Appendix A: Proofs

Additional Notation and Lemma

For the deterioration vector $\boldsymbol{x} = (x_1, \ldots, x_i, \ldots, x_{n_M})$, define $\boldsymbol{x}^{(i)+1} = (x_1^{(i)+1}, \ldots, x_i^{(i)+1}, \ldots, x_{n_M}^{(i)+1}) = (x_1, \ldots, x_i + 1, \ldots, x_{n_M})$. That is, the deterioration condition of the ith asset in $\boldsymbol{x}^{(i)+1}$ is one unit larger than that in $\boldsymbol{x}$, and all other conditions remain unchanged. Next, we establish Lemma 1.

LEMMA 1. *If P has the IFR property and $v(\boldsymbol{x}', l) = v\big((x'_1, ..., x'_i, ..., x'_{n_M}), l\big)$ is nondecreasing in x'_i, then*

$$\sum_{\boldsymbol{x}' \in \mathcal{K}^{n_M}} \prod_{j \in V_M} P_{x_j, x'_j} \cdot v(\boldsymbol{x}', l) \leq \sum_{\boldsymbol{x}' \in \mathcal{K}^{n_M}} \prod_{j \in V_M} P_{x_j^{(i)+1}, x'_j} \cdot v(\boldsymbol{x}', l).$$

Proof. By the IFR property, for all $k \in \mathcal{K}$ we have

$$\sum_{x'_i = k}^{\Delta} P_{x_i, x'_i} \leq \sum_{x'_i = k}^{\Delta} P_{x_i + 1, x'_i} = \sum_{x'_i = k}^{\Delta} P_{x_i^{(i)+1}, x'_i}. \tag{1}$$

Then, from equation (1) and that $v(\boldsymbol{x}', l)$ is nondecreasing in x'_i we have

$$\sum_{\boldsymbol{x}' \in \mathcal{K}^{n_M}} P_{x_i, x'_i} \cdot \Big(\prod_{j \in V_M \setminus \{i\}} P_{x_j, x'_j} \cdot v(\boldsymbol{x}', l) \Big) \leq \sum_{\boldsymbol{x}' \in \mathcal{K}^{n_M}} P_{x_i^{(i)+1}, x'_i} \cdot \Big(\prod_{j \in V_M \setminus \{i\}} P_{x_j, x'_j} \cdot v(\boldsymbol{x}', l) \Big) \tag{2}$$

$$= \sum_{\boldsymbol{x}' \in \mathcal{K}^{n_M}} P_{x_i^{(i)+1}, x'_i} \cdot \Big(\prod_{j \in V_M \setminus \{i\}} P_{x_j^{(i)+1}, x'_j} \cdot v(\boldsymbol{x}', l) \Big). \tag{3}$$

Inequality (2) is a result of Lemma 4.7.2. in Puterman (2014, see p. 106), and equality (3) holds by the definition of $\boldsymbol{x}^{(i)+1}$. The result follows directly. ■

Proof of Proposition 1

We construct the proof by induction on the iterates of the value iteration algorithm described in Puterman (2014, see p. 161). Let $v^k(\boldsymbol{x}, l)$ represent the cost obtained in the kth iteration of the algorithm. Because the value iteration algorithm converges for any set of initial values, without loss of generality we assume that

$$v^0(\boldsymbol{x}, l) = v_0, \text{ for all } (\boldsymbol{x}, l) \in S.$$

It follows directly that for all $(\boldsymbol{x}, l) \in S$, $v^0\big((x_1, \ldots, x_i, \ldots, x_{n_M}), l\big)$ is nondecreasing in x_i with all other state variables fixed. We now assume that for all $(\boldsymbol{x}, l) \in S$, $v^k\big((x_1, \ldots, x_i, \ldots, x_{n_M}), l\big)$ is also nondecreasing in x_i with all other state variables fixed. Thus, to complete the proof of Proposition 1 we need to show that $v^{k+1}\big((x_1, \ldots, x_i, \ldots, x_{n_M}), l\big)$ is nondecreasing in x_i with all other state variables fixed.

For now, we assume that $l \in V_M$ so that it is feasible to choose the repair action. Recall from Section 2 that $c_R(x_l)$ is nondecreasing in x_l. Then, from Lemma 1 we have

