

Article

MATASS: the software for multi-attribute assessment problems

WenJun Zhang¹, YanHong Qi², Xin Li³

¹School of Life Sciences, Sun Yat-sen University, Guangzhou 510275, China; International Academy of Ecology and Environmental Sciences, Hong Kong

²Sun Yat-sen University Libraries, Sun Yat-sen University, Guangzhou 510275, China

³College of Plant Protection, Northwest A & F University, Yangling 712100, Shaanxi, China

E-mail: zhwj@mail.sysu.edu.cn, wjzhang@iaees.org, qiyh@mail.sysu.edu.cn, lixin57@hotmail.com

Received 15 September 2015; Accepted 19 April 2016; Published 1 June 2017



Abstract

In present study, we developed the software for multi-attribute assessment problems, MATASS (Multi-Attribute Assessment System). The procedures of MATASS include, (1) for a multi-attribute assessment problem, there are m attributes for the assessment of n objects (ecosystems, networks, or habitats, etc.), and each attribute is given a weight according to its importance, and each of the attributes, according to its attribute domain, is assigned to one of seven common types, i.e., interval, upper limit, lower limit, weakly determined value, strongly determined value, no upper and lower limits (the bigger the better), and no upper and lower limits (the smaller the better); (2) data matrix is normalized corresponding to the types of attributes; (3) find the objects that do not meet their specified attribute intervals or values, disqualify these objects and remove them from object list, and the remaining objects are identified as the qualified; (4) re-normalize the data matrix for the qualified objects; (5) assess the qualified objects using various multi-attribute assessment methods, like TOPSIS, REVAWEA, SAWA, etc; (6) determine the final ranking of the qualified objects using Copeland method. Full Matlab codes and software of MATASS were given.

Keywords attributes; TOPSIS; REVAWEA; SAWA; Copeland; assessment; software; Matlab.

Computational Ecology and Software
ISSN 2220-721X
URL: <http://www.iaees.org/publications/journals/ces/online-version.asp>
RSS: <http://www.iaees.org/publications/journals/ces/rss.xml>
E-mail: ces@iaees.org
Editor-in-Chief: WenJun Zhang
Publisher: International Academy of Ecology and Environmental Sciences

1 Introduction

In the real world, we face many problems on multi-attribute assessment (von Neumann and Morgenstern, 1944; Navarrete et al., 1979; Chen, 1987). For example, there are some objects (e.g., ecosystems, habitats, networks, or organisms, etc.). We want to assess them based on some of their attributes (i.e., properties, or characteristics), and rank them from the best to the worst. These attributes are different in their allowable ranges of assigning values. Some attributes may take interval values, and some may take upper limit values, etc. Based on previous studies (Chen, 1987; Huang et al., 1988; Qi et al., 1990; Zhang et al., 1991a, 1991b; Zhang and Gu, 1996; Qi et al., 2002; Qi, 2003; Zhang, 2007, 2012, 2016; Ferrarini, 2012), we developed the

software for multi-attribute assessment problems, MATASS (Multi-Attribute Assessment System) in present study. Full Matlab codes and software of MATASS were given.

2 Methods

For a multi-attribute problem, suppose there are m attributes and n objects being assessed. The weights of m attributes are w_1, w_2, \dots, w_m , respectively, which meet $\sum w_i = 1$. The data matrix is $(x_{ij})_{n \times m}$, $i=1,2,\dots,n; j=1,2,\dots,m$. The procedures of the proposed algorithm are as follows.

I. Each of the m attributes, according to its attribute domain, is assigned to one of seven common types (1) interval value: Assigned value must be in an interval; (2) upper limit value: Assigned value should not greater than an upper limit; (3) lower limit value: Assigned value should not less than a lower limit; (4) weakly determined value: more closer a specified value the better; (5) strongly determined value: assigned value must be strictly equal to a specified value; (6) no upper and lower limits: for the assigned value, the bigger the better, and (7) no upper and lower limits: for the assigned value, the smaller the better.

