

Raw Relation Sets and Multiperson Decision Making

Yuxi Chen, Weiyi Meng

Department of Computer Science
SUNY at Binghamton, Binghamton, NY 13902
meng@cs.binghamton.edu

Abstract

In this paper, Multiperson Decision Making is explored based on Raw Relation Sets [1], in which three types of basic order relationships between any two objects are used to define the order relationships among a set of objects. Different order relationships among the same set of objects may be produced by different evaluators (or evaluation criteria) and there is also an order relationship among the evaluators themselves. In this paper, all evaluators are assumed to be of the same importance. This method emphasizes using qualitative (or un-weighted) order relationships but it is also capable of handling weighted order relationships.

1. Introduction

The following problem is frequently encountered in daily lives. A set of objects needs to be evaluated based on one or more criteria in order to rank them. For example, in Multiperson Decision Making [2], multiple people actively participate in arriving at a decision. Also, in the top- N query problem [5], multiple attributes serving as criteria are involved in a condition, we expect the DBMS to return the tuples that best satisfy the query condition, with better-matched tuples ranked ahead. There are many similar applications. Generally, the critical common issues are:

- (1) How to appropriately describe the orders on the same set of objects for different evaluators/criteria?
- (2) How to aggregate these orders into a combined order while taking into consideration the fact that different evaluators/criteria may have different importance/priorities?
- (3) How to convert this combined order into a ranking?

We call the problem with the above three issues the *Order Fusion Problem* [1].

Order Fusion can be considered as a special case of Data Fusion [3], which plays a crucial role for capturing data representation and aggregation. As Data Fusion's foundations, a variety of researches on both mathematically formal methods (e.g., Bayesian, Dempster-Shafer, Fuzzy Logic) and somewhat ad hoc approaches (e.g., voting) have been studied for as long as hundreds of years. These approaches may also be applied to Order Fusion. However, most of these approaches have the following weaknesses regarding the above three issues:

- (1) The lack of a description of qualitative relationships, which may not satisfy transitivity.

- (2) The lack of a formal theory to support aggregation.
- (3) The lack of the ranking methodologies in terms of using qualitative information.

About (1), all the current works prefer quantitative information rather than qualitative data. This leads to the lack of descriptions of the following two aspects:

- Intransitivity: In a competition that t_1 beats t_2 and t_2 beats t_3 does not necessarily mean t_1 can beat t_3 .
- Uncertainty: For instance, we may not know who is better between t_1 and t_2 because the match between them was canceled for some reason, say raining.

About (2), there is a lack of convincing basis to justify why a particular formula used for aggregation is valid in current approaches. For instance, in Top- N query problem, Euclidean functions may be used as the distance function [6]. It is difficult to determine if there is a better function.

Item (3) is the direct outcome of (1). Since most current researches deal with ranking problems involving only quantitative values, they do not need to distinguish the difference between Order and Rank. In this work, *order* means the direct relationships among the objects and *rank* means an overall linear list of the objects. A specific procedure must be developed to map an order to a rank.

Therefore, *Order Fusion* is a hard problem, even for its special case – Multiperson Decision Making where all evaluators are assumed to be of the same importance, there exist difficulties as described in the following examples.

The first example illustrates a case involving 3 objects and we have a good solution for ranking them.

Example 1.1: Suppose chess players take part in tournaments of the same importance. In each tournament, players compete with each other. The order between two players can be decided by counting who wins more between them in all the tournaments. Then, we have a graph of order relationships among all the players. Notice that this graph is aggregated based on all the tournaments and may contain circles. By measuring the difference between games of winning and games of losing for each player based on the graph, we can rank all the players. □

The next example illustrates a case involving only 2 objects, yet we do not have a good solution to rank them.

