Ranking with Optimization Techniques

March 21, 2010
SIAM-SEAS

Amy Langville
Mathematics Department
College of Charleston
langvillea@cofc.edu

Ranking with Optimization Techniques

March 21, 2010
SIAM-SEAS

Kathryn Pedings
Mathematics Department
College of Charleston
kepeding@cofc.edu

Yoshi Yamamoto
Phil Opt Group
University of Tsukuba
yamamoto@sk.tsukuba.ac.jp

Amy Langville
Mathematics Department
College of Charleston
langvillea@cofc.edu

Outline

- Minimum Violations Ranking
  - data differential matrices
  - hillside form
- Evolutionary Optimization
- BILP
- LP
- Applications

Data Differential Matrices

data on pairwise comparisons between items (e.g., teams)

Data:

- points
- rebounds
- fge
- turnovers

Points Examples:

- d_{ij} = 10 and d_{ji} = 0 , if I beats j by 10
- d_{ij} = 10 and d_{ji} = -10 , if I beats j by 10

For multiple games between teams, take average or cumulative differentials.

Hillside Form

Perfect Hillside Form:

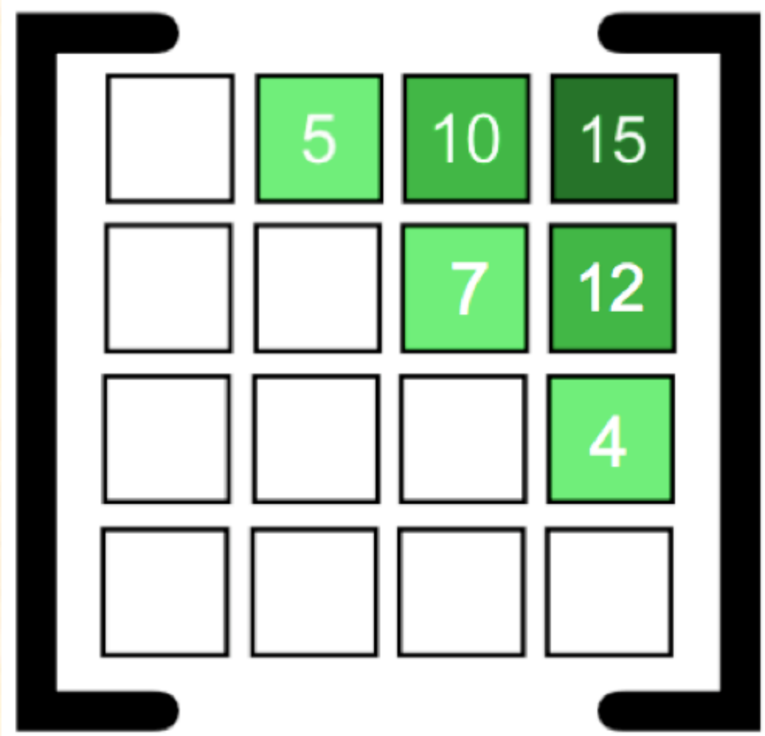
- each row is an increasing sequence
- each column is an decreasing sequence
- (optional) zeros on lower triangular

Hillside Form

Perfect Hillside Form:

- each row is an increasing sequence
- each column is an decreasing sequence
- (optional) zeros on lower triangular

Example of perfect hillside form



Mathematically, P_{mxn} is in perfect hillside form if:

$$P_{i,j} = 0 \quad \forall\ i \le j : i \in \{1,\ldots,m\}, j \in \{1,\ldots,n\}$$
$$P_{i,k} \le P_{j,k} \quad \forall\ i < j : i \in \{1,\ldots,m\}, j \in \{1,\ldots,m\}, k \in \{1,\ldots,n\}$$
$$P_{k,i} \le P_{k,j} \quad \forall\ i < j : i \in \{1,\ldots,n\}, j \in \{1,\ldots,n\}, k \in \{1,\ldots,m\}$$

Hillside Form and Ranking

In a perfect season, the 1st place team beats the 2nd place team by a little, the 3rd place team by a little more, and so on.

