Simulation Modelling Practice and Theory 84 (2018) 222–231

Contents lists available at ScienceDirect

Simulation Modelling Practice and Theory

journal homepage: www.elsevier.com/locate/simpat

Serviceability of large-Scale systems

Zsolt T. Kosztyán

University of Pannonia, Department of Quantitative Methods, Hungary

ARTICLE INFO

Article history:
Received 22 November 2017
Revised 5 March 2018
Accepted 7 March 2018
Available online 10 March 2018

Keywords:
Network science
System reliability
Serviceability

ABSTRACT

One of the most important research fields of network sciences is the robustness of networks. A recently answered important question was the following: Which network topologies are more resistant to random malfunctions and/or direct attacks? Nevertheless, until now, “which system topology can be maintained and how to manage maintenance more efficiently and effectively” have been open questions. However, these questions are the keys both to designing large-scale systems and to scheduling maintenance tasks. This paper proposes a new means to analyze the maintainability of a large system by combining two kinds of networks, i.e., the reliability diagram of the system (1) and the network of scheduled maintenance tasks (2). This paper shows how to assign maintenance task(s) to a system component to increase the reliability of the system. With the proposed method, the maintainability of large-scale systems can be analyzed.

1. Introduction

The robustness and resistance of networks are widely studied in network science (see [1] for an excellent review). Scholars showed that so-called small-world[1] (hereinafter SW, e.g., (electrical) power grids, see [3]) and scale-free (SF, e.g., the Internet and social networks, see [4]) networks are more resistant to random failures than random networks [4,5].

SW and SF networks have common features (see [6] for a great synthesis). These networks can be measured by the average shortest path length, as these networks allow limiting the number of stops (intermediate nodes) between two given nodes, on average. In addition, these networks contain many hubs (bridge nodes) [7]. However, SW and SF networks contain only a few large degree nodes (hubs, in this study, power plants); therefore, these networks (similar to power grids) are slightly resistant to direct attacks [8]. The distribution degree of the SF and, usually, SW networks follows a power law, at least asymptotically. That is, the fraction $P(k)$ of nodes in the network having k connections to other nodes obeys $P(k) \sim k^{-\gamma}$, where γ is a parameter that is typically in the range of [2, 3] for SF networks, although it may occasionally lie outside these bounds. The structure of the power grid can be characterized usually as a planar network (meaning edges do not cross each other). This network is an SF network instead of an SW network; however, the degree of distribution can also follow a power function, and the typical parameter $\gamma \in [1, 2]$. A planar network is more physically constrained and thus is more assortative, with a higher probability of containing a giant component (i.e., a connected subgraph containing a majority of the nodes) [7]. Similar to the SF networks, these networks are also less vulnerable to random failures than random networks and slightly more resistant to a direct attack than SF networks [9].

E-mail addresses: kzst@gtk.uni-pannon.hu, kzst@vision.vein.hu

[1] SW networks exhibit a small average path length between pairs of nodes. For an excellent classification of small-world networks, see Amaral et al. [2].

https://doi.org/10.1016/j.simpat.2018.03.002
1569-190X/

Owing to the knowledge of network topologies, we can now generate robust networks [8], or we can improve an existing system network (see, e.g., [10,11]) to make them more resistant. Adequate network topologies can help us reduce the probability of system failures. However, an open question remains: which *kinds of large-scale systems can be maintained more effectively? Are more robust large-scale systems more effectively maintainable?*

The challenge of this exploration is to address different kinds of structures at the same time. For example, to schedule maintenance tasks,[2] at least two different kinds of structure are usually required. One structure (or one network) (1) is needed to describe the system (e.g., the production system, the Internet, or the electrical power grid), and the other structure (2) is needed for specifying the relation of activities. Therefore, instead of only network thinking (which focuses only on one structure), we need multi-structural thinking. In addition to network topology, we should focus on the interaction between structures. In maintenance management, we need at least two different kinds of structures (see Fig. 1), which are variously organized.

1. The reliability block diagram (RBD)[3] of a production system can be described as a simple so-called serial/parallel (hereinafter S/P) network [15], while a power grid, the Internet and a communication network can follow a scale-free (SF), a small-world (SW) or other networks [5].
2. Small- to large-scale plans can be characterized as the so-called random networks.[4]

In a maintenance plan for all (parts) of system components, we assign at least one so-called corrective/preventive maintenance task (see Fig. 1). If these maintenance activities are completed, the reliability or availability of the maintained equipment will be increased. Therefore, these structure elements impact each other.

The other challenge of using networks in maintenance management is that the maintenance plans should be characterized as a flexible logic plan (as in [17]). Completing all possible maintenance activities has rarely occurred. Instead, the task is to select adequate maintenance tasks to improve system reliability and/or system availability while maintaining budget and deadlines.

A maintenance task can be completed by different means, the so-called completion mode. Generally, we can assume that a lower task duration requires higher cost, and higher growth of component reliability takes more time. If the set of activities is fixed, this problem is a discrete version of a time-quality-cost trade-off problem, where the quality parameter is the growth of component reliability. The discrete version of the time-cost trade-off problem is currently an NP-hard problem [18]; however, in this case, there are additional quality (i.e., growth of component reliability) parameters assigned to maintenance tasks. Moreover, the problem is further complicated by addressing flexible dependencies and uncertain task occurrences. Since [19]'s algorithm can address maintenance plans, this method can only be used for maintaining production systems, which can be characterized as serial-parallel networks.

In this paper, we extend this algorithm to analyze the maintainability of large-scale systems and support decision-makers in finding the most adequate multi-structure.

2. Analyzing the serviceability of the system

In this paper, we focused on the serviceability of the different kinds of system.

