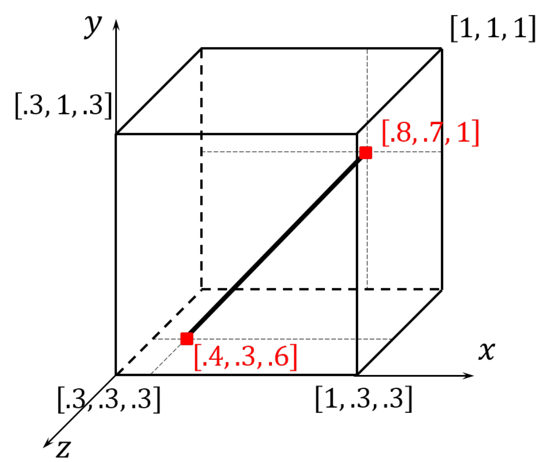


Figure 5.12: The tropical convex hull for the secure partition $L = \emptyset, K = \{1, 2, 3\}$

the segment is again parallel to the main diagonal of the universal set. Then, to reach the point u, the y coordinate should decrease, i.e., the segment continues vertically down.

The generators of the eigenspace for the partition $L = \{1, 2\}, K = \{3\}$ are $u =$

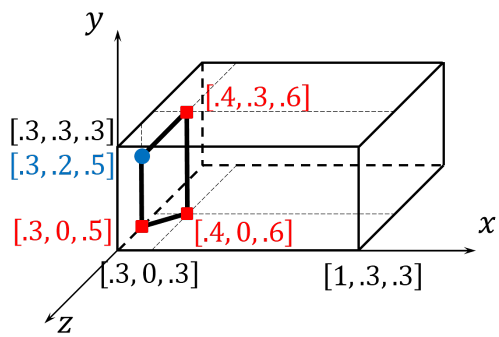


Figure 5.13: Particular solution for the secure partition $L = \{2\}, K = \{1, 3\}$

(0 0 0.3), $v^{(1)} = (0.3\ 0\ 0.5)$, $v^{(2)} = (0\ 0.2\ 0.3)$. The eigenspace for this partition is depicted in Figure 5.14. Observe that the solution set is the union of two two-dimensional figures.

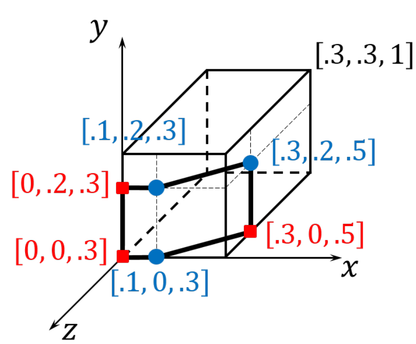


Figure 5.14: Particular solution for the secure partition $L = \{1, 2\}, K = \{3\}$

Lastly, the greatest eigenvector for the partition $L = \{1, 2, 3\}, K = \emptyset$ is (0.1 0.2 0.3). This means that the eigenspace of this partition consists of all vectors satisfying $x_L \leq (0.1\ 0.2\ 0.3)$. The solution set is depicted in Figure 5.15. Observe that unlike the previous solution sets, this solution set is a 3D object: a block.

Two types of generators can be distinguished. For partitions where $K \neq \emptyset$, the so-called weak generators are obtained. The tropical convex hull is generated by

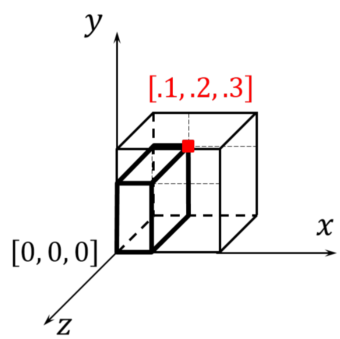


Figure 5.15: Particular solution for the secure partition $L = \{1, 2, 3\}, K = \emptyset$

connecting them to each other. This is a classical concept of generating convex figures. On the other hand, for a partition where $K = \emptyset$, a so-called strong generator is obtained. This means that the particular solution set consists of all vectors that are less than or equal to a strong generator. In other words, the mere connection of the generator with the zero point gives only an incomplete solution.

The solution set of the eigenproblem for a given A and λ is the union of the particular solutions. The final eigenspace is depicted in Figure 5.16. Notice that the whole solution set is also tropically convex.

