Exact Bootstrap and Permutation Distribution
of Wins and Losses in a Hierarchical Trial*

William N. Anderson† and Johan Verbeeck‡

November 24, 2019

Abstract

Finkelstein-Schoenfeld, Buyse, Pocock, and other authors have
developed generalizations of the Mann-Whitney test that allow for
pairwise patient comparisons to include a hierarchy of measurements.
Various authors present either asymptotic or randomized methods for
analyzing the wins. We use graph theory concepts to derive exact
means and variances for the number of wins, as a replacement for
approximate values obtained from bootstrap analysis or random sam-
ping from the permutation distribution. The time complexity of our
algorithm is $O(N^2)$, where $N$ is the total number of patients. In any
situation where the mean and variance of a bootstrap sample are used
to draw conclusions, our methodology will be faster and more accurate
than the randomized bootstrap or permutation test.

KEYWORDS: Pairwise Comparisons, Combining Endpoints, Win Ratio,
Bootstrap, Permutation Test

2010 MATHEMATICS SUBJECT CLASSIFICATIONS: 62G09, 62N99

*This is a revision of an earlier preprint dated January 28, 2019. The differences are
mainly in organization; the results remain the same.
†Carpinteria, California, USA. Email: WNilesAnderson@gmail.com. LinkedIn:
william-anderson-46384b7
‡I-Biostat, University Hasselt, Belgium. Email: johan.verbeeck@uhasselt.be
1 Background

The context is the hierarchical clinical trial methodology of Finkelstein and Schoenfeld (1999), which is a far-reaching generalization of the classical Gehan-Wilcoxon test (Gehan, 1965). The Finkelstein-Schoenfeld test produces a \( p \)-value, based on the the permutation distribution of the trial arms. The \( p \)-value computation uses the difference between treatment wins and control wins, and does not consider the two items separately.

Other authors have considered similar analyses. References include Buyse (2010), Dong (2016), Luo et al. (2015, 2017), Bebu et al. (2016), and Pocock et al. (2012). In these papers the variances for wins and their differences are computed using either asymptotic formulas or randomized sampling. Inferences are then drawn from the computed means and variances.

We use graph theory concepts to derive the exact mean and variance of the trial arm wins for both bootstrap and permutation distribution analysis. The algorithm complexity is \( O(N^2) \) in both time and space. This methodology eliminates the need for asymptotic evaluation of variance matrices. Our algorithm will be both faster and more accurate than randomized bootstrap or permutation tests in any situation where inferences are drawn from the mean and variance of the trial arm wins. For ratios the situation is slightly more complicated. We present R\textsuperscript{®} and SAS\textsuperscript{®} code to implement the computations.

In addition to the theoretical treatment, we illustrate with a simple example. R and SAS code, and a more detailed treatment of the example, are contained in appendices.

1.1 A bit of notation

We will present expected values and variances under two different methodologies: the permutation distribution of trial arms and the bootstrap distribution with sampling from the trial arms separately. We will use the notations \( \text{Exp}_P \), \( \text{Var}_P \), \( \text{Exp}_B \), and \( \text{Var}_B \) to clarify which methodology is involved in the specific formulas.
2 The Finkelstein-Schoenfeld Test

The Finkelstein-Schoenfeld test is based on a hierarchical composition of a number of measures. For each measure every patient is compared to every other patient in a pairwise manner. We define a score, \( u_{ij} \), which is chosen to reflect whether patient \( i \) has had the more favorable outcome than patient \( j \). The concept of more favorable can involve censoring, missing data, and can include a threshold; accordingly patients who cannot be compared may not necessarily be tied. The score is

-1 if patient \( i \) has a less favorable outcome than patient \( j \);

0 if the patients cannot be compared, or are tied;

+1 if patient \( i \) has a more favorable outcome than patient \( j \).

The scores are computed separately for each measure in the hierarchy. The overall comparison for the pair is the first non-zero comparison, where first is defined by the prespecified hierarchy. It is quite possible that for many pairs of patients the overall comparison will remain 0. It is also possible to have non-transitive comparisons: that is, three patients such that \( u_{ij} = 1 \), \( u_{jk} = 1 \), and \( u_{ki} = 1 \) (Verbeeck et al., 2018).

The Finkelstein-Schoenfeld test is a score test based on the sum of the scores for the treated group. Suppose there are \( N \) subjects in the trial, with \( m \) subjects in the treatment group and \( n \) subjects in the control group. Let \( D_i = 1 \) for subjects in the treatment group, and \( D_i = 0 \) for patients in the control group. Using the \( u_{ij} \) for every pair of patients defined above, we assign a score to each subject, \( U_i = \sum_j u_{ij} \).

The test is now based upon

\[
FS = \sum_{i=1}^{N} D_i U_i. \tag{1}
\]

It it easy to see that \( \exp_P(FS) = 0 \), and it can be shown that

\[
\text{Var}_P(FS) = \frac{mn}{N(N - 1)} \sum_{i=1}^{N} U_i^2. \tag{2}
\]
The expected value and variance above furnish the needed information to compute a $z$-statistic and hence a $p$-value, under the null hypothesis of no difference between the trial arms. The formulas are derived in Finkelstein and Schoenfeld (1999).

The Partner IB and TAVR UNLOAD trials have used this methodology prospectively Leon et al. (2010), Spitzer et al. (2016). The method has also been used to furnish alternative analyses to trials with other prespecified analyses Buyse (2010), Dong et al. (2016), Finkelstein and Schoenfeld (1999), Pocock et al. (2012).

### 3 Wins and Losses

The outcome of the trial described in section 2 can be described by two items:

- An $N \times N$ outcome matrix $U$, which is skew and has entries in the set \{-1, 0, +1\}.

- A trial arm vector $D$, which has the value 1 for patients in the treatment arm and 0 for patients in the control arm. There will be $m$ patients in the treatment arm, and $n$ patients in the control arm.

One method to produce such an outcome matrix is analyzing a single variable by use of the Mann-Whitney test, or the Gehan-Wilcoxon test, which allows for censoring. The Finkelstein-Schoenfeld hierarchical analysis allows for consideration of more variables. In the remainder of this manuscript we consider that the outcome matrix $U$ has been produced by some unspecified mechanism; the only restriction is the skewness mentioned above.

We will analyze the means and variances of wins using two different models. In both cases the analysis is based on the observed $U$ matrix and the trial arm assignments.

- **Permutation distribution:** In this analysis all possible permutations of $m$ treatment assignments and $n$ control assignments are considered, and the actual assignments play no special role. This permutation analysis is the analysis used in the classical Wilcoxon test (Hollander et al., 2013), and its generalization as the Gehan-Wilcoxon test (Gehan, 1965) used to analyze a single censored variable. It is also the methodology that was used above in the Finkelstein-Schoenfeld analysis (1999).
Bootstrap: In this analysis a new dataset is obtained by random sampling, with repetition, separately from each of the observed treatment groups. For each sample the numbers of treatment and control wins are computed, and the mean and variance of these numbers over many samples are computed. These means and variances are used to draw inferences about the true numbers of wins. There are \( m^n n^n \) possible samples, each equally likely.

As an alternative to actually performing the bootstrap sampling, we compute the mean and variance over all possible bootstrap samples, using an \( O(N^2) \) algorithm to be described below. This method will be more accurate than actually performing the bootstrap, because the randomization error is eliminated. It will also be faster than evaluating a large number of bootstrap samples, because evaluating only one bootstrap sample is already \( O(N^2) \).

### 3.1 The graphical model

For narrative purposes it will prove convenient to think of the outcome matrix \( U \) as the adjacency matrix of a directed graph \( G \). Our terminology is reasonably standard, and follows Wikipedia (2019).

- If \( u_{ij} = 1 \), the edge \( e = (i, j) \) is an ordered pair of distinct vertices. The head of \( e \) is vertex \( j \) and the tail is vertex \( i \). If \( u_{ij} \neq 1 \), there is no corresponding edge. In a pictorial representation, we draw an arrow from vertex \( i \) to vertex \( j \). If vertex \( v \) is the head or tail of edge \( e \), then we say that \( e \) is adjacent to \( v \). Let \( E \) denote the total number of edges of \( G \).

- For a vertex \( v \), the indegree \( ID(v) \) is the number of edges whose head is the vertex \( v \); equivalently the number of 1 entries in the associated column of \( U \), or the number of pair comparisons that represent a loss for that patient. Similarly the outdegree \( OD(v) \) is the number of edges whose tail is the vertex \( v \); equivalently the number of 1 entries in the associated row of \( U \), or the number of pair comparisons that represent a win for that patient.

- For a specific trial arm assignment \( D \), trial arm wins are defined in the following manner.
An edge \( e = (i, j) \) is a *win* for a treatment patient if \( D_i = 1 \) and \( D_j = 0 \).

