`Transport and Telecommunication, 2011, Volume 12, No 3, 27–40
Transport and Telecommunication Institute, Lomonosova 1, Riga, LV-1019, Latvia

CRITICAL SITUATIONS EVALUATION OF DISCRETE TRANSPORTATION SYSTEMS

Jacek Mazurkiewicz, Tomasz Walkowiak

Institute of Computer Engineering, Control and Robotics
Wroclaw University of Technology
ul. Janiszewskiego 11/17, 50-372 Wroclaw, Poland
E-mails: Jacek.Mazurkiewicz@pwr.wroc.pl, Tomasz.Walkowiak@pwr.wroc.pl

The paper presents the formal model of discrete transportation systems (*DTS*). The modelling methodology is based on the system functional behaviour. Monte Carlo approach is used as a simulation tool. The actual *DTS* situation is measured by the global metric called availability. The proposed solution is very useful for the system owner and manager. The critical situations are caused by reliability, functional and human origin. Absence of restrictions on the system structure and on the kind of distribution describing the system functional and reliability parameters is the main advantage of the approach. The exemplar *DTS* driven into critical situation is presented in the final part. The results show and discuss different ways to restore the normal operating point. The proposed solution seems to be essential for the owner and administrator of the transportation systems.

Keywords: reliability, discrete transportation system, Monte-Carlo simulation, critical sets

1. Introduction

The transportation systems are characterized by a very complex structure. The performance of the system can be impaired by various types of faults related to the transportation vehicles, communication infrastructure or even by traffic congestion or human resource [1]. Each part of the system is characterised by absolutely unique set of features and can cause the critical situation of the whole system if it starts to work in unusual way or the fault or error of it is noticed. It is hard for an administrator, manager or an owner to understand the system behaviour and to combine the large scale of variant states of it in single – easily observable and controlled global metric as a pointer to make the proper decision in a short time period. To overcome this problem we propose a functional approach.

The transportation system is analysed from the functional point of view, focusing on business service implemented by a system [16]. The analysis is following a classical one [4]: modelling and simulation approach. It allows calculating different system measures which can be a base for the decisions related to administration of the transportation systems.

The metric are calculated using Monte Carlo techniques [7]. Absence of the restrictions on the system structure and on a kind of distribution is the main advantage of the method. Such approach allows – forgetting about the classical reliability analysis based on Markov or Semi-Markov processes [2] idealised and hard for reconciliation in practice. The results of the system observation is understand the set of data collected during the simulation process; they are the basis to define the critical situations and they allow providing the probably proper solution to lift-up the systems in effective way if the critical situation occurs. This is the only sensible way, because the critical situations are the real and not removable part of the system life.

Section 2 presents a brief environmental transport systems discussion. The real system – a source and motivation for the work is described in section 3. The developed discrete transportation system model is presented in section 4. The main service given by the post system is the delivery of mails. From the client point of view the quality of the system can be measured by the time of transporting the mail from the source to the destination. A driver is a new element of the system description. We pointed the set of states to characterise the actual driver position including formal law-origin aspects: number of hours he or she can work daily for example. We offer the heuristic approach to the system management (section 5). In our opinion it seems to be the most adequate to the level of detail to provide the well-established description of the critical situation. Next we show the idea, logic and implementation aspects of the prepared and used simulator (section 6).

The quality of the analysed system is measured by the availability defined as an ability to realize the transportation task at a required time (described in section 7). The section defines the critical situation matter and the most important features. Next (section 8), we give an example of using the presented model for the analysis of the Dolny Slask Polish Post regional transportation system.

2. Environmental Transport Systems Discussion

2.1. Traffic Problem

Modelling traffic flow for design, planning and management of transportation systems in urban and highway area has been addressed to since the 1950s mostly by the civil engineering community. The following definitions and concepts of traffic simulation modelling can be found in works such as Gartner et al. [8]. Depending on the level of detail in modelling the granularity of traffic flow, traffic models are broadly divided into two categories: macroscopic and microscopic models. According to Gartner et al. [8], a macroscopic model describes the traffic flow as a fluid process with aggregate variables, such as flow and density. The state of the system is then simulated using the analytical relationships between average variables such as traffic density, traffic volume, and average speed.

On the other hand, a microscopic model reproduces interaction of punctual elements (vehicles, road segments, intersections, etc) in the traffic network. Each vehicle in the system is emulated according to its individual characteristics (length, speed, acceleration, etc.). Traffic is then simulated, using processing logic and models describing vehicle driving behaviour, such as car-following and lane-changing models. These models reproduce the driver-driver and driver-road interactions. Despite its great accuracy level, for many years this highly detailed modelling has been considered a computationally intensive approach. For the last twenty years, with the improvements in processing speed, this microscopic approach has become more attractive. In fact, Ben-Akiva et al. [3], Barcelo et al. [1] and Liu et al. [12] claim that using microscopic approach is essential to track the real-time traffic state and then, to define strategy to decrease congestion in urban transportation networks. For the control of congestion, they explain that the models must accurately capture the full dynamics of time dependant traffic phenomena and must also track vehicles reactions when exposed to Intelligent Transportation Systems (ITS).

From the latter assertions, in order to control the traffic congestion in internal transportation networks it appears that the microscopic modelling will be more appropriate. A common definition of congestion is the apparition of a delay above the minimum travel time needed to traverse a transportation network. As stated in Taylor et al. [14], this notion is context-specific and complex because a delay may always appear in the dynamic transport system, but this delay must exceed a threshold value in order to be considered.

2.2. Microscopic Discussion

Few works have considered the traffic behaviour when studying outdoors vehicle-based internal transport operational problems. In the surface mining environment, pickup and delivery operations involve a fleet of trucks transporting materials from excavation stations to dumping stations, through a designed shared road network. At pickup stations, shovels are continuously digging during a shift according to a pre-assigned mining production plan. Trucks are moving in a cyclical manner between shovels (pickup stations), and dumping areas (delivery stations). A truck cycle time is defined as the time spent by a truck to accomplish an affected mission that consists of travelling to a specific shovel, being serviced by the shovel and hauling material to a specific dumping area. Burt and Caccetta [5] state that mine productivity is very sensitive to truck dispatching decisions which are closely related to the truck cycle time. Thus several papers have studied and proposed algorithms and software to resolve this problematic issue. In fact, this critical decision consists of finding, according to the real environment, to which the best shovel truck must be affected. Such decision has to be generated continuously during a shift, whenever the truck finishes dumping at a delivery station.