According to Blanchard et al. [20], *Maintainabilitydetermines the probability that a failed equipment, machine (=system component), or a system can be restored to its normal operable state within a given make-span, using the prescribed practices and procedures. Its two main components are:*

- *Serviceability (ease of conducting scheduled inspections and servicing) and*
- *Repairability (ease of restoring service after a failure).*

This paper mainly focuses on *serviceability*. In accordance with Lam et. al [12], we consider the time, cost and resource constraints of a schedule of maintenance tasks. At the same time, upon specifying resource availabilities, budgets and deadlines, the minimal growth of system reliability[5] is also specified. The target function was to find the minimal make-span considering the given budget of a schedule and the minimal growth of the system reliability.

2.1. Calculating system reliability

To schedule maintenance tasks, the first step is to characterize the system. One of the most frequently used modeling techniques is the reliability block diagram (RBD). To model a simple production system, an RBD is drawn as a series of blocks connected in parallel or in sequential configuration. Each block represents a component of the system with a failure rate. Parallel paths are redundant, meaning that all the parallel paths must fail for the network to fail. This "redundancy" is important for critical equipment, e.g., in a power plant.

[2] See an excellent review of the specialties of scheduling maintenance tasks in [12].

[3] The reliability block diagram (RBD) is a diagrammatic method for showing how component reliability contributes to the success or failure of a system (see, e.g., [13,14]).

[4] For further literature about random networks, see [16].

[5] In this study, the quality parameter considered is the growth of system reliability.

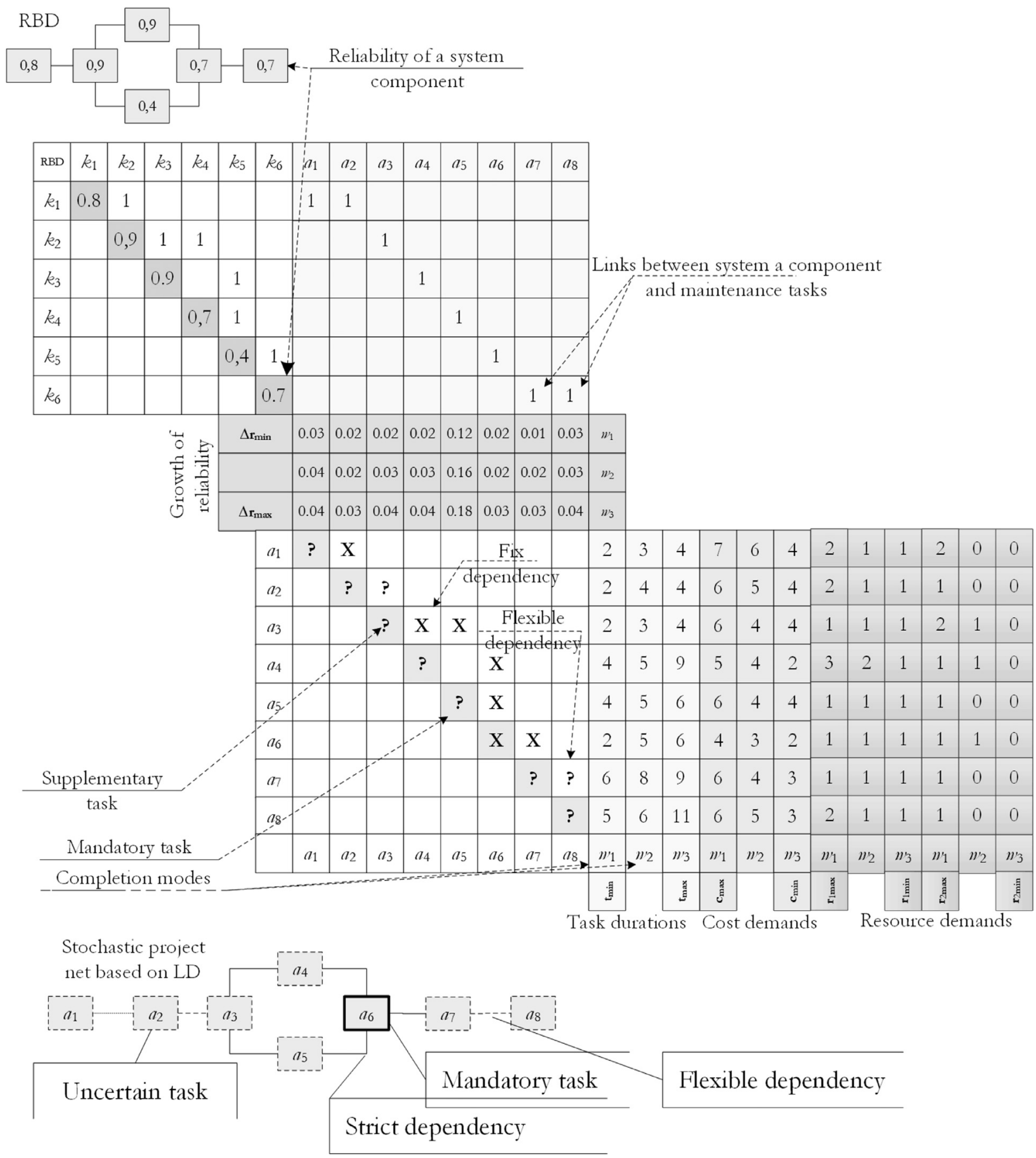

Fig. 1. The applied matrix-based maintenance management model.

In contrast, any failure along a sequential path causes the entire series path to fail. For example, for equipment in a production line that instead can be characterized as a serial (= sequential) path, if any equipment is damaged, this causes the failure of production (on that line).

