Polynomial Time Approximation Schemes for Dense Instances of $\mathcal{NP}$-Hard Problems

Sanjeev Arora   David Karger   Marek Karpinski

STOC '95

GSIA

Polynomial Integer Programs

- A *Polynomial Integer Program* (PIP) in general form, is:

$$\begin{aligned} \min \quad & f(x) \\ \text{subject to} \quad & g_i(x) \geq 0 \\ & x \in \{0,1\}^n \end{aligned}$$

- The polynomials $f(x), g_i(x)$ are $c - smooth$. Recall: $c - smooth$ means the absolute value of each degree i term is at most cn^{d-i}.
- The polynomials are of *low degree.*
- This framework can be used to model various combinatorial optimization problems, as we shall see.

Exhaustive Sampling:

- We want a "good" 0-1 assignment of an $n-$vector x.
- We take a *sample* subset S of size $O(\log n)$.
- We try *all* possible assignments of S. One of these is the optimal assignment. There are $2^{|S|}$ possibilities, but $2^{c \log n} = n^c$, which is polynomial!
- In a dense instance, we can extend the assignment of S to an assignment of the entire vector x, without losing too much in terms of either *optimality* or *feasibility*.

Solution Strategy

$$\boxed{\begin{array}{lr} & \min \quad f(x) \\ \textbf{PIP} & \text{subject to: } \mathcal{P}(x) \\ & x \in \{0,1\}^n \end{array}}$$

$\Downarrow$ Feasibility, Exhaustive Sampling $\Uparrow$

$$\boxed{\begin{array}{lr} \textbf{ILP} & Ax \leq b \\ & x \in \{0,1\}^n \end{array}}$$

$\Downarrow$ Randomized Rounding $\Uparrow$

$$\boxed{\begin{array}{lr} \textbf{LP} & Ax \leq b \\ & 0 \leq x \leq 1 \end{array}}$$

Approximating Smooth PIPs

Lemma 1.1 *A $c-$smooth polynomial of degree d can be written as $p(x) = t + \sum x_i p_i(x)$ where t is a constant and each p_i is a $c-$smooth polynomial of degree $d-1$.*

Proof:

Extract any variable x_i from each monomial term in p, and group all monomials from which x_i was extracted into p_i.

Every degree d' term in p_i corresponds to a degree $d'+1$ term in p.

Hence its coefficient is no more than $cn^{d-(d'+1)} = cn^{(d-1)-d'}$.

Hence it is $c-$smooth.

■

Approximating Smooth PIPs - Absolute value bound

Lemma 1.2 *If $n > 2d$, then the absolute value of a polynomial at any point in $[0,1]^n$ is at most $2cen^d$.*

Proof: For any $0 \le i \le d$, we have at most $\binom{n+i}{i}$ terms of degree i. Each has a coefficient in $[-cn^{d-i}, cn^{d-i}]$. Hence for any point in $[0,1]^n$, an upper bound on the absolute value of the polynomial is:

$$\begin{aligned}
\sum_{i=0}^{d} cn^{d-i}\binom{n+i}{i} &\le \sum_{i=0}^{d} cn^{d-i}\binom{n+d}{i} \\
&\le cn^d \sum_{i=0}^{d}\left(\frac{n+d}{n}\right)^i \frac{1}{i!} \\
&\le cn^d e^{1+d/n} \\
&\le cen^d\left(1+\frac{2d}{n}\right) < 2cen^d
\end{aligned}$$

when $n > 2d$ ■

Reducing Optimization to Feasibility

GSIA

- We need to search in an interval $[-U, U]$ for the maximum value k attained by a feasible solution.

Hence we can transform an Optimization PIP:

$$\begin{aligned} \min \quad & f(x) \\ \text{subject to} \quad & \mathcal{P}(x) \\ & x \in \{0,1\}^n \end{aligned}$$

to a feasibility PIP. We search for a feasible solution to the following:

$$\begin{aligned} & \mathcal{P}(x) \\ & f(x) \geq k \\ & x \in \{0,1\}^n \end{aligned}$$

Approximating Smooth PIPs - Estimating polynomial value

Suppose the optimum of the PIP lies at $x = a$. Given the value of $O(\log n)$ randomly chosen a_i's, we estimate the value of $p(x)$ at a that whp lies in $p(a) \pm \epsilon n^d$.

In the following, let $p(x) = t + \sum x_i p_i(x)$ be the recursive expansion of $p(x)$. Let S be the set of indices for which we know the values a_i. Then we define the function `Eval` as:

$$\texttt{Eval}(p, S, \{a_i : i \in S\}) := t + \frac{n}{|S|} \sum_{i \in S} a_i \texttt{Eval}(p_i, S, \{a_i : i \in S\})$$

We can prove the following lemma about the "goodness" of the estimate returned by `Eval`.