Despite the several proposed dispatching software, the recent articles by Krzyzanowska [11] formally criticize the simplistic assumption behind those software which tend to provide dispatching decisions with the objective to optimise a truck cycle times previously calculated. Generally speaking, those software systems based on the optimisation process of the past period collected data of trucks cycle times and assume that for the next period trucks will spend on average the same time to accomplish missions. But in the reality of mining operation, the duration of truck travel time appears to be very sensitive to the variable traffic state and road conditions. Burt and Caccetta [5] and Krzyzanowska [11], point out the unresolved problematic of truck bunching and platoon formation in mining road network which apparently induce lower productivity.

2.3. Container Operations

Similarly to the material transportation in mining operation, several papers (Ioannou [10], Vis [15]) have provided the methods for improving the container terminal complex operations. In such applications, three types of handling operations are defined: vessel operations, receiving/delivery operations and container handling and storage operations in the stack yards. As we are interested by internal transportation systems, our review concerns the papers dealing with the container handling and storage operations in the stack yards. Generally speaking, vessels bring inbound containers to be picked up by internal trucks and distributed to the respective stocks in the yard.

Once discharged, vessels have to leave with on board outbound containers which also are delivered by internal trucks from the storage yard. For this purpose, trucks are moving through a terminal internal road network. In order to decrease the vessel turnaround time, which is the most important performance measure of the container terminals, it is important to perform those operations as quickly as possible. In fact according to [3], this movement of containers between quay sides and storage yards appears to affect the productivity of containership journey greatly. Vis and Koster [15] gives an extended review of the numerous research papers, providing algorithms to solve this complex routing and scheduling problem. They criticize the lack of consistency of the simplistic assumptions made to solve the proposed models within the real-world highly stochastic environment. The ignored traffic situation in the complex seaport internal transportation network is strongly criticized in recent papers [4], [10].

For example, in [3], a travel time of a container internal truck is modelled as a static mean time of travel, based on the distance and the truck average speed. Duinkerken et al. [6], put a uniform distribution between zero and 30% of the nominal travel time formulation, aiming to assimilate the complexity of traffic. More accurate work to solve this issue is the one provided recently by Liu, Chu and Recker [12]. They integrate a traffic model to the internal service model and reported the effectiveness of this integration which allows analysing the tractor traffic flow in a port container terminal.

Conscious about the critical problem of congestion in the road network inside a terminal, a quantitative measure of congestion to be added as a controllable decision variable have been developed. For this purpose, they considered the road system inside the terminal as a directed network and they measured flows on arcs in units of trucks travelling per unit time.

Those two last works appear as providing the leader approach in term of consideration of congestion and traffic in container terminals; however, their approach is ultimately macroscopic. As we have lately discussed, even if this macroscopic approach allows analysing the traffic behaviour, the highly detailed microscopic model is more efficient for an effective real-time traffic monitoring and control.

3. Real Transportation System

The analysed transportation system is a simplified case of the Polish Post (Fig. 1). The business service provided the Polish Post is delivery of mails. The system consists of a set of nodes placed in different geographical locations. Two kinds of nodes can be distinguished: central nodes (*CR*) and ordinary nodes (*PK*). There are bidirectional routes between the nodes. Mails are distributed among ordinary nodes by trucks, whereas between the central nodes by trucks, railway or by plane.

The mail distribution can be understood by tracing the delivery of some mail from point *A* to point *B*. At first the mail is transported to the nearest A ordinary node. Different mails are collected in ordinary nodes, packed in larger units called containers and then transported by trucks to the nearest central node. In central node containers are repacked and delivered to the appropriate (according to the delivery address of each mail) central node.

In the Polish Post there are 14 central nodes and more than 300 ordinary nodes. There are more than one million mails going through one central node within 24 hours. It gives a very large system to be modelled and simulated. Therefore, we have decided to model only a part of the Polish Post transportation system – one central node with a set of ordinary nodes.

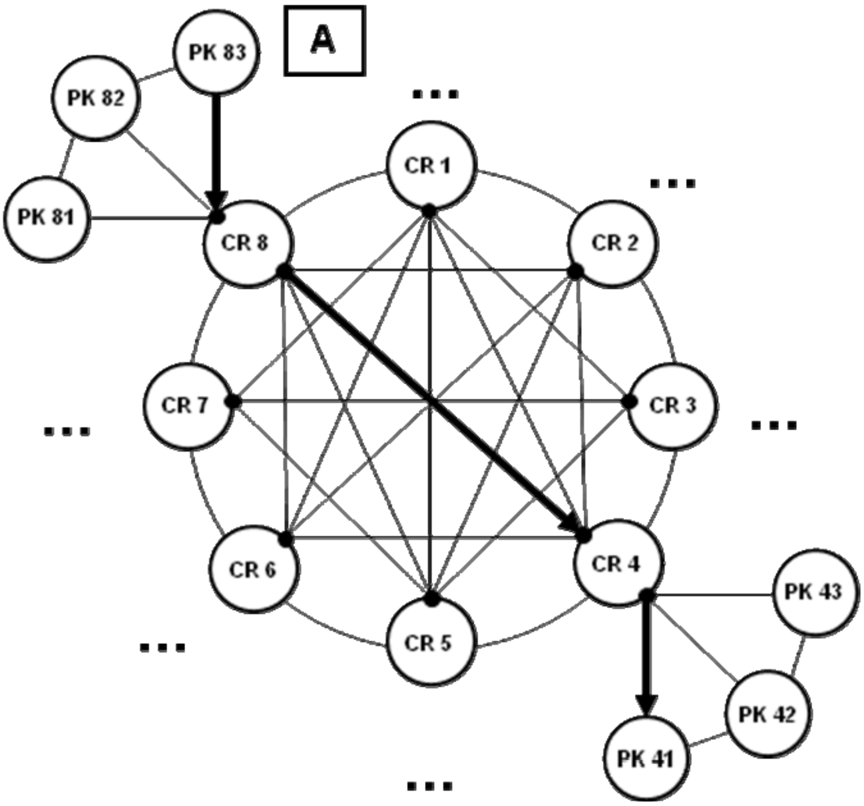


Figure 1. General idea of the analysed *DTS*

The income of mails to the system, or rather addressed containers of mails as it has been discussed above, is modelled by a stochastic process. Each container has a source and destination address. The generation of containers is described by some random process. In case of the central node, there are separate processes for each ordinary node. The containers are transported by vehicles. Each vehicle has a given capacity – maximum number of containers it can haul. The central node is a base place for all vehicles. They start from the central node and the central node is the destination of their travel. The vehicle hauling a commodity is always fully loaded or taking the last part of the commodity if it is less than its capacity. When a vehicle approaches the ordinary node it is waiting in an input queue if there are any other vehicles being loaded/unloaded at the same time. There is only one handling point in each ordinary node. The time of loading/unloading vehicle is described by a random distribution.

