14$^{\text{th}}$ October, 2021

STK3405/4405

Mandatory assignment 2 of 2

Submission deadline

Thursday 28$^{\text{nd}}$ October 2021, 14:30 in Canvas (canvas.uio.no).

Instructions

You can choose between scanning handwritten notes or typing the solution directly on a computer (for instance with LaTeX). The assignment must be submitted as a single PDF file. Scanned pages must be clearly legible. The submission must contain your name, course and assignment number.

It is expected that you give a clear presentation with all necessary explanations. Remember to include all relevant plots and figures. Students who fail the assignment, but have made a genuine effort at solving the exercises, are given a second attempt at revising their answers. All aids, including collaboration, are allowed, but the submission must be written by you and reflect your understanding of the subject. If we doubt that you have understood the content you have handed in, we may request that you give an oral account.

In exercises where you are asked to write a computer program, you need to hand in the code along with the rest of the assignment. It is important that the submitted program contains a trial run, so that it is easy to see the result of the code.

Application for postponed delivery

If you need to apply for a postponement of the submission deadline due to illness or other reasons, you have to contact the Student Administration at the Department of Mathematics (e-mail: studieinfo@math.uio.no) well before the deadline.

All mandatory assignments in this course must be approved in the same semester, before you are allowed to take the final examination.

Complete guidelines about delivery of mandatory assignments:

uio.no/english/studies/admin/compulsory-activities/mn-math-mandatory.html

GOOD LUCK!

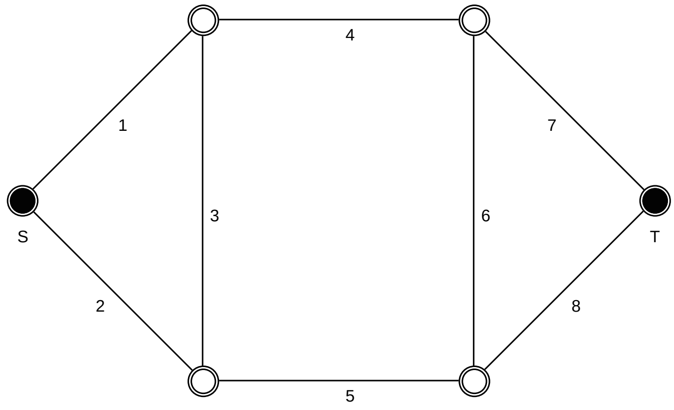

Figure 1: A binary monotone system

Problem 1. In this problem we consider the 2-terminal undirected system (C, ϕ) shown in Figur 1. The nodes of the system are assumed to be functioning *perfectly* with probability 1, and the terminals of the system are the nodes labled S and T. The components in the system are the *edges* in the network, labled $1, \ldots, 8$. Thus, the component set C is the set $\{1, \ldots, 8\}$. We introduce the component state variables $X_1, \ldots, X_8$. We assume that these variables are stochastically independent, and that the component reliabilities are given by:

$$P(X_i = 1) = p_i, \quad i = 1, \ldots, 8.$$

We also introduce the component state vector $\boldsymbol{X} = (X_1, \ldots, X_8)$ and the vector of component reliabilities $\boldsymbol{p} = (p_1, \ldots, p_8)$.

In this assignment we will analyze this system both analytically and by using Monte Carlo simulation. For the simulations we shall use the software program CMCsim™.

IMPORTANT: When solving the different points below, you will create various figures and plots using CMCsim™. All these figures and plots must be *included in the assignment*.

CMCsim™ can be downloaded from:

http://www.riscue.org/cmcsim

NOTE: If you use a Mac, you need to allow the program to be used even though it is initiallly not recognized by the MacOS. A detailed explanation of how to do this is given on the CMCsim™ webpage.

Before CMCsim™ can calculate anything, you must build a *model* of the undirected network system under consideration. This includes building a graphical representation of the system. This includes creating the *nodes* of the network, as well as the edges between the nodes. Below is a brief description of how this is done.

