Research Interests

Optimization: looking for the best solution from among a number of candidates.

Prototypical optimization problem:

$$\begin{array}{lrcl} \min & f(x) & & \\ \text{subject to} & g(x) & \leq & 0 \\ & x & \in & X \subseteq \mathbb{R}^n \end{array}$$

Here, $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}^m$. Functions can be linear or nonlinear.

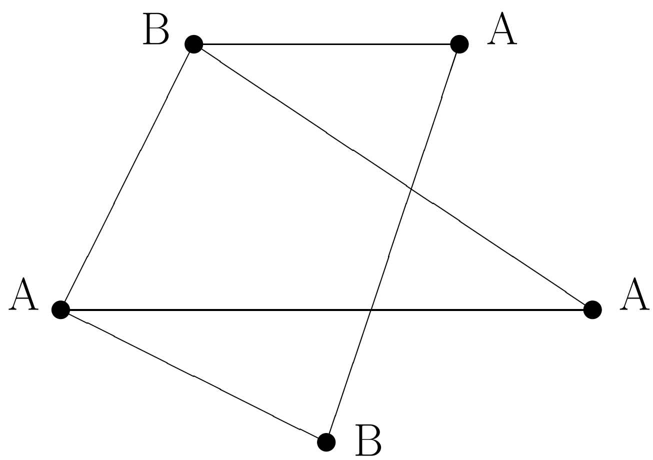
Possible choices for X:

- Nonnegativity, $x \geq 0$.
- Binary, $x \in \{0, 1\}^n$. Arises in combinatorial optimization.
- Semidefinite programming: if x forms a matrix, we may require that this matrix be positive semidefinite.

Applications:

- Find the maximum cut in a graph. One application: finding the ground state of an Ising spin glass.

  Divide vertices into two sets to cut as many edges as possible

- Eigenvalue optimization: choose a matrix satisfying certain conditions that has the smallest maximum eigenvalue. Applications in structural design, control theory, combinatorial optimization, ...
- Linear ordering: place objects in order when there is a cost associated with placing one object before another.
- Portfolio optimization.
- Clustering. Eg: realignment in the NFL.
- Scheduling.

Solution methods

Can often find a good *feasible* solution.

How close is this to *optimal?*

Look at **relaxations** of the original problem to get bounds.

I'm particularly interested in **linear programming relaxations**:

$$\begin{array}{lrcl} \min & c^T x & & \\ \text{subject to} & Ax & = & b \\ & x & \geq & 0 \end{array}$$

How can these relaxations be *tightened?*

How good can we make them?

Typically, solve a sequence of better and better linear programming relaxations. How do we solve this sequence *quickly?*

CUTTING PLANES

Prototypical integer programming problem:

$$\begin{array}{lrcl} \text{min} & c^T x & & \\ \text{subject to} & Ax & = & b \\ & x & \geq & 0 \quad \textbf{and integral} \end{array}$$

LP relaxation:

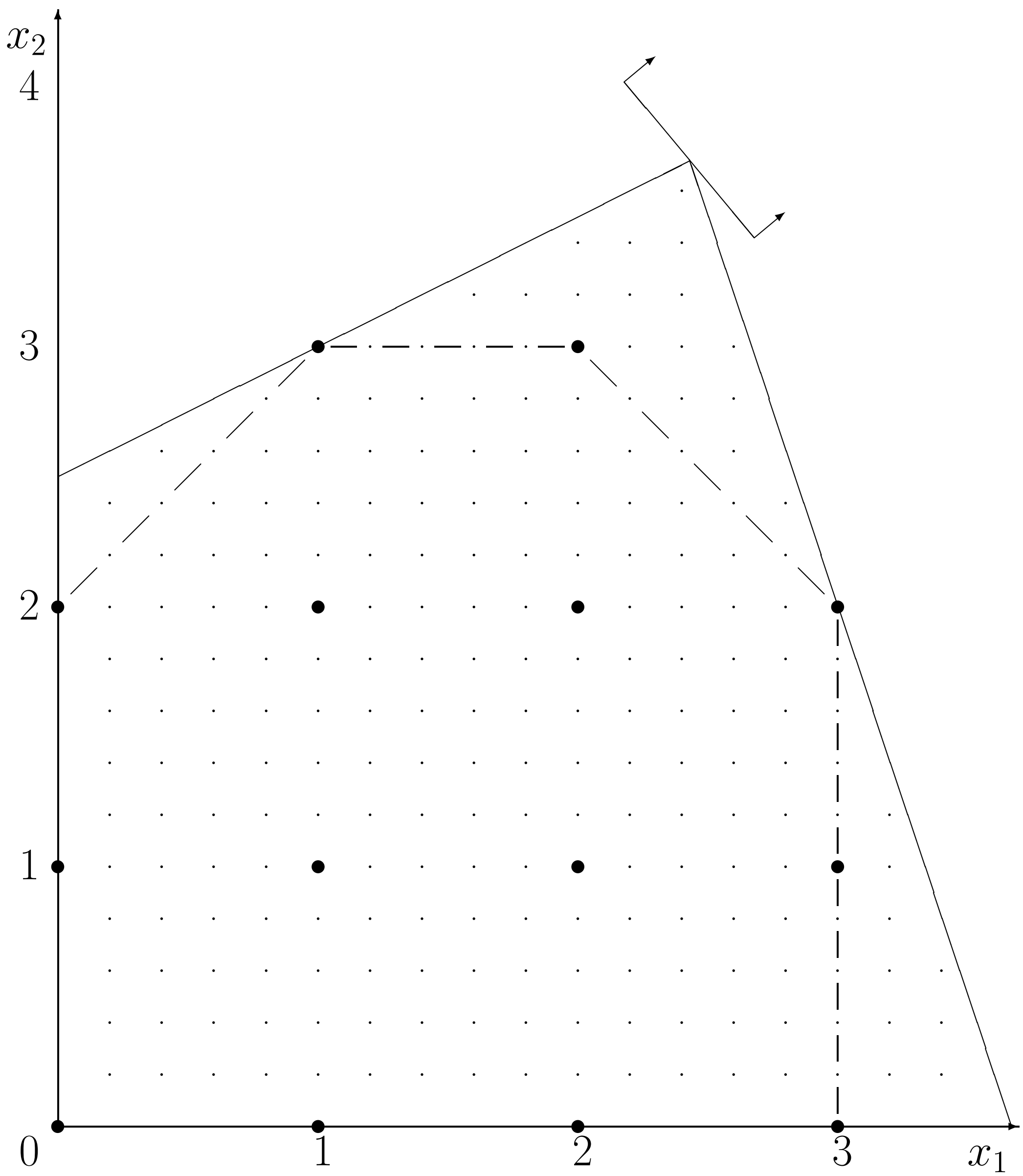
$$\begin{array}{lrcl} \text{min} & c^T x & & \\ \text{subject to} & Ax & = & b \\ & x & \geq & 0 \end{array}$$

