Supplement: D-Efficient Composite-type Second Order Designs via Computer Search

The random search procedures used in the article are based on the following basic algorithms. The procedure Random(S) used in these algorithms, when called with a finite set S as argument, selects an element of S randomly with equal probability.

1. RandomDesign generates a random design.

```
procedure RandomDesign(m, n)
    X ← new n × m matrix
    for i ← 1, 2, ..., n do
        for j ← 1, 2, ..., m do
            X_ij ← Random({−1, 0, 1})        ▷ Select one with equal probability
        end for
    end for
    return X
end procedure
```

2. ImproveDesignGreedy improves a design by greedy search.

```
procedure ImproveDesignGreedy(X)
    m ← number of rows of X
    n ← number of columns of X
    for i ← 1, 2, ..., n do
        for j ← 1, 2, ..., m do
            for x ← −1, 0, 1 do
                E_ijx ← D-efficiency when X_ij is replaced by x
            end for
        end for
    end for
    if max E_ijx > D-efficiency of X then
        return updated X corresponding to max E_ijx
    else
        return X                                  ▷ Unchanged
    end if
end procedure
```

3. ImproveDesignRandom improves a design by random search.

procedure ImproveDesignRandom(X)
  $Y \leftarrow X$
  $m \leftarrow$ number of rows of X
  $n \leftarrow$ number of columns of X
  $i \leftarrow \text{Random}(\{1, 2, \ldots, n\})$
  $j \leftarrow \text{Random}(\{1, 2, \ldots, m\})$
  $Y_{ij} \leftarrow \text{Random}(\{-1, 0, 1\} \setminus \{X_{ij}\})$ ▷ Replace Y_{ij} randomly
  if D-efficiency of $Y >$ D-efficiency of X **then**
    return Y ▷ Updated
  else
    return X ▷ Unchanged
  end if
end procedure