Supplementary Material: Emulation of utility functions over a set of permutations: sequencing reliability growth tasks

July 10, 2017

1 Proof of Proposition 1

The possible values of Kendall's tau distance between any two permutations of length R are $0, 1, \ldots, T$, where $T = R(R-1)/2$. Suppose we order the sequences from highest expected utility to lowest expected utility and, without loss of generality, suppose the ordering is denoted $(1, 2, \ldots, R)$. We name this the base ordering of the sequences.

We can find the total number of permutations, $C_{R,\delta}$, which are at most distance δ from the base ordering. The Kendall's tau distance to the base ordering is the number of pairs of $j, j^{'}$ such that $j < j^{'}$ but $j^{'}$ appears before j in the permutation. Any pair $j, j^{'}$ which has this property is known as an inversion. Any permutation of the base ordering is a permutation of $1, \ldots, R-1$ with R inserted. If we insert R at position l, this creates $R - l$ new inversions. Thus the previous permutation must have had at most $\delta - (R - l)$ inversions. Thus,

$$\begin{aligned} C_{R,\delta} &= \sum_{l=\max(1,R-\delta)}^{\delta} C_{R-1,\delta+R-l}, \\ &= \sum_{l=\delta-R+1}^{\delta} C_{R-1,l}. \end{aligned}$$

From this, we can calculate the number of permutations of the base ordering with a Kendall's tau distance of δ as $N_{R,\delta} = C_{R,\delta} - C_{R,d\delta-1}$. Now, we are interested in the proportion of these permutations which maintain the sequence with highest utility as the first element, second element, etc. Thus, if we know the Kendall's tau distance of function $f(\cdot)$, this will give us the probability that the sequence with highest value of $f(\cdot)$ also has the highest expected utility, etc.

Consider the proportion which maintain sequence 1 as the first element. This implies that in positions $2, \ldots, R$ we require δ inversions and the number of ways this can happen is $N_{R-1,\delta}$. Similarly, the number of ways sequence 1 can

be the second element is $N_{R-1,\delta-1}$ and the number of ways it can be the m'th element is $N_{R-1,\delta-m+1}$. The proportions in each case are given by dividing by the number of permutations with Kendall's tau distance δ, $N_{R,\delta}$. This gives the result.

2 Proof of Proposition 2

The probability that the optimal sequence is in the m candidates, from Corollary 1, is

$$\sum_{m=1}^{M} \frac{C_{R-2,\delta-m+1}}{C_{R-1,\delta}}.$$

Clearly, $C_{R-2,0} = 1$ and $\delta - m + 1 \geq 0 \forall m = 1, \ldots, M$. Specifically, $\delta - M + 1 \geq 0$ and so $M \leq \delta + 1$. Since the optimal sequence must be one of the R permutations,

$$\sum_{m=1}^{\delta+1} \frac{C_{R-2,\delta-m+1}}{C_{R-1,\delta}} = 1$$

and

$$\sum_{m=1}^{\delta} \frac{C_{R-2,\delta-m+1}}{C_{R-1,\delta}} = 1 - \frac{1}{C_{R-1,\delta}}.$$

3 Prior expectation of the reliability under planned development tasks

Following some development tasks, the system reliability, assuming that every fault found is corrected without introducing further faults, is given by

$$\begin{aligned} R(t, \boldsymbol{z}) &= \prod_{i=1}^{I} R_i(t)^{I[z_i > d_i]} \\ &= \prod_{i=1}^{I} R_i(t)^{z_i(1-d_i)} \\ &= \prod_{i=1}^{I} \{1 - z_i(1-d_i)[1 - R_i(t)]\}, \end{aligned}$$

where $I[z_i > d_i]$ is an indicator function which takes the value 1 if $z_i > d_i$ and zero otherwise. Taking the expectation of this with respect to $\boldsymbol{Z} \mid \boldsymbol{D}$, where

$\boldsymbol{Z} = (Z_1, \ldots, Z_I)'$ and $\boldsymbol{D} = (D_1, \ldots, D_I)'$,

$$\begin{aligned} \mathrm{E}_{\boldsymbol{Z}|\boldsymbol{D}}[R(t,\boldsymbol{z})] &= \mathrm{E}_{\boldsymbol{Z}|\boldsymbol{D}}\left[\prod_{i=1}^{I}\{1 - z_i(1-d_i)[1-R_i(t)]\}\right] \\ &= \prod_{i=1}^{I}\mathrm{E}_{Z_i|D_i}\left[\{1 - z_i(1-d_i)[1-R_i(t)]\}\right] \\ &= \prod_{i=1}^{I}\mathrm{E}_{Z_i|D_i}\left[\tilde{R}(t,z_i,d_i)\right], \end{aligned}$$

as $Z_i \mid D_i$ is independent of $Z_j \mid D_j$. Taking the expectation with respect to $\boldsymbol{D}$, gives

$$\mathrm{E}_{\boldsymbol{D}}\left\{\mathrm{E}_{\boldsymbol{Z}|\boldsymbol{D}}\left[R(t,\boldsymbol{z})\right]\right\} = \prod_{i=1}^{I}\mathrm{E}_{D_i}\left\{\mathrm{E}_{Z_i|D_i}\left[\tilde{R}(t,z_i,d_i)\right]\right\}.$$