In general, it is **far easier to solve a linear program** than an integer program of comparable size.

Improve the LP relaxation by adding violated constraints:

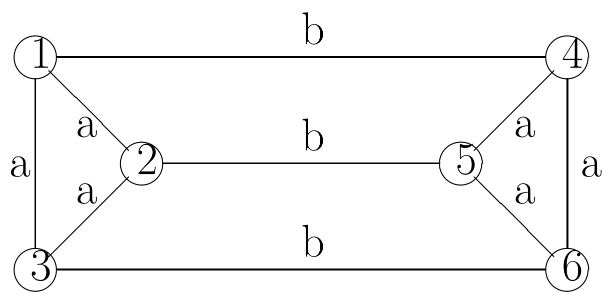
$$\begin{array}{lrcl} \text{min} & c^T x & & \\ \text{subject to} & Ax & = & b \\ & \mathbf{d^T x} & \leq & \mathbf{g} \\ & x & \geq & 0 \end{array}$$

A 2-D integer programming problem



$$\begin{array}{lrcrcl} \min & z := -6x_1 & - & 5x_2 & & \\ \text{subject to} & 3x_1 & + & x_2 & \leq & 11 \\ & -x_1 & + & 2x_2 & \leq & 5 \\ & & & x_1, x_2 & \geq & 0, \ \text{integer.} \end{array}$$

Traveling Salesman Problem example



Relaxation of the TSP:

$$\begin{array}{lrcll} \min & \multicolumn{3}{l}{\Sigma\, c_e x_e} & \\ \text{s.t.} & \Sigma_{e\in\delta(v)}\, x_e & = & 2 & \text{for all vertices } v \qquad (TSP1) \\ & 0 \ \leq \ x_e & \leq & 1 & \text{for all edges } e \end{array}$$

The point $x_{12} = x_{23} = x_{13} = x_{45} = x_{46} = x_{56} = 1$, $x_{ij} = 0$ for all other edges, solves $(TSP1)$.

Any tour must use two of the edges between the set of vertices $\{1, 2, 3\}$ and the set of vertices $\{4, 5, 6\}$.

Add the **subtour elimination constraint**:

$$\sum_{i=1}^{3} \sum_{j=4}^{6} x_{ij} \geq 2$$

INTERIOR POINT METHODS

Simplex is the classical method for solving linear programming problems.

It finds an optimal extreme point.

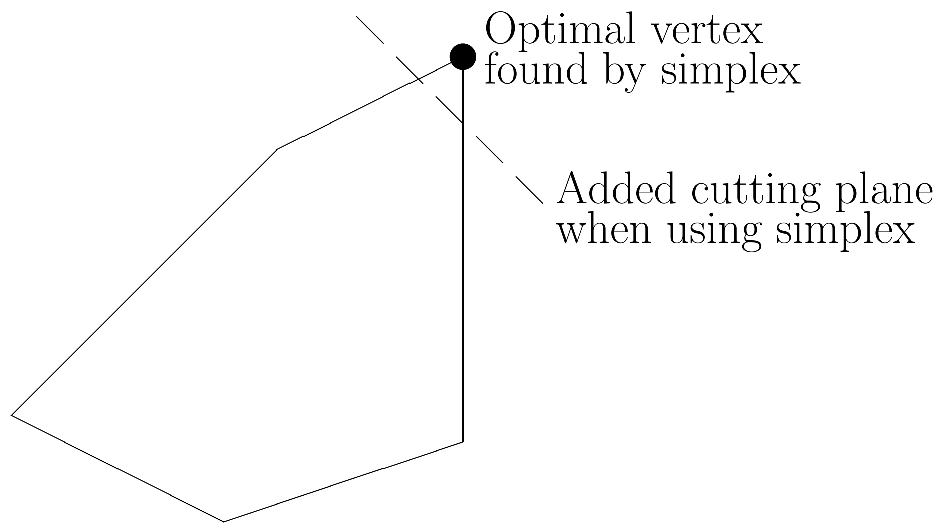
Alternative: use an interior point method