The containers addressed to the given node are unloaded and empty space in the vehicle is filled by containers addressed to a central node. Next, the vehicle waits till the time of leaving the node (set in the time-table) is left and starts its journey to the next node. The operation is repeated in each node on the route and finally the vehicle is approaching the central node when it is fully unloaded and then it is available for the next route. The process of vehicle operation can be stopped at any moment due to a failure. After the failure, the vehicle waits for a maintenance crew, it is repaired (random time) and then it continues its journey [3].

4. Formal Model

We decided to model formally a part of the Polish Post transportation system, one regional section which consists of one central node and a set of ordinary nodes.
An implementation of the system service needs a defined set of technical resources. Moreover, the operating of vehicles transporting mails between system nodes is done according to some rules – some management system. Therefore, we can model the discrete transportation system as a 4-tuple:

$$DTS = \langle Client, BS, TI, MS \rangle, \tag{1}$$

where: *Client* – client model, *BS* – business service, a finite set of service components, *TI* – technical infrastructure, *MS* – management system.

4.1. Technical Aspects

The technical infrastructure of *DTS* can be described by three elements:

$$TI = \langle No, V, MM \rangle , \tag{2}$$

where: No – the set of nodes; V – the set of vehicles; MM – the maintenance model.

The set of nodes (No) consists of a single central node (CR), a given number of ordinary nodes (PK_i). The distance between each two nodes is defined by the function:

$$distance : No \times No \to R_+ . \tag{3}$$

Each node has one functional parameter – the mean (normal distribution) time of loading a vehicle:

$$loading : No \to R_+ . \tag{4}$$

Moreover, the central node (CR) has additional functional parameter: the number of service points (in each ordinary node there is only one service point):

$$servicepoints : CN \to N_+ . \tag{5}$$

Each vehicle is described by the following functional and reliability parameters:

- mean speed of a journey
  $$meanspeed : V \to R_+ , \tag{6}$$
- capacity – number of containers which can be loaded
  $$capacity : V \to R_+ , \tag{7}$$
- mean time to failure
  $$MTTF : V \to R_+ , \tag{8}$$
  a time when failure occurs is given by exponential distribution with mean equal to a value of $MTTF$ function,
- mean repair time
  $$MRT : V \to R_+ , \tag{9}$$

it is modelled by a truncated Gaussian distribution.

The traffic is modelled by a random value of vehicle speed and therefore the time of vehicle (v) going from one node (n_1) to another (n_2) is given by a formula:

$$time(v, n_1, n_2) = \frac{distance(n_1, n_2)}{Normal(meanspeed(v), 0.1 \cdot meanspeed(v))} , \tag{10}$$

where $Normal$ denotes a random value with the truncated Gaussian distribution [17].

The maintenance model (MM) consists of a set of maintenance crews which are identical and unrecognized. The crews are not combined to any node, are not combined to any route, they operate in the whole system and are described only by the number of them.

The time when a vehicle is repaired is equal to the time of waiting for a free maintains crew (if all crews are involved into maintenance procedures) and the time of a vehicle repair which is a random value with the truncated Gaussian distribution: ($Normal(MRT(v), 0{,}1 \cdot MRT(v))$).

4.2. Human Factor

The human infrastructure is composed by the set of drivers. So the description of this part of system infrastructure requires the analysis of the drivers' state and the algorithms, which model the rules of their work. Each driver can be in one of the following states (s_d): rest (not at work), unavailable (illness, vacation, etc.), available (at work – ready to start driving), break (during driving), driving.

The number of driver working hours is limited by the labour law. For analysed Post Transportation System the daily limit for each driver equals to 8 hours and a single driver operates with one truck. It gives a simple algorithm:

- if w_h > *limit* then *state* = "rest" & w_h = 0,
- where w_h – working hours, *limit* = 8 hours

Drivers in Polish Post work in two shifts, morning or afternoon one. So twice a day a driver state and shift type is analysed:

- at 6am for each driver:
  - if *shift* == morning & s_d == rest then s_d = available,
- at 1pm for each driver:
  - if *shift* == afternoon & s_d == rest then s_d = available,

The next problem ought to be modelled is the driver's illness state. We propose the following approach:

- for every driver at 4am:
  - if s_d == rest and *rand()* < d_i then during *x* days (according to a given distribution) the driver is in s_d = unavailable,
    where d_i – driver's illness parameter.

Moreover we propose to categorise the driver's illnesses as follows: short sick: 1 to 3 days, typical illness: 7 to 10 days, long-term illness: 10 to 300 days [21]. We prepare the daily record of the driver. The algorithm to fix the driver to the vehicle is the last part of the driver model:

- if no driver – the vehicle does not start,
- driver can be chosen if: s_d = available
  and w_h + *estimated time of journey* < *limit* * 1.1,
- the driver is chosen randomly or by least value approach:
  abs(limit – w_h – *estimated time of journey*).

4.3. Business Service

Business service [19] (*BS*) is a set of services based on a business logic, that can be loaded and repeatedly used for concrete business handling process. Business service can be seen as a set of service components and tasks that are used to provide the service in accordance with the business logic for this process. Therefore, *BS* is modelled by a set of business service components (*sc*):

$$BS = \{sc_{1,} ..., sc_n\}, n = length(BS) > 0 , \tag{11}$$

the function *length*(*X*) denotes the size of any set or any sequence *X*.

4.4. Client Description

The service implemented by the clients of the transportation system, sending the mails from a source node to a destination one. Client model consist of a set of clients (*C*).

Each client is allocated in one of nodes of the transportation system:

$$allocation : C \to No . \tag{12}$$

A client allocated in an ordinary node is generating containers (since we have decided to monitor the containers but not the separate mails during the simulation) according to the Poisson process with destination address set to ordinary nodes. In the central node, there is a set of clients, one for each ordinary node. Each client generates the containers by a separate Poisson process and is described by the intensity of container generation:

$$intensity : C \to R_+ . \tag{13}$$

The central node is the destination address for all containers generated in ordinary nodes.