Example 1.2: Suppose two soccer teams take part in tournaments of the same importance. In each tournament, teams compete with each other. Suppose we have three tournaments T_1 , T_2 and T_3 involving two teams t_1 and t_2 ; in T_1 and T_2 , t_1 beats t_2 with the same score of 1:0; in T_3 , t_2 beats t_1 with a score of 7:0. Although t_1 beats t_2 twice, it is

still hard to determine which team is better since t_2 beats t_1 with a huge margin in its one win. People may have different opinions on which team is better. \square

For years, many researches have relied on Arrow’s Impossibility Theorem [4]: There is NO consistent method of making a fair choice based on the Fairness Criteria, and believed in “the presence of three or more objects/candidates” being the source of difficulty. However, as we can see from Example 1.2, even with 2 objects (soccer teams), we still cannot reach a convincing answer while we can in Example 1.1 where 3 objects (chess players) are involved. Thus, it appears that rather than the “the presence of three or more candidates” condition being the necessary condition required in Arrow’s Impossibility Theorem, there must exist a more profound hidden factor that contributes making a consistent solution impossible in terms of the fairness criteria in Decision Making. We believe that *it is the weights of the relationships among objects that cause the impossibility*. For instance, we have a good solution in Example 1.1 since no weight is involved. The relationship that one player beats another is merely a qualitative order relationship, i.e., the relationship conveys the qualitative information “who beats whom” but no quantitative information “by what margin”. In other words, the order relationship is not weighted. On the other hand, we don’t have a good solution in Example 1.2 because the relationships are weighted, e.g., a win by 1:0 is different from a win by 7:0.

In this paper, we introduce a new mathematical model - *Raw Relation Sets* - to tackle the issues of object description, object aggregation, and object ranking in Order Fusion, based on the statement above (i.e., weighted relationships are causing the problem) rather than on the Fairness Criteria to overcome the weaknesses of the existing solutions. Considering limited space however, we will emphasize how this new theory can be used to deal with the special case of Order Fusion - Multiperson Decision Making (also known as Voting) [2] as we mentioned above. How to use Raw Relation Sets to address Order Fusion can be found in [1].

The principle of the Raw Relation Sets is to use un-weighted order relationships, instead of numeric values, to represent the information in the real world. We explore a method with which weighed relationships such as those in Example 1.2 can be transformed to equivalent un-weighted relationships such as those in Example 1.1. Therefore, to provide a solution to the former, we only need to provide an answer to the latter. As illustrated in Example 1.1, the solution usually is much easier to obtain when only un-weighted relationships are present.

The remainder of this paper is organized as follows. In Section 2, a detail description of Raw Relation Set is given to deal with the issue of object description. In Section 3, raw relation set operation or aggregation is described to deal with the issue of object aggregation in Multiperson Decision Making. In Section 4, methods of object ranking

are introduced. Finally, a conclusion is given in Section 5.

2. Raw Relation Sets

One thing that makes decision making difficult is the meaning of “binary relationship”, which in fact consists of two pieces of information: *direction* and *weight*. For instance, position 1 is higher than position 3 with a bigger margin than that position 2 is higher than position 3. All the current researches have to evaluate these 2 factors and employ some kind of “balance” during the aggregation. Controversy usually occurs during balancing.

Now, if we can redefine a ranking problem in a model so that a set of binary relationships (it is complete but needs not to be transitive) contains *direction* only, and each relationship between 2 objects is independent of others, then the problem of aggregation will become much easier. This is because in this case, a decision can be obtained for each relationship independently first and some kind of majority rule similar to that used in Example 1.1 can then be applied to these decisions during the aggregation.

In this section, we present how to use Raw Relation Sets to describe preferences among objects with un-weighted relationships. Preference aggregation will be discussed in the next section.

Example 2.1: A contest contains 3 voters: $\{v_1, v_2, v_3\}$ and 3 candidates: $\{c_1, c_2, c_3\}$. The voters are of the same importance, i.e., their votes are of the same importance – this is a traditional ranking situation. The preferences are shown in Table 1. Position 1 is ahead of position 2, which in turn is ahead of position 3. The objective is to find the social choice or the overall ranking of the candidates.

| Voters: | v_1 | v_2 | v_3 |
|------------|-------|-------|-------|
| Position 1 | c_1 | c_2 | c_3 |
| Position 2 | c_2 | c_3 | c_1 |
| Position 3 | c_3 | c_1 | c_2 |

Table 1: Social preferences for 3 candidates.

