1. Computing the vector of criteria weights

   compute normalized pairwise comparison matrix:

$$\overline{a}_{jk} = \frac{a_{jk}}{\sum_{l=1}^{m} a_{lk}}.$$

   $k = 1 : \overline{a_{jk}} = [15/23 \quad 5/23 \quad 3/23]^T$

   $k = 2 : \overline{a_{jk}} = [9/13 \quad 3/13 \quad 1/13]^T$

   $k = 3 : \overline{a_{jk}} = [5/9 \quad 1/3 \quad 1/9]^T$

   criteria weight vector w (that is an m-dimensional column vector):

$$w_j = \frac{\sum_{l=1}^{m} \overline{a}_{jl}}{m}.$$

   $j = 1 : w_j = 0.633$

   $j = 2 : w_j = 0.261$

   $j = 3 : w_j = 0.106$

2. Computing the matrix of option scores

   $B^{(1)}$:

   $k = 1 : \overline{b_{jk}} = [21/31 \quad 7/31 \quad 3/31]^T$

   $k = 2 : \overline{b_{jk}} = [5/7 \quad 5/21 \quad 1/21]^T$

   $k = 3 : \overline{b_{jk}} = [7/13 \quad 5/13 \quad 1/13]^T$

   $j = 1 : s_j = 0.643$

   $j = 2 : s_j = 0.283$

   $j = 3 : s_j = 0.074$

   $s^{(1)} = [0.643 \quad 0.283 \quad 0.074]^T$

$B^{(2)}$:

$k = 1: \overline{b_{jk}} = [1/7 \quad 5/7 \quad 1/7]^T$

$k = 2: \overline{b_{jk}} = [1/7 \quad 5/7 \quad 1/7]^T$

$k = 3: \overline{b_{jk}} = [1/7 \quad 5/7 \quad 1/7]^T$

$j = 1: s_j = 0.143$

$j = 2: s_j = 0.714$

$j = 3: s_j = 0.143$

$s^{(2)} = [0.143 \quad 0.714 \quad 0.143]^T$

$B^{(3)}$:

$k = 1: \overline{b_{jk}} = [45/59 \quad 9/59 \quad 5/59]^T$

$k = 2: \overline{b_{jk}} = [15/19 \quad 3/19 \quad 1/19]^T$

$k = 3: \overline{b_{jk}} = [9/13 \quad 3/13 \quad 1/13]^T$

$j = 1: s_j = 0.748$

$j = 2: s_j = 0.18$

$j = 3: s_j = 0.072$

$s^{(3)} = [0.748 \quad 0.18 \quad 0.072]^T$

3. Ranking the options: $v = S * w = [0.523 \quad 0.385 \quad 0.092]^T$

   ⇒ the first option turns out to be the most preferable.

Checking consistency

Consistency Index:

$$CI = \frac{x - m}{m - 1}$$

A perfectly consistent decision maker should always obtain CI=0, but small values of inconsistency may be tolerated:

$$\frac{CI}{RI} < 0.1$$

RI for our case equal to 0.58.

Calculate maximum eigenvalue for each reciprocal matrix.

```
In [2]: import numpy as np

In [2]: A = np.array([[1, 3, 5], [1/3, 1, 3], [1/5, 1/3, 1]])

In [5]: w, v = np.linalg.eig(A)
        np.amax(w)

Out[5]: (3.0385110905581727+0j)

In [3]: B1 = np.array([[1, 3, 7], [1/3, 1, 5], [1/7, 1/5, 1]])
        w, v = np.linalg.eig(B1)
        np.amax(w)

Out[3]: (3.064887579872819+0j)

In [4]: B2 = np.array([[1, 1/5, 1], [5, 1, 5], [1, 1/5, 1]])
        w, v = np.linalg.eig(B2)
        np.amax(w)

Out[4]: 3.0

In [5]: B3 = np.array([[1, 5, 9], [1/5, 1, 3], [1/9, 1/3, 1]])
        w, v = np.linalg.eig(B3)
        np.amax(w)

Out[5]: (3.029063766798436+0j)
```

$$CI_A = \frac{3.039-3}{3-1} = 0.0195, \qquad CI/RI = 0.034 \Rightarrow consistent$$

$$CI_{B1} = \frac{3.065-3}{3-1} = 0.0325, \qquad CI/RI = 0.056 \Rightarrow consistent$$

$$CI_{B2} = \frac{3-3}{3-1} = 0 \Rightarrow consistent$$

$$CI_{B1} = \frac{3.029-3}{3-1} = 0.0145, \qquad CI/RI = 0.025 \Rightarrow consistent$$

Meaning of the reciprocal matrix A is like individual judgment. By granular reciprocal matrices we mean matrices whose entries are not plain numbers but information granules about pairwise comparison. The criteria weight vector gives us the average priority of the criteria. It could be considered as a membership score of a fuzzy set. This set specifies the membership of each criterion in the set of desired solutions.