

# A high-speed algorithm for repairable stochastic flow networks with converging flows from multiple sources

M. T. Todinov

*Department of Mechanical Engineering and Mathematical Sciences,  
Oxford Brookes University, UK*

## Abstract

An algorithm and software implementation have been developed for a very fast discrete-event simulator for determining the losses from failures in repairable stochastic flow networks with converging flows from multiple sources. This paper shows that the computational speed related to determining the variation of the flow through a stochastic flow network can be improved enormously if the topology of the network is exploited directly. The proposed method is based on new results related to maximising the flow in networks with converging flows. It handles repairable networks with multiple sources of production flow, multi-commodity flows, overlapping failures, multiple failure modes, redundant components and redundant branches of components. The simulator is capable of tracking the cumulative distribution of the potential losses from failures associated with the whole network and with each component in the network. Finally, by using an exemplary stochastic flow network with converging flows, it is demonstrated how the developed fast discrete-event simulator can be used for revealing the distribution of the operational losses and identifying the components with the largest contributions to the total losses from failures.

*Keywords: stochastic flow networks, reliability, availability, maximal flow, tree topology, potential losses from failures.*

## 1 Introduction

At an abstract level, a repairable network with flows from multiple sources can be presented as a number of sources connected to terminals (sinks) through components characterised by flow capacities and time-to-failure distributions. In



order to reveal the variation of the total output flow that the network is capable of transferring to the terminals, a large number of failure histories during the lifecycle of the network must be simulated. This variation, which can be obtained by designing a discrete-event simulator, is key to determining the production availability – a basic performance measure of a repairable network with flows. It is the ratio of the expected amount of flow delivered during a specified time interval and the maximum possible amount of flow that can be delivered during this time interval in the absence of failures. Another important performance measure is the distribution of the potential losses from failures. This distribution is a key to determining the probability that the losses from failures will exceed a maximum acceptable level.

During the design of repairable networks with flows, the speed of analysis is essential in order to keep pace with fast-track projects. In order to optimise production performance, a large number of alternative design layouts must be analysed in a short period of time. The designer must be able to quickly filter out inappropriate designs associated with large losses from failures and select the best solution.

Most of the research on stochastic flow networks is focused on calculating the maximum flow transmitted from the sources to the sink (Ahuja et al. [1]; Cormen et al. [2]). The best of the existing algorithms however are characterised by a polynomial complexity. As a result, these algorithms are suitable for systems with complex topology only if they are run once or a small number of times. Unfortunately, these algorithms are not suitable for large and complex repairable networks incorporating thousands of components. The simulators based on these algorithms will run very slowly, because revealing the variation of the output flow over a large time interval requires the generation of tens of thousands of failure histories, each of which may contain hundreds of failures. As a result, the maximum-flow algorithm will have to be executed not just once or several times, but hundreds of thousands of times. There also exist methods (Lin [3], Yeh [4]) that directly exploit the minimum-cut maximum flow theorem (Elias et al. [5]; Ford and Fulkerson [6]). Again, this approach works satisfactorily for networks comprising a relatively small number of components. With the size of the network increased, the number of minimal cuts increases to such an extent that even their storage and management becomes a problem (Todinov [7]). Here, we show that extremely efficient algorithms, orders of magnitudes faster than any of the existing general-purpose algorithms for maximising the flow in a repairable network, can be designed by exploiting the network topology. Since many repairable networks (e.g. in oil and gas production) are characterised by converging flows from multiple sources, the described algorithm will be based on this common topology.

## 2 A fast algorithm for determining the maximum flow in a repairable stochastic flow network with converging flows

Consider the network with converging flows in Fig. 1. Suppose that two cuts in the network are made such as the cuts  $k_0 - k_0$  and  $k - k$  shown in Fig. 1.



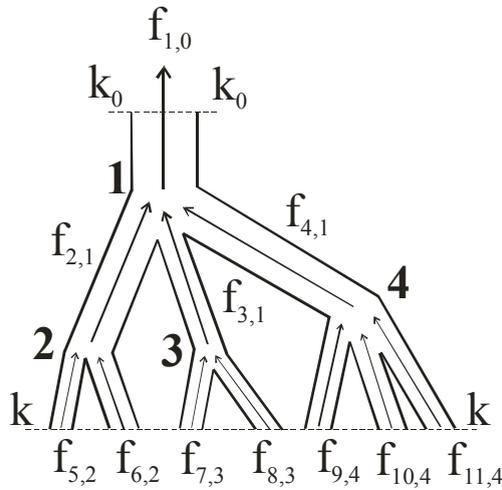


Figure 1: A flow network with converging flows.

