A Minimum Violations Ranking Method

Kathryn E. Pedings, Amy N. Langville ·
Yoshitsugu Yamamoto

Abstract We present a rating method that, given information on the pairwise comparisons of n items, minimizes the number of inconsistencies in the ranking of those items. Our Minimum Violations Ranking (MVR) Method uses a binary linear integer program (BILP) to do this. We prove conditions when the relaxed LP will give an optimal solution to the original BILP. In addition, the LP solution gives information about ties and sensitivities in the ranking. Lastly, our MVR method makes use of bounding and constraint relaxation techniques to produce a fast algorithm for the linear ordering problem, solving an instance with about one thousand items in less than 10 minutes.

1 Introduction

In this paper, we present a rating method that, given information on the pairwise comparisons of n items, minimizes the number of inconsistencies in the ranking of those items. Though Minimum Violations Ranking (MVR) methods have many applications including economic input-output matrices and sequencing of items from archaeological digs [12], we use examples from sports to explain our new MVR method. The matrix **D** below, which we call a point differential matrix, contains pairwise comparison data and is commonly and easily produced for many sports.[1]

Kathryn E. Pedings and Amy N. Langville
Department of Mathematics, 175 Calhoun Street-RSS 339
Charleston, SC 29401, USA College of Charleston
Tel.: 843-953-8021, Fax: 843-953-1410
E-mail: kepeding@edisto.cofc.edu, langvillea@cofc.edu

Yoshitsugu Yamamoto
Graduate School of Systems and Information Engineering
University of Tsukuba, Ibaraki 305-8573, Japan
E-mail: yamamoto@sk.tsukuba.ac.jp

[1] A point differential matrix could also be created so that it includes both positive and negative values for the positive and negative point differentials.

$$
\mathbf{D} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 9 & 0 & 4 & 0 & 2 \\ 5 & 0 & 0 & 0 & 0 \\ 15 & 3 & 8 & 0 & 5 \\ 6 & 0 & 3 & 0 & 0 \end{pmatrix} \end{array}.
$$

The $(2, 3)$-entry means that team 2 beat team 3 by 4 points in their matchup. We will analyze this point differential matrix in order to produce a ranking of these five teams. At this point we introduce a definition.

**Definition 1** A matrix $\mathbf{D}$ is in *hillside form* if

$$
\begin{aligned}
d_{ij} \leq d_{ik} , \quad & \forall\, i \ \text{ and } \ \forall\, j \leq k \qquad \text{(ascending order across rows)} \\
d_{ij} \geq d_{kj} , \quad & \forall\, j \ \text{ and } \ \forall\, i \leq k. \qquad \text{(descending order down columns)}
\end{aligned}
$$

The name is suggestive as a cityplot of a matrix in hillside form looks like a sloping hillside. The matrix $\mathbf{A}$ below is in hillside form, while $\mathbf{B}$ is not.

$$
\mathbf{A} = \begin{pmatrix} 0 & 3 & 5 & 8 & 15 \\ 0 & 0 & 2 & 4 & 9 \\ 0 & 0 & 0 & 3 & 6 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad \mathbf{B} = \begin{pmatrix} 0 & 3 & 5 & 8 & 15 \\ 0 & 0 & 2 & 4 & 9 \\ 7 & 0 & 0 & 3 & 4 \\ 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.
$$

For $n \times n$ matrices in hillside form, the ranking $\mathbf{r}$ of the items is clear: $\mathbf{r} = \begin{pmatrix} 1 & 2 & \cdots & n \end{pmatrix}$. For non-hillside matrices, we can count the number of violations of the hillside conditions. In the above example, $\mathbf{B}$ has 7 violations. Often a matrix that appears to be non-hillside can be symmetrically reordered so that it is in hillside or near hillside form. In fact, the non-hillside matrix $\mathbf{D}$ is the perfect hillside matrix $\mathbf{A}$ when $\mathbf{D}$ is reordered according to the vector $\begin{pmatrix} 5 & 2 & 4 & 1 & 3 \end{pmatrix}$. *Finding such a hidden hillside structure is exactly the aim of our MVR method.*

Our MVR method finds a reordering of the items that when applied to the item-item matrix of differential data forms a matrix that is as close to *hillside form* as possible. Figure 1 summarizes the concept pictorially. On the left is a cityplot of an
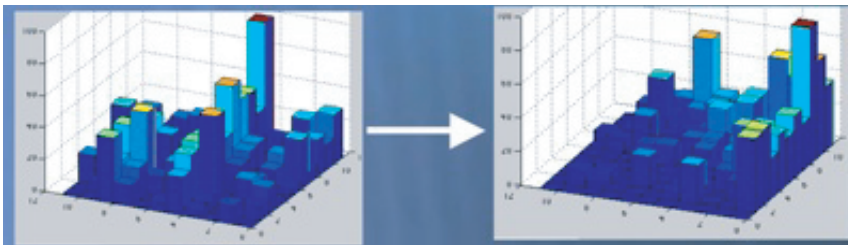
Fig. 1 Cityplot of 11×11 data matrix with original ordering and MVR reordering

11×11 matrix in its original ordering of items, while the right is a cityplot of the same data displayed with the new optimal ordering. In Section 3, we explain how we produce such an optimal ordering.

Hillside form gives a great deal of information about the difference in the strengths of teams. For example, matrix **A** says that not only is team 1 ranked above teams 2, 3, 4, and 5, but we expect team 1 to beat team 2 by some margin of victory, then team 3 by an even greater margin, and so on. Sometimes a data matrix has been reordered to be as close to hillside form as possible, yet violations remain. These violations are of two types: *upsets* and *weak wins*. Nonzero entries in the lower triangular part of the reordered matrix correspond to upsets, i.e., when a lower ranked team beat a higher ranked team. Weak wins mainfest as violations of the hillside conditions that occur in the upper triangular part of the matrix. This is when a high ranked team beats a low ranked team but by a smaller margin of victory than expected. It is possible to weight these two types of violations non-uniformly if the modeler has a greater aversion to one over another. In fact, our MVR method does this inherently. The example matrix **B** above demonstrates this well. Notice that the presence of the 7 in the lower triangular part of the matrix accounted for 6 of the 7 violations.

2 Related Work

Several MVR methods have been proposed [1], [2], [3], [5], [6], and [9]. However, because these MVR methods only consider the upset type of violation, our MVR method is more comprehensive. In fact, since our method considers both upset and weak win violations, it produces a stricter ranking of the items. Very few methods consider tied events as part of the input data and even fewer allow for the possibility of ties in the output ranking [3]. On the other hand, our MVR method allows for both input and output ties. The output ties are a consequence of the beautiful theory of linear programming. (See Section 3.1.3 for more on ties and their relationship to multiple optimal solutions.) In summary, our MVR method contributes three new features: (1) it produces an optimal ranking that minimizes the number of both upset violations and weak win violations, (2) it produces an optimal ranking that may include ties, (3) it identifies alternate optimal rankings, if they exist, (4) it provides sensitivity measures for the optimal ranking, and (5) it is fast, producing an MVR or linear ordering for one thousand items in about 10 minutes.

3 Solving the MVR problem using distance to hillside form

Our MVR problem, i.e., finding a reordering of the items that brings the data matrix $\mathbf{D}_{n \times n}$ as close to hillside form as possible can be stated mathematically as follows. Find the permutation matrix $\mathbf{Q}_{n \times n}$ so that the symmetrically reordered matrix $\mathbf{Q}^T \mathbf{D} \mathbf{Q}$ has minimal hillside violations. The optimization problem below

$$\min_{\mathbf{Q}} \quad \text{\# hillside violations of } \mathbf{Q}^T \mathbf{D} \mathbf{Q}$$

$$s.t. \quad \mathbf{Q}^T \mathbf{e} = \mathbf{e}$$
$$\mathbf{e}^T \mathbf{Q} = \mathbf{e}^T$$
$$q_{ij} \in \{0, 1\}$$

has linear constraints, binary variables, and a quadratic objective function, all of which put it into a challenging class of optimization problems. Fortunately, the alternate formulation of the next section makes the problem much more tractable.

