<u>Making a Decision Using Contest Analysis</u>

Suppose one needed to make a decision which involves several criteria. As an example, we can use cost, time, and quality for three criteria. The decision is a choice between or among two or more options. With two options (for example) and our three criteria, we can set up a scoreboard with 5 participants, numbering them 1 to 3, for the criteria, and 4 to 5, for the options. The entries in the top three rows will be preferences of criteria for options and these can be shown with the symbol A_{ij}, where the first index is for the row and the second is for the column. The scoreboard will look like this:

0	0	0	A_{14}	A_{15}
0	0	0	A_{24}	A_{25}
0	0	0	A_{34}	A_{35}
A_{41}	A_{42}	A_{43}	0	0
A_{51}	A_{52}	A_{53}	0	0

The upper-right part gives the preferences of the criteria for the two options, and the two preferences of each row sum to unity. In the lower-left section, the same preference numbers are used, but with rows and columns interchanged, and the rows do not have to sum to unity. (This simple arrangement no longer applies when the problem becomes more complicated than what we have here so far.)

We will be calculating the weights of criteria and options from the scoreboard. The weight of the criterion depends upon its preferences for the options and the weights of the options. The weight of the option depends upon the preference the criterion has *for it* and the weight of the criterion. (More "weighty" criteria will convey more weight to the option.) If there is an *a priori* weight for the criterion--say if cost will be more important than time--that number can be used as a multiplier in the bottom-left part of the matrix. (Discussion on this point will be given below.)

The contest algorithm computes the weights of the five participants with five coupled equations. The equations all have the same pattern; the first one is written here:

$$(X_1)^2 = [(A_{14})^2/(A_{14} + A_{24} + A_{34})]\ X_4 + [(A_{15})^2/(A_{15} + A_{25} + A_{35})]\ X_5$$

The derivation of this equation is given elsewhere. Solving for the X values is done by iteration. The X value, for participant i, is the magnitude of a vector possessed by participant i which points in the directions of other participants. The components of this vector depend upon the preferences that the other participants have for participant i--so a larger X means more overall preference for a participant. The 'direction' of a participant is the axis in an n-dimensional space which is given to that participant, where n is the number of contestants in the problem.

What we have set up here is a judged contest, similar to a beauty contest, in which the judges are the criteria and they pick a winner from among the options. The options interact with each other only indirectly via the criteria. As an example let us use preference numbers as shown on the scoreboard below.

0	0	0	.3	.7
0	0	0	.1	.9
0	0	0	.5	.5
.3	.1	.5	0	0
.7	.9	.5	0	0

Using the contest analysis algorithm, we get weights for the five participants. After the weights are computed, the algorithm gives the "applied strengths" of the participants. The weight is the magnitude of a vector (in a space of 5 dimensions, for this case), and the participants' vectors have to be formed into a vector sum, representing the "strength vector" of the contest as a whole. Each participant makes a contribution to the magnitude of the contest strength vector, and that contribution is called the "applied strength" of the participant. (It is a scalar quantity, a fraction of the magnitude of the contest vector.) The results for the above example are shown below, where the applied strengths are normalized to unity.

Participant	Weight (X)	Applied Strength
1	.507	0.198
2	.601	0.224
3	.474	0.172
4	.412	0.115
5	.924	0.291

Notice that the second criterion is strongest because it puts more preference on the stronger option. The second option (participant #5) is expected to be stronger because of the preference it received, but it gets an extra boost from the weights of the criteria that vote for it. If we were using preference scores alone, the ratio of the two options would be 0.43. Weighting the judges, as we did, gave a ratio of applied strengths equal to 0.40, an amplification of the difference in their strengths.

When the scoreboard entry is multiplied by the computed weight of the participant scored against, we have the *weighted score*, and the scoreboard can be rewritten with weighted scores as entries. I like to use the terms *attacker* and *defender* for the participants doing the scoring and being scored against. Then the relative applied strength of an attacker turns out to be the sum of his weighted scores against his defenders divided by the sum of all weighted scores in the contest. (The terms attacker and defender are carried over from combat problems and are misnomers here, where strength is given by support from other participants, rather than by defeat of other ones. But the same mathematical procedure works in either case.)

Now suppose one of the criteria of the example were indefinite, subject to a judgment call. Say that the criteria were cost, time, and quality and that cost and time could be given definite numerical values for the two options, but that quality was judgmental. Then we would have a separate judged contest to evaluate for quality, and let us have four judges for that. Each judge will have a vote, to be split as he sees fit for the quality of the two decision options. This is a sub-contest with a scoreboard looking like this:

0	0	0	0	.2	.8
0	0	0	0	.4	.6
0	0	0	0	.5	.5
0	0	0	0	.9	.1
.2	.4	.5	.9	0	0
.8	.6	.5	.1	0	0

The upper four rows give the scores of the judges for the two options and the lower two rows give the scores received by the options. The scores received are equal (row sums 5 and 6), so the raw votes would say that the original 50-50 split on quality is correct. But the judges are quite variable in their voting and we need to assign weights to them and process this sub-contest with another run of the algorithm. Here we number the judges 1-4 and the options 5-6. The results are as follows:

Participant	Weight (X)	Applied Strength
1	.516	0.152
2	.454	0.153
3	.448	0.154
4	.580	0.157
5	.822	0.198
6	.782	0.186

The concentration of the fourth judge on the first option gives gives that judge and that option a little more emphasis, with the options now getting a proportion of 0.52 to 0.48 (in applied strength) instead of the original 0.5 and 0.5. Running the original scoreboard with this slightly more favorable scoring for the first option will give a result showing a bit more overall preference for that option than it had originally.

It is clear now how a complicated decision can be processed as an overall contest with several sub-contests. There can be as many sub-contests as the complexity of the decision dictates. We can extend the above example further by supposing that the three original criteria may not be given equal importance, and that a set of judges should decide the relative weighting. Here we would set up another scoreboard for a judged contest--any number of judges. The application of the contest algorithm to this scoreboard would be similar to what was done for the 'quality' scoreboard just above. The result of this step would give importance numbers for each criterion column of the original scoreboard (lower-left section, multiplying each column by relative criterion importance).

When the original 'master' scoreboard has been adjusted to account for all the judgmental and factual data going into the decision, it can be run with the contest algorithm to yield numerical values for the levels of superiority and inferiority among the options. As a side benefit, similar results will be available for the criteria.