We can express the expectation concerning fault i as

$$\begin{aligned} \mathrm{E}_{D_i}\left\{\mathrm{E}_{Z_i|D_i}\left[\tilde{R}(t,z_i,d_i)\right]\right\} &= \Pr(D_i=1)\Big[\Pr(Z_i=1 \mid D_i=1)\tilde{R}(t,1,1) \\ &\quad + \Pr(Z_i=0 \mid D_i=1)\tilde{R}(t,0,1)\Big] \\ &\quad + \Pr(D_i=0)\Big[\Pr(Z_i=1 \mid D_i=0)\tilde{R}(t,1,0) \\ &\quad + \Pr(Z_i=0 \mid D_i=0)\tilde{R}(t,0,0)\Big]. \quad (1) \end{aligned}$$

We can evaluate each of the quantities in this expression. They are

$$\Pr(D_i=0) = 1-\lambda_i\left[1 - \prod_{j=1}^{J}(1-p_{i,j})^{\kappa_j}\right],\ \Pr(D_i=1) = \lambda_i\left[1 - \prod_{j=1}^{J}(1-p_{i,j})^{\kappa_j}\right],$$

for the unconditional probabilities,

$$\Pr(Z_i=1 \mid D_i=0) = \frac{\lambda_i\prod_{j=1}^{J}(1-p_{i,j})^{\kappa_j}}{1-\lambda_i\left[1-\prod_{j=1}^{J}(1-p_{i,j})^{\kappa_j}\right]},\quad \Pr(Z_i=1 \mid D_i=1) = 1$$

$$\Pr(Z_i=0 \mid D_i=0) = \frac{1-\lambda_i}{1-\lambda_i\left[1-\prod_{j=1}^{J}(1-p_{i,j})^{\kappa_j}\right]}\quad \Pr(Z_i=0 \mid D_i=1) = 0$$

for the conditional probabilities, $\tilde{R}(t,1,1) = \tilde{R}(t,0,1) = \tilde{R}(t,0,0) = 1$ and $\tilde{R}(t,1,0) = R_i(t)$. Thus the expectation in (1) is

$$\begin{aligned}
&\left\{\lambda_i\left[1-\prod_{j=1}^{J}(1-p_{i,j})^{\kappa_j}\right]\right\}[1\times 1+0\times 1]\\
&+\left\{1-\lambda_i\left[1-\prod_{j=1}^{J}(1-p_{i,j})^{\kappa_j}\right]\right\}\left[\frac{\lambda_i\prod_{j=1}^{J}(1-p_{i,j})^{\kappa_j}}{1-\lambda_i\left[1-\prod_{j=1}^{J}(1-p_{i,j})^{\kappa_j}\right]}R_i(t)+\frac{1-\lambda_i}{1-\lambda_i\left[1-\prod_{j=1}^{J}(1-p_{i,j})^{\kappa_j}\right]}\times 1\right]\\
&=\lambda_i-\lambda_i\prod_{j=1}^{J}(1-p_{i,j})^{\kappa_j}+\lambda_i\prod_{j=1}^{J}(1-p_{i,j})^{\kappa_j}R_i(t)+1-\lambda_i\\
&=1-\lambda_i(1-R_i(t))\prod_{j=1}^{J}(1-p_{i,j})^{\kappa_j}.
\end{aligned}$$

Substituting this into the expectation of interest, gives the result:

$$\mathrm{E}_{\boldsymbol{Z}|\boldsymbol{D}}[R(t,\boldsymbol{z})]=\prod_{i=1}^{I}\left[1-\lambda_i(1-R_i(t))\prod_{j=1}^{J}(1-p_{i,j})^{\kappa_j}\right]$$

4 Example of theoretical results

Suppose that there are 3 tasks which we wish to sequence. This means that there are $3! = 6$ possible sequences of the 3 tasks. Define the value of the surrogate function for sequence $\boldsymbol{x}_i$ to be $f_i = f(\boldsymbol{x}_i)$ and the expected utility to be $u_i = U(\boldsymbol{x}_i)$, for $i = 1,\ldots,6$. Suppose that the ordering of the sequences using the surrogate function is $f_1 > f_2 > f_3 > f_4 > f_5 > f_6$, and that the Kendall's tau distance between the expected utility and the surrogate function is 2. This means that $\tau = 11/15$.