3.1 BILP

In this section we formulate our MVR problem as a binary integer linear program (BILP), which is a much more tractable formulation than the quadratic integer program (QIP) above. In order to reach the BILP formulation, we need to define some constants and some decision variables. First, we define the constants c_{ij}.

Definition 2 of C matrix: Let $\mathbf{C} = [c_{ij}] \ \ \forall \, i, j = 1, 2, \ldots, n$ be defined as

$$c_{ij} := \#\{\, k \mid d_{ik} < d_{jk} \,\} + \#\{\, k \mid d_{ki} > d_{kj} \,\}, \tag{1}$$

where $\#$ denotes the cardinality of the corresponding set. Thus, $\#\{\, k \mid d_{ik} < d_{jk} \,\}$ is the number of teams receiving a lower point differential against team i than team j. Similarly, $\#\{\, k \mid d_{ki} > d_{kj} \,\}$ is the number of teams receiving a greater point differential against team i than team j.[2]

Theorem 1 *The cost matrix C defined above can be used to compute the number of violations to hillside form.*

Proof   To count the number of hillside violations associated with a particular ranking, we use that ranking to symmetrically reorder $\mathbf{D}$, then count the number of violations, denoted $\#viol(\mathbf{D})$ according to the rule

$$\#viol(\mathbf{D}) = \#\{\, k \mid d_{ik} < d_{jk}, \ \forall i < j \} + \#\{\, k \mid d_{ki} > d_{kj}, \ \forall i < j \}.$$

Assume the ranking is $1, 2, \ldots, n$. Then in the binary integer program, the solution matrix $\mathbf{X}$ associated with that ranking is an upper triangular matrix of 1s. (Both the integer program and the matrix $\mathbf{X}$ are described on the next page.) As a result, the objective function

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} = (c_{11} + c_{12} + \cdots + c_{1n}) + (c_{23} + \cdots + c_{2n}) + \cdots + (c_{n-1,n}),$$

which is all the elements in the strict upper triangular part of $\mathbf{C}$. Applying the definition for element c_{ij} and summing this for each element in the strict upper triangular part of $\mathbf{C}$, we find that

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} = \#\{\, k \mid d_{ik} < d_{jk}, \ \forall i < j \} + \#\{\, k \mid d_{ki} > d_{kj}, \ \forall i < j \} = \#viol(\mathbf{D}).$$

Without loss of generality we can assume that the ranking is $\{1, 2, \ldots, n\}$ because otherwise, we simply reorder $\mathbf{D}$, $\mathbf{C}$, and $\mathbf{X}$ in which case the ranking associated with the reordered matrices is $\{1, 2, \ldots, n\}$.

---

[2] The matrix $\mathbf{C}$ above counts hillside violations in a binary fashion, however, something more sophisticated can be done. For instance, we can consider weighted violations by summing the difference each time a hillside violation occurs. In this case, $\mathbf{C}$ is defined as $c_{ij} := \sum_{k:d_{ik}<d_{jk}} (d_{jk} - d_{ik}) + \sum_{k:d_{ki}>d_{kj}} (d_{ki} - d_{kj})$.

Consider a data matrix $\mathbf{D}$ of point differentials, where d_{ij} is the number of points winning team i beat losing team j by in their matchup, 0 otherwise. The trick to relating this to MVR is to think of each row and column of $\mathbf{D}$ as a team's ranking of its opponents. For instance, for our 5-team example with

$$
\mathbf{D} = \begin{array}{c} \\ \text{Duke} \\ \text{Miami} \\ \text{UNC} \\ \text{UVA} \\ \text{VT} \end{array}
\begin{array}{ccccc}
\text{Duke} & \text{Miami} & \text{UNC} & \text{UVA} & \text{VT} \\
\left(\begin{array}{ccccc}
0 & 0 & 0 & 0 & 0 \\
45 & 0 & 18 & 8 & 20 \\
3 & 0 & 0 & 2 & 0 \\
31 & 0 & 0 & 0 & 0 \\
45 & 0 & 27 & 38 & 0
\end{array} \right)
\end{array},
$$

the second row of $\mathbf{D}$ tells us that Miami would rank its opponents' defensive ability from strongest to weakest as UVA, UNC, VT, Duke. On the other hand, the first column of $\mathbf{D}$, for example, tells us that Duke would rank its opponents' offensive ability as Miami/VT, UVA, UNC. Consequently, these rankings, both offensive and defensive, for the n teams can be aggregated to create an overall ranking for the season. In fact, our MVR method can be described as a rank aggregation method with a very special definition of the conformity matrix $\mathbf{C}$. (See Section 8.)

Armed with a matrix $\mathbf{C}$ of constants that helps count the number of violations to hillside form, our goal is to create a ranking of the n items that minimizes this number. In order to accomplish this goal, we define decision variables x_{ij} that determine if item i should be ranked above item j. In particular,

$$
x_{ij} = \begin{cases} 1, & \text{if item } i \text{ is ranked above item } j \\ 0, & \text{otherwise.} \end{cases}
$$

To understand the use of the matrix $\mathbf{X}$, consider a small example with $n = 4$ items labeled 1 through 4 and ranked in that order. Then the matrix $\mathbf{X}$ associated with this ranking is

$$
\mathbf{X} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array}
\begin{array}{cccc}
1 & 2 & 3 & 4 \\
\left(\begin{array}{cccc}
0 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0
\end{array} \right)
\end{array},
$$

which indicates that item 1 is ranked above items 2, 3, and 4, while item 2 is ranked above items 3 and 4, and finally, item 3 is ranked above only item 4. In this example, the nice stairstep structure of $\mathbf{X}$ clearly reveals the ranking. At first glance, for other examples, it may not be as clear. Consider the matrices $\mathbf{X}$ and $\mathbf{Y}$ below.

$$
\mathbf{X} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array}
\begin{array}{cccc}
1 & 2 & 3 & 4 \\
\left(\begin{array}{cccc}
0 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0
\end{array} \right)
\end{array}
\quad \text{and} \quad
\mathbf{Y} = \begin{array}{c} \\ 4 \\ 2 \\ 1 \\ 3 \end{array}
\begin{array}{cccc}
4 & 2 & 1 & 3 \\
\left(\begin{array}{cccc}
0 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0
\end{array} \right)
\end{array}.
$$

$\mathbf{Y}$ is simply $\mathbf{X}$ reordered according to the rank ordering of $[4, 2, 1, 3]$. Fortunately, there is no need to actually reorder the matrix $\mathbf{X}$ resulting from the optimization because

it is very easy to identify the ranking from the unordered $\mathbf{X}$. Simply take the column sums of $\mathbf{X}$ and sort them in ascending order to obtain the ranking.[3]

With $\mathbf{X}$ well understood, we now return to the optimization problem. We want to minimize the number of violations to hillside form, which, in terms of our constants c_{ij} and variables x_{ij} becomes

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}\, x_{ij} \quad \text{with } x_{ij} \in \{0, 1\}.$$

However, we must add some constraints that force the matrix $\mathbf{X}$ to have the properties that we observed and exploited in the small 4×4 example above. This can be accomplished by adding constraints of two types:

$$x_{ij} + x_{ji} = 1 \quad \text{for all distinct pairs } (i, j) \qquad \text{(Type 1—antisymmetry)}$$
$$x_{ij} + x_{jk} + x_{ki} \leq 2 \quad \text{for all distinct triples } (i, j, k) \qquad \text{(Type 2—transitivity)}$$

The first constraint is an anti-symmetry constraint, which says that exactly one of x_{ij} and x_{ji} can be turned "on" (i.e., set equal to 1). This captures the fact that there are only two choices describing the relationship of i and j: either i is ranked above j or j is ranked above i. The second constraint is a very clever and compact way to enforce transitivity. That is, if $x_{ij} = 1$ (i is ranked above j) and $x_{jk} = 1$ (j is ranked above k), then $x_{ik} = 1$ (i is ranked above k). Transitivity is enforced by the combination of the Type 1 and Type 2 constraints. Because the decision variables are binary, the Type 2 constraint forbids cycles of length 3 from item i back to item i. The Type 1 constraint forbids cycles of length 2. In fact, these two constraints combine to forbid cycles of any length. A *dominance graph* helps explain this.