5. Dispatching Problem

5.1. Legacy Solution

The management system controls the operation of vehicle. It consists of a sequence of routes:

$$MS = \langle r_1, r_2, ..., r_{nr} \rangle . \tag{14}$$

Each route is a sequence of nodes starting and ending in the central node, times of leaving each node in the route (t_i) and the recommended size of a vehicle (*size*):

$$r = \langle CR, t_0, n_1, t_1, ..., n_m, t_m, CR, size \rangle \quad v_i \in No - \{CR\} \quad 0 \le t_0 < t_1 < ... < t_m < 24h . \tag{15}$$

The routes are defined for one day and are repeated each day. The management system selects vehicles to realise each route in a random way, first of all vehicles (among vehicles available in the central node) with capacity equal to recommended size are taken into consideration. If there is no such vehicle, vehicles with larger capacity are taken into consideration. If still there is no vehicle fulfilling requirements the vehicle of smaller size is randomly selected. If there is no available vehicle a given route is not realised [18]

5.2. Heuristic Approach

The decisions (to send a truck to a given destination node) are taken in the moments when a container arrives to the central node. The truck is send to a trip if:

- the number of containers waiting in for delivery in the central node of the same destination address as that just arrived is larger than a given number,
- there is at least one available vehicle,
- the simulated time is between 6 am and 22 pm minus the average time of going to and returning from the destination node.

The truck is send to a node defined by a destination address of just arrived container. If there is more than one vehicle available in the central node, the vehicle with size that fits best to the number of available containers is selected, i.e. the largest vehicle that can be fully loaded. If there are several trucks with the same capacity available the selection is done randomly. The restriction for the time of truck scheduling (the last point in the above algorithm) are set to model the fact that drivers are working on two 8-hours shifts.

5.3. Soft Computing Idea

The system consists of a multilayer perceptron to decide if and where to send trucks. The input to the neural network consists of:

$$in = \langle pkc_1, pkc_2, ..., pkc_{npk}, crc_1, crc_2, ..., crc_{npk}, nfv \rangle , \tag{16}$$

npk – number of ordinary nodes,
pkc_i – number of containers waiting for delivery in the central node with destination address set to i-th ordinary node,
nfv – number of free vehicles in the central node.

Each output of the network corresponds to each ordinary node:

$$nnout = \langle out_1, out_2, ..., out_{npk} \rangle . \tag{17}$$

The output of the network is interpreted as follows (for sigmoid function used in output layer):

$$j = \arg\max_{i=1...npk} \{out_i\} . \tag{18}$$

If out_j is greater than 0.5 send a vehicle to node j do nothing. If there are more vehicles available in the central node, the largest vehicle that can be fully loaded is selected. If there are available several trucks with the same capacity selection is done randomly. The neural network decision (send a truck or not and where the truck should be sent) are taken in the given moments in time. These moments are defined by the following states of the system:

- the vehicle comes back to the central node and is ready for the next trip,
- if there is at least one available vehicle in central node and the number of containers of the same destination address is larger than the size of the smallest available vehicle.

The neural network used in the management system requires a learning process that will set up the values of its weights. The most typical learning in the case of multilayer perceptron is the back propagation algorithm. However, it cannot be used here since it is impossible to state what should be the proper output values of the neural network. Since it is hard to reconcile what are the results of a single decision made by the management system. The results of the set of decisions are important. Since the business service implemented by transportation system is to move commodities without delays, the neural network should take such decisions that allow reducing delays as much as possible.

To train neural network to perform such task we propose to use genetic algorithm [18, 21]. Similar approach to training neural network is applied in case of computer games. The most important in case of genetic algorithm is a definition of the fitness function. To follow the business service requirements of transportation system we propose the following definition of the fitness function calculated for a given neural network after some time (T) (therefore after a set of decisions taken by the neural network) [17]:

$$fitness(T) = \frac{N_{ontime}(0,T) + N_{ontimeinsystem}(T)}{N_{delivered}(0,T) + N_{insystem}(T)} . \tag{19}$$

It is a ratio of on-time containers (delivered with 24h and being in the system but not longer then 24h) to all containers (that already delivered $N_{delivered}(0,T)$ and still being presented in the system $N_{insystem}(T)$).

6. DTS Simulator

6.1. Event-Driven Simulation

Once a model has been developed, it is executed on a computer. It is done by a computer program which steps through time. One way of doing it is so called event-driven simulation. It is based on the idea of event, which can be described by the time of event occurring and the type of event.

The simulation is done by analyzing the queue of event (sorted by time of event occurring) while updating the states of system elements according to the rules related to a proper type of event. Due to a presence of randomness in the *DTS* model the analysis of it has to be done basing on Monte-Carlo approach [21]. It requires a large number of repeated simulation.

Summarising, the event-driven simulator repeats N-times the following loop:

- beginning state of a *DTS* initialization,
- event state initialization, set time $t = 0$,
- repeat until $t < T$,
- take first event from event list,
- set time equals time of event,
- implement the event – change state of the *DTS* according to the rules related to the proper type of event: change objects attributes describing system state, generate new events and put them into event list, write data into output file.

6.2. Events and Elements of DTS

In case of *DTS* the following events (mainly connected with vehicles) have been defined:

- vehicle failure,
- vehicle starts repair,
- vehicle repaired,
- vehicle reached the node,
- vehicle starts from the node,
- vehicle is ready for the next route,
- time-table (starting the route in the central node).

Processing the events is done in objects representing *DTS* elements. The objects are working in parallel. The following types of system elements have been distinguished: vehicle, ordinary node, central node, time table.

The life cycle of each object consists of waiting for an event directed to this object and then execution of tasks required to perform the event. These tasks include the changes of the internal state of the object (for example when vehicle approaches the node it is unloaded, i.e. the number of hauled containers decreases) and sometimes creating a new event (for example the event "vehicle starts from the node" generates the new event "vehicle reaches the node" – next node in the trip).

The random number generator is used to deal with random events, i.e. failures. It is worth noticing that the currently analysed event not only generates a new event but also can change the time of some future events (i.e. time of approaching the node is changed when failure happens before). The time of a new event is defined by the sum of current time (moment of execution of the current event) and the duration of a given task (for example vehicle repair). Only time of starting a given route (event "vehicle starts from the central node") is predefined (according to the time table). Duration of all other tasks is defined by the system elements states:

- time when vehicle waits in a queue for loading/unloading,
- time when vehicle waits in a queue for maintains crew,

or is given by the random processes:

- time of vehicle going between two nodes,
- time of loading/unloading,
- time to failure,
- repair time.