Creating nodes. A new node is created by pressing the *New Node* button in the toolbar. See Figure 2. A new terminal node is created by pressing

Figure 2: The *New Node* button

the *New Terminal* button in the toolbar. See Figure 3. The node objects

Figure 3: The *New Terminal* button

can be displayed in different sizes. In order to make the nodes have the size shown in Figure 1, we use the menu command: Layout>Icon Size>Small. Note that when using a small icon size, the terminal nodes and the other nodes initially look the same. Thus, in order to see which nodes that are terminals, it is recommended to use a specific color for these nodes. In order to choose a color for a given node, right-click on the node, and choose the color from the popup menu Set color. In Figure 1 the terminal nodes are black.

The name shown in the title field underneath the nodes can be edited by clicking these fields. As in Figure 1 we name the two terminals S and T respectively, and leave the other title fields blank. We then arrange the nodes as shown in Figure 1 by dragging the node objects to the appropriate locations.

You can view the *contents* of a node or terminal by opening it. This is done by double-clicking the node. This brings up a dialog box as the one shown in Figure 4. The *name* of the node is shown in the Name textfield. The *state type* of the node can be chosen from the menu State type. There are three possible options: Random, Always failed and Always functioning.

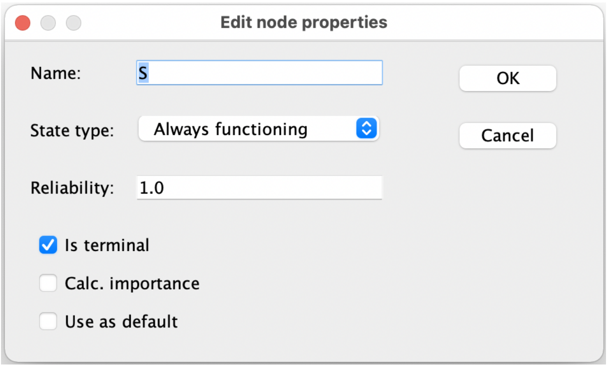

Figure 4: The node dialog box

Since all nodes are assumed to be functioning perfectly, the state type should be Always functioning. The reliability of the node is specified in the Reliability textfield. Note, however, that this value is only used for nodes with state type *random*. Thus, when a node is assumed by be *always functioning*, this textfield does not affect the model of the system and can simply be left as it is. The *terminal status* of a node can be set using the Is terminal checkbox. By using this checkbox we can change whether or not a node should be a terminal or not. Thus, for terminals this box should be left *checked*, while for other nodes this box should be left *unchecked*. If you used the right buttons when creating the nodes and terminals, it should not be necessary to change anything. Still, it is recommended that you verify that the terminal status is correct for all the nodes. The last two checkboxes, Calc. importance and Use as default should be left unchecked.

Creating edges. In order to create an *edge* between two nodes, you drag the first node onto the second node and release the mouse button when the second node becomes highlighted. Create the eight edges of the system as shown in Figure 1. Although not strictly necessary, we recommend that the edges are created in the order $1, 2, \ldots, 8$ as shown in Figure 1.

Note that when you create a new edge, it is *not labled*. In order to create a lable you must *open* the edge. This is done by double-clicking the edge at its midpoint. This is often easier to do if the edge is drawn with a symbol on[1]. To choose e.g., *circular symbols* for the edges, use the menu command: Layout>Edge Symbol>Circle. As a result, the network would look like the

[1]Besides being a simulation software CMCSim™ can also be used as a drawing tool for networks. In order to support different types of network designs CMCSim™ offers different types of edge symbols, including arrows. In all calculations, however, the networks are considered to be *undirected*. Thus, in order to avoid confusion we recommend using a symbol, i.e., a circle, which do not indicate any direction.