The matrix $\mathbf{X}$ from our 4×4 example,

$$\mathbf{X} = \begin{array}{c@{}c} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \left(\begin{array}{cccc} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{array} \right) \end{array},$$

can be alternatively described with the dominance graph of Figure 2. Every ranking vector produces a graph of this sort, which shows the dominance of an item over all items below it. The dominance graph for every ranking vector will contain no *upward arcs* as an upward arc corresponds to a cycle, i.e., a violation of Type 2 transitivity constraints. To see how the Type 1 and 2 constraints combine to forbid any cycles, consider the cycle from $1 \rightarrow 3 \rightarrow 4 \rightarrow 1$, which corresponds to the Type 2 constraint $x_{13} + x_{34} + x_{41} \leq 2$. Because item 1 is ranked above item 3, x_{13} is turned on (i.e., $x_{13} = 1$). Similarly, $x_{34} = 1$. Then according to the Type 2 constraint, x_{41} must equal 0. Combining this with the Type 1 constraint, then x_{14} must equal 1, and thus, transitivity is enforced. In summary, all three types of constraints (Type 1 and Type 2 plus the binary constraint on x_{ij}) combine to produce an $\mathbf{X}$ matrix solution that is a simple reordering away from the stairstep form. Finally, because $\mathbf{X}$ is always a reordering of the stairstep matrix, it has unique row and column sums, and thus,

---

[3] Or the row sums sorted in descending order could be used.
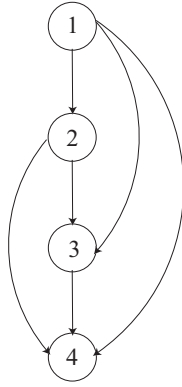
Fig. 2 Dominance graph

produces a unique ranking of the n items. The complete binary integer linear program (BILP) is

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}\, x_{ij}$$

$$x_{ij} + x_{ji} = 1 \quad \text{for all distinct pairs } (i,j) \qquad \text{(Type 1--antisymmetry)}$$

$$x_{ij} + x_{jk} + x_{ki} \le 2 \quad \text{for all distinct triples } (i,j,k) \qquad \text{(Type 2--transitivity)}$$

$$x_{ij} \in \{0,1\} \qquad \text{(Type 3--binary)}$$

Our MVR BILP contains $n(n-1)$ binary decision variables, $n(n-1)$ Type 1 equality constraints, and $n(n-1)(n-2)$ Type 2 inequality constraints. The $O(n^3)$ Type 2 constraints dramatically limit the size of ranking problems that can be solved with this optimal MVR method. Fortunately, there are some strategies (see Section 3.1.2) for sidestepping this issue of scale.

3.1.1 SoCon example

This 12-team example comes from the 2008-2009 Southern Conference (SoCon) basketball season. The MVR definition for the **C** matrix produces

$$\mathbf{C} = \begin{array}{c|cccccccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\
\hline
1 & 0 & 15 & 15 & 14 & 17 & 7 & 4 & 4 & 9 & 2 & 10 & 11 \\
2 & 8 & 0 & 10 & 12 & 18 & 6 & 3 & 3 & 11 & 3 & 7 & 8 \\
3 & 5 & 11 & 0 & 9 & 14 & 6 & 2 & 4 & 9 & 2 & 5 & 9 \\
4 & 5 & 9 & 9 & 0 & 15 & 5 & 0 & 2 & 6 & 3 & 6 & 5 \\
5 & 2 & 2 & 5 & 3 & 0 & 2 & 1 & 2 & 0 & 1 & 1 & 2 \\
6 & 10 & 14 & 16 & 17 & 18 & 0 & 7 & 7 & 12 & 4 & 13 & 15 \\
7 & 15 & 18 & 18 & 20 & 20 & 13 & 0 & 8 & 16 & 10 & 15 & 15 \\
8 & 15 & 20 & 18 & 18 & 20 & 13 & 10 & 0 & 15 & 11 & 14 & 18 \\
9 & 10 & 9 & 11 & 14 & 19 & 7 & 4 & 7 & 0 & 2 & 10 & 9 \\
10 & 17 & 17 & 18 & 18 & 20 & 16 & 7 & 9 & 15 & 0 & 13 & 14 \\
11 & 10 & 14 & 14 & 10 & 18 & 8 & 4 & 4 & 12 & 7 & 0 & 12 \\
12 & 10 & 12 & 11 & 12 & 17 & 7 & 4 & 4 & 10 & 6 & 8 & 0 \\
\end{array}.$$

Solving the BILP produces an optimal objective value of 351. The solution matrix $\mathbf{X}$ and optimal ranking obtained by sorting the column sums of $\mathbf{X}$ in ascending order are below.

$$
\mathbf{X} =
\begin{array}{c}
1\\2\\3\\4\\5\\6\\7\\8\\9\\10\\11\\12
\end{array}
\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0\\
1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1\\
1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\
1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\
1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1\\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0\\
1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0
\end{pmatrix}
\quad \text{and} \quad
\begin{array}{l}
\text{Davidson}\\
\text{CofC}\\
\text{Citadel}\\
\text{Samford}\\
\text{UT Chatt}\\
\text{Wofford}\\
\text{App State}\\
\text{W. Carolina}\\
\text{Elon}\\
\text{UNC-G}\\
\text{Furman}\\
\text{GA Southern}
\end{array}
\begin{pmatrix}
5\\4\\3\\9\\2\\12\\1\\11\\6\\10\\7\\8
\end{pmatrix}.
$$

3.1.2 Solving Large MVR BILPs

Because the $O(n^3)$ Type 2 constraints drastically limit the size of the ranking problems that we can handle, we use a relaxation trick, called *constraint relaxation*, to increase the size of tractable problems. Of the $n(n-1)(n-2)$ Type 2 constraints, $x_{ij} + x_{jk} + x_{ki} \leq 2$, only a very small proportion of these are necessary. The great majority of these will be trivially satisfied—the problem is that we don't know which are necessary and which are unnecessary. In order to find out, we start by assuming all are unnecessary, then slowly add back in the necessary ones as they are identified. The Type 1 and relaxed continuous version of the Type 3 constraints cause no problems, so we leave these unchanged. Below are the steps involved in the constraint relaxation technique.

CONSTRAINT RELAXATION ALGORITHM FOR LARGE MVR BILPs

1. Relax all Type 2 constraints so that the initial set of necessary Type 2 constraints is empty.
2. Solve the BILP with the current set of necessary Type 2 constraints. Form the optimal ranking associated with this solution. This ranking is actually an approximation to the true ranking we desire for the original problem with the full Type 2 constraint set.
3. Determine which Type 2 constraints are violated by the solution from Step 2— these are *necessary* Type 2 constraints. Add these Type 2 constraints to the set of necessary Type 2 constraints and go to Step 2. Repeat until no Type 2 constraints are violated. The BILP solution at this point is an optimal ranking for the original problem with the full constraint set.

In Step 3, the determination of which Type 2 constraints are violated by the current BILP solution is easy and does not require that each constraint be checked one by one. Recall from Figure 2 that violations to the Type 2 transitivity constraints are *upward arcs* in the *dominance graph*. In yet another view, these violations manifest as ones on

the lower triangular part of the rank reordered matrix $\mathbf{X}$. For each identified upward arc (j, i), next find all k such that $x_{ik} = x_{kj} = 1$ and generate the corresponding Type 2 constraint $x_{ji} + x_{ik} + x_{kj} \leq 2$. The matrix $\mathbf{X}$ can be used to quickly find these elements k: form the Hadamard (element-wise) product of the ith row and j^{th} column of $\mathbf{X}$. All nonzero elements in this product satisfy $x_{ik} = x_{kj} = 1$. Because of the upper triangular structure of the reordered matrix $\mathbf{X}$, it takes much less than $O(n)$ work to compute the Hadamard product and form the transitivity constraints associated with each upward arc.

