Journal of Polish Safety and Reliability Association
Summer Safety and Reliability Seminars, Volume 4, Number 2, 2013

Mazurkiewicz Jacek
Wrocław University of Technology, Wrocław, Poland

Softcomputing approach to Discrete Transport System analysis

Keywords

dependability, modelling, complex systems, softcomputing

Abstract

The paper presents an approach to complex analysis of sophisticated Discrete Transport Systems (*DTS*) using softcomputing. The proposed methodology can efficiently substitute the classical approach especially since dependability parameters of the system are mostly approximated by experts instead of classical sources of data. On the other hand we can try to combine reliability and functional aspects. The paper describes the multi-criteria scientific field of the Discrete Transport Systems and shows the way how the softcomputing can be in use in the sensible way. Moreover, results of the numerical experiment performed on a test case scenario related to the reliability and functional aspects using proposed methodology are given. The approach allows reducing the problem of assumptions of reliability distributions and – this way – seems to be very interesting for real systems management and tuning.

1. Introduction

The most often discussed methods of transport are focused on commodity movement according to declared routes, using autonomous equipment (vehicles) characterised by the capacity and based on prepared schedule. The presented class of transport systems is called discrete transport system (*DTS*). We can find an example of *DTS* as a commodity transportation realised by trucks. The schedule of such *DTS* can be not very precise against to coaches, passenger airplanes or trains. Of course there are situations where the schedule ought to be very precise – productive systems working without storehouses – for example – with remote co-operating parties [25], [13]. It is not trivial to model the transportation system properly for quality and efficiency estimation. The discrete transport system definition presented below includes all elements which have effect on service quality served by a supplier according to fixed strategy, real functional and reliability parameters of equipment. Such defined model combines both dependability and functional features. It allows to model discrete transport systems and to analyse the efficiency of the system if the number as well as quality of vehicles changes. We can also test how the system works if the number and location of recipients vary or different types of service strategy are available, or we notice failures [10], [15]. If we think about the transportation system as combination of equipment, infrastructure and human dispatcher we need to substitute the ordinary reliability models by functional and dependability models to check the system reaction for failures as well as to find the system efficiency changes after the dispatcher decisions [27], [6]. It is necessary for functional and dependability models to expand the definition of proper (reliable) state of system. The transportation system works correctly if tasks are realised according to the agreement – it means the commodity is transported on schedule, with declared volume. Failures of vehicles and infrastructure deteriorate the efficiency of the system, but if transportation tasks are realised according to the agreement we can say that the system works correctly. In the real transportation systems it is possible to substitute some functions by similar functions operating by various configurations, using different infrastructure features and redefined schedule [7], [16]. This way the system realises the task based on set of resources called functional configurations. The resources allocation is realised in dynamic way – modifications are driven by the stream of tasks, failures and dispatcher decisions [4], [9]. Complexity of this solutions force lower level of its description but in a same time high level perspective on what is going on in the system. Regardless of the level of abstraction many of parameters should be defined or measured to find the most accurate solution. Since some of them are uneasy to measure we propose to use softcomputing [15] to model the system, since this kind of model can be very useful for further analysis of the system. In result we provide an approach or even an idea of the tool for network system administrator. We call the approach as the functional-reliability models of network system exploitation. In this part we shortly describe elements of transport system Next sections provide the details of the softcomputing approach necessary to reliability and functional description and analysis of the discrete transport system. The results show the essential practical data in the function of the reliability parameters of the system. Paper ends with some general conclusion and remarks for the future works.

2. Discrete Transport System

The Discrete Transport System (*DTS*) is understood as a system of transport resources (e.g. vehicles), transport infrastructure (e.g. roads) and a management system (e.g. a dispatcher supported by a computer system). In this way dependability (functional – reliable) properties of the *DTS* depend not only on technical infrastructure of the system but also on dispatcher decisions [26], [18]. Dispatcher decisions may be a reaction on traffic situations (e.g. a traffic jam, a temporary limitation of vehicle speed on the fixed segment of a road), on infrastructure faults (e.g. a truck with cargo is failed and it has to be repaired), on functional system faults (e.g. a point storehouse is overfilled or already sent parcels are not collected yet) [11], [25], [20]. The dispatcher decisions are taken on the base of such different criteria as financial costs, system performance parameters, availability of renewal teams, possibility to access other routes, acceptability of parcel delaying, etc. The Discrete Transport System *DTS* is defined as [27]:

$$DTS = \langle TI, RES, TT, MS \rangle \quad (1)$$

where:
TI – technical infrastructure of the system,
RES – system resources,
TT – transport tasks,
MS – management system which is called dispatcher.