We can calculate the values of $C_{R,\delta}$ and $N_{R,\delta}$ for $R = 1,\ldots,6$ and $\delta = 1,2,3$, which will allow us to find the probability that the sequence with the maximum expected utility is in the highest M places ordered by surrogate function. The values are in Table 1.

	$C_{R,\delta}$				$N_{R,\delta}$		
R/δ	1	2	3	R/δ	1	2	3
1	1	1	1	1	0	0	0
2	2	2	2	2	1	0	0
3	3	5	6	3	2	2	1
4	4	9	15	4	3	5	6
5	5	14	29	5	4	9	15
6	6	20	49	6	5	14	29

Table 1: Tables showing the values of $C_{R,\delta}$ and $N_{R,\delta}$ for $R = 1,\ldots,6$ and $\delta = 1,2,3$.

We can use the result in Corollary 1 in the main paper as $\delta = 2$, $R = 6$ and so $\delta < R - 1$. We see that $C_{5,2} = 14$, $C_{4,2} = 9$, $C_{4,1} = 4$ and we know that

$C_{4,0} = 1$. Therefore, the probabilities that the sequence with highest expected utility is in the sequences with the highest M values of $f(\cdot)$, for $M = 1, \ldots, 6$ are $(9/14, 13/14, 14/14, 14/14, 14/14, 14/14)$, respectively. We see that, choosing $M = \delta + 1 = 3$, we obtain the optimal sequence with probability 1 as shown by Proposition 2 in the main paper.

In this simple case we can find the probabilities directly. There are $6! = 720$ possible orderings of the expected utilities $u_1, \ldots, u_6$. Of these, 14 have a Kendall correlation $\tau = 11/15$ with the surrogate function ordered as above. Suppose that the true ordering of the expected utilities is some permutation of $u_1, \ldots, u_6$, say $u_{(1)} < u_{(2)} < u_{(3)} < u_{(4)} < u_{(5)} < u_{(6)}$. Then the orderings below of the sequences have the required Kendall correlation (given in order, from first to last).

We see that, of the 14 sequences, in 9 cases the sequence with the highest value for the expected utility is in the first position in the ordering (red), in 4 cases it is in the second position (blue) and in one case it is in the third position. This results in the probabilities given above.

$u_{(1)}, u_{(2)}, u_{(3)}, u_{(5)}, u_{(6)}, u_{(4)}$	$u_{(1)}, u_{(2)}, u_{(3)}, u_{(6)}, u_{(4)}, u_{(5)}$
$u_{(1)}, u_{(2)}, u_{(4)}, u_{(3)}, u_{(6)}, u_{(5)}$	$u_{(1)}, u_{(2)}, u_{(4)}, u_{(5)}, u_{(3)}, u_{(6)}$
$u_{(1)}, u_{(2)}, u_{(5)}, u_{(3)}, u_{(4)}, u_{(6)}$	$u_{(1)}, u_{(3)}, u_{(2)}, u_{(4)}, u_{(6)}, u_{(5)}$
$u_{(1)}, u_{(3)}, u_{(2)}, u_{(5)}, u_{(4)}, u_{(6)}$	$u_{(1)}, u_{(3)}, u_{(4)}, u_{(2)}, u_{(5)}, u_{(6)}$
$u_{(1)}, u_{(4)}, u_{(2)}, u_{(3)}, u_{(5)}, u_{(6)}$	$u_{(2)}, u_{(1)}, u_{(3)}, u_{(4)}, u_{(6)}, u_{(5)}$
$u_{(2)}, u_{(1)}, u_{(3)}, u_{(5)}, u_{(4)}, u_{(6)}$	$u_{(2)}, u_{(1)}, u_{(4)}, u_{(3)}, u_{(5)}, u_{(6)}$
$u_{(2)}, u_{(3)}, u_{(1)}, u_{(4)}, u_{(5)}, u_{(6)}$	$u_{(3)}, u_{(1)}, u_{(2)}, u_{(4)}, u_{(5)}, u_{(6)}$

5 Splitting the training sample: an illustrative example

In the illustrative example of Section 4 in the main paper, we have a training sample of size $N = 60$. Here we split this training sample into $s = 3$ sub-samples of size 20 and fit a regression-adjusted surrogate (using the Benter-Pearson model-correlation combination) to each. Figure 1 plots the logit-transformed expected utilities against the fitted values from the regression-adjusted emulators in each of the sub-samples. For comparison, Figure 1 also contains the fitted values from the regression-adjusted surrogate based on all $N = 60$ training samples.