II. Data matrix is normalized corresponding to the types of attributes. The normalized matrix is $(y_{ij})_{n \times m}$, where the greater y_{ij} is considered as the better, $i=1,2,\dots,n; j=1,2,\dots,m$. Corresponding to the above seven attribute types, the following seven normalization methods are used

(1) Interval value. Suppose the specified interval of the attribute j is $[a, b]$. Let $c=(a+b)/2$, then

$$y_{ij} = 1 - |x_{ij} - c| / |b - c| \quad \text{if } a \leq x_{ij} \leq b$$

$$y_{ij} = -\infty \quad \text{if } x_{ij} < a, x_{ij} > b$$

(2) Upper limit value. Suppose the specified upper limit of the attribute j is b , then

$$y_{ij} = (x_j^{\max} - x_{ij}) / (x_j^{\max} - x_j^{\min}) \quad \text{if } x_{ij} \leq b$$

$$y_{ij} = -\infty \quad \text{if } x_{ij} > b$$

(3) Lower limit value. Suppose the specified lower limit of the attribute j is a , then

$$y_{ij} = (x_{ij} - x_j^{\min}) / (x_j^{\max} - x_j^{\min}) \quad \text{if } x_{ij} \geq a$$

$$y_{ij} = -\infty \quad \text{if } x_{ij} < a$$

(4) Weakly determined value. Suppose the assigned value of the attribute j is better if it is closer to a specified value c , then

$$y_{ij} = 1 - |x_{ij} - c| / (x_j^{\max} + x_j^{\min} + c)$$

(5) Strongly determined value. Suppose the assigned value of the attribute j should be strictly equal to a specified value c , then

$$y_{ij} = 1 \quad \text{if } x_{ij} = c$$

$$y_{ij} = -\infty \quad \text{if } x_{ij} \neq c$$

(6) No upper and lower limits. For the assigned value of the attribute j , the bigger the better, then

$$y_{ij} = (x_{ij} - x_j^{\min}) / (x_j^{\max} - x_j^{\min})$$

(7) No upper and lower limits. For the assigned value of the attribute j , the smaller the better, then

$$y_{ij} = (x_j^{\max} - x_{ij}) / (x_j^{\max} - x_j^{\min})$$

where $x_j^{\max} = \max_i x_{ij}$, $x_j^{\min} = \min_i x_{ij}$, $i=1,2,\dots,n; j=1,2,\dots,m$.

After the above normalization procedure, $0 \leq y_{ij} \leq 1$ if the values fall in the specified scope, otherwise, $y_{ij} = -\infty$, $i=1,2,\dots,n; j=1,2,\dots,m$.

III. Based on the normalized matrix $(y_{ij})_{n \times m}$, find the objects and attributes that meet $y_{ij} = -\infty$, disqualify these objects and remove them from object list, and the remaining p objects ($p < m$) are identified as the qualified.

IV. Re-normalize the data matrix $(x_{ij})_{p \times m}$ for the qualified objects and obtain the re-standardized matrix $(y_{ij})_{p \times m}$.

V. Assess the qualified objects using various multi-attribute assessment methods, like TOPSIS, REVAWEA, SAWA, etc.

(1) Simple Additive Weighting Algorithm (SAWA; Chen, 1987; Zhang and Gu, 1996; Qi, 2003; Zhang, 2007). It assumes that there are linear relationships between attributes and they meet additive criterion. It is similar to Bayesian decision-making. First, compute the benefits of n objects

$$d_k = \sum_{i=1}^m w_i y_{ki} \quad k=1,2,\dots,p$$

Rank d_k following the descending order and obtain the priority ranking of all objects.