In addition, we can take advantage of any approximate rankings that may exist. For example, suppose a fast heuristic method (MVR or otherwise) is run and a ranking produced. This ranking has a one-to-one correspondence to a matrix, let's call it $\bar{\mathbf{X}}$, in hillside form. Computing the objective value $f(\bar{\mathbf{X}}) = \mathbf{C}. * \bar{\mathbf{X}}$ for this approximate solution matrix $\bar{\mathbf{X}}$ gives a useful upperbound on the objective. As the branch and bound BILP procedure explores solutions and encounters a branch with nodes exceeding $f(\bar{\mathbf{X}})$, nodes in that branch no longer need to be explored.

In summary, the constraint relaxation technique is an iterative procedure that solves a series of BILPs whose constraint set gradually grows until all the necessary transitivity constraints are identified. At each iteration the optimal BILP solution is an approximation to the true optimal ranking of the original problem with the full constraint set. The approximations improve until the optimal ranking is reached. Table 2 of Section 9 demonstates this well.

3.1.3 Multiple Optimal Solutions for the BILP

The branch and bound procedure terminates with an optimal solution $\mathbf{X}$. As we saw with our small examples, sorting the column sums in ascending order gives the optimal ranking of the items. In this section, we consider two questions: (1) is the optimal solution unique? and (2) if it is not, can we find alternate optimal solutions?

There is a simple test to determine if the optimal solution to the BILP is unique. Consider each successive pair of items in the optimal ranked list and ask if the two items i and j can be swapped without changing the objective value. Only swaps of rank neighboring items need be considered as these are the only swaps that do not violate the constraints, particularly the transitivity constraints. The objective value will not change if $c_{ij} = c_{ji}$. If this is so, then an alternate optimal solution is one that has these two items swapped. There may indeed be more than a two-way tie at this rank position. For instance, a three-way tie occurs if $c_{ij} = c_{ji} = c_{ik} = c_{ki} = c_{jk} = c_{kj}$ for rank neighboring items i, j, and k. Continue down the optimal ranked list in this fashion detecting any two-way or higher ties at each position. We apply this Tie Detection algorithm to the 2009 SoCon example. From Section 3.1.1, the BILP produced the

optimal ranking of

$$
\begin{array}{l|c}
\text{Davidson} & 5 \\
\text{CofC} & 4 \\
\text{Citadel} & 3 \\
\text{Samford} & 9 \\
\text{UT Chatt} & 2 \\
\text{Wofford} & 12 \\
\text{App State} & 1 \\
\text{W. Carolina} & 11 \\
\text{Elon} & 6 \\
\text{UNC-G} & 10 \\
\text{Furman} & 7 \\
\text{GA Southern} & 8
\end{array}.
$$

We begin the Tie Detection algorithm by comparing the two teams at the top of the list. Because $\mathbf{C}(5,4) \neq \mathbf{C}(4,5)$, these two cannot be swapped. Thus, we move onto the next pair of teams in the ranked list, teams 4 and 3. Because $\mathbf{C}(4,3) = \mathbf{C}(3,4)$, these two can be swapped. This means that teams 3 and 4 can appear in either order in the optimal ranked list, with 4 above 3 as the BILP algorithm found or with 3 above 4, which is an alternate optimal solution since the objective function is unchanged yet feasibility is still maintained. At this point, we have discovered a two-way tie between teams 3 and 4, but a three-way or higher tie may exist. So we check to see if the next team in the list, team 9, satisfies $\mathbf{C}(3,4) = \mathbf{C}(4,3) = \mathbf{C}(4,9) = \mathbf{C}(9,4) = \mathbf{C}(3,9) = \mathbf{C}(9,3)$, which it does not. Thus, the tie at the second rank position is indeed only a two-way tie between teams 3 and 4. We continue down the list, considering 9 and 2, then 2 and 12, and so on, and we find one more two-way tie—this time between teams 1 and 11. As a result, this SoCon example has a total of four binary integer optimal solutions, which are shown below.

$$
\begin{pmatrix} 5 \\ 4 \\ 3 \\ 9 \\ 2 \\ 12 \\ 1 \\ 11 \\ 6 \\ 10 \\ 7 \\ 8 \end{pmatrix},
\begin{pmatrix} 5 \\ 3 \\ 4 \\ 9 \\ 2 \\ 12 \\ 1 \\ 11 \\ 6 \\ 10 \\ 7 \\ 8 \end{pmatrix},
\begin{pmatrix} 5 \\ 4 \\ 3 \\ 9 \\ 2 \\ 12 \\ 11 \\ 1 \\ 6 \\ 10 \\ 7 \\ 8 \end{pmatrix}, \text{ and }
\begin{pmatrix} 5 \\ 3 \\ 4 \\ 9 \\ 2 \\ 12 \\ 11 \\ 1 \\ 6 \\ 10 \\ 7 \\ 8 \end{pmatrix}.
$$

In summary, we know that we can (1) apply a branch and bound procedure to find an optimal solution to the MVR BILP, (2) check the uniqueness of the obtained optimal solution, and (3) if applicable, find *several* alternate optimal solutions with the simple $O(n)$ test described above. As a result, this optimization technique produces an output ranking that may actually contain ties and is a very mathematically appealing and provably optimal ranking method. However, the BILP is much slower than many existing rating and ranking methods. In fact, because of the $O(n^3)$ constraints, in

practice, a commercial BILP solver such as the DASH Optimization software or the NEOS server is limited to a problem with n on the order of a few thousand. Thus, ranking all NCAA Division 1 football or basketball teams is certainly within reach while ranking billions of webpages in cyberspace is not. Yet ranking the top 50 results produced by several popular search engines is not only possible, but fast—as it can be done in real-time. Fortunately, the next section explains how we can further increase the practical limit on n, making the MVR ranking of thousands of items possible.

3.2 LP

In this section, we relax the Type 3 constraints that force the variables x_{ij} to be discrete, specifically, binary since $x_{ij} \in \{0, 1\}$, and allow them to be continuous so that $0 \leq x_{ij} \leq 1$. Actually, the upperbound of the bound constraint $0 \leq x_{ij} \leq 1$ is redundant as this restriction is covered by the Type 1 constraint $x_{ij} + x_{ji} = 1$. Thus, the simplified relaxed LP for the MVR problem is

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}\, x_{ij}$$

$$x_{ij} + x_{ji} = 1 \quad \text{for all distinct pairs } (i, j) \qquad \text{(Type 1–antisymmetry)}$$

$$x_{ij} + x_{jk} + x_{ki} \leq 2 \quad \text{for all distinct triples } (i, j, k) \qquad \text{(Type 2–transitivity)}$$

$$x_{ij} \geq 0 \qquad \text{(Type 3–}continuous\text{)}$$

When some BILPs are solved as LPs the optimal solution to the relaxed problem, the LP, gives a solution with binary values, which is clearly also optimal for the BILP. This is the best-case scenario. The next best scenario is when the optimal solution for the LP contains just a small proportion of fractional values. Often, in this case, these few fractional values can be rounded to the nearest integer giving a slightly suboptimal solution that may adequately approximate the exact optimal integer solution. In this section, we show that the LP gives very interesting results. Many times the LP solution is optimal and, in fact, readily tells us all alternate optimal solutions as well.

Our 12-team SoCon example makes this point well. From Section 3.1.3, we discovered that this example has four binary optimal solutions. One with the teams in the rank order given by

$$\begin{pmatrix} 5 & 4 & 3 & 9 & 2 & 12 & 1 & 11 & 6 & 10 & 7 & 8 \end{pmatrix}^{T},$$

another ranking that is identical yet team 3 is above team 4, still another ranking that is identical to the first yet teams 11 is above team 1, and a final ranking with both 3 above 4 and 11 above 1. Notice how these ties are manifested in the LP solution matrix

X below.

$$\mathbf{X} = \begin{array}{cccccccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0.4951 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\
1 & 1 & 0 & 0.5896 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 0.4104 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0.5049 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0
\end{array}.$$

The locations of the fractional values in **X** correspond precisely to the 2 two-way ties. Because the LP solver terminated with an optimal objective value of 351, which is identical to the objective value of the BILP, the LP's fractional optimal solution lies on the boundary created by the integer optimal solutions of the BILP. In this case, an extreme point LP solver such as the simplex method will terminate at one of the four integer optimal solutions, while an interior point LP solver is likely to terminate at one of the infinitely many fractional optimal solutions, from which optimal integer solutions can be built.