$$v^{k+1}(\boldsymbol{x}, l)$$

$$= \min \left\{ \begin{array}{l} \mathcal{R}(\boldsymbol{x}, l) \equiv c_R(x_l) + c_D + \displaystyle\sum_{j \in V_M \setminus \{l\}} c_D \cdot \mathbf{1}_{\{x_j = \Delta\}} + \lambda \sum_{\substack{\boldsymbol{x}' \in \mathcal{K}^{n_M} \\ \text{s.t. } x'_l = 0}} \prod_{j \in V_M \setminus \{l\}} P_{x_j, x'_j} \cdot v^k(\boldsymbol{x}', l), \\ \mathcal{DN}(\boldsymbol{x}, l) \equiv \displaystyle\sum_{j \in V_M} c_D \cdot \mathbf{1}_{\{x_j = \Delta\}} + \lambda \sum_{\boldsymbol{x}' \in \mathcal{K}^{n_M}} \prod_{j \in V_M} P_{x_j, x'_j} \cdot v^k(\boldsymbol{x}', l), \\ \displaystyle\min_{b:(l,b) \in E} \mathcal{T}_b(\boldsymbol{x}, l) \equiv c_T + \sum_{j \in V_M} c_D \cdot \mathbf{1}_{\{x_j = \Delta\}} + \lambda \min_{b:(l,b) \in E} \sum_{\boldsymbol{x}' \in \mathcal{K}^{n_M}} \prod_{j \in V_M} P_{x_j, x'_j} \cdot v^k(\boldsymbol{x}', b) \end{array} \right\}$$

$$
\begin{aligned}
&\leq \min \left\{ \begin{array}{l}
\mathcal{R}(\boldsymbol{x}^{(i)+1}, l) \equiv c_R(x_l^{(i)+1}) + c_D + \sum_{j \in V_M \setminus \{l\}} c_D \cdot \mathbf{1}_{\{x_j^{(i)+1} = \Delta\}} + \lambda \sum_{\substack{\boldsymbol{x}' \in \mathcal{K}^{n_M} \\ \text{s.t. } x_l' = 0}} \prod_{j \in V_M \setminus \{l\}} P_{x_j^{(i)+1}, x_j'} \cdot v^k(\boldsymbol{x}', l), \\
\mathcal{DN}(\boldsymbol{x}^{(i)+1}, l) \equiv \sum_{j \in V_M} c_D \cdot \mathbf{1}_{\{x_j^{(i)+1} = \Delta\}} + \lambda \sum_{\boldsymbol{x}' \in \mathcal{K}^{n_M}} \prod_{j \in V_M} P_{x_j^{(i)+1}, x_j'} \cdot v^k(\boldsymbol{x}', l), \\
\min_{b:(l,b) \in E} \mathcal{T}_b(\boldsymbol{x}^{(i)+1}, l) \equiv c_T + \sum_{j \in V_M} c_D \cdot \mathbf{1}_{\{x_j^{(i)+1} = \Delta\}} + \lambda \min_{b:(l,b) \in E} \sum_{\boldsymbol{x}' \in \mathcal{K}^{n_M}} \prod_{j \in V_M} P_{x_j^{(i)+1}, x_j'} \cdot v^k(\boldsymbol{x}', b)
\end{array} \right\} \qquad (4) \\
&= v^{k+1}(\boldsymbol{x}^{(i)+1}, l).
\end{aligned}
$$

Inequality (4) holds because $\mathcal{R}(\boldsymbol{x}, l) \leq \mathcal{R}(\boldsymbol{x}^{(i)+1}, l)$, $\mathcal{DN}(\boldsymbol{x}, l) \leq \mathcal{DN}(\boldsymbol{x}^{(i)+1}, l)$, and $\mathcal{T}_b(\boldsymbol{x}, l) \leq \mathcal{T}_b(\boldsymbol{x}^{(i)+1}, l)$ for all b such that $(l, b) \in E$. Hence, the induction hypothesis holds for $n \in \{0, 1, \ldots, k+1\}$. The proof follows similarly when $l \notin V_M$ and only the do nothing and travel actions are allowed, or when $i = l$.

Proof of Theorem 1

Proof under conditions of (*i*). We need to show that if $\mathcal{R}(\boldsymbol{x}, i) < \mathcal{DN}(\boldsymbol{x}, i)$ and $\mathcal{R}(\boldsymbol{x}, i) < \min_{b:(l,b) \in E} \mathcal{T}_b(\boldsymbol{x}, i)$, then $\mathcal{R}(\boldsymbol{x}^{(i)+1}, i) < \mathcal{DN}(\boldsymbol{x}^{(i)+1}, i)$ and $\mathcal{R}(\boldsymbol{x}^{(i)+1}, i) < \min_{b:(l,b) \in E} \mathcal{T}_b(\boldsymbol{x}^{(i)+1}, i)$. First, assume for contradiction that $\mathcal{R}(\boldsymbol{x}^{(i)+1}, i) < \mathcal{DN}(\boldsymbol{x}^{(i)+1}, i)$ does not hold. Hence, we have the following inequalities from $\mathcal{R}(\boldsymbol{x}, i) < \mathcal{DN}(\boldsymbol{x}, i)$ and $\mathcal{R}(\boldsymbol{x}^{(i)+1}, i) \geq \mathcal{DN}(\boldsymbol{x}^{(i)+1}, i)$, respectively:

$$
\begin{aligned}
c_R(x_i) + c_D + \sum_{j \in V_M \setminus \{i\}} c_D \cdot \mathbf{1}_{\{x_j = \Delta\}} + \lambda \sum_{\substack{\boldsymbol{x}' \in \mathcal{K}^{n_M} \\ \text{s.t. } x_i' = 0}} \prod_{j \in V_M \setminus \{i\}} P_{x_j, x_j'} \cdot v(\boldsymbol{x}', i) < \\
\sum_{j \in V_M} c_D \cdot \mathbf{1}_{\{x_j = \Delta\}} + \lambda \sum_{\boldsymbol{x}' \in \mathcal{K}^{n_M}} \prod_{j \in V_M} P_{x_j, x_j'} \cdot v(\boldsymbol{x}', i), \qquad (5)
\end{aligned}
$$

and

$$
\begin{aligned}
c_R(x_i + 1) + c_D + \sum_{j \in V_M \setminus \{i\}} c_D \cdot \mathbf{1}_{\{x_j = \Delta\}} + \lambda \sum_{\substack{\boldsymbol{x}' \in \mathcal{K}^{n_M} \\ \text{s.t. } x_i' = 0}} \prod_{j \in V_M \setminus \{i\}} P_{x_j, x_j'} \cdot v(\boldsymbol{x}', i) \geq \\
\sum_{j \in V_M} c_D \cdot \mathbf{1}_{\{x_j^{(i)+1} = \Delta\}} + \lambda \sum_{\boldsymbol{x}' \in \mathcal{K}^{n_M}} \prod_{j \in V_M} P_{x_j^{(i)+1}, x_j'} \cdot v(\boldsymbol{x}', i). \qquad (6)
\end{aligned}
$$

Subtracting equation (5) from (6), yields

$$
c_R(x_i + 1) - c_R(x_i) > c_D \cdot \mathbf{1}_{\{x_i + 1 = \Delta\}} + \lambda \sum_{\boldsymbol{x}' \in \mathcal{K}^{n_M}} \left\{ \prod_{j \in V_M} P_{x_j^{(i)+1}, x_j'} \cdot v(\boldsymbol{x}', i) - \prod_{j \in V_M} P_{x_j, x_j'} \cdot v(\boldsymbol{x}', i) \right\}. \qquad (7)
$$

The right-hand side of inequality (7) is non-negative by Lemma 1. Moreover, the left-hand side of (7) is 0 because the repair cost is constant under the set of conditions delineated in (i). Hence, inequality (6) cannot hold.

Second, assume for contradiction that $\mathcal{R}(\boldsymbol{x}^{(i)+1}, i) < \min_{b:(l,b) \in E} \mathcal{T}_b(\boldsymbol{x}^{(i)+1}, i)$ does not hold. Hence, we have the following inequalities from $\mathcal{R}(\boldsymbol{x}, i) < \min_{b:(l,b) \in E} \mathcal{T}_b(\boldsymbol{x}, i)$ and $\mathcal{R}(\boldsymbol{x}^{(i)+1}, i) \geq \min_{b:(l,b) \in E} \mathcal{T}_b(\boldsymbol{x}^{(i)+1}, i)$, respectively:

$$
c_R(x_i) + c_D + \sum_{j \in V_M \setminus \{i\}} c_D \cdot \mathbf{1}_{\{x_j = \Delta\}} + \lambda \sum_{\substack{\boldsymbol{x}' \in \mathcal{K}^{n_M} \\ \text{s.t. } x_i' = 0}} \prod_{j \in V_M \setminus \{i\}} P_{x_j, x_j'} \cdot v(\boldsymbol{x}', i) <
$$

$$c_T + \sum_{j \in V_M} c_D \cdot \mathbf{1}_{\{x_j = \Delta\}} + \lambda \min_{b:(l,b)\in E} \sum_{\boldsymbol{x}' \in \mathcal{K}^{n_M}} \prod_{j \in V_M} P_{x_j, x'_j} \cdot v(\boldsymbol{x}', b), \tag{8}$$

and

$$c_R(x_i+1) + c_D + \sum_{j \in V_M \setminus \{i\}} c_D \cdot \mathbf{1}_{\{x_j = \Delta\}} + \lambda \sum_{\substack{\boldsymbol{x}' \in \mathcal{K}^{n_M} \\ \text{s.t. } x'_i = 0}} \prod_{j \in V_M \setminus \{i\}} P_{x_j, x'_j} \cdot v(\boldsymbol{x}', i) \geq$$
$$c_T + \sum_{j \in V_M} c_D \cdot \mathbf{1}_{\{x_j^{(i)+1} = \Delta\}} + \lambda \min_{b:(l,b)\in E} \sum_{\boldsymbol{x}' \in \mathcal{K}^{n_M}} \prod_{j \in V_M} P_{x_j^{(i)+1}, x'_j} \cdot v(\boldsymbol{x}', b). \tag{9}$$