Look for cutting planes prior to optimality

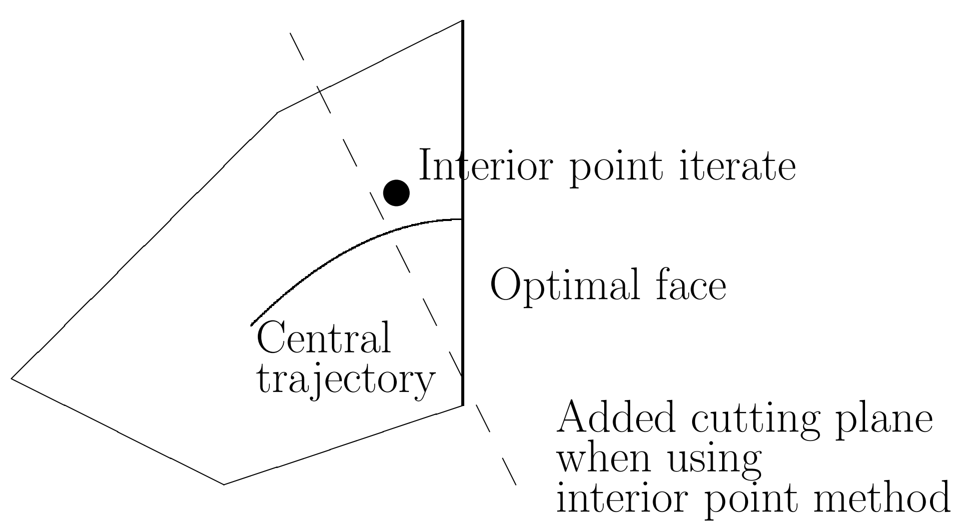
Find **deeper cuts**, so need to look at fewer relaxations

Comparing the strength of simplex and interior point cutting planes

Simplex:

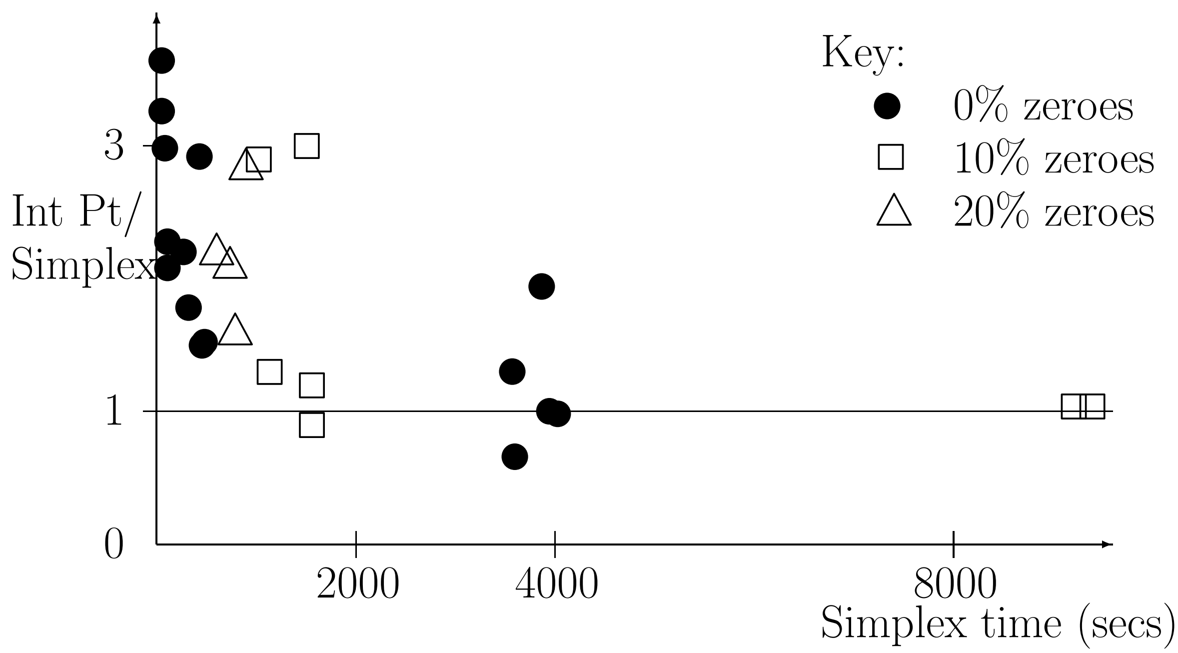


Interior point method:



Large linear ordering problems (up to 250 sectors)

Industrial strength simplex vs homegrown interior point



Require

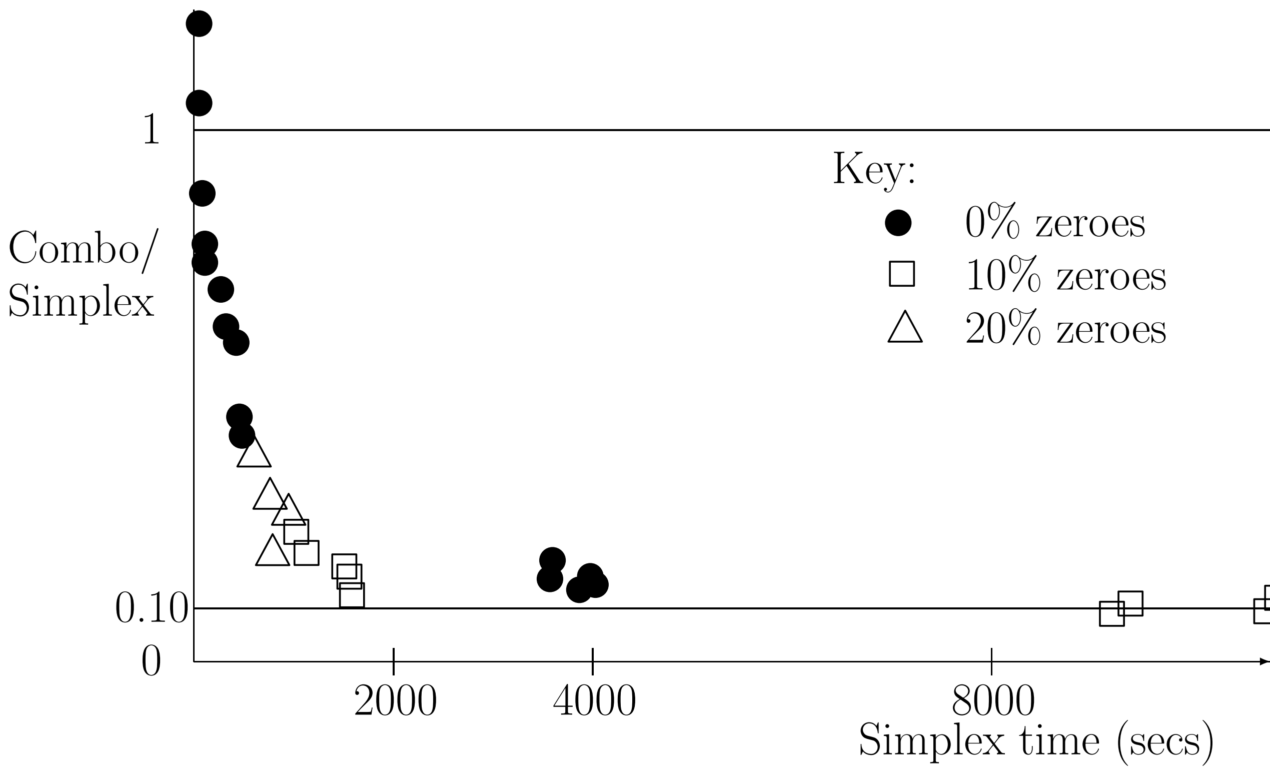
$$x_{ij} = \begin{cases} 1 & \text{if } i \text{ before } j \\ 0 & \text{otherwise} \end{cases}$$

