Online Appendix: Improving the Efficiency and Efficacy of Controlled Sequential Bifurcation for Simulation Factor Screening

A. Appendix: Proofs

A.1. Proof of Lemma 1

For a group $\{k_1, k_1+1, \ldots, k_2\}$ with group effect $\zeta = \sum_{k_1+1}^{k_2} \beta_i$, an arbitrary sample path is $SP_{r_0}(r) = \sum_{\ell=1}^{r}(D_\ell(k_1,k_2) - r_0)$, $r = N_0, N_0+1, \ldots$. If $r_0' < r_0$, then for every sample path, $SP_{r_0'}(r) > SP_{r_0}(r)$ for all r. Thus, every sample path that is classified as unimportant (exits down) under r_0' will also be classified as unimportant under r_0. However, a path that is classified as important (exits up) under r_0', may or may not be classified as important under r_0. Therefore, the probability of declaring a group important decreases in r_0, whereas the probability of declaring it unimportant increases in r_0. □

A.2. Proof of Lemma 2

Suppose the group effect is ζ and the stopping time of the test is T.

$$\begin{aligned}
&\Pr\{\text{Declare group } i \text{ as important} |\, \zeta \le \Delta_0,\, r_0 = \Delta_0\} \\
&= \Pr\left\{ \sum_{\ell=1}^{T} D_\ell(k_1,k_2) - \Delta_0 \ge a(k_1,k_2) - T\lambda \,\middle|\, \zeta \le \Delta_0 \right\} \\
&\le \Pr\left\{ \sum_{\ell=1}^{T} (D_\ell(k_1,k_2) - \Delta_0) \ge a(k_1,k_2) - T\lambda \,\middle|\, \zeta = \Delta_0 \right\} \\
&= 1/2.
\end{aligned}$$

The last equality holds because $\mathrm{E}[\sum_{\ell=1}^{T}(D_\ell(k_1,k_2) - \Delta_0) \mid \zeta = \Delta_0] = 0$; and the region $[\,-a(k_1,k_2) + r\lambda, a(k_1,k_2) - r\lambda\,]$ is symmetric.

Similarly,

$$\begin{aligned}
&\Pr\{\text{Declare group } i \text{ as important} \mid \zeta \ge \Delta_1,\, r_0 = \Delta_1\} \\
&= \Pr\left\{ \sum_{\ell=1}^{T} D_\ell(k_1,k_2) - \Delta_1 \ge a(k_1,k_2) - T\lambda \,\middle|\, \zeta \ge \Delta_1 \right\} \\
&\ge \Pr\left\{ \sum_{\ell=1}^{T} D_\ell(k_1,k_2) - \Delta_1 \ge a(k_1,k_2) - T\lambda \,\middle|\, \zeta = \Delta_1 \right\} \\
&= 1/2.
\end{aligned}$$

Combining this with Lemma 1, we conclude that $\Delta_0 \le r_0 \le \Delta_1$.

A.3. Proof of Theorem 5

To prove the theorem, we need the following lemma:

Lemma 3. *(Hartmann 1991) Let $X_1, X_2, \ldots$ be independent and identically distributed* $\mathrm{Nor}(\Delta, 1)$ *random variables with $\Delta > 0$. Let $S(n) = \sum_{j=1}^{n} X_j$, $L(n) = -a + \lambda n$, and $U(n) = a - \lambda n$ for some $a > 0$ and $\lambda \ge 0$. Let $R(n)$ denote the interval $[L(n), U(n)]$ ($R(n) = \emptyset$ when $L(n) > U(n)$), and let $T = \min\{n : S(n) \notin R(n)\}$ be the first time the partial sum $S(n)$ does not fall in the triangular region defined by $R(n)$. Let $\mathcal{E}$ be the event $S(T) \le L(T)$ and $R(T) \neq \emptyset$, or $S(T) \le 0$ and $R(T) = \emptyset$. Then*

$$\Pr\{\mathcal{E}\} \le \int_{-\infty}^{\infty} \frac{e^{-2\lambda\xi}}{(1+e^{-2\lambda\xi})} \phi\left(\frac{\xi - \Delta a/\lambda}{\sqrt{a/\lambda}}\right) \frac{\mathrm{d}\xi}{\sqrt{a/\lambda}}.$$