It can be seen that there is a good correspondence between the emulator values based on the sub-samples and the emulator values based on the full training sample, as we would expect.

We can also look at the top M sequences under each sub-sample-based emulator and compare with the M sequences under the emulator based on the full training sample. These are shown in Tables 2 to 5.

Again, as expected there is a good correspondence between the M sequences under the different sub-samples and the full training sample, with tasks 8, 6 4 and 3 being proposed early in the sequence and task 5 typically late in the

1	6	8	4	1	3	2	7	5	9
2	6	8	4	1	2	3	7	5	9
3	6	8	4	1	7	3	2	5	9
4	6	8	4	1	5	3	2	7	9
5	6	8	4	3	1	2	7	5	9
6	6	8	4	1	9	3	2	7	5
7	6	8	4	3	2	1	7	5	9
8	6	8	4	3	7	1	2	5	9
9	6	8	4	1	3	2	5	7	9
10	6	8	4	3	5	1	2	7	9
11	6	8	4	2	1	3	7	5	9
12	6	8	4	1	2	3	5	7	9
13	6	8	1	4	3	2	7	5	9
14	8	6	4	1	3	2	7	5	9
15	6	8	4	2	3	1	7	5	9
16	6	8	1	4	2	3	7	5	9
17	6	8	1	4	7	3	2	5	9
18	6	8	4	1	5	3	7	2	9
19	8	6	4	1	2	3	7	5	9
20	8	6	4	1	7	3	2	5	9
21	6	8	4	2	7	1	3	5	9
22	6	8	4	2	5	1	3	7	9
23	6	8	1	4	5	3	2	7	9
24	6	8	4	7	1	3	2	5	9
25	6	8	4	1	9	3	2	5	7
26	8	6	4	1	5	3	2	7	9
27	6	8	4	3	9	1	2	7	5
28	6	8	4	3	1	2	5	7	9
29	6	8	4	7	3	1	2	5	9
30	6	8	4	1	9	3	7	2	5
31	6	8	4	7	2	1	3	5	9
32	8	6	4	3	1	2	7	5	9
33	6	8	4	3	2	1	5	7	9
34	6	8	4	2	9	1	3	7	5
35	6	8	1	4	9	3	2	7	5
36	6	8	1	3	4	2	7	5	9
37	8	6	4	1	9	3	2	7	5
38	6	8	4	1	3	2	7	9	5
39	6	8	4	7	5	1	3	2	9
40	6	8	4	3	5	1	7	2	9

Table 2: Top $M = 40$ sequences under the emulator based on the first subsample of 20 training sequences (corresponding to the black dots in Figure 1).

1	8	6	4	3	9	1	7	2	5
2	8	6	4	3	9	7	1	2	5
3	8	6	4	3	9	2	7	1	5
4	8	6	4	3	9	5	7	1	2
5	8	6	4	3	9	2	1	7	5
6	8	6	4	3	9	5	1	7	2
7	8	6	4	3	9	7	2	1	5
8	8	6	4	3	9	1	2	7	5
9	8	6	4	3	9	5	2	7	1
10	8	6	4	3	9	7	5	1	2
11	8	6	4	3	9	1	5	7	2
12	8	6	4	3	9	2	5	7	1
13	8	6	4	3	9	5	2	1	7
14	8	6	4	3	9	2	5	1	7
15	8	6	3	4	9	1	7	2	5
16	8	6	3	4	9	7	1	2	5
17	8	6	3	4	9	2	7	1	5
18	8	6	3	4	9	5	7	1	2
19	8	6	3	4	9	2	1	7	5
20	8	6	3	4	9	5	1	7	2
21	8	6	3	4	9	7	2	1	5
22	8	6	3	4	9	1	2	7	5
23	8	6	3	4	9	5	2	7	1
24	8	6	3	4	9	7	5	1	2
25	8	6	3	4	9	1	5	7	2
26	8	6	3	4	9	2	5	7	1
27	8	6	3	4	9	5	2	1	7
28	8	6	3	4	9	2	5	1	7
29	8	6	4	3	7	1	9	2	5
30	8	6	4	3	7	9	1	2	5
31	8	6	4	3	7	2	9	1	5
32	8	6	4	3	7	5	9	1	2
33	8	6	4	3	7	2	1	9	5
34	8	6	4	3	7	5	1	9	2
35	8	6	4	3	7	9	2	1	5
36	8	6	4	3	7	1	2	9	5
37	8	6	4	3	7	5	2	9	1
38	8	6	4	3	7	9	5	1	2
39	8	6	4	3	7	1	5	9	2
40	8	6	4	3	7	2	5	9	1

Table 3: Top $M = 40$ sequences under the emulator based on the second subsample of 20 training sequences (corresponding to the red dots in Figure 1.)