The preferences of the voters result in a *Condorcet’s effect*, or *paradox of voting*, or has *cyclical majorities*, in which case c_1 has a 2-to-1 majority over c_2 , c_2 has a 2-to-1 majority over c_3 and c_3 has a 2-to-1 majority over c_1 . Intuitively (a more formal reasoning can also be provided), these candidates should be assigned the same ranking. We now consider another situation by adding a new candidate named c_1' . Suppose c_1' has the same position as c_1 in the preference of each voter, as shown in Table 2.

| Voters: | v_1 | v_2 | v_3 |
|------------|-------------|-------------|-------------|
| Position 1 | c_1, c_1' | c_2 | c_3 |
| Position 2 | c_2 | c_3 | c_1, c_1' |
| Position 3 | c_3 | c_1, c_1' | c_2 |

Table 2: Social preferences for 4 candidates.

For this example, Borda’s method [2] and all other methods that use scoring functions will give the same score to each candidate (see Section 4). Hence, all the 4 candidates will be ranked the same by these methods.

However, in the real world, the value of a position may need to be changed with the distribution of the objects in order to better reflect the reality. Consider the following 2

scenarios in a test:

- S1: All students get A except one who gets C;
- S2: All students get C except one who gets A.

If A is treated as position 1 and C is treated as position 2, then the position 1 in S2 should have a higher value than the position 1 in S1 since only one student gets A in S2. Although we cannot tell how to adjust the values of the position 1 in the two cases, any method that does not consider the difference is probably too simplistic.

Raw Relation Sets provides a platform for consensus ranking while addressing problems like the above. Based on this platform, the final ranking of these candidates is $c_3 > c_1, c_1' > c_2$, i.e., c_3 is ranked ahead of c_1 and c_1' which are in turn ranked ahead of c_2 . The formal justification will be provided in subsequent discussions. Intuitively, this is reasonable because c_3 wins more than it loses, c_2 loses more than it wins, and both c_1 and c_1' have the same numbers of wins and losses. \square

As stated above, we need to build a model to describe the preference for each voter. In this model, a binary relationship means nothing but a *direction*. The *weight* information, if exists, melts in those relationships.

Definition 2.1: Let X be a set of objects. An *Order* on X , denoted as $O(X)$, is a set of all the possible binary relationships among the objects in X . Specifically, for any two objects $x, y \in X$, exactly one of the following three relationships is true:

- x is *Larger* than y : denoted as $\langle x, y \rangle$;
- x is *Equal* to y : denoted as $\langle =x, y \rangle$ or $\langle =y, x \rangle$;
- Unknown*: No idea which one is larger, whether they are equal or comparable, denoted as $\langle x, y \rangle$ or $\langle y, x \rangle$. \square

Notes:

1) The relationship *Larger* has a generic meaning that subsumes *better* than, *more important* than, etc.

2) Since x is *Larger* than y implies that y is smaller than x , it is unnecessary to define *Smaller than* separately.

3) The relationship is not required to be transitive. In other words, that object x is *Larger* than object y , which in turn is *Larger* than object z , does not necessarily mean that x is *Larger* than z . Therefore, it permits the existence of a cycle. For instance, $O(X) = \{(x, y), (y, z), (z, x)\}$ is an order on X . Here, $X = \{x, y, z\}$. This feature also demonstrates another important fact: the relationships are independent to each other. We cannot simply deduce a relationship based on other relationships. Each relationship is obtained by surveying the fact occurring in the real world, e.g., the competing result between two players.

4) *Unknown* does not merely mean that we cannot decide whether there is a *Larger* or *Equal* relationship. More importantly, it sometimes means that the two objects are not comparable. For instance, we are not able to decide who runs faster between object 'Author' and object 'Raw Relation Sets'.

Definition 2.2: Suppose $(x, y) \in O(X)$. The *distance* (*object*) from x to y , denoted by $d(x, y)$, is a qualitative measure reflecting *how much* x is larger than y . \square

Note that distance is only defined for objects in the

Larger relationship. The distance for *Unknown* relationship would be unknown if we must define it. The distance for *Equal* relationship would always be smaller than that for a *Larger* relationship. Distances for *Unknown* and *Equal* relationships would not be useful in our solution and therefore would not be defined.