Hillside Form and Ranking

In a perfect season, the 1st place team beats the 2nd place team by a little, the 3rd place team by a little more, and so on.

GOAL: symmetrically reorder rows and columns of data matrix to bring it as close to hillside form as possible.

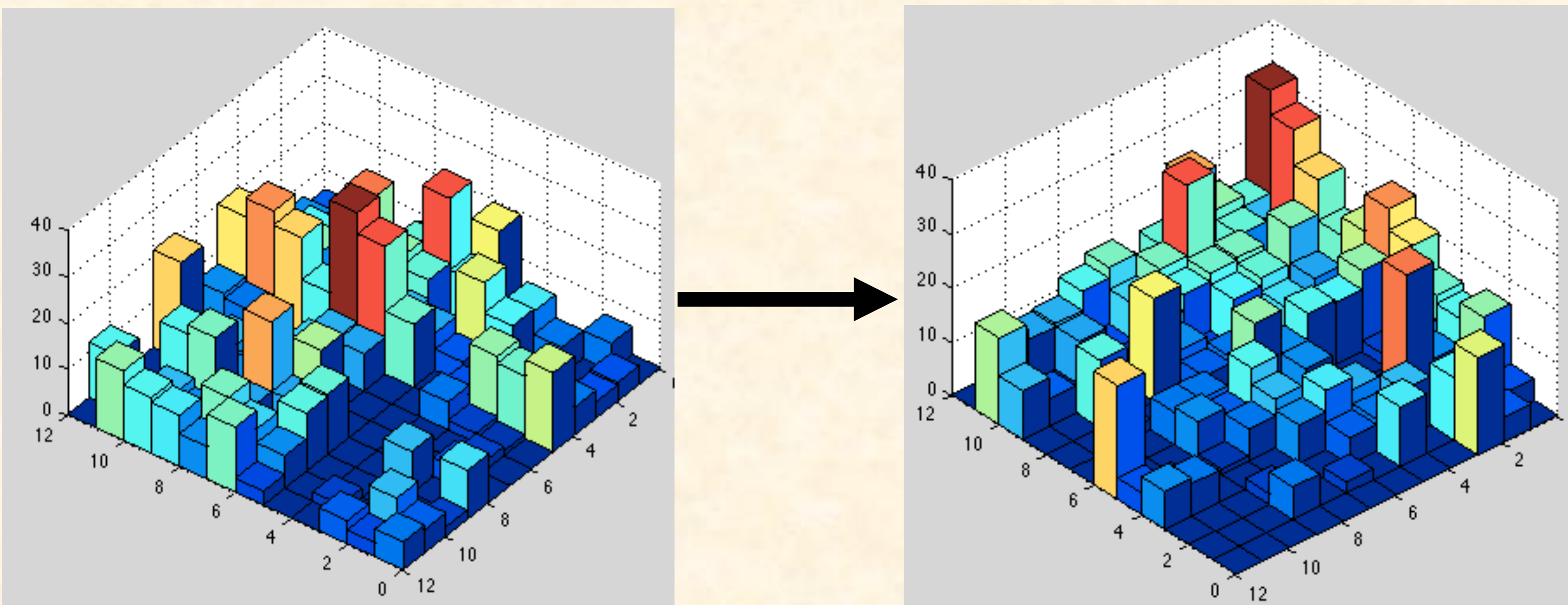
Reordering that does this = ranking vector