The proof of the theorem is a direct application of Lemma 3. Let $\sigma^2 = \mathrm{Var}\left[D_\ell(k_1, k_2)\right]$. When the group effect satisfies $\zeta \le \Delta_0$,

$$\begin{aligned}
&\Pr\left\{\sum_{\ell=1}^{T}(D_\ell(k_1,k_2) - r_0) \le \min\{0, -a(k_1,k_2) + T\lambda\}\right\} \\
&= 1 - \Pr\left\{\sum_{\ell=1}^{T}(r_0 - D_\ell(k_1,k_2)) \le \min\left\{0, -a(k_1,k_2) + T\lambda\right\}\right\} \\
&= 1 - \Pr\left\{\sum_{\ell=1}^{T}\frac{(r_0 - D_\ell(k_1,k_2))}{\sigma} \le \min\left\{0, \frac{-a(k_1,k_2)}{\sigma} + T\frac{\lambda}{\sigma}\right\}\right\} \\
&= 1 - \mathrm{E}\left[\Pr\left\{\sum_{\ell=1}^{T}\frac{(r_0 - D_\ell(k_1,k_2))}{\sigma} \le \min\left\{0, \frac{-a(k_1,k_2)}{\sigma} + T\frac{\lambda}{\sigma}\right\}\middle| S^2(k_1,k_2)\right\}\right] \\
&\ge 1 - \mathrm{E}\left[\int_{-\infty}^{\infty}\frac{e^{-2\frac{\lambda}{\sigma}\xi}}{1+e^{-2\frac{\lambda}{\sigma}\xi}}\phi\left(\frac{\xi - \frac{(r_0-\zeta)}{\sigma}a/\lambda}{\sqrt{a/\lambda}}\right)\frac{d\xi}{\sqrt{a/\lambda}}\right] \text{ by Lemma 3} \\
&\ge 1 - \mathrm{E}\left[\int_{-\infty}^{\infty}\frac{e^{-2\frac{\lambda}{\sigma}\xi}}{1+e^{-2\frac{\lambda}{\sigma}\xi}}\phi\left(\frac{\xi - \frac{(r_0-\Delta_0)}{\sigma}a/\lambda}{\sqrt{a/\lambda}}\right)\frac{d\xi}{\sqrt{a/\lambda}}\right] \text{ since } \Delta_0 \ge \zeta \\
&= 1 - \mathrm{E}\left[\int_{-\infty}^{\infty}\frac{e^{-2\lambda y}}{1+e^{-2\lambda y}}\phi\left(\frac{y\sigma}{\sqrt{a/\lambda}} - \sqrt{a/\lambda}\frac{(r_0-\Delta_0)}{\sigma}\right)\frac{\sigma dy}{\sqrt{a/\lambda}}\right] \\
&= 1 - \mathrm{E}\left[\int_{-\infty}^{\infty}\frac{e^{-2\lambda y}}{1+e^{-2\lambda y}}\phi\left(\frac{y\sigma}{S(k_1,k_2)\sqrt{a_0/\lambda}} - \frac{\sqrt{a_0/\lambda}S(k_1,k_2)(r_0-\Delta_0)}{\sigma}\right)\frac{\sigma dy}{S(k_1,k_2)\sqrt{a/\lambda}}\right] \\
&= 1 - \int_{0}^{\infty}\int_{-\infty}^{\infty}\theta(x)\frac{\nu(y)}{1-\nu(y)}\phi\left(y\theta(x) - \frac{(r_0-\Delta_0)}{\theta(x)}\right) f(x, N_0 - 1)\,\mathrm{d}y\mathrm{d}x \qquad (1) \\
&= 1 - \alpha.
\end{aligned}$$

Since $\theta(x) \to \infty$ when $x \to 0$, Equation (1) relies on the fact that

$$\lim_{x\to 0} \theta(x) \frac{\nu(y)}{1-\nu(y)} \phi\left(y\theta(x) - \frac{(r_0 - \Delta_0)}{\theta(x)}\right) f(x, N_0 - 1) = 0.$$

This is because

$$\lim_{x\to 0} \theta(x) \frac{\nu(y)}{1-\nu(y)} \phi\left(y\theta(x) - \frac{(r_0 - \Delta_0)}{\theta(x)}\right) f(x, N_0 - 1) = \lim_{x\to 0} c_1 x^{(N_0-4)/2} e^{\frac{-c_2}{x}},$$

where c_1 and c_2 are constants. When $N_0 \geq 4$, the limit clearly equals 0. When $N_0 = 2$ or 3, using L'Hopital's rule, it is easy to show that the limit still equals 0. The same result is also used in the following proof.

