

Stochastic Pruning and Its Application for Fast Estimation of the Expected Total Output of Complex Systems

Michael Todinov

Department of Mechanical Engineering and Mathematical Sciences
Oxford Brookes University
Oxford, UK
Email: mtodinov@brookes.ac.uk

Abstract

A powerful method referred to as *stochastic pruning* is introduced for analysing the performance of common complex systems whose component failures follow a homogeneous Poisson process. The method has been applied to create a very fast solver for estimating the production availability of large repairable flow networks with complex topology. It is shown that the key performance measures *production availability* and *system reliability* are all properties of a stochastically pruned network with corresponding pruning probabilities. The high-speed solver is based on an important result regarding the average total output of a repairable system including components characterised by constant failure/hazard rates. The average output over a specified operation time interval is given by the ratio of the expected momentary output of the stochastically pruned system, where the separate components are pruned with probabilities equal to their unavailabilities, and the maximum momentary output in the absence of component failures. The running time of the algorithm for determining the expected total output of the system over a specified time interval is independent of the length of the operational interval and the failure frequencies of the edges. The high-speed solver has been embedded in a software tool, with graphics user interface by which a flow network topology is drawn on screen and the parameters characterising the edges and the nodes are easily specified. The software tool has been used to analyse a gas production network and to study the impact of the network topology on the network performance. It is shown that two networks built with identical type and number of components may have very different performance levels, because of slight differences in their topology.

Keywords:

stochastic pruning, stochastic flow networks, production availability, repairable flow networks, performance, software tool, simulation

1 Introduction

Stochastic systems where the component failures are random events are very common and have been discussed extensively in the system reliability literature related to reliability networks ([1,11,5]). Stochastic systems where the flow capacities of components are random variables have also been considered in the literature dealing with stochastic flow networks ([7,2,10,9,16]). The problem of interest was the probability that, on demand, the throughput flow will be equal or greater than a specified level. Evaluating the reliability of complex systems and maximising the

flow in stochastic flow networks has been traditionally based on methods involving minimal cut sets or minimal paths. A similar approach has been adopted by Jane et al. (1993), where stochastic flow networks with multistate components have been considered. Minimum cut sets have also been used by Fishman (1987) to evaluate the distribution of the maximum flow in a directed network whose edges have random capacities. Although, for small-size reliability networks and flow networks, an approach based on minimal paths or minimal cut sets is acceptable, with increasing the size of the network, the number of minimal paths and cut sets increases exponentially and this approach is no longer feasible. This point has been illustrated with the example in Figure 1 discussed in [14]. The flow network in the figure has $N^N + N$ minimal cut sets and N^{N+1} minimal paths. Even for the moderate $N = 10$, the storage and manipulation of the minimal paths and cut sets is impossible. As a result, an algorithm based on determining all minimal paths or cut sets is very inefficient because it will run *in exponential time*.

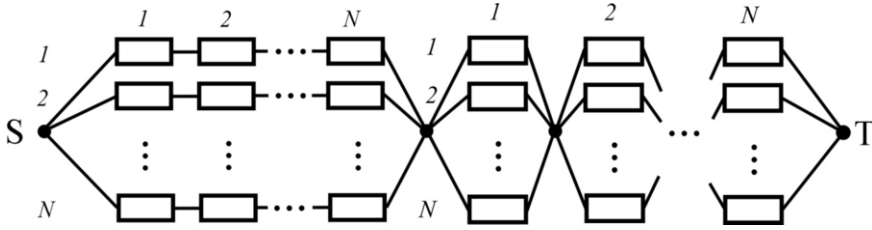


Fig. 1. An example of a flow network where the number of minimal paths and minimal cut sets increases exponentially with increasing the size of the network [14].

A key performance measure of many complex systems over a specified time interval is the *production availability*. This is the expected fraction of total system output during a specified time interval, in the presence of component failures. It is defined as the ratio

$$\psi = \frac{\overline{Q}_T}{Q_{T0}} \quad (1)$$

of the total expected output \overline{Q}_T of the system in the presence of component failures during a specified time interval $[0, a]$ and the total output Q_{T0} of the system that could be obtained in the absence of any component failures. For repairable flow networks (e.g. gas production networks), \overline{Q}_T in equation (1) is the expected maximum throughput flow in the presence of failures of components building the network and Q_{T0} is the total throughput flow that could be obtained during this time interval, in the absence of component failures (Fig.2). For a system of electrical generators connected to an electrical grid, \overline{Q}_T in equation (1) is the expected electrical energy produced by the generators over a specified time interval in the presence of failed generators and Q_{T0} is the total amount of electrical energy that could be produced during this time interval, in the absence of any failed generators. To reveal the variation of the total output, a large number of failure-repair histories during the period of operation of the system must be simulated (Fig.2a).

