INTRAS

Co-funded by the
Erasmus+ Programme
of the European Union

“Traffic modelling and simulation”

Syllabus

Prof. Kyandoghere Kyamakya

Prof. Jean Chamberlain Chedjou

Intelligent Transport Systems: New ICT – based Master’s Curricula in Uzbekistan (INTRAS)
Agreement number: 2017-3516/001-001
Project reference number: 586292-EPP-1-2017-1-PL-EPPKA2-CBHE-JP

GENERAL OVERVIEW OF THE LECTURE: Traffic modelling

- This Lecture focusses on both modelling and simulation of traffic scenarios/phenomena. The scenarios/phenomena are selected in the fields of Road transportation, Railway transportation, supply chain management and Logistics.
- Some selected traffic scenarios/phenomena are considered, and the mathematical modeling is carried out. The mathematical models obtained are expressed in the form of ordinary differential equations (ODEs), partial differential equations (PDEs), and a coupling between ODEs and PDEs.
- The graphical modeling of traffic is further considered and several traffic scenarios/ phenomena (selected in the fields of road transportation, railway transportation, supply chains management and logistics) are modelled in the form of graphs. The graph theory is further considered to investigate important issues like the Shortest Path Problem (SPP), the Traveling Salesman Problem (TSP), the minimum consumption of resources, the optimal dispatching of Goods, rail- vehicles, just to name a few.
- The optimization theory is introduced and used to model the SPP and TSP, the optimal consumption of resources, and the optimal dispatching of Goods and rail vehicles.

GENERAL OVERVIEW OF THE LECTURE: Traffic simulation

- The concept of Neural Networks (NN) is introduced and different Neural Networks architectures are investigated. The architectures of Neural Networks investigated are further used for traffic simulation.
- The Lecture also introduces the simulation software MATLAB and show how MATLAB can be used for traffic simulation. Students learn how to used some basic MATLAB toolboxes for optimization (i.e., Linear Programming (LP) and Quadratic Programming (QP) toolboxes) to simulate/solve the optimization models established/obtained when modeling selected traffic scenarios/phenomena.
- Students also learn how to use the software MATLAB to solve many complex ordinary differential equations (ODEs) and Partial Differential Equations (PDEs) obtained as mathematical models of selected traffic scenarios/phenomena at stake in this Lecture.
- Sample/Selected projects are proposed in road traffic, railway traffic, Supply chain management and Logistics. The corresponding optimization models derived are further solved numerically using the simulation software MATLAB.

Agreement number: 2017-3516/001-001
Project reference number: 586292-EPP-1-2017-1-PL-EPPKA2-CBHE-JP

INTRAS

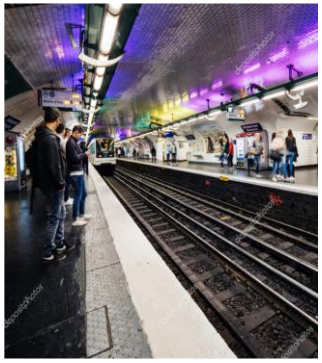
Co-funded by the
Erasmus+ Programme
of the European Union

Chapter 1.
General Introduction

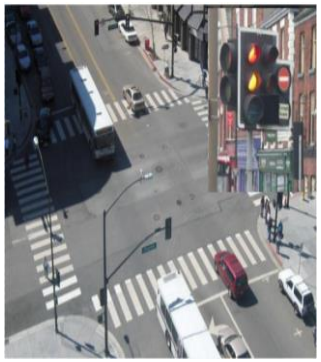
(Full chapter presentation)

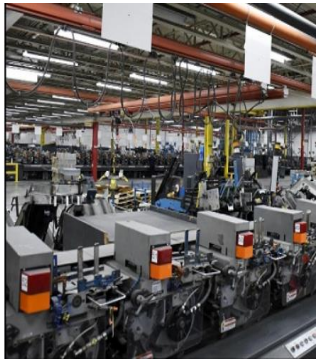
Intelligent Transport Systems: New ICT – based Master's Curricula in Uzbekistan (INTRAS)
Agreement number: 2017-3516/001-001
Project reference number: 586292-EPP-1-2017-1-PL-EPPKA2-CBHE-JP

1.1. Some illustrative examples of: Road traffic, Rail traffic, Air traffic, Boat traffic, and Supply chain (traffic of Goods)







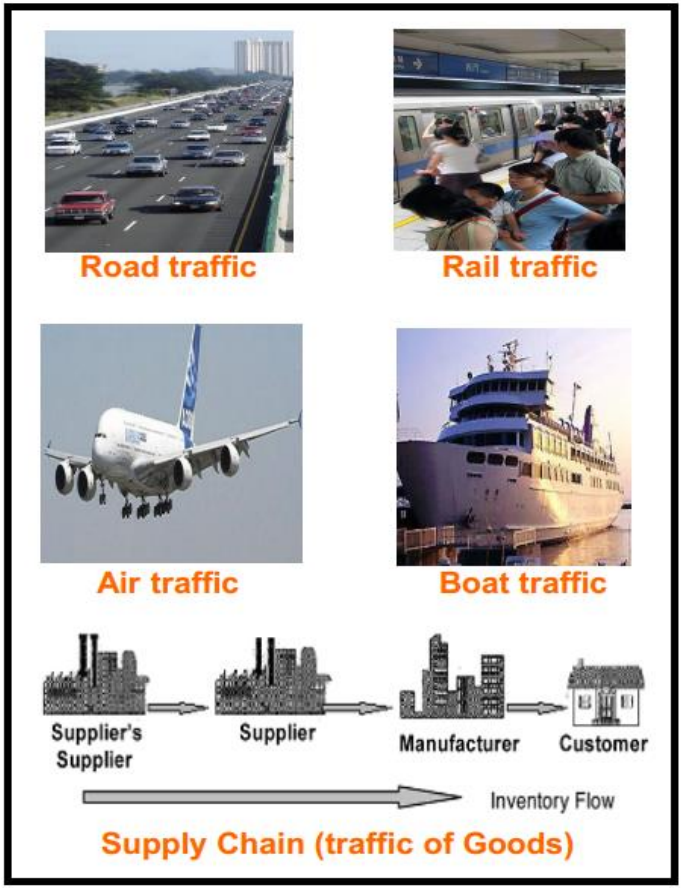
1.2. Definition of some important keywords and their illustration through concrete examples:

- Transportation
- Traffic
- Modelling
- Model
- Informatics
- Traffic theory
- Simulation

1.2. Definition of some important keywords and their illustration through concrete examples:

What is Transportation?

- **Transport (General View):** (Re-) Distribution of matter or other properties of space (e.g. A scalar such as: Mass or energy, A vector such as: Momentum) in space and time.
- **Transportation (General View):** Movement of entities (e.g. People, Goods, Matter, Information/Messages, Energy, Sounds,.....) from one place/point to another.
- **Examples:**
  - ✓ Road transportation
  - ✓ Railway transportation
  - ✓ Waterborne transportation
  - ✓ Air transportation
  - ✓ Information exchange within a supply chain
  - ✓ Etc.



1.2. Definition of some important keywords and their illustration through concrete examples:

What is Traffic?

- **Traffic (General View):** Entities in motion. For example, the movement of entities such as messages/information (in telecommunications & Supply chains), persons/Goods (in Road traffic , Air traffic, Boat traffic, Rail traffic, etc.) from one place/point to another.

What is Modeling?

- **Modeling (General View):** A process of generating abstract, conceptual, graphical, and/or mathematical, models.

What is a Model?

- **A Model is** a Physical, mathematical, graphical, or logical representation of a system entity, phenomenon, or process.
- **A variety of things** are commonly referred to as models: physical objects, fictional objects, set-theoretic structures, descriptions, equations, or combinations of some of these.
- Two models of the same phenomenon may be different (principle of conceptualization).

1.2. Definition of some important keywords and their illustration through concrete examples:

What is Informatics?

- **Informatics (General View):**
  - ✓ Science of information (…the concept of information is closely related to notions of constraint, communication, control, data, instruction, knowledge, meaning, mental stimulus, pattern, perception, representation…),
  - ✓ Practice of information processing,
  - ✓ Engineering of information systems (system of persons, data records,…).

- **Informatics** studies the structure, algorithms, behavior, and interactions of natural and artificial systems that store, process, access and communicate information.

- **Informatics** develops its own conceptual and theoretical foundations and utilizes foundations developed in other fields

1.2. Definition of some important keywords and their illustration through concrete examples:

What is Traffic theory?

- **Traffic theory (General View):** Development of concepts/methods/techniques for the analysis of traffic scenarios/phenomena/events. The aim of the analysis is to:
  - ✓ Understand traffic issues/scenarios/phenomena
  - ✓ Improve traffic issues/scenarios/phenomena
  - ✓ Control traffic issues/scenarios/phenomena
  - ✓ Overcome/tackle/solve traffic problems

What is Simulation?

- **Simulation (General View):** Imitation/Mimic of the reality (i.e., some real things/Behavior/events/phenomena). Examples are state of affairs, behavior, processes, etc. Some concrete examples of simulation types are:
  - ✓ Computer simulation. E.g. simulation of an abstract model of a particular system (algorithmic, numerical codes, etc.)
  - ✓ Simulation game. E.g. simulation (or mimic) of various activities or "real life"

1.3. Pros/Advantages and Cons/Drawbacks of traffic modeling

What are the advantages of traffic modeling?

- **Advantages of traffic modeling (General View):**
  - ✓ Ease control/analysis of traffic systems
  - ✓ Allows the study of the simulated behavior of a traffic system without having the real system. The access to real traffic systems may be very difficult because they are costly or could be difficult to found or to realize.
  - ✓ Less Costly (Financial purposes)
  - ✓ Good security for the Modeler

- **Educational aspect of modeling:**
  - ✓ The modeler learns how to view the natural system under study in its entire complexity and then how to reduce the system to the relevant processes and the relevant parameters.

1.3. Pros/Advantages and Cons/Drawbacks of traffic modeling

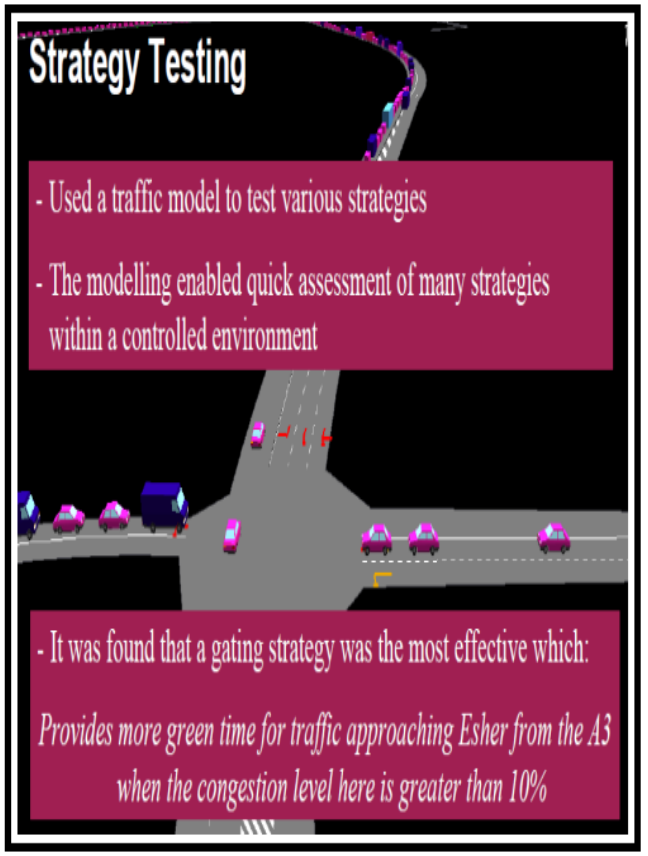
What are the drawbacks of traffic modeling?

- **Drawbacks of traffic modeling** (General View):
  - ✓ Real traffic systems are nonlinear and thus difficult to model.
  - ✓ Nonlinear models to describe real traffic systems are very difficult to propose/obtain (due to the identification process).
  - ✓ Exact analytical solutions of nonlinear models of traffic are impossible (only approximate solutions exist)
  - ✓ Linear models to describe the nonlinear behavior of traffic systems is the most common approach in traffic engineering. This approach cannot give full insights of the real traffic system. Some interesting features such as shockwave, rarefaction wave, torus, chaos, etc.. may not be detected by the mathematical models obtained as results of the modeling process.

1.4. Pros/Advantages and Cons/Drawbacks of traffic simulators

What are the advantages of traffic simulators?

- **Advantages of traffic simulators** (General View):
  - ✓ Simulators can serve as complementary training for drivers
  - ✓ Simulators provide the possibility of opening many research directions
    - Drivers behavior
    - Architecture of roads
    - Pedestrians effects
  - ✓ Simulators will not replace real life training
    - They just help for fast and easy traffic scenarios analyses
  - ✓ Simulator- tools help to meet needs of planning, design, and operational applications
  - ✓ Simulator- tools are with following features: Flexible, Accommodate traffic demands, Easy configuration, good management strategies, etc.



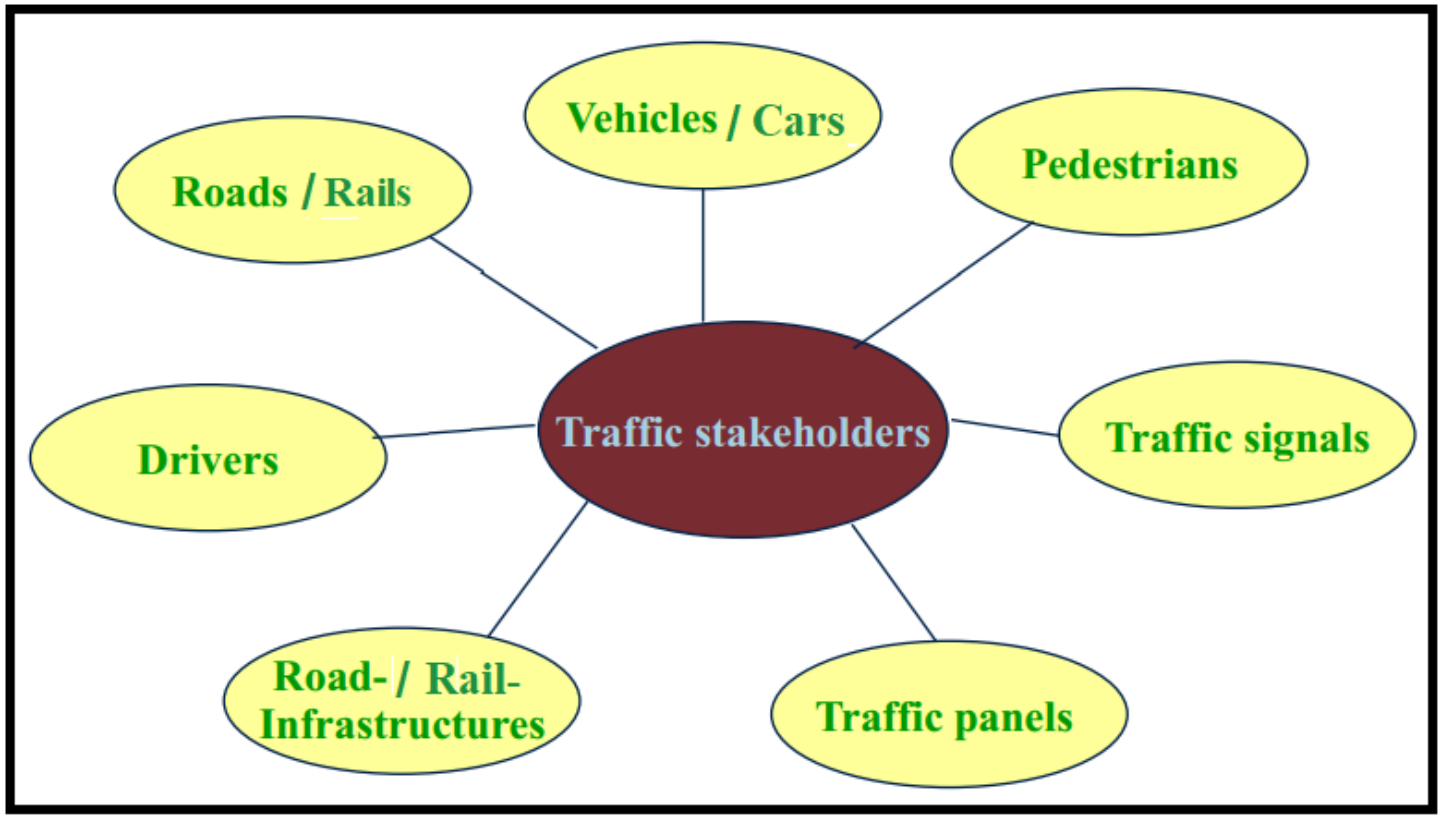
1.4. Pros/Advantages and Cons/Drawbacks of traffic simulators

What are the drawbacks of traffic simulators?

- **Drawbacks of traffic simulators** (General View):
  - ✓ Simulation does not reflect the reality
    - Just a mimic !
  - ✓ Some types of training do not need simulators
    - Because very critical
  - ✓ Professionals do not loss a lot of time
    - Very time consuming
  - ✓ Simulator- tools are not fully reliable
    - Unrealistic driving performances can be simulated.

1.5. Traffic stakeholders and traffic problems: Road- and Rail- traffic.

Selected sample of traffic stakeholders



1.5. Traffic stakeholders and traffic problems: Road- and Rail- traffic.

What are the main traffic problems?

- **Main traffic problems** (General View):
  - ✓ Congestions/queuing
  - ✓ Accidents (Rail- traffic, Road-traffic, Air- traffic, Waterborne traffic, etc....)
  - ✓ Traffic obstructions Rail- traffic, Road-traffic, Air- traffic, Waterborne traffic, etc.)
  - ✓ Packets loss (Telecommunications & information exchange in supply chain management and Logistics)

1.6. Approaches/methods for solving traffic problems.

- **Solution to the congestion problem**:
  - ✓ A realistic solution is integrating traffic management systems.
  - ✓ Another realistic solution is developing new concepts/methods for traffic modeling and simulation, control, optimization, and forecasting.
  - ✓ Another solution might be extending the road/rail network. However, this solution is very expensive/costly
  - ✓ In road transportation, another solution is encouraging people to use public transportation cars instead of private cars.

1.7. Challenges of traffic modeling and simulation.

- **Challenges of traffic modeling** (General View):
  - ✓ Difficult to obtain a model as general framework, which is likely to provide good insights of traffic in all states (i.e. under saturated, saturated, and over saturated traffic states).
  - ✓ The complexity of traffic make-it very difficult to establish/obtain a traffic model, which is likely to depict all insights of the real traffic dynamics. The models provide only an approximate view of the insights of real traffic.
  - ✓ Difficult to obtain a model as a general framework integrating all traffic stakeholders.
  - ✓ Obtaining traffic models is a tedious procedure due to the complexity of traffic.

- **Challenges of traffic simulation** (General View):
  - ✓ Traffic models are very complex and thus very difficult to simulate numerically.
  - ✓ It is difficult to find a traffic simulator/tool, which is likely to simulate traffic at Macroscopic-, Microscopic-, Mesoscopic-, Nanoscopic- levels of details, all simultaneously. Each traffic simulator/tool is built to analyze traffic at a specific level of detail and thus cannot analyze traffic at all level of details simultaneously.
  - ✓ Simulation of dynamic changes in traffic demands is a difficult/challenging task.

1.8. Traffic models classification: Static models; Dynamic models; Continuous model; Discrete model; Deterministic model; Stochastic model; Event-based model.

- **Static models**:
  - ✓ Are used to model average steady-state traffic situations
- **Dynamic models**:
  - ✓ Are used to model changes over time of the traffic situation
- **Discrete models**:
  - ✓ Are used to model traffic situations with finite number of states (dependent variables). Thus the independent variable is expressed as an integer with a finite number of values.
- **Stochastic models**
  - ✓ Are used to capture variation in e.g. reaction time, arrival processes, route choice. But every simulation run results in different outcome, so you need to replicate simulation runs.
- **event based models**
  - ✓ Are used to calculate changes in the system when something 'happens' (events)

Intelligent Transport Systems: New ICT – based Master's Curricula in Uzbekistan (INTRAS)
Agreement number: 2017-3516/001-001
Project reference number: 586292-EPP-1-2017-1-PL-EPPKA2-CBHE-JP

1.9. Physical Interpretation of the modeling procedure: White-box modeling; Black-box modeling; Grey-box modeling.

- Based on their flow and traffic dynamics representation traffic simulation models are characterized as:
  - ✓ **Macroscopic**
    - Fluid representation of flow (Like water flowing through a pipe)
    - Time and space discretization
  - ✓ **Mesoscopic**
    - Individual vehicles (with aggregate behavior) representation
    - Continuous space
    - Usually discrete time
  - ✓ **Microscopic**
    - Individual vehicles (with detailed behavior) representation
    - Traffic dynamics through vehicle interactions and movements
  - ✓ **Nanoscopic**
    - Many common elements with microscopic
    - Detailed representation of vehicle dynamics

1.10. Presentation of traffic simulation tools: Macroscopic; Microscopic; Mesoscopic;

Selected sample of traffic simulation models

AIMSUN 2	Universitat Politècnica de Catalunya,	Spain
ANATOLL	**ISIS and Centre d'Etudes Techniques de l'Equipement**	**France**
ARTEMiS	University of New South Wales, School of Civil Engineering	Australia
ARTIST	***Bosch***	***Germany***
CASIMIR	Institut National de Recherche sur les Transports et la Sécurité	France
CORSIM	***Federal Highway Administration***	***USA***
DRACULA	Institute for Transport Studies, University of Leeds	UK
FLEXSYT II	***Ministry of Transport***	***Netherlands***
FREEVU	University of Waterloo, Department of Civil Engineering	Canada
FRESIM	***Federal Highway Administration***	***USA***
HUTSIM	Helsinki University of Technology	Finland
INTEGRATION	***Queen's University, Transportation Research Group***	***Canada***
MELROSE	Mitsubishi Electric Corporation	Japan
MICROSIM	***Centre of parallel computing (ZPR), University of Cologne***	***Germany***
MICSTRAN	National Research Institute of Police Science	Japan
MITSIMLab	***Massachusetts Institute of Technology***	***USA***
NEMIS	Mizar Automazione, Turin	Italy
PADSIM	***Nottingham Trent University – NTU***	***UK***
PARAMICS	The Edinburgh Parallel Computing Centre and Quadstone	UK

1.10. Presentation of traffic simulation tools: Macroscopic; Microscopic; Mesoscopic;

Selected sample of traffic simulation models

PHAROS	Institute for simulation and training	USA
PLANSIM-T	***Centre of parallel computing (ZPR), University of Cologne***	***Germany***
SIGSIM	University of Newcastle	UK
SIMDAC	***ONERA – Centre d'Etudes et de Recherche de Toulouse***	***France***
SIMNET	Technical University Berlin	Germany
SISTM	***Transport Research Laboratory, Crowthorne***	***UK***
SITRA-B+	ONERA – Centre d'Etudes et de Recherche de Toulouse	France
TRANSIMS	***Los Alamos National Laboratory***	***USA***
THOREAU	The MITRE Corporation	USA
VISSIM	***PTV System Software and Consulting GMBH***	***Germany***
VISUM	***PTV System Software and Consulting GMBH***	***Germany***
LISA++	***PTV System Software and Consulting GMBH***	***Germany***
TRANSYT		UK
SCOOTS	***TRL, Crowthorne, Berkshire (Split Cycle and Offset Optimisation Technique) UK***	
SCATS	RTA, Sydney, NSW (Sydney Coordinated Adaptive Traffic system)	***Australia***
MOTION		***Germany***
SYNCHRO		***USA***

1.10. Presentation of traffic simulation tools: Macroscopic; Microscopic; Mesoscopic;

Selected sample of traffic simulation models

AECOM	Rail Traffic Controller (RTC) by Berkeley Simulation Software	*USA*
AnyLogic	Pedestrian modeling and Railway planning software	*US / Europe*
OpenTrack	Railway planning software	*Switzerland*
AnyLogistix	Software for supply chain modeling	*US / Europe*
PULSim	*User-Based Adaptable Simulation Tool for Railway Planning and Operations*	*Netherlands*
INCONTROL	*Simulation of Railway Networks*	*Netherlands*
Trenissimo	Software for train motion simulation	*Italy*
ROS	*Railway Operation Simulator*	*Japan*
TSW	*Train Simulator World*	*Germany*
RailSys	*Railway planning and routing*	*Germany*
Open Rails	Software for train motion simulation	*US / Europe*
JDA	*Software for Network Design & modeling of Supply Chains*	*USA*
SAP APO	*Software for Advanced Planning and Optimization in Supply Chain Networks*	*USA*
Infor	*Software for Supply Chain Networks Design*	*USA*
Oracle E-Business	*Software for Supply Chain Management*	*USA*
OMP Plus	*Software for Supply Chain Networks Design*	*BELGIUM*

1.11. Presentation of microscopic traffic simulation model

- **Microscopic view.** Maps traffic flow as a set of individual vehicles
  - ✓ Represents
    - integrated networks in detail (freeways and urban streets)
    - wide range of surveillance and control devices
    - various vehicle classes (e.g information access)
  - ✓ Models individual vehicle movements
    - car-following
    - lane changing
    - gap acceptance
    - Merging
    - Buses
  - ✓ Captures travel and driver behavior
    - pre-trip
    - en-route route choice
    - aggressiveness and compliance

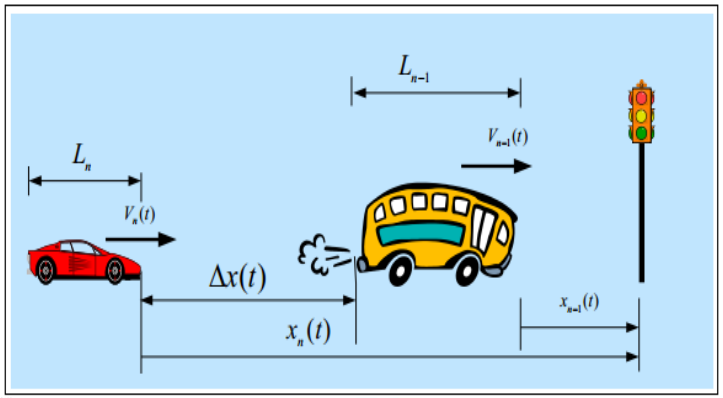
1.11. Presentation of microscopic traffic model

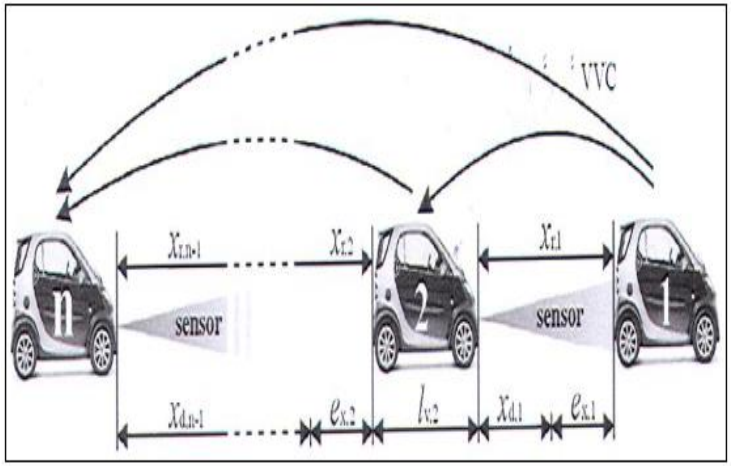
Selected sample of traffic models

- **Mathematical models for microscopic traffic analysis**
  - ✓ Car-Following Model consists of measurement of both Time and Space headways. Three regimes are observed: Free-flow regime, Car-following regime, and Emergency regime.
  - ✓ Coupled model for cooperative ACC

$$\begin{cases} \dfrac{d^2 x_{i-1}}{dt^2} = K_1 \cdot (x_{i-2} - x_{i-1} - h_d \cdot v_{i-1}) + K_2 \cdot (v_{i-2} - v_{i-1}) \quad (1) \\ \dfrac{d^2 x_i}{dt^2} = K_1 \cdot (x_{i-1} - x_i - h_d \cdot v_i) + K_2 \cdot (v_{i-1} - v_i) \quad (2) \end{cases}$$

➢ Acceleration = φ (distance, speed....)