Assume the group effect satisfies $\zeta \geq \Delta_1$,

$$\begin{aligned}
&\Pr\left\{\sum_{\ell=1}^{T}(D_\ell(k_1,k_2) - r_0) \leq \min\{0, -a(k_1,k_2) + T\lambda\}\right\} \\
&= \Pr\left\{\sum_{\ell=1}^{T}(D_\ell(k_1,k_2) - r_0) \leq \min\{0, -a(k_1,k_2) + T\lambda\}\right\} \\
&= \Pr\left\{\sum_{\ell=1}^{T}\frac{(D_\ell(k_1,k_2) - r_0)}{\sigma} \leq \min\left\{0, \frac{-a(k_1,k_2)}{\sigma} + T\frac{\lambda}{\sigma}\right\}\right\} \\
&= \mathrm{E}\left[\Pr\left\{\sum_{\ell=1}^{T}\frac{(D_\ell(k_1,k_2) - r_0)}{\sigma} \leq \min\left\{0, \frac{-a(k_1,k_2)}{\sigma} + T\frac{\lambda}{\sigma}\right\}\middle| S^2(k_1,k_2)\right\}\right] \\
&\leq \mathrm{E}\left[\int_{-\infty}^{\infty}\frac{e^{-2\frac{\lambda}{\sigma}\xi}}{1+e^{-2\frac{\lambda}{\sigma}\xi}}\phi\left(\frac{\xi - \frac{(\zeta - r_0)}{\sigma}a/\lambda}{\sqrt{a/\lambda}}\right)\frac{d\xi}{\sqrt{a/\lambda}}\right] \text{ by Lemma 3} \\
&\leq \mathrm{E}\left[\int_{-\infty}^{\infty}\frac{e^{-2\frac{\lambda}{\sigma}\xi}}{1+e^{-2\frac{\lambda}{\sigma}\xi}}\phi\left(\frac{\xi - \frac{(\Delta_1 - r_0)}{\sigma}a/\lambda}{\sqrt{a/\lambda}}\right)\frac{d\xi}{\sqrt{a/\lambda}}\right] \text{ since } \Delta_1 \leq \zeta \\
&= \mathrm{E}\left[\int_{-\infty}^{\infty}\frac{e^{-2\lambda y}}{1+e^{-2\lambda y}}\phi\left(\frac{y\sigma}{\sqrt{a/\lambda}} - \sqrt{a/\lambda}\frac{(\Delta_1 - r_0)}{\sigma}\right)\frac{\sigma dy}{\sqrt{a/\lambda}}\right] \\
&= \mathrm{E}\left[\int_{-\infty}^{\infty}\frac{e^{-2\lambda y}}{1+e^{-2\lambda y}}\phi\left(\frac{y\sigma}{S(k_1,k_2)\sqrt{a_0/\lambda}} - \frac{\sqrt{a_0/\lambda}S(k_1,k_2)(\Delta_1 - r_0)}{\sigma}\right)\frac{\sigma dy}{S(k_1,k_2)\sqrt{a/\lambda}}\right] \\
&= \int_0^{\infty}\int_{-\infty}^{\infty}\theta(x)\frac{\nu(y)}{1-\nu(y)}\phi\left(y\theta(x) - \frac{(\Delta_1 - r_0)}{\theta(x)}\right) f(x, N_0 - 1)\,\mathrm{d}y\mathrm{d}x \\
&= 1 - \gamma.
\end{aligned}$$