1. Define the \( E \times 1 \) vector \( T \) by \( T_e = k \) if edge \( e \) corresponds to \( k \) wins for treatment.
2. Let \( W_T = \sum_e T_e \) be the total number of wins for treatment patients.

An edge \( e = (i, j) \) is a *win* for a control patient if \( D_i = 0 \) and \( D_j = 1 \). The vector \( C \) is defined analogously to the vector \( T \).

We use the notations \( \mathbb{W}_T \) and \( \mathbb{W}_C \) for the numbers of treatment and control wins in the observed data, as opposed to the numbers in a particular permutation or bootstrap repetition.

The vector \( \mathbf{T}_C \) is the \( 2E \times 1 \) vector with \([TC_1, \ldots, TC_E] = [T_1, \ldots, T_E]\), and \([TC_{E+1}, \ldots, TC_{2E}] = [C_1, \ldots, C_E]\).

In analyzing the observed data and in the permutation distribution analysis, the vector \( \mathbf{T}_C \) will have entries in the set \( \{0,1\} \). Because bootstrap sampling is with repetition, values \( >1 \) are possible in the bootstrap analysis.

Figure 1 gives a small example. The U-matrix is

\[
U = \begin{bmatrix}
0 & -1 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & -1 \\
0 & -1 & 0 & 1 & 0 \\
0 & 0 & -1 & 0 & -1 \\
-1 & 1 & 0 & 1 & 0 \\
\end{bmatrix}.
\]  

This small example illustrates a \( U \) matrix that cannot be obtained from the Mann-Whitney or Gehan-Wilcoxon analysis. First, notice that patient 1 has a more favorable outcome than patient 5, and patient 5 has a more favorable outcome than patient 4; however patient 1 does not have a more favorable outcome than patient 4. Second, note that patients 1, 5, and 2 form a directed cycle, so that there is no way to determine which is superior. It is not difficult to create a hierarchy that will yield these results; we leave the details as an exercise for the reader.
Figure 1: Small Graph example
3.2 Analysis using the graphical model

For further analysis, we consider the outcome matrix $U$ to be fixed, and examine the behavior of the wins $W_T$ and $W_C$ under both analysis models. The goal of this analysis is to compute the means and variances of $W_T$ and $W_C$, and their covariance.

The analysis follows the same general pattern for both methods.

- The expected values $\text{Exp}(TC)$ and $\text{Exp}(TC(TC)^t)$ will be determined, from which we compute $\text{Var}(TC) = \text{Exp}(TC(TC)^t) - \text{Exp}(TC)(\text{Exp}(TC))^t$.

- For each pair of edges $(e, f)$, not necessarily distinct, the expected values $\text{Exp}(T_eT_f)$, $\text{Exp}(T_eC_f)$, $\text{Exp}(C_eT_f)$, and $\text{Exp}(C_eC_f)$ can be computed using counting arguments. The formulas will be different for different edge pairs, depending on the trial arm assignments and the geometric relationship of the edges.

We note that for edge pairs in certain configurations, it will happen that $\text{Exp}(T_eC_f) \neq \text{Exp}(C_eT_f)$. This inequality is not a violation of the symmetry of the variance matrix; the latter requires that $\text{Exp}(T_eC_f) = \text{Exp}(C_fT_e)$ for all edge pairs $(e, f)$.

- These computations are sufficient to compute all the terms of the expected value and variance of the vector $TC$. Define $S_{TT}, S_{TC}, S_{CT},$ and $S_{CC}$ by

\[
\text{Var}(TC) = \begin{bmatrix} S_{TT} & S_{TC} \\ S_{CT} & S_{CC} \end{bmatrix},
\]

where the partitioning corresponds to $T$ and $C$. Finally

\[
\text{Var} \left( \begin{bmatrix} W_T \\ W_C \end{bmatrix} \right) = \begin{bmatrix} \sum S_{TT} & \sum S_{TC} \\ \sum S_{CT} & \sum S_{CC} \end{bmatrix}.
\]

- Because the final variance matrix is symmetric, we do not explicitly need the individual terms $\text{Exp}(C_eT_f)$.

- Since the number of potential edges is $O(N^2)$, the number of edge pairs is $O(N^4)$, and it would not be practical to explicitly compute $\text{Var}(TC)$. Instead we count the number of times each geometric configuration of edges and trial arm assignments occurs, and compute (4) directly. The resulting algorithm will be $O(N^2)$, as will be seen below.
In all cases, the calculation of the expected value and variance terms is an elementary calculation involving binomial coefficients. In the text below we spare the reader the details; gentler derivations are available from the authors.

4 Wins analysis using the permutation distribution of trial arms

**Theorem 1.** The mean and variance of wins under the permutation distribution of trial arms can be computed in \(O(N^2)\) time.

**Proof.**

4.1 Expectations
- For a single edge \(e\), \(\text{Exp}_P(T_e) = \text{Exp}_P(T_e) = mn/N(N - 1)\).
- For the entire graph \(G\), \(\text{Exp}_P(W_T) = \text{Exp}_P(W_C) = Emn/N(N - 1)\).
- It follows that \(\text{Exp}_P(W_T - W_C) = 0\), which is a restatement of \(\text{Exp}_P(FS) = 0\), as was mentioned above.

4.2 Variances
There are \(E^2\) ordered pairs of edges, and the algorithm gives contributions for each possible geometric configuration of the edges. Some algebraic simplification of the formulas below is possible; the terms have been kept separate so as to align with the cases described in the derivation in Appendix C.

The first set of computations is performed separately at each vertex, and is derived from considering pairs of edges that meet at the vertex.

Computations at a single vertex \(v\).
The total count of pairs considered is given by

\[ F_P = \sum_v \left[ \text{ID}(v) + \text{ID}(v)(\text{ID}(v) - 1) + \text{OD}(v)(\text{OD}(v) - 1) + 2\text{ID}(v)\text{OD}(v) \right]. \]

The final variance then is computed by

\[
\text{Exp}_P \left( \begin{bmatrix} W_T & W_C \end{bmatrix} \begin{bmatrix} W_T^t & W_C^t \end{bmatrix} \right) = \sum_v \begin{bmatrix} eP_{TT}(v) & eP_{TC}(v) \\ eP_{CT}(v) & eP_{CC}(v) \end{bmatrix} + (E^2 - F_P) \frac{mn(m-1)(n-1)}{N(N-1)(N-2)(N-3)} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}.
\]

(5)
The last term in (5) comes from Case 6 in the derivation in Appendix C. Finally

\[ \text{Var}_P \left( \begin{bmatrix} W_T \\ W_C \end{bmatrix} \right) = \text{Exp}_P \left( \begin{bmatrix} W_T \\ W_C \end{bmatrix} \begin{bmatrix} W_T \\ W_C \end{bmatrix}^t \right) - E^2 \left( \frac{mn}{N(N-1)} \right)^2 \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}. \] (6)

The algorithm above, plus the discussion of complexity in section 7 completes the proof of theorem 1.

5 Graphical approach to the bootstrap

Since the bootstrap uses the same patients as in the observed data set, the graph \( G \) described in section 3 remains useful. The edges and vertices remain the same as previously.

- The trial arm assignments do not change from the observed data, and we refer to treatment vertices and control vertices based on the trial arms. Now for each bootstrap sample there is one new piece of information: the weight of a vertex will be the number of times that the corresponding patient appears in the bootstrap sample.

- An edge corresponding to a treatment win or a control win in the observed data will be called a treatment edge or a control edge.

- There are \((W_T + W_C)^2\) ordered pairs of edges corresponding to wins. All such pairs contribute to the variance, and will be counted below.

  - Edges joining a pair of treatment vertices or edges joining a pair of control vertices cannot correspond to wins in any bootstrap sample, and hence do not contribute to the final variance terms. In other words, the variance of wins in a bootstrap sample does not depend on within arm comparisons.

  - This situation is in contrast to the permutation variance, where within arm comparisons explicitly contribute to the final variance.

- For each vertex \( v \), let \( \#T_v \) denote the number of treatment edges adjacent to that vertex, and let \( \#C_v \) denote the number of control edges adjacent to that vertex.
As previously, the vector $T$ contains the number of treatment wins for each edge. In section 3, a treatment edge would have had $T_e = 1$, since each patient appears once in the observed data. Now we have $T_e$ equal to the product of the vertex weights, because each instance of patient $i$ in the bootstrap sample will be preferred to each instance of patient $j$. If either patient $i$ or patient $j$ does not appear in the bootstrap, then the corresponding weight will be 0, and there are no corresponding wins; thus $T_e = 0$. The vector $C$ of control wins is extended to the bootstrap context in the same manner.

6 Bootstrapping by sampling from the trial arms separately

In this section the sampling is separate in the two trial arms. With a population of size $m$ treatment patients and $n$ control patients, there are $m^m n^n$ possible bootstrap samples, and our underlying assumption is that all are equally likely.

One could also consider bootstrap sampling from the entire population; the formulas are similar in spirit to those given here, and the details are available from the authors.