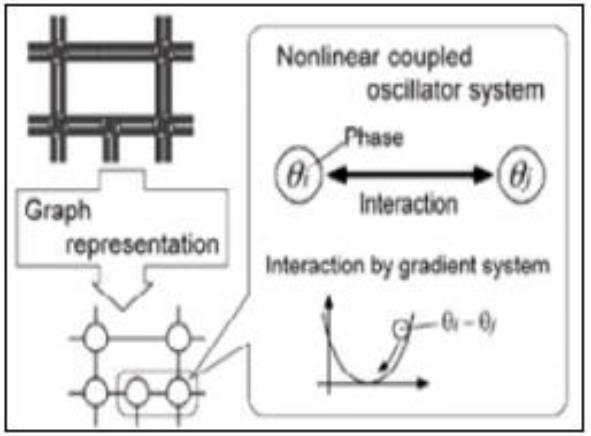


1.11. Presentation of microscopic traffic model

Selected sample of traffic models

- **Mathematical models for microscopic traffic analysis: Modeling traffic control**
  - ✓ Synchronized coupled oscillators (KURAMOTO) can be used to model and optimize real-time self-organized traffic control in a city.
  - ✓ The KURAMOTO model is efficient for pretimed-control. In contrast the KURAMOTO model is not accurate enough for actuated- control.
  - ✓ traffic control in a city.
  - ✓ The KURAMOTO model of the coupled signals at junctions is expressed as follows:

$$\dot{\theta}_i = \omega_i + \sum_{j=1}^{N} K_{ij} \sin(\theta_j - \theta_i), \quad i = 1, \ldots, N$$

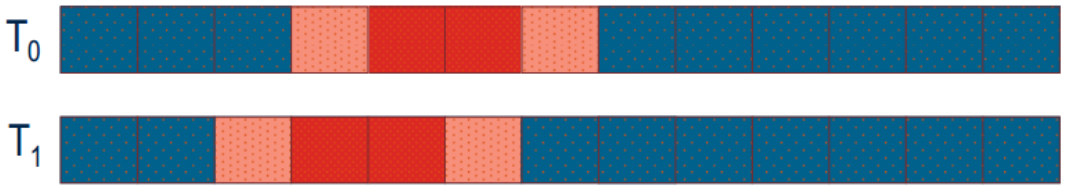


1.12. Advantages and Drawbacks of using the microscopic Car-Following model

- **Advantages**
  - ✓ Ease processing of heterogeneous vehicles
  - ✓ Kinematics of realistic vehicles
  - ✓ Easy implementation of stochastic aspects
  - ✓ Valid for a single lane
- **Drawbacks**
  - ✓ Cannot model overtaking
  - ✓ Cannot model the Driver behavior
  - ✓ Difficulty of calibration and validation (stability ...)
  - ✓ When difference in speed = 0, the acceleration = 0 even if the distance is very small
  - ✓ When small fluctuations in speed-difference result in changing the acceleration : unrealistic that driver can perceive small changes
  - ✓ Drivers are 'dragged along' if the leader accelerates

1.13. Some illustrative examples of macroscopic traffic simulation models

- Types:
  - ✓ Gas-kinetic differential equations (e.g. Prigogine & Herman)
  - ✓ Fluid dynamics differential equations (e.g. Lighthill, Whitham & Richards)
- Discretized over time and space
- Large networks, limited detail



- Maps traffic flow as a continuous unity of fluidized vehicles.
- No vehicle in the traffic flow is identifiable.
- The traffic flow is characterized by macroscopic state values like density, volume and mean velocity.
- The traffic flow is treated as a continuous fluid flow in fluid dynamics.

1.13. Some illustrative examples of macroscopic traffic simulation models

- The three variables are:
  - ✓ Flow: Q(x,t),
  - ✓ Density: K(x,t),
  - ✓ Speed: U(x,t)

- The fundamental relationship between the three variables is as follows:
  - ✓ $\frac{\partial Q}{\partial x} + \frac{\partial K}{\partial t} = 0$ (Conservation equation)
  - ✓ Additional relation between: $Q=KU$)

- Need of a third equation is of necessary importance
  - ✓ Equilibrium relationship $Q = Qe(K)$
  - ✓ $U = Ue(k)$ => first-order models (LWR)

1.14. Advantages and Drawbacks of using the macroscopic model (LWR)

- **Advantages**
  - ✓ Macroscopic models are more suitable for the entire network. In the network-wide application, the characteristics of the global data stream (e.g., shockwaves, rarefaction waves, congestion, delay, etc.) can be easily examined and displayed.
  - ✓ The numerical simulation of macroscopic models leads to lower computational costs than the computational effort of microscopic and mesoscopic models.

- **Drawbacks**
  - ✓ Cannot provide full insight of real- traffic scenario
  - ✓ Not efficient to analyze real- traffic scenarios
  - ✓ Only a global view is provided
  - ✓ Both Time & Space discretization lead to the loose of some key insights or features of the real- traffic dynamics

INTRAS

Co-funded by the
Erasmus+ Programme
of the European Union

Chapter 2.

Overview on traffic models and traffic simulation tools

(Chapter's detailed description)

Intelligent Transport Systems: New ICT – based Master's Curricula in Uzbekistan (INTRAS)
Agreement number: 2017-3516/001-001
Project reference number: 586292-EPP-1-2017-1-PL-EPPKA2-CBHE-JP

2.1. Models of specific road/rail traffic scenarios

- ✓ This section presents and describes some commonly used model in road traffic and railway traffic. Sample modelsare:
  - The train running model,
  - The model of a double-train macroscopic representation emulator (D-TMRE)
  - The Car following model
  - The LWR model
  - The Greenshields model
- ✓ A focus is devoted to Microscopic, Macroscopic, and Mesoscopic levels of details.

2.2. Three phase traffic theory

- ✓ This section explains how the traffic states, namely Free flow, Synchronized flow, and Wide moving Jam manifest themselves in rail traffic and in road traffic, respectively. Concrete examples in railway traffic and road traffic are provide for illustration.

2.3. Some basic terminologies in road and/or rail traffic

- ✓ The following key words are defined in the context of road traffic and railway traffic: Cycle; Cycle length/time; Capacity; Congestion; Coordination; Delay; Density; Detection; Phase; Phase sequence; Split; Phase split; Approach; Actuation; Gap; Gap out; Level of service; Shockwave; Offset of traffic signals; Ramp metering; Pretimed controllers; actuated controllers; semi-actuated controllers; Fully-actuated controllers; Inductive loops; Camera-based controllers;
- ✓ A concrete illustrative example is provided to support each definition.

2.4. Description of some commonly used traffic simulation tools

- ✓ In chapter 1 a list of some commonly used simulation tools has been provided in the fields of railway traffic (1), road traffic (2) and supply chain management & Logistics (3). The aim of this section is to choose at least three simulation tools in each of the aforementioned three fields and describe them.

INTRAS

Co-funded by the
Erasmus+ Programme
of the European Union

Chapter 3.

Basics of neural networks and application in transportaion

(Chapter's detailed description)

Intelligent Transport Systems: New ICT – based Master's Curricula in Uzbekistan (INTRAS)
Agreement number: 2017-3516/001-001
Project reference number: 586292-EPP-1-2017-1-PL-EPPKA2-CBHE-JP

3.1. Introduction to neural networks

- ✓ This section defines the neural network (NN) paradigm, and presents the history of artificial neural networks (ANNs). Some concrete applications of NN are presented.

3.2. Neural networks structures

- ✓ Several structures of Neural Networks (e.g., NN, ANN, Biological NN (BNN)) are presented in this section along with a description of their functioning principle. The fundamental difference between ANN and BNN is further presented.

3.3. Learning/Training phase of artificial neural networks (ANNs)

- ✓ The section presents the three main models of machine learning, namely Unsupervised learning, Supervised learning, and Reinforcement learning. The Learning rules of the neural network (e.g., XOR gate, mean squared error, a system of differential equations) are key/fundamental factors used to improve the Neural Network (NN) performances (e.g., the training time and the accuracy, etc.).

3.4. Usage phase of artificial neural networks (ANNs)

- ✓ ANNs are characterized by a „training phase“ and a „usage phase“. This section is focussed on the usage phase of ANNs. In particular, some concrete ANNs models are presented as case studies and their convergence properties are analyzed.

3.5. Neuron model and network architecture

- ✓ This section present the model of a basic/elementary neuron. The single-input neuron and the multiple-input neuron are analysed and their transfer functions are determined.

3.6. Network architecture

- ✓ The section 3.5 has presented the model of a basic/elementary neuron. The basic neurons are used in this section to design a network architecture. Indeed the basic neurons are further coupled together to form a network architecture. Several network architectures are analysed: A network with one layer and a network with multiple layers. The recurrent neural network is also analysed as new architecture.

3.7. Convergence of neural network architecture (or platform)

- ✓ The section is devoted to the analysis of the convergence properties of both single-layer and multi-layer neural networks architectures.

3.8. Sample application examples of how to solve problems using neural networks

- ✓ The section proposes some concrete problems to be solved using neural nerworks. These problems are: (1) Optimisation of a linear function; (2) Optimization of a nonlinear functions; (3) Shortest path in a Graph; (4) Solving equations (ODEs).

3.9. Some exercises to check the knowledge acquired in the chapter

- ✓ Many exercises/problems are proposed to be solved using Neural networks

INTRAS

Co-funded by the
Erasmus+ Programme
of the European Union

Chapter 4.

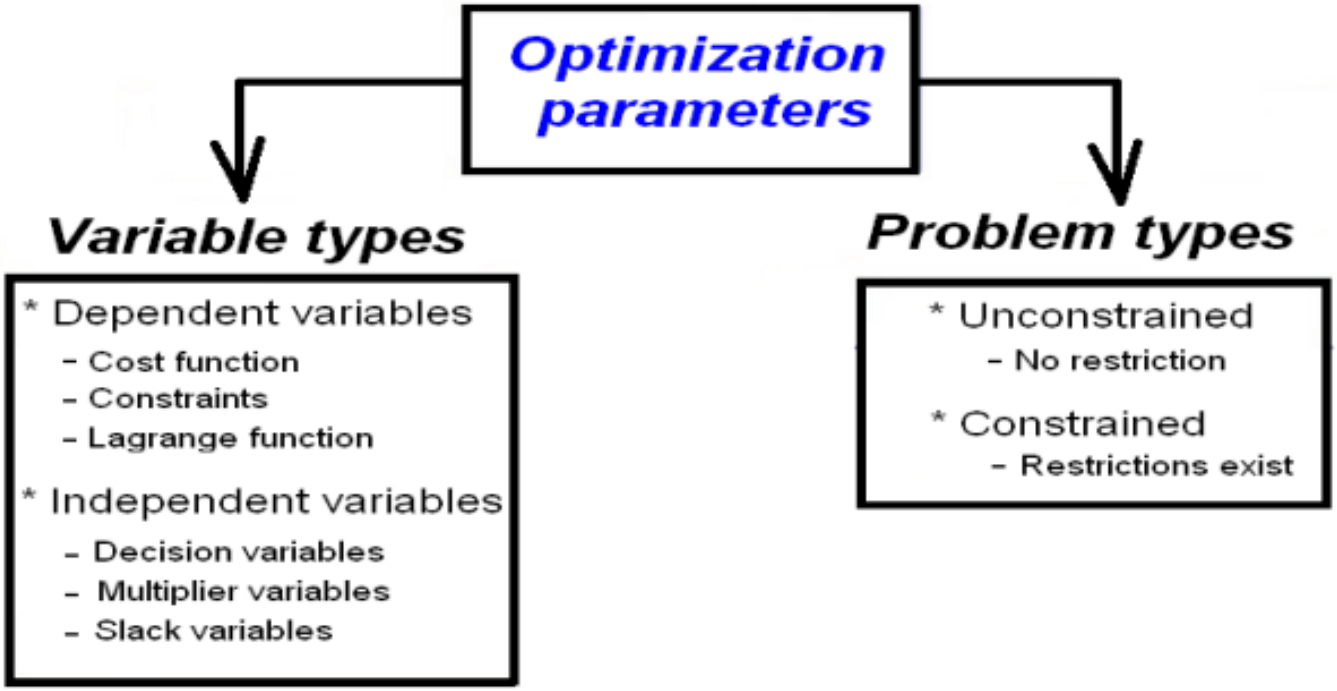
Basics of optimization and simulation algorithms/tools for optimization

(Full chapter presentation)

Intelligent Transport Systems: New ICT – based Master's Curricula in Uzbekistan (INTRAS)
Agreement number: 2017-3516/001-001
Project reference number: 586292-EPP-1-2017-1-PL-EPPKA2-CBHE-JP

4.1. Optimization and related parameters

- ✓ **Optimization is:**
  - a process of seeking for the best solution/result under given circumstances.
  - a process of making decision as fully perfect or as effective as possible.
- ✓ ***Mathematical programming techniques*** are classical optimization methods, which are commonly used in the filed of ***operations research***. This is a branch of Mathematics in which methods/techniques are developed and applied to decision making problems in order to obtain the best/optimum results/solutions.



4.2. Classical Methods used in Operations Research (1)

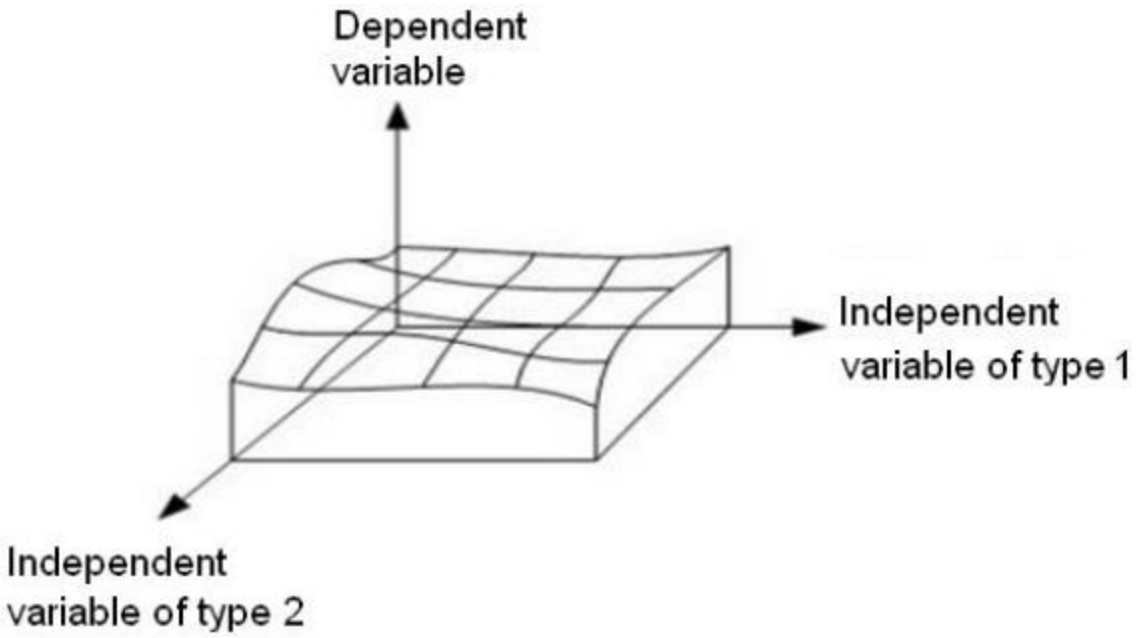
- Commonly used methods in ***Operations Research*** are:
  - ✓ mathematical modeling
  - ✓ Statistics
  - ✓ Algorithms generally based on ***"Max"*** or ***"Min"*** principles/objectives
    - The maxima
      - profit (or incomes)
      - faster assembly line,
      - higher bandwidth, etc.
      - higher life-time
    - The minima
      - cost loss (or expenses)
      - shortest path
      - Traveling salesman problem (TSP) tour
      - energy (power) consumption
      - lowering of risk, etc.
  - ✓ ***The filed of operations research*** develops mathematical techniques, which are used to determine the best possible solution to a problem. In engineering, this problem is generally concerned with the improvement (or optimization) of the system's performance.

4.2. Classical Methods used in Operations Research (2)

Mathematical Programming Techniques	Stochastic Process Techniques	Statistical Methods
Calculus methods	Statistical decision theory	Regression analysis
Calculus of variations	Markov processes	Cluster analysis
Nonlinear programming	Queueing theory	Pattern recognition
Geometric programming	Renewal theory	Design of experiments
Quadratic programming	Simulation methods	Discriminate analysis (factor analysis)
Linear programming	Reliability theory	
Dynamic programming		
Integer programming		
Stochastic programming		
Separable programming		
Multiobjective programming		
Game theory		
Simulated annealing		
Genetic algorithms		
Neural networks		

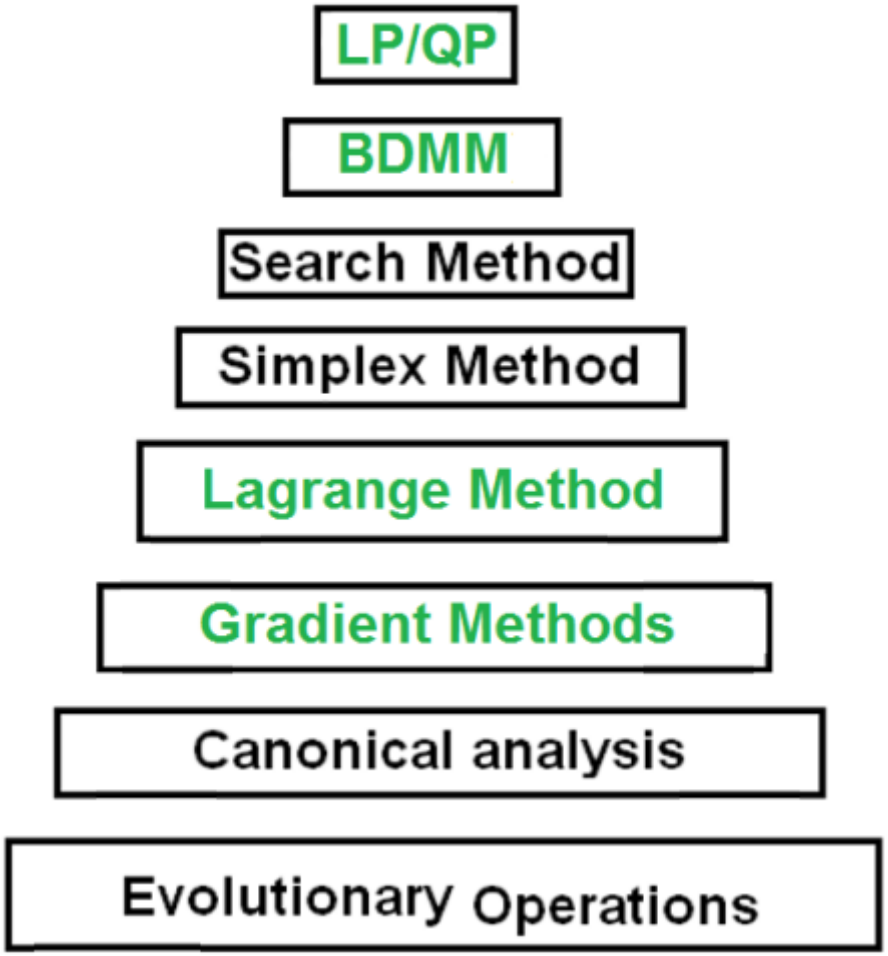
4.3. 3D representation of the optimization problem

- ✓ The 3D representation is an expression of the total function in terms of independent variables (i.e. decision and multiplier variables).
  - the total function represents the energy of the system.
  - total function combines the objective/cost function with constraints (of all types).



Co-funded by the
Erasmus+ Programme
of the European Union

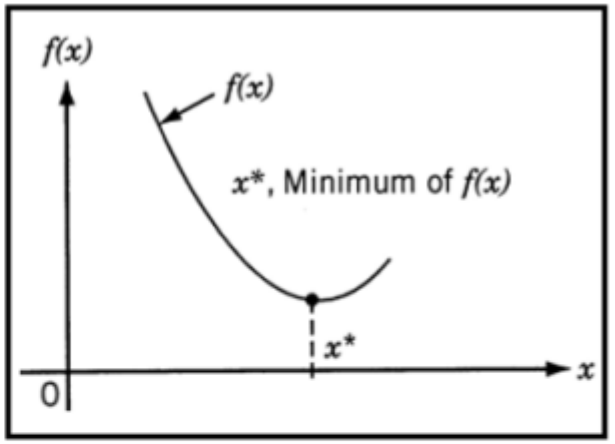
4.4. Some classical optimization methods



4.5. Mathematical formulation of a constrained optimization problem (1)

- **A constrained optimization problem can be formulated as:**

$$\begin{cases} \text{minimize } f(x) \\ \text{subject to } g_i(x) \le b_i, \ i = 1,\ldots,m \end{cases}$$



* $f_0 : \boldsymbol{R}^n \rightarrow \boldsymbol{R}$: objective function
* $x=(x_1,\ldots,x_n)$: Decision variables (unknowns of the problem)
* $g_i : \boldsymbol{R}^n \rightarrow \boldsymbol{R}$: $(i=1,\ldots,m)$: Constraints (imposed by the problem)

4.5. Mathematical formulation of a constrained optimization problem (2)

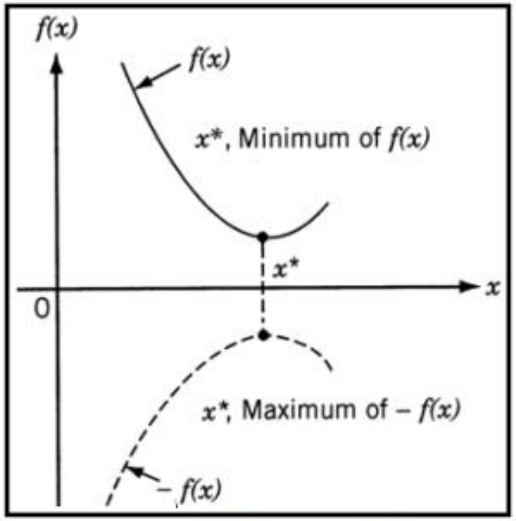
- **An equivalent formulation of the same optimization problem:**

$$\begin{cases} \text{maximize } [f(x)] \\ \text{subject to } g_i(x) \le b_i, \\ i = 1,\ldots,m \end{cases} \Leftrightarrow \begin{cases} \text{minimize } [-f(x)] \\ \text{subject to } g_i(x) \le b_i, \\ i = 1,\ldots,m \end{cases}$$

$$\exists^{x^* \in \Re} /$$

$$Min[f(x^*)] = Max[-f(x^*)]$$

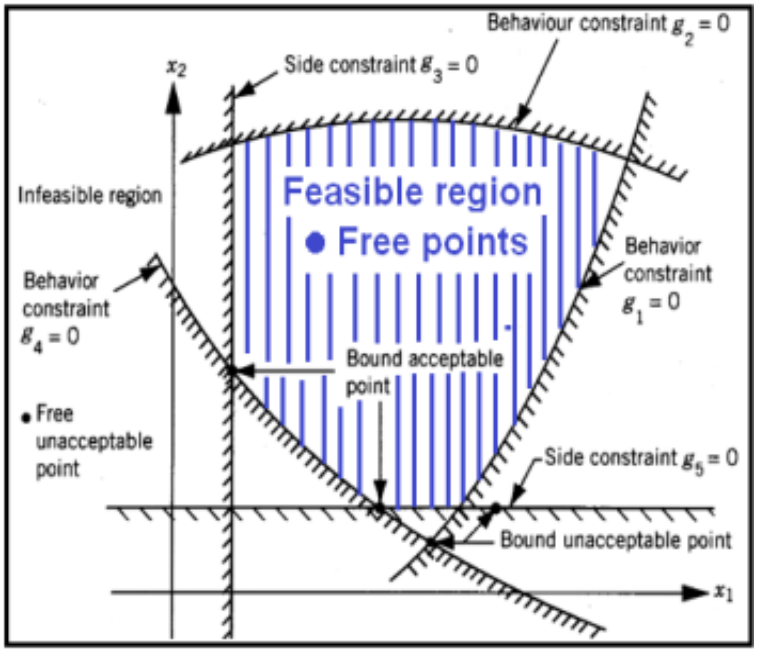
(See figure 2)



* $f: \boldsymbol{R^n} \rightarrow \boldsymbol{R}$: objective function
* $x=(x_1,\ldots,x_n)$: Decision variables (unknowns of the problem)
* $g_i: \boldsymbol{R^n} \rightarrow \boldsymbol{R}$: $(i=1,\ldots,m)$: Constraints (imposed by the problem)

4.6. "Behavior constraints" and "Side constraints"

- **Behavior constraints (also called functional constraints):**
  - ✓ These constraints are used to limit the behavior (or performance) of the system.
  - ✓ Example: $\begin{cases} g_i(x) \le b_i \\ g_i(x) = b_i \\ g_i(x) \ge b_i \end{cases}$ ($g_i(x)$ are constraints)



(Illustration of the different types of constraints)

- **Side constraints:**
  - ✓ These constraints represent physical limitations on design variables such as manufacturing limitations. They are also used to limit (or fix) the range of variation of decision variables.
  - ✓ Example: $\begin{cases} x_i \le b_i \\ x_i = b_i \\ x_i \ge b_i \end{cases}$ (x_i are decision variables)

4.7. Comments on optimization findings

- The objective of the design procedure for optimization is to find an appropriate design, which fulfils/satisfies both „functional requirement“ and „other additional requirements“ of the problem.

- Several acceptable design procedures are generally obtained. Thus, the optimization is used in order to choose the best design procedure.

- The criterion which is generally used to benchmark all the acceptable design procedures in order to choose the best is called: the **objective function** (or **cost function**). This function is expressed in terms of the **design variables** (called **decision variables**).

4.8. Linear programming (LP) problem

- ✓ A linear programming (LP) problem is a problem with a linear objective function and linear constraints.
- ✓ The LP problem can be expressed mathematically as follows:

$$F(\mathbf{X}) = \alpha_0 + \sum_{i=1}^{n} \alpha_i x_i$$

subject to

$$\sum_{i=1}^{n} \beta_{ij} x_i = \gamma_j, \quad j = 1,2,\cdots,m$$

$$x_i \geq 0, \qquad i = 1,2,\cdots,n$$

α_i , β_{ij} and γ_j are constants.

Intelligent Transport Systems: New ICT – based Master's Curricula in Uzbekistan (INTRAS)
Agreement number: 2017-3516/001-001
Project reference number: 586292-EPP-1-2017-1-PL-EPPKA2-CBHE-JP

4.9. Quadratic programming (QP) problem

- ✓ A quadratic programming (QP) problem is a nonlinear programming problem with a quadratic objective function and linear constraints.
- ✓ The QP problem can be expressed mathematically as follows:

$$F(\mathbf{X}) = \alpha_0 + \sum_{i=1}^{n} \alpha_i x_i + \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij} x_i x_j$$

subject to

$$\sum_{i=1}^{n} \beta_{ij} x_i = \gamma_j, \quad j = 1,2,\cdots,m$$

$$x_i \geq 0, \qquad i = 1,2,\cdots,n$$

α_i , β_{ij} , γ_j , and σ_{ij} are constants.

4.10. Integer programming problem

- ✓ All design variables (i.e. decision and multiplier variables) of the optimization problem are integers (i.e. discrete values).

4.11. Real-valued programming problem

- ✓ All design variables (i.e. decision and multiplier variables) of the optimization problem are real values.

4.12. Stochastic programming problem (See Lecture 3)

- ✓ All design variables (i.e. decision and multiplier variables) of the optimization problem are probabilistic (non-deterministic). These variables are described by probability distributions (Gaussian, Poisson, Erlang, Markov, etc.)

4.13. The „cost“ or „objective“ function in Engineering

- **Civil engineering and Transportation:**
  - ✓ The aim is generally to minimize: Time, Costs, Distance, Energy, Pollution, etc.

- **Mechanical engineering:**
  - ✓ The aim is generally to maximize: Mechanical efficiency, Life-time, etc.

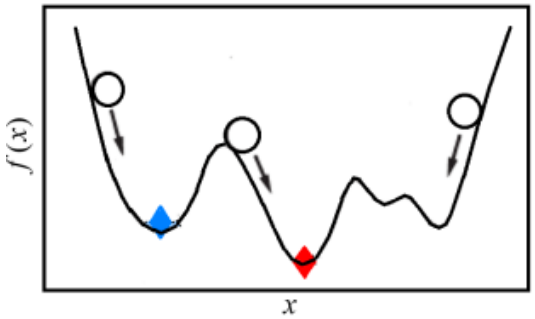
In case several criteria must be satisfied simultaneously, each of which is expressed by the objective function , the optimization problem is defined using a **multi-objective function** expressed as follows:

$$f(\mathbf{X}) = \sum_{i=1}^{m} \alpha_i f_i(\mathbf{X})$$

$\mathbf{X}$: Represents the design variables (i.e. decision variables).

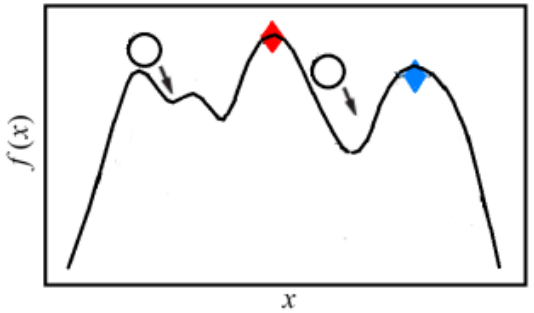
α_i : Constant values expressing the relationship between functions $f_i(\mathbf{X})$

4.14. „Global minimum vs. Local minima“ ; „Global maximum“ vs. „Local maxima“



Global minimum

Local/relative minima



Global maximum

Local/relative maxima

- The function $f(x)$ has a global (or absolute) minimum denoted by $f(x^*)$ at a specific point x^* if and only if : $f(x^*) \leq f(x)$ for all x.
- The function $f(x)$ has a global maximum denoted by $f(x^*)$ at a specific point x^* if and only if : $f(x^*) \geq f(x)$ for all x.