Enforce using triangle inequalities:

$$x_{ij} + x_{jk} + x_{ki} \leq 2$$

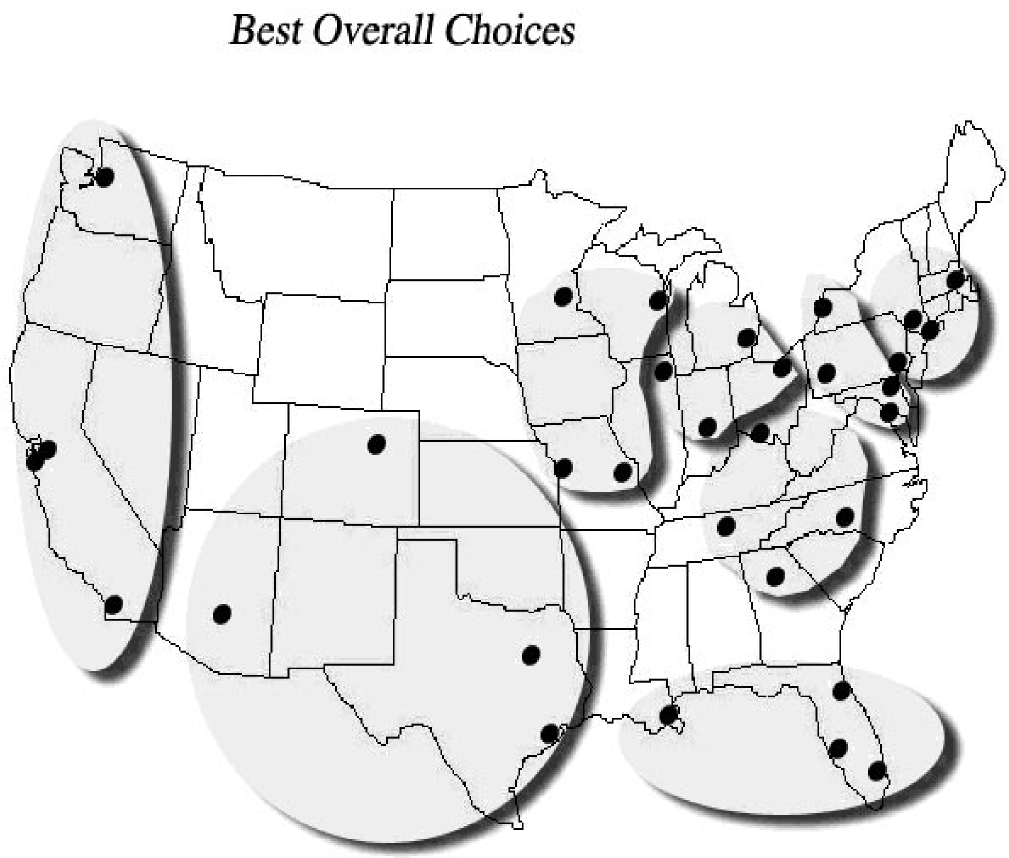
Large linear ordering problems
(up to 250 sectors)

