

Lift and Project Techniques

Jacob Denson

December 9, 2016

One of the most useful applications of linear programming in NP-hard combinatorial optimization is to obtain easy approximation algorithms. Indeed, many NP-hard problems can be fairly effortlessly formulated as integer linear program. If we solve the corresponding non-integer linear program, we obtain a non-integral solution, and if we round this solution, we obtain a solution which likely gives a good approximate solution. Indeed, in many NP-hard problems this gives us approximation algorithms with the best theoretical bound. Similarly, if we have a non-linear program, we can approximate it by a linear program. The Lift-and-Project technique gives a systematic method of placing additional constraints on the program. We *lift* our problem to a new linear program with polynomially many new variables, solve this equation, and then *project* back into the original domain to get more accurate integer solutions. Here we discuss the Sherali Adams projection technique.

1 The Sherali-Adams Lift and Project Method

Consider the zero-one integer linear programming problem. That is, we take a region

$$K = \{x \in \{0,1\}^n : Ax \geq b\} \cap \{0,1\}^n$$

and suppose we wish to find a point $x \in \{0,1\}^n \cap K$ which maximizes λx . Like integer linear programming, this problem is NP hard in general. The standard way to approximate a solution by relaxing the constraints, solving the corresponding linear program K_0 . Without loss of generality, we may assume $K_0 \subset [0,1]^n$, by introducing the constraints $0 \leq x_i \leq 1$ to the problem. The trick is to add additional constraints to K_0 which ‘chop off’ non-integer solutions, while still preserving the ability to approximate an optimal solution.

If we were living in a perfect world, we’d be able to solve quadratic programs in polynomial time, so we could introduce the constraints

$$x_i(1 - x_i) = 0$$

to the problem, and then we’d be done. Unfortunately, in real life we have to find ways around this. For zero-one integer linear programs, noticing a certain algebraic identity on $\{0,1\}$. Indeed, this can be reexpressed as $x_i^2 = x_i$, so that

any polynomial can be reduced to a polynomial which is linear in each variable. The Sherali-Adams technique offers a way to obtain better approximations for zero-one linear programs.

For disjoint subsets $J, K \subset [n]$, consider the polynomials

$$P_{J,K}(X) = \left(\prod_{j \in J} x_j \right) \left(\prod_{k \in K} (1 - x_k) \right)$$

Note that for any $x \in \{0, 1\}^n$, $P_{J,K}(x) \geq 0$. We will say that $P_{J,K}$ is **order** m if $|J \cup K| = m$. For a fixed m , the number of polynomials with order m is

$$\binom{n}{m} 2^m \leq \left(\frac{2^m}{m!} \right) n^m \leq 2n^m = O(n^m)$$

Now consider the following procedure. Add the constraints $P_{J,K}(x) \geq 0$, and multiply the existing constraints in A by $P_{J,K}(x)$ to form a system of polynomial constraints. Perform the reduction $x_i^2 = x_i$ to ensure the polynomial constraints are linear in each variable. Note that these constraints have shrunk our solution set, while still keeping the zero-one solutions in the set. However, we cannot solve a program with polynomial constraints, so we need to perform a further relaxation, so for each $J \subset [n]$, substitute a variable w_J into the constraints above so that the constraints are now linear. The only question now is whether we have introduced any more undesired solutions. Let K_m be the solution set to this equation.

Lemma 1. *For any $1 \leq m < n$, the constraints for polynomials of order m are implied by polynomials of order $m + 1$.*

Proof. Given $P_{J,K}$ of order m , let $p \in [n] - (J \cup K)$. Then

$$P_{J+p,K}(x, w) + P_{J,K+p}(x, w) = P_{J,K}(x, w)$$

If $P_{J+p,K}(x, w) \geq 0$, $P_{J,K+p}(x, w) \geq 0$, then we get $P_{J,K}(x, w) \geq 0$ for free. \square

Thus we have a heirarchy $K_0 \supset K_1 \supset K_2 \supset \dots$, so our solution sets get tighter over time, as we add more and more constraints. Thus we can obtain better approximation algorithms, provided that we are willing to have a more complex algorithm.

Now we'll prove that our approximation sufficiently covers the integer solutions we need. Let $Z_m = \{(x, w) : P_{J,K}(x, w) \geq 0, P_{J,K} \text{ is order } m\}$, so that $Z_m \cap K_0 = K_m$. If we can verify that the binary vectors of Z_m are exactly those of the original set, we'll have shown that we can still find the optimum solution in the set considered.