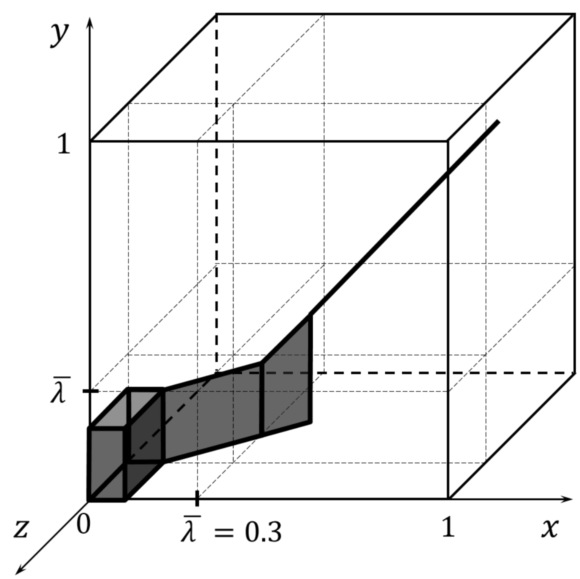


Figure 5.16: The solution set for given A, λ

6. Further research

The popularity of fuzzy algebras has shown that they are a current and important topic. Therefore, the further direction of research is to study further possible applications of fuzzy algebras to solving analogous issues.

One of several possibilities is to develop the Łukasiewicz eigenspace theory. The visualization of an eigenspace as the intersection of some areas can be extended by the description of the intersection with the use of the coordinates of the vertices (vectors). The eigenspace for a matrix of dimension 3 can be still pictured in three-dimensional space, but for higher dimensions a graphical representation is hardly imaginable. Thus, the description by the vectors will be helpful, particularly for the study of the eigenspaces of matrices of higher dimensions.

The present research could be extended in another possible direction: the aim would be to give a systematic characterization of the eigenspaces for other t-norms: Gödel, product and drastic. Very interesting would be their comparison, which could give new perspectives on this topic.

7. Conclusions

This thesis represents the results of research in the field of the control of discrete event systems, focusing on the description of the stable states on the basis of well-known theory. Methods that explore the evolution of the states of complex systems have been described. The algebraic tools have been developed, and they have provided a solid theoretical basis.

The presented results can help to understand the functioning and optimization of the performance of complex systems. Concerning closed linear queuing systems, the development of the queue lengths and the description of the stable states has been given. For open linear queuing systems, the changes in the lengths of the queues have also been described: four types of systems, according to the type of their service time and request flow, have been distinguished. Two methods for the optimization of the system's efficiency through the cost optimization have been suggested. Lastly, the description of the matrix powers in max-drast algebra and the the systematic characterization of the eigenspaces in max-Łukasiewicz algebra have been given.

These research results were published mainly in international conference proceedings focused on Operations Research, or were accepted for the journal *Fuzzy Sets and Systems*.

References

[Baum and Breuer, 2005] Baum, D. and Breuer, L. (2005). *An Introduction to Queueing Theory*. Springer.

[Bobillo and Straccia, 2011] Bobillo, F. and Straccia, U. (2011). Reasoning with the finitely many-valued Łukasiewicz fuzzy description logic. *Information Sciences*, 181(4):758–778.

[Bondy and Murty, 1976] Bondy, J. A. and Murty, U. S. R. (1976). *Graph Theory with Applications*, volume 290. Macmillan London.

[Bouquard et al., 2006] Bouquard, J.-L., Lenté, C., and Billaut, J.-C. (2006). Application of an optimization problem in max-plus algebra to scheduling problems. *Discrete Applied Mathematics*, 154(15):2064–2079. International Symposium on Combinatorial Optimization CO'02.

[Brackley et al., 2012] Brackley, C. A., Broomhead, D. S., Romano, M. C., and Thiel, M. (2012). A max-plus model of ribosome dynamics during mRNA translation. *Journal of Theoretical Biology*, 303(0):128–140.

[Braker, 1993] Braker, J. G. (1993). *Algorithms and applications in timed discrete event systems*. TU Delft, Delft University of Technology.

[Bureš, 2011] Bureš, V. (2011). *Systémové myšlení pro manažery*, volume 1. Professional Publishing.

[Butkovič, 2010] Butkovič, P. (2010). *Max-Linear Systems: Theory and Algorithms*. Springer.

[Butkovič and Cuninghame-Green, 2007] Butkovič, P. and Cuninghame-Green, R. (2007). On matrix powers in max-algebra. *Linear Algebra and its Applications*, 421(2-3):370–381. Special Issue in honor of Miroslav Fiedler.