The **technical infrastructure** *TI* of the discrete transport system is modelled as a directed graph [27], [19]:

$$TI = \langle reloading\ \ places, roads \rangle = \langle RP, R \rangle \quad (2)$$

where:

$$RP = \langle A, B, C, \ldots \rangle \quad (3)$$

- set of reloading places (*Figure 1*),

$$R = \langle AB, AC, BC, \ldots \rangle \quad (4)$$

- set of roads connecting reloading places.

A **reloading place** is a node of the discrete transport system (a node in the *TI* graph) in which such functions as parcels collecting in storehouses, reloading parcels from one transport resources to other one or to a storehouse may be realised. The reloading place may be equipped with a storehouse (with limited capacity; e.g. C_A, C_B, etc.) and needs such “mechanical tools” as cranes or fork-lift track.

Roads are modelled as directed arcs connected to nodes of the *TI* graph. Engineering parameters of the road are integrated into one representative measure called *average speed* of transport resource on this road segment (e.g. v_{AB}). Of course the average speed depends of cargo, transport means type, direction of traffic, day time or month time etc. Sometimes it is possible that $v_{AB} \neq v_{BA}$, but we can also assume the speed values are equal ($v_{AB} = v_{BA}$).

System resources of the *DTS* are understood as collections of transport means, drivers and service teams which the dispatcher may use for transport tasks realisation and for removing some disturbances in the system work. A system resource is described by its functional (e.g. load capacity of a truck), technical (e.g. fuels expendable per kilometre) and reliability parameters (e.g. mean time between failures or mean time renewal) which may have deterministic or probabilistic nature. Drivers create a specific class of the system resource [24], [13], [27].

A **transport task** *TT* is understood as a pickup of a fixed cargo from the start node and a delivery of it to final node according to assumed time-table. Of course the transport task may be defined in more complicated way, e.g. a cargo may be collected in a few nodes and reloaded in several ones. Transport schedule can be defined in different ways, for example a cargo ought to be delivered to the node before the end of fixed time-period, because a train cannot wait for a truck with the cargo.

Faults and renewals of a discrete transport system. There are considered many disruptions in execution of the discrete transport systems. The failures of the *DTS* resources *RES*, e.g. physical failures of trucks or technical infrastructure *TI* (e.g. roads or reloading devices) need to use adequate such the *DTS* system means as service teams, garages, spare elements or substituted routes. Generally, in these situations **“technical” system renewal** processes are started on with assumption of the limited resources [27], [24]. Other sources of the *DTS* system disruptions we can find in organisation and management matters:

1) overloading of the technical infrastructure (roads, reloading machines, etc.),
2) traffic accidents or jams,

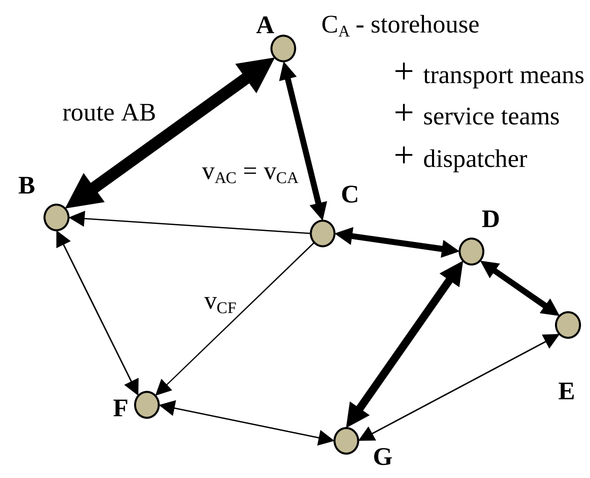

Figure 1. Discrete Transport System – an idea

3) dispatcher faults – he or she is not able to keep up the dynamic changes of the situation in the working *DTS* system. In these cases exploitation system renewal processes are initiated by the system dispatcher. The processes very often consume more time and money than a renewal of a “simple (physical)” broken technical resource, e.g. a repair of a failed truck or a lift.

