**CSC 2429 : Proof Complexity, Mathematical Programming and Algorithms**

# How to Round any CSP

*Date: 28 March 2018* *Scribe: Deeksha Adil, Noah MacAulay*

## 1. Definition of CSP

Here we give our definition of a CSP. Our CSPs will be over a set of $n$ variables $V = \{x_1, ..., x_n\}$, each variable ranging over $q$ discrete values.

Each type, or family of CSPs is defined by a family of payoff functions $\Lambda$. The functions in $\Lambda$ are maps from $[q]^k$ to $[-1, 1]$, for some fixed integer $k$, the arity of the payoff functions.

Given a payoff function $\lambda : [q]^k \to [-1, 1]$ and a set of variables $V = \{x_1, ..., x_n\}$, an application of $\lambda$ to $V$ is a function $P(x_1, ..., x_n) \to [-1, 1]$ such that $P(x_1, ..., x_n) = \lambda(x_{i_1}, ..., x_{i_k})$ for some $i_1, ...i_k \in [n]$. An instance of a $\Lambda$-CSP consists of a set of applications of the payoff functions in $\Lambda$ to $V$. Our goal is to maximize the sum of each of the application of the payoff functions; that is, we wish to find

$$\operatorname*{argmax}_x \sum_{P \in \mathcal{P}} P(x)$$

As an example, let's cast MAX-3-SAT in this format. Here $q = 2$ and $k = 3$. The family $\Lambda$ in this case is the set of 8 possible "or" functions on 3 boolean variables, $\lambda_{000}, \lambda_{001}, ...\lambda_{111}$, e.g.

$$\lambda_{000}(x, y, z) = \begin{cases} -1 & \text{if } x, y, z = 0 \\ 1 & \text{otherwise} \end{cases} \tag{1}$$

$$\lambda_{001}(x, y, z) = \begin{cases} -1 & \text{if } x, y = 0, z = 1 \\ 1 & \text{otherwise} \end{cases} \tag{2}$$

A single application of one of these functions is a clause, and an instance is a formula. Maximizing the sum of the payoff functions is equivalent to maximizing the number of satisfied clauses.

## 2. SDP Relaxation

Our SDP Relaxation will contain two sets of variables, $\{v_{i,a}\}$ and $\{\mu_{P,x}\}$.

For each variable $x_i$ in the original CSP, we introduce $q$ vector-valued variables $v_{i,a}$. The $v_{i,a}$ have dimension $D$ that grows with $n$, say $D = q|V|$. The variable $v_{i,a}$ can be thought of as representing the probability that $x_i = a$.

For each application of a payoff function $P$, we introduce a "local probability distribution" $\mu_P$. These are probability distributions over assignments to the variables $x_{i_1}, ..., x_{i_k}$ that $P$ is assigned to. Each such local distribution has $q^k$ components.

Our goal is now to maximize the "expected value" of the payoff functions under their local distributions:

$$\operatorname*{argmax}_{\mu_{P,x}} \sum_{P \in \mathcal{P}, x \in [q]^k} \mu_{P,x} P(x)$$

To enforce consistency between the different local distributions we introduce the following constraints:

$$\langle v_{i,a}, v_{j,b} \rangle = Pr(x_i = a, x_j = b) = \sum_{x_i=a, x_j=b} \mu_{P,x}$$

$$\langle v_0, v_{i,a} \rangle = Pr(x_i = a) = \sum_{x_i=a} \mu_{P,x}$$

Here $v_0$ is an arbitrary fixed unit vector. The 2nd equality of each line holds for all $P$ containing $x_i$ and $x_j$. We also have constraints enforcing that each $\mu_P$ is a probability distribution: for all $P$, $x$, $\mu_{P,x} \geq 0$, and $\sum_x \mu_{P,x} = 1$.

We note some implications of the constraints. These constraints imply that the pairwise marginals are the same between clauses. They also imply that $\sum_a v_{i,a} = v_0$ for all $i$. To see this, first note that $\sum_a v_{i,a}$ is of unit length, because

$$\langle \sum_a v_{i,a}, \sum_a v_{i,a} \rangle = \sum_{a,b} \langle v_{i,a}, v_{i,b} \rangle = \sum_a Pr(x_i = a) = 1$$

Additionally,

$$\langle \sum_a v_{i,a}, v_0 \rangle = \sum_a \langle v_{i,a}, v_0 \rangle = \sum_a Pr(x_i = a) = 1$$

This implies $\sum_a v_{i,a} = v_0$.