Challenge of optimization: How to avoid **„local minima“** and **„local maxima“**?

4.15. The Hessian Matrix as a test condition

- ✓ The test of the „positive definiteness of the Hessian matrix“ of the objective function at the extremum point can help to know whether the "extremum point" is a "maximum" or a "minimum".
- ✓ The Hessian matrix for an optimization problem with two design variables ($x1$, $x2$) is defined as follows ($f(x1, x2)$ is the objective function):

$$\mathbf{J}\Big|_{(x_1^*, x_2^*)} = \begin{bmatrix} \dfrac{\partial^2 f(x_1, x_2)}{\partial x_1^2} & \dfrac{\partial^2 f(x_1, x_2)}{\partial x_1 \partial x_2} \\ \dfrac{\partial^2 f(x_1, x_2)}{\partial x_1 \partial x_2} & \dfrac{\partial^2 f(x_1, x_2)}{\partial x_2^2} \end{bmatrix}_{(x_1^*, x_2^*)}$$

$$\mathbf{J_1} = \left| \frac{\partial^2 f(x_1, x_2)}{\partial x_1^2} \right|_{(x_1^*, x_2^*)} \qquad \mathbf{J_2} = \begin{vmatrix} \dfrac{\partial^2 f(x_1, x_2)}{\partial x_1^2} & \dfrac{\partial^2 f(x_1, x_2)}{\partial x_1 \partial x_2} \\ \dfrac{\partial^2 f(x_1, x_2)}{\partial x_1 \partial x_2} & \dfrac{\partial^2 f(x_1, x_2)}{\partial x_2^2} \end{vmatrix}_{(x_1^*, x_2^*)}$$

- ✓ The values of $\mathbf{J_1}$ and $\mathbf{J_2}$ are used to deduce the nature of the extrema points **(see application examples below)**.

4.16. The Hessian Matrix as a test condition: Application (1)

Example 1:

- Consider the function $f(x_1, x_2)$ and determine the nature of critical points.

$$f(x_1, x_2) = x_1^3 + x_2^3 + 2x_1^2 + 4x_2^2 + 6$$

Solution 1:

$$\mathbf{J} = \begin{bmatrix} 6x_1 + 4 & 0 \\ 0 & 6x_1 + 8 \end{bmatrix}_{(x_1^*, x_2^*)}$$

$$\mathbf{J_1} = \left| 6x_1 + 4 \right|_{(x_1^*, x_2^*)}$$

$$\mathbf{J_2} = \begin{vmatrix} 6x_1 + 4 & 0 \\ 0 & 6x_1 + 8 \end{vmatrix}_{(x_1^*, x_2^*)}$$

Table. Nature of the 4 extreme points

X	J1	J2	Nature of J	Nature of X	f (X)
(0 , 0)	+4	+32	**Positive definite**	**Local/Relative minimum**	6
(0 , -8/3)	+4	-32	**Indefinite**	**Saddle point**	15.4815
(-4/3 , 0)	-4	-32	**Indefinite**	**Saddle point**	7.1852
(-4/3 , -8/3)	-4	+32	**Negative definite**	**Local/Relative maximum**	16.6667

4.16. The Hessian Matrix as a test condition: Application (2)

Example 2:

- Calculate the Hessian matrix of the function $f(x_1, x_2)$. Analyse the convex or concave form of $f(x_1, x_2)$.

$$f(x_1, x_2) = 2x_1^3 - 6x_2^2$$

Solution 2:

$$\mathbf{J} = \begin{bmatrix} 12x_1 & 0 \\ 0 & -12 \end{bmatrix}_{(x_1^*, x_2^*)}$$

$$\mathbf{J_1} = \left|12x_1\right|_{(x_1^*, x_2^*)}$$

$$\mathbf{J_2} = \begin{vmatrix} 12x_1 & 0 \\ 0 & -12 \end{vmatrix}_{(x_1^*, x_2^*)}$$

For $x_1 \le 0$, **H(X) is** negative semidefinite and $f(x_1, x_2)$ is concave.

$$\begin{cases} J_1 = \dfrac{\partial^2 f}{\partial x_1^2} = 12x_1 \le 0 \\ J_2 = -144x_1 \ge 0 \end{cases} \quad (\text{for } x_1 \le 0)$$

$$\begin{cases} J_1 = \dfrac{\partial^2 f}{\partial x_1^2} = 12x_1 \ge 0 \\ J_2 = -144x_1 \le 0 \end{cases} \quad (\text{for } x_1 \ge 0)$$

Intelligent Transport Systems: New ICT – based Master's Curricula in Uzbekistan (INTRAS)
Agreement number: 2017-3516/001-001
Project reference number: 586292-EPP-1-2017-1-PL-EPPKA2-CBHE-JP

4.17. The Lagrangian method for optimization


Joseph-Louis Lagrange

- **Joseph-Louis Lagrange (1736-1813)**

  - ✓ The main important contributions made by the scientist are:
    1. Calculus of variations,
    2. Minimization of functionals,
    3. Method of optimization for constrained problems)

  - ✓ The Lagrangian method is a commonly used mathematical approach for optimization. The overall approach involves the following 6 steps:

1. Determination of the objective/cost function
2. Determination of constraints
3. Determination of the Lagrange function
4. Partial derivation/differentiation of the Lagrange function
5. Solving the set of equations (obtained through partial differentiation)
6. Calculation of the optimum/best solution

4.18. The Lagrangian method: Case of equality constraints

- **A constrained optimization problem can be formulated as:**

$$\begin{cases} \text{minimize } f_0(x) \\ \text{subject to } g_i(x) = b_i, \quad i = 1,\ldots,m \end{cases}$$

- **The corresponding Lagrange function is formulated as follows:**

$$\exists^{\lambda_i \in \Re} / \quad L(x,\lambda) = f_0(x) + \sum_{i=1}^{m} \lambda_i \left[g_i(x) - b_i \right]$$

* f_0 : $\boldsymbol{R^n \longrightarrow R}$: objective function
* $x=(x_1,\ldots..,x_n)$: Decision variables (unknowns of the problem)
* g_i : $\boldsymbol{R^n \longrightarrow R}$: $(i=1,\ldots,m)$: Constraints (imposed by the problem)
* λ_i: Lagrange Multiplier variables (unknowns of the problem)

4.19. The Lagrangian method with one constraint: Application (1)

Example 3:

- Consider an optimization problem with known „cost function“ $f(x_1, x_2)$ and known „constraints“ $g(x_1, x_2)$.

$$\begin{cases} f(x_1, x_2) = 3x_1^2 + 6x_2^2 - x_1 x_2 \\ g(x_1, x_2) = x_1 + x_2 - 20 \end{cases}$$

- Use the Lagrangian Method to determine the optimum value of the cost function.
- Is the optimum value **„a maximum“** or **„a minimum“**?. Justifies your answer.

Intelligent Transport Systems: New ICT – based Master’s Curricula in Uzbekistan (INTRAS)
Agreement number: 2017-3516/001-001
Project reference number: 586292-EPP-1-2017-1-PL-EPPKA2-CBHE-JP

4.19. The Lagrangian method with one constraint: Application (1)

Solution 3:

- **Step 3**: Determination of the Lagrange function $L(x_1, x_2, \lambda)$.

$$L(x_1, x_2, \lambda) = f(x_1, x_2) + \lambda * g(x_1, x_2)$$

- **Step 4**: Partial differentiation of the Lagrange function.

$$\begin{cases} \dfrac{\partial L(x_1, x_2, \lambda)}{\partial x_1} = 6x_1 - x_2 + \lambda \\ \dfrac{\partial L(x_1, x_2, \lambda)}{\partial x_2} = 12x_2 - x_1 + \lambda \\ \dfrac{\partial L(x_1, x_2, \lambda)}{\partial \lambda} = x_1 + x_2 - 20 \end{cases}$$

4.19. The Lagrangian method with one constraint: Application (1)

Solution 3:

- **Step 5**: Solving of the equations obtained through partial derivatives:

$$\begin{cases} \dfrac{\partial L(x_1, x_2, \lambda)}{\partial x_1} = 0 \\ \dfrac{\partial L(x_1, x_2, \lambda)}{\partial x_2} = 0 \\ \dfrac{\partial L(x_1, x_2, \lambda)}{\partial \lambda} = 0 \end{cases} \Longrightarrow \exists^{(x_1^*, x_2^*, \lambda^*) \in \Re} / \begin{cases} 6x_1^* - x_2^* + \lambda^* = 0 \\ 12x_2^* - x_1^* + \lambda^* = 0 \\ x_1^* + x_2^* - 20 = 0 \end{cases}$$

- **Step 6**: Calculation of the optimum solution.

Solution 3:

$$\begin{cases} x_1^* = 13 \\ x_2^* = 7 \\ f(x_1, x_2)\big|_{optimum} = f(x_1^*, x_2^*) = 710 \end{cases}$$

4.20. The Lagrangian method with two constraint: Application (2)

Example 4:

- Consider an optimization problem with known „cost function“ $f(x,y,z)$ and known „constraints“ $g(x,y,z)$ and $h(x,y,z)$.

$$\begin{cases} f(x,y,z) = x^2 + y^2 + z^2 - 2xy - 2yz \\ g(x,y,z) = x - y = 1 \\ h(x,y,z) = y = 1 \end{cases}$$

Solution 4:

- The Lagrange function is expressed as follows:

$$L(x,y,z) = \left(x^2 + y^2 + z^2 - 2xy - 2yz\right) + \lambda_1 (x - y - 1) + \lambda_2 (y - 1)$$

- The partial derivatives are expressed as follows:

$$\frac{\partial L}{\partial x} = 2x - 2y + \lambda_1 \qquad \frac{\partial L}{\partial z} = 2z - 2y \qquad \frac{\partial L}{\partial \lambda_2} = y - 1$$

$$\frac{\partial L}{\partial y} = 2y - 2x - \lambda_1 + \lambda_2 - 2z \qquad \frac{\partial L}{\partial \lambda_1} = x - y - 1$$

4.20. The Lagrangian method with two constraint: Application (2)

Solution 4:

- Considering all üpartial derivatives equal to zero leads to the following critical $(x^*, y^*, z^*, \lambda_1^*, \lambda_2^*)$ point. This point is obtained as solution of the following algebraic equation:

$$\begin{cases} 2x-2y+\lambda_1=0 \\ -2x+2y-2z-\lambda_1+\lambda_2=0 \\ -2y+2z=0 \\ x-y-1=0 \\ y-1=0 \end{cases} \Leftrightarrow \begin{bmatrix} 2 & -2 & 0 & 1 & 0 \\ -2 & 2 & -2 & -1 & 1 \\ 0 & -2 & 2 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{pmatrix} x^* \\ y^* \\ z^* \\ \lambda_1^* \\ \lambda_2^* \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

- Solving the algebraic system above leads to the following solution:

$$(x^*,\ y^*,\ z^*,\ \lambda_1^*,\ \lambda_2^*) = (2\ ,\ 1\ ,\ 1\ ,\ -2\ ,\ 2)$$

- The value of the optimum value of the function is obtained at the critical point.
- The optimum value corresponds to: $L_{opt} = 0$

Intelligent Transport Systems: New ICT – based Master's Curricula in Uzbekistan (INTRAS)
Agreement number: 2017-3516/001-001
Project reference number: 586292-EPP-1-2017-1-PL-EPPKA2-CBHE-JP

4.21. The Lagrangian method: Case of inequality constraints (1)

- Consider the following optimization problem with inequality constraints:

$$\begin{cases} \text{minimize } f_0(x) \\ \text{subject to } g_i(x) \le 0, \quad i = 1,....,m \end{cases}$$

- The problem can be transformed into an optimization problem with equality constraints as follows:

$$\begin{cases} \text{minimize } f_0(x) \\ \text{subject to } g_i(x) + y_i^2 = 0, \quad i = 1,....,m \end{cases}$$

- y_i^2 are non-negative slack variables with unknown values.

4.21. The Lagrangian method: Case of inequality constraints (2)

- Consider the following optimization problem with inequality constraints:

$$\begin{cases} \text{minimize } f_0(x) \\ \text{subject to } g_i(x) \geq 0, \quad i = 1,....,m \end{cases}$$

- The problem can be transformed into an optimization problem with equality constraints as follows:

$$\begin{cases} \text{minimize } f_0(x) \\ \text{subject to } g_i(x) - y_i^2 = 0, \quad i = 1,....,m \end{cases}$$

- y_i^2 are non-negative surplus variables with unknown values.

4.22. The Lagrangian method with inequality constraints: Application (1)

Example 5:

- Consider an optimization problem with known „cost function“ $f(x_1, x_2)$ and known „inequality constraints“ .

$$\begin{cases} f(x_1, x_2) = 5x_1 + x_2 \\ 2x_1 + x_2 \geq 6 \\ x_1 + x_2 \geq 4 \\ 2x_1 + 10x_2 \geq 20 \\ x_1 \geq 0 \\ x_2 \geq 0 \end{cases}$$

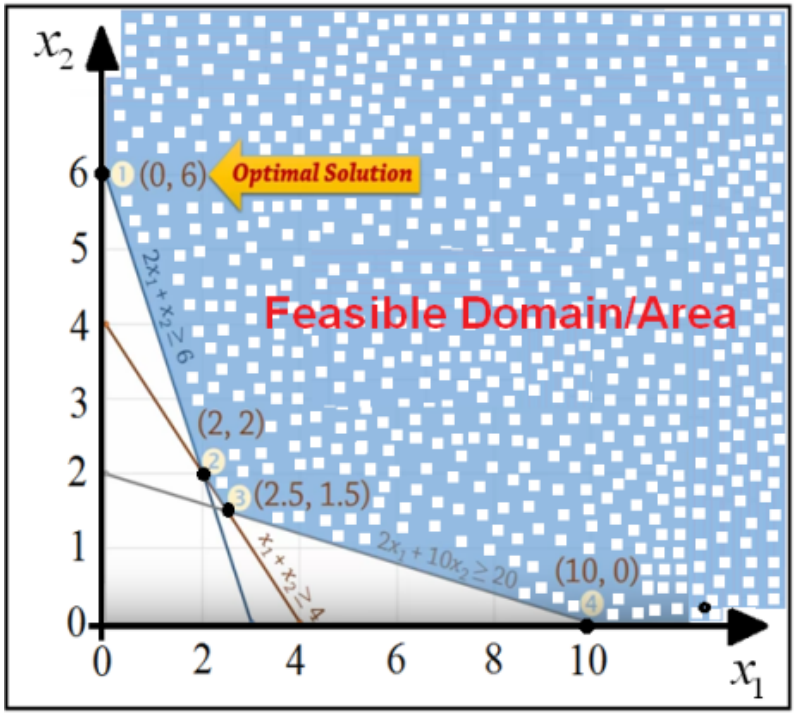
- Solve the LP problem graphically and determine the optimal objective function value.
- Transform the problem into an optimization problem with „equality constraints“.
- Calculte the surplus variables for each constraint.
- Identify Binding and Non-binding constraints.

Intelligent Transport Systems: New ICT – based Master’s Curricula in Uzbekistan (INTRAS)
Agreement number: 2017-3516/001-001
Project reference number: 586292-EPP-1-2017-1-PL-EPPKA2-CBHE-JP

4.22. The Lagrangian method with inequality constraints: Application (1)

Solution 5:

- **Step 1:** Plot of the feasible region and determination of the optimal solution.



(**graphical representation of the feasible region**)

Coordinates of points (x_1, x_2)	Objective/Cost function $f(x_1, x_2) = 5x_1 + x_2$
$(0, 6)$	6
$(2, 2)$	12
$(3/2, 5/2)$	10
$(10, 0)$	50

4.22. The Lagrangian method with inequality constraints: Application (1)

Solution 5:

- **Step 2:** Transformation of the original problem into „equality constraints“

$$\begin{cases} f(x_1, x_2) = 5x_1 + x_2 \\ 2x_1 + x_2 - y_1 = 6 \\ x_1 + x_2 - y_2 = 4 \\ 2x_1 + 10x_2 - y_3 = 20 \\ x_1 \geq 0, x_2 \geq 0 \\ y_1 \geq 0, y_2 \geq 0, y_3 \geq 0 \end{cases}$$

- **Step 3:** Calculation of the surplus variables y_i at the optimum point with coordinates $(0, 6)$. The results obtained are as follows:

$$y_1 = 0 ; y_2 = 2 ; y_3 = 40$$

- **Step 4:** The constraint 1 is Binding, while the constraints 2 and 3 are Non-binding.

4.23. The Lagrangian method with all types of constraints: Application (2)

Example 6:

•Consider an optimization problem with known „cost function“ $f(x_1, x_2)$ and known „inequality constraints“.

$$\begin{cases} Min[f(x_1, x_2) = 8x_1 + 7x_2] \\ 4x_1 + 2x_2 \geq 20 \\ -6x_1 + 4x_2 \leq 6 \\ x_1 + x_2 \geq 4 \\ 2x_1 - x_2 = 2 \\ x_1 \geq 0,\ x_2 \geq 0 \end{cases}$$

- Solve the LP problem graphically and determine the optimal objective function value.
- Transform the problem into an optimization problem with „equality constraints“.
- Calculte the slack/surplus variables for each constraint.
- Identify Binding and Non-binding constraints.

4.23. The Lagrangian method with all types of constraints: Application (2)

Solution 6:

- **Step 2:** Transformation of the original problem into „equality constraints“

$$\begin{cases} Min\left[f(x_1,x_2)=8x_1+7x_2\right] \\ 4x_1+2x_2-y_1=20 \\ -6x_1+4x_2+y_2=6 \\ x_1+x_2-y_3=4 \\ 2x_1-x_2=2 \\ x_1\geq 0,\ x_2\geq 0, \\ y_1\geq 0,\ y_2\geq 0,\ y_3\geq 0 \end{cases}$$

- **Step 3:** Calculation of the slack/surplus variables y_i at the optimum point with coordinates $(3,4)$. The results obtained are as follows:

$$y_1=0 \quad y_2=8 \quad y_3=3$$

- **Step 4:** The constraints 1 and 4 are Binding, while the constraints 2 and 3 are Non-binding.

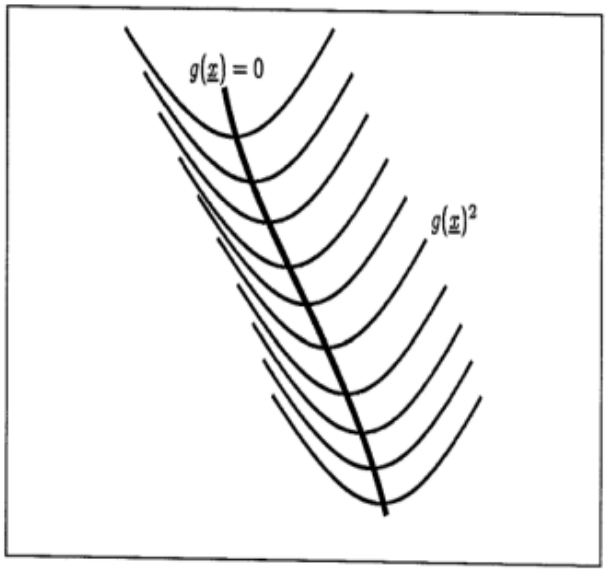
4.24. Classical methods of constrained optimization: „The penalty method“

- This method adds a quadratic energy term in order to penalize violations of constraints. Thus, the penalty method transforms the constrained minimization problem (E1) into an unconstrained minimization problem (E2). The generalization of the penalty method is expressed in (E3).

$$\begin{cases} \text{minimize } f_0(\underline{x}) \\ \text{subject to } g(\underline{x}) = 0, \quad i = 1,\ldots,m \end{cases} \qquad (E_1)$$

$$\text{minimize } \left[\varepsilon_{penalty} = f_0(\underline{x}) + c\left(g(\underline{x})\right)^2\right] \qquad (E_2)$$

$$\text{minimize } \left[\varepsilon_{penalty} = f_0(\underline{x}) + \sum_{i=1}^{n} c_i \left(g_i(\underline{x})\right)^2\right] \qquad (E_3)$$



(State space representation of the penalty method)

- Drawbacks of the penalty method:
  - ✓ Does not fulfil all constraints exactly.
  - ✓ The constraints strengths get harder to set for large dimensionality of $\underline{x}$.

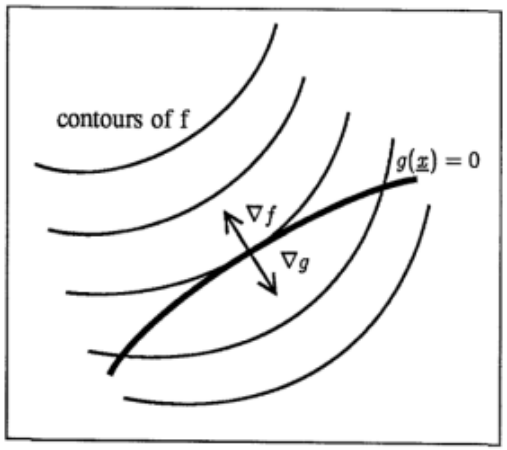
4.25. Classical methods of constrained optimization: „The Lagrange multiplier method“

- This method convert a constrained optimization problem (E1) into an unconstrained optimization problem (E2). A solution to (E1) is also a critical point to the energy in (E2). The parameter λ represents the Lagrange multiplier for the constraint $g(\underline{x}) = 0$. The expression (E2) reduces to the expression (E3) at constrained minima (see Fig. 8).

$$\begin{cases} \text{minimize } f_0(\underline{x}) \\ \text{subject to } g(\underline{x}) = 0, \quad i = 1,\ldots,m \end{cases} \qquad (E_1)$$

$$\text{minimize } \left[\varepsilon_{Lagrange} = f_0(\underline{x}) + \lambda g(\underline{x})\right] \qquad (E_2)$$

$$\nabla f_0(\underline{x}) = -\lambda \nabla g(\underline{x}) \qquad (E_3)$$



(State space representation of the Lagrange method)

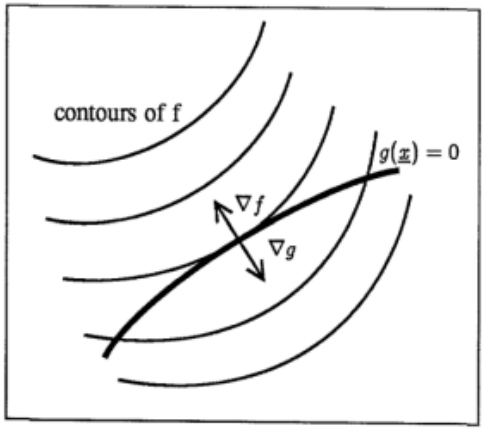
- Remarks:
  - ✓ The gradient of „*f*“ is collinear to the gradient of „g“ at constrained minima.
  - ✓ The collinearity of the gradients of „*f*“ *and* „*g*“ *is very important for the design of the* ***B****asic* ***D****ifferential* ***M****ultiplier* ***M****ethod* „***BDMM***“ *(See next slide).*

4.26. Classical methods of constrained optimization: The Basic Differential Multiplier Method (BDMM) for constrained optimization: „Gradient descent“

- The BDMM is an algorithm for constrained optimization which has not been previously used in neural networks.
- The neural algorithm based on „Gradient descent“ is expressed in the form of coupled ordinary differential equations (ODEs). These ODEs in (E2) are obtained by applying „Gradient descent“ to (E1).

$$\text{minimize}\left[\varepsilon_{Lagrange} = f_0(\underline{x}) + \lambda g(\underline{x})\right] \qquad (E_1)$$

$$\begin{cases} \dfrac{dx_i}{dt} = -\dfrac{\partial \varepsilon_{Lagrange}}{\partial x_i} = -\dfrac{\partial f_0(\underline{x})}{\partial x_i} - \lambda \dfrac{\partial g(\underline{x})}{\partial x_i} \\ \dfrac{d\lambda}{dt} = -\dfrac{\partial \varepsilon_{Lagrange}}{\partial \lambda} = -g(\underline{x}) \end{cases} \qquad (E_2)$$



(State space representation of the Lagrange method)

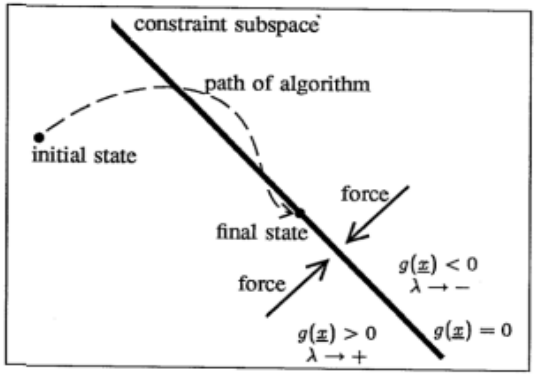
- Drawbacks:
  - ✓ The „gradient descent“ does not work with „Lagrange multiplier“.
  - ✓ The “gradient descent” converges only at “Local minimum”. No convergence is possible at “Global minimum”.

4.27. Classical methods of constrained optimization: The BDMM for constrained optimization: „Gradient descent“ + “„Gradient ascent“

- The BDMM can be obtained by applying „Gradient descent“ on decision variables and „Gradient ascent“ on multiplier variables. Applying the 2 gradients to (E1) leads to the set of coupled ODEs in (E2). This set represents the mathematical model of the BDMM.

$$\text{minimize}\left[\varepsilon_{Lagrange} = f_0(\underline{x}) + \lambda g(\underline{x})\right] \quad (E_1)$$

$$\begin{cases} \dfrac{dx_i}{dt} = -\dfrac{\partial \varepsilon_{Lagrange}}{\partial x_i} = -\dfrac{\partial f_0(\underline{x})}{\partial x_i} - \lambda \dfrac{\partial g(\underline{x})}{\partial x_i} \\ \dfrac{d\lambda}{dt} = +\dfrac{\partial \varepsilon_{Lagrange}}{\partial \lambda} = +g(\underline{x}) \end{cases} \quad (E_2)$$



(State space representation of the BDMM)

- Equation (E2) corresponds to a **Neural Network** with anti-symmetric connections between the multiplier variables/neurons and all the decision variables/neurons. (E2) is the model of **„Dynamic Neural Network“**
- Advantages:
  - ✓ The sign flip (i.e. „gradient ascent“ on multiplier variables and „gradient descent“ on decision variables) makes the BDMM algorithm very stable.