Example 2.2: Let $X = \{x, y, z\}$ be a set and $O(X) = \{(x, y), (y, z), (x, z)\}$. If $d(x, z)$ is *larger than* $d(y, z)$, it reflects that the degree in which x is larger than z is greater than that in which y is larger than z . In other words, it reflects a sense that x is *much larger than* z . \square

Definition 2.3: Let X be a set of objects and $O(X)$ be an *Order* on X . Suppose X_2 is a set of distance objects on $O(X)$. Then, we have a new *Order* $O(X_2)$. Generally, suppose X_i is a set of distance objects on $O(X_{i-1})$ and $O(X_i)$ is an *Order* on $X_i, i = 1, 2, \dots$. Then,

$X^* = \cup X_i$ is the *distance closure* on X , here $X_1 = X$;

$O^*(X) = \cup O(X_i)$ is the *Order closure* on X . \square

Now we define raw relation set.

Definition 2.4: Let X be a universal set. A *raw relation* set (or *raw set* for short) of X is

$R = (X^* \cup O^*(X)) \cup \{ \langle v_i, v_j \rangle \mid v_i \in X_i, v_j \in X_j, \text{ and } i \neq j \}$. \square

The second part of the union above means that the relationships between objects in different levels are always *Unknown* since they belong to different domains.

To make a raw set more visible, an equivalent definition is given in terms of graph theory.

Definition 2.5: A graph G consists of a set of objects called vertices $V(G)$ and a set of edges $E(G)$. Each edge is defined by a pair of ordered or un-ordered distinct vertices of G . $E(G)$ consists of the following types of edges:

$\langle x, y \rangle$: if there is a directed edge from x to y ;

$\langle =x, y \rangle$: if there is an undirected edge between x and y ;

$\langle x, y \rangle$: if there is 'no edge' between x and y . \square

The following is equivalent to Definition 2.4.

Definition 2.6: Let X be a universal set. A raw set R of X is a graph R_G defined as:

$V(R_G) = \cup V(R_{G_k})$;

$E(R_G) = (\cup E(R_{G_k})) \cup \{ \langle v, v' \rangle \mid v \in V(R_{G_i}), v' \in V(R_{G_j}), \text{ and } i \neq j \}$.

Here, R_{G_k} is a cluster of exclusive complete sub-graphs of R_G , defined as:

$V(R_{G_i}) = X$;

$E(R_{G_i}) = \{ (v_i, v_j) \mid v_i \text{ is Larger than } v_j \} \cup$

$\{ \langle =v_i, v_j \rangle \mid v_i \text{ is Equal to } v_j \} \cup$

$\{ \langle v_i, v_j \rangle \mid \text{Unknown between } v_i \text{ and } v_j \}$, here, $v_i, v_j \in X$;

$V(R_{G_k}) = \{ d(x, y) \text{ or } \langle x, y \rangle \mid (x, y) \in E(R_{G_{k-1}}) \}$;

$E(R_{G_k}) = \{ (v_i, v_j) \mid v_i \text{ is Larger than } v_j \} \cup$

$\{ \langle =v_i, v_j \rangle \mid v_i \text{ is Equal to } v_j \} \cup \{ \langle v_i, v_j \rangle \mid$

Unknown between v_i and $v_j \}$, here, $v_i, v_j \in V(R_{G_k})$

and $1 < k \leq h$, and h is a constant integer in $[1, +\infty]$. \square

Note: Any two sub-graphs are connected by special edges of type 'no edge' (for *Unknown* relationships). A graph for a raw set can be considered as consisting of multiple levels of sub-graphs with R_{G_1} being at the first, top, or highest level and R_{G_h} being at the last, bottom, or the lowest level. A lower level graph is built from the

graph at the immediate higher level. Essentially, the directed edges in a graph become the vertices in the graph at its next lower level. The edges in the graph at the lower level describe the distance relationships at the higher level. The number of levels h is referred to as the *height* of the raw set R , denoted as $h = \text{height}(R)$.