Hillside Form and Ranking

2009 SoCon BB teams

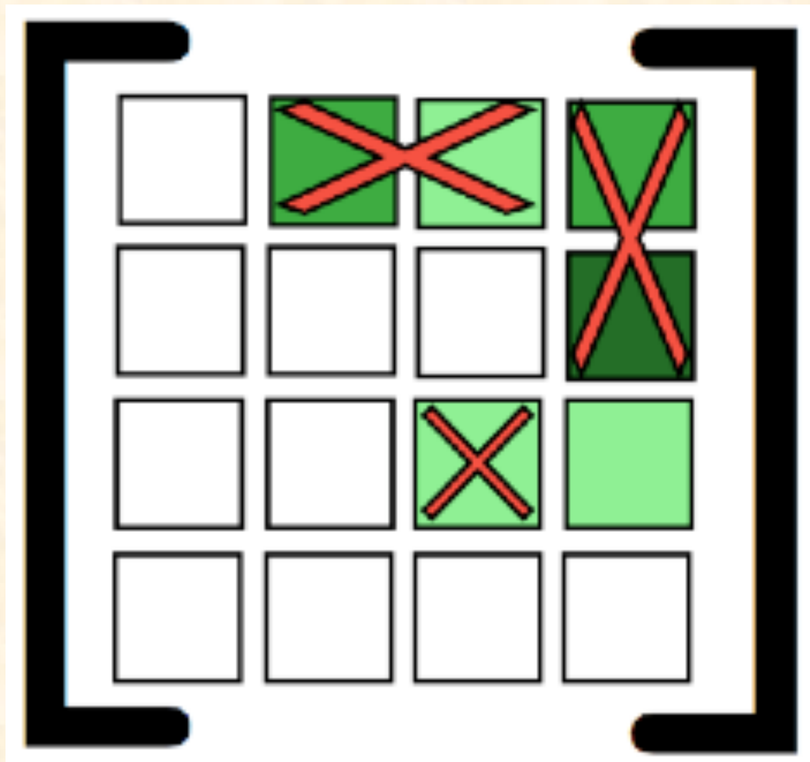
Original ordering

after Reordering



Violations to Hillside Form

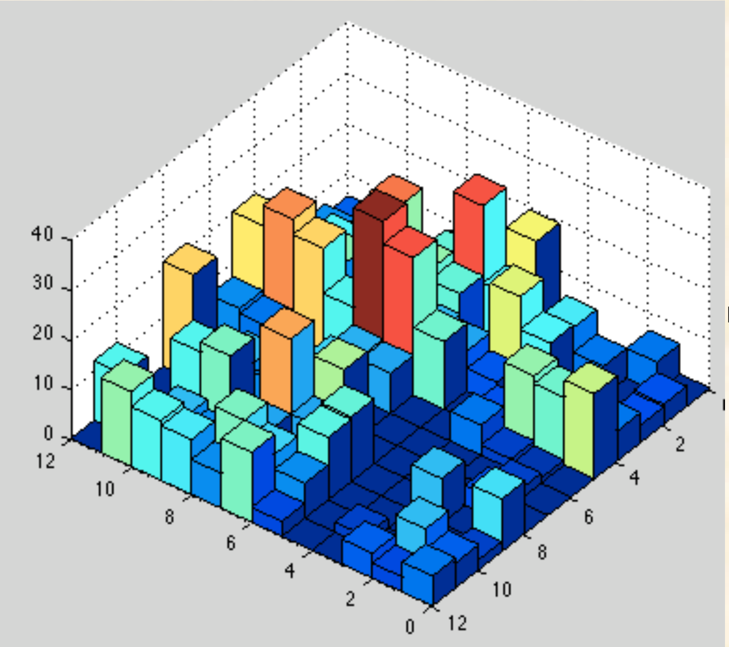
for a reordering (aka, ranking vector), count the number of violations to hillside form