one illlustrated in Figure 5. With these circles in place, you can open an

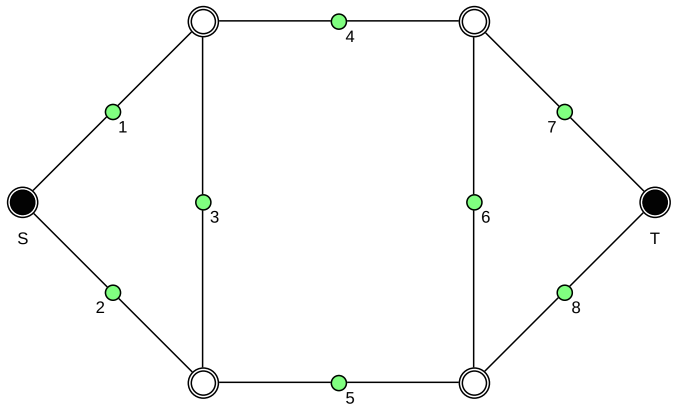

Figure 5: A binary monotone system

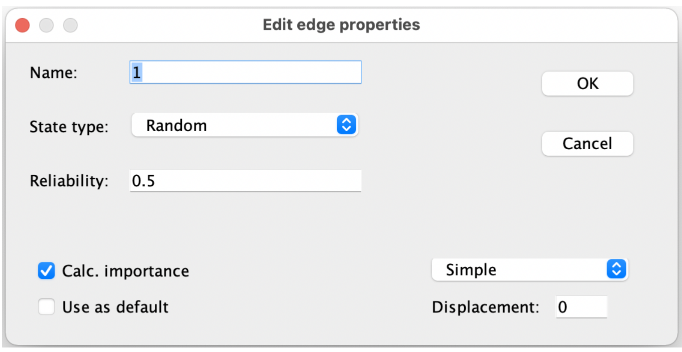

Figure 6: The edge dialog box

edge by double-clicking on the circle symbol on top of it. This brings up a dialog box as the one shown in Figure 6. The Name textfield, State type menu and the Reliability textfield work the same as for nodes. Thus, the *name* of the edge is entered into the Name textfield. Since we assume that the components of the system are the edges of the network, the *state type* should be Random, while the component reliability should be entered into the Reliability textfield. For the edges we leave the Calc. importance checkbox checked and the Use as default checkbox unchecked[2].

[2]The items in the lower right-hand part of the edge dialog box are only relevant when CMCSIM™ is used as a drawing tool, and should be left unchanged in their default states.

a) Let $p_i = 0.5$, $i = 1, \ldots, 8$, and create a model of the system (C, ϕ) using CMCSim™ as explained above. Create a pdf-file showing the system by using the menu command File>Export>As PDF-file.... Make sure that you include the contents of this pdf-file in your assignment.

Running simulations. CMCSim™ can handle several different simulation tasks. To choose which task to run we use the Task type menu at the bottom of the main window. See Figure 7. We start out by using the task types

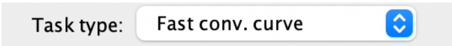

Figure 7: The Task type menu

Crude conv. curve and Fast conv. curve. Both these task types estimate the reliability of the given system by using Monte Carlo simulation. The *crude* method samples N independent binary vectors $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N$, all having the same joint distribution as the component state vector $\boldsymbol{X}$. For each sample, the resulting value of the structure function is calculated. The system reliability $h = E[\phi]$ is then estimated by:

$$\hat{h}_{MC} = \frac{1}{N} \sum_{r=1}^{N} \phi(\boldsymbol{X}_r)$$

When the *fast* method is chosen, the Monte Carlo simulations are done *conditional* on the random variable $S = X_1 + \cdots + X_8$. Thus, we let:

$$\theta_s = E[\phi | S = s], \quad s = 0, 1, \ldots, 8.$$

The reliability h can then be expressed as:

$$h = E[\phi] = \sum_{s=0}^{8} E[\phi | S = s] P(S = s) = \sum_{s=0}^{8} \theta_s P(S = s)$$

CMCSim™ calculates the distribution of S analytically using generating functions, while $\theta_0, \theta_1, \ldots, \theta_8$ are estimated by Monte Carlo simulation. The resulting estimate for h is given by:

$$\hat{h}_{CMC} = \sum_{s=0}^{8} \hat{\theta}_s P(S = s)$$

The corresponding *convergence curves* are obtained by plotting $\hat{h}_{MC}$ and $\hat{h}_{CMC}$ as functions of the sample size N. The convergence curves allow us to examine how fast the Monte Carlo estimates stabilize as the number of samples increase.