[Buzacott and Shanthikumar, 1993] Buzacott, J. A. and Shanthikumar, J. G. (1993). *Stochastic Models of Manufacturing Systems*, volume 4. Prentice-Hall, Englewood Cliffs, NJ.

[Casasnovas and Mayor, 2008] Casasnovas, J. and Mayor, G. (2008). Discrete t-norms and operations on extended multisets. *Fuzzy Sets and Systems*, 159(10):1165–1177.

[Cassandras and Lafortune, 2008] Cassandras, C. G. and Lafortune, S. (2008). *Introduction to Discrete Event Systems*. Springer.

[Cechlárová, 2001] Cechlárová, K. (2001). Solutions of interval linear systems in (max,+)-algebra. In *Proceedings of the 6th International Symposium on Operational Research Preddvor, Slovenia*, pages 321–326.

[Cechlárová, 2005] Cechlárová, K. (2005). Eigenvectors of interval matrices over max-plus algebra. *Discrete Applied Mathematics*, 150(1-3):2–15.

[Cechlárová and Cuninghame-Green, 2002] Cechlárová, K. and Cuninghame-Green, R. (2002). Interval systems of max-separable linear equations. *Lin. Algebra Appl.*, 340:215–224.

[Chaira, 2015] Chaira, T. (2015). Fuzzy mathematical morphology using triangular operators and its application to images. *Journal of Intelligent and Fuzzy Systems*, 28(5):2269–2277.

[Chen and Du, 2015] Chen, X.-G. and Du, H. (2015). Interval-valued triangular fuzzy soft set and its method of dynamic decision making. *Xi Tong Gong Cheng Yu Dian Zi Ji Shu/Systems Engineering and Electronics*, 37(5):1111–1115.

[Cimler and Němcová, 2012] Cimler, R. and Němcová, Z. (2012). Optimalizace nákladů systému hromadné obsluhy. In *Sborník příspěvků IMEA 2012*.

[Cinlar, 2013] Cinlar, E. (2013). *Introduction to Stochastic Processes*. Courier Corporation.

[Cohen et al., 1985] Cohen, G., Dubois, D., Quadrat, J., and Viot, M. (1985). A linear-system-theoretic view of discrete-event processes and its use for performance evaluation in manufacturing. *IEEE Transactions on Automatic Control*, 30(3):210–220.

[Comet, 2003] Comet, J.-P. (2003). Application of max-plus algebra to biological sequence comparisons. *Theoretical Computer Science*, 293(1):189–217.

[Cuninghame-Green, 1960] Cuninghame-Green, R. A. (1960). Process synchronisation in a steelworks–A problem of feasibility. In *Proceedings of the 2nd International Conference on Operational Research*, pages 323–328.

[Cuninghame-Green, 1962] Cuninghame-Green, R. A. (1962). Describing industrial processes with interference and approximating their steady-state behaviour. *Operational Research*, 13:95–100.

[Cuninghame-Green, 1979] Cuninghame-Green, R. A. (1979). *Minimax Algebra. Lecture Notes in Economics and Mathematical Systems* vol. 166. Springer-Verlag, Berlin.

[Cuninghame-Green, 1991] Cuninghame-Green, R. A. (1991). Minimax algebra and applications. *Fuzzy Sets and Systems*, 41:251–267.

[Febbraro et al., 1994] Febbraro, A., Minciardi, R., Parodi, G., Prati, A., Profumo, M., and Sacone, S. (1994). Performance optimization in manufacturing systems by use of max-plus algebraic techniques. volume 2, pages 1995–2001.

[Gavalec, 1997] Gavalec, M. (1997). Computing matrix period in max-min alegebra. *Discrete Applied Mathematics*, 75(1):63–70.

[Gavalec, 2000a] Gavalec, M. (2000a). Computing orbit period in max-min algebra. *Discrete Applied Mathematics*, 100(1-2):49–65.

[Gavalec, 2000b] Gavalec, M. (2000b). Linear matrix period in max-plus algebra. *Linear Algebra and its Applications*, 307(1–3):167–182.

[Gavalec and Němcová, 2010] Gavalec, M. and Němcová, Z. (2010). Eigenproblem for queueing systems in max-plus algebra. In *Mathematical Methods in Economics 2010*.