Upon a component failure, the output of the system usually decreases (Fig.2b). Such are for example the very common systems consisting of interconnected sources

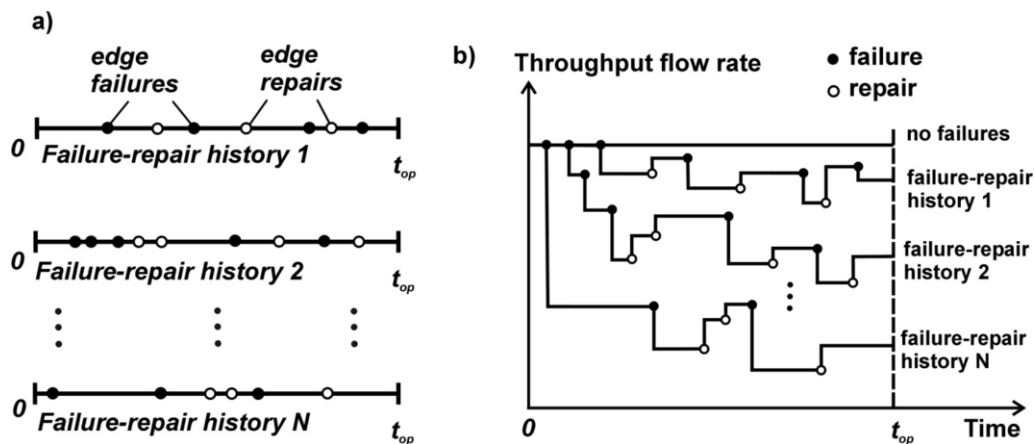


Fig. 2. The variation of the output of a gas production system, as a result of the specific failure-repair history of the system.

providing a particular commodity: electrical generators providing electricity, pump stations distributing water or gas, production wells delivering oil and gas, servers supplying data, etc.

With increasing the number of components in such systems, the likelihood of component failures, whose repair times overlap, increases significantly. This has been illustrated in Fig.2b, which depicts a variation of the output caused by multiple overlapping failures and repairs. If a component failure reduces the output, it causes a dent or a decrease in the output (Fig.2b). On the contrary, a return from repair of a failed component usually causes a rise in the output.

Tracking the variations of the total output caused by component failures whose repair times overlap, is necessary for determining the total output of the system. Because overlapping failures and repairs can be nested in a very complex fashion, tracking the changes of the output is not an easy task and is usually handled by a discrete-event simulator where the failure and repair events are placed in a linked list or a priority queue.

The expected operating periods of real systems (e.g. oil and gas production networks) can be many years, the number of sources of flow (the oil production wells) may reach hundreds, and the components may number thousands. In addition, to optimise production performance, a large number of alternative design layouts must be analysed in a short period of time.

In the area of telecommunication and computer networks, the large size of the network is compounded by a complex network topology.

The running time of a discrete-event simulator developed for tracking the output of a complex system depends strongly on the length of the operational interval and the failure frequencies of the components. If a large running time is combined with large failure frequencies of the components, a single failure-repair history of the simulation may contain thousands of failure/repair events. To reveal the total output with a satisfactory precision, many thousands failure-repair histories are usually required. As a result, the solver must handle tens of millions of failure-repair events

in order to reveal correctly the total output of a large and complex system. In addition, upon each component failure, a problem related to determining the maximum output flow must be solved, whose average running time is at least proportional to the square of the number of components in the system [13]. Consequently, the relatively low efficiency of the conventional discrete-event solvers for large systems does not normally allow embedding them in simulation loops performing topology optimisation.

As a result, a critically important requirement for the discrete-event solvers determining the variation of the total output for large and complex systems is a very high computational speed. This challenge defines the topic of the present work and the presented stochastic pruning method, the ultra-fast algorithm and the software tool are its main contributions.

2 Stochastic pruning and its application for determining the output of a complex system over a time interval

To calculate the production availability ψ in equation (1), a very fast algorithm is proposed, whose computational speed *does not depend on the failure frequencies of the components or the lengths of their repair times*.