Before running the simulations, we must specify the number of simulations, i.e., N. This is entered into the #Simulations textfield at the bottom of the main window. See Figure 8. Ideally, we would choose N as large as possible in order to reduce the uncertainty in the estimates. The simulations run very fast, so even if N is say 100000, we would get the results in seconds. In this assignment we focus on the differences between crude Monte Calo simulations and conditional Monte Carlo simulations. This is easier to see when N is not too large. Thus, we let $N = 1000$.

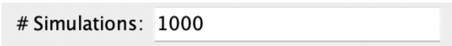

Figure 8: The Simulations textfield

In the simulations we will also estimate the Birnbaum measure of importance for each of the components. Thus, we let the Calc. importance checkbox be checked. See Figure 9. Note that since CMCSIM™ runs separate simulations for each importance measure, this increases the simulation time considerably, especially when the number of simulations, N, is large.

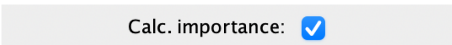

Figure 9: The Calc. importance checkbox

During a Monte Carlo simulation, the progress can be monitored by the Runs field at the bottom of the main window. See Figure 10. When

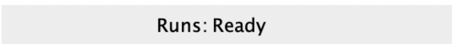

Figure 10: The run counter

the simulations are finished, CMCSIM™ opens a new window where the results are presented as *result nodes*. See Figure 11. The first node is named *System*, and contains the convergence curve for the reliability of the system. The other nodes, named *Comp. 1*, ... , *Comp. 8* respectively, contain convergence curves for the Birnbaum measures of importance. Note that the result nodes for the importance measures will be sorted according to the order which the edges were initially created. Thus, if the edges were created in the order $1, 2, \ldots, 8$ (as previously recommended), the result nodes will automatically be sorted in the same order as well. It is possible to examine

Figure 11: Result nodes

the results by *opening* the nodes. However, it is more convenient to view the results graphically. In order to show a particular convergence curve, select the corresponding node and use the menu command Statistics>Convergence curve. This will bring up a new window where a plot of the convergence curve is shown. See Figure 12. Note that the X-axis ranges from 0 to 100. This scale represents *percentages* of the number of simulation runs. In particular, since N was chosen to be 1000, the plot point with the highest value on the x-axix, i.e., 100, can be interpreted as the estimated value for $N = 1000 \cdot 100\% = 1000$.

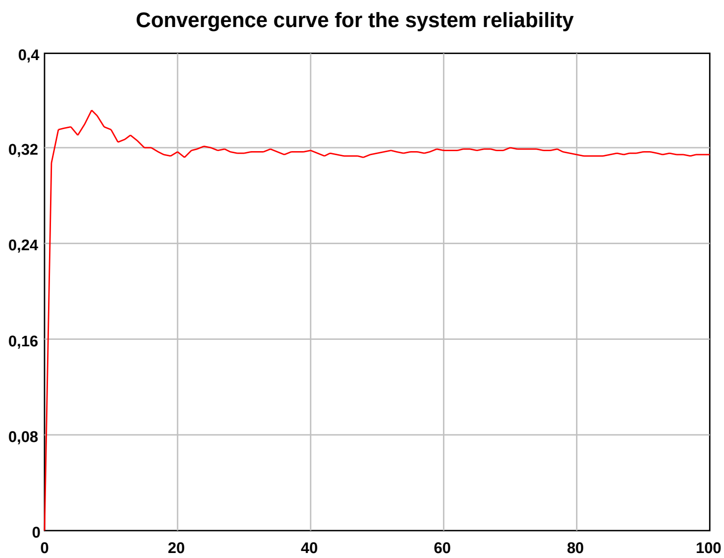

Figure 12: Convergence curve for the system reliability

By selecting multiple nodes it is possible to show convergence for several nodes in the same plot. See Figure 13. The curves will be colored with colors *red*, *green*, *blue*, *purple*, *yellow*, *bluegreen*, *dark red*, *dark green*, *dark blue*, *dark puple*, *dark yellow*, *dark bluegreen*, so that the curve for the first selected node will be *red*, the curve for the second selected node will be *green* etc.