Combining simplex and interior point

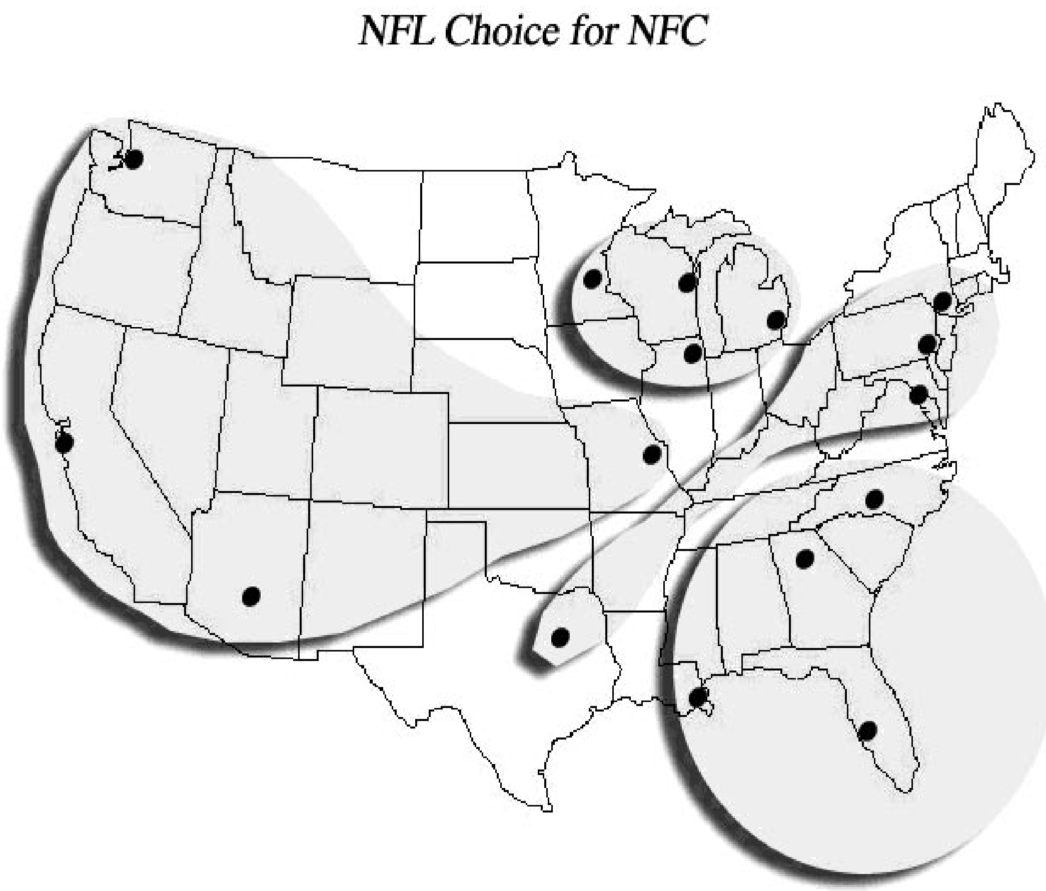


CLUSTERING PROBLEMS

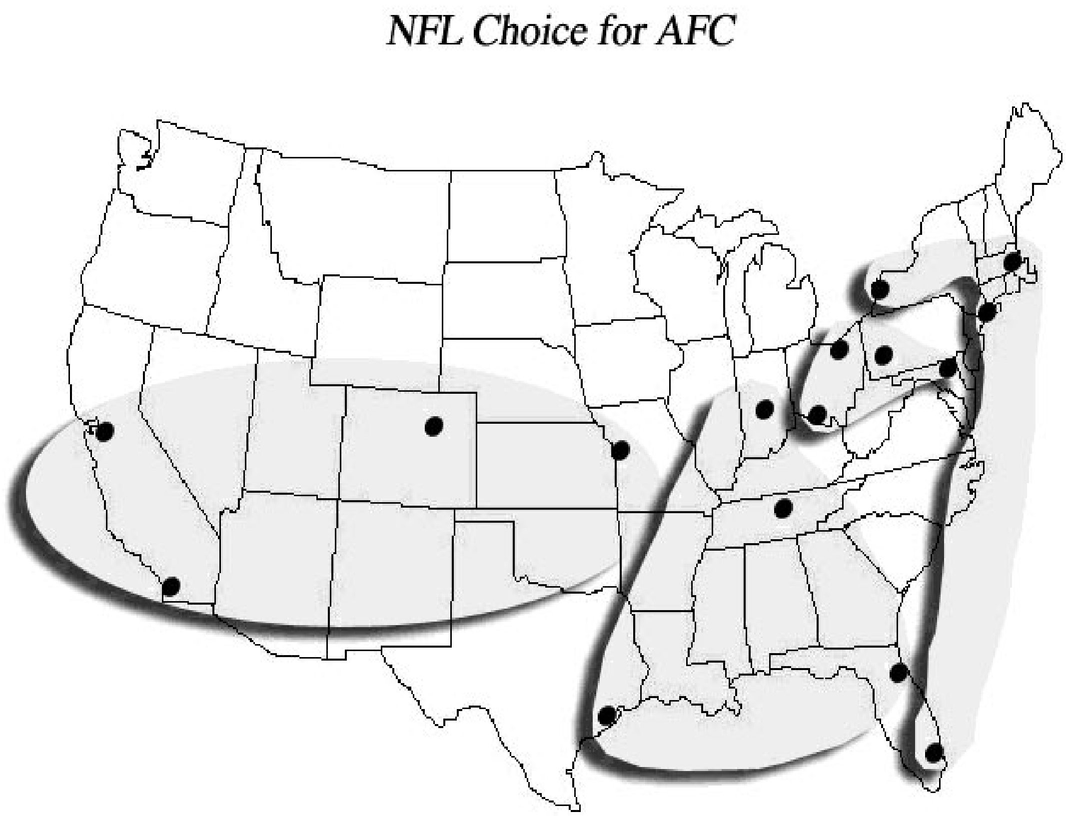
Realignment in the NFL



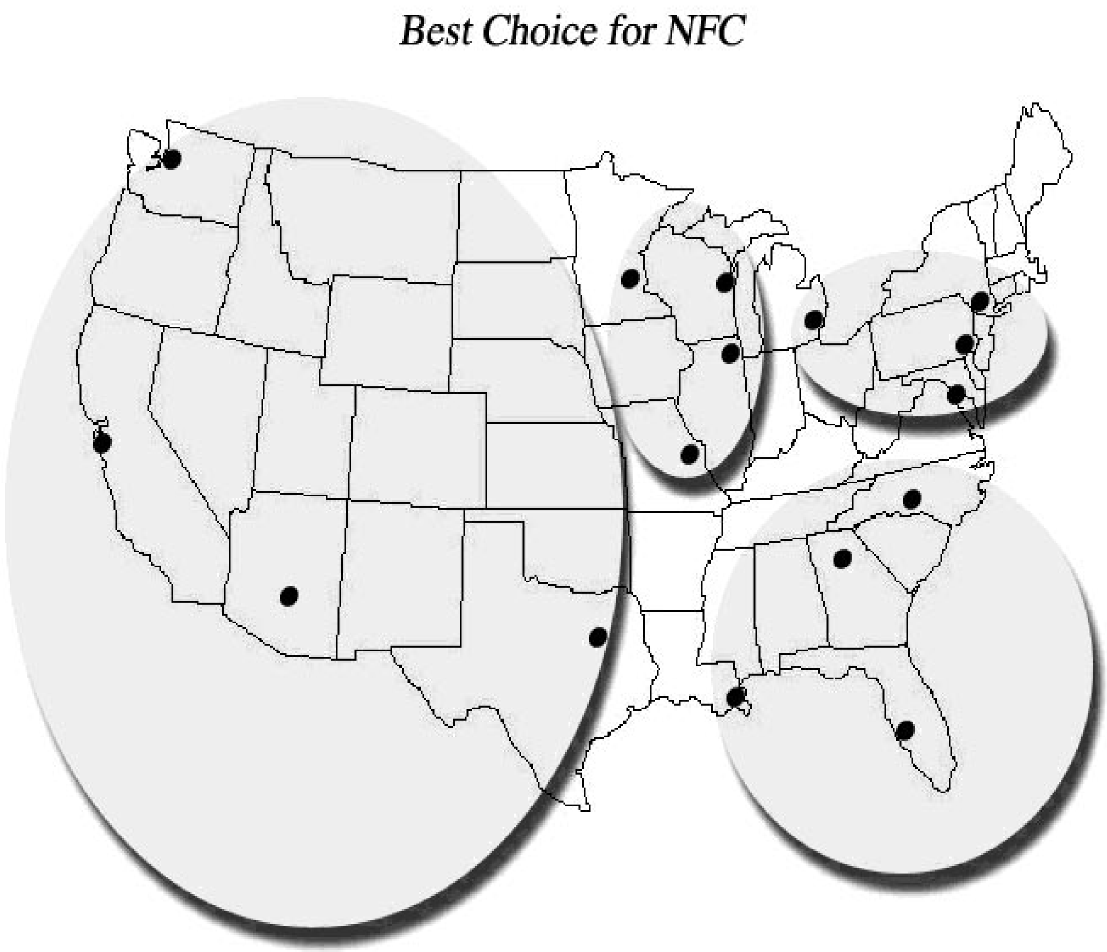
The realignment that minimizes the sum of intradivisional travel distances.