The **dispatcher** organises a work of the *DTS* system - available system resources are assigned to realised tasks. The dispatcher administers logistics of a transport firm based on the signed agreements specifying conditions of correct realisation of a task or sets of tasks [18], [26]. Dispatcher decisions are taken based on needs of assumed transport tasks and according (if it is possible) to assumed proper time-tables. When some disruptions (failures, faults) occur the dispatcher chooses adequate system reactions. The dispatcher is supported by the computer aided tools to improve an assignment of system resources to transport tasks, to compose system traffic time-tables (planned and reserved for emergency conditions), to provide maintenance policies ready to use both for normal and disrupted situation in the system work. Because there are a lot of people involved into the system work the dispatcher should take his/her decisions taking into account not only computer system support results, but also based on his/her experience and his/her intelligence [17], [10], [6]. It is possible to define many classes of dispatchers, who are working in harmony with agreement between an transport employer and an owner of the discrete transport system. A **passive dispatcher** realises transport tasks agree to previously defined conditions and schedules. He or she uses earlier prepared lists of assumed *DTS* disruptions and lists of planned adequate system reactions in case of disruptions. A **task oriented dispatcher** is focused on execution of selected task or its sets. He or she may works agree with such strategy as FIFO, LIFO, FILO etc. A **dynamic dispatcher** is monitoring on-line a system and takes decisions adequate to system situation; of course the dynamic dispatcher cannot work as a fantastic virtuoso manager. If more detailed supporting data are prepared a priori, the real dependability properties (performance and reliability parameters) of the considered *DTS* system are closer to expected. Dependability measures of discrete transport systems are defined as global values (e.g. system efficiency, financial profit or loss) or as more detail measures such as a probability of isolated task execution or a set of tasks realised in a determined time interval [27], [24]. Functional and reliable properties of a discrete transport system have an effect on dependability measures at two fundamental levels:

1. it is possibility to create a functional configuration of the task or the set of tasks, that means it is possibility to allocate needed system resources for the transport task (or tasks) execution,
2. it is possibility that the transport task is correctly realised, that means allocated resources correctly work during assumed time and the assumed cargo is delivered according to assumed time-table.

The resources of all real systems are limited, so the system dispatcher has a significant impact on solving above given problems. His/her decisions concerning allocation technical infrastructure, transport means, service teams or reconfiguration of the system have to be taken up quickly and adequate to the situation [9], [15], [4].

3. Analyzed system

3.1. System overview

Basic elements of system are as follow: store-houses of tradesperson, roads, vehicles, trans-shipping points, store-houses of addressee and transported media (commodities). The commodities are taken from store-houses of tradesperson and transported by vehicles to trans-shipping points. Other vehicles transport commodities from trans-shipping points to next trans-shipping points or to final store-houses of addressees. Moreover, in time of transportation vehicles dedicated to commodities could failed and then they are repaired (*Figure 2*) [24]. Different *commodities* are characterized by common attribute which can be used for their mutual comparison: capacity of commodities. The following assumptions related to the commodities are taken: it is possible to transport *n* different kinds of commodity in the system and each kind of commodity is measured by capacity.

Road is an ordered double of system elements. The first element must be a store-house of tradesperson or trans-shipping point, the second element must be a trans-shipping point or store-house of addressee. Moreover, the road is described by following parameters: length, number of maintain crews, number of vehicles moving on the road. The road is assumed to have no damages. A single *vehicle* transports commodities from start to end point of a single road, return journey realizes in an empty status and the whole cycle is repeated. The assumptions are as follow: a single kind of commodity is transported at the moment, vehicles are universal. The numerous vehicle parameters can be described as random variables using various distributions. The *store-house of tradesperson* is an infinity source of single kind of commodity. *Trans-shipping points* are a transition part of the system which is able to store the commodities. The trans-shipping point is described by following parameters: global capacity, initial state described by capacity vector of commodities stored when the system observation begins, delivery matrix. This matrix defines which road is chosen when each kind of commodity leaves the shipping point. The commodity could be routed to more than one direction. Only one vehicle can be unloaded at the moment. If the vehicle can be unloaded, the commodity is stored in the trans-shipping point. If not, the vehicle is waiting in the only one input queue serviced by FIFO algorithm. Only one vehicle can be loaded at the moment. If the vehicle can be loaded (i.e. the proper commodity is presented and it could be routed a given road) the state of trans-shipping is reduced. If not, the vehicle is waiting in the each output road FIFO queue [19]. The main task of the *store-houses of addressee* is to store the commodity as long as the medium is spent by the recipient. The store-house of addressee is described by following parameters: global capacity, initial state described as for the trans-shipping point, function or rule which describes how each kind of commodity is spent by recipients. Input algorithm is exactly the same as for trans-shipping point. Output algorithm can be described as: stochastic process, continuous deterministic or discrete deterministic one. Moreover, the following assumptions are taken: the capacity of the commodity can't be less than zero, "no commodity state" - is generated when there is a lack of required kind of commodity.

3.2. Economic analysis

The economic analysis is realized from vehicle owner's view-point. The revenue is proportional to number of store-houses of addressee, number of deliveries realized to single store-house of addressee and gain for delivery to store-house of addressee.