Moreover, each object representing a node has additional process (working in parallel) which is responsible for generating containers. The life cycle of this process is very simple: waiting a random time, generating a container with a given destination address (central node for all ordinary nodes, and each ordinary nodes for processing in the central node) and storing a container in the store house (implemented as a queue) of a given node.

6.3. Implementation

The event-simulation program can be written in a general purpose programming language (like C++), in a fast prototyping environment (like Matlab) or a special purpose discrete-event simulation kernel. One of such kernels, is the Scalable Simulation Framework (*SSF*) [17] which is a used for *SSFNet* [17, 18] computer network simulator. *SSF* is an object-oriented *API* – a collection of class interfaces with prototype implementations. It is available in C++ and Java. *SSFAPI* defines just five base classes: Entity, inChannel, outChannel, Process, and Event. The communication between entities and delivery of events is done by channels (channel mappings connects entities).

For the purpose of simulating *DTS* we have used Parallel Real-Time Immersive Modelling Environment (*PRIME*) [17] implementation of *SSF* due to a much better documentation then available for the original *SSF*. We have developed a generic class derived from *SSF* Entity which is a base of classes modelling.

DTS objects which model the behaviour of the discrete transport system presented in section 2 and 4.

As it was mentioned, due to a presence of randomness in the *DTS* model, the Monte-Carlo approach is used. The original *SSF* was not designed for this purpose so some changes in *SSF* core have been done to allow restarting the simulation for several times from time zero within one run of simulation programme.

The statistical analysis of the system behaviour requires a very large number of simulation repetition, therefore the time of performance of the developed simulator is very important.

6.4. Performance

Next, we have tested the DTS simulator performance and scalability. We calculated the time of running one batch of simulation of the exemplar DTS presented above for 100 days on a 2.80 GHz Intel Core Duo machine on Linux system (Tab. 1). The CYGWIN base Windows implementation of PRIME SSF was around 10 times slower then Linux one. Next, we have enlarged the number of containers transported each day 10, 50, 100 and 500 times and proportionally enlarged the transport system (number of trucks, routes and service points). As it can be seen in Table 1 and Figure 2 the memory usage is linearly proportional to a number of containers transported each day. Whereas the simulation time is polynomially proportional (see Fig. 2).

We think that the time and memory effectiveness of simulation done in PRIME environment is very promising. Of course, the time needed to perform one simulation depends on the number of events presented in the system, which is a result DTS configuration.

Table 1. DTS simulator performance

	Reference DTS	10 x larger	50 x larger	100 x larger	500 x larger
Number of trucks	52	520	2 600	5 200	26 000
Number of router per day	184	1 840	9 200	18 400	92 000
Number of containers per day	4 400	44 000	220 000	440 000	2 200 000
Simulation time	0,21s	2,60 s	31,91 s	154,5 s	1910,0 s
Memory usage	2 MB	12 MB	63 MB	125 MB	600 MB

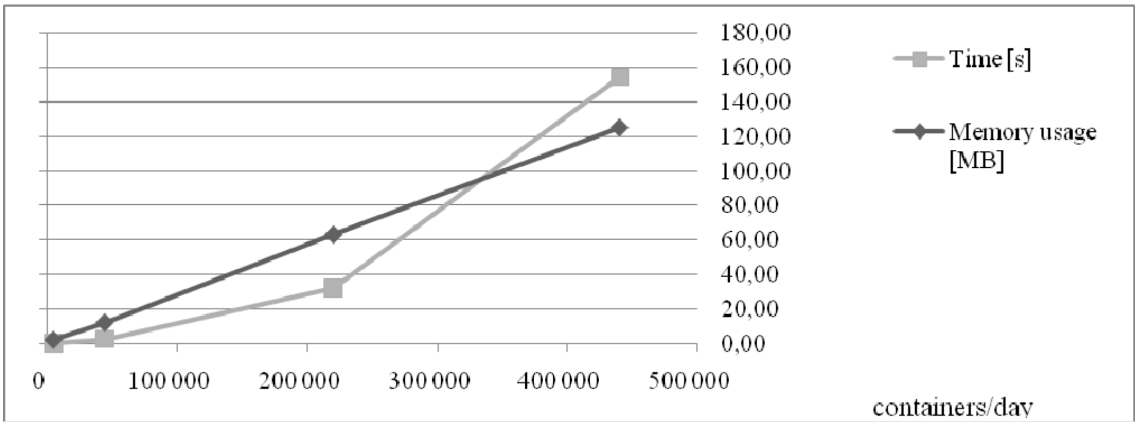


Figure 2. Simualtaion time and memory usage in a function of number of containers per day generated in DTS

7. Critical Situation

7.1. Problem Description

The *DTS* works correctly if there are no problems related to reliability and functional parameters. On the other hand the number and the volume of tasks loaded into the system cannot be above the system possibilities. The sentences sound very trivial, but – in general – it is not so trivial to find the global measure to check if the system is not overloaded.

Of course the correctly tuned system ought to be characterised by the set of fault tolerant features. It means the system is able to realise the loaded tasks even if the different faults occur because of reliability or functional insufficiencies. The problem pointed above needs a multi-criteria solution. In other words it is possible to find a kind of pareto set (Fig. 3.) to guarantee the system as functionally ready for the defined tasks.

The critical situation occurs if an actual operating point of the system is located outside the pareto set. The main goal if the critical situation is noticed is to drive the system to the pareto set as soon as possible. The proper management reaction is the first option to rescue the system situation. We propose the metric called the acceptance ratio to check if the operating point of the system is located at the pareto set [2].