Since most production systems can be characterized as an S/P network, the system reliability can be determined by the *network reduction method* (NRM) [15]. The method's steps reduce the size of the network while preserving its reliability. This method can also be applied to evaluate the reliability of a particular S/P network within the polynomial computational time demand. If the system can be reduced to S/P configurations (we can say it is an S/P reducible network), it is a relatively simple matter to determine the mathematical or analytical formula that describes the system's reliability [15].

Although most production systems' RBD can be characterized as an S/P network, for several other cases, when not only a production system or a production line is modeled, neither the serial nor the parallel configuration can describe the component-system relationship.

Nevertheless, for evaluation of (general) system reliability, further methods should be applied (see, e.g., [15,21,22]). Several methods exist for obtaining the reliability of system (a detailed review can be read in [15]); however, to determine the advantages of a system reliability evaluation on scale-free (SF) and small-world networks [23], we used a modification of the decomposition method (DCM).

DCM is an application of the law of total probability. The application involves choosing a "key", which is usually a high degree component, then calculating the reliability of the system twice: once as if the key component failed and once as if the key component succeeded. These two probabilities are then combined to obtain the reliability of the system, as at any given time, the key component will fail or operate. This calculation property of DCM is a great advantage when evaluating the system reliability of SF and SW networks or other networks that follow a power law.

Since in SF and SW networks, we have high degree nodes, after the decomposition, we obtain certain independent subnetworks; therefore, DCMs can be used to calculate large-scale scale-free networks' system reliability. Although applying DCM is suitable for low-clustered[6] scale-free networks (such as power grids), this method is excessively slow when used for high-clustered networks (such as onion networks) or S/P networks. Therefore, a combination of DCM and NRM is used. After decomposition, NRM is used to reduce the serial/parallel components. If a subsystem is a serial/parallel network, it can be calculated using NRM. Nevertheless, NRM can amass the serial/parallel components of the network, and if a component cannot be reduced by NRM, DCM can decompose it [15]. Repeating the reducing phase (by NRM) and the decomposition phase (by DCM), the reliability of high-clustered networks (such as certain kinds of onion networks) can also be calculated quickly.

For evaluating the system reliability of other networks, other methods, such as the path tracing method (PTM) and the event space method (ESM), can also be used. Since not only serial/parallel but also scale-free and onion networks are used for modeling systems, the combination of DCM and NRM is applied to calculate system reliability.

All of these methods focus only on the evaluation of the system reliability through the system topology and the component reliability; however, they do not address how to maintain the system and how to increase the system reliability. The possible maintenance activities (or tasks) should be organized into a flexible logic plan. Decision makers must make decisions, which necessitates that maintenance tasks should be completed to increase the system reliability. The selected activities should be completed within a specified timeframe and budget and should consider the resource availabilities.

2.2. Modified matrix-based maintenance management model

This paper modifies the matrix-based maintenance management model (hereinafter M^4) (see [19]) and the matrix-based model to address two kinds of structures (reliability diagram of the system structure (1), such as a reliability diagram of a production line or a power grid system, and a maintenance plan (2)). The original version of M^4 focuses only on production lines that could be characterized as an S/P network; therefore, only NRM is used to calculate the reliability or, in other cases, the availability of a system. In this extension of the original approach, we must consider SF, SW and onions networks; therefore, the modified DCM is applied instead of NRM for the calculation of the system reliability. The other problem with the original M^4 is that the method was limited to medium-scale plans ($\sim$75 tasks or equipment) because of the large number of possible maintenance plans. In our current study, larger plans ($\sim$50000 tasks or equipment) should also be investigated; therefore, at the simulation phase, the optimization needed to be parallelized by using CUDA technologies, which was implemented by Matlab. With the Parallel Computing Toolbox, a cluster (containing 16 computers) is speed up to accelerate the calculations.

In M^4, to all equipment (or, more generally, system components), at least one maintenance activity can be assigned. Nevertheless, few are selected for a maintenance plan. Therefore, we used a flexible modeling technique (see [17]). In this study, the matrix-based method is used instead of applying the traditional network-planning techniques because matrix-based methods can combine different kinds of structures in a multiple-domain matrix (MDM) (see [24]) and allow us to distinguish mandatory and to be determined (hereafter undecided) maintenance activities (see [17]). Activities in the first group must be completed. For example, the system component reliability is under the specified threshold. These activities are static components of a possible scenario of a maintenance plan. At the same time, a scenario can contain undecided maintenance activities if the specified maintenance plan based on selected activities can maintain the time/cost/quality/resource constraints.

After using this (parallelized) M^4 algorithm, we will obtain a maintenance plan that contains the set of maintenance tasks. These tasks are proposed to be scheduled to increase the system reliability with minimal duration and to maintain the budget.

Owing to the algorithm, currently not only the reliability but also the maintainability of the system can be explored. Therefore, the main question of this paper is the following: *Are there any maintainability (or more precisely, serviceability) differences among the different kinds of system structures?* (RQ1)

[6] The global clustering coefficient is the number of closed triplets (or 3 × triangles) over the total number of triplets (both open and closed).

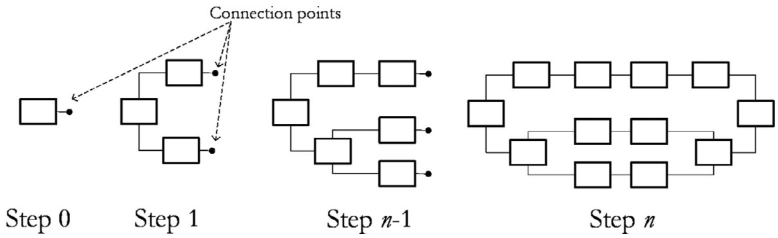

Fig. 2. The steps of generating serial/parallel networks.

