Pair comparison is a measurement method that has been used in two quite different spheres, for similar purposes.

- One is, or perhaps was, as part of the Participatory Rural Appraisal (PRA) package of methods. Here it has been used to generate preference rankings (Example from PRA Notes)
- The other is a voting method (Explanation), which is also about the expressions of preferences. Different voting methods can generate different outcomes, depending on how winning candidate is identified. Pairwise comparison satisfies the Condorcet criteria of fairness

Pairwise comparisons can be used to generate scores for a diverse set of items on a common scale, where none was self evident before. In its simplest form this is an ordinal (ranking) scale, but more sophisticated interval and ratio scales can also be produced.

People usually find pair comparison an easier task than ranking a larger set of items at the same time. It also easier to identify the qualitative differences that people see between the items involved. The more items there are to be compared, the harder the task usually is. However pair comparison can be very time consuming when there are large numbers of items that need to be compared.

(by Judy Horocek, from The Age, July 2014)

One area where pair comparisons could be very useful is in the evaluation of portfolios of activities. These typically have a common theme e.g. improved governance, but no common metric by which all their outcomes can be compared. For example, good governance may be an overall objective, but in practice it can have many different dimensions. This lack of a common basis for comparison is a major obstacle to any attempt at systematic analysis of "what works".

How could pair comparisons be used? In this context we don't want to identify a preference ranking, instead we want to identify a performance ranking. This means that identifying the criteria behind each choice will probably be more important because the final ranking will need to be as transparent as possible, so it can be subject to, and stand up to, careful scrutiny.

The challenge then is to work out the best method of doing so.

The following algorithm is suggested as an approach that should be tested out to see how well it works. There may be more complex and more rigorous methods, but simplicity is arguably as important as rigour. Without it, the most rigorous method may lay unused.

1. Compare one pair of items at a time (e.g. individual projects A and B)
1.1 Identify and list each way in which A>B in terms of its performance
[See rows 3-5, columns A - C in the table below]
1.2 Identify and list each way in which B>A in terms of its performance
[See rows 6-9, columns A - C in the table below]
1.3 Rank all the listed ways (i.e. performance criteria), in terms of their overall importance 1= Most important, n= Least important)
[See column D in the table below]
1.4 Add the rank values for A, where A>B on each criteria
[See column E in the table below]
1.5 Add the rank values for B, where B>A on each criteria
[See column F in the table below]
1.6 Convert both totals into percentages of sum of all rank values (1+2+3=...n)
[See rows 11-12 in the table below]

	A	B	C	D	E	F
1					Weighted scores	
2		A>B	B>A	Ranked criteria	A	B
3	Criteria 1	0	1	6	6	0
4	Criteria 2	0	1	3	3	0
5	Criteria 3	0	1	5	5	0
6	Criteria 4	1	0	3	0	3
7	Criteria 5	1	0	2	0	2
8	Criteria 6	1	0	1	0	1
9	Criteria 7	1	0	4	0	4
10						
11	Totals			24	14	10
12	Totals converted to 1- % of maximum possible				**42%**	**58%**
13						

1.7 Enter each of these percentage values minus 100 as the pair comparison results for A and B in a table like the one below. (minus 100 in order to make high percentages = high rank position)
1.8 Repeat who process until all possible comparisons have been made and the results table is full, like the one below.
1.9 Calculate the average score for each item, as in the rightmost column below.

This is their **overall performance measure on a common measurement scale**.

2. Collate a list of all performance criteria used in all comparisons
[See column A in the table below]
2.1 Calculate the number of all pair comparisons where each criteria was used.
[See column G in the table below]
2.2 Calculate the average ranking given on each criteria
[See column F in the table below]
2.3 Multiply the average ranking (2.2) by percentage (2.1)
[See column H in the table below]
2.4 Convert the result (2.3) in to a percentage of the maximum possible value
[See column I in the table below]

These percentages are in effect **the weighting given to each performance criteria** in the whole exercise.

Postscript: One could get a bit carried away and then use UCINET software to (a) take a pair comparison x criteria matrix (where cells say which criteria were used in which comparisons) then (b) convert it into a criteria x criteria matrix (where cells values = number of times the row and column criteria were used in the same comparison), and then (c) use NETDRAW software to visualise the results, to identify if there any clusters of criteria that were often used together. If so, these clusters would be in effect a typology of the types of projects in the portfolio.

Postscript 2014 08 03: Tom Thomas of Praxis (India) has commented "The pairwise ranking used to be one of the earliest tools and was subsequently abandoned due the the time it took to do these comparisons and people loosing people's attention due to the repetitive nature of this enquiry." This is true, it does take time. The application I have in mind is with program managers in an evaluation context, where the time requirement might be understood and tolerated. That said, I would not want to do the exercise with more than 10, which could involve up to 45 comparisons.

Postscript 2014 08 31: Readers may be interested in LineUp, an OpenSource program for analysis and visualisation of multi-criteria ranking data

Postscript 2014 09 03: Steve Powell has raised an interesting question of whether there might be a more efficient pair comparison process than the exhaustive process outlined above. My interpretation of what he has proposed is as follows:
1. Arrange items in a list, randomly at first
2. Compare any two adjacent items, and place the most successful above the other
3. Repeat with any other pairs of adjacent items not yet compared
[This will result in new pairs of adjacent items, that were not compared before, but now need to be compared]
4. Continue until adjacent pair comparisons cease to generate any change in item positions
[This process may lead to what is called a local rather than global optimum solution]
5. Optionally, make a few pair comparisons of non-adjacent items, to see if the current ranking is only a local optimum. If this induces further changes, go back to Step 4 above.

Note: The process for eliciting performance criteria and then aggregate judgements about which item in a pair comparison is most successful could remain the same as outlined above (Steps 2.1-4)