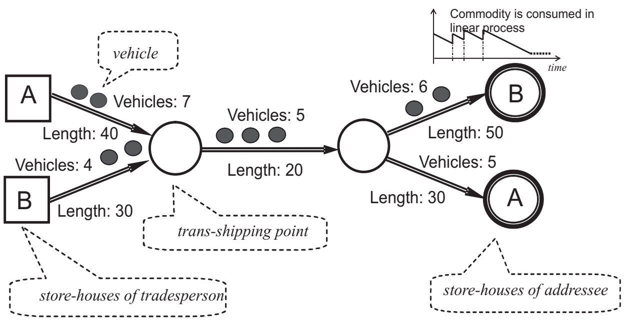

Figure 2. Analysed system – case study

Following costs are taken into account: penalty costs - paid by a transportation firm when there is a lack of commodity in the store-house of addressee, repair costs - proportional to a unit of repair time, vehicle usage costs - in a function of time (salary of drivers) and in a function of distance (i.e. costs of petrol). The economic quality of discrete transport system is described by overall gain function *G(T)* estimated in given time-period *T* as difference between the revenue and costs. We have to remember that the overall gain *G(T)* is a random variable [12].

3.3. System structure

The simulation program generates a description of all changes in the system during simulation (with all events). It is a base for calculation of any functional and reliability measures. The most valuable results of statistical analysis are: time percentage when the vehicle is present in each state, time percentage when the store-house of addressee is present in each state, mean time when the store-house of addressee is empty - this way we can say if "no commodity state" is prolonged or only momentary. We also propose a quantile calculation of time when the store-house of addressee is empty. This is the answer if "no commodity state" situation sometimes lasts significantly longer than the mean time of empty store-house. Moreover, it is possible to observe the influence of changes related to single parameter or a set of parameters - vehicle repair time for example - for other system characteristics - as vehicle utilization level, or commodity accessible in store-houses. The calculated reliability and functional measures could be a base of developing economic measures [22], [14], [21]. Such layered approach allows a high level, economic analysis of the system. It is necessary to check different variants of maintenance organization and to choose the less expensive among them if the reliability criteria are satisfied. It could be done by subsequent Monte-Carlo analysis and calculation of the required economic or functional measures for a set of analyzed parameters. The system model described in previous sections is a subject of computer simulation. A special software package for simulation of the discrete transport system has been developed. The transport system is described in the specially designed script language (with syntax similar to XML) [5]. It is an input for simulator program (written in C++) performing Monte-Carlo simulation [1], [8]. Monte Carlo simulation has an advantage in that it does not constrain the system structure or kinds of distributions used [5]. However, it requires proper data pre-processing, enough time to realize the calculations and efficient calculation engine.

4. Proposed approach

The problem of speeding up functional and reliability analysis of discrete transport system we propose to solve by hybrid system using simulation and neural nets. In many tasks, i.e. in decision systems, there is a need to give an answer in a short time. However Monte-Carlo simulation requires quite a lot of time to realize calculations for a given set of system parameters. To solve this problem we have proposed a use of artificial neural networks [23]. The use of neural network is motivated by its universal approximation capability [3]. Knowing that most of output system parameters are continues we can expect that neural network can approximate any unknown function based on a set of examples. The time needed to get an output from learn neural network is very short. Solution generated by net seems to be satisfactory [23], because we do not need very precise results - time is the most important attribute of the solution. The neural network partly substitutes the simulation process. The neural net module is added to developed simulation software. The aim of this module is to generate an answer how to select the best system parameters (i.e. the maintenance agreements - the average time of vehicle repair) based on the achieved system functional parameters (i.e. the average time of "no commodity" in the store-house of addressee). The process of data analysis will be as follows:

1. set the input parameters for model of discrete transport system;
2. give a range of an analyzed free parameter (parameters);
3. perform initial Monte-Carlo analysis for a few parameters from a given range - calculate all required functional and reliability parameters;
4. build a neural network approximation tool: Multilayer Perceptron; the inputs to the network are analyzed free parameters; the outputs are functional and reliability measures;
5. build the answer about the maintenance agreement based on the output of the neural network and the proper economic measures;
6. communicate with a user: play with functional and reliability data, goto 4.

If more accurate analysis of economic parameter in a function of free parameter is required goto 3 - perform more Monte-Carlo analysis.

4.1. Case study – reliability parameters

The experiment was realized for a simple parallel system (see *Figure 3*), which is composed of two identical elements. Thus there are three reliability states: both elements are working; one element is working, another is failed; both elements are failed. We assume that, time to failure for both elements has an exponential distribution. Time of repair consists of two parts: fixed time of maintenance crew coming and time to repair - normal distribution. Also, failures of the elements are independent. Of course structure of the proposed system is rather trivial, but easily expendable. On the other hand, the distribution of repair time is not so trivial. In general, such example is very good to demonstrate the idea of reliability analysis supported by neural network and can be easily adopted for different real tasks. For a case study we focused on estimating the mean time over one year of being in one of three system stages. The reliability analysis was realized using Monte Carlo simulation. Experiments were performed for given data:

- data related to failures: mean time to failure: 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60 days (these values were taken with an assumption about the exponential distribution of time to failure);

- data related to repairs: repair crew coming time: 0.1, 0.3, 0.5, 0.7, 0.9, 1.1, 1.3, 1.5, 1.7, 1.9, 2.1 days (the additional assumption is that the maintenance crew coming time is fixed), mean time to repair: 0.1, 1, 2, 4, 6, 8, 10 days.