Since for examining serviceability, we must consider two kinds of ((1) system and (2) task schedule) structures, another question is as follows: *Does the system structure or the constraints have more impact on serviceability?* (RQ2)

Owing to the knowledge of networks, we can improve the reliability of a system. For example, we can create an onion (high clustered) network from an SF or an SW network (see, e.g., [10,11]). Therefore, the third question is as follows: *Does the reliability improvement improve the serviceability or not?* (RQ3)

2.3. Generating structures

Two kinds of structures were generated. The first describes the system structure (i.e., the structure of a production system or the structure of power grid), and the second characterizes the schedule of tasks.

2.3.1. Generating system structures

System structures and component reliability values are characterized by the reliability block diagram (RBD). Most production systems can be characterized by S/P networks (see Section 2.1). When generating a network, we should set the rate of the serial blocks ($S\%$). In the initial step, we generate a component with one connection point (see step 0 in Fig. 2). In every step, we generate a pure random number (s) within the interval [0, 1]. If $s > S\%$ ($s \leq S\%$), we connect a parallel (serial) block to connection points (see step 1 in Fig. 2). In the last step, the blocks will be reflected.

If $S\% = 1$, we will only obtain serial blocks; in contrast, if $S\% = 0$, we obtain two interconnected binary trees.

The Internet and the power grid instead follow SF and SW networks. When generating an SF or SW network, we followed Barabási et. al. [4] network growing method. The network begins with an initial connected network of n_0 nodes. New nodes are added to the network one at a time. Each new node is connected to $n \leq n_0$ existing nodes with a probability proportional to the number of links that the existing nodes currently have.

Formally, the probability p_i that the new node is connected to node i is $p_i = \frac{k_i}{\sum_j k_j}$, where k_i is the degree of node i, and the sum includes all pre-existing nodes j (i.e., the denominator results in twice the current number of edges in the network). We generated structures using the poweRlaw package of the R program, where the desired parameter was $\gamma \in \{1.5, 2.0, 2.5\}$. $\gamma = 1.5$ characterizes an SW network, whereas $\gamma = 2.5$ characterizes an SF network.

Heavily linked nodes (so-called “hubs”, in this simulation, e.g., power plants) tend to quickly accumulate more links, while nodes with only a few links are unlikely to be chosen as the destination for a new link. The new nodes have a so-called “preference” to attach themselves to the currently heavily linked nodes.

This algorithm can characterize different kinds of networks [2,4], and this method is also good for characterizing power grids [25]. These networks are more resistant to random failures but less resistant to a direct attack [5].

Therefore, one of the new directions is to improve a network such that it is more resistant to both random failures and direct attacks. The main problem is that if a hub (a high-degree node) is malfunctioning or being attacked, the system can be fragmented [25]. This fragmentation can cause a system failure.

The idea of improving the system reliability is to connect high degree nodes to each other to obtain a more resistant network. This network will be a so-called onion network (see, e.g., [1,10,11]). We used Wu & Holmes’ algorithm [10], where the initial network was a generated SF or SW network, and the output was the most robust onion network.

In all simulation cases and for all structures, the components’ reliability followed a uniform random distribution within the interval [0, 1].

2.3.2. Generating schedules

Most generators (see, e.g., ProGen by Kolisch and Sprecher [26]) generate quasi-random networks of logic plans. In real random networks, every node can be connected to another with the same probability; however, in a schedule, we use directed connections (as dependencies). We can specify the number of stages (N) and maximal number of nodes (n_i) in stage $I, J = 1, 2, .., N$.. To avoid direct cycles, only arcs between a node from stage I and a node from stage J is allowed if $I < J$. Therefore, this structure of scheduled tasks is a so-called quasi-random structure.

We simulated logic plans using ProGen, and the completion sequence of a set of maintenance activities was assumed to be changeable. Therefore, we generated 10 possible completions[7] with the same maximal task time/cost/resource[8] demands. After performing ProGen, we obtained a set of 10 completions with different dependencies and different kinds of time/cost/resource demands. If a dependency between two activities occurred in all completions, this dependency was considered a strict dependency ("X", as in Fig. 1). If a dependency between two nodes never occurred, we ignored this dependency from the schedule (see " " or the empty cells in Fig. 1). However, if a dependency occurred at least once, it was considered an undecided dependency ("?" in Fig. 1). The simulation was followed by the recognition that maintaining equipment can be completed separately and independently [27]; therefore, the ratio of strict dependency between two tasks was only 10%.

A larger plans, such as system shutdowns, requires more mandatory tasks than smaller ones such as continuous preventive maintenance plans. Therefore, to characterize the differences between a system shutdown and continuous maintenance, the ratio of non-mandatory maintenance tasks was $P\% \in \{50\%, 60\%, \ldots, 90\%\}$. Not all the activities were conducted in the event of system shutdown; nevertheless, in continuous maintenance, less repair activity will be performed in a smaller maintenance plan.

After specifying the flexible maintenance plan time/cost, resource demands were randomly selected from the set of completions simulated by ProGen in the previous step.

3. Results

After generating schedule and system structures, the minimal/maximal make-span (TPT = total process time)/cost (TPC = total process cost), the vector of maximal values of the resource demands (total process resource **TPR**) and the minimal/maximal growth of system reliability (ΔTSR = the growth of total system reliability) can be calculated as follows:

TPT_{min} occurs if every flexible dependency is omitted and only mandatory activities are completed within the minimal (so-called crash) task duration [17].

TPT_{max} occurs if every flexible dependency is realized and both mandatory and undecided activities are completed within the maximal (i.e., normal) task duration [17].