Subtracting equation (8) from (9), yields

$$c_R(x_i+1) - c_R(x_i) > c_D \cdot \mathbf{1}_{\{x_i+1=\Delta\}} + \lambda \left\{ \min_{b:(l,b)\in E} \sum_{\boldsymbol{x}' \in \mathcal{K}^{n_M}} \prod_{j \in V_M} P_{x_j^{(i)+1}, x'_j} \cdot v(\boldsymbol{x}', b) - \min_{b:(l,b)\in E} \sum_{\boldsymbol{x}' \in \mathcal{K}^{n_M}} \prod_{j \in V_M} P_{x_j, x'_j} \cdot v(\boldsymbol{x}', b) \right\}. \tag{10}$$

By Lemma 1, for any b we have

$$\sum_{\boldsymbol{x}' \in \mathcal{K}^{n_M}} \prod_{j \in V_M} P_{x_j^{(i)+1}, x'_j} \cdot v(\boldsymbol{x}', b) \geq \sum_{\boldsymbol{x}' \in \mathcal{K}^{n_M}} \prod_{j \in V_M} P_{x_j, x'_j} \cdot v(\boldsymbol{x}', b),$$

and thus the right-hand side of inequality (10) is non-negative. Moreover, the left-hand side of (10) is 0 because the repair cost is constant under the set of conditions delineated in (i). Hence, inequality (9) cannot hold.

Proof under conditions of (ii). The steps for the proof under conditions of (ii) are very similar to those outlined above. Note inequalities (7) and (10) with non-negative right-hand sides. If $x_i < \Delta - 1$, then the left-hand sides of (7) and (10) are non-positive because the repair cost is assumed to be constant for $x_i < \Delta$. On the other hand, if $x_i = \Delta - 1$, by the conditions delineated in (ii) we have $c_R(x_i+1) - c_R(x_i) \leq c_D$. Whereas, by (7) and (10) we have $c_R(x_i+1) - c_R(x_i) > c_D$. Hence, inequalities (6) and (9) cannot hold.

Proof of Theorem 2

We have

$$v(\boldsymbol{x}, l) = \mathcal{T}_b(\boldsymbol{x}, l) = c_T + \sum_{j \in V_M} c_D \cdot \mathbf{1}_{\{x_j = \Delta\}} + \lambda \mathbb{E}\big[v(\boldsymbol{x}', b) | \boldsymbol{X} = \boldsymbol{x}\big] \tag{11}$$
$$\geq \lambda \mathbb{E}\big[v(\boldsymbol{x}', b) | \boldsymbol{X} = \boldsymbol{x}\big] \tag{12}$$
$$\geq \lambda v(\boldsymbol{x}, b), \tag{13}$$

where $\boldsymbol{X}$ is the vector of random variables denoting asset deterioration conditions. Equation (11) holds by definition (see equation (7)), and inequality (12) holds by the non-negativity of travel and downtime costs. Next, with regard to (13), note that under the assumptions of Theorem 2, the transition probability matrix P has the IFR and upper triangular properties. The upper triangular property ensures that assets transition to deterioration levels that are greater than or equal to the current levels. Moreover, the IFR property

is a sufficient condition for the result established in Proposition 1. Consequently, inequality (13) holds by Proposition 1. The first result of Theorem 2 follows directly.

For the second result, we first establish the proof under conditions of (i) followed by that under conditions of (ii).

Proof under conditions of (i). Assume for contradiction that if $v(\boldsymbol{x},l) < \lambda v(\boldsymbol{x},b)$, then $v(\boldsymbol{x},l) \geq \mathcal{T}_b(\boldsymbol{x},l)$. Note that by (4), the latter inequality holds by equality. That is,

$$v(\boldsymbol{x},l) = \mathcal{T}_b(\boldsymbol{x},l) < \lambda v(\boldsymbol{x},b). \tag{14}$$

Moreover, The result follows because inequality (14) contradicts (13).

Proof under conditions of (ii). The steps of the proof are very similar to those under conditions of (i) with the exception that the inequality in (14) is not strict whereas the inequalities in (12) and (13) are strict because $c_T > 0$.

Appendix B: Control-Limit Rule Violation

Here we present two numerical examples in which the conditions of Theorem 1, and thus the control-limit structure, are violated. These resulting optimal policies are depicted in Figure 1. Let $c_R(0) = 0$, $c_R(1) = 0$, $c_R(2) = 40$, $c_D = 41$, $c_T = 0.5$, and

$$P = \begin{bmatrix} 0.98 & 0.01 & 0.01 & 0 \\ 0 & 0.96 & 0.03 & 0.01 \\ 0 & 0 & 0.95 & 0.05 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Moreover, we let $c_R(3) = 40$ in Figure 1a and $c_R(3) = 5000$ in Figure 1b. The control-limit structure is violated in Figure 1a because, all else held equal, it is optimal to repair Asset 1 in conditions 1 and 3, but not in condition 2. In Figure 1b it is optimal to repair Asset 1 in conditions 1 and 2, but not in condition 3, i.e., it is optimal to abandon Asset 1 once it reaches condition 3.