These values were taken with an assumption about the normal distribution of time to repair. Standard deviation of repair time is the last data. We took the following values of this parameter: 0 - fixed time to repair, 0.5, 1.5, 2, 2.5 days. Additionally, all data where standard deviation of repair time was larger two times then mean time to repair where removed. The presented data gave in summary: 2904 performed experiments. For each set of parameters 20000 Monte Carlo simulation where performed. The Multilayer Perceptron was used to estimate the mean time over one year of being in one of three system stages. The network has four input neurons: corresponding to each of system parameters: mean time to failure, crew coming time, mean time to repair and standard deviation of time to repair. Three outputs correspond to mean time over one year of being in one of three system stages (both elements are working; one element is working, another is failed; both are failed.

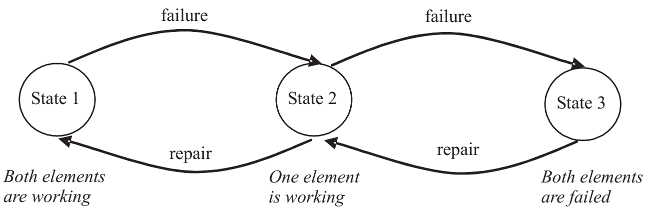

Figure 3. Structure of analysed system – reliability

Outputs of the network where normalized to (0,1) range - divided by 356 (duration of year in days). Moreover, the network has single hidden layer and a sigmoidal transfer function. Number of neurons in the hidden layer was set to 20 by a set of preliminary experiments. The network was trained by Levenberg-Marquardt [3] algorithm using MATLAB package. Input data were divided in random way into two equally sized sets: training and testing. Three different kinds of experimental distance between resulting and original function have been used during testing procedure:

$$L_1 = \frac{1}{N}\sum_{i=1}^{N}\left|y(x_i) - \hat{y}(x_i)\right| \qquad (5)$$

$$L_2 = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y(x_i) - \hat{y}(x_i)\right)^2} \qquad (6)$$

$$L_\infty = \max_{i=1,\dots,N}\left|y(x_i) - \hat{y}(x_i)\right| \qquad (7)$$

where:

$y^{\wedge}(x_i)$ - network output,
$y(x_i)$ - desired output (Monte Carlo simulation),
N - number of examples.

Tests were performed also for different number of hidden neurons: 4, 6, 8, 10, 12, 14, 16, 18 and 20.

4.2. Results: reliability parameters

For each set of parameters 20000 Monte Carlo simulation where performed. Resulting values and data range of them are as follow:

T1 - average time when both objects are working in one year: 35.85 - 362.56 days,
T2 - average time when only one object is working in one year: 2.42 - 165.62 days,
T3 - average time when both objects failed in one year: 0.01 - 178.85 days.

Results related to the tests of number of neurons in hidden layer are presented in *Table 1*. The network with 20 hidden neurons performs in the best way. To compare the neural network performance two additional tests were taken. One was focussed on Monte Carlo data stability. All simulations were performed one time more. The distances for different metrics between these two simulations are presented in *Table 2*. The second test was focussed on a classical method of time calculation of being in a given state based on stationary Markov model (with assumption of all exponential time distributions) [2]. Achieved results were compared with Monte Carlo simulation (*Table 1*).

Table 1. Errors (for different types of distance) in days for different number of hidden neurons for testing set

State number	Distance Type	Number of neurons in hidden layer								
		4	6	8	10	12	14	16	18	20
1	L_1	2.56	0.34	0.25	0.22	0.32	0.15	0.16	0.16	0.17
	L_2	3.55	0.55	0.31	0.29	0.40	0.20	0.23	0.21	0.22
	L_∞	34.74	6.79	1.26	1.20	1.91	1.08	2.45	0.92	0.93
2	L_1	2.02	0.46	0.28	0.21	0.25	0.16	0.15	0.14	0.14
	L_2	3.39	0.63	0.38	0.28	0.35	0.21	0.23	0.18	0.18
	L_∞	30.92	5.13	1.63	1.27	2.87	1.20	3.28	1.22	0.64
3	L_1	1.07	0.26	0.19	0.13	0.22	0.10	0.09	0.11	0.09
	L_2	1.53	0.35	0.24	0.17	0.28	0.14	0.13	0.14	0.13
	L_∞	7.08	2.40	1.20	0.83	0.97	0.68	0.64	0.80	0.65