Violations to Hillside Form

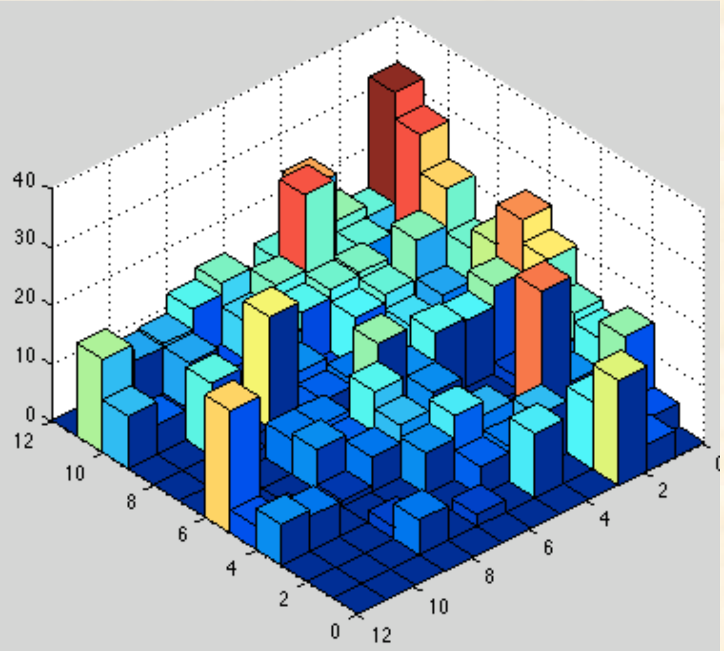
for a reordering (aka, ranking vector), count the number of violations to hillside form

Original ordering



violations = 1329

after Reordering



violations = 351

Hillside Form and Ranking

In a perfect season, the 1st place team beats the 2nd place team by a little, the 3rd place team by a little more, and so on.

GOAL: symmetrically reorder rows and columns of data matrix to bring it **as close to hillside form as possible**.

Reordering that does this = ranking vector

Optimization!

Solving the Optimization Problem

1. EO (evolutionary optimization)
2. BILP (binary integer linear program)
3. LP (linear program)

Evolutionary Optimization

Of all n! permutation vectors, find the one that minimizes the number of hillside violations.

Evolutionary Optimization

Of all n! permutation vectors, find the one that minimizes the number of hillside violations.

Every ranking vector is a permutation vector.

A New Algorithm for Ranking Sports Teams Using Evolutionary Optimization

Kathryn Pedings (kathryn.pedings@gmail.com) and Dr. Amy Langville
College of Charleston   Department of Mathematics

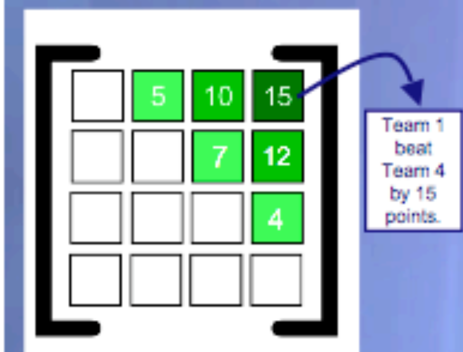
COLLEGE of CHARLESTON
THE GRADUATE SCHOOL

Abstract

Evolutionary optimization is an algorithm using Darwinian ideas of mating, mutating, fitness, and survival of the fittest. Its use has been limited to intractable problems, but it was our belief that the algorithm could be modified to be successful with tractable problems such as sports ranking.

1. Never before used on sports ranking.
2. We have shown that with some datasets our evolutionary algorithm is better than other well-known algorithms.
3. Sophisticated changes made to speed up the algorithm.

Hillside Form and Point-Differential Matrices



Initial Population

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 4 & 2 & 2 & 3 \\ 1 & 1 & 4 & 4 \\ 3 & 4 & 3 & 2 \end{bmatrix}$$

Calculate the Fitness

Get an Offspring

Mating

$$\begin{bmatrix} 2 \\ 4 \\ 1 \\ 3 \end{bmatrix} \text{ and } \begin{bmatrix} 3 \\ 2 \\ 1 \\ 4 \end{bmatrix} \rightarrow \begin{bmatrix} 2 \\ 3 \\ 1 \\ 4 \end{bmatrix}$$

Mutating

$$\begin{bmatrix} 1 \\ 2 \\ 4 \\ 3 \end{bmatrix} \rightarrow \begin{bmatrix} 4 \\ 2 \\ 1 \\ 3 \end{bmatrix}$$

Then calculate the fitness one step at a time.