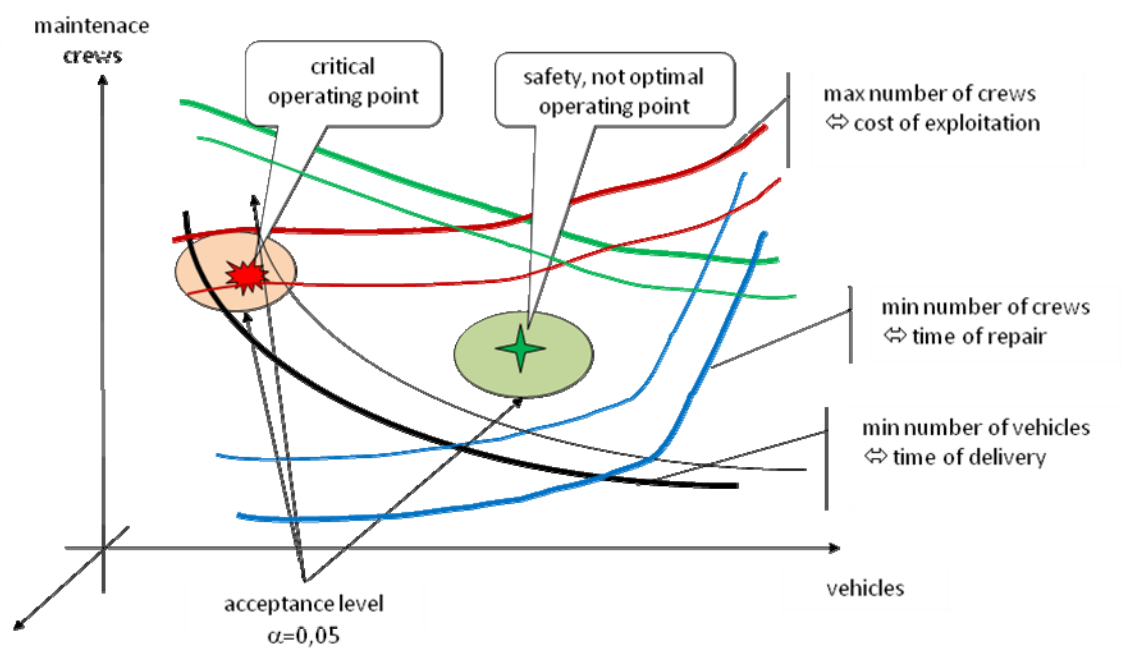


Figure 3. General idea of the pareto set

7.2. Availability – Definition

One can define the availability in different ways, but the value of an availability can always be easyly transformed into economic or functional parameters perfectly understood by owner of the system. The availability is mostly understood as a probability that the system is up and is defined as a ratio of the expected value of the uptime of a system to the observation time. It is a simple definition but it requires defining what it means that the transportation system is working.

The similar metric is the acceptance ratio defined by information as a number of accepted requests to the total number of requests.

In paper [20] we have proposed the definition of up time as a time when the number of delayed containers does not exceed a given threshold. Let us introduce the following notation:

- T – a time measured from the moment when the container has been introduced to the system to the moment when the container has been transferred to the destination (random value);
- T_g – a guaranteed time of delivery, if exceeded the container is delayed;
- $N_{deleyed}(t)$ – a stochastic process describing the number of delayed containers at time t, i.e. the number of containers for which $T > T_g$.

Therefore, the functional availability $A_k(t)$ can be defined as a probability that the number of delayed containers at time t does not exceed k, the value k is the level of acceptable delay:

$$FA_k(t) = \Pr\{N_{deleyed}(t) \le k\}. \tag{20}$$

7.3. Average Functional Availability

The defined in the previous section functional availability describes a state of an analyzed system at a given point of time. In case if somebody wants to analyze the state of the system in a time interval we have proposed in [7] the other metric: average functional availability $AFA_k(t)$. It is defined as an average probability that a system in the time interval from 0 to t is in up-time state (i.e. the number of delayed containers does not exceed threshold k):

$$AFA_k(t) = \frac{1}{t}\int_0^t \Pr\{N_{deleyed}(\tau) \le k\} d\tau . \tag{21}$$

7.4. Acceptance Ratio

A very often used estimation of the availability, which uses its asymptotic property and is based on an assumption of a uniform rate of the clients' requests, is the acceptance ratio. For *DTS*, we have defined it [16] as the ratio of on-time containers (containers for which $T < T_g$) to all containers within a given time of observation $(0, \tau)$. Within the time period a given number of containers are delivered ($N_{delivered}(\tau)$), a part of them or all delivered on time ($N_{ontime}(\tau)$), but in the end of analysed time period there can be some containers not delivered yet (waiting in the source node or being transported) ($N_{insystem}(\tau)$) and all or part of them being not late yet ($N_{ontimeinsystem}(\tau)$). Taking into consideration the introduced symbols the availability can be calculated as the expected value (Monte-Carlo approach) of ratio of on-time containers to all containers:

$$AR_\tau = E\left(\frac{N_{ontime}(\tau) + N_{ontimeinsystem}(\tau)}{N_{delivered}(\tau) + N_{insystem}(\tau)}\right). \tag{22}$$

8. Critical Situation Evaluation

8.1. Exemplar *DTS*

We propose for the case study analysis an exemplar *DTS* based on Polish Post regional centre in Wroclaw. We have modelled a system consisting of one central node (Wroclaw regional centre) and twenty two other nodes – cities where there are local post distribution points in Dolny Slask Province. The length of the roads has been set according to the real road distances between cities used in the analyzed case study. The intensity of generation of containers for all destinations has been set to 4.16 per hour in each direction giving in average 4400 containers to be transported each day.

The vehicles speed has been modelled by Gaussian distribution with 50 km/h of mean value and 5 km/h of standard deviation. The average loading time has been equal to 5 minutes. There have been two types of vehicles: with capacity of 10 and 15 containers. The MTF of each vehicle has been set to 20000. The average repair time has been set to 5h (Gaussian distribution). We also have tried to model the drivers' availability parameters. We have fulfilled this challenge by using the following probability of a given type of sickness: short sick: 0.003, typical illness: 0.001, long-term illness: 0.00025.

8.2. Results – Discussion

We entered the above defined system into critical situation: first of all to observe the actual value of the acceptance ratio – to find how deep degradation we can notice. The second goal was to compare the effectiveness of three possible management systems in case of driving the operating point of the system to the pareto set as soon as possible (Fig. 4). Finally we tried to find the real pareto set for the defined set of task loaded into the system, the adjusted number of vehicles and the adjusted number of drivers (Fig. 6).