One of the cuts is a ‘main cut’ – a single cut through a selected branch of the network (the cut  $k_0 - k_0$  in Fig.1). The other cut is an ‘isolating cut’ – through all flows feeding the flow from the main cut (the cut  $k-k$  in Fig.1). Nodes have been labelled 1,2,3,4,... and flows have been labelled  $f_{i,j}$  where  $i$  and  $j$  stand for the  $i$ -th and the  $j$ -th node. For each node  $i$ , the flow conservation law holds: the sum of the flows entering the node (inflows) is equal to the flow exiting the node (the outflow). The following theorem (whose proof is omitted here because of lack of space) is important to the algorithm described later.

**Theorem 1:** For any pair of a main cut and isolating cut, a flow constrained by a factor  $k$  ( $0 \leq k \leq 1$ ) from the main cut is equal to the sum of the flows constrained by the same factor from the isolating cut:  $f'_{1,0} = kf_{1,0} = \sum_i kf_{i,isolating\ cut}$ . This theorem creates the possibility of

calculating the flow through the main cut upon failure in the branch where the main cut is located without the need to calculate flows between the isolating cut and the main cut. Obtaining the flow through the main cut is done simply by adding the flows reduced by the same factor from the isolating cut feeding the main cut.

The building blocks of the repairable networks with converging flows considered in this paper are sections composed of branches arranged in parallel each of which may contain components arranged in series. The sections are blocks with multiple input flows and a single output flow (Fig. 2), which constitutes a network with converging flows. A characteristic feature of a network with converging flows is that the flows from the sources *do not branch*

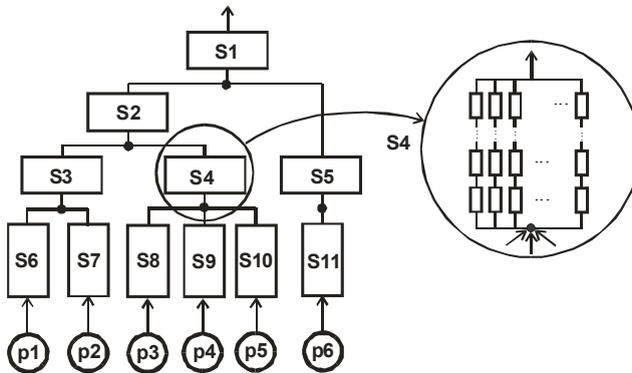


Figure 2: An example of a repairable stochastic flow network with converging flows topology based on six production sources ( $p_1, p_2, \dots, p_6$ ).

*out between the sections.* In other words, if a source feeds a particular section then it is guaranteed that the entire flow from the source passes through the section.

The simplest section is a single component. Each component is characterised by a flow capacity, which decreases to a certain level upon failure of the component. Starting from the top down, a converging flow topology can be constructed if the top section is fed by several other sections, each of which is fed in turn by several other sections, etc. In Figure 2, the sections are numbered  $S_1, S_2, \dots, S_{11}$ . The sections permit nesting complexity within the converging flows topology. They also significantly simplify the representation and analysis of common flow networks because large parts of them are, in effect, long branches with components arranged in series or parallel. The description of the system topology is also simplified because the arrangement of components in a branch belonging to a particular section is not important and may not be specified. Only the number and the type of components in the branch are important and should be specified. This is because the flow in a branch consisting of components in series is constrained from the component with the smallest capacity and does not depend on the location of the components along the branch.

The second theorem is related to determining the maximum flow through a stochastic flow network with converging flows. Its proof is also omitted because of lack of space.

**Theorem 2.** For any set of values of the section capacities, the maximum flow through the network is always obtained after a number of steps equal to the number of sections in the system. Each step consists of considering a new section and if the capacity of the section is smaller than the sum of the flows from the sources feeding the section, the flows from these sources are constrained (choked) proportionally so that the sum of the flows from the sources feeding the section is equal to its flow capacity.

The algorithm starts by calling a procedure performing an initial choking of the production sources. One by one, constraints on the sources of flow are introduced by each section in the network in such a way that the flow through the sections fits the existing capacity of the sections. According to Theorem 2, after the execution of this procedure, a maximum flow through the network will be set.