□

B. Analysis of the Efficiency of CSB-X

Because of the introduction of mirror levels, CSB-X doubles the number of design points relative to CSB. However, unlike the traditional fold-over design, the overall computational effort of CSB-X is not necessarily doubled. For any specific group $\{k_1+1, k_1+2, \ldots, k_2\}$ tested, define $D_{\text{CSB}}(k_1,k_2) = Z(k_2) - Z(k_1)$, and $D_{\text{CSB-X}}\,(k_1,k_2) = (Z(k_2) - Z(-k_2))/2 - (Z(k_1) - Z(-k_1))/2$ for CSB and CSB-X, respectively. Then

$$\text{Var}\left[D_{\text{CSB}}(k_1,k_2)\right] = \text{Var}(Z(k_2)) + \text{Var}(Z(k_1)) - 2\text{Cov}(Z(k_1), Z(k_2)),$$

while

$$\begin{aligned}
\text{Var}\left[D_{\text{CSB-X}}(k_1,k_2)\right] &= 1/4\left\{\text{Var}(Z(k_2)) + \text{Var}(Z(k_1)) + \text{Var}(Z(-k_2)) + \text{Var}(Z(-k_1))\right.\\
&- 2\text{Cov}(Z(k_1), Z(k_2)) - 2\text{Cov}(Z(-k_1), Z(-k_2))\\
&- 2\text{Cov}(Z(k_2), Z(-k_2)) - 2\text{Cov}(Z(k_1), Z(-k_1))\\
&+ \left. 2\text{Cov}(Z(-k_1), Z(k_2)) + 2\text{Cov}(Z(k_1), Z(-k_2))\right\}.
\end{aligned}$$

If $\text{Var}(D_{\text{CSB}}(k_1,k_2)) \geq 2\text{Var}(D_{\text{CSB-X}}(k_1,k_2))$, then the expected number of runs required in each **Test** step for the fold-over design may be smaller than the original design for both the two-stage and fully sequential testing procedures in Wan et al. (2006) and any other test for which the sample size is proportional to the variance of the tested variables. In other words, the expected value of the number of runs in the worst-case scenario for each test is not increased when we use the fold-over design. The intuition is that the fold-over design decreases the variance of the test statistic, so even though the number of design points is doubled, the number of replications taken at each design point decreases. The amount of the decrease depends on the variance and covariance structure between different levels. Here are two special cases:

If the responses at all levels are independent of each other, then

$$\text{Var}\left[D_{\text{CSB}}(k_1,k_2)\right] = \text{Var}(Z(k_2)) + \text{Var}(Z(k_1)),$$

and

$$\text{Var}\left[D_{\text{CSB-X}}(k_1,k_2)\right] = 1/4\left\{\text{Var}(Z(k_2)) + \text{Var}(Z(k_1)) + \text{Var}(Z(-k_2)) + \text{Var}(Z(-k_1))\right\}.$$

If the responses from different levels have the same variance, then the variances of the two test statistics are the same, so CSB-X should be as efficient as CSB at each step.

If we implement CRN across all positive levels and all negative levels separately, leaving the responses between positive levels and negative levels independent, then when CRN is effective,

$$\begin{aligned}\mathrm{Var}[D_{\mathrm{CSB}}(k_1,k_2)] &= \mathrm{Var}(Z(k_2)) + \mathrm{Var}(Z(k_1)) - 2\mathrm{Cov}(Z(k_1),Z(k_2)) \\ &< \mathrm{Var}(Z(k_2)) + \mathrm{Var}(Z(k_1)),\end{aligned}$$

and

$$\begin{aligned}\mathrm{Var}[D_{\mathrm{CSB\text{-}X}}(k_1,k_2)] &= 1/4\left\{\mathrm{Var}(Z(k_2)) + \mathrm{Var}(Z(k_1)) + \mathrm{Var}(Z(-k_2)) + \mathrm{Var}(Z(-k_1))\right. \\ &- \left. 2\mathrm{Cov}(Z(k_1),Z(k_2)) - 2\mathrm{Cov}(Z(-k_1),Z(-k_2))\ \right\} \\ &< 1/4\left\{\mathrm{Var}(Z(k_2)) + \mathrm{Var}(Z(k_1)) + \mathrm{Var}(Z(-k_2)) + \mathrm{Var}(Z(-k_1))\right\}.\end{aligned}$$

In this case, both CSB and CSB-X work more efficiently with CRN than without CRN. If $\mathrm{Cov}(Z(k_1),Z(k_2)) = \mathrm{Cov}(Z(-k_1),Z(-k_2))$ and $\mathrm{Var}(Z(k_1)) + \mathrm{Var}(Z(k_2)) = \mathrm{Var}(Z(-k_1)) + \mathrm{Var}(Z(-k_2))$, then CSB-X is as efficient as CSB in each **Test** step.