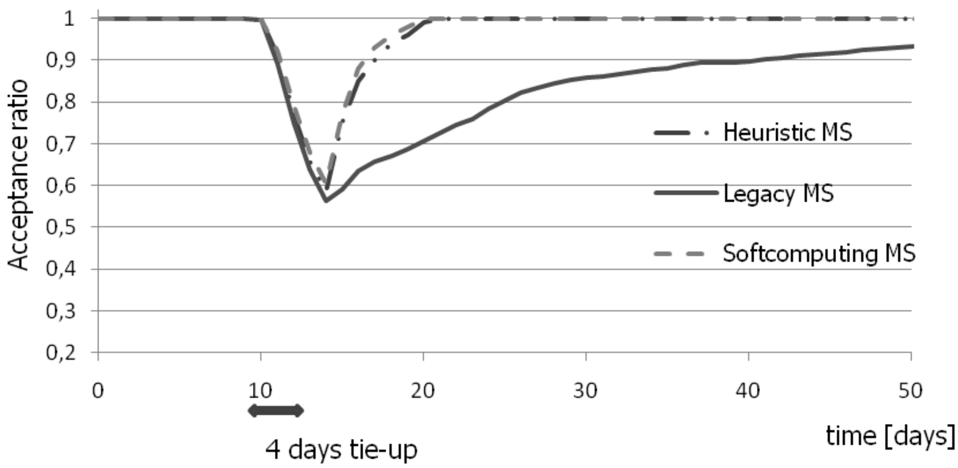
Let us assume that for some days the system is working at 50% level. The tie-up of the system can be caused for example by a driver strike or some contagious diseases resulting in situation that only 50% of vehicles are available. After a given number of days the system is again working fully.

The achieved results (acceptance ratio calculated according to (22)) for 4 and 14 days tie-up are presented in Figure 4. As it can be expected the acceptance ratio in day 10 (when critical situation starts) is starting to drop down and when drivers come back is enlarging.

However, the system with heuristic management as well as the soft computing one is coming back to normal operation much faster than the legacy one. The soft computing one is slightly outperforming the heuristic one.

The Figure 5 presents how many days are needed for the transportation system to achieve the required level (0.9, 0.95 and 0.98) after a tie-up of the different length. The system is driven by the soft computing management system.

a)



b)

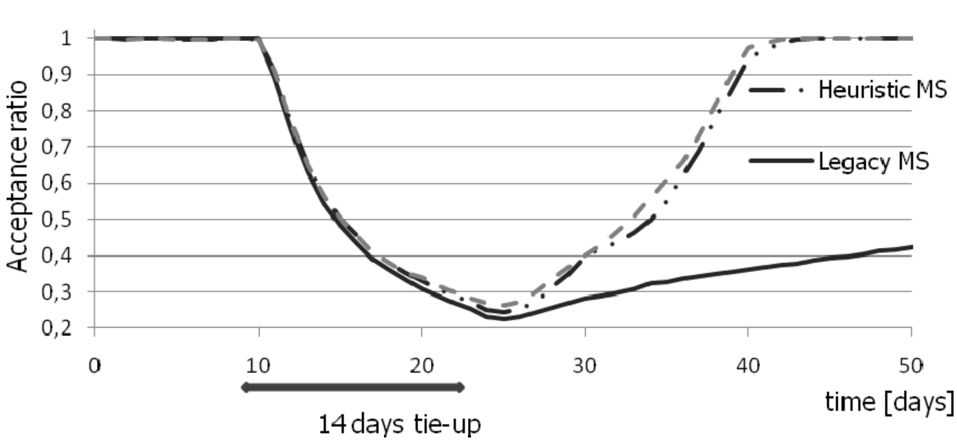


Figure 4. Acceptance ratio for 4 days (a) and 14 days (b) tie-up vehicles for different management system

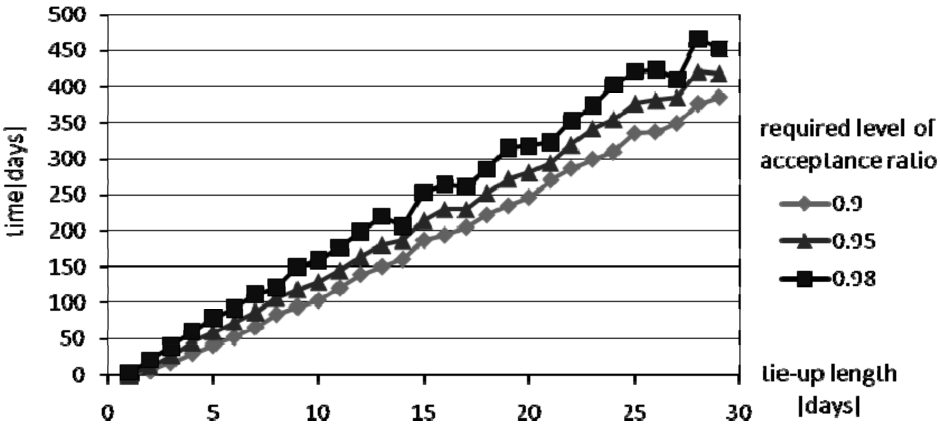


Figure 5. Time required to achieve a given level of acceptance ratio after a tie-up of different length for the exemplar *DTS* driven by the soft computing management system

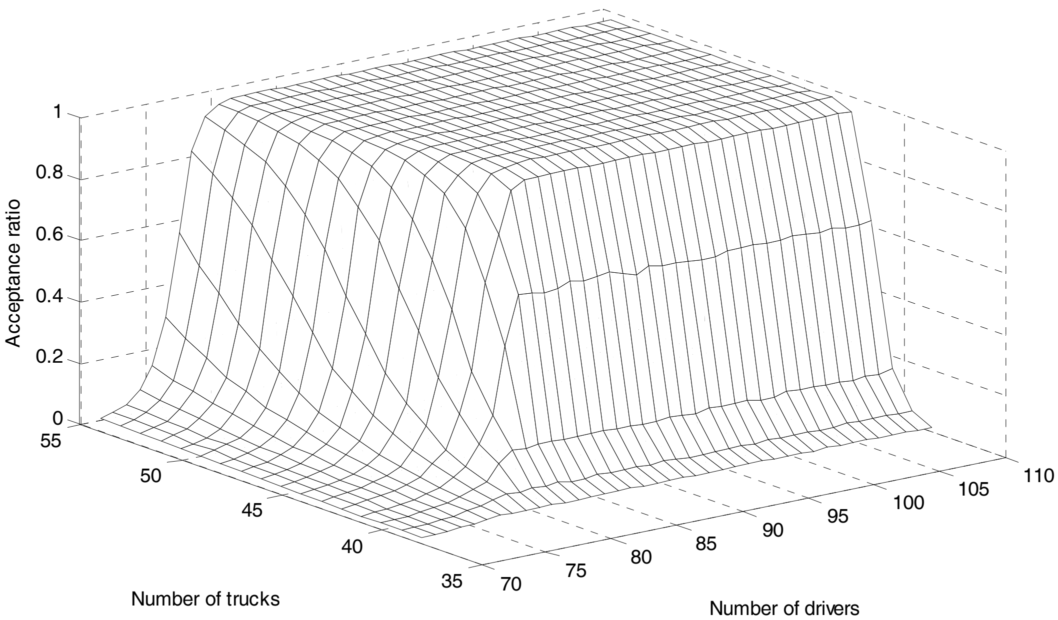


Figure 6. Pareto set in a function of number of trucks and number of drivers for tested *DTS*

9. Conclusion

We have presented a formal model of discrete transportation system (*DTS*) including reliability, functional parameters as well as the human factor component. The *DTS* model is based on the Polish Post regional transportation system. We proposed three different management approaches for the defined system. The critical situation is described and the availability of metric to create the pareto set guarantees the possible safety operating points for actual *DTS*.