1	8	6	4	3	7	9	1	2	5
2	8	6	4	3	7	2	1	9	5
3	8	6	4	3	2	9	1	7	5
4	8	6	4	3	2	7	1	9	5
5	8	6	4	3	7	1	9	2	5
6	6	8	4	3	7	9	1	2	5
7	6	8	4	3	7	2	1	9	5
8	8	6	4	3	7	2	9	1	5
9	8	6	4	3	2	1	9	7	5
10	8	6	4	3	7	1	2	9	5
11	8	6	4	3	7	9	5	2	1
12	6	8	4	3	2	9	1	7	5
13	6	8	4	3	2	7	1	9	5
14	8	6	4	3	7	2	5	9	1
15	8	6	4	3	7	9	2	1	5
16	8	6	4	3	2	7	9	1	5
17	8	6	4	3	2	9	5	7	1
18	8	6	4	3	2	1	7	9	5
19	8	6	4	3	2	7	5	9	1
20	8	6	4	3	9	2	1	7	5
21	8	6	4	3	9	7	1	2	5
22	8	6	4	3	2	9	7	1	5
23	6	8	4	3	7	1	9	2	5
24	6	8	4	3	7	2	9	1	5
25	6	8	4	3	2	1	9	7	5
26	6	8	4	3	7	1	2	9	5
27	6	8	4	3	7	9	5	2	1
28	6	8	4	3	7	2	5	9	1
29	6	8	4	3	7	9	2	1	5
30	6	8	4	3	2	7	9	1	5
31	6	8	4	3	2	9	5	7	1
32	6	8	4	3	2	1	7	9	5
33	6	8	4	3	2	7	5	9	1
34	6	8	4	3	9	2	1	7	5
35	6	8	4	3	9	7	1	2	5
36	6	8	4	3	2	9	7	1	5
37	8	6	4	3	9	1	2	7	5
38	8	6	4	3	9	1	7	2	5
39	8	6	4	3	9	2	5	7	1
40	8	6	4	3	9	7	5	2	1

Table 4: Top $M = 40$ sequences under the emulator based on the third subsample of 20 training sequences (corresponding to the green dots in Figure 1.)

1	8	6	4	3	1	7	2	9	5
2	8	6	4	3	1	7	9	2	5
3	8	6	4	3	7	1	2	9	5
4	8	6	4	3	7	1	9	2	5
5	8	6	4	3	1	9	2	7	5
6	8	6	4	3	1	9	7	2	5
7	8	6	4	3	7	9	2	1	5
8	8	6	4	3	7	9	1	2	5
9	6	8	4	3	1	7	2	9	5
10	8	6	4	3	1	2	9	7	5
11	6	8	4	3	1	7	9	2	5
12	8	6	4	3	1	2	7	9	5
13	6	8	4	3	7	1	2	9	5
14	6	8	4	3	7	1	9	2	5
15	8	6	4	3	7	2	9	1	5
16	6	8	4	3	1	9	2	7	5
17	8	6	4	3	7	2	1	9	5
18	6	8	4	3	1	9	7	2	5
19	6	8	4	3	7	9	2	1	5
20	6	8	4	3	7	9	1	2	5
21	8	6	4	3	9	1	2	7	5
22	6	8	4	3	1	2	9	7	5
23	8	6	4	3	9	7	2	1	5
24	8	6	4	3	1	7	5	9	2
25	6	8	4	3	1	2	7	9	5
26	8	6	4	3	9	1	7	2	5
27	8	6	4	3	7	1	5	9	2
28	6	8	4	3	7	2	9	1	5
29	8	6	4	3	9	7	1	2	5
30	6	8	4	3	7	2	1	9	5
31	8	6	4	3	1	9	5	7	2
32	8	6	4	3	1	7	5	2	9
33	8	4	6	3	1	7	2	9	5
34	8	6	4	3	7	9	5	1	2
35	8	6	4	3	7	1	5	2	9
36	8	4	6	3	1	7	9	2	5
37	8	4	6	3	7	1	2	9	5
38	8	4	6	3	7	1	9	2	5
39	6	8	4	3	9	1	2	7	5
40	8	6	4	3	1	7	9	5	2

Table 5: Top $M = 40$ sequences under the emulator based on the full set of $N = 60$ training sequences (corresponding to the grey circles in Figure 1.)