4.28. The Basic Differential Multiplier Method (BDMM)-Case of one constraint: Application (1)

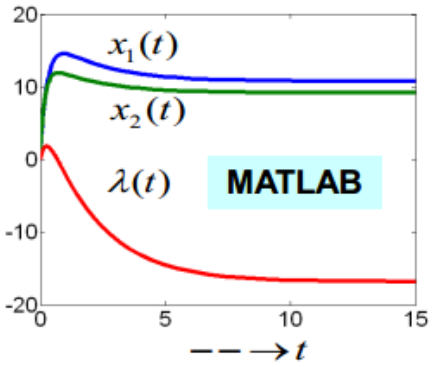
Example 7:

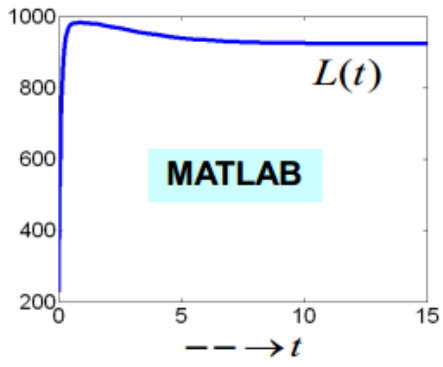
- Use the BDMM to find: $\begin{cases} Max\left[f(x_1,x_2) = -2x_1^2 + 60x_1 - 3x_2^2 + 72x_2 + 100\right] \\ subject\ to\ \left[x_1 + x_2 = 20\right] \end{cases}$

Solution 7:

- The Lagrange function is: $L(x_1, x_2, \lambda) = \left(-2x_1^2 + 60x_1 - 3x_2^2 + 72x_2 + 100\right) + \lambda(x_1 + x_2 - 20) \quad (E_1)$
- Applying BDMM to $-L(x_1, x_2, \lambda)$ leads to (E_2):

$$\begin{cases} \dfrac{dx_1}{dt} = +(-4x_1 + 60 + \lambda) \\ \dfrac{dx_2}{dt} = +(-6x_2 + 72 + \lambda) \\ \dfrac{d\lambda}{dt} = -(x_1 + x_2 - 20) \end{cases} \quad (E_2)$$





- Equation (E_2) is solved using MATLAB and the solution converges to:

$$x_1(t) = 10.8 \quad ; \quad x_2(t) = 9.2 \quad ; \quad \lambda(t) = -16.78 \quad ; \quad L_{max}(t) = 923.17$$

Intelligent Transport Systems: New ICT – based Master's Curricula in Uzbekistan (INTRAS)
Agreement number: 2017-3516/001-001
Project reference number: 586292-EPP-1-2017-1-PL-EPPKA2-CBHE-JP

4.29. The Basic Differential Multiplier Method (BDMM)-Case of two constraints: Application (2)

Example 8:

- Use the BDMM to find: $\begin{cases} Min\left[f(x_1,x_2,x_3)=x_1^2+x_2^2+x_3^2\right] \\ subject\ to \begin{cases}\left[x_1+x_2+x_3=1\right] \\ \left[x_1+2x_2+3x_3=6\right]\end{cases}\end{cases}$

Solution 8:

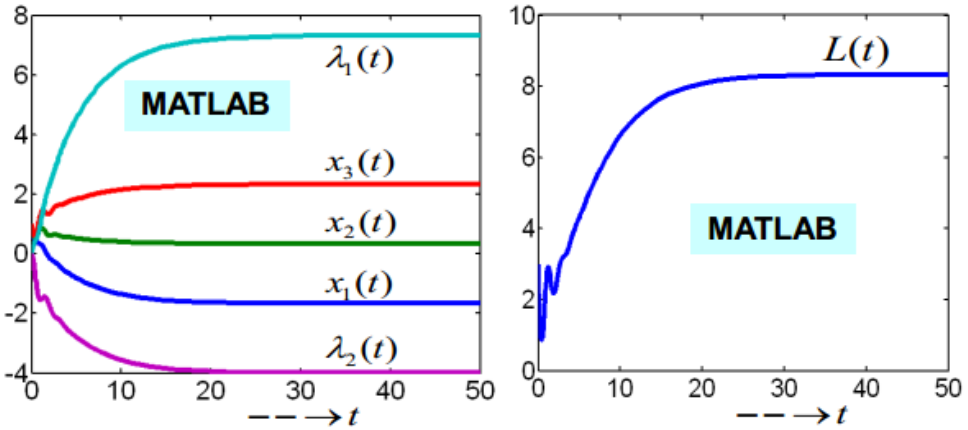
- Lagrange function is:

$$L(x_1,x_2,x_3,\lambda_1,\lambda_2)=\left(x_1^2+x_2^2+x_3^2\right)+\lambda_1(x_1+x_2+x_3-1)+\lambda_2(x_1+2x_2+3x_3-6)\ (E_1)$$

- Applying BDMM to (E_1) leads to (E_2). Finally solving (E_2) leads to:

$$x_1(t)=-1.6666\ ;\quad x_2(t)=0.3333\quad ;\quad x_3(t)=2.3333$$
$$\lambda_1(t)=7.3331\quad ;\quad \lambda_2(t)=-3.9999\ ;\ L_{max}(t)=8.3328$$

$$(E_2)\begin{cases}\frac{dx_1}{dt}=-(2x_1+\lambda_1+\lambda_2)\\ \frac{dx_2}{dt}=-(2x_2+\lambda_1+2\lambda_2)\\ \frac{dx_3}{dt}=-(2x_3+\lambda_1+3\lambda_2)\\ \frac{d\lambda_1}{dt}=+(x_1+x_2+x_3-1)\\ \frac{d\lambda_2}{dt}=+(x_1+2x_2+3x_3-6)\end{cases}$$



4.30. Simulation algorithm/tool: The MATLAB TOOLBOX for Linear Programming (LP)

❑ Definition of Linear Programming (LP)

✓ Linear Programming (LP) is the problem of finding a vector x that minimizes a linear function $f^T x$ subject to linear constraints:

$$\min_x f^T x \quad \text{subject to} \quad \begin{matrix} A * x \leqslant b \\ A_{eq} * x = b_{eq} \\ lb \leqslant x \leqslant ub \end{matrix}$$

Vectors are:

$f, x, b, beq, lb,$ and ub

Matrices are:

A, and Aeq

❑ Syntax of Linear Programming (LP)

```
x = linprog(f,A,b)
x = linprog(f,A,b,Aeq,beq)
x = linprog(f,A,b,Aeq,beq,lb,ub)
x = linprog(f,A,b,Aeq,beq,lb,ub,x0)
x = linprog(f,A,b,Aeq,beq,lb,ub,x0,options)
x = linprog(problem)
[x,fval] = linprog(...)
[x,fval,exitflag] = linprog(...)
[x,fval,exitflag,output] = linprog(...)
[x,fval,exitflag,output,lambda] = linprog(...)
```

❖ Additional information can be obtained by typing the following command in the MATLAB- Prompt:

>> help linprog

4.30. Simulation algorithm/tool: The MATLAB TOOLBOX for Linear Programming (LP)

❑ Description of Linear Programming (LP)

linprog solves linear programming problems.
x = linprog(f,A,b) solves min f'*x such that A*x ≤ b.
x = linprog(f,A,b,Aeq,beq) solves the problem above while additionally satisfying the equality constraints Aeq*x = beq. Set A = [ ] and b = [ ] if no inequalities exist.
x = linprog(f,A,b,Aeq,beq,lb,ub) defines a set of lower and upper bounds on the design variables, x, so that the solution is always in the range lb ≤ x ≤ ub. Set Aeq = [ ] and beq = [ ] if no equalities exist.
x = linprog(f,A,b,Aeq,beq,lb,ub,x0) sets the starting point to x0. linprog uses x0 only with the active-set algorithm. linprog ignores x0 with the interior-point and simplex algorithms.
x = linprog(f,A,b,Aeq,beq,lb,ub,x0,options) minimizes with the optimization options specified in options. Use optimoptions to set these options.
x = linprog(problem) finds the minimum for problem, where problem is a structure described in Input Arguments.
Create the problem structure by exporting a problem from Optimization app, as described in Exporting Your Work.
[x,fval] = linprog(...) returns the value of the objective function fun at the solution x: fval = f'*x.
[x,fval,exitflag] = linprog(...) returns a value exitflag that describes the exit condition.
[x,fval,exitflag,output] = linprog(...) returns a structure output that contains information about the optimization.
[x,fval,exitflag,output,lambda] = linprog(...) returns a structure lambda whose fields contain the Lagrange multipliers at the solution x.

Note: If the specified input bounds for a problem are inconsistent, the output x is x0 and the output fval is [ ].

4.31. Solving an optimization problem using the Linprog Toolbox: Application example (1)

Application 1 (See Example 5 above):

- Use the Linprog toolbox to solve the „Example 5 provided above “.

$$\begin{cases} f(x_1,x_2)=5x_1+x_2 \\ 2x_1+x_2\geq 6 \\ x_1+x_2\geq 4 \\ 2x_1+10x_2\geq 20 \\ x_1\geq 0 \\ x_2\geq 0 \end{cases} \quad (E_1) \quad \Leftrightarrow \quad \begin{cases} f(x_1,x_2)=5x_1+x_2 \\ 2x_1+x_2-y_1=6 \\ x_1+x_2-y_2=4 \\ 2x_1+10x_2-y_3=20 \\ x_1\geq 0,\ x_2\geq 0 \\ y_1\geq 0,\ y_2\geq 0,\ y_3\geq 0 \end{cases} \quad \begin{matrix} y_1=0 \\ y_2=2 \\ y_3=40 \end{matrix} \quad (E_2)$$

Linprog for system (E_1)

```
%%Linprog with inequality Constraints
% Inequality constraints;
A=[-2 -1; -1 -1; -2 -10]; b=[-6,-4,-20];
% Equality constraints;
 Aeq = [ ];     beq = [ ];
%% Side constraints
lb = [0 ; 0]; ub = [ ];
% Objective function;
f = [5 1 ];
% Syntax for optimization
[x,fval]=linprog(f,A,b,Aeq,beq,lb,ub)%
```

Linprog for system (E_2)

```
%% Linprog with equality Constraints
% Inequality constraints;
A = [ ]; b = [ ];
% Equality constraints;
Aeq=[2 1; 1 1; 2 10]; beq=[6;6;60];
%% Side constraints
lb = [0 ; 0]; ub = [ ];
% Objective function;
f = [5 1 ];
% Syntax for optimization
[x,fval]=linprog(f,A,b,Aeq,beq,lb,ub)
```

4.32. Solving an optimization problem using the Linprog Toolbox: Application example (2)

Application 2 (See Example 6 above):

- Use the Linprog toolbox to solve the „Example 6 provided above “.

$$\begin{cases} Min[f(x_1,x_2)=8x_1+7x_2] \\ 4x_1+2x_2 \geq 20 \\ -6x_1+4x_2 \leq 6 \\ x_1+x_2 \geq 4 \\ 2x_1-x_2=2 \\ x_1 \geq 0, x_2 \geq 0 \end{cases} \quad (E_1) \quad \Leftrightarrow \quad \begin{cases} Min\,[f(x_1,x_2)=8x_1+7x_2] \\ 4x_1+2x_2-y_1=20 \\ -6x_1+4x_2+y_2=6 \\ x_1+x_2-y_3=4 \\ 2x_1-x_2=2 \\ x_1 \geq 0, x_2 \geq 0, \\ y_1 \geq 0, y_2 \geq 0, y_3 \geq 0 \end{cases} \quad \begin{matrix} y_1=0 \\ y_2=8 \\ y_3=3 \end{matrix} \quad (E_2)$$

Linprog for system (E_1)

```
%%Linear Program with all Types of Constraints
% Inequality constraints;
A=[-4 -2;-6 4;-1 -1]; b=[-20,6,-4];
% Equality constraints;
 Aeq = [2 -1 ];     beq = [2];
%% Side constraints
lb = [0 ; 0]; ub = [ ];
% Objective function;
f = [8 7];
% Syntax for optimization
[x,fval] = linprog(f,A,b,Aeq,beq,lb,ub)
```

Linprog for system (E_2)

```
%%Linear Program with equality Constraints
% Inequality constraints;
A =[ ]; b=[ ];
% Equality constraints;
Aeq=[4 2 ;-6 +4;1 1; 2 -1];beq=[20,-2,7 2];
%% Side constraints
lb = [0 ; 0]; ub = [ ];
% Objective function;
f = [8 7];
% Syntax for optimization
[x,fval]=linprog(f,A,b,Aeq,beq,lb,ub)
```

4.33. The MATLAB TOOLBOX for quadratic Programming (QP)

- Quadratic programming **(Quadprog)** is a MATLAB TOOLBOX for solving nonlinear programming problems. The QP problem is defined by objective function(s) with nonlinearity of degree 2. The constraints formulated for this type of problem are linear.
- **„The objective of optimization"** consists of **finding the minimum** of a problem modelled mathematically as follows:

$$\begin{cases} \underset{x}{Min}\left[\frac{1}{2}x^T Hx + f^T x\right] \\ subject\ to \begin{cases} A \cdot x \le b \\ Aeq \cdot x = beq \\ lb \le x \le ub \end{cases} \end{cases}$$

Vectors are:

$f, x, b, beq, lb,$ and ub

Matrices are:

$H, A,$ and Aeq

- Additional information can be obtained by typing the following command in the MATLAB- Prompt:

>> help quadprog

- **The syntax used by the "MATLAB TOOLBOX" to evaluate the optimization problem is:**

```
x = quadprog(H,f)
x = quadprog(H,f,A,b)
x = quadprog(H,f,A,b,Aeq,beq)
x = quadprog(H,f,A,b,Aeq,beq,lb,ub)
x = quadprog(H,f,A,b,Aeq,beq,lb,ub,x0)
x = quadprog(H,f,A,b,Aeq,beq,lb,ub,x0,options)
x = quadprog(problem)
[x,fval] = quadprog(H,f,...)
[x,fval,exitflag] = quadprog(H,f,...)
[x,fval,exitflag,output] = quadprog(H,f,...)
[x,fval,exitflag,output,lambda] = quadprog(H,f,...)
```

„H" is a matrix obtained from the non-linear part of the objective function.

$$H = \begin{bmatrix} \frac{\partial^2 f(x_1,x_2)}{\partial x_1^2} & \frac{\partial^2 f(x_1,x_2)}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f(x_1,x_2)}{\partial x_1 \partial x_2} & \frac{\partial^2 f(x_1,x_2)}{\partial x_2^2} \end{bmatrix}$$

„H" must be positive definite. (See condition in Thema 3)

4.34. Solving an optimization problem using the Quadprog Toolbox: Application example (3)

Application 3 (See Example 3 above):

- Use the Quadprog toolbox to solve the „Example 3 provided above “.

$$\begin{cases} Min\left[f(x_1,x_2)=3x_1^2+6x_2^2-x_1x_2\right] \\ \text{subject to } g(x_1,x_2)=x_1+x_2-20 \end{cases} \qquad (E)$$

„Quadprog“ algorithm for system (E)

```
%% Quadratic Program with one equality Constraint
% Inequality constraints;
A = [ ];   b = [ ];
% Equality constraints;
 Aeq = [1   1]; beq = [20];
%% Side constraints
 lb = [ ]; ub = [ ];
%% Matrix "H"
H= [6   -1 ; -1  12 ];
% Objective function;
f = [0 ; 0];
% Syntax for optimization
[x,fval] = quadprog(H, f, A, b, Aeq, beq, lb, ub)
```

Solution 3:

$$\begin{cases} x_1^* = 13 \\ x_2^* = 7 \\ f(x_1,x_2)\big|_{optimum} = f(x_1^*,x_2^*) = 710 \end{cases}$$

4.35. Solving an optimization problem using the Quadprog Toolbox: Application example (4)

Application 4 (See Example 4 above):

$$\begin{cases} Min\left[f(x,y,z)=x^2+y^2+z^2-2xy-2yz\right] \\ \text{Subject to} \begin{cases} g(x,y,z)=x-y=1 \\ h(x,y,z)=y=1 \end{cases} \end{cases} \qquad (E)$$

„Quadprog“ algorithm for system (E)

- Use the Quadprog toolbox to solve the „Example 4 provided above “.

```
%% Quadratic Program with two equality Constraints
% Inequality constraints;
A = [ ];   b = [ ];
% Equality constraints;
Aeq=[1 -1 0;0 1 0];beq=[1 1];
%% Side constraints
 lb = [ ]; ub = [ ];
%% Matrix "H"
H= [2 -2 0 ;-2 2 -2;0 -2 2];
% Objective function;
f = [0 ; 0 ; 0];
% Syntax for optimization
[x,fval] = quadprog(H, f, A, b, Aeq, beq, lb, ub)
```

Solution 4:

$$\begin{cases} x=2 \\ y=1 \\ z=1 \\ f(x,y,z)_{opt}=0 \end{cases}$$

4.36. Solving an optimization problem using the Quadprog Toolbox: Application example (5)

Application 5 (See Example 7 above):

- Use the Quadprog toolbox to solve the "Example 7 provided above".

$$\begin{cases} Max\left[f(x_1,x_2) = -2x_1^2 + 60x_1 - 3x_2^2 + 72x_2 + 100\right] \\ subject\ to\ \left[x_1 + x_2 = 20\right] \end{cases} \qquad (E)$$

„Quadprog“ algorithm for system (E)

```
%% Quadratic Program with one equality Constraint
% Inequality constraints;
A = [ ];   b = [ ];
% Equality constraints;
 Aeq = [1   1];     beq = [20];
%% Side constraints
 lb = [ ]; ub = [ ];
%% Matrix "H"
H= [4   0 ; 0   6];
% Objective function;
f = [-60 ; -72];
% Syntax for optimization
[x,fval] = quadprog(H, f, A, b, Aeq, beq, lb, ub)
```

Solution 5:

$x_1(t) = 10.8$

$x_2(t) = 9.2$

$L_{\min}(t) = -923.20$

4.37. Solving an optimization problem using the Quadprog Toolbox: Application example (6)

Application 6 (See Example 8 above):

- Use the Quadprog toolbox to solve the "Example 7 provided above".

$$\begin{cases} Min\left[f(x_1,x_2)=x_1^2+x_2^2+x_3^2\right] \\ subject\ to \begin{cases} \left[x_1+x_2+x_3=1\right] \\ \left[x_1+2x_2+3x_3=6\right] \end{cases} \end{cases} \qquad (E)$$

„Quadprog“ algorithm for system (E)

```
%% Quadratic Program with two equality Constraints
% Inequality constraints;
A = [ ];  b = [ ];
% Equality constraints;
 Aeq = [1  1 1 ; 1 2 3];  beq = [1  6];
%% Side constraints
 lb = [ ]; ub = [ ];
%% Matrix "H"
H= [1 0 0 ; 0 1 0  ; 0 0 1];
% Objective function;
f = [0 0 0];
% Syntax for optimization
[x,fval] = quadprog(H, f, A, b, Aeq, beq, lb, ub)
```

Solution 6:

$x_1(t) = -1.6667$

$x_2(t) = 0.3333$

$x_3(t) = 2.3333$

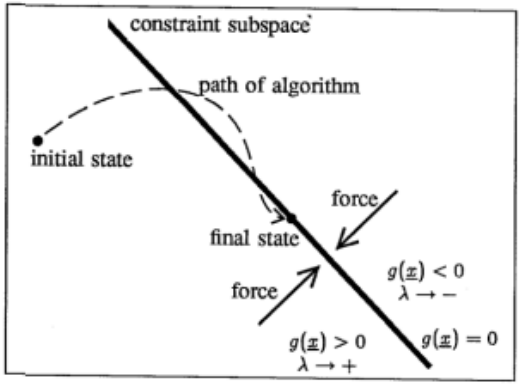
$L_{\min}(t) = 8.3333$

4.38. MATLAB Coding of the BDMM for constrained optimization: „Gradient descent & Gradient ascent "

- The BDMM can be obtained by applying „Gradient descent" on decision variables and „Gradient ascent" on multiplier variables. Applying the 2 gradients to (E1) leads to the set of coupled ODEs in (E2). This set represents the mathematical model of the BDMM.

$$\text{minimize}\left[\varepsilon_{Lagrange} = f_0(\underline{x}) + \lambda g(\underline{x})\right] \quad (E_1)$$

$$\begin{cases} \dfrac{dx_i}{dt} = -\dfrac{\partial \varepsilon_{Lagrange}}{\partial x_i} = -\dfrac{\partial f_0(\underline{x})}{\partial x_i} - \lambda \dfrac{\partial g(\underline{x})}{\partial x_i} \\ \dfrac{d\lambda}{dt} = +\dfrac{\partial \varepsilon_{Lagrange}}{\partial \lambda} = +g(\underline{x}) \end{cases} \quad (E_2)$$



(State space representation of the BDMM)

- Equation (E2) corresponds to a **Neural Network** with anti-symmetric connections between the multiplier variables/neurons and all the decision variables/neurons. (E2) is the model of **„Dynamic Neural Network"**
- Advantages:
  - ✓ The sign flip (i.e. „gradient ascent" on multiplier variables and „gradient descent" on decision variables) makes the BDMM algorithm very stable.

4.39. MATLAB coding of the Basic Differential Multiplier Method: Application example (7)

Application 7:

- Use the BDMM to find: $\begin{cases} Max\left[f(x_1,x_2) = -2x_1^2 + 60x_1 - 3x_2^2 + 72x_2 + 100\right] \\ subject\ to\ \left[x_1 + x_2 = 20\right] \end{cases}$

Solution 7: (This example is proposed above)

- The Lagrange function is: $L(x_1, x_2, \lambda) = \left(-2x_1^2 + 60x_1 - 3x_2^2 + 72x_2 + 100\right) + \lambda(x_1 + x_2 - 20)$ (E_1)
- Applying BDMM to $-L(x_1, x_2, \lambda)$ leads to (E_2):

$$\begin{cases} \dfrac{dx_1}{dt} = +(-4x_1 + 60 + \lambda) \\ \dfrac{dx_2}{dt} = +(-6x_2 + 72 + \lambda) \\ \dfrac{d\lambda}{dt} = -(x_1 + x_2 - 20) \end{cases} \qquad (E_2)$$

4.39. MATLAB coding of the Basic Differential Multiplier Method: Application example (7)

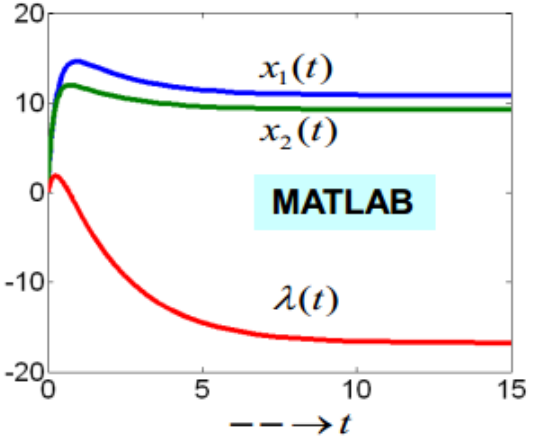
Solution 7:

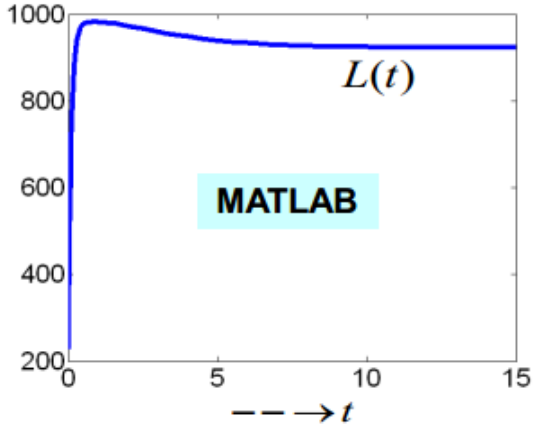
$$\begin{cases} \dfrac{dx_1}{dt} = +(-4x_1 + 60 + \lambda) \\ \dfrac{dx_2}{dt} = +(-6x_2 + 72 + \lambda) \\ \dfrac{d\lambda}{dt} = -(x_1 + x_2 - 20) \end{cases} \qquad (E_1)$$

MATLAB Code for (E_1)

```
function dx=f(t,x)
dx=zeros(3,1);
dx(1)=+( -4*x(1) + 60 + x(3) );
dx(2)=+( -6*x(2) + 72 + x(3) );
dx(3)=-(  x(1) + x(2) -  20 );
end
```

```
>> [t,x]=ode45(@BDMM7, [0 15], [1 0 0])

>> L=(-2*x(:,1).^2+60*x(:,1)-3*x(:,2).^2 +
   72*x(:,2)+100)+x(:,3).*(x(:,1)+x(:,2)-20)
```





Solution 7:

$x_1(t) = 10.8$

$x_2(t) = 9.2$

$\lambda(t) = -16.78$

$L_{\max}(t) = 923.17$

Intelligent Transport Systems: New ICT – based Master's Curricula in Uzbekistan (INTRAS)
Agreement number: 2017-3516/001-001
Project reference number: 586292-EPP-1-2017-1-PL-EPPKA2-CBHE-JP

4.40. MATLAB coding of the Basic Differential Multiplier Method: Application example (8)

Example 8:

- Use the BDMM to find:

$$\begin{cases} Min\left[f(x_1,x_2,x_3)=x_1^2+x_2^2+x_3^2\right] \\ subject\ to \begin{cases} [x_1+x_2+x_3=1] \\ [x_1+2x_2+3x_3=6] \end{cases} \end{cases}$$

Solution 8: (This example is proposed above)

- Lagrange function is:

$$L(x_1,x_2,x_3,\lambda_1,\lambda_2)=\left(x_1^2+x_2^2+x_3^2\right)+\lambda_1(x_1+x_2+x_3-1)+\lambda_2(x_1+2x_2+3x_3-6) \quad (E_1)$$

- Applying BDMM to (E_1) leads to (E_2). Finally solving (E_2) leads to:

$$\begin{cases} \frac{dx_1}{dt}=-(2x_1+\lambda_1+\lambda_2) \\ \frac{dx_2}{dt}=-(2x_2+\lambda_1+2\lambda_2) \\ \frac{dx_3}{dt}=-(2x_3+\lambda_1+3\lambda_2) \\ \frac{d\lambda_1}{dt}=+(x_1+x_2+x_3-1) \\ \frac{d\lambda_2}{dt}=+(x_1+2x_2+3x_3-6) \end{cases} \qquad (E_2)$$

4.40. MATLAB coding of the Basic Differential Multiplier Method: Application example (8)

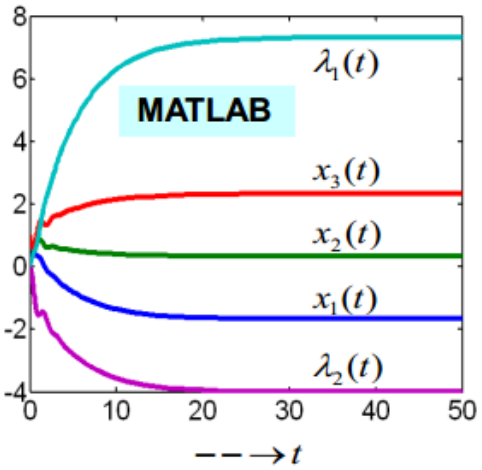
Solution 8:

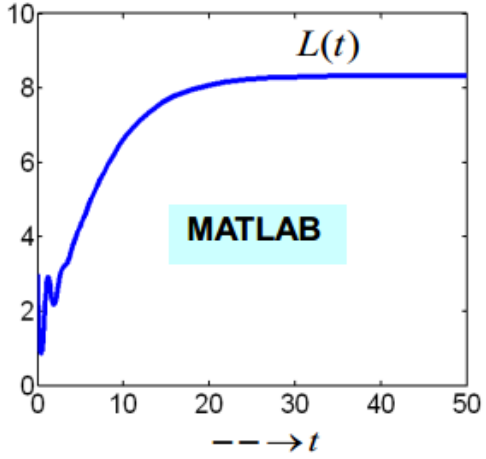
$$\begin{cases} \dfrac{dx_1}{dt} = -(2x_1 + x_4 + x_5) \\ \dfrac{dx_2}{dt} = -(2x_2 + x_4 + 2x_5) \\ \dfrac{dx_3}{dt} = -(2x_3 + x_4 + 3x_5) \\ \dfrac{d\lambda_1}{dt} = +(x_1 + x_2 + x_3 - 1) \\ \dfrac{d\lambda_2}{dt} = +(x_1 + 2x_2 + 3x_3 - 6) \end{cases} \quad (E_1)$$

MATLAB Code for (E_1)

```
function dx=f(t,x)
dx=zeros(5,1);
dx(1)=-( 2*x(1)+x(4)+x(5) );
dx(2)=-( 2*x(2)+x(4)+2*x(5) );
dx(3)=-( 2*x(3)+x(4)+3*x(5) );
dx(4)=+( x(1)+x(2)+x(3)-1 );
dx(5)=+(x(1)+2*x(2)+3*x(3)-6);
end
```

```
>> [t,x]=ode45(@BDMM7, [0 50], [1 0 0 0 0])
>> L=(x(:,1).^2+x(:,2).^2+x(:,3).^2)+x(:,4).*(x(:,1)+
   x(:,2)+x(:,3)-1)+x(:,5).*(x(:,1)+2*x(:,2)+3*x(:,3)-6)
```





Solution 8:

$x_1(t) = -1.6666$

$x_2(t) = 0.3333$

$x_3(t) = 2.3333$

$\lambda_1(t) = 7.3331$

$\lambda_2(t) = -3.9999$

$L_{max}(t) = 8.3328$

INTRAS

Co-funded by the
Erasmus+ Programme
of the European Union

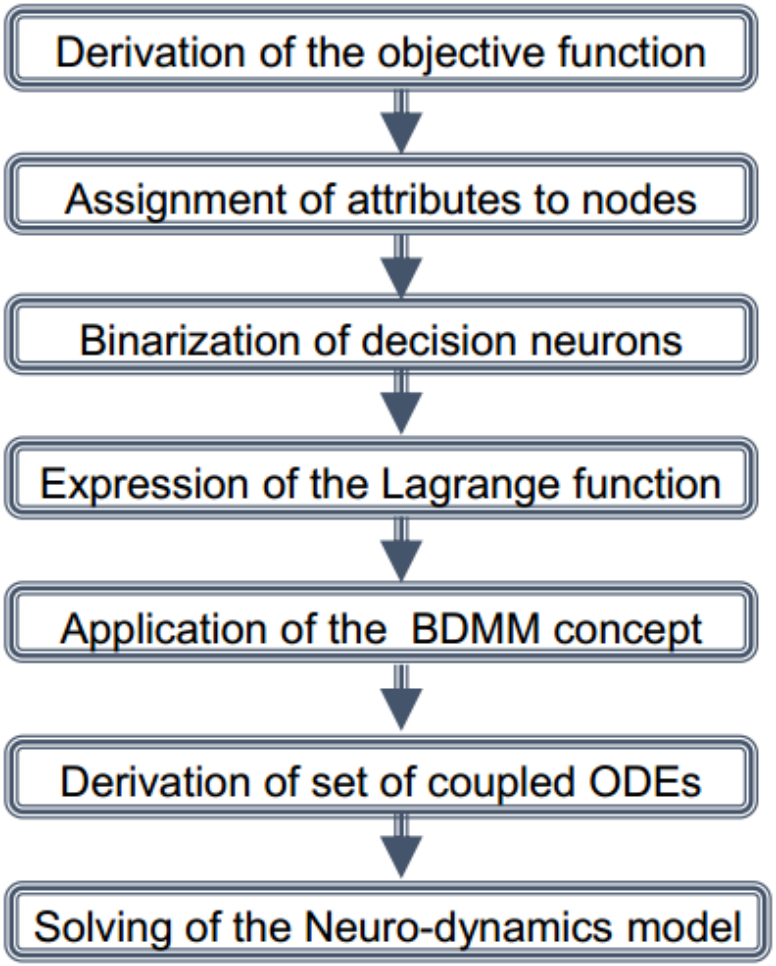
Chapter 5.