Table 2. Errors in days for best neural network, Markov model and other Monte Carlo simulation

State number	Distance Type	Errors in days between Monte Carlo simulation results and		
		other MC simulation	best neural network	Markov model
1	L_1	0.18	0.17	8.28
	L_2	0.23	0.22	10.58
	L_∞	1.01	0.93	23.83
2	L_1	0.13	0.14	6.24
	L_2	0.17	0.18	7.51
	L_∞	0.75	0.64	14.54
3	L_1	0.11	0.09	2.04
	L_2	0.15	0.13	3.34
	L_∞	0.75	0.65	10.84

Looking at results (*Table 2*), it is clear that neural network gives much better results than Markov approach. Moreover, received performance (in meaning of a result accuracy) of the neural network method is very close to the Monte Carlo method stability and probably very close to it is possible optimal performance.

4.3. Case study – functional parameters

To show possibilities of the proposed model and developed software we have analyzed an exemplar transport network presented in *Figure 4*. The network consists of two store-houses of tradesperson (each one producing its own commodity, marked as A and B), one trans-shipping point (with one storehouse for both commodities) and two store-houses of addressee (each one with one storehouse). The commodities are spent by each recipient. The process is continuous deterministic, the amount of consumption in time unit is marked by u with subscripts corresponding to store-houses of addressee and commodity id. It's exemplar values are presented in *Figure 4*. Having lengths of the roads (see *Figure 4*), the amount of commodity consumption in time unit for each store-house of addressee, the capacity of each vehicle (15), vehicle speed (50 and 75 in empty return journey) the number of vehicles for each road could be easy calculated. We have take into account some redundancy due to the fact of car failure (we assumed that the time between failures is 2000 time units) what results in following number of vehicles: road one n_1=40, road two n_2=12, road three n_3=18(A)+6(B)=24 and road four n_4=16(A)+8(B)=24. The analysis time T was equal to 20000. We have analyzed maintenance and service level agreement (SLA) dependency. From one side the transport network operator has to fulfil some service level agreement, i.e. have to deliver commodity in such way that a "no commodity state" is lower than a given stated level. Therefore the analyzed functional measure was a summary time of "no commodity state" during the analyzed time period. It could be only done if a proper maintenance agreement is signed. Therefore the argument of analyzed dependency was an average time of repair of vehicles. We assumed that we have four separated maintenance agreement, one for each for each road (roads 1 and 2 with one maintenance crew, and 3 and 4 with two maintenance crews). Also the exponential distribution of repair time was assumed. Therefore, we have four free parameters with values spanning from 1 to 1200. The system was simulated in 1500 points. For each repair time values set the simulation was repeated 25 times to allow to get some information of summary time of "no commodity" distribution. Two measures were calculated: average time of summary of "no commodity state" and its 4% quantile (i.e. the value of summary "no commodity" time that with probability 96% could be not higher). The achieved data from simulation was divided randomly into two sets: learning and testing. We have used the Multilayer Perceptron architecture with 4 input neurons which correspond to repair time for each road, 10 hidden layer neurons and 2 output neurons corresponding to calculated measures (average time of summary of "no commodity state" and its 4% quantile). The number of neurons in the hidden layer was chosen experimentally. Such network produced best results and higher numbers did not give any improvement. The tan-sigmoid was used as a transfer function in hidden layer and log-sigmoid output layer. Besides that, the output values have been weighted due to the fact the log-sigmoid has values between 0 and 1. The network presented above was trained using the Levenberg-Marquardt algorithm [3]. The achieved results, the mean of absolute value of difference between neural network results (multiplied by time range: 20000) and results from Monte-Carlo simulation, for testing data set was 364 time units and 397 respectively for an average time of summary of "no commodity state" and its 4% quantile. It is in a range of 1-2% of the analyzed transport system time. We have also tested the simulation answer stability, i.e. the difference between two different runs of Monte-Carlo simulations (25 of them each time) for both functional measures (average time of summary of "no commodity state" and its 5% quantile) was 387 time units in an average. Therefore, the neural networks outputs are on the same level of accuracy as Monte-Carlo simulation since it was used for training the neural network the results could not be better. Whereas, there is no comparison between calculation time since the calculation of neural network outputs is negligible compared to Monte-Carlo simulation.