Now let's see the raw relation sets for the preferences in Table 2 of Example 2.1. Three raw sets can be constructed for the three preferences corresponding the three voters. In Figure 2.1, we defined a raw relation set for the preference of voter v_1 . Since v_1 considers c_1 and c_1' are in position 1, we have *Equal* relationship between these two candidates, i.e., there is undirected edge between them in the first level illustrated in the left sub-graph. In addition, because c_2 is in position 2 and c_3 is in position 3, we have *Larger* relationships from c_1 and c_1' to c_2 and c_3 , and from c_2 to c_3 as illustrated in the left sub-graph as well. The directed edges (they correspond to the *Larger* relationships) are labeled and they become the vertices in the next-level sub-graph - the middle sub-graph. Since the distance between position 1 and position 3 is larger than both the distance between position 1 and position 2 and the distance between position 2 and position 3, and the later two distances are equal to each other, we have the corresponding relationships or the edges. Notice that the direct edges in this sub-graph means the distance between "the distance between position 1 and position 3" and "the distance between position 1 and position 2" or "the distance between position 2 and position 3". Hence, those directed edges have the same weight. Thereby in the lowest level - the right sub-graph, we have only *Equal* edges among those distances. Following the same logic, we can construct the other raw relation sets for the preferences of voters v_2 and v_3 (see Figure 2.2 and 2.3).

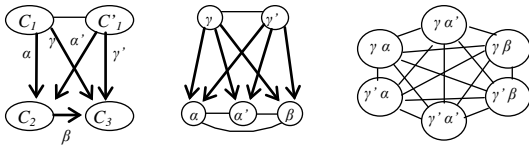


Figure 2.1: Raw set for v_1 's preference.

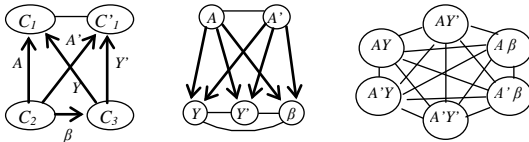


Figure 2.2: Raw set for v_2 's preference.

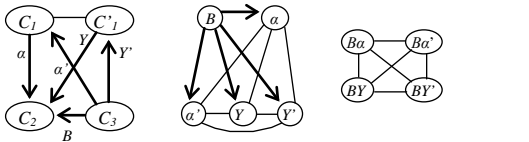


Figure 2.3: Raw set for v_3 's preference.

Now, each preference is defined as a raw set, which is a graph consisting of three-levels of sub-graphs. Notice that there is no weight on any edge. As we mentioned earlier, information regarding weights has melted in the three levels of the graph for each preference.

The following special types of raw sets will be used in later discussions.

Definition 2.7: If the relationships among n objects in X are all *Equal*, the corresponding raw set is referred to as an *Equal n -raw set*, denoted by E_n . \square

3. Raw Set Operation

Different from the traditional set operations, a raw set operation itself is a raw set.

Definition 3.1: A raw set operation (or raw operation for short) is a raw set Op . $V(Op_G_I)$ serves as a *committee* C whose members' priorities on a decision are defined by Op . Each of the n ($= |C| = |V(Op_G_I)|$) members has a decision I_i ($1 \leq i \leq n$) serving as the input, which is a raw set on X . As the output of Op , the aggregated decision $R = Op(I_1, I_2, \dots, I_n)$ of C is also a raw set on X . \square

Intuitively, committee C can be thought of as a set of evaluators. Each I_i is an order among the objects in X produced by an evaluator. There is an order among the evaluator objects in the committee based on the priorities (importance) of these evaluators. The raw operation aggregates the decisions $\{I_i \mid i = 1, 2, \dots, n\}$ into an overall decision (order) on X based on the order on C .

Since most Multiperson Decision Making applications involve E_n raw set operation, i.e., no committee member is more important than any other committee members on the committee's decision, we will focus this raw operation. The discussion on the case of employing general raw operations for Order Fusion is given in [1].

For each pair of objects x and y , each member in the committee has identified a relationship or made a decision for them. In the following we discuss some principles for reaching the committee's decision.

Principle 1: In a committee with only one member, that member's decision is the committee's decision.

This principle is easy to accept. \square

Principle 2: If the number of members with decision (x, y) (or (y, x)) in a committee is larger than that of members with decision (y, x) (or (x, y) , respectively), then the aggregated decision is (x, y) (or (y, x) , respectively). Otherwise, If the numbers are the same, then the aggregated decision is $\langle =x, y \rangle$ unless all of the members' decisions are *Unknown*. In the later case the aggregated decision is $\langle x, y \rangle$. \square

This principle can be explained as follows. Consider two players, if player x wins over player y more than he/she loses, the decision is (x, y) , i.e., x is better than y . If the opposite is true, then (y, x) . If all games are cancelled, then the decision is *Unknown* (i.e., $\langle x, y \rangle$). Otherwise, if they win and lose the same number of times, then the two are tied (i.e., $\langle =x, y \rangle$).