Appendix C: Downtime and Graph Centrality

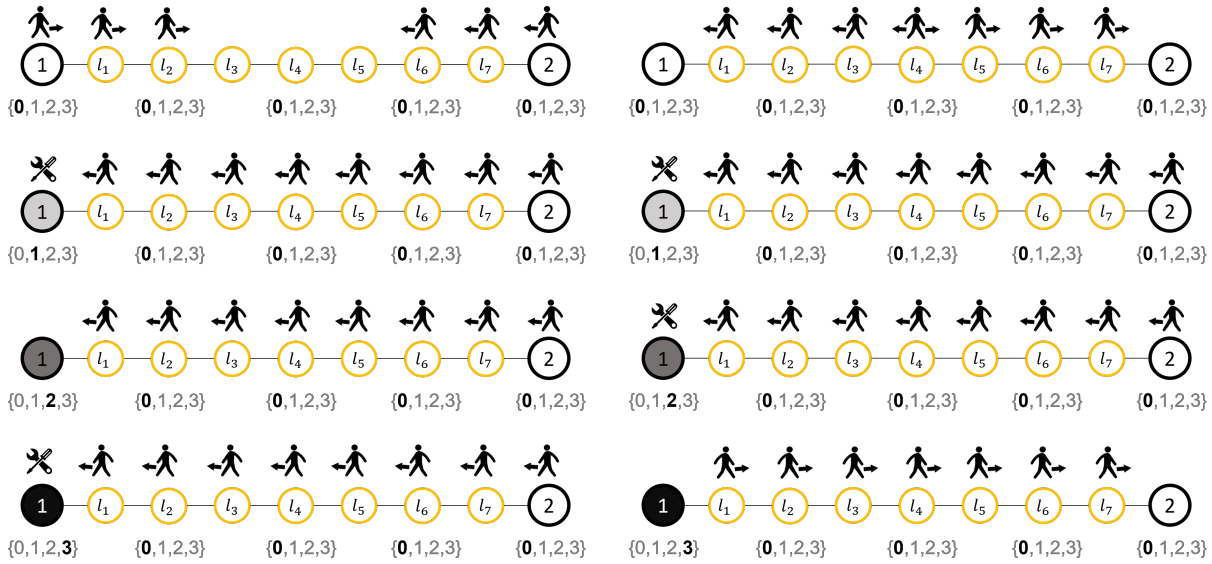
In the examples of Section 5 all assets are in equally central locations, and thus, their average percentage of downtime is almost equal in the long run. In this section, we study the relationship between the centrality of an asset's location and its downtime as well as how the downtime is affected by the graph structure.

Specifically, we consider six assets connected through four different graph configurations, namely, linear, grid, tree, and star configurations. We assume parameter values of $c_R(0) = 1$, $c_R(1) = 2$, $c_R(2) = 3$, $c_D = 1$, $c_T = 1$, $\lambda = 0.995$, and

$$P = \begin{bmatrix} 0.85 & 0.1 & 0.05 \\ 0 & 0.85 & 0.15 \\ 0 & 0 & 1 \end{bmatrix}.$$

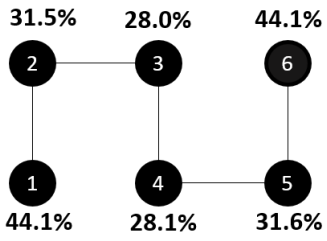
We run a simulation study for 600000 units of time and record the average percentage of downtime of all assets in each graph configuration; the results are depicted in Figure 2.

In Figure 2, first note that assets in less central locations incur more downtime compared to assets in more central locations. In fact, an asset's downtime decreases in its closeness centrality; recall equation (8). Second, assets on the grid graph have the least amount of downtime compared to assets on the other graph configurations, seemingly due to the high level of connectivity in the grid graph (Żochowska and Soczówka 2018). These results imply that assets' downtime is affected by both their relative centralities and network configuration.

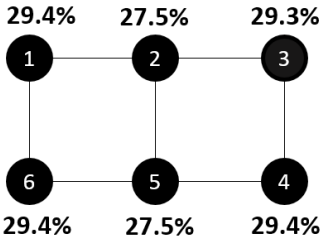


(a) $c_R(3) = 40$ (b) $c_R(3) = 5000$

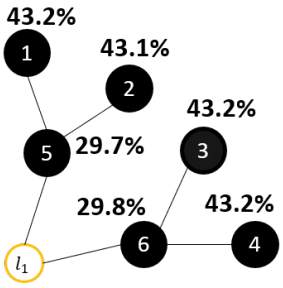
Figure 1 **Excerpt of the optimal policies for the examples in Appendix B. The conditions of Theorem 1 are violated, and thus, the control-limit structure does not hold for Asset 1 in either Example (a) or (b). Moreover, in Example (b), assets are abandoned upon failure.**