This optimization problem can be converted by standard techniques to a SDP of polynomial size, and its optimal solution can be obtained in polynomial time.

## 3. **Rounding by Variable Folding**

We will present a rounding scheme for this SDP which is provably close to optimal.

To define the sense in which it is close to optimal, we consider the integrality gap curve $S_\Lambda$ for this family of CSPs:

$$S_\Lambda(c) = \inf_{sdp(\mathcal{P})=c} opt(\mathcal{P})$$

Here $sdp(\mathcal{P})$ denotes the optimal value of the SDP relaxation of $\mathcal{P}$ and $opt(\mathcal{P})$ denotes the optimal value of the original instance. (Note that we divide this optimum by the number of payoff functions in the instance)

The rounding scheme we present satisfies the following theorem:

**Theorem 3.1.** *For every $\epsilon > 0$ there exists an algorithm which, given a $\Lambda$-CSP instance $\mathcal{P}$ and an optimal solution $P$ to the SDP relaxation of $\mathcal{P}$, returns a solution $P$ of $\mathcal{P}$ with value at least $S_\Lambda(sdp(\mathcal{P}) - \epsilon) - \epsilon$. The algorithm runs in time $poly(n) + exp(exp(poly(\frac{kq}{\epsilon})))$*

The rounding scheme will use variable folding. Given a CSP instance $\mathcal{P}$with variables $V$, we can construct a folding of $V$ by constructing a mapping $\phi : V \to W$ for some set $W$. We obtain a 'folded' CSP in which we identify variables $x_i$, $x_j$ if $\phi(x_i) = \phi(x_j)$. We refer to this folded instance as $\mathcal{P}/\phi$. We note two properties of folding: the number of variables of $\mathcal{P}/\phi$ is less than $|W|$, and $opt(\mathcal{P}/\phi) \leq opt(\mathcal{P})$.

Our overall strategy for rounding will be as follows: We will compute an optimal solution to the SDP relaxation of $\mathcal{P}$. We will use this solution to, first, discard some of the constraints of the original problem, then find a variable folding of this altered problem which approximately preserves the SDP value. Solving this CSP by brute force gives us a solution which is approximately an optimal rounding for our original CSP. In the next section we will show how to construct a variable folding with the following properties:

**Theorem 3.2.** *Given a $\Lambda$-CSP instance $\mathcal{P}$, we can efficiently compute an instance $\mathcal{P}$' and a variable folding $\phi$ such that*

*(1) $\mathcal{P}$' is obtained from $\mathcal{P}$ by discarding a fraction $\epsilon$ of payoffs*
*(2) $sdp(\mathcal{P}'/\phi) \geq sdp(\mathcal{P}) - \epsilon$*
*(3) The variable set of $\mathcal{P}'/\phi$ has cardinality $exp(\frac{kq}{\epsilon})$*

The above theorem implies Theorem 3.1.

*Proof.* By property 3, we can solve the CSP $\mathcal{P}'/\phi$ by brute force in time $exp(exp(\frac{kq}{\epsilon}))$. Let $y$ be the optimal solution we obtain. The value of $y$ on $\mathcal{P}'/\phi$, $val_{\mathcal{P}'\phi}(y)$, is greater than $S_\Lambda(sdp(\mathcal{P}) - \epsilon)$ by property 2 and the definition of $S_\Lambda$. By the definition of variable folding $val_{\mathcal{P}'\phi}(y) = val_{\mathcal{P}'}(y)$. Finally by property 1, $|val_{\mathcal{P}}(y) - val_{\mathcal{P}'}(y)| \leq \epsilon$. Therefore $val_{\mathcal{P}}(y) \geq S_\Lambda(sdp(\mathcal{P}) - \epsilon) - \epsilon$, as desired. □

## 4. Construction of the Variable Folding

In the previous section we see that it is enough to create a variable folding that satisfies the conditions of Theorem 3.2. We are given an instance $\mathcal{P}$ over variable set $V$ and the solution $\{v_{i,a}\}, \{\mu_P\}$ of the SDP relaxation. We want to construct an instance $\mathcal{P}'$ and a function $\phi : V \to W$, such that $|W|$ is constant and $\mathcal{P}'$ is close to $\mathcal{P}$. The following three steps gives an algorithm that returns $\mathcal{P}'$ and $\phi$.