Lemma 2. *If $x \in \{0, 1\}^n$, then $(x, w) \in Z_m$ iff $w_J = \prod_{j \in J} x_j$ for all $|J| \leq m$.*

Proof. If $w_J = \prod_{j \in J} x_j$ holds, then we have already shown that $(x, w) \in Z_m$, because then $P_{J,K}(x, w) = P_{J,K}(x) \geq 0$, because these polynomials only feature

monomials of degree at most m . We prove the converse by induction. Note that it is true by definition for $m = 1$. By the previous lemma, if $(x, w) \in Z_m$, then $(x, w) \in Z_{m-1}$, so $w_J = \prod_{j \in J} x_j$ for all $|J| < m$. Now suppose $|J| = m$. Then if $k \neq l$ are elements of J , then

$$P_{J-\{k\},\{k\}}(x, w) = \left(\prod_{j \neq k} x_j \right) - w_J \geq 0$$

which makes sense, because if we multiply by a number by a number between 0 and 1, we can only decrease the value. And

$$P_{J-\{k,l\},\{k,l\}}(x, w) = (1 - x_k - x_l) \left(\prod_{j \neq k,l} x_j \right) + w_J \geq 0$$

which again makes sense, because if we multiply w_J by a number less than 1, we should get a number smaller than it. Thus

$$(x_k + x_l - 1) \prod_{j \neq k,l} x_j \leq w_J \leq \prod_{j \neq k} x_j = x_l \prod_{j \neq k,l} x_j$$

If $\prod_{j \neq k,l} x_j = 0$, then we obtain the inequality $0 \leq w_J \leq 0$, so

$$w_J = 0 = \prod_{j \in J} x_j$$

Otherwise, $\prod_{j \neq k,l} x_j = 1$, and we divide out and find $x_k + x_l - 1 \leq w_J \leq x_l$. If $x_k = x_l = 1$, then $\prod_{j \in J} x_j = 1$, and we have $1 \leq w_J \leq 1$. We know $w_J \geq 0$, so if $x_l = 0$ then we conclude $w_J = 0 = \prod_{j \in J} x_j$. Similar results hold if $x_k = 0$. This covers all cases, and completes the proof. \square

We conclude that the set of $x \in \{0, 1\}^n$ for which there is w such that $(x, w) \in K_m$ is the same as the set of $x \in K$, and finding optima over K_m will find at least a good approximation as K_0 . It turns out that this ‘heirarchy’ of sets approximates K as much as can be done with a convex set. This is slightly more tricky than the previous result, where we only had to do computation of $\{0, 1\}^n$.

Lemma 3. *For a fixed (x, w) , the linear endomorphism generated by the maps*

$$w_J \mapsto P_{J,J^c}(x, w) = y_J$$

is invertible (where we think of x_i as $w_{\{i\}}$), with inverse

$$w_J = \sum_{K \subset J^c} y_{J \cup K} = \sum_{J \subset K} y_K$$

Proof. The proof is just an algebraic calculation. First, note that

$$P_{J,J^c}(x, w) = \sum_{K \subset J^c} (-1)^{|K|} w_{J \cup K}$$

We have

$$\begin{aligned} \sum_{K \subset J^c} P_{J \cup K, (J \cup K)^c}(x, w) &= \sum_{K \subset J^c} \sum_{L \subset (J \cup K)^c} (-1)^{|L|} w_{J \cup K \cup L} \\ &= \sum_{M \subset J^c} w_{J \cup M} \left(\sum_{L \subset M} (-1)^{|M-J|} \right) \\ &= \sum_{M \subset J^c} w_{J \cup M} \left(\sum_{L \subset M} (-1)^{|L|} \right) \end{aligned}$$

and the sum vanishes for non-empty set above, because we have a bijection that reverse the sign of $(-1)^{|L|}$. \square

Lemma 4. *The extreme points of*

$$Z = \{(x, w) : (\forall J : P_{J,J^c}(x, w) \geq 0)\}$$

are zero-one integral in the projection.

Proof. Consider the set

$$Z = \{(x, w) : (\forall J : P_{J,J^c}(x, w) \geq 0)\}$$

Since $\sum_{j \in J} P_{J,J^c}(x, w) = 1$, we have

$$P_{\emptyset, N} = 1 - \sum_{\emptyset \subsetneq J \subset N} P_{J,J^c}$$

Thus

$$Z = \left\{ (x, w) : (\forall J \neq \emptyset : P_{J,J^c}(x, w) \geq 0), \sum_{\emptyset \subsetneq J \subset N} P_{J,J^c}(x, w) \leq 1 \right\}$$

Now the linear map f transforms Z into

$$S = \left\{ y : \sum_{\emptyset \subsetneq J \subset N} y_J \leq 1, (\forall J \neq \emptyset : y_J \geq 0) \right\}$$

which has the some polyhedral properties as Z , in the sense that extreme points are mapped to extreme points, etc. But the extreme points of S are just the unit vectors or 0, but this implies that the extreme points of Z are in $\{0, 1\}^n$, for this implies that all the $P_{J,J^c}(x, w)$ are equal to zero one in the inverse image. \square

Theorem 5. For any $(x, w) \in Z$, the map $J \mapsto w_J$ is monotone decreasing.