Lemma 1.3 *Let* $a_i \in \{0, 1\}$. *Let* S *be a set of* $O(g \log n)$ *indices chosen from* $[n]$ *randomly with replacement. Then with probability at least* $1 - n^{d-f}$, S *is such that* `Eval`$(p, S, \{a_i : i \in S\})$ *returns a value in* $p(a) \pm \epsilon n^d$, *where* $\epsilon = 4ce\sqrt{f/g}$.

Proof (Intuition):

$$
\begin{aligned}
& t + \frac{n}{|S|} \sum_{i \in S} a_i e_i && \text{(output of } \texttt{Eval}\text{)} \\
\approx \quad & t + \frac{n}{|S|} \sum_{i \in S} a_i \rho_i && \text{(by inductive hypothesis)} \\
\approx \quad & t + \sum_{i} a_i \rho_i && \text{(by Sampling lemma)}
\end{aligned}
$$

■

Approximating Smooth PIPs - Linearizing Constraints

GSIA

Let $L \leq p(x) \leq U$ be a polynomial degree$-d$ costraint, and suppose we have a partial feasible solution vector $\{a_i : i \in S\}$. Let $p(x) = t + \sum x_i p_i(x)$. Then we can use the following algorithm to linearize the constraint recursively:

$\texttt{Linearize}(\text{``}L \leq p(x) \leq U\text{''}, S, \{a_i : i \in S\}, \epsilon)$

- $\texttt{for}$ $i = 1$ $\texttt{to}$ n
  - $e_i = \texttt{Eval}(p_i, S, \{a_i : i \in S\})$
  - $[l_i, u_i] = e_i \mp \epsilon n^{d-1}$
  - $\texttt{Linearize}(\text{``}l_i \leq p_i(x) \leq u_i\text{''}, S, \{a_i : i \in S\}, \epsilon)$
- $\texttt{output}$ $L - \epsilon n^d \leq t + \sum x_i e_i \leq U + \epsilon n^d$

Lemma 1.4 *Let $f, g, c > 0$ be constants. Let* `Linearize` *be given an error parameter $\epsilon = 4ce\sqrt{f/g}$, and a $c-$smooth degree$-d$ polynomial constraint. Let $a \in \{0,1\}^n$ be feasible for the constraint. S is a random sample of $g \log n$ variables, then with probability at least $1 - dn^{d-f}$,* `Linearize` *outputs a set of linear constraints satisfied by a.*

Lemma 1.5 *Every feasible solution $y \in [0,1]^n$ to the set of linear constraints output by* `Linearize` *is approximately feasible for the original (polynomial) constraint:*

$$L - 2d\epsilon n^d \leq p(y) \leq U + 2d\epsilon n^d$$

Lemma 1.6 *Given fractional values* (y_i) *s.t.* $p(y) = b$*, we use randomized rounding to get integer values* z_i*, and* $g = 2ce\sqrt{f}$*. Then with probability at least* $1 - n^{d-f}$*, we have*

$$p(z) \in b \pm gdn^{d-1/2}\sqrt{\log n}$$

Proof:

An extension of the Raghavan-Thompson technique, proved by induction. ■

GSIA

Applications - Dense $k-$subgraph

- Let $k \geq \alpha n$, and the graph be $\delta-$dense.
- Densest subgraph has at least $\alpha^2 \delta n^2/2$ edges.
- 1-Smooth PIP formulation:

$$\begin{aligned} \max \quad & \sum_{(i,j)\in E} x_i x_j \\ \text{subject to} \quad & \sum x_i = k, \qquad x \in \{0,1\}^n \end{aligned}$$

- We get a solution which is *approximately feasible*, ie $\sum x_i \in k \pm g\sqrt{n \log n}$. Hence we move at most $g\sqrt{n \log n}$ vertices to get a feasible solution, losing at most $o(n^2)$ edges.
- Solution is within additive ϵn^2 of optimal.

Applications - MAX $k-$SAT

- Given $m \geq \delta n^k$ clauses.
  - # clauses of size $k = m - O(n^{k-1})$.
  - Random assignment satisfies each with probability $1 - 2^{-k}$
  - $OPT \geq (1 - 2^{-k})(m - O(n^{k-1}))$.
- Standard Arithmetization to represent each clause as a degree$-k$ smooth PIP. Let t_j be the term for the j^{th} clause.
- PIP formulation:

$$\begin{array}{rl} \max & \sum t_j(x) \\ \text{subject to} & x \in \{0,1\}^n \end{array}$$

- Hence in time $O(n^{2^{4k}/\epsilon^2})$, we find an assignment satisfying $OPT - \frac{\epsilon}{2^k} n^k \geq (1-\epsilon)OPT$ clauses.

Applications - Separator

GSIA

- Guess the number k of vertices on one side by trying all $n/3$ possibilities.
- PIP formulation:

$$\begin{array}{rl} \min & \displaystyle\sum_{(i,j)\in E} [x_i(1-x_j) + x_j(1-x_i)] \\ \text{subject to} & \sum x_i = k \\ & x \in \{0,1\}^n \end{array}$$

- We get an *approximately feasible* solution, within additive ϵn^2 of the optimal.