(2) Residual Value Weighting Algorithm (REVAWEA; Zhang and Gu, 1996; Qi, 2003; Zhang, 2012, 2016). It is proposed by Zhang and Gu (1996). The principle is to use the averaged residual benefit of each object being assessed against the remaining objects to determine the priority of every object. First, compute between-object benefit differences

$$z_{ij} = \sum w_i (y_{ik} - y_{jk}) \quad y_{ik} > x_{ik}, i, j=1,2,\dots,p; i \neq j$$

and every object's benefit

$$d_k = \sum_{\substack{j=1 \\ j \neq k}}^p z_{kj} / (p-1) \quad k=1,2,\dots,p$$

Rank d_k following the descending order and obtain the priority ranking of all objects.

(3) The Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS; Hwang and Yoon, 1981; Yoon, 1987; Chen, 1987; Qi et al., 1990; Zhang et al., 1991a, 1991b; Zhang and Gu, 1996; Qi, 2003; Ferrarini, 2012; Zhang, 2007, 2012, 2016). First, let

$$z_{ij} = w_j y_{ij} \quad i=1,2,\dots,p; j=1,2,\dots,m$$

and determine the ideal solution and negative ideal solution

$$(x_j^*) = (\max z_{i1} \quad \max z_{i2} \quad \dots \quad \max z_{im})$$

$$(x_j^-) = (\min z_{i1} \quad \min z_{i2} \quad \dots \quad \min z_{im})$$

Then compute the distances of every object to the ideal solution and negative ideal solution

$$s_i^* = \left[\sum_{j=1}^m (z_{ij} - x_j^*)^2 \right]^{1/2} \quad i=1,2,\dots,p$$

$$s_i^- = \left(\sum_{j=1}^m (z_{ij} - x_j^-)^2 \right)^{0.5} \quad i=1,2,\dots,p$$

Finally, compute the relative closeness (benefit) of every object to the ideal solution d_i

$$d_i = s_i^- / (s_i^- + s_i^*) \quad i=1,2,\dots,p$$

Rank d_i following the descending order and obtain the priority ranking of all objects.

(4) Revised Mix Preference Chart (MIXPREFER; Moody, 1983; Zhang et al., 1996; Qi et al., 2002). The original algorithm (Huang et al., 1988) did not consider the situation of no difference between the objects i and j . Zhang et al. (1996) revised the algorithm by including the situation. First, for the attribute k , construct matrix chart $Z^k = (z_{ij})_{p \times p}$, where

$$z_{ij}^k = 0 \quad \text{if no difference between the objects } i \text{ and } j$$

$$z_{ij}^k = 1 \quad \text{if the object } i \text{ is superior to } j$$

$$z_{ij}^k = -1 \quad \text{if the object } j \text{ is superior to } i$$

$$i, j=1,2,\dots,p; i \neq j; k=1,2,\dots,m$$

in which $z_{ji}^k = -z_{ij}^k, i \neq j$. Then, compute every object's benefit

$$d_i = \sum_j \sum_k w_k z_{ij}^k \quad i=1,2,\dots,p$$

Rank d_i following the descending order and obtain the priority ranking of all objects.

(5) Revised Ranking Method Based on Relative Position Estimate (RELRANK; Navarrete et al., 1979; Chen, 1987; Zhang et al., 1996). First, suppose p objects are $x_i, i=1,2,\dots,p$. $x_i > x_k$, $x_i < x_k$, and $x_i \sim x_k$, denote that x_i is superior to x_k , x_i is inferior to x_k , and x_i is not different from x_k . Let

$$w(x_i > x_k) = \sum_{x_i > x_k} w_j, \quad w(x_i < x_k) = \sum_{x_i < x_k} w_j, \quad w(x_i \sim x_k) = \sum_{x_i \sim x_k} w_j$$

Suppose $\alpha \geq 0$ (here, $\alpha=1$), and

$$A_\alpha(x_i > x_k) = (w(x_i > x_k) + \alpha w(x_i \sim x_k)) / (w(x_i < x_k) + \alpha w(x_i \sim x_k))$$

$$A^{\cdot}(x_i > x_k) = \min_{A_\alpha \geq 1} A_\alpha(x_i > x_k), \quad A^{\prime\prime}(x_i > x_k) = \max_{A_\alpha \leq 1} A_\alpha(x_i > x_k),$$