**Theorem 2.** The mean and variance of wins under bootstrap sampling from the trial arms separately can be computed in $O(N^2)$ time.

*Proof.*

6.1 Expectations

As a first step we compute the expected values of the vectors $T$ and $C$.

- For a single edge $e$ that corresponds to a treatment win, $\text{Exp}_B(T_e) = 1$, and $\text{Exp}_B(C_e) = 0$.
- For the entire graph $G$, the expected number of treatment wins in a bootstrap sample is $W_T$.
- For a single edge $e$ that corresponds to a control win, $\text{Exp}_B(C_e) = 1$, and $\text{Exp}_B(T_e) = 0$.
For the entire graph $G$, the expected number of control wins in a bootstrap sample is $W_C$.

6.2 Variances

There are $(W_T + W_C)^2$ ordered pairs of edges to consider, and the algorithm gives contributions for each possible geometric configuration of the edges. Some algebraic simplification of the formulas below is possible; the terms have been kept separate so as to align with the cases described in the derivation of Appendix D.

The vertex computations are different for treatment and control vertices.

- **Computations at a single treatment vertex $v$.**

  \[
  eB2_{TT}(v) = \frac{\#T_v (2m - 1)(2n - 1)}{2mn} + \#T_v(\#T_v - 1)\frac{(2m - 1)(n - 1)}{mn}.
  \]

  *Case 1*

  \[
  eB2_{CC}(v) = \frac{\#C_v (2m - 1)(2n - 1)}{2mn} + \#C_v(\#C_v - 1)\frac{(2m - 1)(n - 1)}{mn}.
  \]

  *Case 2*

  \[
  eB2_{TC}(v) = \#T_v\#C_v\frac{(2m - 1)(n - 1)}{mn}.
  \]

  *Case 7*

  \[
  eB2_{CT}(v) = \#C_v\#T_v\frac{(2m - 1)(n - 1)}{mn}.
  \]

  *Case 9*

- **Computations at a single control vertex $v$.**
$$eB_{TT}(v) = \frac{\#T_v (2m - 1)(2n - 1)}{2mn} + \frac{\#T_v(\#T_v - 1)(m - 1)(2n - 1)}{mn}. \text{ *Case 1*}$$

$$eB_{CC}(v) = \frac{\#C_v (2m - 1)(2n - 1)}{2mn} + \frac{\#C_v(\#C_v - 1)(m - 1)(2n - 1)}{mn}. \text{ *Case 2*}$$

$$eB_{TC}(v) = \frac{(m - 1)(2n - 1)}{mn}. \text{ *Case 8*}$$

$$eB_{CT}(v) = \frac{(m - 1)(2n - 1)}{mn}. \text{ *Case 10*}$$

**Computations for the remaining edges.** To complete the calculation we need to include pairs of non-intersecting edges. The counts are

- $\text{remeB}_{TT} = W_T(W_T - 1) - \sum_v \#T_v(\#T_v - 1). \text{ *Case 11*}$
- $\text{remeB}_{TC} = W_TW_C - \sum_v \#T_v\#C_v. \text{ *Case 12*}$
- $\text{remeB}_{CT} = W_CW_T - \sum_v \#C_v\#T_v. \text{ *Case 13*}$
- $\text{remeB}_{CC} = W_C(W_C - 1) - \sum_v \#C_v(\#C_v - 1). \text{ *Case 14*}$

The final variance then is computed by

$$\text{Exp}_{B1} \left( \begin{bmatrix} W_T \\ W_C \end{bmatrix} \begin{bmatrix} W_T \\ W_C \end{bmatrix}^T \right) = \sum_v \begin{bmatrix} eB_{TT}(v) \\ eB_{TC}(v) \end{bmatrix} + \begin{bmatrix} (m - 1)(n - 1) \end{bmatrix} \begin{bmatrix} \text{remeB}_{TT} & \text{remeB}_{TC} \\ \text{remeB}_{CT} & \text{remeB}_{CC} \end{bmatrix},$$

and

$$\text{Var}_B \left( \begin{bmatrix} W_T \\ W_C \end{bmatrix} \right) = \text{Exp}_B \left( \begin{bmatrix} W_T \\ W_C \end{bmatrix} \begin{bmatrix} W_T \\ W_C \end{bmatrix}^T \right) - \begin{bmatrix} W_T^2 \\ W_C^2 \\ W_TW_C \\ W_CW_T \end{bmatrix}. \text{ (7)}$$
The algorithm above, plus the discussion of complexity in section 7 completes the proof of theorem 2.

R code for the algorithm is given in appendix A.2.

7 Complexity

- Since the mechanism of producing the matrix $U$ is undefined, all above diagonal entries must be examined at least once. Accordingly no algorithm for the permutation or bootstrap analysis can have time complexity faster than $O(N^2)$.

- Computing the total number of edges in the graph, and the numbers of treatment and control wins is $O(N^2)$.

- For each vertex, computing the indegree, the outdegree, and the counts $\#C_v$ and $\#T_v$, is performed by examining the entries in the corresponding row of the matrix $U$ and the entries in the trial arm vector. These operations are $O(N)$ for each vertex; since there are $N$ vertices the overall complexity of these computations is $O(N^2)$.

- The remaining computations are $O(1)$.

- Accordingly the overall complexity of each algorithm is $O(N^2)$.

When the Finkelstein-Schoenfeld method is used to compute the $U$ matrix, a bounded number of computations is performed for each pair of patients, so that the complexity of computing $U$ is also $O(N^2)$. One would anticipate that a considerably larger constant factor occurs in the complexity for computing $U$ than for computing the variance. Accordingly computing the variance and covariance of the wins, using any of the algorithms, would add only a negligible time to the overall computation.

One could in principle consider all possible permutations or bootstrap samples, compute the numbers of treatment and control wins for each, and then compute means and variances. Such a computation would be exponential in $N$, and hence completely unsatisfactory for a real example. We have performed such computations in very small examples as a test of our algorithm.
8 Win Ratio

Various authors including Pocock et al. (2012), Dong (2016), and Bebu (2016) examine the win ratio, which is 
\[ R_W = \frac{W_T}{W_C}, \]
and our methodology can yield information about the ratio also. As is standard with the analyses of frequencies (Agresti 2012), we analyze the log of the ratio because it is expected to be closer to normal than the ratio itself.

8.1 Pocock method

Pocock suggests the following method for analyzing the win ratio. From the observed win ratio \( R_W \), compute \( \log(R_W) \). Let \( z \) be the standardized normal deviate obtained from the Finkelstein-Schoenfeld significance test for the null hypothesis of no difference between the trial arms. Then an approximate standard error for \( \log(R_W) \) is \( s = \log(R_W) \div z \). This method is derived from the implicit assumption that the significance test of no difference between the trial arms should yield the same result whether working with win differences or win ratios. In the case of ordinary frequencies, this assumption is very close to being true, and it is surely reasonable in the present context. Once an approximate standard error has been computed, then confidence intervals can be computed and other inferences drawn.

As an alternative, Pocock suggests using the bootstrap, with separate sampling from the two treatment groups. Our formulas for the bootstrap could be used for this purpose.

8.2 Delta method

Dong uses maximum likelihood and U-statistics theory to compute means and variances for \( W_T \) and \( W_C \), and then uses delta method transformations to compute means and variances of the ratio. Even though the present means and variances for \( \text{Exp}(W_T) \) and \( \text{Exp}(W_C) \) are based on counting rather than on maximum likelihood, the delta method transformations could still be applied to our variance formulas. The transformations indeed do produce reasonable values; we will deal with such issues in a subsequent paper.
9 Discussion

The methodology above gives an efficient method of computing the exact variance of the trial arm wins $W_T$ and $W_C$. Whether or not this method constitutes a closed form depends on the definition of the term. The same comment would apply to the formula of Finkelstein-Schoenfeld for the variance of $W_T - W_C$. However, it seems to us that equation (2) is considerably simpler to apply in practice than the more classical formula involving ties, as given in Hollander and Wolfe (2013, equation (4.13)). Moreover, the derivation of the Finkelstein-Schoenfeld formula is vastly simpler than that for the classical formula.

In the case of a single variable Mann-Whitney test, the variances from the classical formula, from the Finkelstein-Schoenfeld formula, and from our formula will agree, because all are based on counting arguments using the exact permutation distribution of the trial arms. An algebraic derivation of the identity would seem to be very complicated, and hardly worth the effort. One can, of course, compute as many numerical examples as desired. For this purpose the SAS PROC NPAR1WAY and R function `wilcox.test` would be useful, but one must be careful to turn off the continuity correction. A similar note applies to the censored situation with a single variable. Both the Finkelstein-Schoenfeld formula and our formula implement the Gehan-Wilcoxon test as formulated by Gehan (1965); the implementations in the SAS PROC LIFETEST and R function `survdiff` are very slightly different.