(1) **Dimension reduction:** The vectors $\{v_{i,a}\}$ have a really high dimension, say $D$. We create a vector $u_{i,a}$ corresponding to each vector $v_{i,a}$ of constant dimension, say $d$. Let $M$ be a matrix of dimension $d \times D$ such that each entry of $M$ is independent and drawn from a Gaussian distribution, $\mathcal{N}(0, 1/d)$. Now, define:

$$u_{i,a} = Mv_{i,a}$$

(2) **Discarding bad CSP constraints:** The new vectors, $\{u_{i,a}\}$ and distributions $\{\mu_P\}$ may not satisfy the SDP constraints. For every payoff function $P$ we have around $\mathcal{O}(k^2q^2)$ constraints in the SDP. Let $B_\varepsilon$ denote the set of payoff functions $P$ for which at least one SDP constraint is not satisfied up to an additive error of $\varepsilon$. We now define a new instance $\mathcal{P}'$ which does not contain the payoff functions in $B_\varepsilon$, i.e., $\mathcal{P}' = \mathcal{P} \setminus B_\varepsilon$.

(3) **Discretization:** Every vector $\{u_{i,a}\}$ has a norm at most $1 + \varepsilon$ (follows from the SDP constraints). Let us look at the unit ball in $d$ dimensions and the position vectors of $\{u_{i,a}\}$ in it. Let $N$ denote a grid in this $d$ dimensional space where each grid hyper-square has length $\varepsilon$. Let $|N|$ denote the number of grid points in $N$. Note that $|N| \leq \big(c/\varepsilon\big)^d$ for some constant $c$. For every $u_{i,a}$, let $w_{i,a}$ denote the closest point (standard euclidean distance) on the grid to $u_{i,a}$. Formally our function $\phi : V \to N^q$ is:

$$\phi(i) = (w_{i,1}, w_{i,2}, ..., w_{i,q})$$

Return $\mathcal{P}'/\phi$ where $\mathcal{P}'$ is as described in (2) and $\phi$ is as described in (3). We need to prove that $\mathcal{P}'$ and $\phi$ satisfy 3.2. We first prove that not too many payoff functions are discarded in step 2.

**Lemma 4.1.** *For any two vectors $v_1$ and $v_2$ in $\mathbb{R}^D$ in the unit ball*

$$\Pr_M \big[|\langle Mv_1, Mv_2\rangle - \langle v_1, v_2\rangle| \geq \varepsilon\big] \leq \mathcal{O}(\frac{1}{\varepsilon^2 d})$$

*Proof.*

$$\begin{aligned}
\mathbf{E}[\langle Mv_1, Mv_2\rangle] &= \mathbf{E}[\sum_{i=1}^{d}\sum_j\sum_k M_{ij}M_{ik}v_{1j}v_{2k}] \\
&= \mathbf{E}[\sum_{i=1}^{d}(\sum_j M_{ij}^2 v_{1j}v_{2j} + \sum_{j\neq k} M_{ij}M_{ik}v_{1j}v_{2k})] \\
&= \sum_j \langle v_{1j}, v_{2j}\rangle \sum_{i=1}^{d}\mathbf{E}[M_{ij}^2] + \sum_{i=1}^{d}\sum_{j\neq k}\mathbf{E}[M_{ij}]\,\mathbf{E}[M_{ik}]v_{1j}v_{2k} \\
&= \sum_j \langle v_{1j}, v_{2j}\rangle \cdot d \cdot 1/d + 0 \\
&= \langle v_1, v_2\rangle
\end{aligned}$$

Also, standard deviation of $\langle Mv_1, Mv_2\rangle$ can be shown to be $\mathcal{O}(1/\sqrt{d})$. We now use Chebyshev's inequality which proves the lemma. □