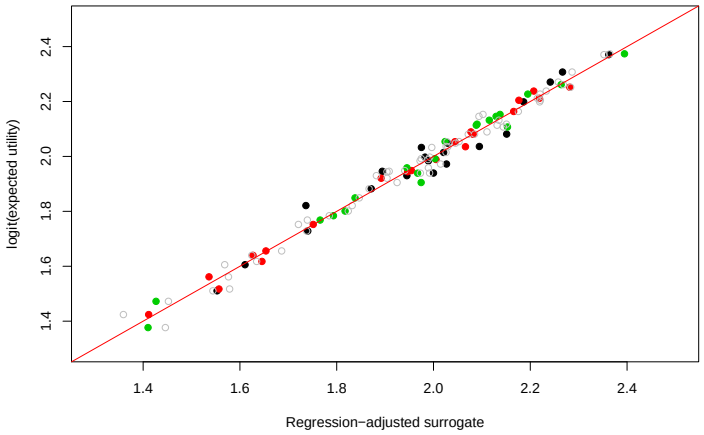

Figure 1: Logit transformed expected utilities versus regression-adjusted surrogate values based on the full training sample (grey circles) and on three subsamples of size 20 (black, red and green dots). The red line as zero intercept and unit gradient.

sequence. It can be seen that there is more variability in the M sequences from the first sub-sample (Table 2) and these seem less similar to the M sequences from the full training sample than those from the other two subsamples.

Whilst sub-sampling like this may seem like it has the potential to provide information about the stability of the emulator and of the putative optimal sequences it is not clear how to leverage this information in a real example. We would argue that we have not gained very much insight in the above illustrative example by adopting this approach. There are still no guarantees regarding the optimality of the putative optimal sequence.

Furthermore, it is difficult to conceive of a situation in which the sub-samples would not lead to an emulator that is worse than that based on the full training sample because they are based on a smaller sample, and smaller samples will naturally lead to poorer emulators. Therefore, with a given budget B, we do not see any strong arguments in favour of splitting the training sample.

6 Guidance on choice of N and M: simulation study

An extensive simulation study was carried out to explore the effect of the choice of N (for a given budget B) on finding the optimal sequence over a range of values for the model parameters $J, I, R_0, \epsilon, \lambda, p, q$ and so on.

A total of 10000 "problem scenarios" were generated randomly from the

following distributions:

$$\begin{aligned}
J &\sim U[6,10] \\
I &\sim U[5,15] \\
R_0 &\sim U(0.7,0.95) \\
\epsilon_i = \epsilon &\sim \text{Gamma}(10,500), \qquad i = 1,\ldots,I \\
\lambda_i &\sim U(0,0.5), \qquad i = 1,\ldots,I \\
p_{i,j} &\sim U(0,0.5)\mathbb{I}(U(0,1) > 0.5), \qquad i = 1,\ldots,I, \quad j = 1,\ldots,J,
\end{aligned}$$

with (q_1, q_2, q_3) sampled uniformly from the discrete distribution with sample space $\{(1/4,1/4,1/2),(1/4,1/3,5/12),(1/4,1/2,1/4),(1/4,2/3,1/12)$, $(1/4,3/4,0),(1/3,1/4,5/12),(1/3,1/3,1/3),(1/3,1/2,1/6)$, $(1/3,2/3,0),(1/2,1/4,1/4),(1/2,1/3,1/6),(1/2,1/2,0)$, $(2/3,1/4,1/12),(2/3,1/3,0),(3/4,1/4,0)\}$. In the above, and what follows, $U(a,b)$ denotes the continuous uniform distribution with sample space given by the open interval (a,b), and $U[c,d]$ denotes the discrete uniform distribution with sample space $\{c, c+1, \ldots, d\}$.

These distributions reflect the range of values that one might expect for problems of this type. We restricted attention to a maximum of 10 tasks to allow complete enumeration of all possible sequences in a reasonable time.

For each scenario $k = 1,\ldots,10000$ we sampled a value of N, $N_k \sim U[25,200]$ and a value of M, $M_k \sim U[10,200]$ and performed the emulation as described in the main paper, using the Benter-Pearson surrogate-correlation combination. For each scenario k we recorded whether the optimal sequence was found in the $B_k = N_k + M_k$ evaluations; giving $Y_k = 1$ if the optimal sequence was found, and $Y_k = 0$ otherwise.