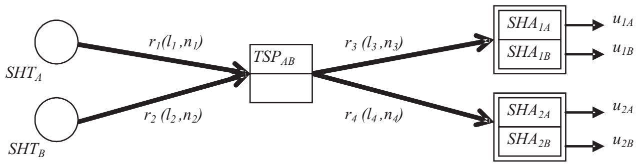

Figure 4. Structure of analysed system – functional parameters: l_1=120, l_2=90, l_3=90, l_4=120, u_{1A}=60, u_{1B}=20, u_{2A}=40, u_{2B}=20

4.4. Softcomputing approach to dispatching

As it was mentioned in the introduction we also proposed the management system based on a neural network based [24]. The system consists of a Multilayer Perceptron to decide if and where to send trucks. The input to the neural network consists of:

$$in = \left\langle pkc_1, pkc_2, ..., pkc_{npk}, crc_1, crc_2, ..., crc_{npk}, nfv \right\rangle, \quad (8)$$

npk – number of nodes of *DTS* infrastructure,
pkc_i – number of containers waiting for delivery with destination address set to *i*-th node,
nfv – number of free vehicles in the vehicle base,
Each output of the network corresponds to each node:

$$nnout = \left\langle out_1, out_2, ..., out_{npk} \right\rangle, \quad (9)$$

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

$$j = \arg\max_{i=1...npk}\{out_i\}. \quad (10)$$

If out_j is greater than 0.5 send a vehicle to node *j* else do nothing. If there are more vehicles available in the base, the largest vehicle that could 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 given moments in time. These moments are defined by following states of the system:

- the vehicle comes back to the base and is ready for the next trip,
- if in base there is at least one available vehicle 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. Important are results of the set of decisions. Since the business service realised by transport system is to move commodities without delays, the neural network should take such decisions that allows to reduce delays as much as possible. To train neural network to perform such task we propose to use genetic algorithm [20, 24]. 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 business service requirements of transport system we propose following definition of the fitness function calculated for a given neural network after some time (*T*) (therefore after a set of decisions taken by neural network):

$$fitness(T) = \frac{N_{ontime}(0,T) + N_{ontimeinsystem}(T)}{N_{delivered}(0,T) + N_{insystem}(T)}. \quad (11)$$

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

5. Conclusion

Results of functional and reliability analysis of exemplar discrete transport system are very promising. The best neural network could estimate the time of being in a given state with an error much smaller then one day in one year period. The time needed to achieve the answer is very fast. Therefore, the network could be used in any software package supporting decision process based of a reliability analysis. Time necessary for whole neural network training is less (in average 4 times) then time necessary for a single training vector preparation (run of 25 simulations for a single set of free parameters). An error related to the network answer - when the already trained network is tested by the input data which are not used during training - is in the range of disperse related to results of simulation. Of course there is an important aspect of avoiding over fitting or under training by neural network. At this stage of work it was done manually by observing the global error in function of training epochs and stopping training when the curve stops to decrease. The other interesting aspect of presented approach is the scalability projections. Increasing the number of modeled vehicles or system elements increases the Monte Carlo simulation time significantly. In case of training time of neural network (classification time is negligible) increasing a number of simulated entities has not direct influence. However, if one wants to analyze more sophisticated relation between input parameters and output measures, i.e. increases the number of input parameters, it results in an increase of input neurons, therefore needs a larger number of training data and results in a longer training time. The network solution is not free of problems. The main disadvantage is that the particular net is correctly fixed only to a single structure of the system. For each model different neural network must be trained. The overall structure of the net could be the same but a new set of weights must be estimated. Future work is planned on checking the extrapolation features of the neural network. We are going to analyze the answer of the network for input data with range outside the training set.

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] Banks, J., Carson, J.S. & Nelson, B.N. (1996). *Discrete-Event System Simulation*. 2nd Edition. Prentice Hall, Upper Saddle River, NJ.

[2] Barlow, R. & Proschan, F. (1996). *Mathematical Theory of Reliability*. Philadelphia: Society for Industrial and Applied Mathematics.

[3] Bischop, Ch. (1996). *Neural Networks for Pattern Recognition*. Clarendon Press Oxford.

[4] Burt, C.N. & Caccetta, L. (2007). Match Factor for Heterogeneous Truck and Loader Fleets. *International Journal of Mining, Reclamation and Environment*, Vol. 21, 262-270.

[5] Caban, D. & Walkowiak, T. (2002). Computer Simulation of Discrete Transport System. *Proc. XXX Winter School of Reliability*, 93-103.

[6] Caban, D. & Zamojski, W. (2008). Dependability Analysis of Information Systems with Hierarchical Reconfiguration of Services. *Proc. the Second International Conference on Emerging Security Information, Systems and Technologies*, 350-355.

[7] Duinkerken, M.B., Dekker, R., Kurstjens, S.T.G.L., Ottjes, J.A., & Dellaert, N.P. (2006). Comparing Transportation Systems for Inter-Terminal Transport at the Maasvlakte Container Terminals. *OR Spectrum*, Vol. 28, 469-493.

[8] Fishman, G. (1996). *Monte Carlo: Concepts, Algorithms, and Applications*. Springer-Verlag.

[9] Gartner, N., Messer, C.J. & Rathi, A.K. (1998). *Traffic Flow Theory and Characteristics*. In: T.R. Board (Ed.). Texas: University of Texas at Austin.

