Packing via Covering and LP–Relative Approximation

Jochen Könemann, Ojas Parekh, David Pritchard

Department of Combinatorics & Optimization, U Waterloo, ON

CANADAM 2009, May 27
Linear Program–Based Combinatorial Optimization in a Nutshell

- LPs used for exact algorithms since mid-century (e.g., flow, matching, matroids)
  - Literature developed lots of proof techniques & algorithmic techniques (LP duality, total unimodularity, uncrossing, ellipsoid algorithm)

- Broad base of knowledge applied also to approximation algorithms ~30 years ago
  - Spurred more techniques, e.g. primal–dual schema, randomized rounding, scaling, grouping, iterated rounding/relaxation
Technique–Based Theory

- LP–based combinatorial optimization thereby exhibits the best & worst of mathematical problem-solving:
Technique–Based Theory

- LP–based combinatorial optimization thereby exhibits the best & worst of mathematical problem–solving:
  - Best: techniques are elementary when considered individually, and combining them gives some very great results
  - Worst: it can be very hard to figure out how to combine them!
Approximation algorithms

For NP-hard optimization problems optimal value OPT cannot be found by a poly-time algorithm (unless P=NP)

Thus in poly-time the best we can do is find an approximately optimal answer

Our convention: An $\alpha$-approximation algorithm for a “max” problem always returns feasible solution with value at least OPT/$\alpha$. For “min” problem, always returns value at most OPT*$\alpha$. ($\alpha \geq 1$; exact $\equiv \alpha = 1$)
Iterated LP Relaxation

- Background: Iterated relaxation (Lau, Naor, Salavatipour, Singh, STOC 2007) finds solutions with additive violation of constraints instead of multiplicative factor

  - Bounded degree spanning tree: can find a spanning tree with super-optimal weight but violating degree bounds by up to +1 (SL ‘07)
  - Integral multicommodity flow in a tree: can find a flow with super-optimal weight but violating edge capacities by up to +2 (KPP ‘08)
Contribution of Our Work

- Can iterated rounding help when constraints are inflexible?
Contribution of Our Work

- Can iterated rounding help when constraints are inflexible?
- Our contribution: Techniques that remove violation (at expense of value)
  - General approach that works in most iterated rounding situations (where a “counting lemma” exists)
- Gives new algorithmic results
  - Ultimate goal: understanding best possible approximation ratios for problems
Rest of Talk

- Def\(^n\): Multicommodity Flow in a Tree (MFT)
- Def\(^n\): LP–Relative Approximation Ratio
- Counting lemma for MFT and consequences
- Results in case of MFT:
  - \(1+1/O(\text{minimum capacity})\) approx algorithms
- Mini–Technique: LP–Relative Covers Help Cover
- Technique: LP–Relative Covers Help Pack
- General Form of Results, Open Problems
Integer multicommodity flow in a tree (MFT)

- **Input**: tree with edge capacities $c_e$; pairs of terminals, profit $w_i$ for each commodity $i$.
  - Let $path(i)$ denote path between terminals for $i$
- **Goal**: integers $x_i \geq 0$ such that for each $e$
  \[
  \sum_{i : e \in path(i)} x_i \leq c_e
  \]
  such that $\sum_i w_i x_i$ is maximized
  - E.g. ship kegs on tree network
- APX–complete (GVY ‘93),
  4–apx (CMS ‘03)
Key Notion: LP–Relative Approximation

- *Natural LP* for MFT: flow values $x$ can be fractional, otherwise the same. LP–OPT can be found in poly–time. Note LP–OPT $\geq$ OPT.
- Def. An *LP-relative $\alpha$–approx. algorithm* for a “max” problem always returns value at least LP–OPT/$\alpha$; “min” is analogous
  - Same as “$\alpha$–approx.” def with OPT $\Rightarrow$ LP–OPT
  - Stronger notion
  - Abundant in papers but no common term?
    - E.g. 4–approx for MFT, min degree+1 spanning tree
(KPP 08) “Let \(x\) be an extreme solution to the natural LP for MFT. If \(x_i < 1\) for every commodity \(i\), then some edge \(e^*\) has the following property:
- At most 3 commodities \(i\) have \(x_i > 0\) and \(i\) in \(\text{path}(e^*)\)”