Right-clicking in the middle of the plot brings up a pop-up menu with various formatting options. In particular, it is possible to change the scales of the plot. Thus, e.g., in Figure 12 and Figure 13 the vertical ranges of the plots have been changed to $[0, 0.4]$. Right-clicking the plot title brings up a pop-up menu which allows the title to be edited or hidden[3].

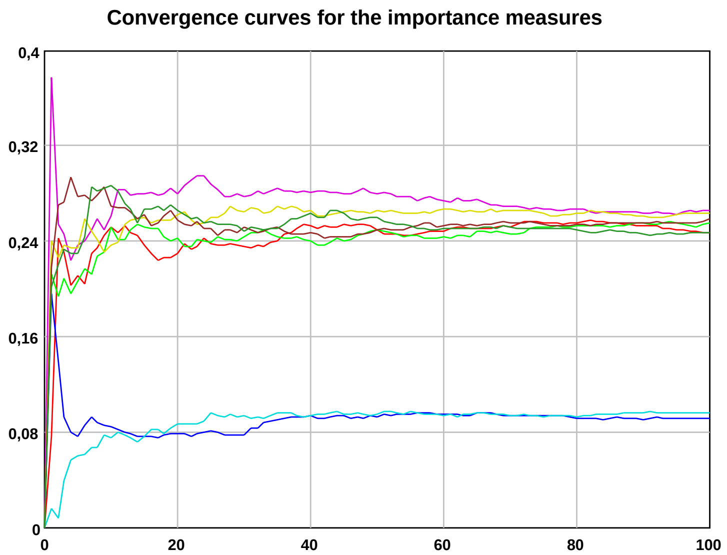

Figure 13: Convergence curve for the importance measures

b) Run two simulations on the created model, one simulation using task type Crude conv. curve and one simulation using task type Fast conv. curve. For each of the two simulations create one plot showing the convergence curve for the *system reliability* (similar to Figure 12), and one plot showing the convergence curves for the importance measure of the eight components (similar to Figure 13). Save all plots as pdf-files by using the menu command File>Export>As PDF-file.... Make sure that you include all the plots in your assignment. Compare the results from the two simulations. Moreover, compare the estimated importance measures for the eight components. Which components appear to be least important?

For each of the two simulations open the *System* node. This brings up a window with two columns of numbers. The left-hand column

[3] If the plot title has been hidden, it can be brought back again by right-clicking in the empty area above the plot and choosing Show title.

contains percentage values (i.e., the x-values of the convergence curve) while the right-hand column contains estimates of the system reliability (i.e., the y-values of the convergence curve). Scroll down to the very last row where the percentage value is 100, and note the corresponding reliability estimate. Include these numbers in your assignment. We will later compare these numbers to the exact reliability.

We now consider the task types Crude path sets, Fast path sets, Crude cut sets and Fast cut sets. Running a Monte Carlo simulation on the first two of these will do a random search for *minimal path sets*, while the last two will do a random search for *minimal cut sets*. When the crude task types are used, the search is completely random. If on the other hand the fast task types are used, the search is done conditionally on the sum S. This means that the algorithm systematically considers sets of sizes corresponding to the possible values of S. As a result the fast method tends to find the minimal path and cut sets faster than when the crude methods are used. Neither of the methods are perfect. For large-scale complex systems one is not guaranteed that all the minimal path or cut sets are found. The methods may also produce non-minimal path and cut sets. For moderate sized networks, however, the methods usually work fine. Moreover, as the number of simulations grows, eventually all minimal sets will be found and the non-minimal sets will be filtered out. It should also be noted that the random search methods tend to find the *most important* path and cut sets, i.e., the sets that contributes the most to the system reliability (or unreliability).