Due to the Type 1 constraints, fractional values must always occur in pairs. Each fractional pair occupies mirrored positions about the diagonal of the **X** matrix. A set of fractional pairs is classified as either isolated or non-isolated. If the set of fractional pairs contains no overlapping indices, then it is said to be *isolated*. The 2 two-way tie SoCon example above contained two fractional pairs ($\{3,4\}$ and $\{1,11\}$), which produced an isolated set and hence, $2^2 = 4$ optimal solutions. The next example demonstrates the effect of a *non-isolated* set of fractional pairs. The same set of 12 SoCon teams with slightly different point data created the following **C** matrix.

$$\mathbf{C} = \begin{array}{r|cccccccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\
\hline
1 & 0 & 14 & 15 & 15 & 17 & 11 & 9 & 8 & 12 & 12 & 12 & 11 \\
2 & 9 & 0 & 12 & 13 & 16 & 11 & 9 & 5 & 11 & 10 & 9 & 12 \\
3 & 9 & 12 & 0 & 11 & 16 & 7 & 7 & 9 & 10 & 5 & 9 & 10 \\
4 & 8 & 11 & 13 & 0 & 16 & 9 & 9 & 8 & 11 & 9 & 10 & 11 \\
5 & 7 & 8 & 8 & 8 & 0 & 6 & 3 & 6 & 9 & 5 & 6 & 8 \\
6 & 12 & 13 & 17 & 15 & 18 & 0 & 13 & 12 & 13 & 8 & 12 & 15 \\
7 & 15 & 15 & 17 & 15 & 21 & 11 & 0 & 12 & 17 & 12 & 14 & 15 \\
8 & 16 & 19 & 14 & 16 & 18 & 12 & 12 & 0 & 14 & 11 & 15 & 13 \\
9 & 12 & 13 & 14 & 12 & 15 & 11 & 7 & 10 & 0 & 9 & 11 & 13 \\
10 & 12 & 13 & 19 & 15 & 18 & 15 & 11 & 13 & 15 & 0 & 13 & 16 \\
11 & 12 & 14 & 15 & 14 & 18 & 12 & 10 & 9 & 12 & 11 & 0 & 14 \\
12 & 13 & 12 & 14 & 13 & 16 & 8 & 9 & 10 & 11 & 8 & 10 & 0
\end{array}.$$

The relaxed LP generates the solution matrix

$$
\mathbf{X} = \begin{array}{c c}
& \begin{array}{c c c c c c c c c c c c}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12
\end{array} \\
\begin{array}{c}
1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12
\end{array} &
\left(\begin{array}{c c c c c c c c c c c c}
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.4291 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0.5727 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.5709 & 0.4273 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0
\end{array}\right)
\end{array}.
$$

As in the previous example, there are two fractional pairs: $\{6, 8\}$ and $\{7, 8\}$. However, this time the set of pairs is non-isolated since item 8 is shared. This means that there will be fewer than $2^2 = 4$ integer optimal solutions as some violate the Type 2 feasibility constraint. Table 1 shows that we cannot slide the values in every fractional pair to their extreme values of 0 and 1. In particular, because x_{76} is fixed at 1, the second row

Table 1 Non-isolated set violates feasibility

x_{68}	x_{86}	x_{78}	x_{87}	feasible?
0	1	0	1	yes
1	0	0	1	no
0	1	1	0	yes
1	0	1	0	yes

of the table violates the Type 2 constraint $x_{76} + x_{68} + x_{87} \leq 2$. Thus, there are only three integer optimal solutions for this non-isolated SoCon example.

Though the above two SoCon examples end in fractional solutions from which optimal binary solutions can be constructed, this is not guaranteed. In fact, we constructed a 9-item example with a unique fractional optimal solution. Thus, when binary solutions are constructed, the objective value is not as good as that produced by the unique fractional solution. However, this example with a unique fractional solution was hard to construct. In fact, to locate such an instance, we randomly generated a hundred **C** matrices with entries uniformly distributed between -1 and 1 before we encountered the unique fractional 9-item example. Thus, though unique fractional solutions are possible, it is more likely in practice that the relaxed LP formulation of our MVR problem will result in non-unique fractional solutions from which multiple optimal binary solutions can be constructed. Empirical studies by Reinelt et al. [10], [11] add further evidence that the LP results for ranking problems are exceptionally good and often optimal and binary in practice.

4 Theorem for the Relationship between the BILP and LP

In which cases can we be certain that the optimal LP solution is also optimal for the BILP? Remember, after all, that the BILP is truly the problem of interest for us. Of course, if the LP solution is binary, then that solution is optimal for the BILP. But even when the LP solution is fractional, there are instances in which it is optimal for the BILP. Recall the two fractional optimal SoCon examples from Section 3.2. The theorem of this section identifies conditions on the LP solution that guarantee that this solution is also optimal for the BILP. In addition, the theorem below connects the presence of multiple optimal solutions for the BILP, which indicate the presence of ties in the ranking, to fractional values in the LP solution.

Theorem 2 *Suppose* $\mathbf{X}$, *the fractional optimal solution to the relaxed LP, satisfies the following two conditions:*

1.   The g fractional pairs in $\mathbf{X}$ *form an isolated set. Due to the Type 1 antisymmetry constraint, each fractional pair can be expressed as* $x_{ij} = \alpha$, $x_{ji} = 1 - \alpha$, *for* $0 < x_{ij} < 1$.
2.   For each fractional pair (i, j), *the corresponding elements in the cost matrix satisfy* $c_{ij} = c_{ji}$.

Then 2^g *binary solutions that are optimal for the BILP can be built from* $\mathbf{X}$ *by setting the fractional parameter* α *of each fractional pair to its two possible extreme values of 0 and 1.*

Proof The matrix $\mathbf{X}$ is the optimal solution of the relaxed LP, which occurs when the integer constraint $x_{ij} \in \{0, 1\}$ of the BILP is relaxed to $x_{ij} \in [0, 1]$. We will show that the solution $\mathbf{X}$ can be written as a convex combination of binary matrices that are feasible and optimal for both LP and the BILP. As a result, $\mathbf{X}$ is not an optimal *extreme* point of the feasible region but rather an optimal *boundary* point. Note that we use "boundary point" to mean a non-extreme boundary point.

CASE of $g = 1$**:** We begin with the simplest case, with just $g = 1$ fractional pair. Without loss of generality, assume that the fractional values occur at x_{ij} and x_{ji}. For $g = 1$, we will show that $\mathbf{X}$ can be written as a convex combination of two binary matrices $\mathbf{Y}$ and $\mathbf{Z}$. Let $\mathbf{Y}$ and $\mathbf{Z}$ be defined so that they share all the elements in $\mathbf{X}$ except $y_{ij} = 1$ and $y_{ji} = 0$ and $z_{ij} = 0$ and $z_{ji} = 1$. In other words, $\mathbf{Y}$ and $\mathbf{Z}$ can be thought of as the two only possible binary ("rounded") versions of $\mathbf{X}$ that still satisfy the Type 1 antisymmetry constraint. As a result, $\mathbf{X} = x_{ij} \ \mathbf{Y} + x_{ji} \ \mathbf{Z} = x_{ij} \ \mathbf{Y} + (1 - x_{ij}) \ \mathbf{Z}$, which clearly shows that $\mathbf{X}$ is a convex combination of the binary matrices $\mathbf{Y}$ and $\mathbf{Z}$.