Based on these two principles, we now have a postulate described as an algorithm to summarize the above explanations.

The members' decisions can be divided into the following four groups:

G_x : all members in this group have decision (x, y) ;

G_y : all members in this group have decision $\langle y, x \rangle$;
 G_e : all members in this group have decision $\langle =x, y \rangle$;
 G_n : all members in this group have decision $\langle x, y \rangle$.

Postulate (Majority Rule): Suppose C is a committee with $Num_x = |G_x|$, $Num_y = |G_y|$, $Num_e = |G_e|$, and $Num_n = |G_n|$. The aggregated decision is returned based on the following algorithm:

Decision ($Num_x, Num_y, Num_e, Num_n$)

BEGIN

- (1) If ($Num_x = Num_y = Num_e = 0$) return $\langle x, y \rangle$;
- (2) Else If ($Num_x > Num_y$) return $\langle x, y \rangle$;
- (3) Else If ($Num_y > Num_x$) return $\langle y, x \rangle$;
- (4) Else return $\langle =x, y \rangle$.

END □

The Majority Rule works for a pair of objects only. When we apply it to all the relationships of the relevant raw sets, we need to repeat using the Majority Rule for each relationship. Hence, we have the following procedure, which is a special case studied in [1].

Definition 3.2: Op is a raw operation on raw sets, if:

- Op is a raw set, whose first level vertices $N_1, N_2, \dots, N_n \in V(Op_G_1)$ serve as the parameters;
- I_1, I_2, \dots, I_n are all raw sets on domain X , which will be taken as the inputs of $V(Op_G_1)$;
- R is a raw set on domain X , as the output of Op .

RawOp(Op, I_1, I_2, \dots, I_n)

BEGIN

1. $h \leftarrow \text{MAX}(\text{Height}(I_1), \text{Height}(I_2), \dots, \text{Height}(I_n))$;
2. **FOR** $k \leftarrow 1$ **TO** h **DO**
3. $E(R_G_k) \leftarrow \text{Empty}$;
4. **IF** ($k = 1$) **THEN**
5. $V(R_G_k) \leftarrow X$;
6. **ELSE**
7. $V(R_G_k) \leftarrow$ all the directed edges in R_G_{k-1} ;
8. **FOR** each pair ' x, y ', $x, y \in V(R_G_k)$ **DO**
9. **FOR** $i \leftarrow 1$ **TO** n **DO**
10. Get the relationship of ' x, y ' in $I_i_G_k$;
11. Put these relationships into the corresponding vertices in Op_G_1 ;
12. **ENDFOR**(i)
13. Get $Num_x, Num_y, Num_e, Num_n$;
14. $r = \text{Decision}(Num_x, Num_y, Num_e, Num_n)$
15. Add r into $E(R_G_k)$;
16. **ENDFOR**(each pair)
17. **ENDFOR**(k)

END

Notes:

1) The algorithm employs the top-down procedure to produce raw set R . It finds the relationships for any 2 objects x and y from all the input raw sets (Line 10) and applies the Majority Rule to those relationships so as to produce the aggregated relationship for x and y .

2) The loop controller in Line 2 is used to scan all the levels, while the loop controller in Line 8 is used to find all the relationships in a single level.

3) In Line 14, the Majority Rule applies on those individual decisions. The result or the aggregated decision

is a part of k -th level of the final aggregated decision or the aggregated raw set, which is filled gradually in Line 15.

4) Some other issues need to be addressed in this algorithm. Due to space limitation, they are not addressed here but they can be found in [1].

Example 3.1 Multiperson decision making (continue Example 2.1). Now we apply the Algorithm *RawOp*, which is E_3 since we have 3 voters and all of them have the same importance, to the raw sets in Example 2.1.

