Prioritizers and a New Limit Matrix Calculation

Bill Adams
Decision Lens Inc
DeLand FL, USA
wjadams@decisionlens.com

March 1, 2011

Abstract

Current ANP theory has a mixture of pairwise, direct data, ratings, and subnetworks used at specified places throughout it. A unifying concept of a prioritizer is put forward in this paper to combine these all under one heading. In addition, a change to the limit matrix calculation is suggested that aids in handling the new prioritizer underpinning of ANP models.

1 Introduction

ANP theory, as described in [1], has evolved over the past few decades to the current structures. Originally there were only ANP limit matrix calculations involved. Later, a ratings structure was added to the theory as well as the ability to utilize subnetworks. As these new ideas were introduced the associated calculations were bolted on to the existing ANP calculations. Thus, as the theory currently stands a full ANP model with ratings and subnetworks goes through the following steps in calculations.

1. Calculate local priorities from either direct data or pairwise data and fill in the unscaled and scaled supermatrices.
2. Calculate the limit matrix and the global priorities vector.
3. Use the global priorities and ratings data to calculate total ratings scores.
4. Use those scores to feed upward any alternative values in subnetworks, using the appropriate synthesis formula.

In this paper, we propose a relatively simple change to both the structure of ANP models, and the calculations performed on them to address the following difficulties.

Calculation issues: In the previous outline of a full ANP model's calculation, one notices that the last two calculations happen after the most interesting ANP calculation has already happened (namely the limit matrix calculation). That is, any rating data we put into the model cannot be used in any feedback purpose, and likewise for subnetwork data. It is as if, after we have done the difficult problem of handling feedback, we have bolted on two additional calculations.

BOCR can only be a tree: What we mean by this is the following. Consider a typical model with BOCR and Social, Political, Economic underneath. Essentially we have a tree structure with a goal node, BOCR directly underneath, and Social, Political, and Economic beneath each of the BOCR nodes respectively (and finally the alternatives under those). There is no way in this setup to have, for instance, Social Benefits interact with Social Costs in any sort of feedback.

Complicated view of a full model: If one looks at a typical ANP model with subnetworks in SuperDecisions, one only sees the top level with BOCR in it. One has no immediate idea of what alternatives are involved in the decision, or even if a Social, Political, Economic level is underneath BOCR without drilling down into each model. Granted, this is partly a user interface issue. However, that issue arises from the way subnetworks are handled in a standard ANP model.

Ad hoc: What we mean here is the ad hoc nature of where one can put subnetworks and where one can rate. It would be nice if we could rate at any place it seems convenient for the model at hand (and likewise for subnetworks).

2 Prioritizer Definition

We propose the following definition to unify the various concepts of pairwise data, direct data, ratings data, optimization (in the ANP context), and even ANP models themselves.

Definition 1 (prioritizer)**.** *A **prioritizer** is something that assigns priorities (numerical values) to a list of **alternatives**.*

Note 1. The term **alternatives** here simply means a collection of things we wish to have prioritized. We do realize there is a bit of conflict with the notion of alternatives in an AHP or ANP model. However, one may think of ANP and AHP models as prioritizers and the **prioritizer alternatives** of the ANP/AHP model are precisely the regular old alternatives of ANP/AHP model as defined in [1].

In order to see how the concept of a prioritizer unifies the various ideas of ratings, pairwise, etc, let us go through each case and describe them from the prioritizer perspective.

Direct: Let us say we are assigning direct data priorities to a collection of nodes in an ANP model (with respect to some other node). For specificity, let us call them Crit1, Crit2, and Crit3, and further specify the values of 0.3, 0.6, and 0.1 respectively for their direct data. We can view this information as a prioritizer. That prioritizer would have three alternatives, Crit1, Crit2, and Crit3. The numerical values they are assigned would be 0.3, 0.6, and 0.1 respectively.