Next we show that $\mathbf{Y}$ and $\mathbf{Z}$ are both feasible and optimal for the BILP. First, feasibility. Clearly, $\mathbf{Y}$ and $\mathbf{Z}$ satisfy the Type 3 binary constraint for the BILP. It is also trivial to check that $\mathbf{Y}$ and $\mathbf{Z}$ satisfy the Type 1 antisymmetry constraint. It takes a bit more work to show that the remaining constraint, the Type 2 transitivity constraint, is satisfied. The LP solution $\mathbf{X}$ satisfies transitivity so that $x_{ij} + x_{jk} + x_{ki} \leq 2$. Because $\mathbf{Y}$ only differs from $\mathbf{X}$ in the (i, j) and (j, i) elements, we only need to check that $y_{ij} + x_{jk} + x_{ki} \leq 2$ and $y_{ji} + x_{ik} + x_{kj} \leq 2$. Since $0 < x_{ij} < 1$ and x_{jk} and x_{ki} are binary, this implies that $x_{jk} + x_{ki} \leq 1$. Thus, $y_{ij} + x_{jk} + x_{ki} = 1 + x_{jk} + x_{ki} \leq 2$, establishing transitivity. Similarly, $y_{ji} + x_{ik} + x_{kj} = 0 + x_{ik} + x_{kj} \leq 2$. The same argument is used to show that $\mathbf{Z}$ is also feasible for the BILP.

Lastly, we show that $\mathbf{Y}$ and $\mathbf{Z}$ are optimal for the BILP. We use the notation $f(\mathbf{X}) = \mathbf{C}. * \mathbf{X}$ to represent the Hadamard (element-wise) product of the coefficient matrix $\mathbf{C}$ with a solution matrix $\mathbf{X}$. Thus, $f(\mathbf{X})$ is the objective function value of solution matrix $\mathbf{X}$. To prove that $\mathbf{Y}$ and $\mathbf{Z}$ are optimal for the BILP, we will show that $f(\mathbf{X}) = f(\mathbf{Y}) = f(\mathbf{Z})$. Both $\mathbf{Y}$ and $\mathbf{Z}$ are feasible to LP, hence $f(\mathbf{X}) \leq f(\mathbf{Y})$ and $f(\mathbf{X}) \leq f(\mathbf{Z})$. By the linearity of the objective function and the construction of $\mathbf{Y}$ and $\mathbf{Z}$, we see that $f(\mathbf{X}) = x_{ij} \, f(\mathbf{Y}) + (1 - x_{ij}) \, f(\mathbf{Z})$. Note that $x_{ij} > 0$ and $1 - x_{ij} > 0$. Suppose one or both of $f(\mathbf{Y})$ and $f(\mathbf{Z})$ are greater than $f(\mathbf{X})$, then we have

$$f(\mathbf{X}) = x_{ij} \, f(\mathbf{Y}) + (1 - x_{ij}) \, f(\mathbf{Z}) > x_{ij} \, f(\mathbf{X}) + (1 - x_{ij}) \, f(\mathbf{X}) = f(\mathbf{X}).$$

This is a contradiction. Therefore, we know that $f(\mathbf{X}) = f(\mathbf{Y}) = f(\mathbf{Z})$.

As a result, the fractional LP solution is actually a non-extreme boundary optimal solution on the boundary between the two alternate optimal *binary* solutions $\mathbf{Y}$ and $\mathbf{Z}$. In this case, for $g = 1$ we have constructed $2^g = 2$ binary optimal solutions for the BILP from the fractional LP solution.

CASE of $g = 2$: Assume, without loss of generality, that x_{ij}, x_{ji}, x_{kl} and x_{lk} are the only fractional values in $\mathbf{X}$. Let $\mathbf{Y}$ and $\mathbf{Z}$ be defined as above and create two additional matrices $\mathbf{S}$ and $\mathbf{T}$ that are identical to $\mathbf{X}$ except $s_{kl} = 1$ and $s_{lk} = 0$ and $t_{kl} = 0$ and $t_{lk} = 1$. Then $\mathbf{X}$ can be written as a convex combination of the binary matrices $\mathbf{Y}$, $\mathbf{Z}$, $\mathbf{S}$, and $\mathbf{T}$.

$$\mathbf{X} = \frac{x_{ij}}{2} \, \mathbf{Y} + \frac{1 - x_{ij}}{2} \, \mathbf{Z} + \frac{x_{kl}}{2} \, \mathbf{S} + \frac{1 - x_{kl}}{2} \, \mathbf{T}.$$

Next we show that these four matrices are feasible for the BILP. It is trivial to show that they each satisfy the Type 1 antisymmetry and Type 3 binary constraints. In order to show Type 2 transitivity is satisfied, we use precisely the same logic as the $g = 1$ case. It is the isolated nature of the set of fractional pairs that enables us to prove transitivity.

Lastly, we show that $\mathbf{Y}$, $\mathbf{Z}$, $\mathbf{S}$, and $\mathbf{T}$ are optimal for the BILP by showing that the objective value at each of the four binary solution matrices match the objective value at $\mathbf{X}$. From the $g = 1$ case, we have already shown that $f(\mathbf{X}) = f(\mathbf{Y}) = f(\mathbf{Z})$. It remains to show that $f(\mathbf{X}) = f(\mathbf{S}) = f(\mathbf{T})$. Both $\mathbf{S}$ and $\mathbf{T}$ are feasible to LP, hence $f(\mathbf{X}) \leq f(\mathbf{S})$ and $f(\mathbf{X}) \leq f(\mathbf{T})$. By the linearity of the objective function and the construction of $\mathbf{S}$ and $\mathbf{T}$, we see that $f(\mathbf{X}) = x_{kl} \, f(\mathbf{S}) + (1 - x_{kl}) \, f(\mathbf{T})$. Note that $x_{kl} > 0$ and $1 - x_{kl} > 0$. Suppose one or both of $f(\mathbf{S})$ and $f(\mathbf{T})$ are greater than $f(\mathbf{X})$, then we have

$$f(\mathbf{X}) = x_{kl} \, f(\mathbf{S}) + (1 - x_{kl}) \, f(\mathbf{T}) > x_{kl} \, f(\mathbf{X}) + (1 - x_{kl}) \, f(\mathbf{X}) = f(\mathbf{X}).$$

This is a contradiction. Therefore, we know that $f(\mathbf{X}) = f(\mathbf{S}) = f(\mathbf{T})$.

For $g = 2$ we have constructed $2^g = 4$ binary optimal solutions for the BILP from the fractional LP solution.

GENERAL CASE: This constructive logic can be applied for any number of fractional pairs g. For each fractional pair, a pair of matrices are formed. Each matrix is identical to $\mathbf{X}$ except in the two locations corresponding to that particular fractional pair. The coefficients in the convex combination are simply the fractional value itself divided by g, e.g., $\frac{x_{ij}}{g}, \frac{1 - x_{ij}}{g}, \frac{x_{kl}}{g}, \frac{1 - x_{kl}}{g}, \frac{x_{ik}}{g}, \frac{1 - x_{ik}}{g}, \ldots$. Thus, matrices can always be created so that the optimal LP solution $\mathbf{X}$ can be written as a convex combination.

And each matrix in the convex combination can be shown to be both feasible and optimal for the BILP.

As a consequence, **X** is not an extreme point of the feasible region. Instead, it is a non-extreme boundary point. In fact, it is on the boundary formed by the convex hull of the matrices in the convex combination. Since **X** is optimal, so are all other points on this boundary including the points defined by the matrices in the convex combination.

Theorem 2 carries computational consequences as well. The original LP solver, the famous simplex method, is not the method of choice for solving our MVR problem. The simplex method is an extreme point method, meaning that it moves from one extreme point to the next, in an ever-improving direction until it reaches an optimal solution, which will, of course, also be an extreme point. In contrast, interior point LP solvers move through the interior of the feasible region until they converge on an optimal solution that may be an extreme point or a boundary point depending on the path taken through the interior of the feasible region. For us, it is the non-extreme boundary optimal points (which contain fractional values) that give us so much more information than extreme optimal points (which are the integer-only solutions). Thus, we always choose a non-extreme point, non-simplex LP solver when solving the relaxed LP associated with our MVR problem. Finally, we note that the same constraint relaxation technique of p. 8 that we advocated for solving large MVR BILPs can be used to solve large MVR LPs.