[Gavalec and Němcová, 2011] Gavalec, M. and Němcová, Z. (2011). Queues in deterministic linear manufacturing systems. In *Mathematical Methods in Economics 2011*.

[Gavalec and Němcová, 2012] Gavalec, M. and Němcová, Z. (2012). Matrix period in max-drast fuzzy algebra. In *Mathematical Methods in Economics 2012*.

[Gavalec and Němcová, 2014] Gavalec, M. and Němcová, Z. (2014). Deterministic queuing models in a service line. In *Mathematical Methods in Economics 2014*.

[Gavalec and Němcová, 2016] Gavalec, M. and Němcová, Z. (2016). Steady states of max-Lukasiewicz fuzzy systems. *Fuzzy Sets and Systems*. Under review.

[Gavalec et al., 2015] Gavalec, M., Němcová, Z., and Sergeev, S. (2015). Tropical linear algebra with the Lukasiewicz t-norm. *Fuzzy Sets and Systems*, 276:131–148.

[Gavalec and Plavka, 2010] Gavalec, M. and Plavka, J. (2010). Monotone interval eigenproblem in max-min algebra. *Kybernetika*, 46(3):387–396.

[Gavalec et al., 2011] Gavalec, M., Plavka, J., and Tomaskova, H. (2011). Eigenspace of interval matrices in max-min algebra. In *Mathematical Methods in Economics 2011*. Professional Publishing, Prague.

[Gavalec et al., 2014] Gavalec, M., Plavka, J., and Tomášková, H. (2014). Interval eigenproblem in max–min algebra. *Linear Algebra and its Applications*, 440:24–33.

[Gottwald, 2001] Gottwald, S. (2001). *A Treatise on Many-Valued Logics*, volume 3. Research Studies Press Baldock.

[Gottwald, 2005] Gottwald, S. (2005). Mathematical fuzzy logic as a tool for the treatment of vague information. *Information Sciences*, 172(1):41–71.

[Goverde, 2010] Goverde, R. (2010). A delay propagation algorithm for large-scale railway traffic networks. *Transportation Research Part C: Emerging Technologies*, 18(3):269–287.

[Goverde, 2007] Goverde, R. M. (2007). Railway timetable stability analysis using max-plus system theory. *Transportation Research Part B: Methodological*, 41(2):179–201.

[Goverde et al., 2011] Goverde, R. M., Heidergott, B., and Merlet, G. (2011). A coupling approach to estimating the Lyapunov exponent of stochastic max-plus linear systems. *European Journal of Operational Research*, 210(2):249–257.

[Goverde and Odijk, 2002] Goverde, R. M. and Odijk, M. A. (2002). Performance evaluation of network timetables using peter. *Computers in Railways VIII*, pages 731–740.

[Gross and Yellen, 2004] Gross, J. L. and Yellen, J. (2004). *Handbook of Graph Theory*. CRC Press.

[Gursoy et al., 2013] Gursoy, B. B., Mason, O., and Sergeev, S. (2013). The analytic hierarchy process, max algebra and multi-objective optimisation. *Linear Algebra and its Applications*, 438(7):2911–2928.

[Hájek et al., 1996] Hájek, P., Godo, L., and Esteva, F. (1996). A complete many-valued logic with product-conjunction. *Archive for Mathematical Logic*, 35(3):191–208.

[Hao and Yifei, 2011] Hao, T. and Yifei, T. (2011). Study on queuing system optimization of bank based on {BPR}. *Procedia Environmental Sciences*, 10, Part A(0):640–646. 2011 3rd International Conference on Environmental Science and Information Application Technology {ESIAT} 2011.

[Heidergott, 1998] Heidergott, B. (1998). Max plus algebra and queues.

[Heidergott, 2000] Heidergott, B. (2000). A characterisation of (max,+)-linear queueing systems. *Queueing Systems*, 35(1–4):237–262.

[Heidergott and de Vries, 2001] Heidergott, B. and de Vries, R. (2001). Towards a (max,+) control theory for public transportation networks. *Discrete Event Dynamic Systems*, 11(4):371–398.

[Heidergott et al., 2006] Heidergott, B., Olsder, G. J., and van der Woude, J. W. (2006). *Max Plus at Work: Modeling and Analysis of Synchronized Systems: A Course on Max-Plus Algebra and its Applications*, volume 13. Princeton University Press.