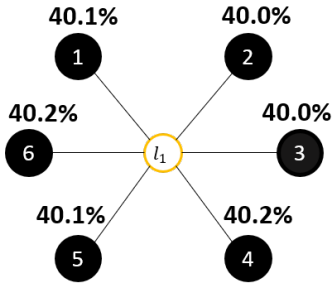
(a) Linear



(b) Grid



(c) Tree

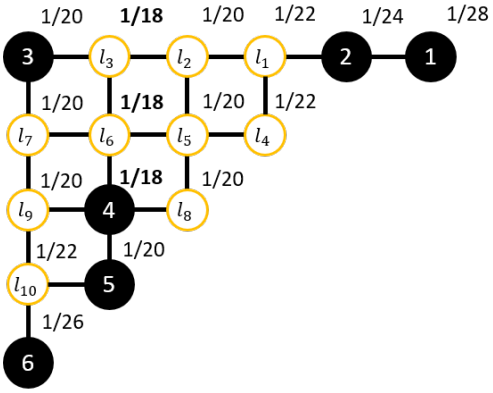


(d) Star

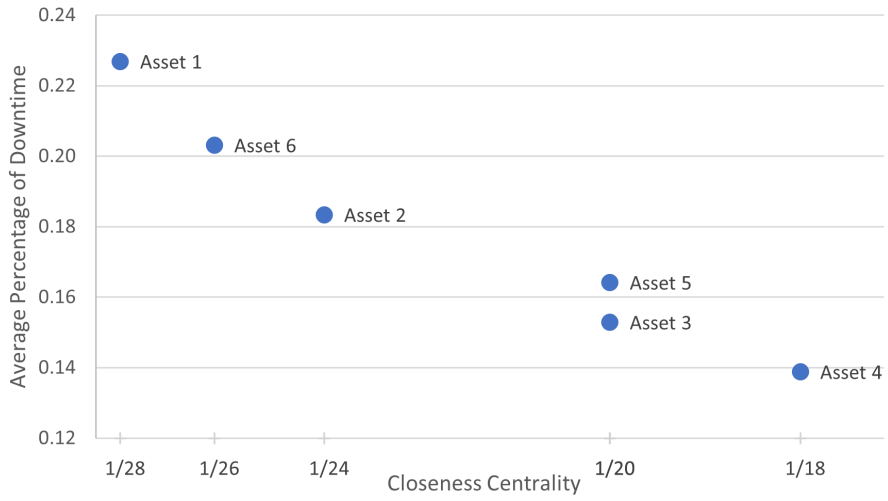
Figure 2 **Average percentage of downtime for six assets in different graph configurations. The assets incur the lowest downtime in the grid graph due to the high connectivity. Moreover, in each graph, asset downtime is nondecreasing in closeness centrality.**

Next, we present another example to examine the relationship between closeness centrality and downtime. Here, we let $c_R(0) = 1$, $c_R(1) = 2$, $c_R(2) = 3$, $c_D = 1$, $c_T = 1$, $= 0.995$, and

$$P = \begin{bmatrix} 0.95 & 0.03 & 0.02 \\ 0 & 0.95 & 0.05 \\ 0 & 0 & 1 \end{bmatrix}.$$



(a) Graph for the example of Appendix C. Closeness centrality score, see equation (8), is labeled on the top of each node.



(b) Percentage of downtime versus closeness centrality of each asset

Figure 3 **Plot of the average percentage of downtime against the closeness centrality of six assets with the graph configuration depicted in (a). Asset downtime is decreasing in closeness centrality.**

We run a simulation study for 2 million units of time to record the average percentage of downtime for each asset. Figure 3a depicts the graph configuration and Figure 3b plots the percentage of downtime against the closeness centrality of each asset.

The plot in Figure 3b implies that asset downtime decreases in the measure of closeness centrality. Furthermore, note that Asset 5 incurs a higher percentage of downtime compared to Asset 3 even though their measures of closeness centrality are equal. This difference can be explained by other measures of centrality such as eccentricity (i.e., the longest shortest path between the node of interest and all other asset nodes). In this example, Asset 5 has a larger eccentricity index compared to Asset 3.

Appendix D: Deterioration Probability Matrices of Section 6

In the computational study of Section 6.3, we let the deterioration probability matrices be as follows:

$$P = \begin{bmatrix} 0.988 & 0.01 & 0.02 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.988 & 0.01 & 0.02 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.988 & 0.01 & 0.02 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.988 & 0.01 & 0.02 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.988 & 0.01 & 0.02 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.988 & 0.01 & 0.02 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.988 & 0.01 & 0.02 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.988 & 0.01 & 0.02 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.988 & 0.12 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$P' = \begin{bmatrix} 0.98 & 0.012 & 0.005 & 0.002 & 0.001 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.98 & 0.012 & 0.005 & 0.002 & 0.001 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.98 & 0.012 & 0.005 & 0.002 & 0.001 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.98 & 0.012 & 0.005 & 0.002 & 0.001 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.98 & 0.012 & 0.005 & 0.002 & 0.001 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.98 & 0.012 & 0.005 & 0.002 & 0.001 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.98 & 0.012 & 0.005 & 0.003 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.98 & 0.012 & 0.008 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.98 & 0.02 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$P'' = \begin{bmatrix} 0.97 & 0.021 & 0.006 & 0.002 & 0.001 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.97 & 0.021 & 0.006 & 0.002 & 0.001 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.97 & 0.021 & 0.006 & 0.002 & 0.001 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.97 & 0.021 & 0.006 & 0.002 & 0.001 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.97 & 0.021 & 0.006 & 0.002 & 0.001 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.97 & 0.021 & 0.006 & 0.002 & 0.001 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.97 & 0.021 & 0.006 & 0.003 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.97 & 0.021 & 0.009 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.97 & 0.03 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