Optimization in graph networks with applications in transportation

(Full chapter presentation)

Intelligent Transport Systems: New ICT – based Master's Curricula in Uzbekistan (INTRAS)
Agreement number: 2017-3516/001-001
Project reference number: 586292-EPP-1-2017-1-PL-EPPKA2-CBHE-JP

5. 1. Transforming Graph Theoretical problems into set of ODEs: The Neurocomputing concept (BDMM)

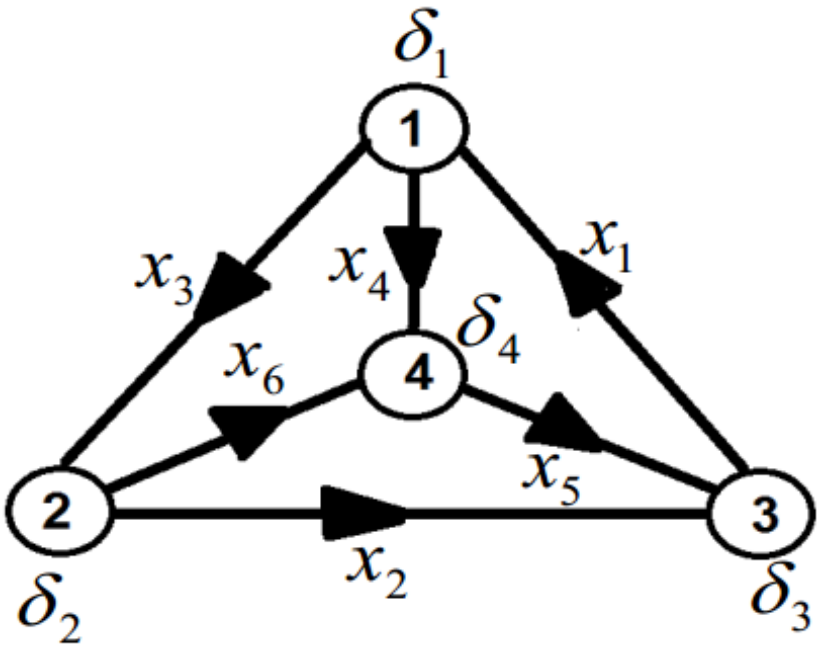


Intelligent Transport Systems: New ICT – based Master's Curricula in Uzbekistan (INTRAS)
Agreement number: 2017-3516/001-001
Project reference number: 586292-EPP-1-2017-1-PL-EPPKA2-CBHE-JP

5. 2. Mathematical modeling of the shortest path problem (SPP) in a directed graph of magnitude 4 and size 6

- **Case study 1: A simple directed graph of magnitude 4 and size 6 (see below)**

- **TASK: Mathematical modelling of the SPP using the theory of optimization in chap. 4**
  - ✓ In this section we use the knowledge acquired in chapter 4 to model the SPP (in the graph below) mathematically.



Intelligent Transport Systems: New ICT – based Master's Curricula in Uzbekistan (INTRAS)
Agreement number: 2017-3516/001-001
Project reference number: 586292-EPP-1-2017-1-PL-EPPKA2-CBHE-JP

- **STEP 1: Expression of the objective function.**
  - ✓ This function is expressed as the minimization of the total cost of the graph.

$$Min\left[f(x_i,c_i)=\left(c_1x_1+c_2x_2+c_3x_3+c_4x_4+c_5x_5+c_6x_6\right)\right]$$

- **STEP 2: Formulation of the optimization constraints: "Connectivity of nodes"**
  - ✓ Constraints are obtained through the assignment of attributes to all nodes of the graph. Three possible attributes are defined in the case of SPP: The source Node, The intermediate node, and the destination node. These attributes are modeled by the set of equations below:

$$g(x_i,\delta_i)=\begin{cases}\text{Node1}: \left(x_3+x_4-x_1\right)=\delta_1\\ \text{Node2}: \left(x_2+x_6-x_3\right)=\delta_2\\ \text{Node3}: \left(x_1-x_5-x_2\right)=\delta_3\\ \text{Node4}: \left(x_5-x_6-x_4\right)=\delta_4\end{cases}$$

$$(\text{Source Node}: \delta_i=1)$$
$$(\text{Intermediate Node}: \delta_i=0)$$
$$(\text{Destination Node}: \delta_i=-1)$$

- **STEP 3: Formulation of the optimization constraint: "Binarization"**
  - ✓ This constraint insures, that each edge of the graph takes only two possible values, which are "0" or "1" (Binarization):

$$h(x_i)=\begin{cases} x_1(x_1-1)=0 \\ x_2(x_2-1)=0 \\ x_3(x_3-1)=0 \\ x_4(x_4-1)=0 \\ x_5(x_5-1)=0 \\ x_6(x_6-1)=0 \end{cases}$$

- **STEP 4: Expression of the Lagrange function as the total energy of the system**
  - ✓ The Lagrange function is obtained by combining the objective function with constraints. This combination is achieved using the multiplier variables λ_i.

$$L=(c_1x_1+c_2x_2+c_3x_3+c_4x_4+c_5x_5+c_6x_6)+\lambda_1(x_3+x_4-x_1-\delta_1)$$
$$+\lambda_2(x_2+x_6-x_3-\delta_2)+\lambda_3(x_1-x_5-x_2-\delta_3)+\lambda_4(x_5-x_6-x_4-\delta_4)$$
$$+\lambda_5x_1(x_1-1)+\lambda_6x_2(x_2-1)+\lambda_7x_3(x_3-1)+\lambda_8x_4(x_4-1)+\lambda_9x_5(x_5-1)+\lambda_{10}x_6(x_6-1)$$

- **STEP 5: Application of the BDMM concept and derivation of the coupled ODEs**
  - ✓ The gradient descent (see chapter 4) is applied to decision neurons/variables to obtain the set of coupled ODEs below:

$$\begin{cases} \dfrac{dx_1}{dt} = -\dfrac{\partial L}{\partial x_1} = -\left[c_1 - \lambda_1 + \lambda_3 + \lambda_5(2x_1 - 1)\right] \\ \dfrac{dx_2}{dt} = -\dfrac{\partial L}{\partial x_2} = -\left[c_2 + \lambda_2 - \lambda_3 + \lambda_6(2x_2 - 1)\right] \\ \dfrac{dx_3}{dt} = -\dfrac{\partial L}{\partial x_3} = -\left[c_3 + \lambda_1 - \lambda_2 + \lambda_7(2x_3 - 1)\right] \\ \dfrac{dx_4}{dt} = -\dfrac{\partial L}{\partial x_4} = -\left[c_4 + \lambda_1 - \lambda_4 + \lambda_8(2x_4 - 1)\right] \\ \dfrac{dx_5}{dt} = -\dfrac{\partial L}{\partial x_5} = -\left[c_5 - \lambda_3 + \lambda_4 + \lambda_9(2x_5 - 1)\right] \\ \dfrac{dx_6}{dt} = -\dfrac{\partial L}{\partial x_6} = -\left[c_6 + \lambda_2 - \lambda_4 + \lambda_{10}(2x_6 - 1)\right] \end{cases}$$

Intelligent Transport Systems: New ICT – based Master's Curricula in Uzbekistan (INTRAS)
Agreement number: 2017-3516/001-001
Project reference number: 586292-EPP-1-2017-1-PL-EPPKA2-CBHE-JP

- **STEP 6: Application of the BDMM concept and derivation of the coupled ODEs**
  - ✓ The gradient ascent (see chapter 4) is applied to the **GROUP 1** of multiplier neurons/variables (ensuring the connectivity of nodes) to obtain the set of coupled ODEs below:

$$\begin{cases} \dfrac{d\lambda_1}{dt} = +\dfrac{\partial L}{\partial \lambda_1} = +\left[x_3 + x_4 - x_1 - \delta_1\right] \\ \dfrac{d\lambda_2}{dt} = +\dfrac{\partial L}{\partial \lambda_2} = +\left[x_2 + x_6 - x_3 - \delta_2\right] \\ \dfrac{d\lambda_3}{dt} = +\dfrac{\partial L}{\partial \lambda_3} = +\left[x_1 - x_5 - x_2 - \delta_3\right] \\ \dfrac{d\lambda_4}{dt} = +\dfrac{\partial L}{\partial \lambda_4} = +\left[x_5 - x_6 - x_4 - \delta_4\right] \end{cases}$$

- **STEP 7: Application of the BDMM concept and derivation of the coupled ODEs**
  - ✓ The gradient ascent (see chapter 4) is applied to the **GROUP 2** of multiplier neurons (ensuring the binarization) to obtain the set of coupled ODEs below:

$$\begin{cases} \dfrac{d\lambda_5}{dt} = +\dfrac{\partial L}{\partial \lambda_5} = +\left[x_1(x_1-1)\right] \\ \dfrac{d\lambda_6}{dt} = +\dfrac{\partial L}{\partial \lambda_6} = +\left[x_2(x_2-1)\right] \\ \dfrac{d\lambda_7}{dt} = +\dfrac{\partial L}{\partial \lambda_7} = +\left[x_3(x_3-1)\right] \\ \dfrac{d\lambda_8}{dt} = +\dfrac{\partial L}{\partial \lambda_8} = +\left[x_4(x_4-1)\right] \\ \dfrac{d\lambda_9}{dt} = +\dfrac{\partial L}{\partial \lambda_9} = +\left[x_5(x_5-1)\right] \\ \dfrac{d\lambda_{10}}{dt} = +\dfrac{\partial L}{\partial \lambda_{10}} = +\left[x_6(x_6-1)\right] \end{cases}$$

Intelligent Transport Systems: New ICT – based Master's Curricula in Uzbekistan (INTRAS)
Agreement number: 2017-3516/001-001
Project reference number: 586292-EPP-1-2017-1-PL-EPPKA2-CBHE-JP

❑ STEP 8: Mathematical model of the Neuro-processor for the SPP

✓ The Mathematical model of the Neuro-processor is obtained as a combination of the set of equations obtained in STEPS 5, 6 and 7.

$$\begin{cases}
\frac{dx_1}{dt} = -[c_1 - \lambda_1 + \lambda_3 + \lambda_5(2x_1 - 1)] \\
\frac{dx_2}{dt} = -[c_2 + \lambda_2 - \lambda_3 + \lambda_6(2x_2 - 1)] \\
\frac{dx_3}{dt} = -[c_3 + \lambda_1 - \lambda_2 + \lambda_7(2x_3 - 1)] \\
\frac{dx_4}{dt} = -[c_4 + \lambda_1 - \lambda_4 + \lambda_8(2x_4 - 1)] \\
\frac{dx_5}{dt} = -[c_5 - \lambda_3 + \lambda_4 + \lambda_9(2x_5 - 1)] \\
\frac{dx_6}{dt} = -[c_6 + \lambda_2 - \lambda_4 + \lambda_{10}(2x_6 - 1)] \\
\frac{d\lambda_1}{dt} = +[x_3 + x_4 - x_1 - \delta_1] \\
\frac{d\lambda_2}{dt} = +[x_2 + x_6 - x_3 - \delta_2]
\end{cases}
\qquad
\begin{cases}
\frac{d\lambda_3}{dt} = +[x_1 - x_5 - x_2 - \delta_3] \\
\frac{d\lambda_4}{dt} = +[x_5 - x_6 - x_4 - \delta_4] \\
\frac{d\lambda_5}{dt} = +[x_1(x_1 - 1)] \\
\frac{d\lambda_6}{dt} = +[x_2(x_2 - 1)] \\
\frac{d\lambda_7}{dt} = +[x_3(x_3 - 1)] \\
\frac{d\lambda_8}{dt} = +[x_4(x_4 - 1)] \\
\frac{d\lambda_9}{dt} = +[x_5(x_5 - 1)] \\
\frac{d\lambda_{10}}{dt} = +[x_6(x_6 - 1)]
\end{cases}$$

❑ STEP 8: Equivalent form of the mathematical model of the Neuro-processor for the SPP

✓ The system above can be expressed/written into the following form:

$$\begin{cases}
\frac{dx_1}{dt} = -\left[c_1 - x_7 + x_9 + x_{11}(2x_1 - 1)\right] \\
\frac{dx_2}{dt} = -\left[c_2 + x_8 - x_9 + x_{12}(2x_2 - 1)\right] \\
\frac{dx_3}{dt} = -\left[c_3 + x_7 - x_8 + x_{13}(2x_3 - 1)\right] \\
\frac{dx_4}{dt} = -\left[c_4 + x_7 - x_{10} + x_{14}(2x_4 - 1)\right] \\
\frac{dx_5}{dt} = -\left[c_5 - x_9 + x_{10} + x_{15}(2x_5 - 1)\right] \\
\frac{dx_6}{dt} = -\left[c_6 + x_8 - x_{10} + x_{16}(2x_6 - 1)\right] \\
\frac{dx_7}{dt} = +\left[x_3 + x_4 - x_1 - \delta_1\right] \\
\frac{dx_8}{dt} = +\left[x_2 + x_6 - x_3 - \delta_2\right]
\end{cases}
\qquad
\begin{cases}
\frac{dx_9}{dt} = +\left[x_1 - x_5 - x_2 - \delta_3\right] \\
\frac{dx_{10}}{dt} = +\left[x_5 - x_6 - x_4 - \delta_4\right] \\
\frac{dx_{11}}{dt} = +\left[x_1(x_1 - 1)\right] \\
\frac{dx_{12}}{dt} = +\left[x_2(x_2 - 1)\right] \\
\frac{dx_{13}}{dt} = +\left[x_3(x_3 - 1)\right] \\
\frac{dx_{14}}{dt} = +\left[x_4(x_4 - 1)\right] \\
\frac{dx_{15}}{dt} = +\left[x_5(x_5 - 1)\right] \\
\frac{dx_{16}}{dt} = +\left[x_6(x_6 - 1)\right]
\end{cases}$$

$$\lambda_1 = x_7;\quad \lambda_2 = x_8;\quad \lambda_3 = x_9;\quad \lambda_4 = x_{10};\quad \lambda_5 = x_{11};\quad \lambda_6 = x_{12};\quad \lambda_7 = x_{13};\quad \lambda_8 = x_{14};\quad \lambda_9 = x_{15};\quad \lambda_{10} = x_{16};$$

5.3. MATLAB-CODING of the mathematical model on the Neuro-processor for SPP

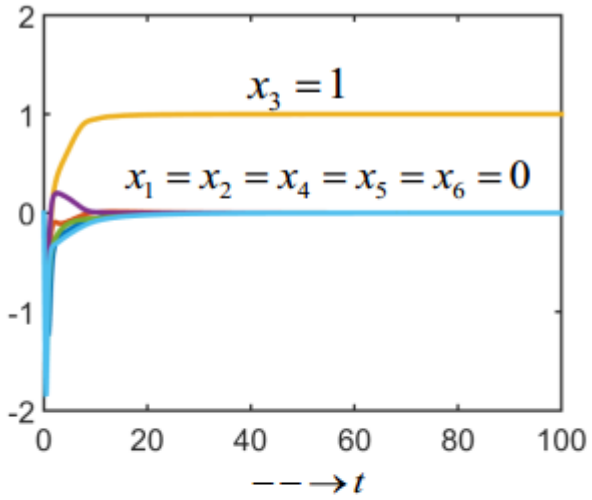
- **MATLAB-code of the mathematical model of the Neuro-processor for solving SPP**

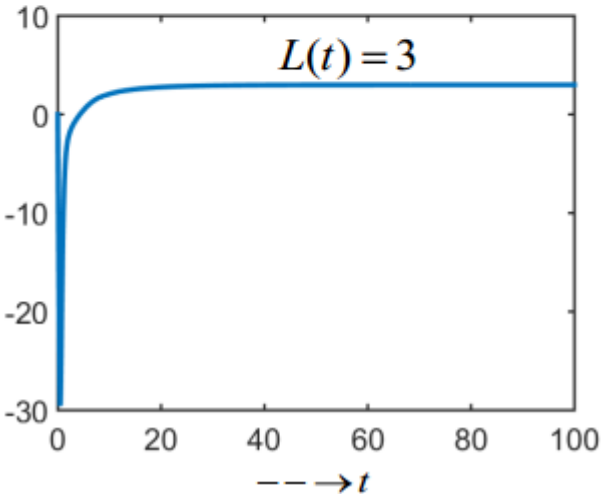
```
function dx=f(t,x)
dx=zeros(16,1);
c1=1; c2=2; c3=3; c4=4; c5=5; c6=6;
s1=1; s2=0; s3=0; s4=-1;
dx(1)=-(c1 - x(7) + x(9) )-x(11)*(2*x(1)-1);
dx(2)=-(c2 + x(8) - x(9) )-x(12)*(2*x(2)-1);
dx(3)=-(c3 + x(7) - x(8) )-x(13)*(2*x(3)-1);
dx(4)=-(c4 + x(7) - x(10))-x(14)*(2*x(4)-1);
dx(5)=-(c5 - x(9) + x(10))-x(15)*(2*x(5)-1);
dx(6)=-(c6 + x(8) - x(10))-x(16)*(2*x(6)-1);
dx(7)=+( x(3) + x(4) - x(1) - s1 );
dx(8)=+( x(2) + x(6) - x(3) - s2 );
dx(9)=+( x(1) - x(5) - x(2) - s3 );
dx(10)=+( x(5) - x(6) - x(4) - s4 );
dx(11)=+x(1)*(x(1)-1);
dx(12)=+x(2)*(x(2)-1);
dx(13)=+x(3)*(x(3)-1);
dx(14)=+x(4)*(x(4)-1);
dx(15)=+x(5)*(x(5)-1);
dx(16)=+x(6)*(x(6)-1);
end
```

Intelligent Transport Systems: New ICT – based Master's Curricula in Uzbekistan (INTRAS)
Agreement number: 2017-3516/001-001
Project reference number: 586292-EPP-1-2017-1-PL-EPPKA2-CBHE-JP

5.4. Results of the numerical computation of the SPP using the MATLAB-code above

- **Numerical results obtained by the MATLAB-code: N1 (Source) and N2 (destination)**

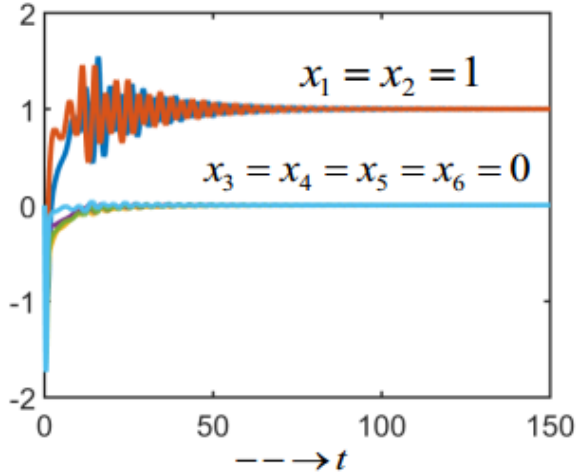


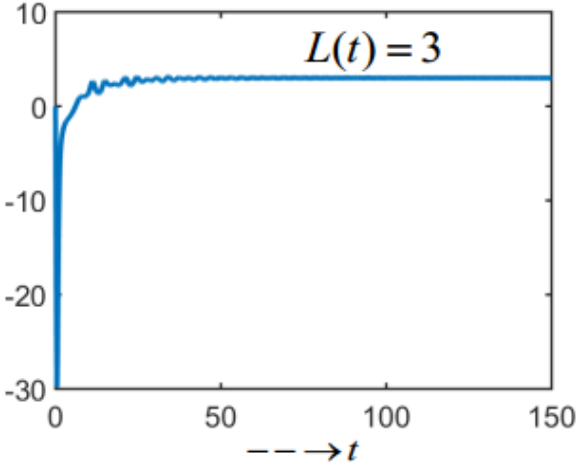



```
function dx=f(t,x)

dx=zeros(16,1);

c1=1; c2=2; c3=3; c4=4; c5=5; c6=6;
s1=1; s2=-1; s3=0; s4=0;

dx(1)=-(c1-x(7)+x(9) )-x(11)*(2*x(1)-1);
dx(2)=-(c2+x(8)-x(9) )-x(12)*(2*x(2)-1);
dx(3)=-(c3+x(7)-x(8) )-x(13)*(2*x(3)-1);
dx(4)=-(c4+x(7)-x(10))-x(14)*(2*x(4)-1);
dx(5)=-(c5-x(9)+x(10))-x(15)*(2*x(5)-1);
dx(6)=-(c6+x(8)-x(10))-x(16)*(2*x(6)-1);

dx(7)=+( x(3) + x(4) - x(1) - s1 );
dx(8)=+( x(2) + x(6) - x(3) - s2 );
dx(9)=+( x(1) - x(5) - x(2) - s3 );
dx(10)=+( x(5) - x(6) - x(4) - s4 );

dx(11)=+x(1)*(x(1)-1);
dx(12)=+x(2)*(x(2)-1);
dx(13)=+x(3)*(x(3)-1);
dx(14)=+x(4)*(x(4)-1);
dx(15)=+x(5)*(x(5)-1);
dx(16)=+x(6)*(x(6)-1);

end
```

5.5. Results of the numerical computation of the SPP using the MATLAB-code above

- **Numerical results obtained by the MATLAB-code: N2 (Source) and N1 (destination)**

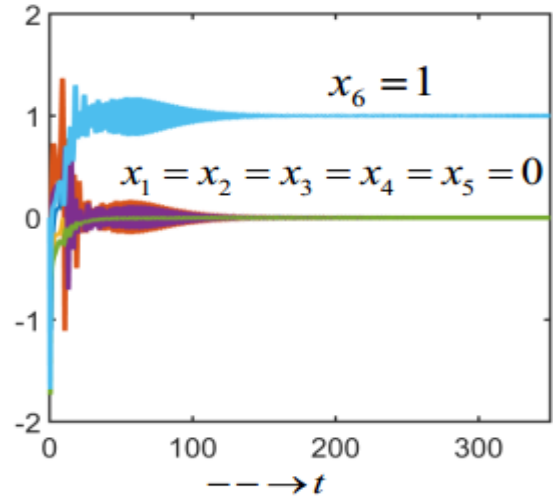


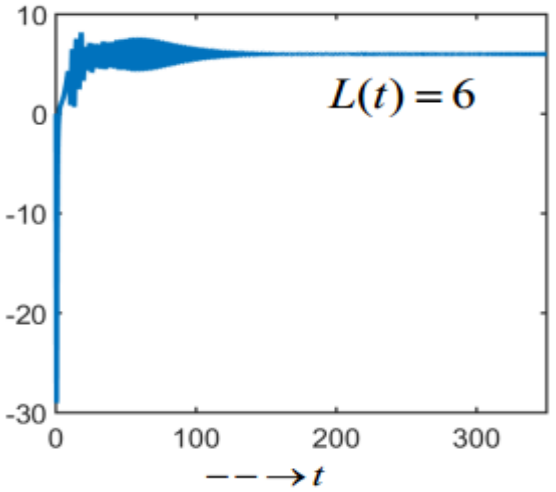



```
function dx=f(t,x)

dx=zeros(16,1);

c1=1; c2=2; c3=3; c4=4; c5=5; c6=6;
s1=-1; s2=+1; s3=0; s4=0;

dx(1)=-( c1 - x(7) + x(9) )-x(11)*(2*x(1)-1);
dx(2)=-( c2 + x(8) - x(9) )-x(12)*(2*x(2)-1);
dx(3)=-( c3 + x(7) - x(8) )-x(13)*(2*x(3)-1);
dx(4)=-( c4 + x(7) - x(10))-x(14)*(2*x(4)-1);
dx(5)=-( c5 - x(9) + x(10))-x(15)*(2*x(5)-1);
dx(6)=-( c6 + x(8) - x(10))-x(16)*(2*x(6)-1);

dx(7)=+( x(3) + x(4) - x(1) - s1 );
dx(8)=+( x(2) + x(6) - x(3) - s2 );
dx(9)=+( x(1) - x(5) - x(2) - s3 );
dx(10)=+( x(5) - x(6) - x(4) - s4 );

dx(11)=+x(1)*(x(1)-1);
dx(12)=+x(2)*(x(2)-1);
dx(13)=+x(3)*(x(3)-1);
dx(14)=+x(4)*(x(4)-1);
dx(15)=+x(5)*(x(5)-1);
dx(16)=+x(6)*(x(6)-1);

end
```

5.6. Results of the numerical computation of the SPP using the MATLAB-code above

- **Numerical results obtained by the MATLAB-code: N2 (Source) and N4 (destination)**

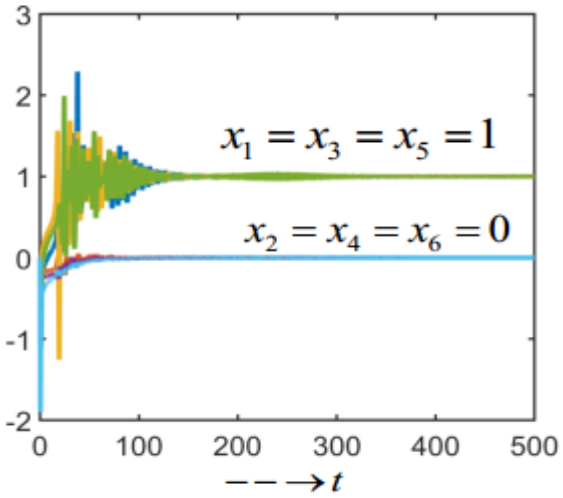


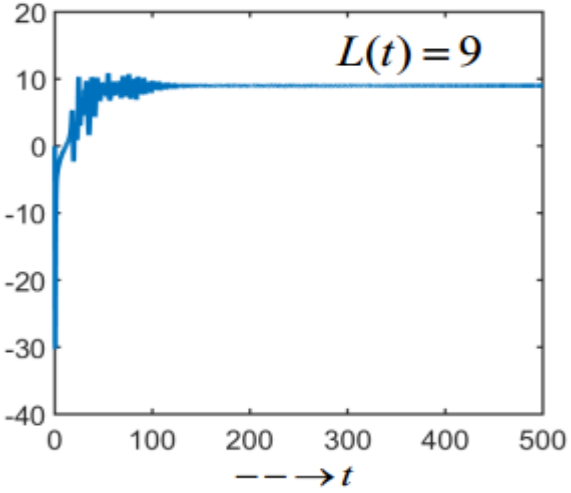



```
function dx=f(t,x)

dx=zeros(16,1);

c1=1; c2=2; c3=3; c4=4; c5=5; c6=6;
s1=0; s2=1; s3=0; s4=-1;

dx(1)=-( c1 - x(7) + x(9) )-x(11)*(2*x(1)-1);
dx(2)=-( c2 + x(8) - x(9) )-x(12)*(2*x(2)-1);
dx(3)=-( c3 + x(7) - x(8) )-x(13)*(2*x(3)-1);
dx(4)=-( c4 + x(7) - x(10))-x(14)*(2*x(4)-1);
dx(5)=-( c5 - x(9) + x(10))-x(15)*(2*x(5)-1);
dx(6)=-( c6 + x(8) - x(10))-x(16)*(2*x(6)-1);

dx(7)=+( x(3) + x(4) - x(1) - s1 );
dx(8)=+( x(2) + x(6) - x(3) - s2 );
dx(9)=+( x(1) - x(5) - x(2) - s3 );
dx(10)=+( x(5) - x(6) - x(4) - s4 );

dx(11)=+x(1)*(x(1)-1);
dx(12)=+x(2)*(x(2)-1);
dx(13)=+x(3)*(x(3)-1);
dx(14)=+x(4)*(x(4)-1);
dx(15)=+x(5)*(x(5)-1);
dx(16)=+x(6)*(x(6)-1);

end
```

5.7. Results of the numerical computation of the SPP using the MATLAB-code above

- **Numerical results obtained by the MATLAB-code: N4 (Source) and N2 (destination)**



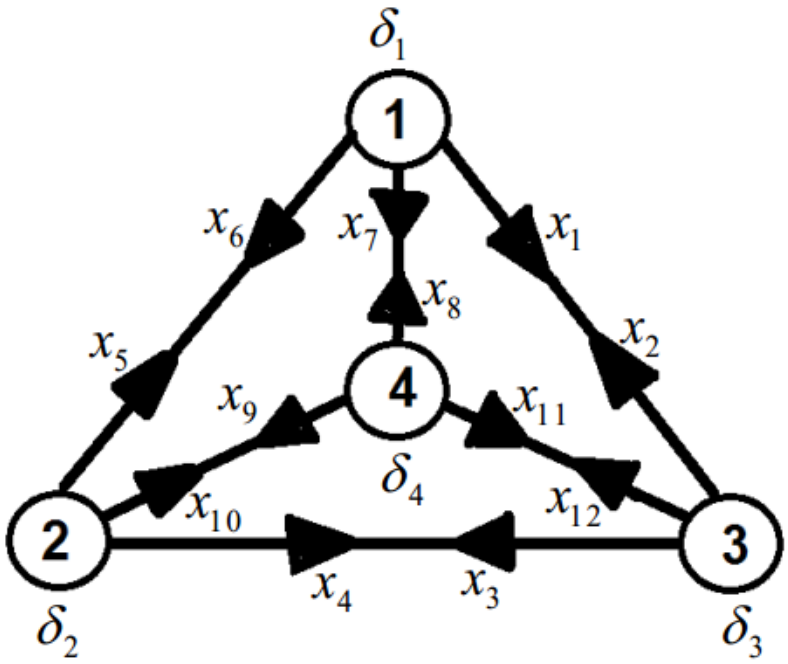



```
function dx=f(t,x)

dx=zeros(16,1);

c1=1; c2=2; c3=3; c4=4; c5=5; c6=6;
s1=0; s2=-1; s3=0; s4=1;

dx(1)=-( c1 - x(7) + x(9) )-x(11)*(2*x(1)-1);
dx(2)=-( c2 + x(8) - x(9) )-x(12)*(2*x(2)-1);
dx(3)=-( c3 + x(7) - x(8) )-x(13)*(2*x(3)-1);
dx(4)=-( c4 + x(7) - x(10))-x(14)*(2*x(4)-1);
dx(5)=-( c5 - x(9) + x(10))-x(15)*(2*x(5)-1);
dx(6)=-( c6 + x(8) - x(10))-x(16)*(2*x(6)-1);

dx(7)=+( x(3) + x(4) - x(1) - s1 );
dx(8)=+( x(2) + x(6) - x(3) - s2 );
dx(9)=+( x(1) - x(5) - x(2) - s3 );
dx(10)=+( x(5) - x(6) - x(4) - s4 );

dx(11)=+x(1)*(x(1)-1);
dx(12)=+x(2)*(x(2)-1);
dx(13)=+x(3)*(x(3)-1);
dx(14)=+x(4)*(x(4)-1);
dx(15)=+x(5)*(x(5)-1);
dx(16)=+x(6)*(x(6)-1);

end
```

5.8. Mathematical modeling of the shortest path problem (SPP) in an undirected graph of magnitude 4 and size 12“

- **Case study 2: A simple undirected graph of magnitude 4 and size 12 (see below)**

- **TASK: Mathematical modelling of the SPP using the theory of optimization in chap. 4**
  - ✓ In this section we use the knowledge acquired in chapter 4 to model the SPP (in the graph below) mathematically.