Suppose that the system is composed of m independently working components, $k = 1, 2, \dots, m$, each characterised by an unavailability $p_k(t)$, which is, in general, a function of the time t . For components characterised by constant hazard (failure) rates, the probability that a particular component will be in a failed state is given by

$$p_k = \frac{MTTR_k}{MTTF_k + MTTR_k} \quad (2)$$

where $MTTR_k$ is the mean time to repair of the k th component and $MTTF_k$ is the mean time to failure of the k th component [15].

Definition. A stochastically pruned system is the original system from which individual components are removed/deleted with specified probabilities.

The momentary output of the system stochastically pruned with probabilities $p_k(t)$ at time t , is denoted by $Q^P(t)$. The expected value of the momentary outputs $Q_i^P(t)$ $i = 1, 2, \dots, n$ of n stochastically pruned systems (with probabilities $p_k(t)$) is denoted by $\overline{Q}^P(t)$. This quantity is defined by

$$\overline{Q}^P(t) = \lim_{n \rightarrow \infty} \left(\frac{1}{n} \sum_{i=1}^n Q_i^P(t) \right) \quad (3)$$

where n is the number of stochastically pruned systems.

A stochastic pruning of the k th component from the system is done by eliminating its connection from the rest of the system, with a probability $p_k(t)$ equal to the unavailability of the component.

Suppose that the components are characterised by constant failure rates and Q_0 is the momentary output of the system, in the absence of component failures. Then, the following general result holds.

Theorem 2.1 *The production availability of a complex system is given by the ratio of the expected momentary output of the stochastically pruned system (with probabilities equal to the unavailabilities $p_k(t)$ of the components) and the momentary output of the system which has not been pruned:*

$$\psi = \frac{\overline{Q}^P}{Q_0} \quad (4)$$

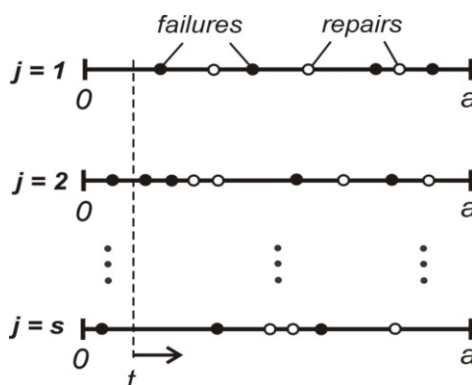


Fig. 3. Stochastic pruning eliminates the need for generating failure-repair histories.

Proof. For a large operational cycle with length a , the total output of a complex system during the operational time interval $[0, a]$, for the j th failure-repair history ($j = 1, 2, \dots, s$) is given by

$$Q_{j,T} = \int_0^a Q_j^P(t) dt \quad (5)$$

This is the integral of the momentary output $Q_j^P(t)dt$ of the system at time t , in the j th failure-repair history (simulation trial). The index "P" stands for "pruned". For the time interval $[0, a]$, the expected value of the total output over s failure-repair histories is given by:

$$\overline{Q}_T = \frac{1}{s} \sum_{j=1}^s \int_0^a Q_j^P(t) dt \quad (6)$$

Swapping the summation sign and the integral in (6) gives:

$$\overline{Q}_T = \int_0^a \left(\frac{1}{s} \sum_{j=1}^s Q_j^P(t) \right) dt \quad (7)$$

The value $\frac{1}{s} \sum_{j=1}^s Q_j^P(t)$ is essentially the expected value of the momentary output at time t (Fig.3), for s stochastically pruned systems, whose components have been

deleted with probabilities equal to their unavailabilities $p_k(t)$, at time t :

$$\overline{Q}^P(t) = \frac{1}{s} \sum_{j=1}^s Q_j^P(t) \quad (8)$$

As a result, the expected value \overline{Q}_T of the total system output during the interval $(0,a)$ becomes:

$$\overline{Q}_T = \int_0^a \overline{Q}^P(t) dt \quad (9)$$

The integral (9), however, can be determined with any specified precision by a Monte Carlo integration. This involves a sufficiently large number ($r = 1, 2, \dots, N$) of random times r sampling the time interval $(0,a)$, at each of which the expected momentary output $\overline{Q}_r^P(t)$ is determined. According to the theory of Monte Carlo integration (Glasserman 2003),

$$\int_0^a \overline{Q}^P(t) dt = \lim_{N \rightarrow \infty} \left(\frac{a}{N} \sum_{r=1}^N \overline{Q}_r^P \right) = a \lim_{N \rightarrow \infty} \left(\frac{1}{N} \sum_{r=1}^N \overline{Q}_r^P \right) \quad (10)$$