[Houssin et al., 2007] Houssin, L., Lahaye, S., and Boimond, J.-L. (2007). Just in time control of constrained (max, plus)-linear systems. *Discrete Event Dynamic Systems—Theory and Applications*, 17(2):159–178. Conference on Modelling Reactive Systems, Autrans, FRANCE, Oct., 2005.

[Iliadis et al., 2008] Iliadis, L., Spartalis, S., and Tachos, S. (2008). Application of fuzzy t-norms towards a new artificial neural networks evaluation framework: A case from wood industry. *Information Sciences*, 178(20):3828–3839. Special Issue on Industrial Applications of Neural Networks 10th Engineering Applications of Neural Networks 2007.

[Jablonský, 2002] Jablonský, J. (2002). *Operační výzkum: kvantitativní modely pro ekonomické rozhodování*. Professional Publishing.

[Jantzen, 2007] Jantzen, J. (2007). *Foundations of Fuzzy Control*. John Wiley & Sons.

[Klement et al., 2000] Klement, E. P., Mesiar, R., and Pap, E. (2000). *Triangular Norms*. Springer.

[Klement et al., 2004] Klement, E. P., Mesiar, R., and Pap, E. (2004). Triangular norms. Position paper I: Basic analytical and algebraic properties. *Fuzzy Sets and Systems*, 143(1):5–26.

[Konigsberg, 2008] Konigsberg, Z. R. (2008). A generalized eigenmode algorithm for reducible regular matrices over the max-plus algebra with applications to the metro-bus public transport system in Mexico City. *Nonlinear Analysis: Hybrid Systems*, 2(4):1205–1216.

[Kosko, 1986] Kosko, B. (1986). Fuzzy entropy and conditioning. *Information sciences*, 40(2):165–174.

[Lawrence and Pasternack, 1997] Lawrence, J. A. and Pasternack, B. A. (1997). *Applied Management Science*. John Wiley & Sons.

[Le Boudec and Thiran, 2001] Le Boudec, J.-Y. and Thiran, P. (2001). *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*. Springer-Verlag, Berlin.

[Liang et al., 2014] Liang, C., Zhao, S., and Zhang, J. (2014). Aggregation operators on triangular intuitionistic fuzzy numbers and its application to multi-criteria decision making problems. *Foundations of Computing and Decision Sciences*, 39(3):189–208.

[Liu et al., 2016] Liu, C.-C., Wu, Y.-K., Lur, Y.-Y., and Tsai, C.-L. (2016). On the power sequence of a fuzzy matrix with convex combination of max-product and max-min operations. *Fuzzy Sets and Systems*, 289:157–163.

[Łukasiewicz, 1930] Łukasiewicz, J. (1930). Philosophische bemerkungen zu mehrwertigen systemen des aussagenkalk. *C. R. Soc. Sci. Lett. Varsovie*, 23:51–77.

[Łukasiewicz, 1970] Łukasiewicz, J. (1970). *Studies in Logic and the Foundations of Mathematics*. North-Holland.

[Menger, 1942] Menger, K. (1942). Statistical metrics. *Proceedings of the National Academy of Sciences*, 28(12):535–537.

[Miller, 2009] Miller, B. M. (2009). Optimization of queuing system via stochastic control. *Automatica*, 45(6):1423–1430.

[Mills, 1993] Mills, J. W. (1993). Lukasiewicz' insect: The role of continuous-valued logic in a mobile robot's sensors, control, and locomotion. In *Proceedings of The Twenty-Third International Symposium on Multiple-Valued Logic, 1993*, pages 258–263. IEEE.

[Molnárová, 2005] Molnárová, M. (2005). Generalized matrix period in max-plus algebra. *Linear Algebra and its Applications*, 404:345–366.

[Myšková, 2012] Myšková, H. (2012). Interval max-plus systems of linear equations. *Linear Algebra and its Applications*, 437(8):1992–2000.

[Myšková, 2016] Myšková, H. (2016). Interval max-plus matrix equations. *Linear Algebra and its Applications*, 492:111–127.

[Myšková and Plavka, 2013] Myšková, H. and Plavka, J. (2013). X-robustness of interval circulant matrices in fuzzy algebra. *Linear Algebra and its Applications*, 438(6):2757–2769.

[Myšková and Plavka, 2014] Myšková, H. and Plavka, J. (2014). The robustness of interval matrices in max-plus algebra. *Linear Algebra and its Applications*, 445(0):85–102.