Then

$$x_i > x_k \quad \text{if } A_\alpha(x_i > x_k) > A^{\cdot}(x_i > x_k)$$

$$x_i < x_k \quad \text{if } A_\alpha(x_i > x_k) < A^{\prime\prime}(x_i > x_k)$$

$$x_i \sim x_k \quad \text{if } A^{\cdot}(x_i > x_k) \leq A_\alpha(x_i > x_k) \leq A^{\prime\prime}(x_i > x_k)$$

Define two sets covering all objects

$$P(x_i) = \{x_k | x_k \neq x_i, x_i > x_k, \text{ not } x_k > x_i\}, \quad Q(x_i) = \{x_k | x_k \neq x_i, x_k > x_i, \text{ not } x_i > x_k\}$$

Suppose $p(x_i)$ and $q(x_i)$ are the number of elements in $P(x_i)$ and $Q(x_i)$ respectively. Every object's benefit is

$$d_i = p(x_i) - q(x_i) \quad i=1,2,\dots,p$$

Rank d_i following the descending order and obtain the priority ranking of all objects.

In the original algorithm (Chen, 1987), A^{\cdot} and $A^{\prime\prime}$ were arbitrarily specified, which cannot guarantee the symmetry of superiority vs. inferiority relationship. Zhang et al. (1996) revised the algorithm. We used $\min A_\alpha(x_i > x_k)$ and $\max A_\alpha(x_i > x_k)$ to replace A^{\cdot} and $A^{\prime\prime}$, respectively, and thus if $x_i > x_k$, then $x_k < x_i$.

VI. Determine the final ranking of the qualified objects using Copeland method. The principle of Copeland method is that, for any two objects i and j , compute the numbers of the object i being superior and inferior to the object j in every assessment method. If the number of being superior is larger than that of being inferior, then the object i wins j . Otherwise i loses to j . We assume that the object i loses to itself. Comparing i and j , and if the number of superiority is equal to that of inferiority, then both i and j lose. Finally, we obtain a win-lose matrix. Compute the winning number along rows and columns of the matrix respectively, and for every object, minus winning number of rows by that of columns to obtain the win values of the object. Rank win values following the descending order and obtain the priority ranking of all objects (Chen, 1987; Qi, 2003; Zhang, 2007).