For a network with converging flows consisting of  $E$  sections (which for the sake of simplicity are assumed to be single components) and  $N_p$  sources, for each section a maximum of  $N_p$  checks are needed. In the case where the  $E$  sections form a balanced binary tree, the complexity of the algorithm to determine the maximum flow in the network is  $O(\log_2(E+1) \times N_p)$  which is equal to the maximum number of updates done on the sources. In the worst case scenario, where each section is fed by all of the sources, the maximum number of updates done on the sources does not exceed  $E \times N_p$ . Clearly, the computational speed of the proposed algorithm for determining the maximum flow in the network is very high.

Next, for all components, the times to failure characterising all failure modes are generated by using the *inverse transformation method*. The time to failure distribution characterising the  $j$ -th failure mode of the  $i$ -th component has been assumed to be the two-parameter Weibull distribution:

$$F(t) = 1 - \exp\left(-\left(t/\eta_{ij}\right)^{\beta_{ij}}\right).$$

For the  $i$ -th component, characterised by  $im$  failure modes, the scale parameters  $\eta_{ij}$ , and shape parameters  $\beta_{ij}$ ,  $j = 1, im$  are specified for each failure mode. The Weibull distribution is very flexible because it permits describing failure modes from all regions of the bath-tub curve. The factors with which the component flow capacity decreases upon failure are also specified for each failure mode. For each component, the failure mode that will occur first is identified and this is the failure mode that will fail the component. Next, for all components, the component with the smallest time to failure is identified and this is the component that will fail first. The flow capacity of the failed component for a particular failure mode is determined by multiplying its capacity in a non-failed state by the 'capacity reduction factor' corresponding to the failure mode.

Failure of a component affects the flow indirectly, through the section to which the component belongs, by decreasing the flow capacity of the section. This determines the next step – determining the flow capacity of the section to which the component belongs after the component failure. Failure of a component in a particular section may or may not restrict the flow through the section. A section consisting of  $N$  parallel branches containing components in series has a flow capacity equal to the sum of the flow capacities of the parallel branches.

A check is then performed whether the deteriorated flow capacity of the section has created a bottleneck (has constrained the flow from the sources). A



bottleneck caused by a failure of a component, constrains the flows from the production sources feeding the section to which the failed component belongs. Whether there will be a bottleneck after a component failure in the section depends on the ratio of the degraded flow capacity of the section and the sum of the flow rates of the sources feeding the section. If this ratio is smaller than one, then a bottleneck is present. The sources feeding the section are then constrained proportionally so that the sum of their flow rates becomes equal to the flow rate of the failed section.

For example, a decrease in the flow capacity of section S2 in Fig. 2 constrains the flows from sources  $p1, p2, p3, p4$  and  $p5$  feeding the section. The flow rate through section 2 is then equal to the sum of the flow rates of sources  $p1-p5$  feeding section 3. If the actual flow through section S2 is needed, it is obtained simply by adding the current flow rates of production sources  $p1-p5$ . The flows through the intermediate sections S3, S4, S6-S10 are not needed. Upon deterioration of the flow capacity of section S2, the flows from the production sources  $p1-p5$  originating flow paths entering the section are decreased proportionally to fit the reduced flow capacity of section S2.

The state of the flow network at any time is fully determined by the current flows from the production sources. *There is no need for a system reliability analysis and analysis of the flows in the branches in the network.* This circumstance speeds up the analysis enormously.

If all sources are producing at their maximal flow rates (there is no initial choking) for a flow network with converging flows there is no possibility of compensating for the loss from failure of a particular section by increasing production from sources along whose flow paths no bottlenecks are present. As a result, if no initial choking of production sources is present, constrained production flows due to failure of components are lost and the availability analysis is reduced to updating the flows from the production sources upon degradation of the section capacities.

All failed components are pushed into a stack. Losses are calculated only when a component failure constrains some of the production sources. This event initiates intervention for repair. One by one, the failed components are removed from the stack and the losses associated with failures of the separate components (downtime, lost production flows, cost of intervention for repair/replacement) are calculated and accumulated. The simulation continues by repairing the removed components to *as good as new* condition, after which they are 'put back in operation'.

The losses calculated upon removing failed components from the stack, serve as a basis for calculating the production availability of the flow network and producing reports related to the losses associated with the flow network and the separate components.