Fitness is Bad

Fitness is Good

Population Changes Drastically

$$\begin{bmatrix} 1 & 3 & 2 & 4 \\ 3 & 2 & 1 & 3 \\ 2 & 4 & 3 & 1 \\ 4 & 1 & 4 & 2 \end{bmatrix}$$

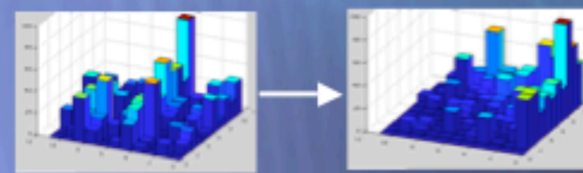
Population Changes Very Little

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 3 & 3 & 2 & 4 \\ 2 & 4 & 3 & 3 \\ 4 & 2 & 4 & 2 \end{bmatrix}$$

STOP! Print out the best parent solution; this is your team ranking!

Results

The dataset used is the basketball data for the Southern Conference for the 2007-2008 season.



Massey Ranking	Evolutionary Optimization	Colley Ranking
# of Viol = 269	# of Viol = 265	# of Viol = 278
Davidson	Davidson	Davidson
UNC - G	UNC - G	App St.
GA So.	GA So.	Chatt.
App St.	App St.	GA So.
Chatt.	Chatt.	UNC - G
CofC	Elon	Elon
Elon	CofC	CofC
Wofford	W. Car.	Wofford
W. Car.	Wofford	Furman
Furman	Furman	W. Car.
Citadel	Citadel	Citadel

Future Work

- Use to rank other items such as books, movies, graduate school student applicants, etc.
- Enter a bracket to ESPN for March Madness.

Work Cited

Michalewicz, Zbigniew, and David B. Fogel. How to Solve It. New York: Springer, 2004.

Goodson, Neil, and Colin Stephenson. "March "Mathness": Sports Ranking Using Linear Algebra." Math 481 Final Presentation. College of Charleston, Charleston, SC. 17 Apr. 2008.

Acknowledgements

- Emmie Douglas for her input and scholarly advice.
- Dr. Amy Langville for her original Evolutionary Optimization code, datasets, and continued guidance throughout the research process.

Evolutionary Optimization

Pros:

1. Simple structure, easy to code.
2. Can handle large datasets.
3. With enough randomization and time, will reach global solution.
4. Early termination gives meaningful answer.

Cons:

1. Can get stuck in local solution for a while.
2. Can take a long time to converge.
3. Due to randomization, different runs produce different solutions.

Solving the Optimization Problem

1. EO (evolutionary optimization)
2. BILP (binary integer linear program)
3. LP (linear program)

BILP

Finds global minimum, solves MVR problem optimally.

BILP

Finds global minimum, solves MVR problem optimally.

Feb. 2009 trip to University of Tsukuba

BILP

Finds global minimum, solves MVR problem optimally.

YOSHI'S RANK AGGREGATION BILP

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

$$\begin{aligned}
&\text{maximize} && \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ij} && \\
&\text{subject to} && x_{ij} + x_{ji} = 1 && \forall \text{ distinct pairs } (i,j) \in N \times N \\
& && x_{ij} + x_{jk} + x_{ki} \leq 2 && \forall \text{ distinct triples } (i,j,k) \in N \times N \times N \\
& && x_{ij} \in \{0,1\} && \forall (i,j) \in N \times N \\
& && x_{ii} = 0 && \forall i \in N
\end{aligned}$$

maximizes agreement among *k* input ranked lists

C_{ij} = # of input lists having *i* above *j*

BILP

Related to the **linear ordering** (LOR) problem and **minimum feedback arc** problem; Reinelt et al.

Finds global minimum, solves MVR problem optimally.