The analysis presented here using the bootstrap distribution is similar in spirit to the analysis of the permutation distribution. Because of the special structure of the analysis considered here, we are able to determine the expected mean and variance from a bootstrap sample without actually doing the randomization. These values are effectively what would be found if an infinite number of bootstrap samples were taken. Efron has computed exact values in a different context (Efron, 1982, section 10.3).

Our computation of the bootstrap mean and variance is as easy as that for the permutation mean and variance used for the Mann-Whitney, Wilcoxon, or Gehan-Wilcoxon analyses. Accordingly the bootstrap evaluation could be considered a practical alternative to the original permutation test based analyses. Because of the extensive historical use of Mann-Whitney, we would be highly reluctant to recommend making such a change. All we suggest is that one should proceed carefully in cases where the classical method and the bootstrap produce different results from a significance test.
The derivations of the variance for wins need to consider only pairs of edges. If one wishes to further investigate the distribution of wins, there are a number of constraints involving more than two edges. For example, in an undirected cycle of 4 edges, it is possible for all four edges to correspond to wins. (In the example of figure 1 let $D_2 = 1, D_3 = 0, D_4 = 1, \text{and } D_5 = 0$. Then the cycle gives 1 treatment win and 3 control wins.) On the other hand, in an undirected cycle of 3 edges, there can be at most 2 wins, since at least one edge must have both vertices with the same trial arm assignment. Odd cycles play a pivotal role in maximum matching algorithms (Anderson, 1974), (Edmonds, 1965), and conceivably they will play a role in further investigations of the distribution.

The theoretical background for assuming normality in these analyses comes from the area of U-statistics. These items are discussed in Dong (2016), Finkelstein-Schoenfeld (1999), Lehmann (2006), and references therein. In this manuscript we deal with rather simpler combinatorial issues; we have nothing to add to the U-statistics literature.

10 A small example

The small example of figure 1 illustrates many of the cases above. Because of the small size, it is possible to work out the complete permutation distribution of trial arms, all the bootstrap samples, and compare the results with the formulas. Here we present just the results; further details are contained in the appendices.

The comparison matrix is defined by

$$U = \begin{bmatrix} 0 & -1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & -1 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & -1 \\ -1 & 1 & 0 & 1 & 0 \end{bmatrix}.$$  

The trial arms are

$$D = [1 \ 1 \ 0 \ 0 \ 0].$$
10.1 Permutation Distribution

The expectation and variance are
\[
\text{Exp}_P \frac{W_T}{W_C} = \frac{1}{5} [9]. \tag{8}
\]
and
\[
\text{Var}_P \frac{W_T}{W_C} = \frac{1}{100} \begin{bmatrix} 76 & -24 \\ -24 & 56 \end{bmatrix}. \tag{9}
\]

Then for the win difference,
\[
\text{Exp}_P (W_T - W_C) = 0.
\]

The same results can be obtained by applying the Finkelstein-Schoenfeld formulas to the example.

10.2 Bootstrap distribution – two sample

The expectation and variance are
\[
\text{Exp}_B \frac{W_T}{W_C} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}. \tag{10}
\]
and
\[
\text{Var}_B \frac{W_T}{W_C} = \frac{1}{6} \begin{bmatrix} 10 & -1 \\ -1 & 9 \end{bmatrix}. \tag{11}
\]

Then for the win difference,
\[
\text{Exp}_B (W_T - W_C) = 1.
\]
\[
\text{Var}_B (W_T - W_C) = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \text{Var}_B \begin{bmatrix} W_T \\ W_C \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix}^t = \frac{1}{6} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \begin{bmatrix} 10 & -1 \\ -1 & 9 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix}^t = \frac{21}{6} = 3.5.
\]

References

Agresti, A. (2012). *Categorical Data Analysis* (3rd ed.). New York: Wiley.

Anderson, W. (1974). Maximum matching and the game of slither. *Journal of Combinatorial Theory, series B* 17, 234–239.

Bebu, I. and J. Lachin (2016). Large sample inference for a win ratio analysis of a composite outcome based on prioritized components. *Biostatistics* 17, 178–187.

Buyse, M. (2010). Generalized pairwise comparisons of prioritized outcomes in the two-sample problem. *Statistics in Medicine* 29, 3245–3257.

Dong, G., D. Li, S. Ballerstedt, and M. Vandemeulebroecke (2016). A generalized analytic solution to the win ratio to analyze a composite endpoint considering the clinical importance order among components. *Pharmaceutical Statistics* 15, 430–437.

Edmonds, J. (1965). Paths, trees, and flowers. *Canadian Journal of Mathematics* 17, 449–467.

Efron, B. (1982). *The Jackknife, the Bootstrap and Other Resampling Plans*. Philadelphia: SIAM.

Finkelstein, D. and D. Schoenfeld (1999). Combining mortality and longitudinal measures in clinical trials. *Statistics in Medicine* 18, 1341–1354.

Gehan, E. (1965). A generalized wilcoxon test for comparing arbitrarily singly-censored samples. *Biometrika* 52, 203–223.
Hollander, M., D. Wolfe, and E. Chicken (2013). *Nonparametric Statistical Methods* (3rd ed.). New York: Wiley.

Lehmann, E. (2006). *Nonparametrics Statistical Methods Based on Ranks* (Revised First ed.). New York: Springer.

Leon, M., C. Smith, M. Mack, D. Miller, J. Moses, L. Svensson, E. Tuzcu, J. Webb, G. Fontana, R. Makkar, D. Brown, P. Block, R. Guyton, A. Pichard, J. Bavaria, H. Herrmann, P. Douglas, J. Petersen, J. Akin, W. Anderson, D. Wang, and S. Pocock (2010). Transcatheter aortic-valve implantation for aortic stenosis in patients who cannot undergo surgery. *New England Journal of Medicine* 363, 1597–1607.

Luo, X., J. Qui, S. Baib, and H. Tianc (2017). Weighted win loss approach for analyzing prioritized outcomes. *Statistics in Medicine* 36, 24522465.

Luo, X., H. Tian, S. Mohanty, and W. Tsai (2015). An alternative approach to confidence interval estimation for the win ratio statistic. *Biometrics* 71, 139145.

Pocock, S., C. Ariti, T. Collier, and D. Wang (2012). The win ratio: a new approach to the analysis of composite endpoints in clinical trials based on clinical priorities. *European Heart Journal* 33, 176182.

Spitzer, E., N. Van Mieghem, P. Pibarot, R. Hahn, S. Kodali, M. Maurer, T. Nazif, J. Rods-Cabau, J.-M. Paradis, A.-P. Kappetein, O. Ben-Yehuda, G.-A. van Es, F. Kallel, W. Anderson, J. Tijssen, and M. Leon (2016). Rationale and design of the transcatheter aortic valve replacement to unload the left ventricle in patients with advanced heart failure (TAVR UNLOAD) trial. *American Heart Journal* 182, 80–88.

Verbeeck, J., E. Spitzer, T. de Vries, G.-A. van Es, W. Anderson, N. Van Mieghem, M. Leon, G. Molenberghs, and J. Tijssen (2018). Generalized pairwise comparison methods to analyze (non)-hierarchical composite endpoints. *Under Review*.

Wikipedia contributors (2019). Directed graph — Wikipedia, the free encyclopedia. [Online; accessed 29-Mar-2019].
Appendices

There are number of appendices, which support and illustrate the derivations above.

- Appendix A: R code for the two situations.
- Appendix B: SAS Code for the two situations.
- Appendix C: Details of wins analysis using the Permutation model.
- Appendix D: Details of wins analysis using the two-sample bootstrap model.

In addition to these appendices, the following items are available from either of the authors.

- A gentle derivation of The Finkelstein-Schoenfeld Test, in the spirit of this manuscript.
- A derivation of the multinomial coefficient identities used in analyzing the bootstrap.
- Formulas and detailed derivations of wins analysis for bootstrapping from the entire population, rather than from the trial arms separately.
- R validation code for all three distributions. This file contains analysis of some examples using the the practical code, all permutations or bootstrap samples considered by hand, and randomized permutations or bootstrap samples.