and

$$P''' = \begin{bmatrix} 0.95 & 0.041 & 0.006 & 0.002 & 0.001 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.95 & 0.041 & 0.006 & 0.002 & 0.001 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.95 & 0.041 & 0.006 & 0.002 & 0.001 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.95 & 0.041 & 0.006 & 0.002 & 0.001 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.95 & 0.041 & 0.006 & 0.002 & 0.001 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.95 & 0.041 & 0.006 & 0.002 & 0.001 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.95 & 0.041 & 0.006 & 0.003 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.95 & 0.041 & 0.009 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.95 & 0.05 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Appendix E: Choosing the Simulation Run Length

We use simulation studies throughout the paper to quantify, e.g., the proportion of time spent idling at different nodes (Section 4.3), operational metrics (Section 5), and average cost-rate incurred under the optimal policy and the heuristics (Section 6). To specify a proper simulation run length, we consider the instance size and the computational burden of the problem at hand. Although not presented in these sections, we set the simulation runs long enough such that the results produced in various runs are reasonably close.

Take a large graph instance of Section 6.3 as an example. Table 1 presents the confidence intervals over the expected cost-per-unit time incurred under the three heuristics. In particular, for each heuristic policy, we repeat the simulation 20 times for the same problem instance and record the expected cost-per-unit times. Then, we compute the 95% confidence interval using the t-distribution. We conduct this study for the benchmark scenario with transition probability matrix P as well as matrices P^{+}, P^{++} and P^{+++} (recall Appendix D).

Table 1 suggests that the 200000 simulation run length set in Section 6.3 produces sufficiently tight confidence intervals across the three heuristics and different values of transition probabilities. We can also argue that there is not much benefit in further increasing this run length to 300000 time units because the widths of the intervals are very close.

Please note that the instance sizes in the other studies (e.g., Section 4.3, Section 5, Section 6.2) are significantly smaller than those in Section 6.3. Therefore, simulation run lengths set for those studies are also sufficiently long and should in fact produce tighter confidence intervals than those presented in Table 1.

Simulation	P			P^{+}			P^{++}			P^{+++}		
time units	CM	CR	DR	CM	CR	DR	CM	CR	DR	CM	CR	DR
20000	27.21	13.36	2.79	18.24	19.14	7.71	16.36	17.31	10.79	16.82	16.44	19.70
	± 2.49	± 0.09	± 0.47	± 0.72	± 1.66	± 0.09	± 0.08	± 0.83	±0.15	± 1.43	± 1.61	± 0.89
200000	27.01	12.96	2.80	18.26	19.06	7.70	17.12	17.02	10.79	16.21	16.01	19.71
	± 0.89	± 0.47	± 0.09	± 0.44	± 0.42	± 0.05	± 0.05	± 0.40	±0.06	± 0.11	± 0.44	± 0.06
300000	27.07	12.93	2.80	18.22	19.01	7.70	17.09	17.01	10.78	16.23	16.15	19.70
	± 0.77	± 0.45	± 0.03	± 0.42	± 0.41	± 0.03	± 0.04	± 0.35	±0.02	± 0.10	± 0.32	± 0.03

Table 1 95% Confidence intervals over cost-per-unit time incurred under each heuristic policy for a problem instance of Section 6.3 and different simulation run lengths. The parameter setting is the same as the benchmark setting, except that we change the transition probabilities in the subsequent columns (recall that the values of P-P^{+++} are set according to Appendix D.) Note that the simulation run length of 200000 time units (as set in Section 6.2) produces sufficiently tight confidence intervals.

Lastly, recall that we consider a warm-up period in our simulations during which we let the states of the MDP transition according to the simulation, but, we calculate the metrics of interest only after the warm-up period. This extra step ensures that we only capture the metrics during the stationary states of the model and eliminate the effect of the nonstationary states.

We base our choice of the warm-up period largely on the deterioration transition probabilities because those probabilities dictate how quickly the Markov Process transitions into the stationary states. However, we want to note that our simulation run times are set long enough to mitigate the effect of the nonstationary states (especially because our metrics of interest are reported as averages across the time units). Therefore, our choices of warm-up period lengths affect the reported metrics by only a negligible amount. That being said, we believe that setting a nonzero warm-up period to rule out the beginning states of the simulation is a good practice in general and can make up for inadequate run lengths.