[10] Ioannou, P.A. (2008) *Intelligent Freight Transportation.* Carolina: Taylor and Francis Group.

[11] Liu, H., Chu, L. & Recker, W. (2004). Performance Evaluation of ITS Strategies Using Microscopic Simulation. *Proc. of the 7th International IEEE Conference on Intelligent Transportation Systems* 255-270.

[12] Mazurkiewicz, J., & Walkowiak, T. (2004). *Fuzzy Economic Analysis of Simulated Discrete Transport System*. In: Rutkowski, L., Siekmann, J.H., Tadeusiewicz, R., Zadeh, L.A. (eds.) ICAISC 2004. LNCS (LNAI) Springer, Heidelberg, vol. 3070, 1161–1167.

[13] Michalska K., Zamojski W. & Mazurkiewicz J. (2009. Functional-Dependability Metrics for Business Service Oriented Complex Information Systems. *Proc. Mendel 2009: 15th International Conference on Soft Computing*, Brno, Czech Republic, 234-238.

[14] Podofillini L., Zio E. & Marella, M. (2005). A multi-state Monte Carlo Simulation Model of a Railway Network System. In Advances in Safety and Reliability. *Proc. European Safety and Reliability Conference - ESREL 2005,* 1567-1575.

[15] Sanso, B. & Milot, L. (1999). Performability of a Congested Urban-Transportation Network when Accident Information is Available. *Transportation Science*, Vol. 33, No 1. 10-21.

[16] Taylor, M.A.P., Woolley, J.E. & Zito, R. (2000). Integration of the Global Positioning System and Geographical Information Systems for Traffic *Congestion Studies.* Transportation Research, Part C (Emerging Technologies), Vol. 8C, 257-285.

[17] Vis, I.F.A. (2006). Survey of Research in the Design and Control of Automated Guided Vehicle Systems. *European Journal of Operational Research,* Vol. 170. 677-709.

[18] Walkowiak, T. & Mazurkiewicz, J. (2010). *Algorithmic Approach to Vehicle Dispatching in Discrete Transport Systems*. In: Technical approach to dependability /Ed. by Jarosław Sugier, et al. Wroclaw: Oficyna Wydawnicza Politechniki Wroclawskiej. 173-188.

[19] Walkowiak, T. & Mazurkiewicz, J. (2008). Availability of Discrete Transport System Simulated by SSF Tool. *Proc. of International Conference on Dependability of Computer*

Systems, Szklarska Poreba, Poland, Los Alamitos: IEEE Computer Society Press. 430-437.

[20] Walkowiak, T. & Mazurkiewicz, J. (2008). Functional Availability Analysis of Discrete Transport System Realized by SSF Simulator. *In Computational Science – ICCS 2008, 8th International Conference,* Krakow, Poland, June 2008. Springer-Verlag, LNCS 5101, Part I. 671-678.

[21] Walkowiak, T. & Mazurkiewicz, J. (2006). Fuzzy Approach to Economic Analysis of Dispatcher Driven Discrete Transport Systems. *DepCoS-RELCOMEX'06 International Conference,* IEEE Press, Poland, Szklarska Poreba. 366-373.

[22] Mazurkiewicz, J.& Walkowiak, T. (2004). *Fuzzy Economic Analysis of Simulated Discrete Transport System*. Artificial Intelligence and Soft Computing - ICAISC 2004, Springer-Verlag, LNAI 3070. 1161-1167.

[23] Walkowiak, T. & Mazurkiewicz, J. (2004). Hybrid Approach to Reliability and Functional Analysis of Discrete Transport System. *Proc. International Conference on Computational Science,* ICCS 2004, Springer-Verlag, LNCS 3037. 241-248.

[24] Walkowiak, T. & Mazurkiewicz, J. (2010). Soft Computing Approach to Discrete Transport System Management. *In: Lecture Notes in Computer Science. Lecture Notes in Artificial Intelligence.* Springer-Verlag, Vol. 6114. 675-682.

[25] Zamojski, W. (2009). Dependability of Services Networks. *Proc. of the Third Summer Safety and Reliability Seminars - SSARS 2009,* Vol. 2, 387-396.

[26] Zamojski, W. & Caban, D. (2006). Impact of Software Failures on the Reliability of a Man-Computer System. *International Journal of Reliability, Quality, and Safety Engineering,* Vol. 13, No. 2. 149-156.

[27] Zamojski, W. & Caban, D. (2007). Maintenance Policy of a Network with Traffic Reconfiguration. *Proc. of International Conference on Dependability of Computer Systems.* DepCoS - RELCOMEX 2007, Szklarska Poręba, Poland, IEEE Computer Society Press. 213-220.