Other strategies can be analyzed similarly; they may increase or decrease the efficiency of CSB and CSB-X, depending on the specific covariance structure. But our overall conclusion is that we expect CSB-X to be roughly as efficient, and sometimes more efficient, than CSB while also correctly identifying important factors more effectively than CSB.

C. MatLab Code to Find a_0 and r_0

```
%main File
%the following values should be set by the user
global lambda; %The slope
global alpha; %required Type I Error
global gamma; %Required power
global n0; %initial sample size
global delta1; %threshold of critical effects
global delta0; %threshold of important effects
alpha = 0.30;
gamma = 0.95;
delta0=2;
delta1=4;
lambda=(delta1-delta0)/4
n0=25;
if(abs(alpha-(1-gamma))<1e-8) %symmetric case
    angle =(alpha +1-gamma)*0.5
```

```
    psi=2*log(2*angle)/(1-n0)
    ata = (exp(psi)-1)/2;
    a0 = 2*ata*(n0-1)/(delta1-delta0)
    r0=(delta0+delta1)/2
    fprintf(1, 'Symmetric ranking and Selection Procedure, Solve directly\n');
else
    par=SolveaK(1e-8);
    a0 = par(1)
    r0=par(2)
end
******************************************************************************
function result = SolveaK(error)
global lambda;
global alpha;
global gamma;
global n0;
global delta1;
global delta0;
step =0.01;
angle =(alpha +1-gamma)*0.5
psi=2*log(2*angle)/(1-n0)
ata =(exp(psi)-1)/2;
aInitial = 2*ata*(n0-1)/(delta1-delta0);
KInitial=(delta0+delta1)/2 e=10^-10;
findResult = 0;
options=optimset('Display','iter','TolFun',error);
t=cputime;
[x,fval]=fminsearch(@sumInt,[aInitial,KInitial], options)
FuncInt1S(x(1),x(2))
FuncInt2S(x(1),x(2)) timeUse = cputime-t
result=x
*******************************************************************
function func1 = FuncInt1S(a,K)
global lambda;
global alpha;
global gamma;
global n0;
global delta1;
global delta0;
global numberOfRuns;
 func1 = dblquad(@FuncProb1S,0.01,75, -15,15,10^-6,[],
a, K)- alpha;
**********************************************************************
function func2 = FuncInt2S(a, K)
global lambda;
global alpha;
global gamma;
global n0;
global delta1;
global delta0;
global numberOfRuns;
func2 = dblquad(@FuncProb2S,0.1,75, -15,15,10^-5,[],
a, K)- (1-gamma);
************************************************************************
```

```
function ProbDev1 = FuncProb1S(x, y, a, K)
global lambda;
global alpha;
global gamma;
global n0;
global delta1;
global delta0;
global numberOfRuns; temp1= exp(-2*lambda.*y);
temp=y.*sqrt(n0-1)./(sqrt(a./lambda).*sqrt(x)) -
sqrt(a./lambda).*(K-delta0).*sqrt(x)./sqrt(n0-1);
temp2=sqrt(n0-1)./(sqrt(a./lambda).*sqrt(x));
ProbDev1 = temp2*(temp1/(1+temp1)).*FunSiS(temp).*chi2pdf(x,n0-1);
*****************************************************************
function ProbDev2 = FuncProb2S (x, y, a, K)
global lambda;
global alpha;
global gamma;
global n0;
global delta1;
global delta0;
global numberOfRuns;
temp1= exp(-2*lambda.*y);
temp=y.*sqrt(n0-1)./(sqrt(a./lambda).*sqrt(x)) -
sqrt(a./lambda).*(delta1-K).*sqrt(x)./sqrt(n0-1);
temp2 =sqrt(n0-1)./(sqrt(a/lambda).*sqrt(x));
ProbDev2 =temp2*(temp1/(1+temp1)).*FunSiS(temp).*chi2pdf(x,n0-1);
*************************************************************
function si = FunSiS (x)
si = exp(-(x.^2)/2)/sqrt(2*3.1416);
```