- **STEP 1: Expression of the objective function.**
  - ✓ This function is expressed as the minimization of the total cost of the graph.

$$Min\left[f(x_i,c_i)=c_1x_1+c_2x_2+c_3x_3+c_4x_4+c_5x_5+c_6x_6 + c_7x_7+c_8x_8+c_9x_9+c_{10}x_{10}+c_{11}x_{11}+c_{12}x_{12}\right]$$

- **STEP 2: Formulation of the optimization constraints: "Connectivity of nodes"**
  - ✓ Constraints are obtained through the assignment of attributes to all nodes of the graph. Three possible attributes are defined in the case of SPP: The source Node, The intermediate node, and the destination node. These attributes are modeled by the set of equations below:

$$g(x_i,\delta_i)=\begin{cases}\text{Node1}: (x_1+x_6+x_7-x_2-x_5-x_8)=\delta_1\\ \text{Node2}: (x_4+x_5+x_{10}-x_3-x_6-x_9)=\delta_2\\ \text{Node3}: (x_2+x_3+x_{12}-x_1-x_4-x_{11})=\delta_3\\ \text{Node4}: (x_8+x_9+x_{11}-x_7-x_{10}-x_{12})=\delta_4\end{cases}$$

$$(\text{Source Node}: \delta_i=1)$$
$$(\text{Intermediate Node}: \delta_i=0)$$
$$(\text{Destination Node}: \delta_i=-1)$$

- **STEP 3: Formulation of the optimization constraint: "Binarization"**
  - ✓ This constraint insures, that each edge of the graph takes only two possible values, which are "0" or "1" (Binarization):

$$h(x_i)=\begin{cases} x_1(x_1-1)=0 \\ x_2(x_2-1)=0 \\ x_3(x_3-1)=0 \\ x_4(x_4-1)=0 \\ x_5(x_5-1)=0 \\ x_6(x_6-1)=0 \end{cases} \qquad h(x_i)=\begin{cases} x_7(x_7-1)=0 \\ x_8(x_8-1)=0 \\ x_9(x_9-1)=0 \\ x_{10}(x_{10}-1)=0 \\ x_{11}(x_{11}-1)=0 \\ x_{12}(x_{12}-1)=0 \end{cases}$$

- **STEP 4: Expression of the Lagrange function as the total energy of the system**
  - ✓ The Lagrange function is obtained by combining the objective function with constraints. This combination is achieved using the multiplier variables λ_i.

$$\begin{aligned} L=&\left(c_1x_1+c_2x_2+c_3x_3+c_4x_4+c_5x_5+c_6x_6+c_7x_7+c_8x_8+c_9x_9+c_{10}x_{10}+c_{11}x_{11}+c_{12}x_{12}\right) \\ &+\lambda_1\left(x_1+x_6+x_7-x_2-x_5-x_8-\delta_1\right)+\lambda_2\left(x_4+x_5+x_{10}-x_3-x_6-x_9-\delta_2\right) \\ &+\lambda_3\left(x_2+x_3+x_{12}-x_1-x_4-x_{11}-\delta_3\right)+\lambda_4\left(x_8+x_9+x_{11}-x_7-x_{10}-x_{12}-\delta_4\right) \\ &+\lambda_5x_1\left(x_1-1\right)+\lambda_6x_2\left(x_2-1\right)+\lambda_7x_3\left(x_3-1\right)+\lambda_8x_4\left(x_4-1\right)+\lambda_9x_5\left(x_5-1\right)+\lambda_{10}x_6\left(x_6-1\right) \\ &+\lambda_{11}x_7\left(x_7-1\right)+\lambda_{12}x_8\left(x_8-1\right)+\lambda_{13}x_9\left(x_9-1\right)+\lambda_{14}x_{10}\left(x_{10}-1\right)+\lambda_{15}x_{11}\left(x_{11}-1\right)+\lambda_{16}x_{12}\left(x_{12}-1\right) \end{aligned}$$

- **STEP 5: Application of the BDMM concept and derivation of the coupled ODEs**
  - ✓ The gradient descent (see chapter 4) is applied to decision neurons/variables to obtain the set of coupled ODEs below:

$$\begin{cases}
\frac{dx_1}{dt} = -\frac{\partial L}{\partial x_1} = -\left[c_1 + \lambda_1 - \lambda_3 + \lambda_5(2x_1 - 1)\right] \\
\frac{dx_2}{dt} = -\frac{\partial L}{\partial x_2} = -\left[c_2 - \lambda_1 + \lambda_3 + \lambda_6(2x_2 - 1)\right] \\
\frac{dx_3}{dt} = -\frac{\partial L}{\partial x_3} = -\left[c_3 - \lambda_2 + \lambda_3 + \lambda_7(2x_3 - 1)\right] \\
\frac{dx_4}{dt} = -\frac{\partial L}{\partial x_4} = -\left[c_4 + \lambda_2 - \lambda_3 + \lambda_8(2x_4 - 1)\right] \\
\frac{dx_5}{dt} = -\frac{\partial L}{\partial x_5} = -\left[c_5 - \lambda_1 + \lambda_2 + \lambda_9(2x_5 - 1)\right] \\
\frac{dx_6}{dt} = -\frac{\partial L}{\partial x_6} = -\left[c_6 + \lambda_1 - \lambda_2 + \lambda_{10}(2x_6 - 1)\right]
\end{cases}$$

$$\begin{cases}
\frac{dx_7}{dt} = -\frac{\partial L}{\partial x_7} = -\left[c_7 + \lambda_1 - \lambda_4 + \lambda_{11}(2x_7 - 1)\right] \\
\frac{dx_8}{dt} = -\frac{\partial L}{\partial x_8} = -\left[c_8 - \lambda_1 + \lambda_4 + \lambda_{12}(2x_8 - 1)\right] \\
\frac{dx_9}{dt} = -\frac{\partial L}{\partial x_9} = -\left[c_9 - \lambda_2 + \lambda_4 + \lambda_{13}(2x_9 - 1)\right] \\
\frac{dx_{10}}{dt} = -\frac{\partial L}{\partial x_{10}} = -\left[c_{10} + \lambda_2 - \lambda_4 + \lambda_{14}(2x_{10} - 1)\right] \\
\frac{dx_{11}}{dt} = -\frac{\partial L}{\partial x_{11}} = -\left[c_{11} - \lambda_3 + \lambda_4 + \lambda_{15}(2x_{11} - 1)\right] \\
\frac{dx_{12}}{dt} = -\frac{\partial L}{\partial x_{12}} = -\left[c_{12} + \lambda_3 - \lambda_4 + \lambda_{16}(2x_{12} - 1)\right]
\end{cases}$$

- **STEP 6: Application of the BDMM concept and derivation of the coupled ODEs**
  - ✓ The gradient ascent (see chapter 4) is applied to the **GROUP 1** of multiplier neurons/variables (ensuring the connectivity of nodes) to obtain the set of coupled ODEs below:

$$\begin{cases} \dfrac{d\lambda_1}{dt} = +\dfrac{\partial L}{\partial \lambda_1} = +\left[x_1 + x_6 + x_7 - x_2 - x_5 - x_8 - \delta_1\right] \\ \dfrac{d\lambda_2}{dt} = +\dfrac{\partial L}{\partial \lambda_2} = +\left[x_4 + x_5 + x_{10} - x_3 - x_6 - x_9 - \delta_2\right] \\ \dfrac{d\lambda_3}{dt} = +\dfrac{\partial L}{\partial \lambda_3} = +\left[x_2 + x_3 + x_{12} - x_1 - x_4 - x_{11} - \delta_3\right] \\ \dfrac{d\lambda_4}{dt} = +\dfrac{\partial L}{\partial \lambda_4} = +\left[x_8 + x_9 + x_{11} - x_7 - x_{10} - x_{12} - \delta_4\right] \end{cases}$$

- **STEP 7: Application of the BDMM concept and derivation of the coupled ODEs**
  - ✓ The gradient ascent (see chapter 4) is applied to the **GROUP 2** of multiplier neurons (ensuring the binarization) to obtain the set of coupled ODEs below:

$$\begin{cases} \frac{d\lambda_5}{dt} = +\frac{\partial L}{\partial \lambda_5} = +\left[x_1(x_1-1)\right] \\ \frac{d\lambda_6}{dt} = +\frac{\partial L}{\partial \lambda_6} = +\left[x_2(x_2-1)\right] \\ \frac{d\lambda_7}{dt} = +\frac{\partial L}{\partial \lambda_7} = +\left[x_3(x_3-1)\right] \\ \frac{d\lambda_8}{dt} = +\frac{\partial L}{\partial \lambda_8} = +\left[x_4(x_4-1)\right] \\ \frac{d\lambda_9}{dt} = +\frac{\partial L}{\partial \lambda_9} = +\left[x_5(x_5-1)\right] \\ \frac{d\lambda_{10}}{dt} = +\frac{\partial L}{\partial \lambda_{10}} = +\left[x_6(x_6-1)\right] \end{cases} \qquad \begin{cases} \frac{d\lambda_{11}}{dt} = +\frac{\partial L}{\partial \lambda_{11}} = +\left[x_7(x_7-1)\right] \\ \frac{d\lambda_{12}}{dt} = +\frac{\partial L}{\partial \lambda_{12}} = +\left[x_8(x_8-1)\right] \\ \frac{d\lambda_{13}}{dt} = +\frac{\partial L}{\partial \lambda_{13}} = +\left[x_9(x_9-1)\right] \\ \frac{d\lambda_{14}}{dt} = +\frac{\partial L}{\partial \lambda_{14}} = +\left[x_{10}(x_{10}-1)\right] \\ \frac{d\lambda_{15}}{dt} = +\frac{\partial L}{\partial \lambda_{15}} = +\left[x_{11}(x_{11}-1)\right] \\ \frac{d\lambda_{16}}{dt} = +\frac{\partial L}{\partial \lambda_{16}} = +\left[x_{12}(x_{12}-1)\right] \end{cases}$$

❑ STEP 8: Mathematical model of the Neuro-processor for the SPP

✓ The Mathematical model of the Neuro-processor is obtained as a combination of the set of equations obtained in STEPS 5, 6 and 7.

$$\begin{cases}
\dot{x}_1 = -\left[c_1 + \lambda_1 - \lambda_3 + \lambda_5(2x_1 - 1)\right] \\
\dot{x}_2 = -\left[c_2 - \lambda_1 + \lambda_3 + \lambda_6(2x_2 - 1)\right] \\
\dot{x}_3 = -\left[c_3 - \lambda_2 + \lambda_3 + \lambda_7(2x_3 - 1)\right] \\
\dot{x}_4 = -\left[c_4 + \lambda_2 - \lambda_3 + \lambda_8(2x_4 - 1)\right] \\
\dot{x}_5 = -\left[c_5 - \lambda_1 + \lambda_2 + \lambda_9(2x_5 - 1)\right] \\
\dot{x}_6 = -\left[c_6 + \lambda_1 - \lambda_2 + \lambda_{10}(2x_6 - 1)\right] \\
\dot{x}_7 = -\left[c_7 + \lambda_1 - \lambda_4 + \lambda_{11}(2x_7 - 1)\right] \\
\dot{x}_8 = -\left[c_8 - \lambda_1 + \lambda_4 + \lambda_{12}(2x_8 - 1)\right] \\
\dot{x}_9 = -\left[c_9 - \lambda_2 + \lambda_4 + \lambda_{13}(2x_9 - 1)\right] \\
\dot{x}_{10} = -\left[c_{10} + \lambda_2 - \lambda_4 + \lambda_{14}(2x_{10} - 1)\right] \\
\dot{x}_{11} = -\left[c_{11} - \lambda_3 + \lambda_4 + \lambda_{15}(2x_{11} - 1)\right] \\
\dot{x}_{12} = -\left[c_{12} + \lambda_3 - \lambda_4 + \lambda_{16}(2x_{12} - 1)\right]
\end{cases}$$

$$\begin{cases}
\dot{\lambda}_1 = +\left[x_1 + x_6 + x_7 - x_2 - x_5 - x_8 - \delta_1\right] \\
\dot{\lambda}_2 = +\left[x_4 + x_5 + x_{10} - x_3 - x_6 - x_9 - \delta_2\right] \\
\dot{\lambda}_3 = +\left[x_2 + x_3 + x_{12} - x_1 - x_4 - x_{11} - \delta_3\right] \\
\dot{\lambda}_4 = +\left[x_8 + x_9 + x_{11} - x_7 - x_{10} - x_{12} - \delta_4\right]
\end{cases}$$

$$\begin{cases}
\dot{\lambda}_5 = +\left[x_1(x_1 - 1)\right] \\
\dot{\lambda}_6 = +\left[x_2(x_2 - 1)\right] \\
\dot{\lambda}_7 = +\left[x_3(x_3 - 1)\right] \\
\dot{\lambda}_8 = +\left[x_4(x_4 - 1)\right] \\
\dot{\lambda}_9 = +\left[x_5(x_5 - 1)\right] \\
\dot{\lambda}_{10} = +\left[x_6(x_6 - 1)\right] \\
\dot{\lambda}_{11} = +\left[x_7(x_7 - 1)\right] \\
\dot{\lambda}_{12} = +\left[x_8(x_8 - 1)\right] \\
\dot{\lambda}_{13} = +\left[x_9(x_9 - 1)\right] \\
\dot{\lambda}_{14} = +\left[x_{10}(x_{10} - 1)\right] \\
\dot{\lambda}_{15} = +\left[x_{11}(x_{11} - 1)\right] \\
\dot{\lambda}_{16} = +\left[x_{12}(x_{12} - 1)\right]
\end{cases}$$

- **STEP 8: Equivalent form of the mathematical model of the Neuro-processor for the SPP**
  - ✓ The system above can be expressed/written into the following form:

$$\begin{cases}
\dot{x}_1 = -\left[c_1 + x_{13} - x_{15} + x_{17}(2x_1 - 1)\right] \\
\dot{x}_2 = -\left[c_2 - x_{13} + x_{15} + x_{18}(2x_2 - 1)\right] \\
\dot{x}_3 = -\left[c_3 - x_{14} + x_{15} + x_{19}(2x_3 - 1)\right] \\
\dot{x}_4 = -\left[c_4 + x_{14} - x_{15} + x_{20}(2x_4 - 1)\right] \\
\dot{x}_5 = -\left[c_5 - x_{13} + x_{14} + x_{21}(2x_5 - 1)\right] \\
\dot{x}_6 = -\left[c_6 + x_{13} - x_{14} + x_{22}(2x_6 - 1)\right] \\
\dot{x}_7 = -\left[c_7 + x_{13} - x_{16} + x_{23}(2x_7 - 1)\right] \\
\dot{x}_8 = -\left[c_8 - x_{13} + x_{16} + x_{24}(2x_8 - 1)\right] \\
\dot{x}_9 = -\left[c_9 - x_{14} + x_{16} + x_{25}(2x_9 - 1)\right] \\
\dot{x}_{10} = -\left[c_{10} + x_{14} - x_{16} + x_{26}(2x_{10} - 1)\right] \\
\dot{x}_{11} = -\left[c_{11} - x_{15} + x_{16} + x_{27}(2x_{11} - 1)\right] \\
\dot{x}_{12} = -\left[c_{12} + x_{15} - x_{16} + x_{28}(2x_{12} - 1)\right]
\end{cases}$$

$$\begin{cases}
\dot{x}_{13} = +\left[x_1 + x_6 + x_7 - x_2 - x_5 - x_8 - \delta_1\right] \\
\dot{x}_{14} = +\left[x_4 + x_5 + x_{10} - x_3 - x_6 - x_9 - \delta_2\right] \\
\dot{x}_{15} = +\left[x_2 + x_3 + x_{12} - x_1 - x_4 - x_{11} - \delta_3\right] \\
\dot{x}_{16} = +\left[x_8 + x_9 + x_{11} - x_7 - x_{10} - x_{12} - \delta_4\right]
\end{cases}$$

$$\begin{cases}
\dot{x}_{17} = +\left[x_1(x_1 - 1)\right] \\
\dot{x}_{18} = +\left[x_2(x_2 - 1)\right] \\
\dot{x}_{19} = +\left[x_3(x_3 - 1)\right] \\
\dot{x}_{20} = +\left[x_4(x_4 - 1)\right] \\
\dot{x}_{21} = +\left[x_5(x_5 - 1)\right] \\
\dot{x}_{22} = +\left[x_6(x_6 - 1)\right] \\
\dot{x}_{23} = +\left[x_7(x_7 - 1)\right] \\
\dot{x}_{24} = +\left[x_8(x_8 - 1)\right] \\
\dot{x}_{25} = +\left[x_9(x_9 - 1)\right] \\
\dot{x}_{26} = +\left[x_{10}(x_{10} - 1)\right] \\
\dot{x}_{27} = +\left[x_{11}(x_{11} - 1)\right] \\
\dot{x}_{28} = +\left[x_{12}(x_{12} - 1)\right]
\end{cases}$$

5.9. MATLAB-CODING of the mathematical model on the Neuro-processor for SPP

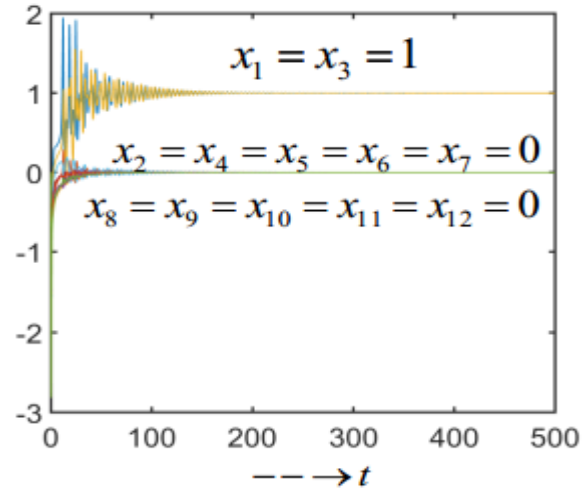
- **MATLAB-code of the mathematical model of the Neuro-processor for solving SPP**

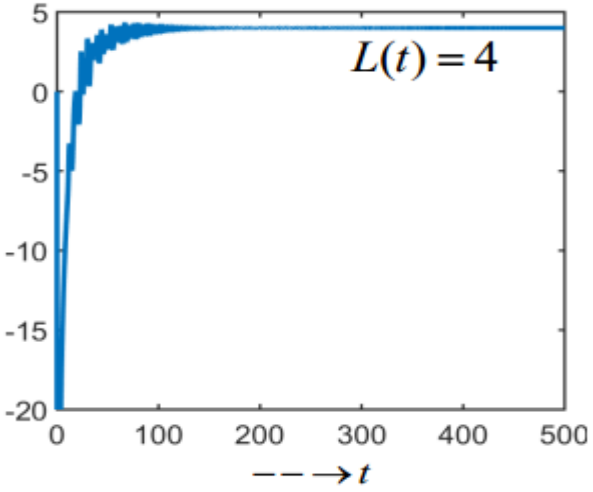
```
function dx=f(t,x)
dx=zeros(28,1);
c1=1;c2=2;c3=3;c4=4;c5=5;c6=6;c7=7;c8=8;c9=9;c10=10;
c11=11;c12=12;                s1=1; s2=-1; s3=0; s4=0;

dx(1)=-( c1 + x(13) - x(15) + x(17)*(2*x(1)-1) );
dx(2)=-( c2 - x(13) + x(15) + x(18)*(2*x(2)-1) );
dx(3)=-( c3 - x(14) + x(15) + x(19)*(2*x(3)-1) );
dx(4)=-( c4 + x(14) - x(15) + x(20)*(2*x(4)-1) );
dx(5)=-( c5 - x(13) + x(14) + x(21)*(2*x(5)-1) );
dx(6)=-( c6 + x(13) - x(14) + x(22)*(2*x(6)-1) );
dx(7)=-( c7 + x(13) - x(16) + x(23)*(2*x(7)-1) );
dx(8)=-( c8 - x(13) + x(16) + x(24)*(2*x(8)-1) );
dx(9)=-( c9 - x(14) + x(16) + x(25)*(2*x(9)-1) );
dx(10)=-( c10 + x(14) - x(16) + x(26)*(2*x(10)-1) );
dx(11)=-( c11 - x(15) + x(16) + x(27)*(2*x(11)-1) );
dx(12)=-( c12 + x(15) - x(16) + x(28)*(2*x(12)-1) );

dx(13)=+(x(1)+x(6)+x(7) - x(2)- x(5) - x(8) - s1);
dx(14)=+(x(4)+x(5)+x(10)- x(3)- x(6) - x(9) - s2);
dx(15)=+(x(2)+x(3)+x(12)- x(1)- x(4) - x(11)- s3);
dx(16)=+(x(8)+x(9)+x(11)- x(7)- x(10)- x(12)- s4);

dx(17)=+x(1)*(x(1)-1);
dx(18)=+x(2)*(x(2)-1);
dx(19)=+x(3)*(x(3)-1);
dx(20)=+x(4)*(x(4)-1);
dx(21)=+x(5)*(x(5)-1);
dx(22)=+x(6)*(x(6)-1);
dx(23)=+x(7)*(x(7)-1);
dx(24)=+x(8)*(x(8)-1);
dx(25)=+x(9)*(x(9)-1);
dx(26)=+x(10)*(x(10)-1);
dx(27)=+x(11)*(x(11)-1);
dx(28)=+x(12)*(x(12)-1);
end
```

Intelligent Transport Systems: New ICT – based Master's Curricula in Uzbekistan (INTRAS)
Agreement number: 2017-3516/001-001
Project reference number: 586292-EPP-1-2017-1-PL-EPPKA2-CBHE-JP

5.10. Results of the numerical computation of the SPP using the MATLAB-code above

- **Numerical results obtained by the MATLAB-code: N1 (Source) and N2 (destination)**

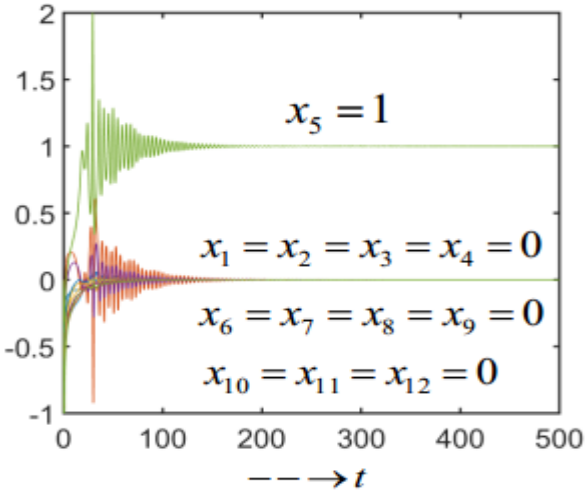


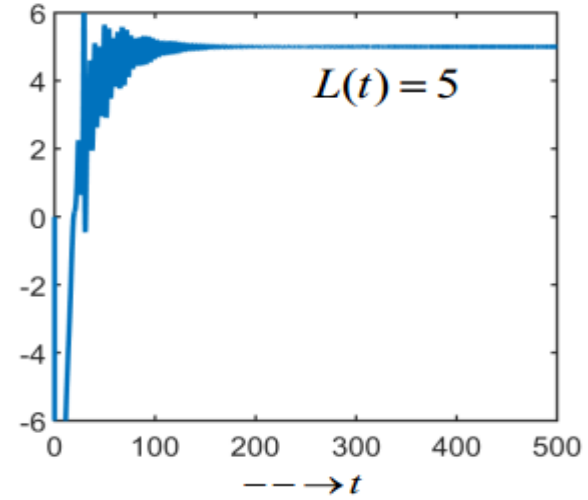



```
function dx=f(t,x)
dx=zeros(28,1);
c1=1;c2=2;c3=3;c4=4;c5=5;c6=6;c7=7;c8=8;c9=9;c10=10;
c11=11;c12=12;                s1=1; s2=-1; s3=0; s4=0;
dx(1)=-( c1 + x(13) - x(15) + x(17)*(2*x(1)-1) );
dx(2)=-( c2 - x(13) + x(15) + x(18)*(2*x(2)-1) );
dx(3)=-( c3 - x(14) + x(15) + x(19)*(2*x(3)-1) );
dx(4)=-( c4 + x(14) - x(15) + x(20)*(2*x(4)-1) );
dx(5)=-( c5 - x(13) + x(14) + x(21)*(2*x(5)-1) );
dx(6)=-( c6 + x(13) - x(14) + x(22)*(2*x(6)-1) );
dx(7)=-( c7 + x(13) - x(16) + x(23)*(2*x(7)-1) );
dx(8)=-( c8 - x(13) + x(16) + x(24)*(2*x(8)-1) );
dx(9)=-( c9 - x(14) + x(16) + x(25)*(2*x(9)-1) );
dx(10)=-( c10 + x(14) - x(16) + x(26)*(2*x(10)-1) );
dx(11)=-( c11 - x(15) + x(16) + x(27)*(2*x(11)-1) );
dx(12)=-( c12 + x(15) - x(16) + x(28)*(2*x(12)-1) );
dx(13)=+( x(1) + x(6)+x(7) - x(2) - x(5) - x(8) - s1 );
dx(14)=+( x(4) + x(5)+x(10) - x(3) - x(6) - x(9) - s2 );
dx(15)=+( x(2) + x(3)+x(12) - x(1) - x(4) - x(11) - s3 );
dx(16)=+( x(8) + x(9)+x(11) - x(7) - x(10) - x(12) -s4 );
dx(17)=+x(1)*(x(1)-1);
dx(18)=+x(2)*(x(2)-1);
dx(19)=+x(3)*(x(3)-1);
dx(20)=+x(4)*(x(4)-1);
dx(21)=+x(5)*(x(5)-1);
dx(22)=+x(6)*(x(6)-1);
dx(23)=+x(7)*(x(7)-1);
dx(24)=+x(8)*(x(8)-1);
dx(25)=+x(9)*(x(9)-1);
dx(26)=+x(10)*(x(10)-1);
dx(27)=+x(11)*(x(11)-1);
dx(28)=+x(12)*(x(12)-1);
end
```

5.11. Results of the numerical computation of the SPP using the MATLAB-code above

- **Numerical results obtained by the MATLAB-code: N2 (Source) and N1 (destination)**