The following are Matlab codes for MATASS, MATASS.m

```
valmat=input('Input the excel file name of data matrix, e.g., raw.xls (for data matrix x). x=(xij)n×m, rows are objects being assessed, and columns are attributes: ','s');
x=xlsread(valmat);
m=size(x,2); n=size(x,1);
stand=input('Input the excel file name of attribute information, e.g., attri.xls: ','s');
disp('Each of the m attributes, according to its attribute domain, is classified as one of seven common types (1) interval value: Assigned value must be in an interval; (2) upper limit value: Assigned value should not greater than an upper limit; (3) lower limit value: Assigned value should not less than a lower limit; (4) weakly determined value: more closer a specified value the better; (5) strongly determined value: assigned value must be strictly equal to a specified value; (6) no upper and lower limits: for the assigned value, the bigger the better, and (7) no upper and lower limits: for the assigned value, the smaller the better. In the data file, the 1st row stores the weights of attributes, the 2nd row stores the types of attributes according to the specified IDs for attributes as listed in previous section. The 3rd and 4th rows store the permitted intervals of value-assigning for attributes, and correspond to 7 types of attributes, we have (1) lower limits for the 3rd row and upper limits for the 4th row; (2) zeros for the 3rd row and upper limits for the 4th row; (3) lower limits for the 3rd row and zeros for the 4th row; (4) the same weak-determined values for both the 3rd row and the 4th row; (5) the same strong-determined values for both the 3rd row and the 4th row; (6) ones for both the 3rd row and the 4th row, and (7) zeros for both the 3rd row and the 4th row.')
```

```
stand=xlsread(stand);
de=std(x);
```

```

if (sum(de==0)~=0)
disp('The following attributes are nonsense. Just remove them out of data matrix and attribute matrix and restart the algorithm.');
```

```

find(de);
pause;
end
y=zeros(n,m);
w=stand(1,:);
y=normalization(x,stand);
idunq=find(sum(y')===-Inf);
idqua=find(sum(y')~=-Inf);
s=size(idunq,2);
if (s>=1)
disp('Unqualified objects')
disp([idunq])
if (s<n)
disp('Qualified objects')
disp([idqua])
end
xx=x(idqua(1:n-s),:);
yy=normalization(xx,stand);
n=n-s;
else
disp(' Qualified objects being assessed')
disp([idqua])
xx=x; yy=y;
end
rnk=zeros(5,n);
for i=1:5
switch i
case 1
d=sawa(w,yy); rp=sortrows([(1:n)' d'],-2); rnk(1,:)=rp(:,1);
rr=sortrows([idqua' d'],-2);
fprintf('SAWA Ranking\n Ranking Object Benefit\n');
rkk=[(1:n)' rr];
disp([rkk])
case 2
d=revawea(w,yy); rp=sortrows([(1:n)' d'],-2); rnk(2,:)=rp(:,1);
rr=sortrows([idqua' d'],-2);
fprintf('REVAWEA Ranking\n Ranking Object Benefit\n');
rkk=[(1:n)' rr];
disp([rkk])
case 3
d=topsis(w,yy); rp=sortrows([(1:n)' d'],-2); rnk(3,:)=rp(:,1);
rr=sortrows([idqua' d'],-2);
fprintf('TOPSIS Ranking\n Ranking Object Benefit\n');
rkk=[(1:n)' rr];
disp([rkk])
case 4
d=mixprefer(w,yy); rp=sortrows([(1:n)' d'],-2); rnk(4,:)=rp(:,1);
rr=sortrows([idqua' d'],-2);
fprintf('MIXPREFER Ranking\n Ranking Object Benefit\n');
rkk=[(1:n)' rr];
disp([rkk])
case 5
d=relrank(w,yy); rp=sortrows([(1:n)' d'],-2); rnk(5,:)=rp(:,1);
rr=sortrows([idqua' d'],-2);
fprintf('REL RANK Ranking\n Ranking Object Benefit\n');
rkk=[(1:n)' rr];

```

```

        disp([rkk])
    end
end
rk=Copeland(rnk);
rr=sortrows([idqua' rk'],-2);
rank=[(1:n)' rr];
disp('Copeland Ranking')
disp('Ranking   Object   Win value')
disp([rank])

```

The following are Matlab functions for normalization and various assessment methods and Copeland method, normalization.m, sawa.m, revaweaa.m, topsis.m, mixprefer.m, relrank.m, and Copeland.m

```

function y=normalization(x,stand)
%x: the value matrix.
%stand: matrix for information of standards.
mmax=max(x);
mmin=min(x);
m=size(x,2); n=size(x,1);
y=x;
for i=1:m
k=stand(2,i);
for j=1:n
switch k
case 1
c=(stand(3,i)+stand(4,i))/2;
if ((x(j,i)>=stand(3,i)) & (x(j,i)<=stand(4,i))) y(j,i)=1-abs(x(j,i)-c)/(stand(4,i)-stand(3,i));
else y(j,i)=-Inf;
end
case 2
if (x(j,i)<=stand(4,i)) y(j,i)=(mmax(i)-x(j,i))/(mmax(i)-mmin(i));
else y(j,i)=- Inf;
end
case 3
if (x(j,i)>=stand(3,i)) y(j,i)=(x(j,i)-mmin(i))/(mmax(i)-mmin(i));
else y(j,i)=- Inf;
end
case 4
y(j,i)=1-abs(x(j,i)-stand(3,i))/(mmax(i)+mmin(i)+stand(3,i));
case 5
if (x(j,i)==stand(3,i)) y(j,i)=1;
else y(j,i)=- Inf;
end
case 6
y(j,i)=(x(j,i)-mmin(i))/(mmax(i)-mmin(i));
case 7
y(j,i)=(mmax(i)-x(j,i))/(mmax(i)-mmin(i));
end
end; end