First, the inputs of *RawOp* are three preferences, which are the three raw sets ($n = 3$) described in Figure 2.1, Figure 2.2, and Figure 2.3. Since each of these raw sets has three levels, we have $h = 3$ in Line 1. In Line 2, k is set to 1 to indicate we start from level 1 of those three raw sets. Since, we want to find all the relationships (edges) of level 1 in the aggregated raw set, we first initialize the set of those relationships by setting it empty in Line 3. We also initialize the set of the related objects, which is $V(R_G_1)$ in Line 5 (when we come back again to this point next time, i.e., for other levels, we use Line 7 instead) by assigning the value $\{c_1, c_1', c_2, c_3\}$.

Now in Line 8 we get into another loop to deal with each pair of objects in $V(R_G_1)$, which is $\{c_1, c_1', c_2, c_3\}$ at this point. Suppose the first pair is ' c_1, c_2 ', i.e., c_1 is x while c_2 is y . Since $n = 3$ in this example, we need to get each relationship between ' c_1, c_2 ' from each input raw set by running the loop in Line 10. Note that for this example, *RawOp* (i.e., the raw sets of the committee consisting of the three voters) has only one level because all the voters are of the same importance. Op_G_1 contains three voters: v_1, v_2, v_3 .

Therefore in Line 11, we set v_1 's decision as $\langle c_1, c_2 \rangle$ that is defined in v_1 's preference. By the same logic, we set v_2 's decision as $\langle c_2, c_1 \rangle$ and set v_3 's decision as $\langle c_1, c_2 \rangle$. Then in Line 13, we have $Num_x = 2$ since c_1 is *Larger* than c_2 two times. Also we have $Num_y = 1$, $Num_e = 0$, and $Num_n = 0$. In Line 14, case (2) of the Majority Rule occurs, leading to $r = \langle c_1, c_2 \rangle$. This result is put into $E(R_G_1)$ in Line 15.

Then we go back to Line 8 five times and find out that the other aggregated decisions are $\langle c_3, c_1 \rangle$, $\langle c_2, c_3 \rangle$, $\langle =c_1, c_1' \rangle$, $\langle c_1', c_2 \rangle$, $\langle c_3, c_1' \rangle$. These six relationships (including $\langle c_1, c_2 \rangle$) construct $E(R_G_1)$, which is the first level of the aggregated raw set illustrated as the left sub-graph in Figure 3.1. In $E(R_G_1)$ we have 5 *Larger* relationships, labeled $\alpha, \alpha', \beta, Y, Y'$, respectively, as shown in Figure 3.1.

Now we go back to Line 2 again. This time k is 2, which means we are going to deal with the second level of raw sets. Therefore we will execute Line 7 instead of Line 5 and $V(R_G_2)$ is $\{\alpha, \alpha', \beta, Y, Y'\}$, which will be the vertices of the second level of the aggregated raw sets as shown in the right sub-graph in Figure 3.1.

Again, we run into Line 8. Suppose the pair ' x, y ' is ' α, β '. First, $\langle =\alpha, \beta \rangle$ is in the second level of v_1 's preference (raw set). We have $\langle \alpha, \beta \rangle$ according the second level of v_2 's preference because α does not exist in the second level of v_2 's preference we have no way to decide the relationship between them. Following the same logic, we

have $\langle \alpha, \beta \rangle$ according the second level of v_3 's preference.

Therefore this time we have $Num_x = 0$, $Num_y = 0$, $Num_e = 1$, and $Num_n = 2$. In Line 14 we obtain $r = \langle \alpha, \beta \rangle$, which is illustrated in the right sub-graph of Figure 3.1, serving as another relationship in the aggregated raw set.

In the same way, we can find all the other relationships for the second level of the aggregated raw set. Because all those relationships are *Equal*, we do not have level 3, the algorithm will then stop immediately. Figure 3.1 is the aggregated raw sets representing the committee's decision.

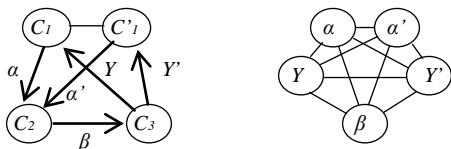


Figure 3.1 The raw sets for the final decision.

The raw set contains cycles. The raw set describing the decision does not provide a linear object rank to which people often would like to see in practice. In next section we discuss how to convert a raw set to a linear list or how to rank the objects in a raw set.