## 2.1 Input data

Information about the sources, sections and components is read from input files. For each production source, the capacity (maximum flow rate) is specified as well as the composition of the flow (e.g. the fractions of waste liquid or gas per



unit of mixed flow). Next, information about the existing sections is read. Each section is described by a record with fields corresponding to the number of sources feeding the section, the number of branches in the section, the flow capacity of the section and the type of the section. In each section record, dynamic arrays are defined holding the capacity of each branch in the section and the indexes of the sources feeding the section. Memory for the dynamic arrays is allocated during run time, after the actual number of sources and branches in each section become available from the input data. Finally, information about the components and their failure modes is read. Each component record has fields specifying to which section and branch in the section the component belongs. In dynamic arrays defined in the records, all parameters  $\eta_{ij}$  and  $\beta_{ij}$  in the Weibull distributions related to the times to occurrence of each failure mode of the components are specified. Dynamic arrays holding the parameters of the time to repair distributions characterising all failure modes are also specified. For each failure mode, a constant repair time, or normally-distributed or log-normally distributed repair time can be specified. For each failure mode, the type of intervention and its cost are also specified. This information is necessary to correctly determine the distribution of the potential losses from failures, characterising the stochastic flow network.

Dynamic memory allocation during run time is a powerful implementation technique because it permits handling sections with the number of components, the number of branches and the number of failure modes, limited only by the available memory.

### 3 A solved example

Figure 3 is an example of a simple stochastic flow network with six sources of multi-commodity flow which is a mixture of useful liquid chemical *A*, waste liquid *B* and useful gas *C*, produced from 6 sources p1-p6. The system consists of 34 components distributed in 18 sections. Eight sections contain a single component, seven other sections contain two components working in parallel and each of three other sections contain four components, distributed in two parallel branches each containing two components in series.

The mixed flow is delivered from sources p1 to p6 to the separator *S* (1), which provides 100% separation of the incoming mixed flow to liquid *A*, gas *C* and waste liquid *B*. Upon failure of a section, its flow capacity is reduced and the production from the corresponding sources feeding the section is cut back proportionally in order to accommodate the flow into the reduced capacity of the section. Failure of the section composed of components 30 and 31 in the waste liquid *B* line is associated with loss of capacity for the waste liquid *B* and initiates a cut back of production from the sources with the largest content of waste liquid *B* so that the produced waste liquid fits in the remaining flow capacity of the section composed of components 30 and 31, and at the same time, the production of useful liquid *A* is maximised. Failure of the section



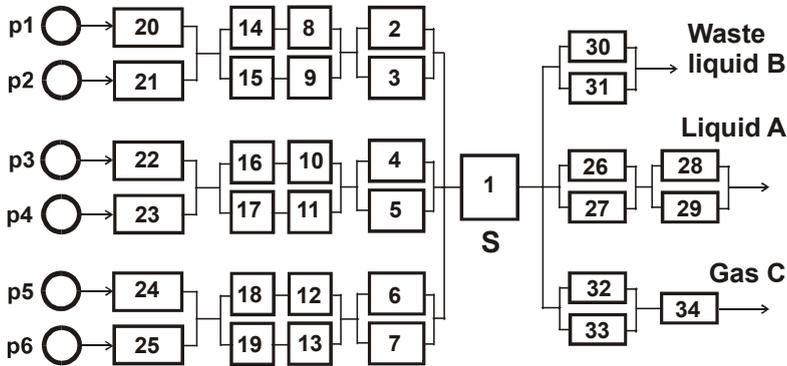


Figure 3: A simple generic stochastic flow network.

Table 1: Production sources, liquid rate, and composition of the flow.

Source number	Liquid rate $\times 10^3$ l/per day)	Waste liquid B content %	Gas to liquid A ratio [ $m^3/10^3$ litres]
p1-p3	20	20	0.42
p6-p10	20	34	0.21

composed of components 32 and 33 or failure of the section composed of component 34 only is associated with loss of gas flow capacity and results in a cut back of production from the sources with the largest gas C-to-liquid A ratio, so that gas production fits in the gas section with the smallest capacity and, at the same time, production of liquid A is maximised. The flow rates for the six sources are shown in table 1.

Each component has the following capacity, reliability data and capacity reduction if failure occurs.

The cost of intervention associated with the separate components is as follows: Separator 1 (\$2000), components 2-7 (\$600), components 8-19 (\$250), components 20-25 (\$400) and components 26-34 (\$300). The cost of 1000 litres liquid A is \$120 and the cost of 1000  $m^3$  gas C is \$40.

The discrete-event simulator was implemented in C/C++. It simulated 1000 histories of a 20 year life-cycle in significantly less than a second! The calculated results and the execution time on a laptop with an *Intel(R) T7200 @ 2.00 GHz* processor were as follows:

- Production availability (Liquid A): 98.4%
- Production availability (Gas C): 98.4%
- Execution time: 0.26 seconds!