We then fitted a series of logistic regression models to the binary response variable Y_k (for $k = 1,\ldots,10000$) which is the indicator of whether the optimal sequence was found or not. The explanatory variables were the values of N_k, B_k, and functions thereof. The models were fitted via maximum likelihood by using the `glm()` function in R. Using the Akaike Information Criterion (AIC) to compare the fit of different models we found that the model with

$$\ell(B,N) = \text{logit}(E[Y]) = \hat{\beta}_0 + \hat{\beta}_1 N + \hat{\beta}_2 N^2 + \hat{\beta}_3 N^2/B^2$$

gave the optimal fit (an AIC of 10035). So, given values for B and N we can use the above expression to predict the probability that we will find the optimal sequence (in generic problems of this type). Treating the logit-transformed success probability $\ell(B,N)$ as a function of N for fixed B, we can maximise $\ell(B,N)$ with respect to N to find an expression for the optimal N. Differentiating $\ell(B,N)$ with respect to N and setting the derivative equal to zero gives

$$N = \frac{-\hat{\beta}_1}{2(\hat{\beta}_2 + \hat{\beta}_3/B^2)},$$

which can be shown to be a maximum in the usual way. Therefore, for a given budget B, our simulations would suggest that the optimal choice of N is given

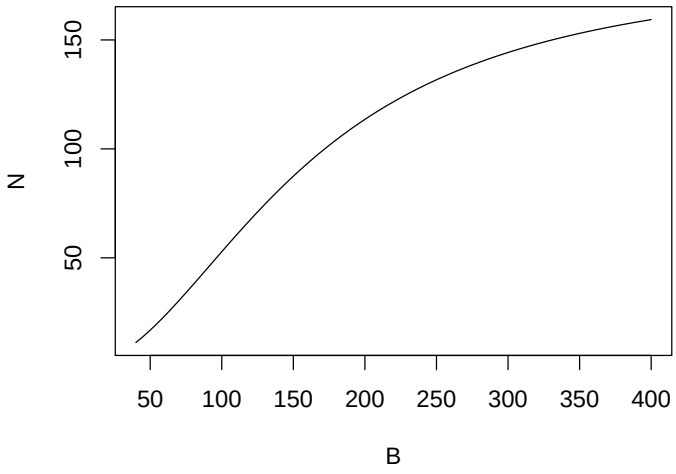

Figure 2: Optimal choice of N for a given budget B based on logistic regression analysis of simulation results

by the above expression. Rounding the estimated logistic regression coefficients to 4 decimal places for clarity we have

$$\ell(B, N) = \text{logit}(E[Y]) = 0.3884 + 0.0199N - 0.0001N^2 - 1.3429N^2/B^2, \quad (2)$$

and therefore,

$$N = \frac{0.0199}{2(0.0001 + 1.3429/B^2)}. \quad (3)$$

So, for example, when $B = 100$ we get $N \simeq 53$ (to the nearest integer) using the unrounded values for the logistic regression coefficients. The estimated relationship between the "optimal" N and B is displayed in Figure 2.

Plugging the optimal N, as determined through Equation (3), into the logistic regression equation (2) for a range of values of B gives estimates for the probabilities of finding the optimal sequence in the B evaluations. These probabilities are displayed in the solid black line in Figure 3. Also displayed in Figure 3 by the red dashed line are the estimated probabilities of finding the optimal sequence when we simply choose $N = B/2$ (rounded to the nearest integer), in other words we choose $N = M$; training samples and evaluation samples of equal sizes. This clearly indicates that choosing $N = M$ is only marginally sub-optimal. Therefore due to its simplicity our recommendation to practitioners, based on our simulations, would be that choosing an even split of the budget B between N and M is a reasonable default choice.