**Lemma 4.2.** *With high probability* $||\mathcal{P} - \mathcal{P}'||_1 \leq \varepsilon$

*Proof.* While creating the instance $\mathcal{P}'$, we discard all payoff functions $P$ from $\mathcal{P}$ for which even one SDP constraint is not satisfied within an additive error of $\varepsilon$. For every $P \in PP$, we have $\mathcal{O}(k^2q^2)$ SDP constraints. Probability that a particular constraint fails is $\mathcal{O}(\frac{1}{\varepsilon^2 d})$ from lemma 4.1. So the probability that one of the SDP constraints corresponding to $P$ fails is $\mathcal{O}(\frac{k^2q^2}{\varepsilon^2 d})$. We thus have,

$$\Pr_M[P \in B_\varepsilon] \leq \mathcal{O}(\frac{k^2q^2}{\varepsilon^2 d})$$

Setting $d = poly(kq/\varepsilon)$, we get, $\mathbf{Pr}_M[P \in B_\varepsilon] \leq \varepsilon$. This means we have $||\mathcal{P} - \mathcal{P}'||_1 \leq \varepsilon\mathcal{P} \leq \varepsilon$, since $\mathcal{P} \leq 1$. □

The above lemma shows that we haven't discarded too many pay off functions. Also the values attained by the instances $\mathcal{P}$and $\mathcal{P}'$ on any assignment $x$ differ by at most $\varepsilon$. We next show that the instance we return, $\mathcal{P}'/\phi$ has constant number of variables.

**Corollary 4.3.** *The variable set of* $\mathcal{P}'/\phi$ *has cardinality at most* $2^{poly(kq/\varepsilon)}$

*Proof.* The number of variables in the instance $\mathcal{P}'/\phi$ is at most $|N|^q \leq \big(c/\varepsilon\big)^d$. Since we have $d = poly(kq/\varepsilon)$, we get, $|N|^q \leq \big(c/\varepsilon\big)^{poly(kq/\varepsilon)} = 2^{poly(kq/\varepsilon)}$. □

Now, we know that our solution $\{u_{i,a}\}, \{\mu_P\}$ satisfies every SDP constraint of the instance $\mathcal{P}'$ up to an additive error of $\varepsilon$. We still need to know how the solution $\{w_{i,a}\}, \{\mu_P\}$ behaves with respect to the SDP corresponding to $\mathcal{P}'$.

**Lemma 4.4.** *For small enough* $\varepsilon > 0$*, suppose vectors* $\{u_{i,a}\}$ *satisfy all constraints corresponding to some payoff function* $P \in \mathcal{P}'$ *up to an error* $\varepsilon$*. Then the vectors* $\{w_{i,a}\}$ *satisfy all constraints corresponding to* $P$ *up to an error of* $4\varepsilon$*.*

*Proof.* Every vector $w_{i,a} \leq u_{i,a} + \varepsilon\hat{v}$ where $\hat{v}$ is some unit vector. From the SDP constraints we know that $|u_{i,a}|^2 \leq 1 + \varepsilon$.

$$\begin{aligned}
|\langle w_{i,a}, w_{j,b}\rangle - \langle u_{i,a}, u_{j,b}\rangle| &\leq |\langle u_{i,a} + \varepsilon\hat{v_1}, u_{j,b} + \varepsilon\hat{v_2}\rangle - \langle u_{i,a}, u_{j,b}\rangle| \\
&= \varepsilon|u_{i,a} + u_{j,b}| + \varepsilon^2 \\
&\leq \varepsilon(|u_{i,a}| + |u_{j,b}|) + \varepsilon \\
&\leq 2\varepsilon\sqrt{1+\varepsilon} + \varepsilon \\
&\leq (2\sqrt{2}+1)\varepsilon \leq 4\varepsilon
\end{aligned}$$

□

We are only left with proving (2) of theorem 3.2 which follows from the following theorem (for which we sketch out the proof later).

**Theorem 4.5.** *Let $\mathcal{P}$ be a $\Lambda$-CSP instance on variable set $V$. Suppose $\{v_{i,a}\}, \{\mu_P\}$ satisfy the SDP constraints for $\mathcal{P}$ up to an additive constant $\varepsilon$ and have a SDP value $\alpha$, then*

$$sdp(\mathcal{P}) \geq \alpha - \sqrt{\varepsilon} \cdot poly(kq)$$

**Corollary 4.6.**

$$sdp(\mathcal{P}'/\phi) \geq sdp(\mathcal{P}) - \varepsilon$$

*Proof.* Solution $\{w_{i,a}\}, \{\mu_P\}$ satisfies every SDP constraint for the instance $\mathcal{P}'$ up to an additive error of $4\varepsilon$. The value of the SDP solution for $\mathcal{P}'$ is at least $sdp(\mathcal{P}) - ||\mathcal{P} - \mathcal{P}'|| \geq sdp(\mathcal{P}) - \varepsilon$. Note that this is also a solution for the instance $\mathcal{P}'/\phi$ with the same SDP value. Now using Theorem 4.5, we get that $sdp(\mathcal{P}'/\phi) \geq sdp(\mathcal{P}) - \varepsilon - \sqrt{\varepsilon} \cdot poly(kq) \geq sdp(\mathcal{P}) - \sqrt{\varepsilon} \cdot poly(kq)$. □