Appendix F: A Report on Computation Times

We code our studies to Python 3.7 and perform the experiments on a Windows PC with a 3.7 GHz CPU and 12 GB RAM. We solve the optimality equation (4) using the value iteration algorithm from Python's pymdptoolbox module. The pymdptoolbox value iteration algorithm performs notably faster than our implementation of the value iteration algorithm; hence, we use this toolbox in all of the experiments.

Computation Time for the Performance Under the Optimal Policy. In the numerical study of Section 6.2, the overall run time of solving for an optimal policy and computing the expected cost-rate is comprised of three main components. *i*) Constructing the transition and reward matrices of the MDP that we then feed into the value iteration algorithm. (We want to note that even though the construction of these matrices takes a considerable portion of time, the operations of the value iteration algorithm are performed much faster in matrix forms.) *ii*) The termination of the value iteration algorithm. *iii*) The time spent simulating the performance of the optimal policy.

We record the overall time spent for calculating the expected cost-rate under the optimal policy and the benchmark parameter setting for the 10 problem instances in Table 4. The average run times of the three components mentioned above are 5, 7, and 10 seconds, respectively. Note that in the examples of Section 6.2, we assume 4 assets with 4 possible deterioration conditions and 25 possible locations for the maintenance resource. Consequently, the run times are short due to the small problem size and would otherwise increase for larger problem instances. For example, if we assume 6 assets instead, then the average run time for constructing the matrices (i.e., component *i* above) and terminating the value iteration algorithm (i.e., component *ii* above) increases to 108, 184 seconds, respectively.

Computation Time for the Performance Under the Heuristic Policies. In Sections 6.2 and 6.3, we report the performance of the heuristic policies. For these policies, we run a simulation for different combinations of idling (ι), travel (τ), and maintenance thresholds (μ). (Note that we no longer require solving for the optimal policy.) Please recall our discussions in Section 6.1 that because the cost-per-unit time is seemingly unimodal with respect to the threshold values, we do not need to run a simulation for all the possible combinations of the above thresholds, in that we can stop the search for the best values once we reach a local minimum. Hence, the overall computation time is comprised of the time spent for one simulation times the number of possible combinations we need to iterate before finding a local minimum. The latter is affected by the parameter values, the graph structure, and the overall problem setting.

In the small problem instances of Section 6.2 where we run the simulation for 40000 time units, under the benchmark scenario (please refer to Table 4), the average time spent for one simulation run under the three heuristics is 10 seconds. Moreover, the overall time spent for the CM, CR, and DR heuristic policies is 51, 103, and 62 seconds, respectively. Note that the overall run time of the CR policy is larger than that of the CM and DR policies. This observation holds because the CR policy requires optimizing for three threshold values ι, τ, and μ, whereas the CM and DR policies require optimizing for two threshold values (ι and τ for CM, ι and μ for DR).

Lastly, in the large problem instances of Section 6.2 where we run the simulation for 200000 time units, under the benchmark scenario and the tree configuration (please refer to Figure 10), the average time spent

for one simulation run under the three heuristics is 57 seconds. Moreover, the overall time spent for the CM, CR, and DR heuristic policies is 232, 457, and 301 seconds, respectively. In the grid configuration (please refer to Figure 11), the average time spent for one simulation run under the three heuristics is also approximately 57 seconds. The overall time spent for the CM, CR, and DR heuristic policies is 251, 480, and 417 seconds, respectively.

Appendix G: Formal Definition for Destination Asset

Recall from Section 6.1 that under the Committed Maintenance and Committed Routing heuristic policies, the maintenance resource "commits" to traveling to a destination asset and is not allowed to change course toward another asset until the destination asset is repaired.

When simulating these heuristic policies, to update the destination asset in the current time unit, we require the state of the MDP in the current time unit and the destination asset specified in the previous time unit, here referred to as d_{previous}. Based on our definition, we update the destination asset at each time unit as below:

$$d(\boldsymbol{x}, l, d_{\text{previous}}) = \begin{cases} \emptyset & \text{if the action is repairing } j' = d_{\text{previous}} \in V_M \text{ and } \max\limits_{j \in \{1,\ldots,n_M\}} x_j \leq \iota^*, \\ j \in V_M & \text{if the action is repairing } \ j' = d_{\text{previous}} \in V_M, \\ & \quad d_{\text{previous}} = \emptyset, x_j \geq \tau^* \text{ and } j \text{ is the closest asset to } l, \\ d_{\text{previous}} & \text{otherwise} \end{cases} \tag{15}$$

Equation (15) implies that the destination asset is only updated once the maintenance resource repairs that asset. Note that in the first time unit, we let $d_{\text{previous}} = \emptyset$ if no assets are deteriorated worse than τ^*; otherwise we let it be the closest asset with a deterioration condition worse than τ^*.

References

Puterman ML, 2014 *Markov Decision Processes: Discrete Stochastic Dynamic Programming* (John Wiley & Sons).

Żochowska R, Soczówka P, 2018 *Analysis of selected transportation network structures based on graph measures. Zeszyty Naukowe. Transport/Politechnika Śląska* .