YOSHI'S RANK AGGREGATION BILP

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

$$\begin{aligned}
&\text{maximize} && \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ij} && \\
&\text{subject to} && x_{ij} + x_{ji} = 1 && \forall \text{ distinct pairs } (i,j) \in N \times N \\
& && x_{ij} + x_{jk} + x_{ki} \le 2 && \forall \text{ distinct triples } (i,j,k) \in N \times N \times N \\
& && x_{ij} \in \{0,1\} && \forall (i,j) \in N \times N \\
& && x_{ii} = 0 && \forall i \in N
\end{aligned}$$

maximizes agreement among *k* input ranked lists

C_{ij} = # of input lists having *i* above *j*

MVR and Rank Aggregation

Every team ranks its opponents with its point differentials, from both an offensive and a defensive viewpoint.

max agreement = *min* disagreement = *min* # violations

Definition of C matrix

$$c_{ij} := \left|\left\{k \in N \middle| d_{kj} < d_{ki}\right\}\right| + \left|\left\{k \in N \middle| d_{ik} < d_{jk}\right\}\right|$$

$$w_{ij} := \sum_{k \in N | d_{kj} < d_{ki}} \left(d_{kj} - d_{ki}\right) + \sum_{k \in N | d_{ik} < d_{jk}} \left(d_{ik} - d_{jk}\right)$$

BILP: calculating C

Every team ranks its opponents with its point differentials, from both an offensive and a defensive viewpoint.

Point Differential Matrix

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 45 & 0 & 18 & 8 & 20 \\ 3 & 0 & 0 & 2 & 0 \\ 31 & 0 & 0 & 0 & 0 \\ 45 & 0 & 27 & 38 & 0 \end{bmatrix} \longrightarrow$$

C Matrix

$$\begin{bmatrix} 0 & \boxed{8} & 6 & 5 & 7 \\ 0 & 0 & 0 & 0 & 2 \\ 0 & 6 & 0 & 2 & 4 \\ 0 & 7 & 3 & 0 & 5 \\ 0 & 2 & 1 & 1 & 0 \end{bmatrix}$$

C_{12} = (# of entries in col. 2 less than col. 1)

\+ (# of entries in row 1 less than row 2)

$= \boxed{4} + \boxed{4} = \boxed{8}$

BILP: multiple optimal solutions

Criteria for finding some m.o.s.:

1. $C_{ij} = C_{ji}$
2. Teams *i* and *j* are neighbors in ranked list.

Test:

Systematically descend down the ranked list searching for sets of teams satisfying criteria.

BILP: multiple optimal solutions

2009 SoCon Example

$$\begin{bmatrix}5\\4\\3\\9\\2\\12\\1\\11\\6\\10\\7\\8\end{bmatrix}\qquad\begin{bmatrix}5\\\boxed{\begin{matrix}3\\4\end{matrix}}\\9\\2\\12\\1\\11\\6\\10\\7\\8\end{bmatrix}\qquad\begin{bmatrix}5\\4\\3\\9\\2\\12\\\boxed{\begin{matrix}11\\1\end{matrix}}\\6\\10\\7\\8\end{bmatrix}\qquad\begin{bmatrix}5\\\boxed{\begin{matrix}3\\4\end{matrix}}\\9\\2\\12\\\boxed{\begin{matrix}11\\1\end{matrix}}\\6\\10\\7\\8\end{bmatrix}$$

2 two-way ties

BILP

Pros:

1. Optimal solution!
2. Can find some multiple optimal solutions.
3. Fast, with good initial feasible solution and bounds.

Cons:

1. $O(N^3)$ transitivity constraints limit runtime and problem size.

BILP

Pros:

1. Optimal solution!
2. Can find some multiple optimal solutions.
3. Fast, with good initial feasible solution and bounds.

Cons:

1. $O(N^3)$ transitivity constraints limit runtime and problem size.

Constraint relaxation is huge help here.