Proof. Now we just verify that for any $(x, w) \in Z$, the map $J \mapsto w_J$ is monotone decreasing. Indeed, if $J \subset K$, and from the non-negativity of the y in S , and since $J^c = K^c \cup (K - J)$

$$w_K = \sum_{L \subset K^c} y_{L \cup K} \geq \sum_{L \subset J^c} y_{L \cup (K - J) \cup J} = w_J$$

So the inequality is proven. \square

Theorem 6. K_n is the convex hull of K .

Proof. Suppose that $(x, w) \in K_n$ is an extreme point. Then $w_J = \prod_{j \in J} x_j$ for all J . If we can verify that the w_J are binary, then x must be binary, so $x \in K$, and this shows that K_n is the convex hull of K . Now we assume that K is defined by the inequalities

$$\sum a_i^j x_i \leq b_j$$

Then K_n is the set of points (x, w) defined by

$$\left(\sum a_i^j - b_j \right) P_{J, \bar{J}}(x, w) \geq 0$$

Define E to be the set of J such that $\sum_{j \in J} a_j^k - b_k < 0$ holds for some k . If $J \in E$, then the constraints corresponding to J can be rephrased as $P_{J, \bar{J}}(x, w) = 0$, while for $J \in E^c$ this is equivalent to $P_{J, \bar{J}}(x, w) \geq 0$. Hence

$$K_n = \{(x, w) : (\forall J \in E : P_{J, \bar{J}}(x, w) = 0), (\forall J \in E^c : P_{J, \bar{J}}(x, w) \geq 0)\}$$

But then K_n is just a subface of the set Z considered in the last proof, and so the extreme points of K_n are all extreme points of Z , and we just verified these are binary. \square

Note – if the optimal non-integer solution lies in the convex hull of the integer points of K , then we can always construct a randomized algorithm which achieves optimality in expectation. Indeed, if $x \in K_n$ is optimum, and if

$$x = \sum a_i x_i$$

where $x_i \in K$, then if we choose x_i with probability a_i , we achieved the value of x in expectation. Even just choosing an x_i results in an optimal solution, because $\lambda x = \sum a_i \lambda x_i$, and if $\lambda x_i > \lambda x$, then we could conclude $\sum a_i \lambda x_i > \lambda x$. Unfortunately, using K_n often results in an exponential time algorithm, and even if we could find $x \in K_n$ in polynomial time, it is unlikely we can compute the a_i in polynomial time, because the x might be written as a linear combination of exponentially many points x_i . These ideas return in more sophisticated heirarchical lift and project techniques, like in the Lessaare heirarchy.

There are some details we haven't discussed here. Using some clever algebraic manipulations, we can reduce the number of constraints we add in the

heirarchy. We also note that we can extend these methods to quadratic programming, in essentially the same way. If you're curious, the paper is out there for you to read.

In 1973, Chvatal proved that any linear inequality in an integer program can be derived through some constraint surrogations, followed by rounding, but it wasn't until this paper that there was a constructive characterization of the convex hull. Consideration of K_1 has already been done, but the Sherali Adam's heirarchy provides a full range of approximations of the convex hull, and this allows us to calculate explicit inequalities for approximation algorithms.

2 A Quick Example: Vertex Cover

Let's give a quick example of how the method applies to a real problem. The vertex cover problem on a graph $G = (V, E)$ can be formulated as the integer linear program of minimizing $\sum x_i$, such that $x_i + x_j \geq 1$ for $ij \in E$. If we just relax the integer linear program to $0 \leq x_i \leq 1$, then we have an integrability gap of 2 – that is, the minimum $\sum x_i$ we find over the linear domain is at most twice the size of the integer solution.

So let's use the heirarchy to construct a better linear program. If we consider the first level extension, then we first construct the quadratic program. For all i, j , we should have

$$(1 - x_i)(1 - x_j), x_i(1 - x_j), x_i x_j \geq 0$$

and if $ij \in E$, then

$$(x_i + x_j)(1 - x_i)(1 - x_j) \geq (1 - x_i)(1 - x_j)$$

$$(x_i + x_j)x_i(1 - x_j) \geq x_i(1 - x_j)$$

There are certainly some limitations to this method, and the integrability gap has been heavily studied in other papers, but that's not the topic of my talk!

References

- [1] Thomas Rothvoß, *The Lasserre Heirarchy in Approximation Algorithms*, MAPSP 2013 Tutorial, 2013.
- [2] Sherali, Adams, *A Heirarchy of Relaxations Between the Continuous and Convex Hull Representations for Zero-One Programming Problems*, SIAM Journal of Discrete Mathematics, 1990.
- [3] Chlamtac, Tulsiani, *Convex Relaxations and Integrality Gaps*, 2013.