TPC_{min} occurs if only mandatory tasks are completed utilizing minimal costs [17].

TPC_{max} occurs if both mandatory and undecided tasks are completed utilizing maximal (so-called crash) costs [17].

$\mathbf{TPR}_{min}$ occurs if every flexible dependency is realized but only mandatory activities are completed within the maximal task duration [17].

$\mathbf{TPR}_{max}$ occurs if every flexible dependency is omitted and both mandatory and undecided tasks are completed within the minimal task duration [17].

$\Delta\mathrm{TSR}_{min}$ occurs if only mandatory tasks are competed and the impact of maintenance activities is minimal.

$\Delta\mathrm{TSR}_{max}$ occurs if both mandatory and undecided tasks and the impact of maintenance activities is maximal.

To compare results, we use constraint ratios, which can be calculated as follows:

$$C_x\% = \frac{C_x}{X_{\max} - X_{\min}} \cdot 100 \qquad (1)$$

where $X \in \{\mathrm{TPT}, \mathrm{TPC}, \Delta\mathrm{TSR}\}$; $C_x \in \{X_{\min}, X_{\max}\}$ is the constraint. Note that $X \in \{\mathrm{TPT}, \mathrm{TPC}, \Delta\mathrm{TSR}\}$ specifies a traditional (so-called iron) triangle (see, e.g., [28]), where TPT is the make-span, TPC is the realized budget, and ΔTSR characterizes the quality of the schedule. In this simulation, $C_x\% \in \{50\%, 60\%, \ldots, 90\%\}$. Different kinds of constraints can be set for the different kinds of resources. However, to avoid generating a large number of cases, resource constraints are set to be 75% for all resources.

In addition to the schedule parameters, system structural parameters are considered. The number of equipment (n) in the first simulation was $n \in \{25, 50, 75\}$; however, for exploring large-scale networks, $n \in \{25000, 50000, 75000\}$. The ratio of non-mandatory activities in both simulations was $P\% \in \{50\%, 60\%, \ldots, 90\%\}$.

In each simulation, we investigated the maintainability (specifically, the serviceability) of both small-scale and large-scale structures. Therefore, the target was to complete maintenance tasks as soon as possible within a given timeframe and budget; in addition, the growth of the system reliability must be greater than a specified value. We constructed a maintainability ratio as follows:

$M\% = 1 - \frac{\mathrm{TPT} - \mathrm{TPT}_{min}}{\mathrm{TPT}_{max} - \mathrm{TPT}_{min}} \in [0, 1]$, where TPT is the make-span of the optimal solution. If this value is 1, it means that we can find the minimal make-span within the constraints.

The scope of the simulation results was to answer the research questions (RQ1-RQ3).

After simulation, we validated the answers of RQ1-RQ3 considering the Hungarian power grid structure; see Fig. 4.

[7] Where the coefficient of network complexity (arcs/nodes) is set to be 1.3.

[8] In this simulation, the number of resources is set to be 3.

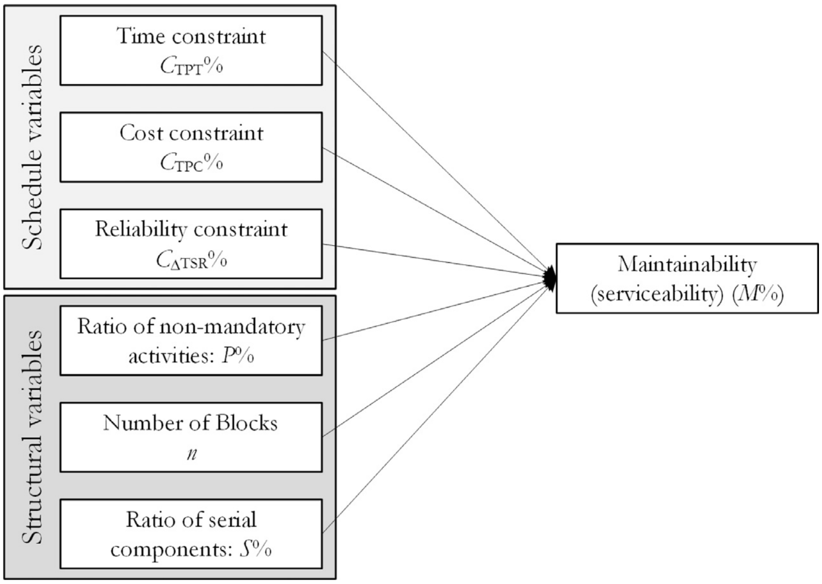

Fig. 3. Dependent and independent variables for simulation.

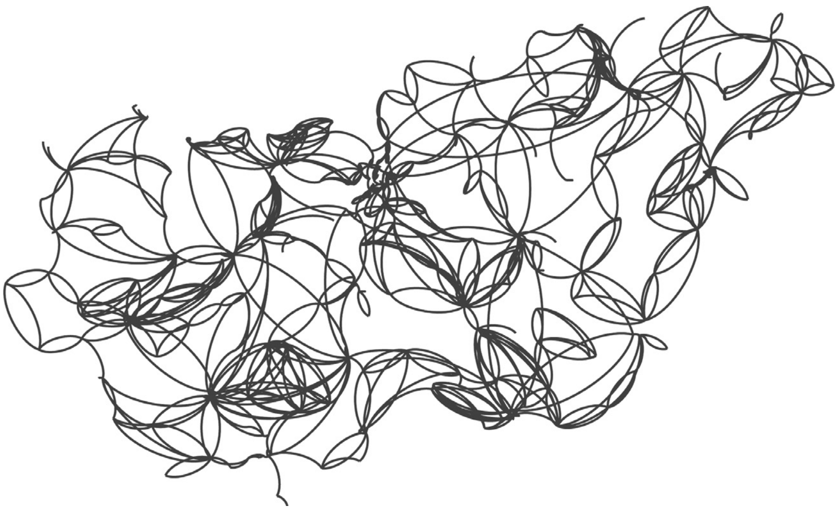

Fig. 4. Hungarian power grid topology (where the load of electric wires is greater than 10 kV; average path length = 3.74).