BILP

$$
\begin{array}{lll}
\text{maximize} & \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ij} & \\
\text{subject to} & x_{ij} + x_{ji} = 1 & \forall \text{ distinct pairs } (i,j) \in N \times N \\
& x_{ij} + x_{jk} + x_{ki} \le 2 & \forall \text{ distinct triples } (i,j,k) \in N \times N \times N \\
& x_{ij} \in \{0,1\} & \forall (i,j) \in N \times N \\
& x_{ii} = 0 & \forall i \in N
\end{array}
$$

Cons:

1. $O(N^3)$ transitivity constraints limit runtime and problem size.

Constraint relaxation is huge help here.

Solving the Optimization Problem

1. EO (evolutionary optimization)
2. BILP (binary integer linear program)
3. LP (linear program)

LP

Rank Aggregation/MVR LP

minimize	$\sum_{i \in N} \sum_{j \in N} c_{ij} x_{ij}$	$\left(c_{ij}\text{ can be replaced with } w_{ij}\right)$
subject to	$x_{ij} + x_{ji} = 1$	$\forall$ distinct pairs $(i,j) \in N \times N$
	$x_{ij} + x_{jk} + x_{ki} \le 2$	$\forall$ distinct triples $(i,j,k) \in N \times N \times N$
	$\boxed{0 \le x_{ij} \le 1}$	$\forall (i,j) \in N \times N$
	$x_{ii} = 0$	$\forall i \in N$

LP: multiple optimal solutions

2009 SoCon Example ***X*** matrix

$$
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & \boxed{.5130} & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\
1 & 1 & 0 & \boxed{.4839} & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & \boxed{.5161} & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & & \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
\boxed{.4870} & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0
\end{bmatrix}
$$

Fractional Locations:
(3,4) and (4,3)

(1,11) and (11,1)

2 two-way ties

LP: multiple optimal solutions

2009 SoCon Example

$$\begin{bmatrix} 5 \\ 4 \\ 3 \\ 9 \\ 2 \\ 12 \\ 1 \\ 11 \\ 6 \\ 10 \\ 7 \\ 8 \end{bmatrix} \quad \begin{bmatrix} 5 \\ 3 \\ 4 \\ 9 \\ 2 \\ 12 \\ 1 \\ 11 \\ 6 \\ 10 \\ 7 \\ 8 \end{bmatrix} \quad \begin{bmatrix} 5 \\ 4 \\ 3 \\ 9 \\ 2 \\ 12 \\ 11 \\ 1 \\ 6 \\ 10 \\ 7 \\ 8 \end{bmatrix} \quad \begin{bmatrix} 5 \\ 3 \\ 4 \\ 9 \\ 2 \\ 12 \\ 11 \\ 1 \\ 6 \\ 10 \\ 7 \\ 8 \end{bmatrix}$$

2 two-way ties

LP: multiple optimal solutions

Theorem: If fractional pairs in LP solution are *isolated*, then

1. Locations of fractional values give information on multiple optimal solutions.
2. Objective values for BILP and LP are equal.
3. If LP solution is binary, MVR solution is unique.

LP: sensitivity analysis

Range on C_{ij} values

	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]

Tight bounds:

1 and 9

2 and 9

9 and 12

LP: sensitivity analysis

Range on C_{ij} values

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

$$\begin{bmatrix} 5 \\ 4 \\ 3 \\ 9 \\ 2 \\ 12 \\ 1 \\ 11 \\ 6 \\ 10 \\ 7 \\ 8 \end{bmatrix}$$

Tight bounds:

1 and 9

2 and 9

9 and 12

LP

Pros:

1. Nearly always finds optimal solution. Otherwise, finds excellent near-optimal soln.
2. Easy identification of multiple optimal solns.
3. Fast, with good initial feasible solution and bounds.
4. Sensitivity analysis gives confidence measures.

Cons:

1. $O(N^3)$ transitivity constraints limit runtime and problem size.