The proposed approach allows performing reliability and functional analysis of the *DTS*, for example:

- determine what will cause a ”local” change in the system,
- make experiments in case of increasing the number of containers per day incoming to system,
- identify the weak points of the system by comparing a few of its configuration,
- understand better how the system behaves.

We have introduced the exemplar *DTS* into critical situation to discuss how deep breakdown it causes and to test which management approach can drive the system operating point into pareto set as soon as possible. The *DTS* described as the case study is based on the real Polish Post Wroclaw area. The presented solution can be used as a practical tool for defining an organization of the vehicle maintenance and transportation system logistics.

Acknowledgment

Work reported in this paper was sponsored by a grant No. N N509 496238, (years: 2010–2013) from the Polish Ministry of Science and Higher Education.

References

1. Barcelo, J., Codina, E., Casas, J., Ferrer, J.L. and D. Garcia. Microscopic Traffic Simulation: a Tool for the Design, Analysis and Evaluation of Intelligent Transportation Systems, *Journal of Intelligent and Robotic Systems: Theory and Applications*, Vol. 41, 2005, pp. 173–203.
2. Barlow, R., Proschan F . *Mathematical Theory of Reliability*. Philadelphia: Society for Industrial and Applied Mathematics, 1996. 345 p.
3. Ben-Akiva, M., Cuneo, D., Hasan, M., Jha, M. and Q. Yang. Evaluation of Freeway Control Using a Microscopic Simulation Laboratory, *Transportation Research,* Part C (Emerging Technologies), 11C, 2003, pp. 29–50.
4. Birta, L., Arbez G. *Modelling and Simulation: Exploring Dynamic System Behaviour*. London: Springer, 2007. 246 p.
5. Burt, C.N. and L. Caccetta. Match Factor for Heterogeneous Truck and Loader Fleets, *International Journal of Mining, Reclamation and Environment*, Vol. 21, 2007, pp. 262–270.
6. Duinkerken, M.B., Dekker, R., Kurstjens, S.T.G.L., Ottjes, J.A., and N.P. Dellaert. Comparing Transportation Systems for Inter-Terminal Transportation at the Maasvlakte Container Terminals, *OR Spectrum*, Vol. 28, 2006, pp. 469–493.
7. Fishman, G. *Monte Carlo: Concepts, Algorithms, and Applications*. Springer-Verlag, 1996. 129 p.
8. Gartner, N., Messer, C.J. and A.K. Rathi. *Traffic Flow Theory and Characteristics / T.R. Board, ed.* Texas: University of Texas at Austin, 1998. 125 p.
9. Gold, N., Knight, C., Mohan, A., Munro, M. Understanding service-oriented software, *IEEE Software*, Vol. 21, 2004, pp. 71–77.
10. Ioannou, P.A. *Intelligent Freight Transportation.* Carolina: Taylor and Francis Group, 2008. 245 p.
11. Krzyzanowska, J. The Impact of Mixed Fleet Hauling on Mining Operations at Venetia Mine, *Journal of The South African Institute of Mining and Metallurgy*, Vol. 107, 2007, pp. 215–224.
12. Liu, H., Chu, L, Recker W. Performance Evaluation of ITS Strategies Using Microscopic Simulation, *Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems*, 2004, pp. 255–270.
13. Sanso, B. and L. Milot. Performability of a Congested Urban-Transportation Network when Accident Information is Available, *Transportation Science*, Vol. 33, No 1, 1999, pp. 10–21.
14. Taylor, M.A.P., Woolley, J.E., Zito, R. Integration of the Global Positioning System and Geographical Information Systems for Traffic Congestion Studies, *Transportation Research* Part C (Emerging Technologies) 8C, 2000, pp. 257–285.
15. Vis, I.F.A. Survey of Research in the Design and Control of Automated Guided Vehicle Systems, *European Journal of Operational Research*, Vol. 170, 2006, pp. 677–709.
16. Walkowiak, T., Mazurkiewicz, J. Analysis of Critical Situations in Discrete Transportation Systems. In : *Proceedings of International Conference on Dependability of Computer Systems*, *Brunow, Poland, June 30-July 2, 2009*. Los Alamitos: IEEE Computer Society Press, 2009, pp. 364–371.
17. Walkowiak, T., Mazurkiewicz, J. Availability of Discrete Transportation System Simulated by SSF Tool. In: *Proceedings of International Conference on Dependability of Computer Systems*, *Szklarska Poreba*, *Poland, June, 2008*. Los Alamitos: IEEE Computer Society Press, 2008, pp. 430–437.
18. Walkowiak, T., Mazurkiewicz, J. Functional Availability Analysis of Discrete Transportation System Realized by SSF Simulator. In: *Computational Science – ICCS 2008, 8th International Conference*, *Krakow, Poland, June 2008*. Springer-Verlag, LNCS 5101, 2008. Part I, pp. 671–678.
19. Walkowiak, T., Mazurkiewicz, J. Algorithmic Approach to Vehicle Dispatching in Discrete Transportation Systems. In: *Technical approach to dependability / Eds. Jarosław Sugier [at al].* Wroclaw: Oficyna Wydawnicza Politechniki Wroclawskiej, 2010. pp. 173–188.
20. Walkowiak, T. and J. Mazurkiewicz. Functional Availability Analysis of Discrete Transportation System Simulated by SSF Tool, *International Journal of Critical Computer-Based Systems*, Vol. 1, No 1–3, 2010, pp. 255–266.
21. Walkowiak, T. and J. Mazurkiewicz. Soft Computing Approach to Discrete Transportation System Management. In: *Lecture Notes in Computer Science. Lecture Notes in Artificial Intelligence*. Springer-Verlag, 2010. Vol. 6114, pp. 675–682.