Agenda
1) Review
   - generic hierarchy scheme
     - Sharai: Adams (SA)
     - Lasserre
     - Louiz Schrijver

2) Applications to the knapsack problem
3) Questions / Misc.

Review

Big picture idea of hierarchy:
- working with optimization problems (node-cut, axis-vertex cover, etc.), often NP-hard
- true combinatorial optimum: OPT
- relax the problem to an LP/SDP and put: FPRAC
- rounding procedure to get: ROAED

For maximization problems: ROAED \leq OPT \leq FPRAC
For minimization problems: FPRAC \leq OPT \leq ROAED

Big idea: add additional constraints so as to get better and better approximations
often derived from the non-linear constraints of the original problem
- e.g. to force \( y_i \) be 0/1, want to write \( y_i^2 = y_i \), but that's not linear.
- write other constraints to approximate this requirement

- LP/SDP hierarchy: define "levels / rounds" of a problem, each level higher and more powerful than the previous

Notes: For \( n \) variables \( \{0,1\} \), the \( n^{th} \) level has int. gap = 1 (i.e. exact solution)

- Lasserre Schrijver
  - define an operator \( \mathcal{N} \) on the convex relaxation \( P \) of a 0/1 linear program
  - each application of \( \mathcal{N} \) adds additional variables and constraints
  - final relaxation: \( \mathcal{N}(\mathcal{N}(... \mathcal{N}(\text{original linear program}) ...)) \) is an approx. of the decision variables

- Sharai: Adams
  - fixed size problem, LS whose generation of successive relaxation matrices depends on \( \mathcal{N} \)
  - in new variables were introduced
  - instead of introducing variables immediately, they get added one at a time

- Lasserre: Kimmer
  - provides a hierarchy for SDP relaxation of quadratic 0/1 program
  - introduced a new variable \( y_i^T \) for the product of every \( + \) variables in the original program

- max ind. set is normally have \( y_i \), \( y_j = 0 \) \( V(i,j) \in E \)

- now have \( y_S \) for each \( S \in V \), \( y_i = 1 \) iff \( i \in S \) are in ind. set
Before continuing, worth noting:

1) In any of the added constraints to the base, integral solution still works. Thus, the process is indeed a relaxation.

2) While the n-th level of the hierarchy provided the optimal integer solution, we've spent exponential time generating the additional constraints, as we may have just done the brute force $O(2^n)$ procedure.

Expected Linear Formulation for SA-Hierarchy

For any constraint $g_e(x) \geq 0$, in the original problem, we can write:

$$
\sum_{i \in I} \sum_{j \in J} y_{i,j} \Pi_{i \in I} \Pi_{j \in J} (1-x_{i,j}) \geq 0
$$

For any subsets $I, J \subseteq V$ where $V$ is the original set.

In plain English, we just multiply the constraint by some $x_i$ and $1-x_j$ and since each $x_i, x_j \in \{0, 1\}$, $\Pi_{i \in I} \Pi_{j \in J} (1-x_{i,j}) \in \{0, 1\}$, so the above constraint will hold when $g_e(x) \geq 0$.

When we decompose a variable inside $g_e(x)$ in our products, then we'll get $x_i^k$ in our expansion for $k \geq 1$, and $i$ being the index of the variable duplicated.

Since $x_i \in \{0, 1\}$ in the LP, we can rewrite the integrality constraint as $x_i^k = x_i$ to $Y_i, Y_i^k$ and replace the $x_i^k$'s in our expansion accordingly.

As explained, noted, we replace one of the products with "big" variables. Namely, we replace each $\Pi_{i \in I} x_i$ by $Y_i$, after having expanded.

Together, we can formally define the $k$-th level of the SA hierarchy as:

$$
SA^+(k) = \{ y \in P_i(V) | Y_{i,n} = 1 \text{ and } \sum_{i \in I} \sum_{j \in J} y_{i,j} \Pi_{i \in I} \Pi_{j \in J} (1-x_{i,j}) = 0 \}
$$

for any $k$ and $I, J$ s.t. $|I \times J| = 1$.

Where $g_e'$ is obtained by:

1. Meet $g_e(x)$ by $\Pi_{i \in I} \Pi_{j \in J} (1-x_{i,j})$.
2. Expand and replace $x_i^k$ by $Y_i^k$.
3. Replace each $\Pi_{i \in I} x_i$ by $Y_i$.

A point $x \in \{0, 1\}^V$ belongs to $SA^+(k)$ if $\exists y \in SA^+(k)$ s.t. $y_{1,i} = x_i$ for all $i \in V$. 