[Ng and Soong, 2008] Ng, C.-H. and Soong, B.-H. (2008). *Queueing Modelling Fundamentals: With Applications in Communication Networks*. John Wiley & Sons.

[Nitica and Singer, 2007] Nitica, V. and Singer, I. (2007). The structure of max-plus hyperplanes. *Linear Algebra and its Applications*, 426(2):382–414.

[Němcová, 2011a] Němcová, Z. (2011a). Max-plus algebra v teorii hromadné obsluhy. In *Sborník recenzovaných příspěvků HED 2011*.

[Němcová, 2011b] Němcová, Z. (2011b). Queues in the closed and open queuing system. In *Sborník příspěvků z letní školy "Mezioborové přístupy informatiky a kognitivní vědy"*.

[Němcová, 2014] Němcová, Z. (2014). Náklady detedeterministických systémů hromadné obsluhy. In *Sborník příspěvků IMEA 2014*.

[Papadopoulos and Heavey, 1996] Papadopoulos, H. and Heavey, C. (1996). Queueing theory in manufacturing systems analysis and design: A classification of models for production and transfer lines. *European Journal of Operational Research*, 92(1):1–27.

[Pardo and de la Fuente, 2008] Pardo, M. J. and de la Fuente, D. (2008). Design of a fuzzy finite capacity queuing model based on the degree of customer satisfaction: Analysis and fuzzy optimization. *Fuzzy Sets and Systems*, 159(24):3313–3332.

[Pedrycz and Gomide, 2007] Pedrycz, W. and Gomide, F. (2007). *Fuzzy Systems Engineering: Toward Human-centric Computing*. John Wiley & Sons.

[Perovic et al., 2011] Perovic, A., Ognjanovic, Z., Raskovic, M., and Radojevic, D. (2011). Finitely additive probability measures on classical propositional formulas definable by Gödel's t-norm and product t-norm. *Fuzzy Sets and Systems*, 169(1):65–90.

[Plavka, 2014] Plavka, J. (2014). The weak robustness of interval matrices in max-plus algebra. *Discrete Applied Mathematics*, 173(0):92–101.

[Reddy et al., 2009] Reddy, Y., Janardhana, R., and Kun, K. (2009). Modular modelling and performance evaluation of manufacturing systems using max-plus algebra. *International Journal of Industrial and Systems Engineering*, 4(5):577–596.

[Ren et al., 2007] Ren, S. C., Xu, D., Wang, F., and Tan, M. (2007). Timed event graph-based cyclic reconfigurable flow shop modelling and optimization. *International Journal of Production Research*, 45(1):143–156.

[Rohn, 1989] Rohn, J. (1989). Systems of linear interval equations. *Linear Algebra and its Applications*, 126:39–78.

[Rudas et al., 2008] Rudas, I. J., Batyrshin, I. Z., Zavala, A. H., Nieto, O. C., Horváth, L., and Vargas, L. V. (2008). Generators of fuzzy operations for hardware implementation of fuzzy systems. In *MICAI 2008: Advances in Artificial Intelligence*, pages 710–719. Springer.

[Schutter and van den Boom, 2001] Schutter, B. D. and van den Boom, T. (2001). Model predictive control for max-plus-linear discrete event systems. *Automatica*, 37(7):1049–1056.

[Schweizer et al., 1960] Schweizer, B., Sklar, A., et al. (1960). Statistical metric spaces. *Pacific J. Math*, 10(19601):313–335.

[Seleim and ElMaraghy, 2014] Seleim, A. and ElMaraghy, H. (2014). Parametric analysis of mixed-model assembly lines using max-plus algebra. *{CIRP} Journal of Manufacturing Science and Technology*, 7(4):305–314.

[Semančíková, 2006] Semančíková, B. (2006). Orbits in max-min algebra. *Linear Algebra and its Applications*, 414(1):38–63.

[Semančíková, 2007] Semančíková, B. (2007). Orbits and critical components of matrices in max-min algebra. *Linear Algebra and its Applications*, 426(23):415–447.

[Sergeev, 2009] Sergeev, S. (2009). Max algebraic powers of irreducible matrices in the periodic regime: An application of cyclic classes. *Linear Algebra and its Applications*, 431(8):1325–1339.

[Stevenson, 2014] Stevenson, W. J. (2014). *Operations Management*. McGraw-Hill.

[Tempelmeier, 2003] Tempelmeier, H. (2003). Practical considerations in the optimization of flow production systems. *International Journal of Production Research*, 41(1):149–170.