3.1. Simulation results

Three kinds of simulations were performed. In the first simulation of only serial/parallel systems, structures were generated to characterize the production system (see the research model in Fig. 3). In this case, the ratio of serial components was $S\% \in \{50\%, 60\%, \ldots, 100\%\}$.

In the second simulation, we compared the results between serial/parallel (S/P) and small-world/scale-free (SW/SF) networks. To compare the maintainability of these two structures, the coefficient of network complexity (CNC) was the same for both S/P and SW/SF networks.

In the third simulation, in accordance with Wu & Holmes' algorithm [10], we constructed an onion network from a scale-free network and compared the maintainability.

3.1.1. Analyzing the maintainability of S/P networks

Table 1 shows parameter estimates of the simulation result.

The positive significant coefficients show that if we increase these values, we can improve the maintainability. Table 1 shows that structural parameters are more important than the constraints (see RQ3).

Table 1
Parameter estimation (* significance level is less than 0.05, *** significance level is less than 0.0001, adj.R^2 = 0.6466, $N = 16200$).

Parameter estimates					
Term	Std.err.	t value	Prob > |t|	Std.beta	Rel.imp.%
Intercept***	4.210E-04	2270.935	< 2E-16	0.000E-00	–
$C_{TPT}\%^{*}$	2.312E-04	−2.338	0.0194	−1.083E-02	0.018%
$C_{TPC}\%$	2.312E-04	0.308	0.7583	1.426E-03	0.001%
$C_{\Delta TSR}\%^{***}$	2.792E-04	−14.909	< 2E-16	−6.910E-02	0.732%
$P\%^{***}$	2.792E-04	53.523	< 2E-16	2.481E-01	9.436%
n^{***}	1.934E-06	163.580	< 2E-16	7.582E-01	88.144%
$S\%^{***}$	2.312E-04	−22.512	< 2E-16	−1.043E-01	1.669%

Table 2
Total system reliability, with equipment reliability following a uniform distribution.

Structure	Mean	Median	Std.dev.	Min.	Max.
S/P	8.749E-03	3.337E-03	1.579E-02	1.606E-05	1.970E-01
SF/SW	2.698E-01	2.599E-01	1.045E-01	2.742E-02	6.810E-01

Table 3
The results of parameter estimates, *** significance level is less than 0.0001, adj.R^2 = 0,5091, $N = 5400$.

Parameter estimates					
Term	Std.err.	t value	Prob > |t|	Std.beta	Rel.imp.%
Intercept***	3.182E-03	297.991	< 2.000E-16	0.000E-01	–
$C_{TPT}\%$	1.906E-03	−1.879	6.030E-02	−1.792E-02	0.063%
$C_{TPC}\%$	1.906E-03	−0.260	7.948E-01	−2.481E-03	–
$C_{\Delta TSR}\%^{***}$	2.302E-03	−23.874	< 2.000E-16	−2.277E-01	10.173%
$P\%^{***}$	2.302E-03	23.874	< 2.000E-16	2.277E-01	10.177%
n^{***}	1.595E-06	−4.189	2.850E-05	−3.994E-02	0.313%
'S/P', 'SW/SF'***	6.510E-04	66.643	< 2.000E-16	6.355E-01	79.274%

Since a higher flexible ratio ($P\%$)[9] instead models continuous maintenance, the positive coefficient of $P\%$ shows that *continuous maintenance can be more effective* (see RQ2).

It is easy to realize that more parallel blocks (less serial blocks) in a system produce higher system reliability; however, considering the negative coefficient of $S\%$, we obtain an interesting result. The effectiveness of maintenance is higher in a more reliable system, where $S\%$ is lower (see RQ1).

The negative coefficient of the increase of the reliability parameter ($C_{\Delta TSR}\%$) is trivial. If we increase the minimal improvement of the system reliability, the distance between the feasible make-span and the ideal, which equals the minimal make-span, will be increased.

3.1.2. Exploring the maintainability of scale-free/small-world large-scale networks

If the structure of the system can be characterized as a scale-free or a small-world network, the total system reliability is significantly higher than that for the traditional serial/parallel systems (see Table 2).

Our results confirm the results of [29] that scale-free/small-world networks are more reliable than random or S/P networks. SF/SW networks are more resistant against random failures [30]. However, *the novel result was that not only the system reliability but also the system serviceability mainly depends on the system structure* (see RQ3 and the greatest relative importance value for the system topology ('S/P' vs 'SF/SW')).

The different kinds of schedule and structure variables specify 5400 simulations. Table 3 shows that only one schedule variable is significant, and the structure parameters are more important than the schedule parameters.

One of the most interesting results is that the changes of the cost and time constraints are not significant in this model. *The other considerable result is that system structures that follow power law networks can be maintained faster than traditional serial/parallel system structures* (see RQ1).

The results show us that if a maintenances plan is flexible, the make-span can be more significantly reduced (see RQ2).

[9] = the ratio of non-mandatory tasks.

3.1.3. Exploring the maintainability of onion networks

In accordance with Wu and Holme's [10] algorithm, we constructed onion networks from the former specified SF/SW networks, and we compared their maintainability.