```

```

function d=sawa(w,y)
%w is the weight vector, and y is normalized value matrix
m=size(y,2);
n=size(y,1);
for i=1:n
d(i)=sum(y(i,:).*w);
end

```

```

function d=revawea(w,y)
%w is the weight vector, and y is normalized value matrix
m=size(y,2);
n=size(y,1);
for i=1:n
for j=1:n
z(i,j)=0;
for k=1:m
if (y(i,k)>y(j,k)) z(i,j)=z(i,j)+w(k)*(y(i,k)-y(j,k)); end
end; end; end
for k=1:n
d(k)=0;
for j=1:n
if (j~=k) d(k)=d(k)+z(k,j)/(n-1); end;
end; end

```

```

function d=topsis(w,y)
%w is the weight vector, and y is normalized value matrix
m=size(y,2);
n=size(y,1);
for j=1:m
z(:,j)=w(j)*y(:,j);
end
xstar=max(y);
xminus=min(y);
for i=1:n
sstar(i)=sqrt(sum((z(i,:)-xstar).^2));
sminus(i)=sqrt(sum((z(i,:)-xminus).^2));
end
d=sminus./(sminus+sstar);

```

```

function d=mixprefer(w,y)
%w is the weight vector, and y is normalized value matrix
m=size(y,2);
n=size(y,1);
d=zeros(1,n);
yy=zeros(m,n,n);
s=zeros(n,n);
for j=1:n
for k=1:n
for i=1:m
if (y(j,i)>y(k,i)) yy(i,j,k)=1;
elseif (y(j,i)<y(k,i)) yy(i,j,k)=-1;
else yy(i,j,k)=0;
end;
s(j,k)=s(j,k)+yy(i,j,k)*w(i);
end; end; end
for i=1:n
d(i)=0;
for j=1:n
d(i)=d(i)+s(i,j);
end; end

```

```

function d=relrank(w,y)
%w is the weight vector, and y is normalized value matrix
m=size(y,2);
n=size(y,1);

```

```

p=zeros(1,n); q=zeros(1,n);
s=zeros(n,n); t=zeros(n,n); r=zeros(n,n);
u=zeros(n,n); yy=zeros(m,n,n);
alpha=1;
for j=1:n
for k=1:n
for i=1:m
if (y(j,i)>y(k,i)) yy(i,j,k)=1;
elseif (y(j,i)<y(k,i)) yy(i,j,k)=-1;
else yy(i,j,k)=0;
end; end
end; end
for i=1:n
for j=1:n
for k=1:m
if (yy(k,i,j)==1) t(i,j)=t(i,j)+w(k); end
if (yy(k,i,j)==-1) s(i,j)=s(i,j)+w(k); end
if (yy(k,i,j)==0) r(i,j)=r(i,j)+w(k); end
end
if ((s(i,j)+alpha*r(i,j))==0) u(i,j)=1; continue; end
a(i,j)=(t(i,j)+alpha*r(i,j))/(s(i,j)+alpha*r(i,j));
end; end
a1=1e+50;
a2=-1e+50;
for j=1:n
for i=1:n
if (i==j) continue; end
if ((a(j,i)>=1) & (a(j,i)<a1)) a1=a(j,i); end
if ((a(j,i)<=1) & (a(j,i)>a2)) a2=a(j,i); end
end; end
for j=1:n
for i=1:n
if (i==j) continue; end
if ((a(j,i)>a1) | (u(j,i)==1)) p(j)=p(j)+1; end
if (a(j,i)<a2) q(j)=q(j)+1; end
end
d(j)=p(j)-q(j);
end