4. Ranking

Ranking vertices is one of the primary issues in Graph Theory. The methods are mostly divided into two groups: *path-oriented* (PO) and *weight-oriented* (WO) [7]. Since WO works for any cases even when a cycle is involved, we usually apply WO to implement object ranking. Besides, WO always yield a unique solution

To apply a WO approach to rank vertices (objects) in an aggregated raw set, we need to *bring weight back to our directed graph after aggregation has been done*. Once a new directed graph with weighted edges is obtained, we assign a weight to each vertex, which is the difference between the sum of the weights on inbound edges and that on outbound edges. We then rank the vertices according to their weights. This ranking procedure may repeat many times from the bottom to the top in the aggregated raw set until we obtain the top level ranking or the candidate ranking. The intuition behind this method of assigning a weight to a node is that an inbound edge corresponds to a loss and an outbound edge corresponds to a win and thus the weight reflects the difference between wins and losses. Due to space limitation and the fact that some issues have already been discussed in Graph Theory, we will not go any further with object ranking in this paper.

According to the discussion above, we have the final ranking for Example 2.1, which is:

$$c_3 > c_1, c_1' > c_2.$$

This is because, based on Figure 3.1, all the directed edges are considered having the same weight since no one prevails over others based on the second level (right sub-graph) of Figure 3.1. Based on the first level, c_3 wins more than it loses while c_2 loses more than it wins. Both c_1 and c_1' have the same numbers of wins and losses.

It can be seen that no variation of Borda methods will

obtain the above ranking for this example. This is because no matter how people assign values to the positions, all the candidates will have the same score. Suppose a candidate receives score s_j if s/he is ranked at the first position by any voter. This assumption makes sense because the voters in our example are of the same importance. Similarly, let s_2 and s_3 be the scores associated to the second and third positions, respectively. Now suppose we have a scoring function $B(p_1, p_2, p_3)$, which combines individual scores of a candidate into a combined score, where p_1, p_2 and p_3 are the scores that a candidate receives from v_1, v_2 and v_3 , respectively. Since those voters are of the same importance, we should have the following equalities: $B(p_1, p_2, p_3) = B(p_1, p_3, p_2) = B(p_2, p_1, p_3) = B(p_2, p_3, p_1) = B(p_3, p_1, p_2) = B(p_3, p_2, p_1)$, i.e., the order of the parameters is not significant. This means that the combined scores for the three candidates have the same value. Therefore, the three candidates will be ranked the same. As traditional distance/aggregate functions such as min, max, sum and Euclidean are variations of Borda functions, it raises question about their validity for generating combined rankings. A good method should consider both the ranking positions of different objects in individual orders and the distributions of these objects at different positions.

5. Conclusions

This paper proposed a new method to solve the *Multiperson Decision Making* problem when all evaluators are of the same importance. This method is based on the theory of Raw Relation Sets [1]. By concentrating on un-weighted direct relationships between objects and by converting weighted relationships into equivalent un-weighted ones, a simple and powerful rule – majority rule can be applied to perform preference aggregation. Through a simple example, it was shown that the proposed method can provide more reasonable solutions than any variation of the Borda method.

References

- [1] Y. Chen. Raw Relation Sets, Order Fusion and Top- N Query Problem. Ph.D. Dissertation, Department of Computer Science, Binghamton University, 2002.
- [2] P. Fishburn. Multiperson Decision Making: A Selective Review. J. Kacprzyk and M. Fedrizzi(eds.), Multiperson Decision Making Using Fuzzy Sets and Possibility Theory, p17. ©1990 Kluwer Academic Publishers. Printed in the Netherlands.
- [3] Richard T. Antony, "Principles of Data Fusion Automation". Artech House, Inc. 1995.
- [4] K. Arrow. Social Choice and Individual Values. Wiley, New York, 1963.
- [5] M. Carey and D. Kossmann. On saying "Enough Already!" in SQL. In Proc. of the 1997 ACM International Conf on Management of Data (SIGMOD'97), May 1997. p219-230.
- [6] S. Chaudhuri, and L. Gravano. Evaluating top-k selection queries. In Proc. of the Twenty-fifth International Conference on Very Large Data Bases (VLDB'99), 1999.
- [7] J. Bondy and U. Murty, Graph Theory with Applications, Macmillan, London, 1976.