The quantity $\frac{1}{N} \sum_{r=1}^N \overline{Q}_r^P$ is the expected momentary output from the stochastically pruned system, over the entire time interval $[0,a]$: $\overline{Q}^P = \frac{1}{N} \sum_{r=1}^N \overline{Q}_r^P$. The total output in the absence of failures is $Q_{0T} = aQ_0$ (Q_0 is the momentary output in the absence of failures) and the production availability becomes:

$$\psi = \frac{a(1/n) \sum_{r=1}^n Q_r^P}{aQ_0} = \frac{(1/n) \sum_{r=1}^n Q_r^P}{Q_0} \quad (11)$$

which proves the theorem. \square

The result given by equation (11) creates the basis of extremely fast solvers for the total output of complex repairable systems, orders of magnitudes faster than discrete-event simulators using linked lists or priority queues. This approach works for both constant failure rates and time-dependent failure rates of the components. *Because no failure-repair histories are generated, the running time of the proposed algorithm is independent of the failure frequencies of the components.* There is no need to build discrete-event simulators for determining production availability. The efficiency of this algorithm creates the possibility of embedding it in a simulation loop performing topology optimisation of large and complex systems. If the component hazard (failure) rates are constant, the failure times follow a negative exponential distribution and the probability that a component will be in a working or failed state does not depend on the actual time along the time interval $(0,a)$. This is because of the memoryless property of the negative exponential distribution. Constant failure rates and the negative exponential distribution of the times to failure are always in place if the conditional probability of failure, given that the component has survived time t , does not practically depend on the age t of the component (Todinov 2016). In this case, failure is caused by random overloads,

not by a wearout. The negative exponential distribution is very common. It describes the flat region of the bathtub reliability curve, typical for many components and systems. From equation (11), for constant failure rates of the components, the production availability becomes:

$$\psi = \frac{(1/n) \sum_{r=1}^n Q_r^P}{Q_0} \quad (12)$$

where the quantity $\bar{Q}^P = (1/n) \sum_{r=1}^n Q_r^P$ does not depend on the sampling time. It is the expected momentary output from a number (n) of stochastically pruned systems.

3 A fast algorithm based on stochastic pruning for determining the production availability of repairable flow networks

The production availability of a complex flow network, including independently working edges characterised by constant hazard rates, can be determined very efficiently by the next very fast algorithm, based on stochastic pruning.

Algorithm 1

```
function MaxFlow();
function RealRandom();

function ProductionAvailability();
{
  SumMomentFlows=0;

  for i=1 to NumPrunedNetworks do
  {
    Stochastic pruning of the network
    for k=1 to m do (m is the number of edges in the network)
    {
      tmp = RealRandom();
      if (tmp < p[k]) then reduce the flow capacity of edge k to zero;
    }
    maxf = MaxFlow();
    SumMomentFlows = SumMomentFlows + maxf;
    Restore the flow capacities of all failed components to their original values;
  }
  return SumMomentFlows / NumPrunedNetworks;
}
```

The array $p[]$ contains the probabilities that the separate edges will be in a failed

state upon demand. In a nested loop controlled by the variable 'k', the separate edges are pruned stochastically ('m' denotes the number of edges/components in the network). The state of the k th edge is tested by generating a uniformly distributed random number between 0 and 1, by the statement 'tmp = RealRandom()', and comparing it with the probability $p[k]$ that the k th edge will be in a failed state. The edges are assumed to be characterised by constant failure rates (negative exponential time to failure distribution). Consequently, the probability that a demand at a specified point in time will sample a failed state for the k th edge can be approximated by $p_k = \frac{MTTR_k}{MTTF_k + MTTR_k}$, where p_k is the average unavailability of the k th edge. All probabilities ($k = 1, 2, \dots, m$) are pre-calculated and stored in the array $p[]$. If the generated random number 'tmp', uniformly distributed between 0 and 1, is smaller than p_k , the k th edge is in a failed state and is pruned (deleted) from the network. If the converse is true, the edge remains in the network.

After the state of all edges has been determined, the function 'MaxFlow()' calculates the maximum throughput flow in the pruned network. The maximum throughput flow is calculated by using the two-stage augmentation algorithm whose details are given in (Todinov 2011).