```

```
function v=Copeland(rnk)
```

```
%rnk(i,1:n): rankings; i=1,2,..., mean the rankings from SAWA, REVAWEA, TOPSIS, MixPrefer, RelRank, etc.
```

```
%v: rankings based efficacy of Copeland method.
```

```

m=size(rnk,1);
n=size(rnk,2);
tem=zeros(n,n);
am=zeros(1,n);
v=zeros(1,n);
vv=zeros(1,n);
for i=1:n-1
for j=i+1:n
q1=0;q2=0;pr1=0;pr2=0;
for k=1:m
for kk=1:n
if (rnk(k,kk)==i) pr1=kk;
elseif (rnk(k,kk)==j) pr2=kk;
end
end
if (pr1<pr2) q1=q1+1; end

```



```

if (pr1>pr2) q2=q2+1; end
end
if (q1>q2) tem(i,j)=1; tem(j,i)=0;
elseif (q1<q2) tem(i,j)=0; tem(j,i)=1;
else tem(i,j)=0; tem(j,i)=0;
end
end; end
for i=1:n
am(i)=0;
for j=1:n
if (i==j) continue;
elseif (tem(i,j)==1) am(i)=am(i)+1;
end
end; end
for j=1:n
vv(j)=0;
for i=1:n
if (i==j) continue;
elseif (tem(i,j)==1) vv(j)=vv(j)+1;
end
end; end
for i=1:n
v(i)=am(i)-vv(i);
end

```

Also, the full Matlab codes listed above, an example data file, an attribute file, and an executable file package, as the supplementary material (Zhang-Supplementary-Material.rar) of the present article, can be downloaded at [http://www.iaees.org/publications/journals/ces/articles/2017-7\(2\)/2017-7\(2\).asp](http://www.iaees.org/publications/journals/ces/articles/2017-7(2)/2017-7(2).asp).

3 Application Example

Suppose there are 9 attributes and 10 objects. The data matrix is

2	2	2	3	4	9	2	12	2
2	4	2	13	7	3	2	4	3
6	2	3	2	2	6	6	2	6
11	6	6	5	5	4	3	6	1
5	2	2	7	2	4	2	2	2
2	6	2	7	2	2	4	3	2
8	3	2	2	3	4	2	8	3
5	4	10	5	2	2	3	10	7
2	7	2	3	5	3	4	10	3
2	9	5	11	3	3	2	2	4

and the matrix for attribute information is

0.15	0.1	0.15	0.1	0.1	0.15	0.1	0.05	0.1
6	7	6	1	3	2	4	6	7
1	0	1	2	1	0	5	1	0
1	0	1	10	0	8	5	1	0

in which the 1st row stores the weights of attributes, the 2nd row stores the types of attributes according to the specified IDs for attributes as listed in previous section. The 3rd and 4th rows store the permitted intervals of value-assigning for attributes, and correspond to 7 types of attributes, we have (1) lower limits for the 3rd row

and upper limits for the 4th row; (2) zeros for the 3rd row and upper limits for the 4th row; (3) lower limits for the 3rd row and zeros for the 4th row; (4) the same weak-determined values for both the 3rd row and the 4th row; (5) the same strong-determined values for both the 3rd row and the 4th row; (6) ones for both the 3rd row and the 4th row, and (7) zeros for both the 3rd row and the 4th row.

Table 1 Assessment results of various methods.