5 Sensitivity Analysis

Another advantage of the LP over the BILP relates to the natural sensitivity measures produced when solving a linear program. In this case, we are interested in changes in the objective coefficients c_{ij}. Slight changes in the input data (specifically the differential matrix **D** that creates the objective coefficient matrix **C**) could change the optimal solution, and hence, optimal ranking of the teams. The Xpress-MP optimization software computes the following ranges on the objective coefficients for the 12-team SoCon example.

```
            1           2           3           4           5           6           7           8           9          10          11          12
----------------------------------------------------------------------------------------------------------------------------------------------
 1 : [   0,   0][   9, Inf][   6, Inf][   5, Inf][   2, Inf][-Inf,   9][-Inf,  15][-Inf,  15][   9,  10][-Inf,  17][   9, Inf][  10, Inf]
 2 : [-Inf,  14][   0,   0][  10, Inf][   9, Inf][   3, Inf][-Inf,  13][-Inf,  20][-Inf,  20][  11,  12][-Inf,  18][-Inf,  13][-Inf,  10]
 3 : [-Inf,  14][-Inf,  11][   0,   0][-Inf,   9][   5, Inf][-Inf,  17][-Inf,  19][-Inf,  19][-Inf,  14][-Inf,  18][-Inf,  15][-Inf,  14]
 4 : [-Inf,  14][-Inf,  13][   9, Inf][   0,   0][   4, Inf][-Inf,  17][-Inf,  20][-Inf,  19][-Inf,  16][-Inf,  18][-Inf,  11][-Inf,  14]
 5 : [-Inf,  17][-Inf,  16][-Inf,  14][-Inf,  14][   0,   0][-Inf,  17][-Inf,  21][-Inf,  22][-Inf,  18][-Inf,  20][-Inf,  19][-Inf,  17]
 6 : [   8, Inf][   7, Inf][   5, Inf][   5, Inf][   3, Inf][   0,   0][-Inf,  14][-Inf,  11][   8, Inf][-Inf,  16][   8, Inf][   7, Inf]
 7 : [   4, Inf][   2, Inf][   1, Inf][   0, Inf][   0, Inf][   6, Inf][   0,   0][   7,   9][   3, Inf][   7, Inf][   1, Inf][   2, Inf]
 8 : [   4, Inf][   3, Inf][   3, Inf][   1, Inf][   0, Inf][   9, Inf][   9,  11][   0,   0][   6, Inf][-Inf,  11][   3, Inf][   3, Inf]
 9 : [   9,  10][   8,   9][   6, Inf][   4, Inf][   1, Inf][-Inf,  12][-Inf,  17][-Inf,  15][   0,   0][-Inf,  14][  10, Inf][   9,  10]
10 : [   2, Inf][   3, Inf][   2, Inf][   3, Inf][   1, Inf][   4, Inf][-Inf,   9][   9, Inf][   3, Inf][   0,   0][   5, Inf][   5, Inf]
11 : [-Inf,  11][   8, Inf][   4, Inf][   5, Inf][   0, Inf][-Inf,  13][-Inf,  18][-Inf,  16][-Inf,  11][-Inf,  15][   0,   0][-Inf,  10]
12 : [-Inf,  11][  10, Inf][   6, Inf][   5, Inf][   2, Inf][-Inf,  15][-Inf,  17][-Inf,  19][   9,  10][-Inf,  15][  10, Inf][   0,   0]
```

Fig. 3 Sensitivity ranges on the objective coefficients c_{ij}

Most c_{ij} coefficients have very loose bounds. The exceptions are the pairs of 1 and 9, 2 and 9, and 9 and 12, which have tight bounds. For this dataset, these ranges warn us that we are less certain of the rank ordering of teams in the middle of the

pack. Changes of the objective coefficients outside of the given ranges, can change the ranking. On the other hand, we have more confidence in our ranking of teams at the top and bottom of the list.

6 Bounding Techniques to Improve Convergence

Bounding techniques are extremely useful in optimization. In particular, solution techniques for integer programs rely heavily on bounding. In this section, we apply such bounding to the Iterative LP method to accelerate convergence and produce optimality guarantees. Recall that the Iterative LP method relaxes the transitivity constraint set. In fact, at the first iteration, the LP is solved with no transitivity constraints. The optimal solution matrix $\mathbf{X}$ at this iteration creates an objective function value that we denote $\underline{f}$ since it is a lowerbound of the optimal objective value f^*. The solution at this iteration almost always violates some transitivity constraints and thus is not a feasible solution for the original LP. However, it can be used to form a very useful approximate solution. This is done by computing the row sums of $\mathbf{X}$. The ith row sum is a good indicator of how many opponents the ith team will beat. As the iterative LP method proceeds, we use the row sums of each iteration's solution matrix $\mathbf{X}$ to compute an approximate ranking. This approximation gets closer to the optimal ranking as the iterations proceed. Because every ranking, including these approximate ones, has a one-to-one correspondence with an $\mathbf{X}$ matrix, we can compute the objective function value for each approximate ranking, which we denote $\overline{f}$ since it an upperbound of f^*. Thus, we know that

$$\underline{f} \leq f^* \leq \overline{f}.$$

Next we bound the relative error $\frac{\overline{f} - f^*}{f^*}$ associated with the approximate ranking. Because $\underline{f}$, f^*, and $\overline{f}$ all have the same sign, we can bound the relative error, which involves the unknown f^* with the known quantities $\underline{f}$ and $\overline{f}$.

$$\frac{\overline{f} - f^*}{|f^*|} \leq \frac{\overline{f} - \underline{f}}{|f^*|} \leq \frac{\overline{f} - \underline{f}}{|\overline{f}|}.$$

Further, when all elements of the objective coefficient matrix $\mathbf{C}$ are integral and $\overline{f} - \underline{f} \leq 1$, the approximate ranking is the optimal solution of the original ranking problem. Even when we are no so lucky as to be able to guarantee optimality, we can give a guarantee on the error associated with the near-optimal solution. We can guarantee that the optimal objective value is between $\underline{f}$ and $\overline{f}$ and the relative error is not greater than $\frac{\overline{f} - \underline{f}}{|\overline{f}|}$. This is another great advantage of the bounding version of our Iterative LP method. Not only does it require fewer iterations but it also allows the user to stop the iterative procedure as soon as an acceptable relative error is reached.

7 Linear Ordering Polytope

The convex hull of all of the binary integer points satisfying the constraints of our MVR BILP has been well-studied and is called the linear ordering polytope [13]. It is helpful to study the relationship of the LP's polytope to the feasible region of the

original BILP. Of course, the feasible region of the BILP is contained within the feasible region of the LP. The best scenario is when the LP's feasible region is as tight as possible to the BILP's feasible region. In other words, the LP's feasible region is the convex hull of the points in the BILP's feasible region. For our MVR problem, the good news is that all of the inequality constraints (Type 2 transitivity and Type 3 nonnegativity) are facet-defining inequalities for the linear ordering polytope. This means that *these* inequalities are as tight as possible. However, the set of constraints for the LP does not cover all facet-defining inequalites for the linear ordering polytope. Sophisticated valid inequalities such as the so-called fence and Möbius ladder inequalities create stronger LP relaxations, but unfortunately they are too costly to generate [8], [10], and [12].

8 Connection to Rank Aggregation

Our MVR optimization formulation is a special case of a related ranking problem, the rank aggregation problem. In rank aggregation, k rankings must be aggregated into one unified ranking. The only difference between the BILP optimization formulations of our MVR method and the rank aggregation method is in the definition of the $\mathbf{C}$ matrix of objective coefficients. In rank aggregation, we define c_{ij} as a measure of the conformity between items i and j. One conformity definition uses $c_{ij} = (\# \text{ lists}$ having i above j). In order to parallel the MVR formulation which was a minimization problem, here for rank aggregation we also use a minimization formulation. Thus, we minimize the negation of conformity. The advantage is that all of our findings on the relationship of the BILP to the LP, the existence and discovery of alternate optimal solutions, and the discovery of ties applies to the rank aggregation problem as well. There is, however, one practical difference between the MVR problem and the rank aggregation problem. Compared to the MVR problem, the rank aggregation problem typically requires many fewer iterations of the Iterative BILP (or LP) Method. In fact, our experiments show that when the lists to be aggregated are largely in agreement from the start, then only a handful of iterations are required.