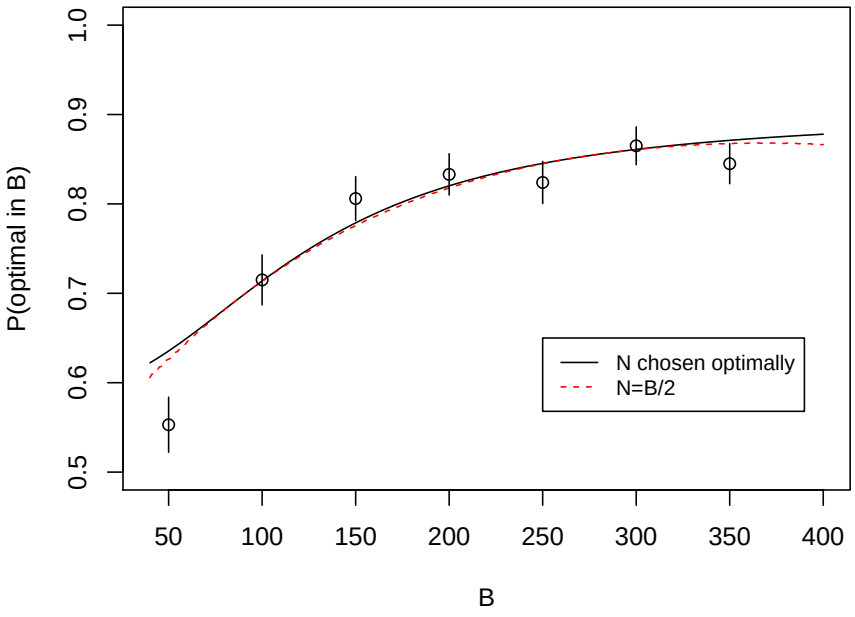

Figure 3: Estimated probability of finding the optimal sequence in the B evaluations, using the "optimal" choice of N (black solid line), and using $N = B/2$ (red dashed line). The circles represent the observed proportion of successful optimal sequence discoveries in 1000 independent simulations, for a range of values of B. The vertical lines represent the usual large sample 95% confidence intervals for a binomial proportion.

To validate whether the logistic regression model (2) provides a good fit to the simulated data, and hence whether our simple guidance on the choice of N will be of practical relevance, we performed a validation simulation study. For each value of B in $\{50, 100, 150, 200, 250, 300, 350\}$ we generated 1000 "problem scenarios" randomly using the procedure detailed above. For each scenario we performed the usual emulator fitting/evaluation process on a simple random sample of sequences of size $N = B/2$. We recorded the proportion of the scenarios in which the optimal sequence was found. These empirical proportions are unbiased estimates of the true probabilities that the optimal sequence would be found in the B evaluations and are represented by the circles in Figure 3; the vertical lines give approximate 95% confidence intervals for the true probabilities. Our estimates of the probabilities from the independent validation simulations match up well with those predicted from our logistic regression model, apart from when $B = 50$ where we see that the logistic regression model overestimates the probability slightly. Overall, these simulations would suggest that the logistic regression model is sufficiently accurate at predicting the probability of successfully finding the optimal sequence, over a wide range of plausible scenarios. They also suggest that choosing $N = B/2$ seems a sensible, simple default option in the absence of other specific information.

7 Cross entropy optimization-based training sample simulations

As an alternative to choosing the training sample via a random sample, we have investigated the following simple alternative. For a training sample of size N we take a random sample of size $\lfloor N/2 \rfloor$ from the set of all $J!$ sequences, then we fit the emulator based on maximising the correlation function between emulator output and actual expected utility for these training sequences. We then sample the remaining $N - \lfloor N/2 \rfloor$ training samples from the probabilistic model associated with the emulator (Benter, PL or RPL) at the optimized parameter values. The full set of $\lfloor N/2 \rfloor$ original and $N - \lfloor N/2 \rfloor$ new sequences make up our full training sample. We then proceed to fit the emulator as described in the main paper. As such this training sample is based on two-stages of a cross-entropy optimization (CEO) algorithm [Rubinstein and Kroese, 2004]. We present results in the following table for the same problem set-up as in Section 5.1 of the main paper.

			M				
N	Correlation	Model	10	20	50	100	200
100	Pearson	PL	0.21	0.35	0.52	0.75	0.84
		RPL	0.42	0.59	0.74	0.92	0.96
		B	0.58	0.75	0.86	0.94	0.97
	Spearman	PL	0.16	0.20	0.29	0.40	0.60
		RPL	0.20	0.29	0.50	0.65	0.73
		B	0.27	0.41	0.50	0.64	0.79
	Kendall	PL	0.11	0.18	0.28	0.39	0.56
		RPL	0.23	0.34	0.46	0.61	0.74
		B	0.17	0.26	0.43	0.57	0.75
200	Pearson	PL	0.34	0.56	0.76	0.90	0.98
		RPL	0.63	0.75	0.91	0.99	1.00
		B	0.66	0.80	0.93	0.98	1.00
	Spearman	PL	0.14	0.23	0.41	0.69	0.85
		RPL	0.40	0.58	0.80	0.90	0.98
		B	0.54	0.67	0.83	0.93	0.95
	Kendall	PL	0.22	0.35	0.64	0.82	0.89
		RPL	0.33	0.54	0.73	0.85	0.94
		B	0.50	0.69	0.84	0.91	0.96

Table 6: Estimated probability from 100 simulations that the optimal sequence is found in an initial sample of N sequences chosen via two steps of a cross-entropy optimization algorithm followed by the top M sequences under various surrogates: PL=Plackett-Luce, RPL=Reverse Plackett-Luce, B=Benter.