The expected value of the momentary maximum throughput flow is obtained by dividing the accumulated in the variable SumMomentFlows to the number of stochastically pruned systems. At the end of each simulation trial, the flow capacities of all failed components are restored to their original levels. The proposed algorithm is very simple, very efficient and avoids building a discrete-event simulator for revealing the production availability.

4 Stochastic pruning of reliability networks to determine the reliability of complex systems

Stochastic pruning can also be applied with success to determine the reliability of complex systems. In case of a reliability network of a complex system, the probability that the system will still be operational at the end of the time interval $[0, a]$ (the system reliability) is equal to the probability that paths through working edges from the start node to all end nodes will exist in the reliability network at time $t = a$, which marks the end of the time interval of operation [14]. The separate edges of the reliability network fail with probabilities $F_k(a)$, where $F_k(t)$ are the time to failure distributions characterising the edges of the network ($k = 1, \dots, m$). At the end of the time interval $[0, a]$, the reliability network is effectively stochastically pruned with probabilities $F_k(a)$. The probability of existence of paths to all terminal nodes in the stochastically pruned reliability network is equal to the fraction of stochastically pruned reliability networks for which paths through working edges exist from the start node to all terminal nodes, at the end of the operational time interval $[0, a]$. The reliability of the system is therefore given by $R_{sys} = \lim_{n \rightarrow \infty} \frac{n_A}{n}$, where n_A is the number of stochastically pruned reliability networks for which paths from the start node to all terminal nodes exist and n is the total number of stochastically pruned reliability networks.

Here is the algorithm for determining system reliability by stochastic pruning.

Algorithm 2

```

function PathsToAllTerminalNodes();
function RealRandom();

function SystemReliability()
SuccessCount=0;
for i=1 to NumPrunedNetworks do
{
    Stochastic pruning of the reliability network
    for k=1 to m do (m is the number of edges in the network)
    {
        tmp = RealRandom();
        if (tmp < p[k]) then mark edge 'k' as 'failed';
    }
    PathsExist=PathsToAllTerminalNodes();
    if(PathsExist=1) then SuccessCount=SuccessCount+1;
    Restore the failed edges in the reliability network;
}
return SuccessCount/NumPrunedNetworks;
}

```

The array $p[]$ contains the probabilities of failure of the separate edges, which are given by $p[k] = F_k(a)$. The number of edges is m . In a nested loop controlled by the variable 'k', the state of the separate edges (working/failed) is determined. The state of the k th edge is tested by generating a uniformly distributed random number between 0 and 1 from the statement 'tmp = RealRandom()' and comparing it with the probability $p[k]$ that the k th edge will be in a failed state. A failed edge no longer provides a connection between its corresponding nodes and is essentially excluded from the reliability network. After determining the state of all edges, the function PathsToAllTerminalNodes() establishes whether there exist connections through working edges from the start node to each terminal node. The ratio of the number of pruned reliability networks for which a connection from the start node to each of the terminal nodes exists and the total number of pruned reliability networks is an estimate of the reliability of the system. At the end of each simulation trial, all edges marked as 'failed' are restored as working edges. A failed edge that has been restored resumes the connection between the corresponding nodes of the edge and is essentially included in the reliability network. In the case of very small probabilities of failure characterising the edges, the precision of the presented Monte Carlo crude sampling can be increased by applying stratified sampling without replacement.

5 Analysis of the performance of repairable flow networks

The high-speed algorithm for revealing the production availability of repairable flow networks has been embedded in a software tool, with graphics user interface (Fig.4) by which the network topology is drawn on screen and the capacities, the failure frequencies and the repair times of the edges and nodes are specified. The software tool includes a menu from which an embedded function performing a desired service is called. The parameters characterising each edge and node can be specified individually, by double-clicking them (Fig.4). For each edge or a node, a large number of failure modes can be specified and for each failure mode, the failure frequency and the time to repair can be specified.

A copy/paste function has been provided, for quickly transferring the parameters from one edge/node to another edge/node. Functions for zooming, panning, auto-arranging, saving and loading of flow networks have also been provided. Attributes of edges and nodes can be viewed and listed on the screen. Nodes and edges can be easily deleted and added to the network, in order to achieve quickly the desired topology.