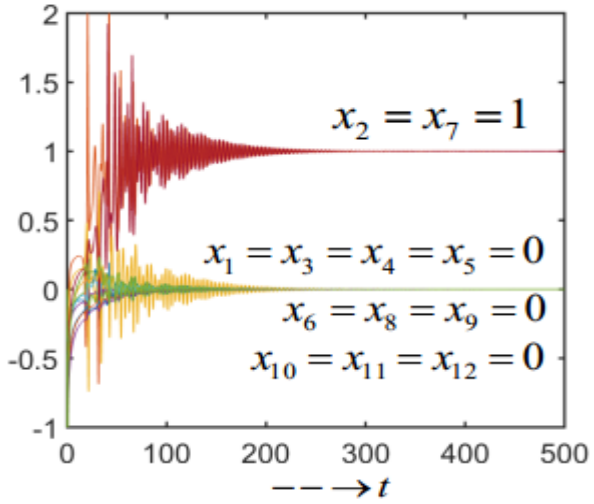


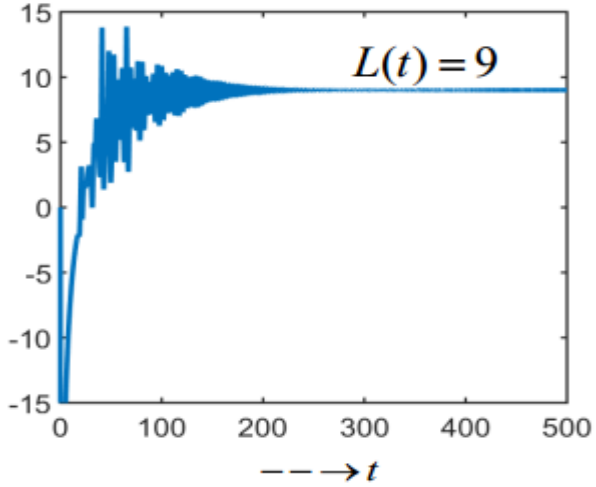



```
function dx=f(t,x)
dx=zeros(28,1);
c1=1;c2=2;c3=3;c4=4;c5=5;c6=6;c7=7;c8=8;c9=9;c10=10;
c11=11;c12=12;                s1=-1; s2=1;  s3=0; s4=0;

dx(1)=-( c1 + x(13) - x(15) + x(17)*(2*x(1)-1) );
dx(2)=-( c2 - x(13) + x(15) + x(18)*(2*x(2)-1) );
dx(3)=-( c3 - x(14) + x(15) + x(19)*(2*x(3)-1) );
dx(4)=-( c4 + x(14) - x(15) + x(20)*(2*x(4)-1) );
dx(5)=-( c5 - x(13) + x(14) + x(21)*(2*x(5)-1) );
dx(6)=-( c6 + x(13) - x(14) + x(22)*(2*x(6)-1) );
dx(7)=-( c7 + x(13) - x(16) + x(23)*(2*x(7)-1) );
dx(8)=-( c8 - x(13) + x(16) + x(24)*(2*x(8)-1) );
dx(9)=-( c9 - x(14) + x(16) + x(25)*(2*x(9)-1) );
dx(10)=-( c10 + x(14) - x(16) + x(26)*(2*x(10)-1) );
dx(11)=-( c11 - x(15) + x(16) + x(27)*(2*x(11)-1) );
dx(12)=-( c12 + x(15) - x(16) + x(28)*(2*x(12)-1) );
dx(13)=+( x(1) + x(6)+x(7) - x(2) - x(5) - x(8) - s1 );
dx(14)=+( x(4) + x(5)+x(10) - x(3) - x(6) - x(9) - s2 );
dx(15)=+( x(2) + x(3)+x(12) - x(1) - x(4) - x(11) - s3 );
dx(16)=+( x(8) + x(9)+x(11) - x(7) - x(10) - x(12) -s4 );
dx(17)=+x(1)*(x(1)-1);
dx(18)=+x(2)*(x(2)-1);
dx(19)=+x(3)*(x(3)-1);
dx(20)=+x(4)*(x(4)-1);
dx(21)=+x(5)*(x(5)-1);
dx(22)=+x(6)*(x(6)-1);
dx(23)=+x(7)*(x(7)-1);
dx(24)=+x(8)*(x(8)-1);
dx(25)=+x(9)*(x(9)-1);
dx(26)=+x(10)*(x(10)-1);
dx(27)=+x(11)*(x(11)-1);
dx(28)=+x(12)*(x(12)-1);
end
```

5.12. Results of the numerical computation of the SPP using the MATLAB-code above

- **Numerical results obtained by the MATLAB-code: N3 (Source) and N4 (destination)**

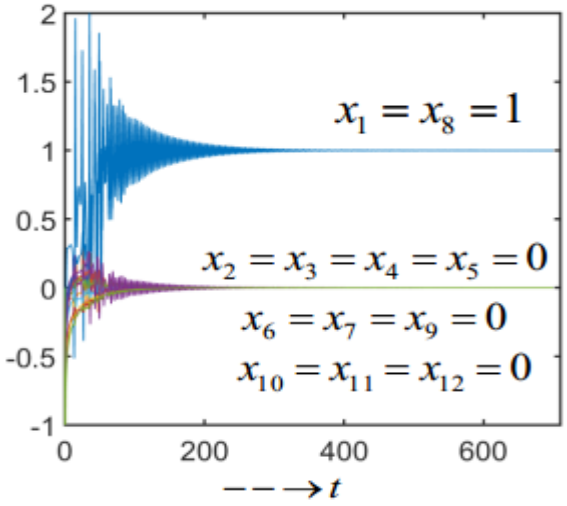


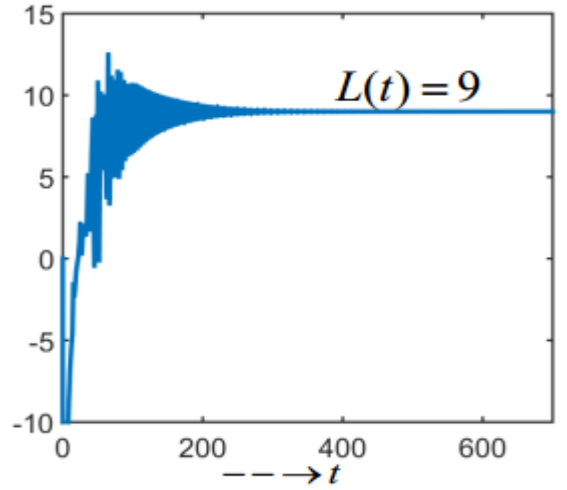



```
function dx=f(t,x)
dx=zeros(28,1);
c1=1;c2=2;c3=3;c4=4;c5=5;c6=6;c7=7;c8=8;c9=9;c10=10;
c11=11;c12=12;                s1=0; s2=0; s3=1; s4=-1;

dx(1)=-( c1 + x(13) - x(15) + x(17)*(2*x(1)-1) );
dx(2)=-( c2 - x(13) + x(15) + x(18)*(2*x(2)-1) );
dx(3)=-( c3 - x(14) + x(15) + x(19)*(2*x(3)-1) );
dx(4)=-( c4 + x(14) - x(15) + x(20)*(2*x(4)-1) );
dx(5)=-( c5 - x(13) + x(14) + x(21)*(2*x(5)-1) );
dx(6)=-( c6 + x(13) - x(14) + x(22)*(2*x(6)-1) );
dx(7)=-( c7 + x(13) - x(16) + x(23)*(2*x(7)-1) );
dx(8)=-( c8 - x(13) + x(16) + x(24)*(2*x(8)-1) );
dx(9)=-( c9 - x(14) + x(16) + x(25)*(2*x(9)-1) );
dx(10)=-( c10 + x(14) - x(16) + x(26)*(2*x(10)-1) );
dx(11)=-( c11 - x(15) + x(16) + x(27)*(2*x(11)-1) );
dx(12)=-( c12 + x(15) - x(16) + x(28)*(2*x(12)-1) );

dx(13)=+( x(1) + x(6) + x(7) - x(2) - x(5) - x(8) - s1 );
dx(14)=+( x(4) + x(5) + x(10) - x(3) - x(6) - x(9) - s2 );
dx(15)=+( x(2) + x(3) + x(12) - x(1) - x(4) - x(11) - s3 );
dx(16)=+( x(8) + x(9) + x(11) - x(7) - x(10) - x(12) - s4 );

dx(17)=+x(1)*(x(1)-1);
dx(18)=+x(2)*(x(2)-1);
dx(19)=+x(3)*(x(3)-1);
dx(20)=+x(4)*(x(4)-1);
dx(21)=+x(5)*(x(5)-1);
dx(22)=+x(6)*(x(6)-1);
dx(23)=+x(7)*(x(7)-1);
dx(24)=+x(8)*(x(8)-1);
dx(25)=+x(9)*(x(9)-1);
dx(26)=+x(10)*(x(10)-1);
dx(27)=+x(11)*(x(11)-1);
dx(28)=+x(12)*(x(12)-1);
end
```

5.13. Results of the numerical computation of the SPP using the MATLAB-code above

- **Numerical results obtained by the MATLAB-code: N4 (Source) and N3 (destination)**



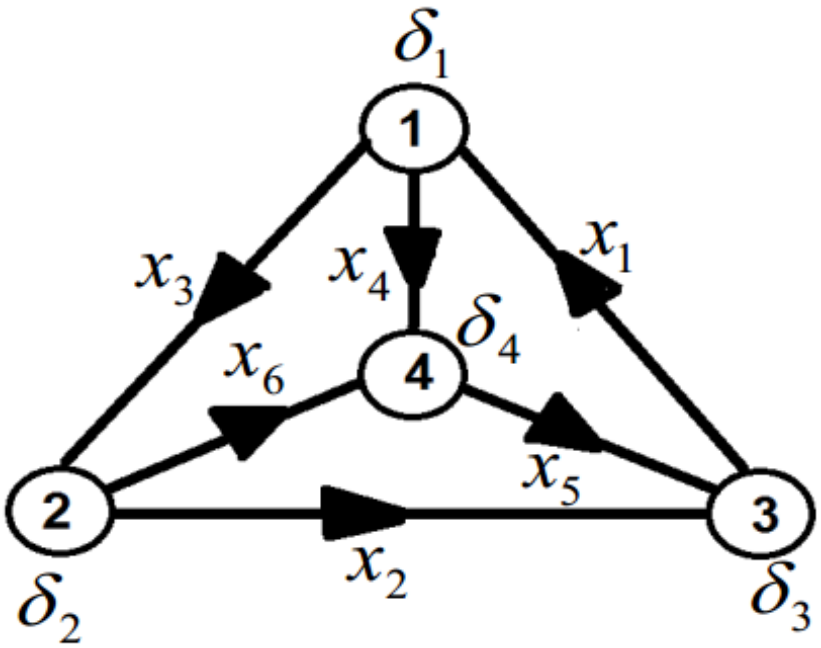



```
function dx=f(t,x)
dx=zeros(28,1);
c1=1;c2=2;c3=3;c4=4;c5=5;c6=6;c7=7;c8=8;c9=9;c10=10;
c11=11;c12=12;                s1=0; s2=0; s3=-1; s4=1;

dx(1)=-( c1 + x(13) - x(15) + x(17)*(2*x(1)-1) );
dx(2)=-( c2 - x(13) + x(15) + x(18)*(2*x(2)-1) );
dx(3)=-( c3 - x(14) + x(15) + x(19)*(2*x(3)-1) );
dx(4)=-( c4 + x(14) - x(15) + x(20)*(2*x(4)-1) );
dx(5)=-( c5 - x(13) + x(14) + x(21)*(2*x(5)-1) );
dx(6)=-( c6 + x(13) - x(14) + x(22)*(2*x(6)-1) );
dx(7)=-( c7 + x(13) - x(16) + x(23)*(2*x(7)-1) );
dx(8)=-( c8 - x(13) + x(16) + x(24)*(2*x(8)-1) );
dx(9)=-( c9 - x(14) + x(16) + x(25)*(2*x(9)-1) );
dx(10)=-( c10 + x(14) - x(16) + x(26)*(2*x(10)-1) );
dx(11)=-( c11 - x(15) + x(16) + x(27)*(2*x(11)-1) );
dx(12)=-( c12 + x(15) - x(16) + x(28)*(2*x(12)-1) );

dx(13)=+( x(1) + x(6) + x(7) - x(2) - x(5) - x(8) - s1 );
dx(14)=+( x(4) + x(5) + x(10) - x(3) - x(6) - x(9) - s2 );
dx(15)=+( x(2) + x(3) + x(12) - x(1) - x(4) - x(11) - s3 );
dx(16)=+( x(8) + x(9) + x(11) - x(7) - x(10) - x(12) - s4 );

dx(17)=+x(1)*(x(1)-1);
dx(18)=+x(2)*(x(2)-1);
dx(19)=+x(3)*(x(3)-1);
dx(20)=+x(4)*(x(4)-1);
dx(21)=+x(5)*(x(5)-1);
dx(22)=+x(6)*(x(6)-1);
dx(23)=+x(7)*(x(7)-1);
dx(24)=+x(8)*(x(8)-1);
dx(25)=+x(9)*(x(9)-1);
dx(26)=+x(10)*(x(10)-1);
dx(27)=+x(11)*(x(11)-1);
dx(28)=+x(12)*(x(12)-1);
end
```

5. 14. Mathematical modeling of the Traveling Salesman problem (TSP) in a directed graph of magnitude 4 and size 6

- **Case study 1: A simple directed graph of magnitude 4 and size 6 (see below)**

- **TASK: Mathematical modelling of the TSP using the theory of optimization in chap. 4**
  - ✓ In this section we use the knowledge acquired in chapter 4 to model the TSP (in the graph below) mathematically.



Intelligent Transport Systems: New ICT – based Master's Curricula in Uzbekistan (INTRAS)
Agreement number: 2017-3516/001-001
Project reference number: 586292-EPP-1-2017-1-PL-EPPKA2-CBHE-JP

- **STEP 1: Expression of the objective function.**
  - ✓ This function is expressed as the minimization of the total cost of the graph.

$$Min\left[f(x_i, c_i) = \left(c_1x_1 + c_2x_2 + c_3x_3 + c_4x_4 + c_5x_5 + c_6x_6\right)\right]$$

- **STEP 2: Formulation of the optimization constraints: "Connectivity of nodes"**
  - ✓ Constraints are formulated/obtained through the assignment of a unique attribute to all nodes of the graph. The attribute of intermediate node is assigned to each node of the graph. This ensure the belonging of all nodes to the TSP solution.

$$g(x_i, \delta_i) = \begin{cases} \text{Node1}: \left(x_3 + x_4 - x_1\right) = \delta_1 \ \cap \ \left(x_3 + x_4 + x_1\right) = \delta_2 \\ \text{Node2}: \left(x_2 + x_6 - x_3\right) = \delta_3 \ \cap \ \left(x_2 + x_6 + x_3\right) = \delta_4 \\ \text{Node3}: \left(x_1 - x_5 - x_2\right) = \delta_5 \ \cap \ \left(x_1 + x_5 + x_2\right) = \delta_6 \\ \text{Node4}: \left(x_5 - x_6 - x_4\right) = \delta_7 \ \cap \ \left(x_5 + x_6 + x_4\right) = \delta_8 \end{cases} \qquad \begin{cases} \delta_i = 0 \ \ (i: \text{odd}) \\ \delta_i = 2 \ \ (i: \text{even}) \end{cases}$$

Intelligent Transport Systems: New ICT – based Master's Curricula in Uzbekistan (INTRAS)
Agreement number: 2017-3516/001-001
Project reference number: 586292-EPP-1-2017-1-PL-EPPKA2-CBHE-JP

- **STEP 3: Formulation of the optimization constraint: "Binarization"**
  - ✓ This constraint insures, that each edge of the graph takes only two possible values, which are "0" or "1" (Binarization):

$$h(x_i)=\begin{cases} x_1(x_1-1)=0 \\ x_2(x_2-1)=0 \\ x_3(x_3-1)=0 \\ x_4(x_4-1)=0 \\ x_5(x_5-1)=0 \\ x_6(x_6-1)=0 \end{cases}$$

- **STEP 4: Expression of the Lagrange function as the total energy of the system**
  - ✓ The Lagrange function is obtained by combining the objective function with constraints. This combination is achieved using the multiplier variables λ_i.

$$L=(c_1x_1+c_2x_2+c_3x_3+c_4x_4+c_5x_5+c_6x_6)+\lambda_1(x_3+x_4-x_1-\delta_1)+\lambda_2(x_3+x_4+x_1-\delta_2)+\lambda_3(x_2+x_6-x_3-\delta_3)$$
$$+\lambda_4(x_2+x_6+x_3-\delta_4)+\lambda_5(x_1-x_5-x_2-\delta_5)+\lambda_6(x_1+x_5+x_2-\delta_6)+\lambda_7(x_5-x_6-x_4-\delta_7)+\lambda_8(x_5+x_6+x_4-\delta_8)$$
$$+\lambda_9x_1(x_1-1)+\lambda_{10}x_2(x_2-1)+\lambda_{11}x_3(x_3-1)+\lambda_{12}x_4(x_4-1)+\lambda_{13}x_5(x_5-1)+\lambda_{14}x_6(x_6-1)$$

- **STEP 5: Application of the BDMM concept and derivation of the coupled ODEs**
  - ✓ The gradient descent (see chapter 4) is applied to decision neurons/variables to obtain the set of coupled ODEs below:

$$\begin{cases} \dfrac{dx_1}{dt} = -\dfrac{\partial L}{\partial x_1} = -\left[c_1 - \lambda_1 + \lambda_2 + \lambda_5 + \lambda_6 + \lambda_9(2x_1 - 1)\right] \\ \dfrac{dx_2}{dt} = -\dfrac{\partial L}{\partial x_2} = -\left[c_2 + \lambda_3 + \lambda_4 - \lambda_5 + \lambda_6 + \lambda_{10}(2x_2 - 1)\right] \\ \dfrac{dx_3}{dt} = -\dfrac{\partial L}{\partial x_3} = -\left[c_3 + \lambda_1 + \lambda_2 - \lambda_3 + \lambda_4 + \lambda_{11}(2x_3 - 1)\right] \\ \dfrac{dx_4}{dt} = -\dfrac{\partial L}{\partial x_4} = -\left[c_4 + \lambda_1 + \lambda_2 - \lambda_7 + \lambda_8 + \lambda_{12}(2x_4 - 1)\right] \\ \dfrac{dx_5}{dt} = -\dfrac{\partial L}{\partial x_5} = -\left[c_5 - \lambda_5 + \lambda_6 + \lambda_7 + \lambda_8 + \lambda_{13}(2x_5 - 1)\right] \\ \dfrac{dx_6}{dt} = -\dfrac{\partial L}{\partial x_6} = -\left[c_6 + \lambda_3 + \lambda_4 - \lambda_7 + \lambda_8 + \lambda_{14}(2x_6 - 1)\right] \end{cases}$$

- **STEP 6: Application of the BDMM concept and derivation of the coupled ODEs**
  - ✓ The gradient ascent (see chapter 4) is applied to the **GROUP 1** of multiplier neurons/variables (ensuring the connectivity of nodes) to obtain the set of coupled ODEs below:

$$\begin{cases} \dfrac{d\lambda_1}{dt} = +\dfrac{\partial L}{\partial \lambda_1} = +\left[x_3 + x_4 - x_1 - \delta_1\right] \\ \dfrac{d\lambda_2}{dt} = +\dfrac{\partial L}{\partial \lambda_2} = +\left[x_3 + x_4 + x_1 - \delta_2\right] \\ \dfrac{d\lambda_3}{dt} = +\dfrac{\partial L}{\partial \lambda_3} = +\left[x_2 + x_6 - x_3 - \delta_3\right] \\ \dfrac{d\lambda_4}{dt} = +\dfrac{\partial L}{\partial \lambda_4} = +\left[x_2 + x_6 + x_3 - \delta_4\right] \end{cases}$$

$$\begin{cases} \dfrac{d\lambda_5}{dt} = +\dfrac{\partial L}{\partial \lambda_5} = +\left[x_1 - x_5 - x_2 - \delta_5\right] \\ \dfrac{d\lambda_6}{dt} = +\dfrac{\partial L}{\partial \lambda_6} = +\left[x_1 + x_5 + x_2 - \delta_6\right] \\ \dfrac{d\lambda_7}{dt} = +\dfrac{\partial L}{\partial \lambda_7} = +\left[x_5 - x_6 - x_4 - \delta_7\right] \\ \dfrac{d\lambda_8}{dt} = +\dfrac{\partial L}{\partial \lambda_8} = +\left[x_5 + x_6 + x_4 - \delta_8\right] \end{cases}$$

δ_2

- **STEP 7: Application of the BDMM concept and derivation of the coupled ODEs**
  - ✓ The gradient ascent (see chapter 4) is applied to the **GROUP 2** of multiplier neurons (ensuring the binarization) to obtain the set of coupled ODEs below:

$$\begin{cases} \frac{d\lambda_9}{dt} = +\frac{\partial L}{\partial \lambda_9} = +\left[x_1(x_1-1)\right] \\ \frac{d\lambda_{10}}{dt} = +\frac{\partial L}{\partial \lambda_{10}} = +\left[x_2(x_2-1)\right] \\ \frac{d\lambda_{11}}{dt} = +\frac{\partial L}{\partial \lambda_{11}} = +\left[x_3(x_3-1)\right] \\ \frac{d\lambda_{12}}{dt} = +\frac{\partial L}{\partial \lambda_{12}} = +\left[x_4(x_4-1)\right] \\ \frac{d\lambda_{13}}{dt} = +\frac{\partial L}{\partial \lambda_{13}} = +\left[x_5(x_5-1)\right] \\ \frac{d\lambda_{14}}{dt} = +\frac{\partial L}{\partial \lambda_{14}} = +\left[x_6(x_6-1)\right] \end{cases}$$

❑ STEP 8: Mathematical model of the Neuro-processor for the TSP

✓ The Mathematical model of the Neuro-processor is obtained as a combination of the set of equations obtained in STEPS 5, 6 and 7.

$$\begin{cases}
\frac{dx_1}{dt} = -\frac{\partial L}{\partial x_1} = -\left[c_1 - \lambda_1 + \lambda_2 + \lambda_5 + \lambda_6 + \lambda_9(2x_1 - 1)\right] \\
\frac{dx_2}{dt} = -\frac{\partial L}{\partial x_2} = -\left[c_2 + \lambda_3 + \lambda_4 - \lambda_5 + \lambda_6 + \lambda_{10}(2x_2 - 1)\right] \\
\frac{dx_3}{dt} = -\frac{\partial L}{\partial x_3} = -\left[c_3 + \lambda_1 + \lambda_2 - \lambda_3 + \lambda_4 + \lambda_{11}(2x_3 - 1)\right] \\
\frac{dx_4}{dt} = -\frac{\partial L}{\partial x_4} = -\left[c_4 + \lambda_1 + \lambda_2 - \lambda_7 + \lambda_8 + \lambda_{12}(2x_4 - 1)\right] \\
\frac{dx_5}{dt} = -\frac{\partial L}{\partial x_5} = -\left[c_5 - \lambda_5 + \lambda_6 + \lambda_7 + \lambda_8 + \lambda_{13}(2x_5 - 1)\right] \\
\frac{dx_6}{dt} = -\frac{\partial L}{\partial x_6} = -\left[c_6 + \lambda_3 + \lambda_4 - \lambda_7 + \lambda_8 + \lambda_{14}(2x_6 - 1)\right]
\end{cases}$$

$$\begin{cases}
\frac{d\lambda_1}{dt} = +\frac{\partial L}{\partial \lambda_1} = +\left[x_3 + x_4 - x_1 - \delta_1\right] \\
\frac{d\lambda_2}{dt} = +\frac{\partial L}{\partial \lambda_2} = +\left[x_3 + x_4 + x_1 - \delta_2\right] \\
\frac{d\lambda_3}{dt} = +\frac{\partial L}{\partial \lambda_3} = +\left[x_2 + x_6 - x_3 - \delta_3\right] \\
\frac{d\lambda_4}{dt} = +\frac{\partial L}{\partial \lambda_4} = +\left[x_2 + x_6 + x_3 - \delta_4\right] \\
\frac{d\lambda_5}{dt} = +\frac{\partial L}{\partial \lambda_5} = +\left[x_1 - x_5 - x_2 - \delta_5\right] \\
\frac{d\lambda_6}{dt} = +\frac{\partial L}{\partial \lambda_6} = +\left[x_1 + x_5 + x_2 - \delta_6\right] \\
\frac{d\lambda_7}{dt} = +\frac{\partial L}{\partial \lambda_7} = +\left[x_5 - x_6 - x_4 - \delta_7\right] \\
\frac{d\lambda_8}{dt} = +\frac{\partial L}{\partial \lambda_8} = +\left[x_5 + x_6 + x_4 - \delta_8\right]
\end{cases}$$

$$\begin{cases}
\frac{d\lambda_9}{dt} = +\frac{\partial L}{\partial \lambda_9} = +\left[x_1(x_1 - 1)\right] \\
\frac{d\lambda_{10}}{dt} = +\frac{\partial L}{\partial \lambda_{10}} = +\left[x_2(x_2 - 1)\right] \\
\frac{d\lambda_{11}}{dt} = +\frac{\partial L}{\partial \lambda_{11}} = +\left[x_3(x_3 - 1)\right] \\
\frac{d\lambda_{12}}{dt} = +\frac{\partial L}{\partial \lambda_{12}} = +\left[x_4(x_4 - 1)\right] \\
\frac{d\lambda_{13}}{dt} = +\frac{\partial L}{\partial \lambda_{13}} = +\left[x_5(x_5 - 1)\right] \\
\frac{d\lambda_{14}}{dt} = +\frac{\partial L}{\partial \lambda_{14}} = +\left[x_6(x_6 - 1)\right]
\end{cases}$$

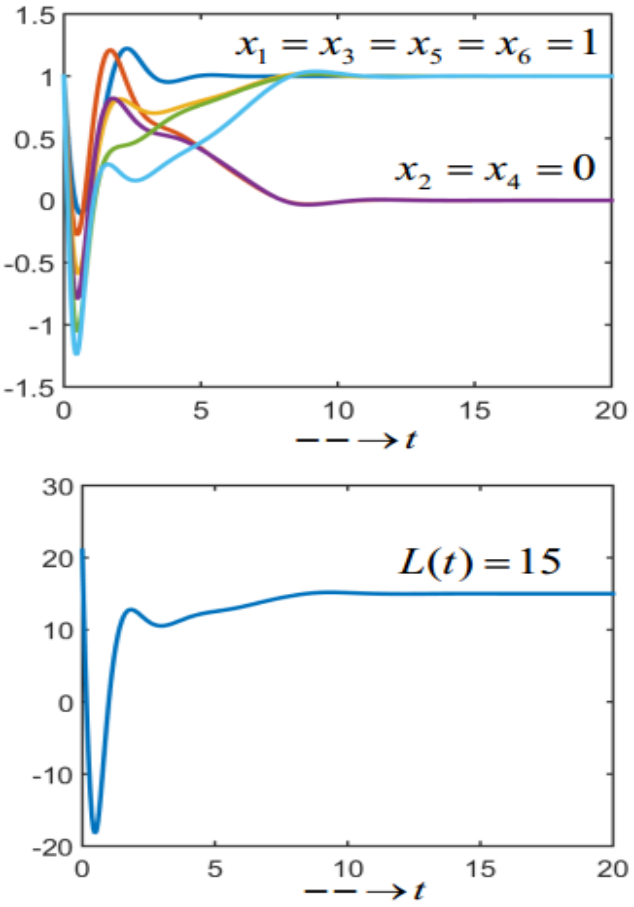
5.15. MATLAB-CODING of the mathematical model on the Neuro-processor for TSP

- **MATLAB-code of the mathematical model of the Neuro-processor for solving TSP**

```
function dx=f(t,x)
dx=zeros(20,1);

c1=1; c2=2; c3=3; c4=4; c5=5; c6=6;

s1=0; s2=2;   s3=0; s4=2;   s5=0; s6=2;   s7=0; s8=2;

dx(1)=-( c1 - x(7) + x(8) + x(11)+ x(12)+ x(15)*(2*x(1)-1) );
dx(2)=-( c2 + x(9) + x(10) - x(11)+ x(12) + x(16)*(2*x(2)-1) );
dx(3)=-( c3 + x(7) + x(8) - x(9)+ x(10)+ x(17)*(2*x(3)-1) );
dx(4)=-( c4 + x(7) + x(8) - x(13)+ x(14) + x(18)*(2*x(4)-1) );
dx(5)=-( c5 - x(11) + x(12) + x(13)+ x(14)+ x(19)*(2*x(5)-1));
dx(6)=-( c6 + x(9) + x(10) - x(13)+ x(14)+ x(20)*(2*x(6)-1) );

dx(7)=+( x(3) + x(4) - x(1) - s1 );
dx(8)=+( x(3) + x(4) + x(1) - s2 );

dx(9)=+( x(6) + x(2) - x(3) - s3 );
dx(10)=+( x(6) + x(2) + x(3) - s4 );

dx(11)=+( x(1) - x(2) - x(5) - s5 );
dx(12)=+( x(1) + x(2) + x(5) - s6 );

dx(13)=+( x(5) - x(4) - x(6) - s7 );
dx(14)=+( x(5) + x(4) + x(6) - s8 );

dx(15)=+x(1)*(x(1)-1);
dx(16)=+x(2)*(x(2)-1);
dx(17)=+x(3)*(x(3)-1);
dx(18)=+x(4)*(x(4)-1);
dx(19)=+x(5)*(x(5)-1);
dx(20)=+x(6)*(x(6)-1);
end
```

5.16. Results of the numerical computation of the TSP using the MATLAB-code above

- **Numerical results obtained by the MATLAB-code:**

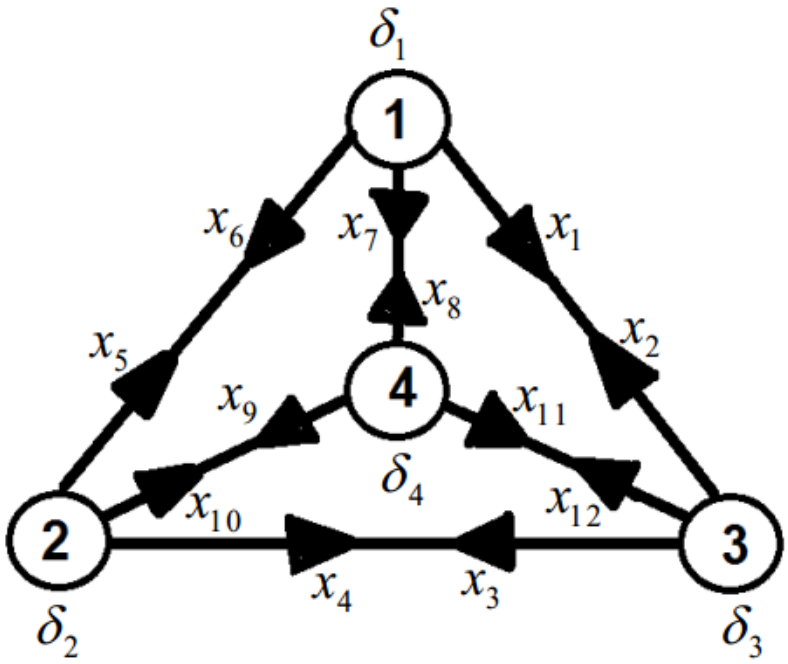



```
[t,x]=ode45(@TSP1, [0 20], [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ]);
```

5. 17. Mathematical modeling of the Traveling Salesman problem (TSP) in an undirected graph of magnitude 4 and size 12

- **Case study 1: A simple undirected graph of magnitude 4 and size 12 (see below)**

- **TASK: Mathematical modelling of the TSP using the theory of optimization in chap. 4**
  - ✓ In this section we use the knowledge acquired in chapter 4 to model the TSP (in the graph below) mathematically.