Applications

Sports

1. March Madness
2. Go
3. Sumo wrestling

Recommendation systems

1. Netflix
2. Amazon

Webpages

1. Meta-search
2. Google, Yahoo, Ask

Applications: March Madness

Table 3: Computational Results for Iterative LP method *with bounding* on 347 NCAA teams

iteration	LP time	Obj. value	best rank	ConGen time	# con.added
1	1.19	-1778224.00	-1777474.00	0.28	11885
2	1.58	-1777829.00	-1777516.00	0.27	6900
3	2.24	-1777801.50	-1777712.00	0.16	644
4	2.13	-1777800.50	-1777779.00	0.13	179
5	2.22	-1777800.50	-1777787.00	0.13	38
6	2.17	-1777800.50	-1777787.00	0.14	35
7	2.33	-1777800.50	-1777793.00	0.13	29
8	2.09	-1777800.50	-1777793.00	0.13	54
9	2.22	-1777800.50	-1777793.00	0.14	30
10	2.19	-1777800.50	-1777793.00	0.13	14
11	2.14	-1777800.50	-1777793.00	0.14	19
12	2.27	-1777800.50	-1777793.00	0.14	28
13	2.30	-1777800.50	-1777793.00	0.13	20
14	2.16	-1777800.50	-1777793.00	0.13	4
15	2.39	-1777800.50	-1777793.00	0.13	20
16	2.16	-1777800.50	-1777793.00	0.14	7
17	2.30	-1777800.50	-1777793.00	0.14	25
18	2.36	-1777800.50	-1777793.00	0.14	10
19	2.13	-1777800.50	-1777793.00	0.13	19
20	2.34	-1777800.50	-1777793.00	0.14	12
21	2.22	-1777800.50	-1777793.00	0.13	26
22	2.34	-1777800.50	-1777793.00	0.13	10
23	2.34	-1777800.50	-1777793.00	0.13	0
total	49.78			3.33	20008

Applications: March Madness

Table 3: Computational Results for Iterative LP method *with bounding* on 347 NCAA teams

iteration	LP time	Obj. value	best rank	ConGen time	# con.added
1	1.19	-1778224.00	-1777474.00	0.28	11885
2	1.58	-1777829.00	-1777516.00	0.27	6900
3	2.24	-1777801.50	-1777712.00	0.16	644
4	2.13	-1777800.50	-1777779.00	0.13	179
5	2.22	-1777800.50	-1777787.00	0.13	38
6	2.17	-1777800.50	-1777787.00	0.14	[illegible]
7	2.33	-1777800.50	-1777793.00	0.13	[illegible]
8	2.09	-1777800.50	-1777793.00	0.13	[illegible]
9	2.22	-1777800.50	-1777793.00	0.14	30
10	2.19	-1777800.50	-1777793.00	0.13	14
11	2.14	-1777800.50	-1777793.00	0.14	19
12	2.27	-1777800.50	-1777793.00	0.14	28
13	2.30	-1777800.50	-1777793.00	0.13	20
14	2.16	-1777800.50	-1777793.00	0.13	4
15	2.39	-1777800.50	-1777793.00	0.13	20
16	2.16	-1777800.50	-1777793.00	0.14	7
17	2.30	-1777800.50	-1777793.00	0.14	25
18	2.36	-1777800.50	-1777793.00	0.14	10
19	2.13	-1777800.50	-1777793.00	0.13	19
20	2.34	-1777800.50	-1777793.00	0.14	12
21	2.22	-1777800.50	-1777793.00	0.13	26
22	2.34	-1777800.50	-1777793.00	0.13	10
23	2.34	-1777800.50	-1777793.00	0.13	0
total	49.78			3.33	20008

< .05% of 4.1 million transitivity constraints

Applications: March Madness

Algorithm	ESPN Score
Colley Bi-weekly step	1420
BILP C- Half Count	1360
BILP W- Half Count	1360
BILP C- Count	1350
BILP C- Weight	1350
President Obama	1230

Future Work

Much more sensitivity analysis

Duality from LP

Conclusions

Ranking and RankAgg with ties and sensitivity measures

Gains made by considering 3 different optimization viewpoints

LOR vs. TSP