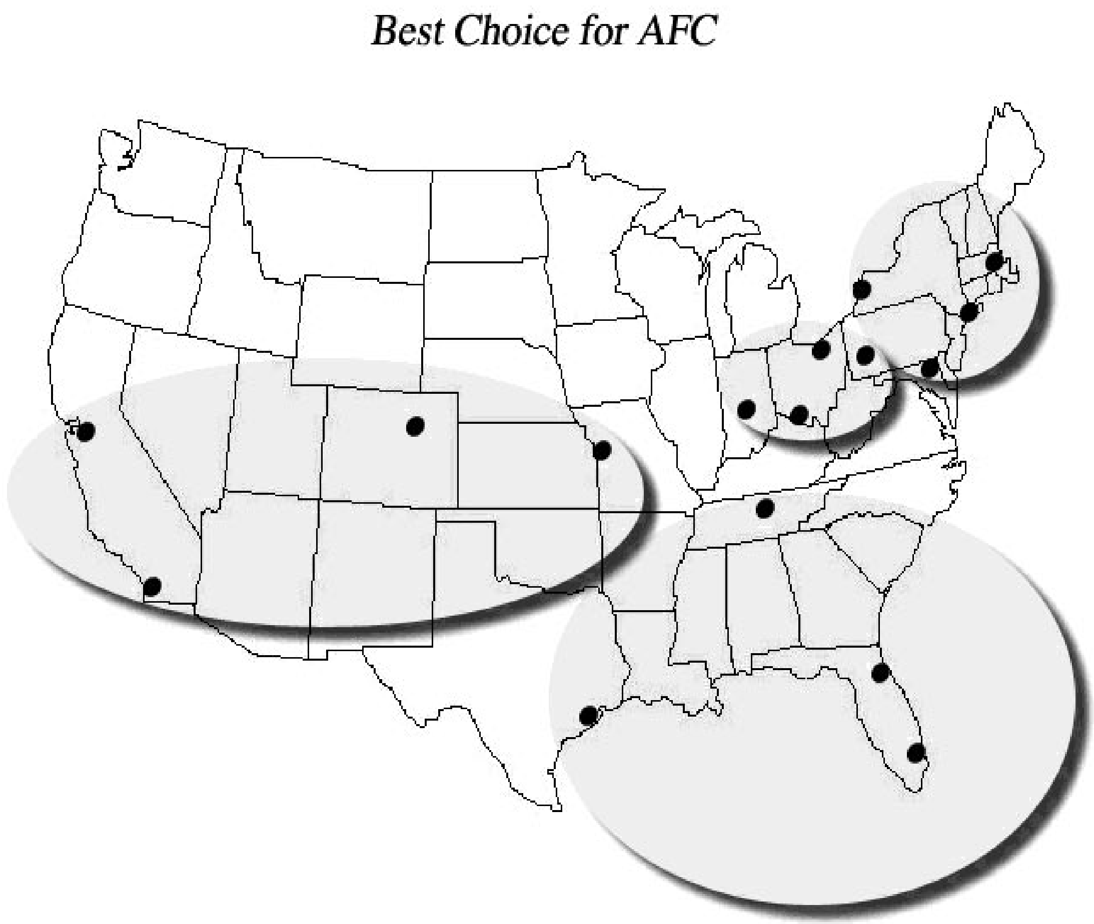
The realignment chosen by the NFL for the NFC.



The realignment chosen by the NFL for the AFC.



The optimal realignment for the NFC.



The optimal realignment for the AFC.

Clustering problems

The realignment problem is a **clustering** problem.