A R code

This appendix contains R code for the practical computations. The code uses only base R functions, and has been tested using R version 3.5.1. The code follows closely the algorithm presented in sections 4 and 6.
A.1 R code for the permutation distribution

# Compute means and variances for trial arm wins,  
# based on the permutation distribution,  
# using an already computed win matrix and trial arm counts.  
# The algorithm is O(N^2) in time and space.

onevertex_P <- function(indegree, outdegree){
  # indegree and outdegree correspond to a specific vertex  
  # subscripts correspond to case numbering in manuscript  
  c(indegree, indegree*(indegree - 1), outdegree*(outdegree - 1),  
  indegree*outdegree, indegree*outdegree)
}

winsmeanandvariance_P <- function(winmatrix, trialarms){
  # code is modeled after manuscript section 4; some simplification is possible  
  # winmatrix is the skew matrix of wins, perhaps from a hierarchical evaluation  
  # m, n are the number of treatment and control patients: m + n = nrow(winmatrix)  
  # function computes the expected number of treatment and control wins, and their variance  
  # the actual trial assignments are not relevant for this function, just the counts  
  m <- sum(trialarms == 1); n <- sum(trialarms == 0)  
  N <- m + n  
  # had better be the dimension of winmatrix  
  indegrees <- colSums(winmatrix == 1)  
  outdegrees <- rowSums(winmatrix == 1)  
  # add the cases from all the vertices  
  casecounts <- rowSums(sapply(onevertex_P, indegrees, outdegrees))  
  expedge <- m*n/(N*(N-1))  
  expected = casecounts[1]*rep(expedge, 2)  
  names(expected) <- c("Test Wins", "Control Wins")  
  TTerms <- c(m*n/(N*(N - 1)), m*n*(m - 1)/(N*(N - 1)*(N - 2)),  
  m*n*(n - 1)/(N*(N - 1)*(N - 2)), 0, 0)  
  CTTerms <- c(m*n/(N*(N - 1)), m*n*(n - 1)/(N*(N - 1)*(N - 2)),  
  m*n*(m - 1)/(N*(N - 1)*(N - 2)), 0, 0)  
  TCTerms <- c(0, 0, 0, m*n*(n - 1)/(N*(N - 1)*(N - 2)),  
  m*n*(m - 1)/(N*(N - 1)*(N - 2)))  
  CTCTerms <- c(0, 0, 0, m*n*(m - 1)/(N*(N - 1)*(N - 2)),  
  m*n*(n - 1)/(N*(N - 1)*(N - 2)))  
  FP <- sum(casecounts)  
  expectedforvarianceterm1 <- c(sum(casecounts*TTerms), sum(casecounts*TCTerms),  
  sum(casecounts*CTTerms), sum(casecounts*CTCTerms))  
  expectedforvarianceterm2 <- (casecounts[1]^2 - FP)*m*n*(m - 1)*(n - 1)/  
  (N*(N - 1)*(N - 2)*(N - 3))  
  variance <- expectedforvarianceterm1 + expectedforvarianceterm2 -  
  casecounts[1]^2*(m*n/(N*(N - 1)))^2  
  variance <- matrix(variance, nrow = 2)  
  rownames(variance) <- c("Test Wins", "Control Wins");

colnames(variance) <- c("Test Wins", "Control Wins")
expdiff <- expected[1] - expected[2]; names(expdiff) <- NULL
vardiff <- variance[1, 1] - 2*variance[1, 2] + variance[2, 2]
list(Expected = expected, Variance = variance,
     'Expected Difference' = expdiff, 'Variance Difference' = vardiff)

A.2 R code for the two-sample bootstrap distribution

onevertex_B2 <- function(T, C, arm){
    # T and C are the number of treatment and control wins involving that vertex
    # arm is the trial arm for that vertex
    # subscripts correspond to case numbering in manuscript
    if (arm == 1) cases <- c(T/2, C/2, T*(T-1), 0, C*(C-1), 0, T*C, 0, C*T, 0)
    else cases <- c(T/2, C/2, 0, T*(T-1), 0, C*(C-1), 0, T*C, 0, C*T)
    cases
}

winsmeanandvariance_B2 <- function(winmatrix, trialarms){
    Observed <- wincomputations(winmatrix, trialarms)
    names(Observed) <- c("Test Wins", "Control Wins")
    N <- length(trialarms)
    treatmentarms <- trialarms == 1; m <- sum(treatmentarms)
    controlarms <- trialarms == 0; n <- sum(controlarms)
    # compute vectors giving the numbers of treatment and control wins for each vertex
    treatmentwinmatrix <- outer(treatmentarms, controlarms)*(winmatrix == 1)
    treatmentwinmatrix <- treatmentwinmatrix + t(treatmentwinmatrix)
    controlwinmatrix <- outer(controlarms, treatmentarms)*(winmatrix == 1)
    controlwinmatrix <- controlwinmatrix + t(controlwinmatrix)
    vertextreatmentwins <- rowSums(treatmentwinmatrix)
    vertexcontrolwins <- rowSums(controlwinmatrix)
    # compute treatment and control wins
    Twins <- sum(vertextreatmentwins)/2 # remove double count
    Cwins <- sum(vertexcontrolwins)/2 # remove double count
    # use the counted wins to count various cases corresponding to that vertex
    casecounts <- rowSums(mapply(onevertex_B2, vertextreatmentwins, vertexcontrolwins, trialarms))
    expedge <- 1 # expected wins for one edge
    expected = expedge*c(Twins, Cwins)
    names(expected) <- c("Test Wins", "Control Wins")
    TTerms <- c((2*m - 1)*(2*n - 1), 0, (2*m - 1)*(n - 1), (m - 1)*(2*n - 1),
                 rep(0, 6))/(m*n)
    CCTerms <- c(0, (2*m - 1)*(2*n - 1), 0, 0, (2*m - 1)*(n - 1), (m - 1)*(2*n - 1),
\( \text{rep}(0, 4)/(m \times n) \)

\[
\text{TTTerms} \leftarrow \text{c}\left(\text{rep}(0, 6), (2m - 1)(n - 1), (m - 1)(2n - 1), \text{rep}(0, 2)\right)/(m \times n)
\]

\[
\text{CTTerms} \leftarrow \text{c}\left(\text{rep}(0, 8), (2m - 1)(n - 1), (m - 1)(2n - 1)\right)/(m \times n)
\]

\[
\text{expectedforvarianceterm1} \leftarrow \text{c}\left(\text{sum}(\text{casecounts} \times \text{TTTerms}), \text{sum}(\text{casecounts} \times \text{CTTerms}), \text{sum}(\text{casecounts} \times \text{CTTerms}), \text{sum}(\text{casecounts} \times \text{CTTerms})\right)
\]

\[
\text{expectedforvarianceterm2} \leftarrow (m - 1)(n - 1)/(m \times n) \times \text{c}(\text{Twins} \times (\text{Twins} - 1) - \text{sum}(\text{casecounts}[c(3, 4)]),
\text{Twins} \times \text{Cwins} - \text{sum}(\text{casecounts}[c(7, 8)]),
\text{Cwins} \times \text{Twins} - \text{sum}(\text{casecounts}[c(9, 10)]),
\text{Cwins} \times (\text{Cwins} - 1) - \text{sum}(\text{casecounts}[c(5, 6)]))
\]

\[
\text{variance} \leftarrow \text{expectedforvarianceterm1} + \text{expectedforvarianceterm2} - \text{c}(\text{Twins} \times \text{Twins}, \text{Twins} \times \text{Cwins}, \text{Cwins} \times \text{Twins}, \text{Cwins} \times \text{Cwins})
\]

\[
\text{variance} \leftarrow \text{matrix}(\text{variance}, \text{nrow} = 2)
\]

\[
\text{rownames}(\text{variance}) \leftarrow \text{c}(\text{"Test Wins"}, \text{"Control Wins"})
\]

\[
\text{colnames}(\text{variance}) \leftarrow \text{c}(\text{"Test Wins"}, \text{"Control Wins"})
\]

\[
\text{expdiff} \leftarrow \text{expected}[1] - \text{expected}[2]
\]

\[
\text{vardiff} \leftarrow \text{variance}[1, 1] - 2 \times \text{variance}[1, 2] + \text{variance}[2, 2]
\]

\[
\text{list}(\text{Expected} = \text{expected}, \text{Variance} = \text{variance},
\text{‘Expected Difference‘} = \text{expdiff}, \text{‘Variance Difference‘} = \text{vardiff})
\]

\}

**B  SAS code**

This section contains various items of SAS code, which have been tested using SAS version 9.4.