Knapsack Problem

given a set of n items, each with cost \(c_i\) and weight \(w_i\), and some knapsack capacity \(C\):
find a subset \(S \subseteq \{1, 2, \ldots, n\}\) of cost \(\sum_{i \in S} c_i \leq C\) that maximizes total value \(\sum_{i \in S} v_i\).

ILP: max \(\sum_{i \in S} v_i \) subject to \(\sum_{i \in S} c_i \leq C\)

Simple LP relaxation gives int. gap of 2: \(\text{FDAC}_{\text{OPT}}\)

Greedy solution performs poorly: (e.g., set by \(\frac{c_i}{v_i}\), choose in order)

- performs badly w/ 2-item case
  - \(c_1 = 2\), \(c_2 = 1\), \(v_1 = v_2 = 1\), \(n \gg 1\)
  - \(C = 6\)

picks the first object, w/ no room for second, would have been better off picking the second!

[Khanna, Kim, 1994]: show that 3 FPTAS that approximate Knapsack

[Khanna, Madan, 2002]: show that even with good approximations, applying the Sh hierarchy does not quickly reduce the integrality gap

Formally, the strong integrality convergence statement can be expressed as the following two:

- **Theorem 1**: For every \(\epsilon, \delta \geq 0\), the integrality gap at the \(t\)th level of the Sh hierarchy for knapsack where \(t \leq 5n\) is at least \((2 - \epsilon)(1/(1 + \delta))\)

**Proof sketch**: (full proof given in handout in lecture)

1) Consider an instance \(K\) of knapsack with \(n\) variables / objects
2) Let \(K\) be instance of knapsack where \(c_i = v_i = 1\) (uniform)
3) Let \(C = 2(1 - \epsilon)\) so that we can only fit 1 object in the bag.

Thus, \(\text{OPT} = 1\)

4) Claim: for any \(y \in \{0, 1\}^n\), \(y_i = \frac{C}{(1 + (1 - \epsilon)(1 + \epsilon))} \sum_{j \in [n]} y_j v_j \geq 1\)

is in \(S^+(n)\)

Note: this point gives \(k_i = \frac{C}{(1 + (1 - \epsilon)(1 + \epsilon))} \sum_{j \in [n]} y_j v_j\)

and that \(\sum_{i \in [n]} y_i c_i = \sum_{i \in [n]} y_i c_i \geq \sum_{i \in [n]} \frac{C}{(1 + (1 - \epsilon)(1 + \epsilon))} \sum_{j \in [n]} y_j v_j \geq \frac{Cn}{(1 + (1 - \epsilon)(1 + \epsilon))} \geq (2 - \epsilon)\) when \(t \geq 5n\)
Key takeaway: applying SA to simple $k$ is not as good as we thought.

What to do?

Rewrite LP as feasibility LP, then SA, give much better ind. gap.

Find $y_1, y_2, \ldots, y_n$ such that

$$\sum y_i v_i \geq R \quad g'(y)$$
$$\sum y_i c_i \leq C \quad g^2(y)$$
$$0 \leq y_i \leq 1 \quad \forall i \in [n]$$

**Theorem 3**

Integrality gap of 1-level SA applied to the feasibility LP is $\leq \frac{1}{1+\epsilon}$. 

Key takeaway: By manipulating the LP before applying hierarchies, we can do much better.

**Lemma**

Let $S_{i-1} = \{ i : i > 0 \text{ or } i < 1 \}$. At most $(1+\epsilon)$ items from $S_{i-1}$ can fit in the bag if $j \in S_{i-1}$ with non-zero LP value, add constraint on LP to pick $x_i$.

Almost, we repeat (1+1) steps and the remaining items in $S_{i-1}$ have LP value = 0.

$K_0$ set of items chosen (non-zero LP value from before). $R_0$ = $r(K_0)$ removed.

Consider current LP solution $y'$ restricted to $(n) \setminus S_{i-1}$. New LP value = $\sum_{j \in S_{i-1}} y'_j \geq R - R_0$. Apply greedy alg. to remaining items, give additional reward $R_y$.

From lemma:

$$R - R_0 \leq \sum_{j \in S_{i-1}} y'_j \leq R_g + \max_{i \in S_{i-1}} c_i \leq R_g + \frac{1}{1+\epsilon} \leq R \left( 1 - \frac{1}{1+\epsilon} \right)$$

Value of returned solution $R_0 + R_g \geq R \left( 1 - \frac{1}{1+\epsilon} \right)$