Require each cluster to contain **exactly** four vertices.
Can find families of cutting planes for this problem.

In some settings (eg microaggregation), want instead each cluster to be **no smaller than a given size**.

Xiaoyun Ji (Sharron) has been working on this problem with me.

She has found some new families of constraints, and she has implemented her results.

Positioning of rotamers in **computational biology** can be expressed as a variant of a clustering problem.

THEORETICAL ISSUES

If you can find a violated cutting plane in polynomial time, can you solve the optimization problem in polynomial time?

Yes, if you use the ellipsoid algorithm.

But the ellipsoid algorithm is slow in practice.

Interior point methods: only known method requires that unimportant constraints be dropped in order to guarantee that the algorithm keeps making progress.

Srini Ramaswamy and I refined this approach to integrate the optimization aspect more efficiently.

Luc Basescu and I have looked at the convergence of extensions of these algorithms.

Open question: Is there an interior point column generation algorithm that converges in polynomial time and does not require that unimportant constraints be dropped?

SEMIDEFINITE PROGRAMMING

$$\begin{array}{lrcl} \min & C \bullet X & & \\ \text{s.t.} & A_i \bullet X & = & b_i \quad i = 1, \ldots, m \\ & X & \succeq & 0 \end{array}$$

X, C, A_i are symmetric square matrices.

X is constrained to be positive semidefinite (psd).

The symbol $\bullet$ denotes the **Frobenius inner product**:

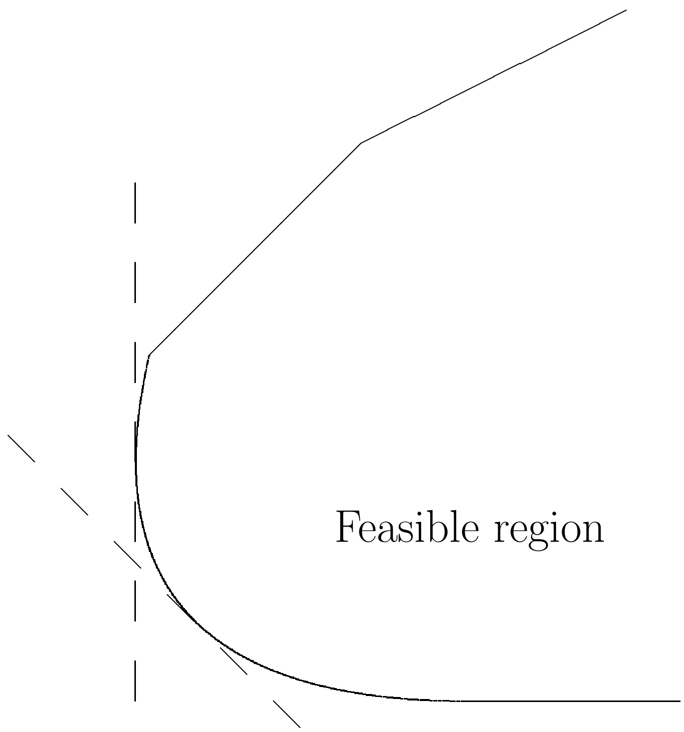
$$\begin{array}{rcl} C \bullet X & := & \Sigma_{i=1}^n \Sigma_{j=1}^n C_{ij} X_{ij} \\ & = & \text{trace}(CX) \quad \text{for symmetric } C, X \end{array}$$

Can get **tighter relaxations** of some combinatorial optimization problems by using semidefinite programming.

Typically, X is an outer product $X = xx^T$ for some vector x. **Relax** the requirement that X have **rank one**, only require X to be symmetric and positive semidefinite.

Also has applications in **control theory** and elsewhere.

Kartik Krishnan and I investigated replacing the semidefiniteness constraint with linear constraints.



Variational characterization: a matrix X is psd if and only if $d^T X d \geq 0$ for all vectors d.

Find appropriate vectors d to use as **cutting planes**.

Duality in SDP

The dual problem is

$$\begin{array}{lrcl} \max & b^T y & & \\ \text{s.t.} & \Sigma_{i=1}^n y_i A_i + S & = & C \\ & S & \succeq & 0 \end{array}$$

The optimal X and S can be simultaneously diagonalized so that

$$X = \begin{bmatrix} P & Q \end{bmatrix} \begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} P^T \\ Q^T \end{bmatrix} = P\Lambda P^T$$

and

$$S = \begin{bmatrix} P & Q \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & \Gamma \end{bmatrix} \begin{bmatrix} P^T \\ Q^T \end{bmatrix} = Q\Gamma Q^T$$

Recently, Kartik and I have looked at trying to exploit this duality relationship in order to improve our algorithm.

QUADRATIC CONSTRAINTS

Semidefinite relaxations:

Steve Braun and I looked at relaxing **complementarity requirements**:

- Require $x_i x_j = 0$ for a pair of variables.
- Change variables to $X = xx^T$. Relax to require X be psd and symmetric.
- Complementarity constraint is **linear** in the new variables: namely, $X_{ij} = 0$.

This idea needs investigation for extension to more general mathematical programs with equilibrium constraints.

Second order cone programming (SOCP)

Constraints of the form

$$\sum_{i=1}^{n} x_i^2 \leq t^2$$

where x and t are variables.