**B.1  SAS Code for the permutation distribution**

/*This code creates an example U matrix*/

Data U;
  input U1 U2 U3 U4 U5;
  datalines;
  0 -1 0 0 1
  1 0 1 0 -1
  0 -1 0 1 0
  0 0 -1 0 -1
  -1 1 0 1 0
;
/*Assign values to be used later*/
%let nT=2; %let nC=3; %let N=%eval(&nT+&nC);
%let expedge=%sysevalf(&nT*&nC/(&N*(&N-1)));

/*Calculate the indegree, outdegree and counts for ID, 2, 3, 4 and 5*/
Data example;
set U;
array U U1-U&N;
indegree=0;
outdegree=0;
do i=1 to &N;
if U[i]=-1 then indegree= indegree+1;
   if U[i]=1 then outdegree = outdegree+1;
end;
drop i;
case2=indegree*(indegree-1);
case3=outdegree*(outdegree-1);
case4_5=indegree*outdegree;
run;

/*Sum indegree, and cases 2, 3, 4 and 5 */
proc means data=example (keep=outdegree indegree case2 case3 case4_5);
   output out=variance sum=;
run;

/*Calculate Fp counts and the variance for the number of wins for the treatment arm (TT), the variance for the number of wins for the control arm (CC) and their covariance (CT). Finally calculate the expectation (Exp_WD) and variance for the win difference (Var_WD)*/
data variance;
set variance;
Fp=indegree-case2-case3-2*case4_5;
factor1= &nT*(&nT - 1)*&nC/(&N*(&N - 1)*(&N - 2));
factor2= &nT*&nC*(&nC - 1)/(&N*(&N - 1)*(&N - 2));
TT= &expedge*indegree + factor1*case2 + factor2*case3;
CC= &expedge*indegree + factor2*case2 + factor1*case3;
TC= factor1*case4_5 + factor2*case4_5;
Exp_WD=outdegree*&expedge - indegree*&expedge;
Var_WD=TT+CC-2*TC;
drop _TYPE_ _FREQ_;
run;

proc print data=variance (keep=Exp_WD Var_WD TT CC TC); run;

**B.2 SAS Code for the two-sample bootstrap distribution**

/*Assign values to be used later*/
%let nT=2; %let nC=3; %let N=%eval(&nT+&nC); %let expedge=1;
%let C=%eval(&nT+1);

/*create treatment win matrix (UT) and control win matrix (UC) from U */
Data UT1;
set U (obs=&nT);
array Uc U&C-U&N;
array Ut U1-U&nT;
do i=1 to &nC;
if Uc[i]<0 then Uc[i]=0;
end;
do j=1 to &nT;
Ut[j]=0;
end;
drop i j;
run;

Data UT2;
set U (firstobs=&C);
array Uc U&C-U&N;
array Ut U1-U&nT;
do i=1 to &nC;
Uc[i]=0;
end;
do j=1 to &nT;
if Ut[j]<0 then Ut[j]=1; else Ut[j]=0;
end;
drop i j;
run;

Data UT;
set UT1 UT2;
run;

Data UC1;
set U (obs=&nT);
array Uc U&C-U&N;
array Ut U1-U&nT;
do i=1 to &nC;
  if Uc[i]=-1 then Uc[i]=1; else Uc[i]=0;
end;
do j=1 to &nT;
  Ut[j]=0;
end;
drop i j;
run;

Data UC2;
set U (firstobs=&C);
array Uc U&C-U&N;
array Ut U1-U&nT;
do i=1 to &nC;
  Uc[i]=0;
end;
do j=1 to &nT;
  if Ut[j]<0 then Ut[j]=0;
end;
drop i j;
run;

Data UC;
set UC1 UC2;
run;

/*sum the wins per row for both win matrices*/
Data ST;
set UT;
array U U1-U5;
T=0;
do i=1 to &N;
if U[i]=1 then T= T+1;
end;
drop i;
run;

Data SC;
set UC;
array U U1-U5;
C=0;
do i=1 to &N;
if U[i]=1 then C= C+1;
end;
drop i;
run;

/*merge the two sums columns and calculate case 3-10 */
data S;
merge ST(keep=T) SC(keep=C);
run;
data sumT;
set S (obs=&nT);
case3=T*(T-1);
case5=C*(C-1);
case7_9=T*C;
run;

data sumC;
set S (firstobs=&C);
case4=T*(T-1);
case6=C*(C-1);
case8_10=T*C;
run;
data cases;
set sumT sumC;
run;

proc means data=cases;
output out=variance sum=;
run;

/*Calculate case 1-10 counts and the variance for the number of wins for the treatment arm (TT), the variance for the number of wins for the control arm (CC) and their covariance (CT or TC). Finally calculate the expectation (Exp_WD) and variance for the win difference (Var_WD)*/
data variance;
set variance;
case1=T/2;
case2=C/2;
factor0 = (2*&&nT-1)*(2*&&nC - 1)/(&&nT*&&nC);
factor1t = (2*&&nT-1)*(&&nC-1)/(&&nT*&&nC);
factor1c = (&&nT-1)*(2*&&nC-1)/(&&nT*&&nC);
factor2 = (&&nT-1)*(&&nC-1)/(&&nT*&&nC);
remTT=case1*(case1-1)-(case3+case4);
remCC=case2*(case2-1)-(case5+case6);
remTC=case7_9*factor1t + case8_10*factor1c + remTC*factor2-case1*case2;
TT = case1*factor0 + case3*factor1t + case4*factor1c + remTT*factor2-case1**2;
CC = case2*factor0 + case5*factor1t + case6*factor1c + remCC*factor2-case2**2;
TC = case7_9*factor1t + case8_10*factor1c + remTC*factor2-case1*case2;
Exp_WD=case1*expedge - case2*expedge - case2*expedge;
Var_WD=TT+CC-2*TC;
drop _TYPE_ _FREQ_; 
run;

proc print data=variance (keep=Exp_WD Var_WD); run;
C Detailed wins analysis using the permutation model

In this section we give detailed derivations of the formulas used in section 4.

C.1 Expectations

The total number of trial arm assignments is \( \binom{N}{m} \). For a specific edge \( e = (i, j) \) the number of assignments with \( D_i = 1 \) and \( D_j = 0 \) is \( \binom{N-2}{m-1} \). Hence for all edges \( e \),

\[
\text{Exp}_P(T_e) = \frac{(N-2)}{(m-1)} \binom{N}{m} = \frac{mn}{N(N-1)}, \tag{C1}
\]

and

\[
\text{Exp}_P(W_T) = E \frac{mn}{N(N-1)}. \tag{C2}
\]

Since having \( D_i = 1 \) and \( D_j = 0 \) is equally likely to having \( D_i = 0 \) and \( D_j = 1 \), it follows that \( \text{Exp}_P(C_e) = \text{Exp}_P(T_e) \), and \( \text{Exp}_P(W_C) = \text{Exp}_P(W_T) \).

C.2 Variances

The variance is found by considering the geometric configuration of pairs of edges. The description for each case is merely for ease of following the computations; the actual algorithm uses only the counts and the expected value formulas.

The first computations are performed separately at each vertex, and the formulas below give \( \text{Exp}_P(T_eT_f) \), where \( (e, f) \) is an ordered pair of edges, possibly the same.

Computations at a single vertex \( v \).

Case 1: A single edge \( e \) with head \( v \):

- The term appears \( \text{ID}(v) \) times.
- Since \( T_e \) is 0 or 1, we have using (C1)

\[
\text{Exp}_P(T_e^2) = \text{Exp}_P(T_e) = \frac{mn}{N(N-1)}.
\]
Similarly, \( \exp_P(C_e^2) = \exp_P(C_e) = \frac{mn}{N(N - 1)}. \)

- Since an edge cannot be simultaneously a win for treatment and for control, we have \( T_tC_e = 0 \) for all edges \( e \) and all permutations, and thus \( \exp_P(T_tC_e) = \exp_P(C_eT_e) = 0. \)

Case 2: An ordered pair \((e, f)\) of distinct edges, each with head \( v \):

- Edges \( (3, 4) \) and \( (5, 4) \) in figure 1 are an example of distinct edges with a common head.
- Because the pair is ordered, both \((e, f)\) and \((f, e)\) will appear in the computation. Accordingly the term appears \( \text{ID}(v)(\text{ID}(v) - 1) \) times.
- Consider an ordered pair of edges \( e = (v_i, v_j) \) and \( f = (v_k, v_j) \) where \( \{v_i, v_j, v_k\} \) are distinct; that is vertex \( v_j \) is the head for both edges. Then \( T_eT_f = 1 \) iff \( D_i = D_k = 1 \) and \( D_j = 0 \). There remain \( N - 3 \) patients to be assigned trial arms, with \( m - 2 \) treatment patients and \( n - 1 \) control patients. The number of trial arm permutations satisfying these conditions is \( \binom{N - 3}{m - 2} \). Accordingly.

\[
\exp_P(T_eT_f) = \frac{(N - 3)}{(m - 2)} / \binom{N}{m} = \frac{mn(m - 1)}{N(N - 1)(N - 2)}.
\]

- Similarly,

\[
\exp_P(C_eC_f) = \frac{(N - 3)}{(n - 2)} / \binom{N}{m} = \frac{mn(n - 1)}{N(N - 1)(N - 2)}.
\]

- \( T_e \) and \( C_f \) cannot both be 1, because the former requires \( D_j = 0 \), and the latter requires \( D_j = 1 \). Hence \( T_eC_f = 0 \) always. Accordingly

\[
\exp_P(T_eC_f) = \exp_P(C_eT_f) = 0.
\]

Case 3: An ordered pair \((e, f)\) of distinct edges, each with tail \( v \):

- Edges \( (5, 2) \) and \( (5, 4) \) in figure 1 are an example of distinct edges with a common tail.
• Because the pair is ordered, both \((e, f)\) and \((f, e)\) will appear in the computation. Accordingly the term appears \(\text{OD}(v)(\text{OD}(v) - 1)\) times in the computations for vertex \(v\).