[Tutte, 1984] Tutte, W. T. (1984). *Graph Theory, Volume 21 of Encyclopedia of Mathematics and its Applications*. Addison-Wesley, Reading, MA.

[Vorobjov, 1967] Vorobjov, N. N. (1967). Extremal Algebra of positive Matrices. *Datenverarbeitung und Kybernetik*, 3:37–71.

[Zadeh, 1965] Zadeh, L. (1965). Fuzzy sets. *Information and Control*, 8:338–353.

[Zeigler et al., 2000] Zeigler, B. P., Praehofer, H., and Kim, T. G. (2000). *Theory of modeling and simulation: Integrating discrete event and continuous complex dynamic systems*. Academic Press.

[Zimmermann, 1976] Zimmermann, K. (1976). *Extremal Algebra*. Ekon. Ústav ČSAV Praha.

[Zimmermann, 2003] Zimmermann, K. (2003). Disjunctive optimization, max-separable problems and extremal algebras. *Theoretical Computer Science*, 293(1):45–54.

[Zimmermann, 1981] Zimmermann, U. (1981). Linear and Combinatorial Optimization in Ordered Algebraic Structure. *Ann. Discrete Math.*, No. 10, North-Holland, Amsterdam.

List of Author's Publications

Publications in Preparation

M. Gavalec and Z. Němcová. Deterministic and stochastic queuing model in a service line. 2016. To be published in E+M.

M. Gavalec and Z. Němcová. Steady states of max-lukasiewicz fuzzy systems. *Fuzzy Sets and Systems*, 2016. Under review.

Z. Němcová. Cost optimizing methods for deterministic queuing systems. 2016. To be published in LNCS.

P. Tučník, Z. Němcová, and T. Nacházel. Management of industrial production in agent-based artificial economic system. 2016. To be published in LNCS.

Articles

M. Gavalec, Z. Němcová, and S. Sergeev. Tropical linear algebra with the lukasiewicz t-norm. *Fuzzy Sets and Systems*, 276:131–148, 2015.

P. Tučník and Z. Němcová. Production unit supply management solution in agent-based computational economics model. In *New Trends in Intelligent Information and Database Systems*, volume 598 of *Studies in Computational Intelligence*, pages 343–352. 2015.

Papers in Proceedings

R. Cimler and Z. Němcová. Optimalizace nákladů systému hromadné obsluhy. In *Sborník příspěvků IMEA 2012*, 2012.

M. Gavalec and Z. Němcová. Eigenproblem for queueing systems in max-plus algebra. In *Mathematical Methods in Economics 2010*, 2010.

M. Gavalec and Z. Němcová. Queues in deterministic linear manufacturing systems. In *Mathematical Methods in Economics 2011*, 2011.

M. Gavalec and Z. Němcová. Matrix period in max-drast fuzzy algebra. In *Mathematical Methods in Economics 2012*, 2012.

M. Gavalec and Z. Němcová. Orbit period in max-t fuzzy algebra. In *Mathematical methods in economy and industry*, 2012.

M. Gavalec and Z. Němcová. Deterministic queuing models in a service line. In *Mathematical Methods in Economics 2014*, 2014.

M. Šimková, H. Tomášková, and Z. Němcová. Mobile education in tools. In *Procedia-Social and Behavioral Sciences*, volume 47, pages 10–13, 2012.

Z. Němcová. Cost optimization for tandem networks. In *Proceedings of the 18th Czech-Japan Seminar on Data Analysis and Decision Making under Uncertainty*, 2015.

Z. Němcová. Steady states of max-lukasiewicz fuzzy systems. In *Proceedings of the 2015 Conference of the International Fuzzy Systems Association and the European Society for Fuzzy Logic and Technology*, 2015.

Z. Němcová. Max-plus algebra v teorii hromadné obsluhy. In *Sborník recenzovaných příspěvků HED 2011*, 2011.

Z. Němcová. Queues in the closed and open queuing system. In *Sborník příspěvků z letní školy Mezioborové přístupy informatiky a kognitivní vědy*, 2011.

Z. Němcová. Náklady detedeterministických systémů hromadné obsluhy. In *Sborník příspěvků IMEA 2014*, 2014.

H. Tomášková, Z. Němcová, and M. Šimková. Usage of virtual communication in university environment. In *Procedia-Social and Behavioral Sciences*, volume 28, pages 360–364, 2011.