Same works for covering version of MFT

(‘09) Also holds for arc- and vertex-capacitated versions of MFT, with 3 replaced by 7
- Won’t state these versions explicitly from now on but all results in this talk go through
Counting Lemma Consequences

- i) Capacitated covering–MFT: some var has value $\geq 1/3 \Rightarrow 3$–apx by iterated rounding

- ii) (1, +2)–approximation for MFT by iterated relaxation as mentioned earlier

- iii) (1, –2)–approximation for covering–MFT

Key: these results are all LP–relative!
Results as Applied To MFT

- Let $\mu$ be minimum (edge/arc/vertex) capacity
- We get $1 + O(1/\mu)$ approximation algorithm
  - Previous best ratio: constant (4/4/5)
  - Asymptotically optimal in terms of $\mu$: $\exists \epsilon > 0 \forall \mu$, we get $1 + O(\epsilon/\mu)$ inapproximability
  - Also get $1 + O(1/\mu)$ approximation algorithm for covering-MFT

- Next up: sketches of the techniques
Mini–Technique: LP–Relative Covers Help Cover (e.g.: MFT)

- There is a $\left(1 + \frac{2}{\mu}\right)$-apx alg for covering-MFT

Proof

- Artificially increase all requirements $c_e$ by 2
- Scaling: if $x$ is an optimum to the old LP, then $x^*\left(2 + \mu\right)/\mu$ is a solution to the new LP
- So LP–OPT’ $\leq$ LP–OPT*$\left(2 + \mu\right)/\mu$
- Now apply the LP–relative (1, –2) approximation to the new requirements, gives a solution which meets old requirements and has cost at most

$$1*\text{LP–OPT’} \leq \text{LP–OPT}*\left(2 + \mu\right)/\mu$$
Technique: LP–Relative Covers Help Pack (e.g.: MFT)

- There is a $1+O(1/\mu)$-apx alg for MFT

Proof
- Let $x$ be output of $(1,+2)$ approx. algorithm
- For each $e$ let $f_e$ be overload of $e$ by $x$ ($0 \leq f_e \leq 2$)
- Look @ capacitated MFT–covering w/ requirements $f$ and capacities $x$:
  - Scaling: $x^*2/(2+\mu)$ is a feasible fractional covering
  - So LP–OPT’ $\leq c(x)^*2/(2+\mu)$
- Run 3–apx alg for capacitated cover–MFT $\Rightarrow y$
- $x – y$ is a solution to original MFT instance and $c(x–y) \geq c(x)^*(1–3*2/(2+\mu)) \geq \text{LP–OPT}/(1+O(1/\mu))$
Generalized Results

- Consider a family of integer linear programs \((A, b)\)
- Define \(\mu := b_{\min}\)
- If “counting lemma” \(\Rightarrow\) known LP-rel consequences
  - \((\alpha', -v')\) for uncapacitated covering \(\min \{cx | Ax \geq b, 0 \leq x\}\)
  - \((\alpha, +v)\) for capacitated packing \(\max \{cx | Ax \leq b, 0 \leq x \leq d\}\)
  - \(\beta\) for capacitated covering \(\min \{cx | Ax \geq b, 0 \leq x \leq d\}\)
- Our techniques give in addition:
  - \(\alpha'(1+v'/\mu) = \alpha'(1+O(1/\mu))\) for covering
  - \(\alpha/(1- \beta v/(\mu+v)) = \alpha(1+O(1/\mu))\) for capacitated packing
- **Remark**: can’t hope to \(1+O(1/\mu)\)-approx **capacitated covering** in general since we can increase \(\mu\) artificially
Related Open Problems

- Smallest $k$-edge-connected subgraph [GG 08]: best known apx ratio is $1 + 1/2k + O(1/k^2) + \ldots$
  - What about *min-cost* $k$-edge-connected multisubgraph?
    - Best known apx is 2, can we get $1+O(1/k)$?

- [*] Our packing techniques give no result when $\mu$ is too small. Is this avoidable or not?
  - E.g. *column-sparse packing ILPs* admit a counting lemma but need ad-hoc techniques to get const ratio for small $\mu$.

- Demand multicommodity flow in a tree?
  - Case of a star (*demand matching*) is well-studied and also a special case of column-sparse integer programs
    - $1+O(dem_{\max}/\mu)^{1/2}$ best known, can we get $1+O(dem_{\max}/\mu)$?
Merci!