Running a Monte Carlo simulation with any of these four task types will produce result in the form of a single result node named either Path sets or Cut sets depending on the task type. By opening the result nodes the identified sets are shown. Note that it is possible to select all the text in the result window, and then copy this text to the clipboard by the standard key combination, i.e., either Ctrl-C or Cmd-C depending on which operating system that is being used.

c) Find the minimal path and cut sets of the system manually by inspecting the network. Then try to find the same sets by using the four task types Crude path sets, Fast path sets, Crude cut sets and Fast cut sets. Repeat this for $N = 200, 400, 600, 800, 1000$. How many simulations were needed to get correct results when the crude methods were used? How many simulations were needed to get correct results when the fast methods were used? Describe and discuss your findings. Note that since the searches for the minimal path and cuts sets are random, there is no unique answer to these questions.

We now consider the problem of calculating the reliability of the system analytically. In the remaining part of this problem we make the simplifying assumption that all the component reliabilities are equal:

$$P(X_i = 1) = p_i = p, \quad i = 1, \ldots, 8,$$

where p denotes the common component reliability. Our goal is to find $h = E[\phi]$ expressed as a function, $h(p)$, of the common component reliability p. We also introduce the random variable B defined as follows:

$$B = X_3 + X_6$$

d) Given a practical interpretation of B, and explain briefly why the distribution of B is given by:

$$P(B = 0) = (1-p)^2, \quad P(B = 1) = 2p(1-p), \quad P(B = 2) = p^2.$$

e) Show that:

$$\begin{aligned} E[\phi|B = 0] &= p^3 \amalg p^3 \\ E[\phi|B = 1] &= (p^2 \amalg p^2) \cdot (p \amalg p) \\ E[\phi|B = 2] &= (p \amalg p)^3 \end{aligned}$$

and use this in combination with the results from (d) to obtain $h(p)$.

f) Calculate $h(\frac{1}{2})$ and compare the result to the Monte Carlo estimates you found in (b).

g) We now introduce:

b_s = the number of path sets with s components, $\quad s = 0, 1, \ldots, 8.$

Explain why we have that:

$$\theta_s = E[\phi|S = s] = \frac{b_s}{\binom{8}{s}}, \quad s = 0, 1, \ldots, 8.$$

Figure 14: The Calc. importance checkbox unchecked

We go back to CMCSIM™ again, and consider the task type Fast reliability curve. Before running the Monte Carlo simulation, we uncheck the Calc. importance checkbox. See Figure 14.

After running a Monte Carlo simulation with this task type, CMCSIM™ opens a new window where the results are presented as a single node called *System*. As before the result node can be opened by double-clicking it. This brings up a window with the results. In the upper part of this window estimates of the conditional reliabilities, $\theta_0, \theta_1, \ldots, \theta_8$ are given. The remaining results represent the reliability curve, $h(p)$, in the form of two columns. The first column contains the component reliability, p, ranging from 0 to 1, while the second column contains the corresponding values of the reliability curve, $h(p)$.

The reliability curve can also be displayed graphically by selecting the result node *System*, and choosing the menu command Statistics>Reliability curve.

h) Run a Monte Carlo simulation with $N = 1000$ simulations and with task type Fast reliability curve. Display the results graphically in the form of the estimated reliability curve $h(p)$ as explained above. Save the plot as a pdf-file by using the menu command File>Export>As PDF-file.... Make sure that you include the plot in your assignment.

   Moreover, find the Monte Carlo estimates of the conditional reliabilities, $\theta_0, \theta_1, \ldots, \theta_8$ inside the *System* node. Denote these estimates by $\hat{\theta}_0, \hat{\theta}_1, \ldots, \hat{\theta}_8$ respectively.

i) Estimate $b_0, b_1, \ldots, b_8$ by:

$$\hat{b}_s = \text{Round}(\hat{\theta}_s \cdot \binom{8}{s}), \quad s = 0, 1, \ldots, 8.$$

   where the function $\text{Round}(x)$ returns x rounded off to the nearest integer. Explain why $\hat{b}_0, \hat{b}_1, \ldots, \hat{b}_8$ are reasonable estimates of $b_0, b_1, \ldots, b_8$ respectively.

j) Explain why $h(\frac{1}{2})$ can be estimated by:

$$\hat{h} = \frac{\sum_{s=0}^{8} \hat{b}_s}{2^8}$$

Calculate $\hat{h}$ and compare this to the results in (b) and (f). Discuss your findings.