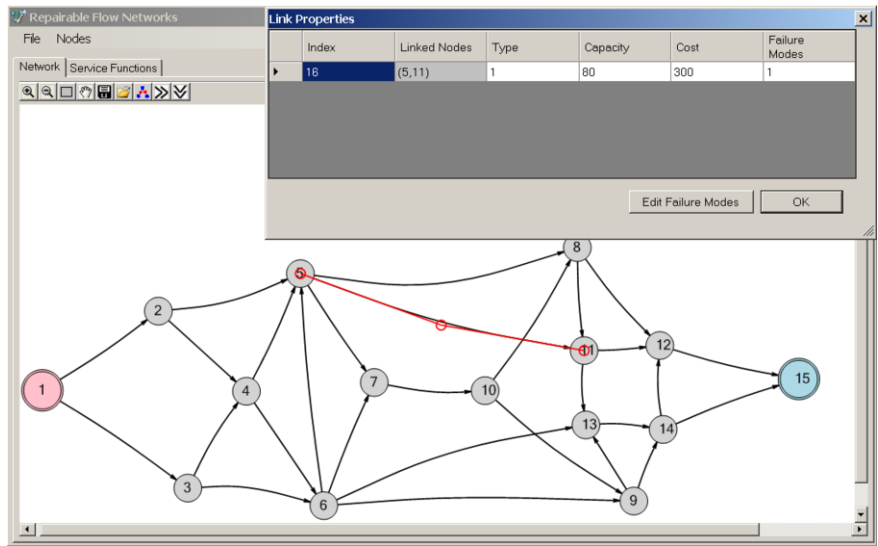


Fig. 4. Specifying the parameters of an edge by using the developed software tool

The developed software tool has been applied for calculating the variation of the total throughput flow in a section of a repairable gas production network (Fig.5). The considered gas production section consists of three sources (initial injection stations) s1, s2 and s3. Each injection station (source) has a production capacity of $70 \times 10^3 \text{ m}_3/\text{day}$. From the initial injection stations, through a system of pipelines, compressors and valves, the gas is delivered to the sink t . Each edge models a pipeline section, with unreliable compressor transporting the gas through it. Each pipeline section is therefore associated with a particular failure rate, as well as flow capacity. It is also assumed that failure of a section (edge) causes the flow through

the corresponding edge to stop.

Constant hazard/failure rates have been assumed for the pipeline sections (edges) and the initial injection stations. On each edge, the hazard rates (in years^{-1}) and the capacities of the edges (in thousands of cubic meters per day), have been given as two consecutive numbers. For example, 4/60 stands for a hazard (failure) rate 4 year^{-1} and flow capacity of 60000 m^3 a day. The initial injection stations have hazard rates 2 year^{-1} . The downtime for repair of each component has been assumed to be the same: $d = 11$ days. If no failures occur, the maximum throughput flow in the network obtained by the algorithm is $150 \times 103 \text{ m}^3/\text{day}$.

The production network in Fig.5a also contains the cycle (4,6,5,4). In Fig.5b, the production network with the three sources (s1,s2 and s3) from Fig.4a has been transformed into a single-source network. The edges (0,1), (0,2) and (0,3), connecting the sources s1, s2 and s3 to the super-source s, are characterised by the failure rates and the flow capacities of the sources s1, s2 and s3 they replace.

For the network in Fig.5 $\psi = \frac{\bar{Q}^P}{Q_0} = 0.85$ has been determined by the proposed method of stochastic pruning. One hundred thousand stochastically pruned systems on a laptop Dell Precision M4800 with processor Intel(R) Core(TM) i7-4810MQ@2.8 GHz, took only 0.27 seconds. The system in Fig.5 was chosen to provide a comparison with the result (0.85) for the production availability produced by a discrete-event simulator based on linked lists, discussed in [12] for the same system. The results from the two different algorithms coincide which confirms the validity of the proposed method of stochastic pruning for estimating production availability.

The running time of the stochastic pruning algorithm however, does not depend on the failure frequencies of the components and *with increasing the failure frequencies of the components the execution time does not increase*. The running time of the discrete-event simulator based on linked lists is significantly larger than the running time of the proposed algorithm. In fact, by increasing the failure frequency of the components, the running time of the discrete-event simulator can be made many orders of magnitude larger than the running time of the proposed algorithm. This is because, with increasing the failure rates of the components, the number of events in the linked list increases proportionally. As a result, in each of the hundreds of thousands generated failure-repair histories, there is a very large number of events, which entails a proportionally large execution time. Consequently, the running time of the proposed algorithm can be orders of magnitudes smaller than the running time of the conventional discrete-event simulator based on linked lists or priority queues. This was confirmed during the simulations.