• Consider an ordered pair of edges \(e = (v_i, v_j)\) and \(f = (v_i, v_l)\) where \(\{v_i, v_j, v_l\}\) are distinct; that is vertex \(v_i\) is the tail for both edges. Then \(T_eT_f = 1\) iff \(D_i = 1\) and \(D_j = D_l = 0\). The number of trial arm permutations satisfying these conditions is \(\binom{N-3}{m-1}\).

Accordingly
\[
\text{Exp}_P(T_eT_f) = \frac{\binom{N-3}{m-1}}{\binom{N}{m}} = \frac{mn(n-1)}{N(N-1)(N-2)}.
\]

Similarly,
\[
\text{Exp}_P(C_eC_f) = \frac{\binom{N-3}{n-1}}{\binom{N}{m}} = \frac{mn(m-1)}{N(N-1)(N-2)}.
\]

• \(T_e\) and \(C_f\) cannot both be 1, because the former requires \(D_i = 1\), and the latter requires \(D_i = 0\). Hence \(T_eC_f = 0\) always. Accordingly
\[
\text{Exp}_P(T_eC_f) = \text{Exp}_P(C_eT_f) = 0.
\]

Case 4: An ordered pair \((e, f)\) of distinct edges, where vertex \(v\) is the tail of edge \(e\) and the head of edge \(f\):

• Edges \((1, 5)\) and \((2, 1)\) in figure 4 are an example of this situation.

• The term appears \(\text{ID}(v)\text{OD}(v)\) times.

• Consider a pair of edges \(e = (v_i, v_j)\) and \(f = (v_k, v_i)\) where \(\{v_i, v_j, v_k\}\) are distinct; that is vertex \(v_i\) is the tail of edge \(e\) and the head of edge \(f\). Then \(T_e\) and \(T_f\) can never both be 1, because the former requires \(D_i = 1\) and the latter requires \(D_i = 0\). Accordingly
\[
\text{Exp}_P(T_eT_f) = \text{Exp}_P(C_eC_f) = 0.
\]

• \(T_eC_f = 1\) iff \(D_i = 1\) and \(D_j = D_k = 0\). Accordingly
\[
\text{Exp}_P(T_eC_f) = \frac{\binom{N-3}{m-1}}{\binom{N}{m}} = \frac{mn(n-1)}{N(N-1)(N-2)}.
\]

33
\[ C_e T_f = 1 \text{ iff } D_i = 0 \text{ and } D_j = D_k = 1. \] Accordingly

\[
\text{Exp}_P(C_e T_f) = \left( \frac{N - 3}{n - 1} \right) \left/ \binom{N}{m} \right. = \frac{mn(m - 1)}{N(N - 1)(N - 2)}.
\]

Case 5: An ordered pair \((e, f)\) of distinct edges, where vertex \(v\) is the head of edge \(e\) and the tail of edge \(f\):

- Edges \((2, 1)\) and \((1, 5)\) in figure 1 are an example of this situation. Note that these are the same edges mentioned in the previous case, but the order is different.
- The term appears \(ID(v)OD(v)\) times.
- Consider a pair of edges \(e = (v_i, v_j)\) and \(f = (v_j, v_k)\) where \(\{v_i, v_j, v_k\}\) are distinct; that is vertex \(v_j\) is the head of edge \(e\) and the tail of edge \(f\). Then \(T_e\) and \(T_f\) can never both be 1, nor can \(C_e\) and \(C_f\). Accordingly

\[
\text{Exp}_P(T_e T_f) = \text{Exp}_P(C_e C_f) = 0.
\]

- \(T_e C_f = 1\) iff \(D_i = D_k = 1\) and \(D_j = 0\). Accordingly

\[
\text{Exp}_P(T_e C_f) = \left( \frac{N - 3}{m - 2} \right) \left/ \binom{N}{m} \right. = \frac{mn(m - 1)}{N(N - 1)(N - 2)}.
\]

Similarly,

- \(\text{Exp}_P(C_e T_f) = \left( \frac{N - 3}{n - 2} \right) \left/ \binom{N}{m} \right. = \frac{mn(n - 1)}{N(N - 1)(N - 2)}.

**Computation for the remaining edge pairs.**

Case 6: An ordered pair \((e, f)\) of non-intersecting edges:

- Edges \((2, 3)\) and \((1, 5)\) in figure 1 are an example of non-intersecting edges.
- The computations in cases 1 - 5 above have considered all possibilities for single edges and for intersecting edges. There are a total
of \( E^2 \) ordered pairs of edges, the total count of pairs considered is given by

\[
F = \sum_v \left[ \text{ID}(v) + \text{ID}(v)(\text{ID}(v) - 1) + \text{OD}(v)(\text{OD}(v) - 1) + 2\text{ID}(v)\text{OD}(v) \right].
\]

The remaining count is \( E^2 - F \).

- Consider a pair of edges \( e = (v_i, v_j) \) and \( f = (v_k, v_l) \) where \( \{v_i, v_j, v_k, v_l\} \) are distinct; that is the edges do not meet. Then \( T_eT_f = 1 \) iff \( D_i = D_k = 1 \) and \( D_j = D_l = 0 \). The number of trial arm permutations satisfying these conditions is \( \binom{N-4}{m-2} \). The same counting argument applies to \( C_eC_f \), \( T_eC_f \), and \( C_eT_f \). Accordingly

\[
\text{Exp}_p(T_eT_f) = \text{Exp}_p(T_eC_f) = \text{Exp}_p(C_eT_f) = \text{Exp}_p(C_eC_f)
= \binom{N-4}{m-2} / \binom{N}{m}
= \frac{mn(m-1)(n-1)}{N(N-1)(N-2)(N-3)}.
\]

D Detailed wins analysis using the two-sample bootstrap

In this section we give detailed derivations of the formulas used in section 6. The computations use elementary sums of multinomial coefficients; these are the same sums that one would evaluate in computing moments of the multinomial distribution.

D.1 Expectations

Consider a treatment edge \( e \). Then the head corresponds to a control patient, and the tail to a treatment patient. The sampling for head and tail is from the two different trial arms, and for the expected value of \( T_e \), we have
\[ \text{Exp}_B(T_e) = \frac{1}{m^m n^n} \sum_{k_1 + \ldots + k_m = m} k_1 \left( \begin{array}{c} m \\ k_1, k_2, \ldots, k_m \end{array} \right) \sum_{l_1 + \ldots + l_l = l} l_1 \left( \begin{array}{c} n \\ l_1, l_2, \ldots, l_m \end{array} \right) \]
\[ = \frac{1}{m^m n^n} m^m n^n = 1. \] (D1)

For this edge we will have \( \text{Exp}_B(C_e) = 0 \), since this edge cannot correspond to a control win in any bootstrap sample. Similarly, if an edge \( f \) is a control edge, then \( \text{Exp}_B(T_f) = 0 \) and \( \text{Exp}_B(C_f) = 1 \).

### D.2 Variances

**Computations at a single vertex.**

Case 1: A treatment edge, adjacent to the vertex \( v \):

- The term appears \( \#T_v/2 \) times. The factor 1/2 is due to the fact that the same edge will be counted at its other end also.
- Let the edge be \( e = (v_1, v_2) \), with corresponding vertex weights \( k_1, k_2 \). The number of possibilities for \( T_e > 0 \) is \( k_1 k_2 \). Then

\[ \text{Exp}_B(T_e^2) = \frac{1}{m^m n^n} \sum_{k_1 + \ldots + k_m = m} k_1^2 \left( \begin{array}{c} m \\ k_1, k_2, \ldots, k_m \end{array} \right) \times \sum_{l_1 + \ldots + l_l = l} l_1^2 \left( \begin{array}{c} n \\ l_1, l_2, \ldots, l_m \end{array} \right) \]
\[ = \frac{1}{m^m n^n} (2m - 1) m^{m-1} (2n - 1) n^{n-1} \]
\[ = \frac{(2m - 1) (2n - 1)}{m n}. \] (D2)

For this same edge, \( C_e = 0 \) for all bootstrap samples, and hence

\[ \text{Exp}_B(C_e^2) = \text{Exp}_B(T_e C_e) = \text{Exp}_B(C_e T_e) = 0. \]

Case 2: A control edge, adjacent to the vertex \( v \):

- The term appears \( \#C_v/2 \) times. The factor 1/2 is due to the fact that the same edge will be counted at its other end also.
• The remaining computations are similar to case 1.

\[ \text{Exp}_{B_1}(T_e^2) = \text{Exp}_{B_1}(T_eC_e) = \text{Exp}_{B_1}(C_eT_e) = 0. \]

\[ \text{Exp}_{B_1}(C_e^2) = \frac{(2m - 1)(2n - 1)}{mn}. \]

Case 3: An ordered pair \((e, f)\) of distinct treatment edges, each adjacent to the treatment vertex \(v\):

• The term appears \(\#T_v(\#T_v - 1)\) times.