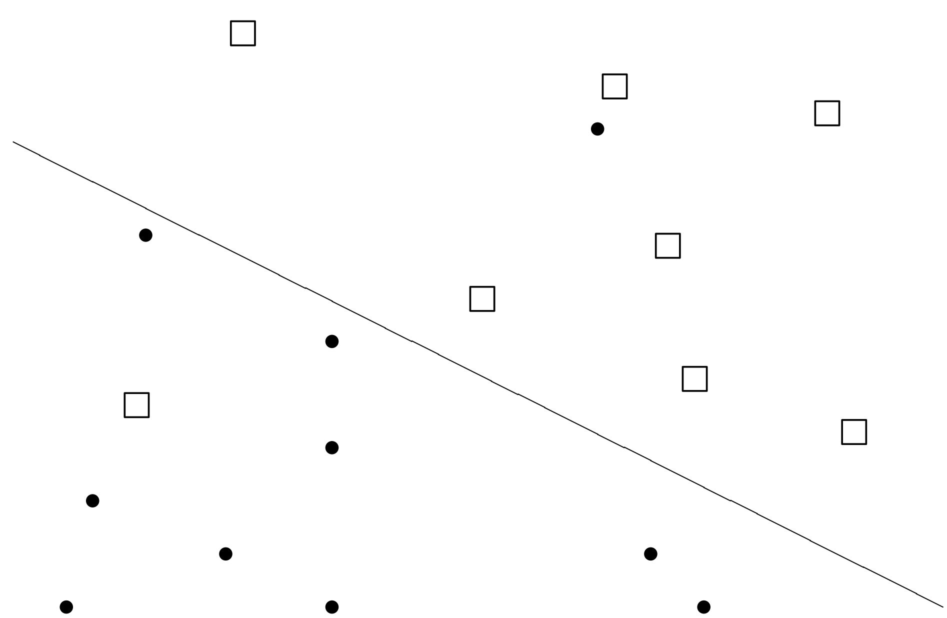
Arise when have norm constraints, for example.

Luc Basescu has proved some nice theoretical results for column generation methods with generalized versions of these constraints.

He is starting work on an implementation.

An SOCP column generation example in data mining:

- Have thousands of points $\{x^i, i = 1, \ldots, m\}$ in $I\!R^n$ which belong to one of two sets.
- Want to find a plane $w^T x = b$ to separate the points, if possible.
- If the points cannot be separated, want to choose the "best" plane.
- Measure the error for the ith point as the euclidean distance from the plane to x^i: this gives an SOCP constraint.
- Only generate these constraints as needed.



COURSES

Core:

- MATP 6600: Nonlinear programming
- MATP 6620: Combinatorial optimization and integer programming
- MATP 6640: Linear programming

Also useful:

- MATH 6220: Intro to functional analysis
- MATH 6800: Computational linear algebra
- various DSES, CIVL, ECSE, CS courses

Other courses are useful depending on the research topic. For example, topics in control theory rely on a good knowledge of differential equations.

SUMMARY

Solve hard optimization problems by looking at a **relaxation** of the problem and repeatedly improving the relaxation.

Possible relaxations: LP relaxation, semidefinite programming relaxation, second-order cone program,...

Can often find a good **feasible** solution.

How close is this to **optimal?**

How can these relaxations be **tightened?**

How good can we make them?

For example, solve a sequence of better and better linear programming relaxations. How do we solve this sequence **quickly?**

References

[1] S. Braun and J. E. Mitchell. A semidefinite programming heuristic for quadratic programming problems with complementarity constraints. Technical report, Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180, November 2002.

[2] K. Krishnan and J. E. Mitchell. Semi-infinite linear programming approaches to semidefinite programming (SDP) problems. Technical report, Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180, August 2001. Accepted for publication in the Fields Institute Communications Series, Volume 37, "Novel approaches to hard discrete optimization problems", edited by P. Pardalos and H. Wolkowicz, pages 121–140, 2003.

[3] K. Krishnan and J. E. Mitchell. Cutting plane methods for semidefinite programming. Technical report, Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180, November 2002.

[4] J. E. Mitchell. Computational experience with an interior point cutting plane algorithm. *SIAM Journal on Optimization*, 10(4):1212–1227, 2000.

[5] J. E. Mitchell. Realignment in the NFL. Technical report, Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180, November 2000. Accepted for publication in *Naval Research Logistics*.

[6] J. E. Mitchell. Branch-and-cut algorithms for integer programming. In C. A. Floudas and P. M. Pardalos, editors, *Encyclopedia of Optimization*. Kluwer Academic Publishers, Dordrecht, The Netherlands, August 2001.

[7] J. E. Mitchell. Branch-and-cut for the k-way equipartition problem. Technical report, Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180, January 2001.

[8] J. E. Mitchell. Cutting plane algorithms for integer programming. In C. A. Floudas and P. M. Pardalos, editors, *Encyclopedia of Optimization*. Kluwer Academic Publishers, Dordrecht, The Netherlands, August 2001.

[9] J. E. Mitchell. Restarting after branching in the SDP approach to MAX-CUT and similar combinatorial optimization problems. *Journal of Combinatorial Optimization*, 5(2):151–166, 2001.

[10] J. E. Mitchell. Branch-and-cut algorithms for combinatorial optimization problems. In P. M. Pardalos and M. G. C. Resende, editors, *Handbook of Applied Optimization*, pages 65–77. Oxford University Press, January 2002.

[11] J. E. Mitchell and B. Borchers. Solving real-world linear ordering problems using a primal-dual interior point cutting plane method. *Annals of Operations Research*, 62:253–276, 1996.

[12] J. E. Mitchell and B. Borchers. Solving linear ordering problems with a combined interior point/simplex cutting plane algorithm. In H. L. Frenk *et al.*, editor, *High Performance Optimization*, chapter 14, pages 349–366. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2000.

[13] J. E. Mitchell and S. Braun. Rebalancing an investment portfolio in the presence of transaction costs. Technical report, Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180, November 2002.

[14] J. E. Mitchell, P. M. Pardalos, and M. G. C. Resende. Interior point methods for combinatorial optimization. In D.-Z. Du and P. M. Pardalos, editors, *Handbook of Combinatorial Optimization*, volume 1, pages 189–297. Kluwer Academic Publishers, 1998.

[15] J. E. Mitchell and S. Ramaswamy. A long-step, cutting plane algorithm for linear and convex programming. *Annals of Operations Research*, 99:95–122, 2000.