6 Comparing the performance of competing network topologies

A very important issue related to repairable flow networks is the link between network topology and network performance. An important question here is what features in the network topology and structure (e.g. connectivity, reliability of compo-

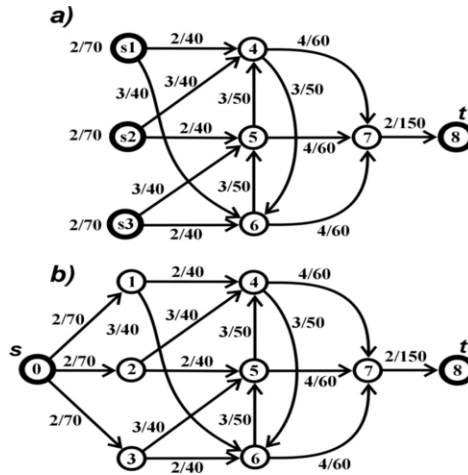


Fig. 5. a) A repairable gas production network based on three initial injection stations s1,s2 and s3; b) Transforming the production network into a single-source network.

nents, spare capacity, etc.) make the throughput flow least sensitive to component failures. An important application of the fast production availability algorithm and the developed software tool is in comparing quickly the performance of competing network topologies and selecting the topology with the best performance.

Consider for example the two competing production network designs in Fig.6a and Fig.6b, where the capacities of the edges are according to the figure and all edges have the same failure rate of 4 expected failures per year and downtime for repair 10 days. This has been done to isolate and analyse only the impact of the network topology. Without loss of generality, it is assumed that only edges can fail while the nodes are perfectly reliable. If unreliable nodes exist, any unreliable node can be split into two perfectly reliable nodes connected by an unreliable edge which has the failure frequency of the unreliable node it replaces.

By using the developed software tool for determining production availability, it is easy to establish which competing topology is superior and by how much. Applying the proposed algorithm yields production availability (expected fraction of the transmitted throughput flow in the presence of failures) of $\psi_a = 79.2\%$ for the network in Fig.6a and only $\psi_b = 73\%$ for the network in Fig.6b. Despite the seemingly insignificant differences between the competing topologies, the impact on the production availability is very big. A difference of 6.2% in production availability translates over the years in huge quantity of lost production and can make the difference between a highly profitable and unprofitable production system.

One of the reasons for the superior performance of network topology 'a' is that redundant edges (2,7) and (3,7) from topology 'a' bypass more unreliable edges compared to the corresponding redundant edges (4,7) and (5,7) from topology 'b'. Indeed, failure of the unreliable edge (2,4) in topology 'a' does not cause a loss of flow because the flow through the failed edge (2,4) can be redirected through the edge (2,7). Failure of edge (2,4) in topology 'b' however, causes loss of flow through edge (2,4) because there is no possibility of redirecting the flow along alternative paths. Consequently, in improving the availability of production networks, redundancies

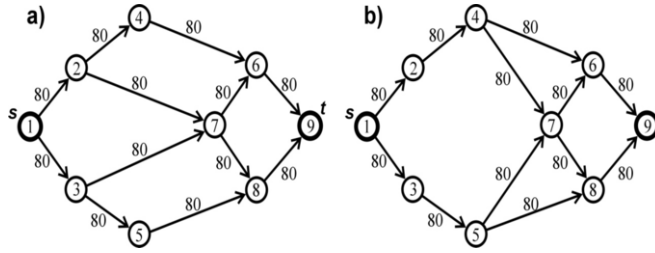


Fig. 6. Two competing networks with different types of redundancy.

need to be placed in such a way that they bypass as many unreliable edges as possible.

Next, consider, the competing production networks in Fig.7a and Fig.7b, where again, to isolate the influence of the topology, all edges have the same flow rate capacity of 40 flow units per day, hazard rate of 4 expected failures per year and downtime for repair 10 days. Edges (3,8) and (4,9) from network 'a' and edges (2,8) and (4,10) from network 'b' are redundant. Without the support of the software tool, it is difficult to infer which network topology is superior. Applying the software tool yields production availability of $\psi_a = 70.5\%$ for the network in Fig.7a and $\psi_b = 75.2\%$ for the network in Fig.7b. As can be verified, despite the seemingly insignificant differences in the competing topologies, the impact on the production availability is very big.