Comparative calculations performed on simpler systems confirmed the validity of the results obtained by the simulator.

Table 2: Component capacities, reliabilities, time for repair and capacity reduction factors.

Component	Capacity l/day, m <sup>3</sup> /day	$\beta$	$\eta$	Time for repair (days)	Capacity reduction factor
Separator (1)	120 x 10 <sup>3</sup> l/day	1.5	4.5	12	0.4
2-7	20 x 10 <sup>3</sup> l/day	1	3.5	6	0.3
8-19	20 x 10 <sup>3</sup> l/day	1.2	8.5	5	0.5
20-25	20 x 10 <sup>3</sup> l/day	1.7	5.3	9	0.2
26-29	60 x 10 <sup>3</sup> l/day	1	6.3	7	0.4
30-31	17 x 10 <sup>3</sup> l/day	1	6.3	7	0.4
32-33	19 m <sup>3</sup> /day	1	6.3	7	0.4
34	38 m <sup>3</sup> /day	1	6.3	7	0.4

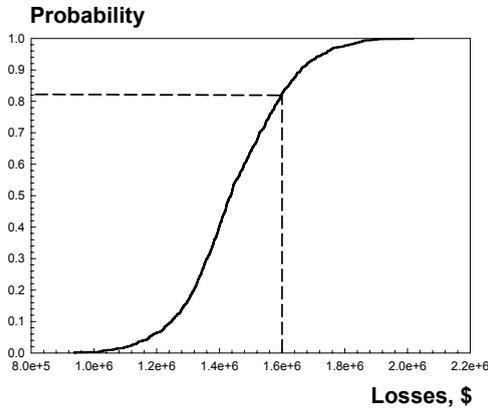


Figure 4: Distribution of the potential losses from failures.

Figure 4 gives the distribution of the potential losses from failures for the stochastic flow network in Fig. 3. The probability that potential losses will exceed a particular budget of \$1.6 million US dollars allocated for covering losses from failures is approximately 18%. The expected value of the potential losses from failures is  $\$1.4 \times 10^6$ .

The analysis of the expected potential losses associated with the separate components revealed that the separator (1) is responsible for approximately 27% of the total system losses. This component needs attention and its reliability should be improved if the overall system losses are to be reduced. Reliability can for example be improved by providing redundancy for this component.

## 4 Conclusions

1. An algorithm and a software implementation of a very fast discrete-event simulator for repairable stochastic networks with converging flows have been developed. It is shown that the state of such networks at any time is fully determined by the values of the flows from the production sources.

2. The maximum flow through a network with converging flows is obtained after a finite number of steps equal to the number of sections in the network. Each step consists of considering a new section and if the capacity of the section is smaller than the sum of the flows from the sources feeding the section, the flows from the sources are constrained so that the sum of the flows feeding the section equals the section capacity.

3. The proposed algorithm handles very large systems, containing hundreds of sources and thousands of components, multi-commodity flows, overlapping failures and the existence of redundant components and branches.

4. The simulator tracks the cumulative distribution of the potential losses from failures and the losses associated with each component in the system.

5. The simulator can be used by engineer-designers as a sound basis for assessing operational losses associated with competing design solutions and identifying components with large contributions to the total potential losses from failures.

## References

- [1] Ahuja R.K., T.L. Magnanti, J.B. Orlin, *Network flows: Theory, Algorithms and Applications*, Prentice Hall, 1993.
- [2] Cormen, Thomas H., Leiserson, Charles E., Rivest, Ronald L. and Stein C. (2001). *Introduction to Algorithms*, second edition, MIT Press and McGraw-Hill, 2001.
- [3] Lin Y.K., Evaluate the performance of a stochastic-flow network with cost attribute in terms of minimal cuts, *Reliability Engineering and Systems Safety*, 91, pp. 539–545, 2006.
- [4] Yeh W.C., A simple algorithm to search for all minimal cutsets with unreliable nodes, *Complexity International*, 8, pp. 1–7, 2001.
- [5] Elias P., A. Feinstein and C.E. Shannon, Note on maximum flow through a network, *IRE Transactions on Information Theory*, IT2, pp. 117–119, 1956.
- [6] Ford L. and D. Fulkerson, Maximal flow through a network, *Canadian Journal of Mathematics*, pp. 399–404, 1956.
- [7] Todinov M.T. *Risk-based reliability analysis and generic principles for risk reduction*, Elsevier, 2007.