Our findings consist of Parandehgheibi and Modiano's [5] results. Our simulation shows that the mean of the transformed onion networks' system reliability is 82.8% more (nearly twice as) reliable as the original SF/SW networks, while the maintainability of the onion network ($M_{\text{'O'}}\%$) is significantly better than the maintainability of the SF/SW networks ($M_{\text{'SF/SW'}}\%$); in addition, the improvement ratio is $\frac{M_{\text{'O'}}\%}{M_{\text{'SF'}}\%} = 1.08$.

The simulation result also shows us that the more reliable networks' serviceability is higher (see RQ1).

3.2. Application of the study

Several scholars showed that the power grid is a typical example of a planar SW network (see [4]). Most of the real networks' degree distributions follow a power law of the form $P(k) \approx k^{-\gamma}$ with the exponent γ. In Rosas-Casals et al.'s [25] publication, the average exponent $\gamma = 1.8$ (where the minimal one was in the United Kingdom, $\gamma_{\text{UK}} = 0.91$, and the maximal one was in Portugal, $\gamma_{\text{PT}} = 2.72$). Rosas-Casals et al. [25] showed that the difference in the network response to errors and attacks appears to be related only to network size and not to other topological measures of network complexity such as the mean degree or the exponent γ of the degree distribution. Therefore, until now, we addressed SW/SF networks that were treated uniformly.

Nevertheless, in this study, we addressed maintainability instead of reliability or resistance. We compared the maintainability of the power grids of Portugal, the United Kingdom and Hungary. For the first two countries, we used Rosas-Casals et al.'s [25] published data; however, for Hungary, we used the 2015 power grid structure (see Fig. 4). The exponent was $\gamma_{\text{HU}} = 1.935$. The average weighted degree was 2.51. The average path length was 5.01. The next study focused on the serviceability analysis of the countries' power grids; however, we found that the serviceability is different if we consider the UK's, Portugal's or Hungary's topology. Using the proposed simulation framework, we obtain significant differences between serviceability: $\frac{M_{\text{'UK'}}\%}{M_{\text{'HU'}}\%} = 0.972$, while $\frac{M_{\text{'PT'}}\%}{M_{\text{'HU'}}\%} = 1.031$.

This result shows us that, in contrast to the reliability, the maintainability may depend on the structural parameters.

According to Section 3.1.3, in addition to the reliability, the maintainability can be improved if high-degree nodes are connected by using Wu and Holme's [10] algorithm. In this case, the maintainability of the Hungarian power grid can be improved to 5.27%.

4. Summary and conclusion

This paper addressed multi-structural networks, where system structure and the maintenance activities are investigated separately. This paper showed that the maintainability mainly influenced the topology of the system. More resistant networks' maintainability is greater than that of less reliable networks.

4.1. Implications for scholars

Exploring maintainability is one suitable example of the power of multi-structural thinking. With the proposed method, at least two different kinds of structures and the interactions between components can be modeled. The results show us that the maintainability strongly depends on the system structures (RQ1) but only slightly depends on the constraints (RQ3). The main message of this paper is that *more reliable systems can be maintained more effectively*. If we want to improve the effectiveness of the maintenance of a (large-scale) system, we should focus on the system structure, and we should make it more reliable.

Similar to the modeling agile vs traditional project management approaches (see [17]), with the flexibility parameter (rate of non-mandatory tasks), different kinds of maintenance management strategies can be modeled (RQ2). Similarly, with the improvement of the effectiveness of agile approaches in project management, the more flexible continuous maintenance plans can be more effective. Therefore, the other message is that *the flexibility of planning can improve its effectiveness.*

4.2. Implications for decision makers

The proposed multi-domain matrix-based maintenance management (M^4) technique can be used to simulate the effects of different kinds of maintenance strategies. The results show us that the continuous preventive maintenance strategy can be more effective than periodic stand downs (RQ2). This simulation result is another argument for the continuous preventive maintenance approaches, which were also proposed both in the total productive maintenance (TPM) and in the risk-based maintenance (RBM) strategies.

Decision makers and planners can mainly focus on the constraints and the completions. Therefore, in the short run, these individuals can improve the effectiveness of maintainability if they act in accordance with the flexible, continuous maintenance strategies. However, significant improvements in effectiveness can be achieved if the topology of the system structure can be more reliable.

5. Limitations and future works

The proposed framework can open multi-structural investigations. This paper focused on serviceability differences in different kinds of system structures. Only two stated (success/failure) components are considered, where only the component reliability is treated. How to model the serviceability of multi-state systems by considering degradation of the components could be an interesting question.

The proposed model showed how to connect two different structures, where one (i.e., the structure of the schedule) is stochastic. In management science problems, we must usually consider more than two structures because we need one structure for the system, one for performing the activities, and another for the organization. This multi-structural thinking can be extended to connect and model interactions on more than two different structures with different topologies and different (deterministic/stochastic) natures.

Acknowledgment

This research is supported by a János Bolyai Fellowship.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.simpat.2018.03.002.