Consider now the network in Fig.8 where again, all edges have the same flow rate capacity of 40 flow units per day, hazard rate of 4 expected failures per year and downtime for repair 10 days. If the redundancies are placed such as it has been shown in Fig.8, the availability of the network is even higher: 80.5%.

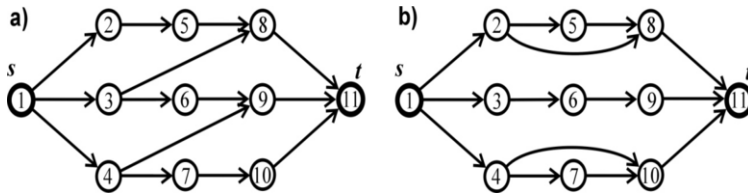


Fig. 7. Two competing networks with different types of redundancy topology.

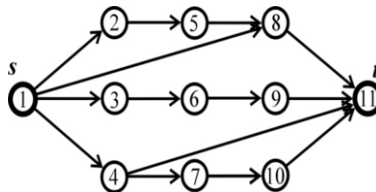


Fig. 8. A network with the same number of redundant links and superior topology.

7 Conclusions

Stochastic pruning is a powerful alternative to the traditional approach for analysing the performance of complex systems. It avoids building discrete-event

simulator for analysing the production availability of complex systems.

Stochastic pruning yields ultra-fast algorithms whose computational speed does not depend on the length of the operating interval and the failure frequencies of the components.

An important result related to the production availability of complex systems has been established: The production availability is equal to the ratio of the expected momentary output from a number of stochastically pruned systems and the output from the system which has not been pruned.

The key performance measures production availability and system reliability are average properties related to stochastically pruned systems.

Reliability of a system can be redefined as: the fraction of stochastically pruned reliability networks in which paths through working components from the start node to all terminal nodes exist.

The high-speed algorithm for stochastic pruning has been embedded in a software tool with a graphics user interface.

The topology of repairable flow networks has a significant impact on their performance. Two networks built with identical number and type of components may have very different production availabilities because of slight differences in their topology.

References

- [1] Billinton R. and Allan R.N., "Reliability evaluation of engineering systems", 2nd ed., Plenum press, 1992.
- [2] Evans JR, *Maximal flow in probabilistic graphs the discrete case*, Networks. **6** (1976), 161-183.
- [3] Fishman G.S., *The distribution of maximum flow with applications to multistate reliability systems*, Oper Res. **35**(4), (1987), 607-618.
- [4] Glasserman P., "Monte Carlo methods in financial engineering", Springer, NewYork, 2003.
- [5] Hoyland A., Rausand M., "System reliability theory", John Wiley and Sons, 1994.
- [6] Jane C, Lin J., Yuan J., *Reliability evaluation of a limited-flow network in terms of minimal cut sets*, IEEE Trans. Reliab. **42**(3), (1993) 354-368.
- [7] Lee SH, *Reliability evaluation of a flow network*, IEEE Trans. Reliab., R-29(1), (1980), 24-26.
- [8] Lin Y.K., *A simple algorithm of reliability evaluation of a stochastic flow network with node failure*, Comput. Oper. Res. **28**(13), (2001) 1277-1285.
- [9] Lin Y.K., Yuan J, *A new algorithm to generated-minimal paths in a multi-state flow network with non-integer arc capacities*, Int.J. Reliab. Qual. Saf. Eng. **5**(3), (1998), 269-285.
- [10] Lin J.C., Jane C., Yuan J, *On reliability evaluation of a capacitated flow network in terms of minimal pathsets*, Networks. **25**, (1995), 131-138.

- [11] Ramakumar R., "Engineering reliability, fundamentals and applications", Prentice Hall, 1993.
- [12] Todinov M.T., *A discrete-event simulator for repairable flow networks with complex topology*, In Proceedings of the Second International Conference on Computational Intelligence, Communication Systems and Networks, Liverpool, 2010.
- [13] Todinov M.T., *Analysis and optimization of repairable flow networks with complex topology*, IEEE Trans.Reliab. **60**(1), (2011), 111-124.
- [14] Todinov M.T., "Reliability and risk models: setting reliability requirements", 2nd ed., Wiley, 2016.
- [15] Trivedi K.S., "Probability and statistics with reliability, queuing and computer science applications", John Wiley and Sons, NewYork, 2002.
- [16] Yeh W.C., *A simple approach to search for all d-MCs of a limited-flow network*, Reliab. Eng. Syst. Saf. **71**(1), (2001), 15-19.