• Let the edges be \(e = (v_1, v_2)\) and \(f = (v_1, v_3)\), with corresponding vertex weights are \(k_1, k_2, k_3\). Then the common vertex \(v_1\) is a treatment vertex. The number of possibilities for \(T_e > 0\) is \(k_1k_2\), and for \(T_f > 0\) is \(k_1k_3\). Then the number of samples that produce both \(T_e > 0\) and \(T_f > 0\) is \(k_1^2k_2k_3\), and

\[ \text{Exp}_B(T_eT_f) = \frac{1}{m^m n^n} \sum_{k_1+\ldots+k_m=m} k_1^2 \left( \begin{array}{c} N \\ k_1, k_2, \ldots, k_N \end{array} \right) \times \sum_{l_1+\ldots+l_n=n} l_2l_3 \left( \begin{array}{c} n \\ l_1, l_2, \ldots, l_n \end{array} \right) \]

\[ = \frac{1}{m^m n^n} (2m - 1)m^{m-1}(n - 1)n^{n-1} \]

\[ = \frac{(2m - 1)(n - 1)}{mn}. \]

Because both edges correspond to treatment wins, we have

\[ \text{Exp}_B(C_eC_f) = \text{Exp}_B(T_eC_f) = \text{Exp}_B(C_eT_f) = 0. \]

Case 4: An ordered pair \((e, f)\) of distinct treatment edges, each adjacent to the control vertex \(v\):

• The term appears \(\#T_v(\#T_v - 1)\) times.

• Let the edges be \(e = (v_1, v_2)\) and \(f = (v_3, v_2)\), with corresponding vertex weights are \(k_1, k_2, k_3\). Then the common vertex \(v_2\) is a control vertex. The number of possibilities for \(T_e > 0\) is \(k_1k_2\), and
for $T_f > 0$ is $k_3k_2$. Then the number of samples that produce both $T_e > 0$ and $T_f > 0$ is $k_1k_2^2k_3$, and

$$\text{Exp}_B(T_eT_f) = \frac{1}{mn} \sum_{k_1+\ldots+k_m=m} k_1k_3 \binom{N}{k_1, k_2, \ldots, k_N}$$

$$\times \sum_{l_1+\ldots+l_n=n} l_2^2 \binom{n}{l_1, l_2, \ldots, l_n}$$

$$= \frac{1}{mn}(m-1)m^{m-1}(2n-1)n^{n-1}$$

$$= \frac{(m-1)(2n-1)}{mn}$$

Because both edges correspond to treatment wins, we have

$$\text{Exp}_B(C_eC_f) = \text{Exp}_B(T_eC_f) = \text{Exp}_B(C_eT_f) = 0.$$

Case 5: An ordered pair $(e, f)$ of distinct control edges, each adjacent to the treatment vertex $v$:

- The term appears $\#C_v(\#C_v - 1)$ times.
- The expected value computations are similar to those of case 3.

$$\text{Exp}_B(C_eC_f) = \frac{(2m-1)(n-1)}{mn}.$$

$$\text{Exp}_B(T_eT_f) = \text{Exp}_B(T_eC_f) = \text{Exp}_B(C_eT_f) = 0.$$

Case 6: An ordered pair $(e, f)$ of distinct control edges, each adjacent to the control vertex $v$:

- The term appears $\#C_v(\#C_v - 1)$ times.
- The expected value computations are similar to those of case 4.

$$\text{Exp}_B(C_eC_f) = \frac{(m-1)(2n-1)}{mn}.$$

$$\text{Exp}_B(T_eT_f) = \text{Exp}_B(T_eC_f) = \text{Exp}_B(C_eT_f) = 0.$$

Case 7: An ordered pair $(e, f)$ of edges, where $e$ is a treatment edge and $f$ is a control edge, each adjacent to the treatment vertex $v$: 38
• The term appears \#T_v\#C_v times.
• The expected value computations are similar to those of case 3.

\[ \text{Exp}_B(T_eC_f) = \frac{(2m-1)(n-1)}{mn} \]
\[ \text{Exp}_B(T_eT_f) = \text{Exp}_B(C_eT_f) = \text{Exp}_B(C_eC_f) = 0. \]

Case 8: An ordered pair \((e, f)\) of edges, where \(e\) is a treatment edge and \(f\) is a control edge, each adjacent to the control vertex \(v\):

• The term appears \#T_v\#C_v times.
• The expected value computations are similar to those of case 4.

\[ \text{Exp}_B(T_eC_f) = \frac{(m-1)(2n-1)}{mn} \]
\[ \text{Exp}_B(T_eT_f) = \text{Exp}_B(C_eT_f) = \text{Exp}_B(C_eC_f) = 0. \]

Case 9: An ordered pair \((e, f)\) of edges, where \(e\) is a treatment edge and \(f\) is a control edge, each adjacent to the treatment vertex \(v\):

• The term appears \#C\#T_v times.
• The expected value computations are similar to those of case 3.

\[ \text{Exp}_B(C_eT_f) = \frac{(2m-1)(n-1)}{mn} \]
\[ \text{Exp}_B(T_eT_f) = \text{Exp}_B(T_eC_f) = \text{Exp}_B(C_eC_f) = 0. \]

Case 10: An ordered pair \((e, f)\) of edges, where \(e\) is a control edge and \(f\) is a treatment edge, each adjacent to the control vertex \(v\):

• The term appears \#C\#T_v times.
• The expected value computations are similar to those of case 4.

\[ \text{Exp}_B(C_eT_f) = \frac{(m-1)(2n-1)}{mn} \]
\[ \text{Exp}_B(T_eT_f) = \text{Exp}_B(T_eC_f) = \text{Exp}_B(C_eC_f) = 0. \]

Computations for the remaining edge pairs.
Case 11: An ordered pair \((e, f)\) of non-intersecting edges, both treatment edges:

- The term appears \(W_T(W_T - 1) - \sum_v \#T_v(\#T_v - 1)\) times. This count comes from \(W_T(W_T - 1)\), which is the total number of ordered pairs of distinct treatment edges, minus the pairs considered in cases 3 and 4 above.

- Let the edges be \(e = (v_1, v_2)\) and \(f = (v_3, v_4)\), with corresponding vertex weights are \(k_1, k_2, k_3, k_4\). The number of possibilities for \(T_e > 0\) is \(k_1k_2\), and for \(T_f > 0\) is \(k_3k_4\). Then the number of samples that produce both \(T_e > 0\) and \(T_f > 0\) is \(k_1k_2k_3k_4\), and

\[
\text{Exp}_B(T_eT_f) = \frac{1}{m^m n^n} \sum_{k_1 + \ldots + k_m = m} k_2k_4 \binom{N}{k_1, k_2, \ldots, k_N} \times \sum_{l_1 + \ldots + l_n = n} l_1l_3 \binom{n}{l_1, l_2, \ldots, l_n} \\
= \frac{1}{m^m n^n} (m-1)m^{m-1}(n-1)n^{n-1} \\
= \frac{(m-1)(n-1)}{mn}.
\]

Because both edges correspond to treatment wins, we have

\[\text{Exp}_B(C_eC_f) = \text{Exp}_B(T_eC_f) = \text{Exp}_B(C_eT_f) = 0.\]

Case 12: An ordered pair \((e, f)\) of non-intersecting edges, both control edges. Reasoning as in case 11 we have

The term appears \(W_C(W_C - 1) - \sum_v \#C_v(\#C_v - 1)\) times.

\[
\text{Exp}_{B1}(T_eT_f) = \text{Exp}_{B1}(T_eC_f) = \text{Exp}_{B1}(C_eT_f) = 0. \\
\text{Exp}_{B1}(C_eC_f) = \frac{(m-1)(n-1)}{mn}.
\]

Case 13: An ordered pair \((e, f)\) of non-intersecting edges, where \(e\) is a treatment edge and \(f\) is a control edge. Reasoning as in case 11 we have

The term appears \(W_TW_C - \sum_v \#T_v\#C_v\) times.

\[
\text{Exp}_{B1}(T_eT_f) = \text{Exp}_{B1}(C_eT_f) = \text{Exp}_{B1}(C_eC_f) = 0.
\]
\[ \text{Exp}_{B_1}(T_eC_f) = \frac{(m - 1)(n - 1)}{mn}. \]

Case 14: An ordered pair \((e, f)\) of non-intersecting edges, where \(e\) is a control edge and \(f\) is a treatment edge. Reasoning as in case 11 we have

The term appears \(\mathcal{W}_C \mathcal{W}_T - \sum_v \#C_v \#T_v\) times.

\[ \text{Exp}_{B_1}(T_eT_f) = \text{Exp}_{B_1}(T_eC_f) = \text{Exp}_{B_1}(C_eC_f) = 0. \]

\[ \text{Exp}_{B_1}(C_eT_f) = \frac{(m - 1)(n - 1)}{mn}. \]