Ranking	SAWA		REVAWEA		TOPSIS		Revised Mix Preference Chart		Revised Plan Ranking based on Estimation of Relative Ranking		Copeland	
	Object	Benefit	Object	Benefit	Object	Benefit	Object	Benefit	Object	Benefit	Object	Win
1	4	0.7171	4	0.337	4	0.236	4	2.2	8	6	4	6
2	8	0.6321	8	0.293	7	0.234	8	1.05	4	4	8	4
3	7	0.5194	9	0.177	8	0.234	6	-0.15	6	1	9	2
4	9	0.484	7	0.166	9	0.229	9	-0.65	5	-2	7	0
5	5	0.4728	5	0.13	5	0.228	7	-0.75	9	-2	5	-2
6	6	0.4394	6	0.129	3	0.225	3	-0.85	3	-3	6	-4
7	3	0.3444	3	0.104	6	0.225	5	-0.85	7	-4	3	-6

The Copeland results show that the objects 1, 2, and 10 are unqualified and the objects 3, 4, 5, 6, 7, 8, and 9 are qualified. The objects 4, 8 and 9 are the most qualified objects (Table 1).

Acknowledgment

I am thankful to the support of Discovery and Crucial Node Analysis of Important Biological and Social Networks (2015.6-2020.6), from Yangling Institute of Modern Agricultural Standardization, High-Quality Textbook *Network Biology* Project for Engineering of Teaching Quality and Teaching Reform of Undergraduate Universities of Guangdong Province (2015.6-2018.6), from Department of Education of Guangdong Province, China.

References

- Chen T. 1987. Decision-making Analysis. Science Press, Beijing, China
- Ferrarini A. 2012. WORTHY: a new model for ecological ranking and evaluation. *Computational Ecology and Software*, 2(4): 213-219
- Hwang CL, Yoon K. 1981. Multiple Attribute Decision Making: Methods and Applications. Springer-Verlag, New York, USA
- Huang MF, Zhang H, Yuan W. 1988. New Practical Techniques for Decision-making. Guizhou People's Press, Guiyang, China
- Moody PE. 1983. Decision Making Proven Methods for Better Decisions. McGraw-Hill, New York, USA
- Qi YH. 2002. An improved algorithm and network computing software of Mix Preference Map and its application in evaluation of books and journals. *Modern Computer*, 11: 14-16, 56
- Qi YH. 2003. An internet based computational software for assessment analysis of books and journals. *Journal of Information*, 22(1): 43-45
- Qi YH, Wang DX, Zhang WJ. 1990. Analysis of resistance attributes of wheat for *Fusarium graminearum*

- Schw. and identification in batches using computers. *ACTA PHYTOPATHOLOGICA SINICA*, 20(2): 147-152
- Hwang CL, Yoon K. 1981. *Multiple Attribute Decision Making: Methods and Applications*. Springer-Verlag, New York, USA
- Navarrete Jr N, Fukushima M, Mine H. 1979. A new ranking method based on relative position estimate and its extensions. *IEEE Transactions on Systems, Man, and Cybernetics*, 9: 681
- von Neumann J, Morgenstern O. 1944. *The Theory of Games and Economic Behavior*. Princeton University Press, New Jersey, USA
- Yoon K. 1987. A reconciliation among discrete compromise situations. *Journal of Operational Research Society*, 38: 277–286
- Zhang WJ. 2007. *Methodology on Ecological Research*. Sun Yat-sen University Press, Guangzhou, China
- Zhang WJ. 2012. *Computational Ecology: Graphs, Networks and Agent-based Modeling*. World Scientific, Singapore
- Zhang WJ. 2016. *Selforganizology: The Science of Self-Organization*. World Scientific, Singapore
- Zhang WJ, Gu DX. 1996. Kind of algorithm for stochastic decision-making in insect pest management. *Bulletin of Science and Technology*, 12(5): 288-293
- Zhang WJ, Li ZQ, Qi YH. 1991a. A study on appraising disease resistance of wheat using MOD Theory. *ACTA PHYTOPATHOLOGICA SINICA*, 21(3): 223-228
- Zhang WJ, Qi YH, Gu DX. 1996. Objective system and benefit evaluation for microbial control of pests. *Journal of Biomathematics*, 11(5): 43-51
- Zhang WJ, Qi YH, Wang DX. 1991b. Algorithm of drought resistance attributes and evaluation of its decision-making in wheat seedling stage. *Agricultural Research in the Arid Areas*, 2: 100-105