Pairwise: Let us say again we are pairwise comparing three nodes in an ANP model, Crit1, Crit2, and Crit3, with a pairwise comparison matrix of

$$\begin{bmatrix} 1 & 3 & 6 \\ \frac{1}{3} & 1 & 2 \\ \frac{1}{6} & \frac{1}{2} & 1 \end{bmatrix}.$$

The priorities resulting from this (normalized) would be

$$\begin{bmatrix} 0.1 & 0.3 & 0.6 \end{bmatrix}.$$

This combined data can be thought of as a prioritizer with three alternatives (Crit1, Crit2, and Crit3) giving priorities of $(0.1, 0.3, 0.6)$.

Ratings: To understand how ratings fits into prioritization let us assume we have 3 alternatives (Alt1, Alt2, Alt3) that we are rating with respective to a particular criteria (Crit1). Assume further that the ratings scores for them are $.95, .75, .25$ (notice that these are ideal scores, not normalized scores, we will discuss this difference in prioritizers later). This collection of data makes up a prioritizer (the prioritizer which prioritizes Alt1, Alt2, and Alt3 with respect to Crit1).

ANP Model: In any ANP model there are always a collection of alternatives. Although there may be many criteria, subcriteria, subnetworks, ratings systems, direct data, etc, the real point of the ANP model is to prioritize its alternatives (and as such it is a prioritizer).

Note 2. Although a prioritizer's end result is to give priorities to its alternatives, it may have significant amounts of other input data. In the case of pairwise prioritizers, there are the input pairwise comparison matrices. For ratings there are the ratings of each alternative, plus numerical information about how much each rating is worth. For ANP models, there are the nodes and clusters (as well as the alternatives cluster) combined with the connections and all of the data that eventually goes into the supermatrices (and perhaps subnetworks as well). Nonetheless every prioritizer has as its "end goal", the prioritization of its alternatives.

Note 3. The prioritizers above sometimes return priorities that are normalized (pairwise does this) and sometimes return ideal priorities (ratings does this). To accommodate these varieties, any prioritizer should be able to return priorities in the following formats.

Normalized: These are the priorities scaled so that they sum to one.

Idealized: These are the priorities scaled so that the largest value is less than or equal to one. For a ratings prioritizer, these are the values it returns by default. For pairwise prioritizers, it would idealize the normalized priorities.

Raw: These are the priorities in the natural state for the given prioritizer. For ratings they would be the ideal scores calculated. For pairwise they would be the normalized scores.

3 Prioritizers in ANP Models

With the definition of a prioritizer we can simplify tremendously the data which goes into an ANP model. At any point in an ANP model one would normally pairwise, or do direct data, or do ratings, instead we now simply say there is a prioritizer there. Let us look at the individual places where pairwise, direct data, ratings and subnetworks are used in a traditional ANP model and how they are effected by using prioritizers.

Node → Cluster connection: In traditional ANP models there would either be pairwise data here, or direct data. Now there is merely a prioritizer that prioritizes the nodes in the given cluster that are connected from the given node. That prioritizer may be a pairwise prioritizer, or a direct data one. However it could also be a ratings prioritizer, or even a subnetwork prioritizer (see section 4 for more information about how subnetworks work).

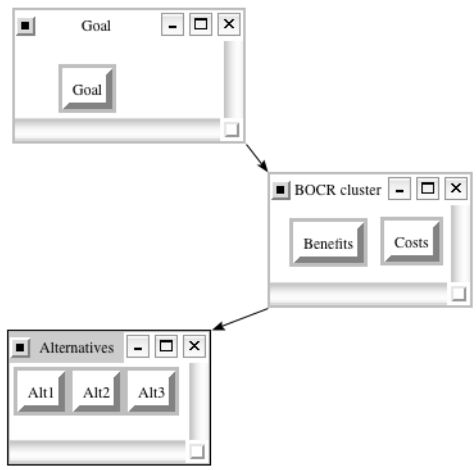
Cluster priorities: In traditional ANP models there would be pairwise (or perhaps direct) data to give priorities for other clusters connected from a given cluster. Now this cluster prioritization is handled by a prioritizer.