References

[1] V.H.P. Louzada, F. Daolio, H.J. Herrmann, M. Tomassini, Smart rewiring for network robustness, J. Complex Netw. 1 (2) (2013) 150–159, doi:10.1093/comnet/cnt010.
[2] L.A.N. Amaral, A. Scala, M. Barthélémy, H.E. Stanley, Classes of small-world networks, Proc. Natl. Acad. Sci. 97 (21) (2000) 11149–11152, doi:10.1073/pnas.200327197.
[3] D.J. Watts, S.H. Strogatz, Collective dynamics of'small-world'networks, Nature 393 (6684) (1998) 440–442.
[4] A.-L. Barabási, R. Albert, H. Jeong, Mean-field theory for scale-free random networks, Physica A 272 (1–2) (1999) 173–187, doi:10.1016/S0378-4371(99)00291-5.
[5] M. Parandehgheibi, E. Modiano, Robustness of interdependent networks: the case of communication networks and the power grid, in: GLOBECOM(IEEE), 2013, pp. 2164–2169, doi:10.1109/GLOCOM.2013.6831395.
[6] M. Newman, Networks: An Introduction, First, Oxford University Press, Oxford, 2010.
[7] C. Ducruet, L. Beauguitte, Spatial science and network science: review and outcomes of a complex relationship, Netw. Spatial Econ. 14 (3) (2014) 297–316, doi:10.1007/s11067-013-9222-6.
[8] V.H. Louzada, F. Daolio, H.J. Herrmann, M. Tomassini, Generating Robust and Efficient Networks under Targeted Attacks, in: D. Król, D. Fay, B. Gabryś (Eds.), Propagation Phenomena in Real World Networks, Intelligent Systems Reference Library, 85, Springer International Publishing, 2015, pp. 215–224, doi:10.1007/978-3-319-15916-4_9.
[9] Y. Hayashi, J. Matsukubo, Geographical effects on the path length and the sobustness in complex networks, Phys. Rev. E 73 (6) (2006) 66–113.
[10] Z.-X. Wu, P. Holme, Onion structure and network robustness, Phys. Rev. E 84 (2011) 026106, doi:10.1103/PhysRevE.84.026106.
[11] T. Tanizawa, S. Havlin, H.E. Stanley, Robustness of onionlike correlated networks against targeted attacks, Phys. Rev. E 85 (2012) 046109, doi:10.1103/PhysRevE.85.046109.
[12] E.W. Lam, A.P. Chan, D.W. Chan, Benchmarking success of building maintenance projects, Facilities 28 (5/6) (2010) 290–305, doi:10.1108/02632771011031529.
[13] M. Rausand, A. Hoyland, System Reliability Theory, Models, Statistical Methods and Applications, Second, Wiley New York, 2004.
[14] D. Bhattacharjya, L. Deleris, From reliability block diagrams to fault tree circuits, Decision Anal. 9 (2) (2012) 128–137, doi:10.1287/deca.1120.0231.
[15] M. Shiker, Some methods of calculating the reliability of mixed models, J. Babylon University 21 (3) (2013) 770–774.
[16] A. Rényi, P. Erdős, On random graphs, Publicationes Mathematicae 6 (5) (1959) 290–297.
[17] Z.T. Kosztyán, Exact algorithm for matrix-based project planning problems, Expert Syst. Appl. 42 (9) (2015) 4460–4473, doi:10.1016/j.eswa.2015.01.066.
[18] P. De, E.J. Dunne, J.B. Ghosh, C.E. Wells, Complexity of the discrete time-cost tradeoff problem for project networks, Oper. Res. 45 (2) (1997) 302–306, doi:10.1287/opre.45.2.302.
[19] Z.T. Kosztyán, C. Hegedűs, J. Kiss, A. Németh, Handling Maintenance Projects with Matrix-Based Methods, Springer New York, New York, NY, pp. 357–366. 10.1007/978-1-4614-3558-7_29
[20] B.S. Blanchard, D. Verma, E.L. Peterson, Maintainability: A Key to Effective Serviceability and Maintenance Management, 13, John Wiley & Sons, 1995.
[21] Y.-M. Wang, T. Elhag, Evidential reasoning approach for bridge condition assessment, Expert Syst. Appl. 34 (1) (2008) 689–699, doi:10.1016/j.eswa.2006.10.006.
[22] A. Verma, A. Srividya, D. Karanki, Basic Reliability Mathematics, Springer London, 2010, doi:10.1007/978-1-84996-232-2_2.
[23] Z. Yuan, C. Zhao, Z. Di, W.-X. Wang, Y.-C. Lai, Exact controllability of complex networks, Nat. Commun. 4 (2013) 2447.
[24] M. Danilovic, T.R. Browning, Managing complex product development projects with design structure matrices and domain mapping matrices, Int. J. Project Manage. 25 (3) (2007) 300–314, doi:10.1016/j.ijproman.2006.11.003.
[25] M. Rosas-Casals, S. Valverde, R.V. Solé, Topological vulnerability of the european power grid under errors and attacks, Int. J. Bifurcation Chaos 17 (7) (2007) 2465–2475, doi:10.1142/S0218127407018531.
[26] R. Kolisch, A. Sprecher, {PSPLIB} - a project scheduling problem library: {OR} software - {ORSEP} operations research software exchange program, Eur. J. Oper. Res. 96 (1) (1997) 205–216, doi:10.1016/S0377-2217(96)00170-1.
[27] H. Wang, A survey of maintenance policies of deteriorating systems, Eur. J. Oper. Res. 139 (3) (2002) 469–489, doi:10.1016/S0377-2217(01)00197-7.
[28] R. Atkinson, Project management: cost, time and quality, two best guesses and a phenomenon, its time to accept other success criteria, Int. J. Project Manage. 17 (6) (1999) 337–342, doi:10.1016/S0263-7863(98)00069-6.
[29] R. Cohen, K. Erez, D. Ben-Avraham, S. Havlin, Breakdown of the internet under intentional attack, Phys. Rev. Lett. 86 (16) (2001) 3682.
[30] J.-P. Onnela, J. Saramäki, J. Hyvönen, G. Szabó, D. Lazer, K. Kaski, J. Kertész, A.-L. Barabási, Structure and tie strengths in mobile communication networks, Proc. Natl. Acad. Sci. 104 (18) (2007) 7332–7336.