Intelligent Transport Systems: New ICT – based Master's Curricula in Uzbekistan (INTRAS)
Agreement number: 2017-3516/001-001
Project reference number: 586292-EPP-1-2017-1-PL-EPPKA2-CBHE-JP

- **STEP 1: Expression of the objective function.**
  - ✓ This function is expressed as the minimization of the total cost of the graph.

$$Min\left[f(x_i,c_i)=c_1x_1+c_2x_2+c_3x_3+c_4x_4+c_5x_5+c_6x_6 + c_7x_7+c_8x_8+c_9x_9+c_{10}x_{10}+c_{11}x_{11}+c_{12}x_{12}\right]$$

- **STEP 2: Formulation of the optimization constraints: "Connectivity of nodes"**
  - ✓ Constraints are obtained through the assignment of a unique attribute to all nodes of the graph. The attribute of intermediate node is assigned to each node of the graph. This ensure the belonging of all nodes to the TSP solution.

$$g(x_i,\delta_i)=\begin{cases}\text{Node1}: \left(x_1+x_6+x_7-x_2-x_5-x_8\right)=\delta_1 \cap \left(x_1+x_6+x_7+x_2+x_5+x_8\right)=\delta_2\\ \text{Node2}: \left(x_4+x_5+x_{10}-x_3-x_6-x_9\right)=\delta_3 \cap \left(x_4+x_5+x_{10}+x_3+x_6+x_9\right)=\delta_4\\ \text{Node3}: \left(x_2+x_3+x_{12}-x_1-x_4-x_{11}\right)=\delta_5 \cap \left(x_2+x_3+x_{12}+x_1+x_4+x_{11}\right)=\delta_6\\ \text{Node4}: \left(x_8+x_9+x_{11}-x_7-x_{10}-x_{12}\right)=\delta_7 \cap \left(x_8+x_9+x_{11}+x_7+x_{10}+x_{12}\right)=\delta_8\end{cases} \qquad \begin{cases}\delta_i=0 \ (i:\text{odd})\\ \delta_i=2 \ (i:\text{even})\end{cases}$$

- **STEP 3: Formulation of the optimization constraint: "Binarization"**
  - ✓ This constraint insures, that each edge of the graph takes only two possible values, which are "0" or "1" (Binarization):

$$h(x_i)=\begin{cases} x_1(x_1-1)=0 \\ x_2(x_2-1)=0 \\ x_3(x_3-1)=0 \\ x_4(x_4-1)=0 \\ x_5(x_5-1)=0 \\ x_6(x_6-1)=0 \end{cases} \qquad h(x_i)=\begin{cases} x_7(x_7-1)=0 \\ x_8(x_8-1)=0 \\ x_9(x_9-1)=0 \\ x_{10}(x_{10}-1)=0 \\ x_{11}(x_{11}-1)=0 \\ x_{12}(x_{12}-1)=0 \end{cases}$$

- **STEP 4: Expression of the Lagrange function as the total energy of the system**
  - ✓ The Lagrange function is obtained by combining the objective function with constraints. This combination is achieved using the multiplier variables λ_i.

$$L=(c_1x_1+c_2x_2+c_3x_3+c_4x_4+c_5x_5+c_6x_6+c_7x_7+c_8x_8+c_9x_9+c_{10}x_{10}+c_{11}x_{11}+c_{12}x_{12})+\lambda_1(x_1+x_6+x_7-x_2-x_5-x_8-\delta_1)$$
$$+\lambda_2(x_1+x_6+x_7+x_2+x_5+x_8-\delta_2)+\lambda_3(x_4+x_5+x_{10}-x_3-x_6-x_9-\delta_3)+\lambda_4(x_4+x_5+x_{10}+x_3+x_6+x_9-\delta_4)$$
$$+\lambda_5(x_2+x_3+x_{12}-x_1-x_4-x_{11}-\delta_5)+\lambda_6(x_2+x_3+x_{12}+x_1+x_4+x_{11}-\delta_6)+\lambda_7(x_8+x_9+x_{11}-x_7-x_{10}-x_{12}-\delta_7)$$
$$+\lambda_8(x_8+x_9+x_{11}+x_7+x_{10}+x_{12}-\delta_8)+\lambda_9x_1(x_1-1)+\lambda_{10}x_2(x_2-1)+\lambda_{11}x_3(x_3-1)+\lambda_{12}x_4(x_4-1)+\lambda_{13}x_5(x_5-1)$$
$$+\lambda_{14}x_6(x_6-1)+\lambda_{15}x_7(x_7-1)+\lambda_{16}x_8(x_8-1)+\lambda_{17}x_9(x_9-1)+\lambda_{18}x_{10}(x_{10}-1)+\lambda_{19}x_{11}(x_{11}-1)+\lambda_{20}x_{12}(x_{12}-1)$$

- **STEP 5: Application of the BDMM concept and derivation of the coupled ODEs**
  - ✓ The gradient descent (see chapter 4) is applied to decision neurons/variables to obtain the set of coupled ODEs below:

$$
\begin{cases}
\dot{x}_1 = -\left[c_1 + x_{13} + x_{14} - x_{17} + x_{18} + x_{21}(2x_1 - 1)\right] \\
\dot{x}_2 = -\left[c_2 - x_{13} + x_{14} + x_{17} + x_{18} + x_{22}(2x_2 - 1)\right] \\
\dot{x}_3 = -\left[c_3 - x_{15} + x_{16} + x_{17} + x_{18} + x_{23}(2x_3 - 1)\right] \\
\dot{x}_4 = -\left[c_4 + x_{15} + x_{16} - x_{17} + x_{18} + x_{24}(2x_4 - 1)\right] \\
\dot{x}_5 = -\left[c_5 - x_{13} + x_{14} + x_{15} + x_{16} + x_{25}(2x_5 - 1)\right] \\
\dot{x}_6 = -\left[c_6 + x_{13} + x_{14} - x_{15} + x_{16} + x_{26}(2x_6 - 1)\right] \\
\dot{x}_7 = -\left[c_7 + x_{13} + x_{14} - x_{19} + x_{20} + x_{27}(2x_7 - 1)\right] \\
\dot{x}_8 = -\left[c_8 - x_{13} + x_{14} + x_{19} + x_{20} + x_{28}(2x_8 - 1)\right] \\
\dot{x}_9 = -\left[c_9 - x_{15} + x_{16} + x_{19} + x_{20} + x_{29}(2x_9 - 1)\right] \\
\dot{x}_{10} = -\left[c_{10} + x_{15} + x_{16} - x_{19} + x_{20} + x_{30}(2x_{10} - 1)\right] \\
\dot{x}_{11} = -\left[c_{11} - x_{17} + x_{18} + x_{19} + x_{20} + x_{31}(2x_{11} - 1)\right] \\
\dot{x}_{12} = -\left[c_{12} + x_{17} + x_{18} - x_{19} + x_{20} + x_{32}(2x_{12} - 1)\right]
\end{cases}
$$

- **STEP 6: Application of the BDMM concept and derivation of the coupled ODEs**
  - ✓ The gradient ascent (see chapter 4) is applied to the **GROUP 1** of multiplier neurons/variables (ensuring the connectivity of nodes) to obtain the set of coupled ODEs below:

$$
\begin{cases}
\dot{x}_{13} = \dot{\lambda}_1 = +\left[x_1 + x_6 + x_7 - x_2 - x_5 - x_8 - \delta_1\right] \\
\dot{x}_{14} = \dot{\lambda}_2 = +\left[x_1 + x_6 + x_7 + x_2 + x_5 + x_8 - \delta_2\right] \\
\dot{x}_{15} = \dot{\lambda}_3 = +\left[x_4 + x_5 + x_{10} - x_3 - x_6 - x_9 - \delta_3\right] \\
\dot{x}_{16} = \dot{\lambda}_4 = +\left[x_4 + x_5 + x_{10} + x_3 + x_6 + x_9 - \delta_4\right] \\
\dot{x}_{17} = \dot{\lambda}_5 = +\left[x_2 + x_3 + x_{12} - x_1 - x_4 - x_{11} - \delta_5\right] \\
\dot{x}_{18} = \dot{\lambda}_6 = +\left[x_2 + x_3 + x_{12} + x_1 + x_4 + x_{11} - \delta_6\right] \\
\dot{x}_{19} = \dot{\lambda}_7 = +\left[x_8 + x_9 + x_{11} - x_7 - x_{10} - x_{12} - \delta_7\right] \\
\dot{x}_{20} = \dot{\lambda}_8 = +\left[x_8 + x_9 + x_{11} + x_7 + x_{10} + x_{12} - \delta_8\right]
\end{cases}
$$

- **STEP 7: Application of the BDMM concept and derivation of the coupled ODEs**
  - ✓ The gradient ascent (see chapter 4) is applied to the **GROUP 2** of multiplier neurons (ensuring the binarization) to obtain the set of coupled ODEs below:

$$\begin{cases} \dot{x}_{21} = \dot{\lambda}_9 = +\left[x_1\left(x_1 - 1\right)\right] \\ \dot{x}_{22} = \dot{\lambda}_{10} = +\left[x_2\left(x_2 - 1\right)\right] \\ \dot{x}_{23} = \dot{\lambda}_{11} = +\left[x_3\left(x_3 - 1\right)\right] \\ \dot{x}_{24} = \dot{\lambda}_{12} = +\left[x_4\left(x_4 - 1\right)\right] \\ \dot{x}_{25} = \dot{\lambda}_{13} = +\left[x_5\left(x_5 - 1\right)\right] \\ \dot{x}_{26} = \dot{\lambda}_{14} = +\left[x_6\left(x_6 - 1\right)\right] \end{cases} \qquad \begin{cases} \dot{x}_{27} = \dot{\lambda}_{15} = +\left[x_7\left(x_7 - 1\right)\right] \\ \dot{x}_{28} = \dot{\lambda}_{16} = +\left[x_8\left(x_8 - 1\right)\right] \\ \dot{x}_{29} = \dot{\lambda}_{17} = +\left[x_9\left(x_9 - 1\right)\right] \\ \dot{x}_{30} = \dot{\lambda}_{18} = +\left[x_{10}\left(x_{10} - 1\right)\right] \\ \dot{x}_{31} = \dot{\lambda}_{19} = +\left[x_{11}\left(x_{11} - 1\right)\right] \\ \dot{x}_{32} = \dot{\lambda}_{20} = +\left[x_{12}\left(x_{12} - 1\right)\right] \end{cases}$$

❑ STEP 8: Mathematical model of the Neuro-processor for the TSP

✓ The Mathematical model of the Neuro-processor is easily obtained by combining the set of equations obtained in STEPS 5, 6 and 7.

$$\begin{cases}
\dot{x}_1 = -\left[c_1 + x_{13} + x_{14} - x_{17} + x_{18} + x_{21}(2x_1 - 1)\right] \\
\dot{x}_2 = -\left[c_2 - x_{13} + x_{14} + x_{17} + x_{18} + x_{22}(2x_2 - 1)\right] \\
\dot{x}_3 = -\left[c_3 - x_{15} + x_{16} + x_{17} + x_{18} + x_{23}(2x_3 - 1)\right] \\
\dot{x}_4 = -\left[c_4 + x_{15} + x_{16} - x_{17} + x_{18} + x_{24}(2x_4 - 1)\right] \\
\dot{x}_5 = -\left[c_5 - x_{13} + x_{14} + x_{15} + x_{16} + x_{25}(2x_5 - 1)\right] \\
\dot{x}_6 = -\left[c_6 + x_{13} + x_{14} - x_{15} + x_{16} + x_{26}(2x_6 - 1)\right] \\
\dot{x}_7 = -\left[c_7 + x_{13} + x_{14} - x_{19} + x_{20} + x_{27}(2x_7 - 1)\right] \\
\dot{x}_8 = -\left[c_8 - x_{13} + x_{14} + x_{19} + x_{20} + x_{28}(2x_8 - 1)\right] \\
\dot{x}_9 = -\left[c_9 - x_{15} + x_{16} + x_{19} + x_{20} + x_{29}(2x_9 - 1)\right] \\
\dot{x}_{10} = -\left[c_{10} + x_{15} + x_{16} - x_{19} + x_{20} + x_{30}(2x_{10} - 1)\right] \\
\dot{x}_{11} = -\left[c_{11} - x_{17} + x_{18} + x_{19} + x_{20} + x_{31}(2x_{11} - 1)\right] \\
\dot{x}_{12} = -\left[c_{12} + x_{17} + x_{18} - x_{19} + x_{20} + x_{32}(2x_{12} - 1)\right]
\end{cases}$$

$$\begin{cases}
\dot{x}_{21} = \dot{\lambda}_9 = +\left[x_1(x_1 - 1)\right] \\
\dot{x}_{22} = \dot{\lambda}_{10} = +\left[x_2(x_2 - 1)\right] \\
\dot{x}_{23} = \dot{\lambda}_{11} = +\left[x_3(x_3 - 1)\right] \\
\dot{x}_{24} = \dot{\lambda}_{12} = +\left[x_4(x_4 - 1)\right] \\
\dot{x}_{25} = \dot{\lambda}_{13} = +\left[x_5(x_5 - 1)\right] \\
\dot{x}_{26} = \dot{\lambda}_{14} = +\left[x_6(x_6 - 1)\right]
\end{cases}
\qquad
\begin{cases}
\dot{x}_{27} = \dot{\lambda}_{15} = +\left[x_7(x_7 - 1)\right] \\
\dot{x}_{28} = \dot{\lambda}_{16} = +\left[x_8(x_8 - 1)\right] \\
\dot{x}_{29} = \dot{\lambda}_{17} = +\left[x_9(x_9 - 1)\right] \\
\dot{x}_{30} = \dot{\lambda}_{18} = +\left[x_{10}(x_{10} - 1)\right] \\
\dot{x}_{31} = \dot{\lambda}_{19} = +\left[x_{11}(x_{11} - 1)\right] \\
\dot{x}_{32} = \dot{\lambda}_{20} = +\left[x_{12}(x_{12} - 1)\right]
\end{cases}$$

$$\begin{cases}
\dot{x}_{13} = \dot{\lambda}_1 = +\left[x_1 + x_6 + x_7 - x_2 - x_5 - x_8 - \delta_1\right] \\
\dot{x}_{14} = \dot{\lambda}_2 = +\left[x_1 + x_6 + x_7 + x_2 + x_5 + x_8 - \delta_2\right] \\
\dot{x}_{15} = \dot{\lambda}_3 = +\left[x_4 + x_5 + x_{10} - x_3 - x_6 - x_9 - \delta_3\right] \\
\dot{x}_{16} = \dot{\lambda}_4 = +\left[x_4 + x_5 + x_{10} + x_3 + x_6 + x_9 - \delta_4\right] \\
\dot{x}_{17} = \dot{\lambda}_5 = +\left[x_2 + x_3 + x_{12} - x_1 - x_4 - x_{11} - \delta_5\right] \\
\dot{x}_{18} = \dot{\lambda}_6 = +\left[x_2 + x_3 + x_{12} + x_1 + x_4 + x_{11} - \delta_6\right] \\
\dot{x}_{19} = \dot{\lambda}_7 = +\left[x_8 + x_9 + x_{11} - x_7 - x_{10} - x_{12} - \delta_7\right] \\
\dot{x}_{20} = \dot{\lambda}_8 = +\left[x_8 + x_9 + x_{11} + x_7 + x_{10} + x_{12} - \delta_8\right]
\end{cases}$$

Intelligent Transport Systems: New ICT – based Master's Curricula in Uzbekistan (INTRAS)
Agreement number: 2017-3516/001-001
Project reference number: 586292-EPP-1-2017-1-PL-EPPKA2-CBHE-JP

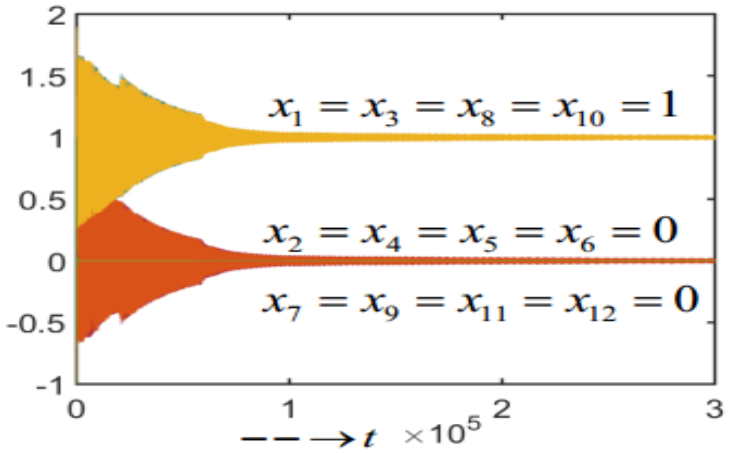
5.18. MATLAB-CODING of the mathematical model on the Neuro-processor for TSP

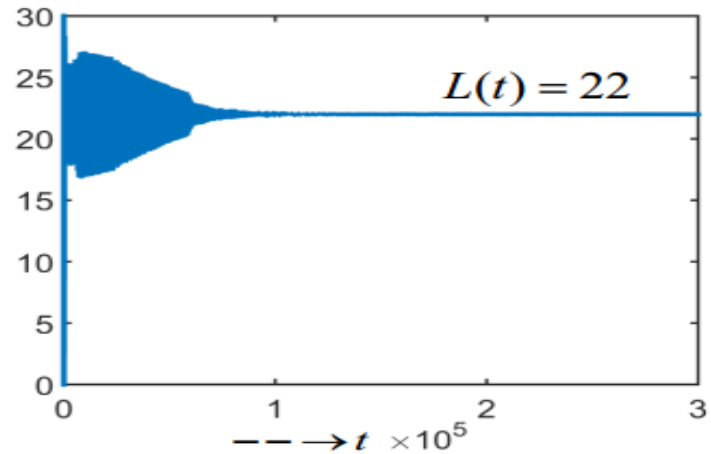
- **MATLAB-code of the mathematical model of the Neuro-processor for solving TSP**

```
function dx=f(t,x)
dx=zeros(32,1);
c1=1; c2=2; c3=3; c4=4; c5=5; c6=6; c7=7; c8=8; c9=9; c10=10;
c11=11; c12=12; s1=0; s2=2; s3=0; s4=2; s5=0; s6=2; s7=0; s8=2;
dx(1)=-( c1 + x(13) + x(14) - x(17) + x(18) + x(21)*(2*x(1)-1) );
dx(2)=-( c2 - x(13) + x(14) + x(17) + x(18) + x(22)*(2*x(2)-1) );
dx(3)=-( c3 - x(15) + x(16) + x(17) + x(18) + x(23)*(2*x(3)-1) );
dx(4)=-( c4 + x(15) + x(16) - x(17) + x(18) + x(24)*(2*x(4)-1) );
dx(5)=-( c5 - x(13) + x(14) + x(15) + x(16) + x(25)*(2*x(5)-1) );
dx(6)=-( c6 + x(13) + x(14) - x(15) + x(16) + x(26)*(2*x(6)-1) );
dx(7)=-( c7 + x(13) + x(14) - x(19) + x(20) + x(27)*(2*x(7)-1) );
dx(8)=-( c8 - x(13) + x(14) + x(19) + x(20) + x(28)*(2*x(8)-1) );
dx(9)=-( c9 - x(15) + x(16) + x(19) + x(20) + x(29)*(2*x(9)-1) );
dx(10)=-( c10 + x(15)+ x(16)-x(19)+ x(20) + x(30)*(2*x(10)-1) );
dx(11)=-( c11 - x(17)+ x(18)+x(19)+ x(20) + x(31)*(2*x(11)-1) );
dx(12)=-( c12 + x(17)+ x(18)-x(19)+ x(20) + x(32)*(2*x(12)-1) );
dx(13)=+( x(1) + x(6) + x(7) - x(2) - x(5) - x(8) - s1 );
dx(14)=+( x(1) + x(6) + x(7) + x(2) + x(5) + x(8) - s2 );
dx(15)=+( x(4) + x(5) + x(10) - x(3) - x(6) - x(9) - s3 );
dx(16)=+( x(4) + x(5) + x(10) + x(3) + x(6) + x(9) - s4 );
dx(17)=+( x(2) + x(3) + x(12) - x(1) - x(4) - x(11) - s5 );
dx(18)=+( x(2) + x(3) + x(12) + x(1) + x(4) + x(11) - s6 );
dx(19)=+( x(8) + x(9) + x(11) - x(7) - x(10) - x(12) - s7 );
dx(20)=+( x(8) + x(9) + x(11) + x(7) + x(10) + x(12) - s8 );
dx(21)=+x(1)*(x(1)-1);
dx(22)=+x(2)*(x(2)-1);
dx(23)=+x(3)*(x(3)-1);
dx(24)=+x(4)*(x(4)-1);
dx(25)=+x(5)*(x(5)-1);
dx(26)=+x(6)*(x(6)-1);
dx(27)=+x(7)*(x(7)-1);
dx(28)=+x(8)*(x(8)-1);
dx(29)=+x(9)*(x(9)-1);
dx(30)=+x(10)*(x(10)-1);
dx(31)=+x(11)*(x(11)-1);
dx(32)=+x(12)*(x(12)-1);
end
```

5.19. Results of the numerical computation of the TSP using the MATLAB-code above

- **Numerical results obtained by the MATLAB-code:**






```
[t,x]=ode45(@TSP2, [0 300000], [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ]);
```

INTRAS

Co-funded by the
Erasmus+ Programme
of the European Union

Chapter 6.

Optimization in transportation

(Chapter's detailed description)

Intelligent Transport Systems: New ICT – based Master's Curricula in Uzbekistan (INTRAS)
Agreement number: 2017-3516/001-001
Project reference number: 586292-EPP-1-2017-1-PL-EPPKA2-CBHE-JP

6.1. Mathematical modeling of the traffic signal control principle at isolated (local) traffic junction: Traffic signals splitting, Green signal sharing, etc.

- ✓ This section explain how the traffic signal control strategy can be modeled using the mathematical models of nonlinear oscillators. The signal control strategy developed for a single traffic junction (also called isolated traffic junction) is based on the Cellular Neural Networks (CNN) paradigm. Overall, this section is based on three main points:
  - Presentation of the paradigm of Cellular Neural Networks (CNN)
  - Design of a signal control strategy based on CNN.
  - Case-study. Solving a concrete example of a known isolated junction.

6.2. Mathematical modeling of the traffic signal control principle in a network of coupled traffic junctions: Traffic signals splitting, Green signal sharing, etc.

- ✓ This section explain how the traffic signal control strategy (in a network of coupled traffic junctions) can be modelled using the KURAMOTO model. Overall, this section is based on three main points:
  - Presentation of the paradigm of KURAMOTO oscillator
  - Design of a signal control strategy based on KURAMOTO.
  - Case-study. Solving a concrete example of a known isolated junction.

INTRAS

Co-funded by the
Erasmus+ Programme
of the European Union

Chapter 7.

Optimization in railway transportation

(Chapter's detailed description)

Intelligent Transport Systems: New ICT – based Master's Curricula in Uzbekistan (INTRAS)
Agreement number: 2017-3516/001-001
Project reference number: 586292-EPP-1-2017-1-PL-EPPKA2-CBHE-JP

7.1. Mathematical modelling of the train dynamics and optimization of the Energy consumption

- ✓ This section studies a mathematical model for the modeling of the motion of a train (rail-cars). The control/optimization of energy consumption is also investigated.

7.2. Mathematical modeling and optimization of the railway blocking problem

- ✓ This section studies a mathematical system for the modeling of the well-known railway blocking problem. The control of the railway blocking phenomenon is further investigated through tuning of the parameters of the mathematical model.

7.3. Optimization of the train trajectory: Modelling concept and simulation algorithm

- ✓ This section presents two mathematical models for the investigation of the trajectory of a train. The first model is the case of traffic without delay and the second model considers the case of traffic with delay. In both cases, the dynamics of the train is further investigated through numerical simulation of the mathematical models. Three main aspects are investigated, namely the effects of traffic lights/signals, the signal response policy, and the green wave policy.

7.4. Extension of the study in section 7.3 to the derivation of the "Optimization model"

- ✓ This section extends the study in section 7.3 to the case of two-train trajectory.

INTRAS

Co-funded by the
Erasmus+ Programme
of the European Union

Chapter 8.

Supply chain networks: Modelling and Analysis of the Dynamics of Supply Chain Networks

(Chapter's detailed description)

Intelligent Transport Systems: New ICT – based Master's Curricula in Uzbekistan (INTRAS)
Agreement number: 2017-3516/001-001
Project reference number: 586292-EPP-1-2017-1-PL-EPPKA2-CBHE-JP

8.1. Overview of supply chain networks

- ✓ This section describes two selected concrete examples of supply chain networks. The cooperation within the supply chain network is further discussed.

8.2. Modelling principle of supply chain networks

- ✓ This section explains the important steps in modeling supply chain networks. The pros and cons of modeling are also presented.

8.3. Modelling and optimization of the assignment problem in a supply chain network

- ✓ The well-known “assignment problem in a supply chain network” is modeled in this section. The problem is further optimized using the classical constrained optimization technique that consists of deriving the objective function with related constraints.

8.4. Supply Chain Management Optimization Problem

- ✓ The strategy of managing supply chain networks is discussed. In particular, the collaboration (e.g., information exchange) within supply chain networks is discussed and the effects on the ordering, supply and/or delivery of Goods are discussed. Finally the optimization of the management policy within supply chain networks is investigated. Some concluding remarks are formulated w.r.t the management policy.

INTRAS

Co-funded by the
Erasmus+ Programme
of the European Union

Chapter 9.

Scheduling. Fundamentals of Scheduling and Applications in Transportation

(Chapter's detailed description)

Intelligent Transport Systems: New ICT – based Master's Curricula in Uzbekistan (INTRAS)
Agreement number: 2017-3516/001-001
Project reference number: 586292-EPP-1-2017-1-PL-EPPKA2-CBHE-JP

9.1. Overview of scheduling

- ✓ This section describes the scheduling problem and discusses challenges related to scheduling. Some examples of scheduling (in transportation) are presented.

9.2. Principles of scheduling

- ✓ The steps of the modeling of scheduling problems are presented and are discussed. The modeling of job-shop scheduling problems is considered as application example.

9.3. Railway scheduling by network optimization problem

- ✓ This section presents optimization algorithms/methods to solve problems involving allocation of shared resources (e.g., sections of railway track).

9.4. Modeling of the railway scheduling problem and time-tables optimization

- ✓ The train scheduling problem is considered. This problem involves the assignment of trains to a group/set of trips generated by a train timetable. The description and optimization of this problem are considered in this section.

9.5. Modeling and optimization of the Crew scheduling problem in Railway transportation

- ✓ The crew scheduling problem consist of assigning crews to trains, which operate at a given schedule. This section considers the modeling and optimization of the problem.

INTRAS

Co-funded by the
Erasmus+ Programme
of the European Union

Intelligent Transport Systems: New ICT – based Master's Curricula in Uzbekistan (INTRAS)
Agreement number: 2017-3516/001-001
Project reference number: 586292-EPP-1-2017-1-PL-EPPKA2-CBHE-JP