Ratings scales: In a traditional ANP model there would either be direct data or pairwise data to signify how important the ratings scale components are (the Hi/Med/Low's). Now there is a prioritizer who's alternatives are those scale components.

Subnetworks: In a traditional ANP model, subnetworks are attached to nodes, and alternative scores feed up through that node. In this new structure things happen differently. Instead subnetworks are attached to connections from a node to a cluster (where a pairwise or direct prioritizer might normally have been). See section 4 for a detailed explanation of how this works.

4 ANP Subnetworks Using Prioritizers

As mentioned above, subnetworks are handled slightly differently using this prioritizer perspective. In particular subnetworks are no longer attached to nodes, but rather to connections from a node to a cluster. It is easiest to see this via an example. Let us say we have a 2 level BOCR model with only Benefits and Costs. In the new structure our top-level model would look like the following



There would be subnetworks in each of the following locations in the situation pictured above.

Goal → BOCR cluster: Given the connection from the Goal node to the BOCR cluster, there needs to be a prioritizer prioritizing Benefits and Costs with respect to the Goal. This prioritizer would be an ANP model whose alternatives would be Benefits and Costs. That ANP model would certainly have several other clusters with many other nodes perhaps. However, it's sole purpose is to tell us how important Benefits and Costs are for the model we have pictured above.

Benefits → Alternatives cluster: Given the connection from the Benefits node to the Alternatives cluster there needs to be a prioritizer prioritizing Alt1, Alt2, and Alt3 with respect to Benefits. This prioritizer would be an ANP model whose alternatives are Alt1, Alt2, and Alt3 (and the resulting values from synthesizing said network would tell us how important Alt1, Alt2, and Alt3 are with respect to the goal node).

Costs → Alternatives cluster: Similarly for this connection.

Note 4. In the above top-level picture for our model we can easily see the overall structure (that we have broken our decision down into Benefits and Costs) and we additionally see immediately what our alternatives are (in a traditional ANP model one has to drill down into the bottom level subnetworks to see the alternatives).

Note 5. If we wished to add a Social and Political level to the above picture, we would simply add another cluster for Social and Political nodes [1], and wedge

[1] Perhaps with separate Social and Political nodes for Benefits and Costs.

them between the BOCR cluster and the alternatives. There would be subnetworks for the connection $Social \to Alternatives$ and $Political \to Alternatives$ under each of the Benefits and Costs. This would give us the equivalent of a traditional three level ANP model (two levels of subnetworks underneath the top level model) but we have only two levels now.

5 Limit Matrix Calculations in ANP Models

With prioritizers attached to connections from a node to a cluster (or cluster to a cluster) and in particular subnetworks attached there, synthesizing with BOCR formulas becomes more difficult. This is because the synthesis, including the inversion of Costs and Risks, needs to happen during the limit matrix calculation.

The remedy is to change the "multiplication" step of calculating powers of the scaled supermatrix. Instead we perform a modified matrix multiplication which takes into account the inversion at the Risks and Costs nodes. Of course, this is applied generically (we do not key in on the words Risks and Costs, for instance). Instead, some extra information is included along with every node and cluster prioritizer in an ANP model. The two extra pieces of data are:

Synthesizer: A synthesizer is a formula describing how to combine scores through the nodes that are prioritized. The standard synthesizer is the "multiply and add" synthesizer (which corresponds to standard matrix multiplication). The only other synthesizer we need at the moment is an exponentiate and multiply – which corresponds to a formula like

$$\frac{B^b O^o}{C^c R^r}.$$

Inverter: An inverter is associated to each alternative in the prioritizer in an ANP model. The empty inverter does nothing, it just feeds up scores. The probabilistic inverter inverts using the $(1 - x)$ formula. The negative inverter inverts by doing $-x$. There are also multiplicative inverters. In models from the current ANP theory, every node except the Costs and Risks node would get the empty inverter. Depending upon which formula we wish to use we would set the inverter for Costs and Risks as appropriate.