9 Computational Examples

9.1 Disconnected Items

Here we consider an unusual case: ranking items from two disparate groups, from which we have no overlapping data. For example, the data matrix $\mathbf{D}$ below contains point scores from teams in two separate conferences. Conference A consists of teams 1, 2, and 3, while the Conference B contains teams 4 through 8.

$$\mathbf{D} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array} \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ \begin{pmatrix} 0 & 1 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 & 3 & 4 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}.$$

The MVR method can be applied to each conference separately, which produces the unique optimal ranking of teams in Conference A as $\mathbf{r}_A = (\,1\quad 2\quad 3\,)$ and the unique optimal ranking of teams in Conference B as $\mathbf{r}_B = (\,4\quad 5\quad 6\quad 7\quad 8\,)$. However, we are interested in the behavior of the MVR method when applied to the full problem of all eight teams.

For the full 8-team case, the MVR method produces objective values for the BILP and the relaxed LP that are identical. Thus, we can conclude that the relaxed LP solution lives on the boundary between several optimal BILP solutions. The solution matrix $\mathbf{X}$ below of the relaxed LP has several fractional values, which are denoted by the * symbol. These fractional values indicate that there are several ties, from which several optimal solutions can be constructed.

$$
\mathbf{X} =
\begin{array}{c}
\\1\\2\\3\\4\\5\\6\\7\\8
\end{array}
\begin{array}{c}
\begin{array}{cccccccc}1&2&3&4&5&6&7&8\end{array}\\
\left(\begin{array}{cccccccc}
0 & 1 & 1 & 0 & * & 1 & 1 & 1\\
0 & 0 & 1 & 0 & 0 & * & 1 & 1\\
0 & 0 & 0 & 0 & 0 & 0 & * & 1\\
1 & 1 & 1 & 0 & 1 & 1 & 1 & 1\\
* & 1 & 1 & 0 & 0 & 1 & 1 & 1\\
0 & * & 1 & 0 & 0 & 0 & 1 & 1\\
0 & 0 & * & 0 & 0 & 0 & 0 & 1\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right)
\end{array}.
$$

This solution creates an optimal ranking of $(\,4\quad 1\quad 5\quad 6\quad 2\quad 7\quad 3\quad 8\,)$. The location of the fractional values indicates the presence of three isolated two-way ties. Nodes 1 and 5 may be swapped. Nodes 2 and 6 can be swapped and nodes 3 and 7 can be swapped. Thus, in total, there are $2^3 = 8$ optimal binary solutions for this problem. Perhaps most interesting in this example is that the optimal ordering of each conference is maintained in all optimal solutions of the full problem. That is, 1 is always above 2 is always above 3, and these are interleaved with the optimal ordering from the other conference.

Just for comparison sake, the popular ranking method of Massey [7] cannot be executed on a disconnected system.[4] Another popular ranking method, the Colley method [4], is computable for disconnected systems, however, it does not produce an output ranking with ties, which, it could be argued in this example, is less appropriate than the MVR ranking which has ties.

9.2 Large n

In order to demonstrate the size of the LPs to which the MVR ranking techniques of this paper can be applied, we ranked the 347 teams in NCAA college basketball for the 2008-2009 season. To solve this large MVR LP, we used the same iterative constraint relaxation trick that we presented in Section 3.1.2 to solve large MVR BILPs. We used the conditions of Theorem 2 and our computational and bounding results to conclude that the Iterative LP method produced a non-unique fractional solution that, when converted to a binary solution using the "rounding" rules of p. 14, is optimal for the original BILP. Just .066% of the nonzero values in the optimal LP solution are

---

[4] In this case, the coefficient matrix for the Massey system is not full rank, and thus, is not invertible.

fractional. In addition, the fractional values, and hence, the ties, occured in positions of lower rank, particularly rank positions 252 through 272.

Table 2 shows the breakdown of how much time is spent at each iteration of the Iterative LP method for the full 347-team example. For example, at `iteration 1`, solving the LP required 4.11 seconds and produced an objective value of 1171616, while finding the necessary Type 2 constraints required .24 seconds and generated 11230 additional constraints to be added to the LP formulation to be solved at Iteration 2. In total, executing all 5 iterations and generating all 18,926 constraints required just 20.90 seconds on a laptop machine. Another observation from Table 2 concerns the remarkable value of the constraint relaxation technique described on p. 8. Just .046% of the total original Type 2 constraints are necessary. This is a huge savings and makes even larger ranking problems within reach. One final observation from Table 2 is in order. Notice that by `iteration 3`, the Iterative LP method has reached a solution that is on the optimal face of the feasible region, yet is infeasible. At each subsequent iteration, the solution is improved in terms of feasibility, not optimality. In other words, the solutions remain on the optimal hyperplane yet move closer to the feasible region at each iteration.

Table 2 Computational Results for Iterative LP method *with bounding* on 347-team example

iteration	LP time	Obj. value	best rank	ConGen time	# con.added
1	4.11	1171616.00	1172359.00	0.24	11230
2	3.70	1172002.00	1172338.00	0.22	6887
3	4.17	1172023.00	1172069.00	0.11	560
4	4.12	1172023.00	1172039.00	0.11	249
5	4.13	1172023.00	1172023.00		
total	20.23			.67	18926

Our experiments show that the bounding technique brings two very significant advantages. First, using the bounding version of the algorithm significantly reduces the overall run time. Second, the bounding version terminates with a solution that is optimal or within some reported percentage of optimal. For example, for another 347-team dataset, the bounding algorithm terminated with a solution that was not proven to be optimal, yet was guaranteed to be very near the optimal solution since the relative error is no greater than .000422%. We were ultimately able to conclude that this solution is indeed optimal since the Iterative BILP method returned the same objective function value as the Iterative LP method.

10 Conclusion and Future Work

We have created a Minimum Violations Rating (MVR) method for ranking items. Our method provides three new contributions to the MVR literature. First, it produces an optimal ranking of the items that minimizes not only upset violations but also weak win violations. Second, unlike most other MVR rankings, our MVR optimal ranking may include ties. Third, our MVR method can identify alternate optimal rankings, if they exist. One drawback of our MVR method concerns scalability. Because the model formulation requires $O(n^3)$ constraints, there is a practical limit on n, the number of items being ranked. We proposed a few solutions to this computational issue, including

a constraint relaxation and a bounding technique. Both performed very well on the 347-item examples from NCAA basketball that we tested. Finally, we proved conditions on which the optimal solution of the relaxed LP is optimal for the original BILP. As future work, we plan to explore sensitivity analysis beyond the basic results described here.

References

1. Iqbal Ali, Wade D. Cook, and Moshe Kress. On the minimum violations ranking of a tournament. *Management Science*, 32(6):660–672, 1986.
2. C. Richard Cassady, Lisa M. Maillart, and Sinan Salman. Ranking sports teams: A customizable quadratic assignment approach. *INFORMS: Interfaces*, 35(6):497–510, 2005.
3. B. Jay Coleman. Minimizing game score violations in college football rankings. *INFORMS: Interfaces*, 35(6):483–496, 2005.
4. Wesley N. Colley. Colley's bias free college football ranking method: The colley matrix explained, 2002.
5. John G. Kemeny. Mathematics without numbers. *Daedalus*, 88(4):577–591, 1959.
6. Antoon W. J. Kolen and Jan Karel Lenstra. Combinatorics in operations research. pages 1875–1910, 1995.
7. Kenneth Massey. Statistical models applied to the rating of sports teams. Bachelor's thesis, Bluefield College, 1997.
8. Alantha Newman and Santosh Vempala. Fences are futile: On relaxations for the linear ordering problem. *Lecture Notes in Computer Science*, 2081:333–347, 2001.
9. Juyong Park. On minimum violations ranking in paired comparisons, Nov 2005.
10. Gerhard Reinelt. *The Linear Ordering Problem: Algorithms and Applications*. Heldermann Verlag, 1985.
11. Gerhard Reinelt, M. Grötschel, and M. Jünger. Optimal triangulation of large real world input-output matrices. *Statistical Papers*, 25(1):261–295, 1983.
12. Gerhard Reinelt, M. Grötschel, and M. Jünger. A cutting plane algorithm for the linear ordering problem. *Operations Research*, 32(6):1195–1220, 1984.
13. Gerhard Reinelt, M. Grötschel, and M. Jünger. Facets of the linear ordering polytope. *Mathematical Programming*, 33:43–60, 1985.