The new limit matrix calculation then takes the scaled supermatrix and the data of synthesizers and inverters and uses that to do a modified matrix multiplication to arrive at the "powers" that we then check for convergence and cycling.

6 Example Calculation

In order to see how this calculation works in practice, consider an ANP BOCR model with a Goal node connecting down to Benefits and Costs, which connect to the alternatives Alt1 and Alt2. Assume we have subnetworks for the $Goal \rightarrow BOCR$ connection and $Benefits \rightarrow Alternatives$ and $Costs \rightarrow Alternatives$. Further assume we have weights of $Benefits = 0.6$ and $Costs = 0.4$ with respect to Goal (from that subnetwork), $Alt1 = 0.8\, Alt2 = 0.2$ with respect to Benefits, and $Alt1 = 0.3\, Alt2 = 0.7$ with respect to Costs (both from the respective subnetworks). Then the supermatrix (both scaled and unscaled) is the following (with the nodes ordered Goal, Benefits, Costs, Alt1, Alt2).

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0.6 & 0 & 0 & 0 & 0 \\ 0.4 & 0 & 0 & 0 & 0 \\ 0 & 0.8 & 0.3 & 0 & 0 \\ 0 & 0.2 & 0.7 & 0 & 0 \end{bmatrix}$$

The only further pieces of information needed are the synthesizers and inverters to use. To mimic the $Bb+C(1-c)$ formula we use the standard multiply and add synthesizer and the probabilistic inverter on Costs. The only place we get a non-zero matrix multiplication column is in going through the Benefits and Costs to the Goal. The Benefits column and inverted Costs column are as follows.

$$B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.8 \\ 0.2 \end{bmatrix} \quad InvertedC = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1-0.3 \\ 1-0.7 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.7 \\ 0.3 \end{bmatrix}$$

The multiply and add synthesizer will give the Goal column the value of

$$NewGoalColumn = 0.6B + 0.4InvertedC = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.48+0.28 \\ 0.12+0.12 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.76 \\ 0.24 \end{bmatrix}$$

And thus after one level of multiplication the result is

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0.76 & 0 & 0 & 0 & 0 \\ 0.24 & 0 & 0 & 0 & 0 \end{bmatrix}$$

7 Conclusions

The concept of a prioritizer unifies the various ideas of pairwise, ratings, direct data, and ANP model under a single idea. One can then adjust the setup of an ANP model to take a prioritizer any place where one might use pairwise, direct, ratings, or subnetworks. This approach has several benefits. First, at any point in an ANP model where one needs priorities for a collection of objects, one is free to use the prioritizer which best suits the situation. Second, by recognizing subnetworks as attached to *node* $\rightarrow$ *cluster* connections, ANP models with subnetworks only ever go one subnetwork down from the top model (and the top level model shows a convenient overview of the model as it breaks down into BOCR and the alternatives [2]). Third, this new structuring of subnetworks allows for feedback that is impossible in a traditional ANP model with subnetworks (between Social Benefits and Social Risks for instance). Finally, with the concept of a prioritizer abstracted out of an ANP model, we are free to create new types of prioritizers [3].

References

[1] T.L. Saaty. *Decision Making with Dependence and Feedback The Analytic Network Process.* RWS Publications, Pittsburgh, 1996.

[2] In a traditional ANP model one has to drill down to the bottom level network to see what are the alternatives.

[3] For instance, one can envision a "Sorting Prioritizer". In such a prioritizer a user would sort the alternatives in a list, from best to worst. The sorting prioritizer would then apply a standard set of priorities to that list of sorted alternatives.