**Proof of Theorem 4.5**

*Proof.* We need to show that there exists at least one feasible solution to the SDP for $\mathcal{P}$, that is not far from the SDP value of $\mathcal{P}$. Starting with the solution $\{v_{i,a}\}, \{\mu_P\}$, we construct a feasible solution $\{w_{i,a}\}, \{\mu'_P\}$. Notice the SDP constraints. There are two types of them,

$$\langle v_{i,a}, v_{i,b}\rangle = \mathbf{Pr}[x_i = a, x_i = b] = 0$$

$$\langle \sum_a v_{i,a}, v_0\rangle = 1$$

So we first create a new solution $\{u_{i,a}\}$ which at least satisfies these constraints. This new solution is also infeasible but by a different amount. The following lemma does it.

**Lemma 4.7.** *The vectors $\{v_{i,a}\}$ can be transformed to vectors $\{u_{i,a}\}$ such that for all $a, b \in [q]$ and all $i \in V$,*

$$\langle u_{i,a}, u_{i,b}\rangle = 0$$

,

$$\sum_a u_{i,a} = v_0$$

*Furthermore, for $i \in V$ and $a \in [q]$,*

$$||u_{i,a} - v_{i,a}|| \leq \sqrt{\varepsilon} \cdot poly(q)$$

*In particular, the SDP solution $\{u_{i,a}\}, \{\mu_P\}$ is $\eta$-infeasible for $\eta = \sqrt{\varepsilon} \cdot poly(q)$*

Next we modify the local distributions by a small amount to satisfy a certain property given in the lemma below.

**Lemma 4.8.** *The local distributions $\{\mu_P\}$ can be transformed to distributions $\{\mu_P\}$ such that for all $P$ , $i \neq j \in V(P)$, and $a, b \in [q]$,*

$$\mathop{\mathbf{Pr}}_{x\sim\mu'}[x_i = a, x_j = b] = (1-\delta)\langle u_{i,a}, u_{j,b}\rangle + \delta \cdot \frac{1}{q^2}$$

*where $\delta = q^4k^2\eta$. Furthermore, for every $P$,*

$$||\mu_P - \mu'_P|| \leq 3\delta$$

The proof of the theorem follows from the above two lemmas. Construct $\{u_{i,a}\}$ as in lemma 4.7 and $\{\mu'_P\}$ as in lemma 4.8. Define new vectors,

$$w_{i,a} = \sqrt{1-\delta}u_{i,a} \oplus \sqrt{\delta}u'_{i,a}$$

Here $\{u'_{i,a}\}$ corresponds to the centre of the convex hull of the feasible solutions. Basically we are taking a suitable convex combination of the centre of the convex hull with vectors $u'_{i,a}s$ to bring it inside the convex hull. The vectors $\{w_{i,a}\}$ is this required convex combination. It can be seen that these vectors with $\{\mu'_P\}$ form a feasible solution. We now need to estimate the SDP value of this solution.

$$\begin{aligned}
\mathop{\mathbf{E}}_{P\sim\mathcal{P}} \mathop{\mathbf{E}}_{x\sim\mu'_P} P(x) &= \alpha - \mathop{\mathbf{E}}_{P\sim\mathcal{P}} P(x)(\mu(x) - \mu'(x)) \\
&\geq \alpha - \mathop{\mathbf{E}}_{P\sim\mathcal{P}} ||(\mu(x) - \mu'(x)||_1, \text{ since } P(x) \leq 1 \\
&\geq \alpha - \eta \cdot poly(kq), \text{follows from lemma 4.8 and putting in the value of } \delta
\end{aligned}$$

This completes the proof of the theorem. □

## 5. References


(1) Prasad Raghavendra, David Steurer. How to Round any CSP. FOCS '09 Proceedings of the 2009 50th Annual IEEE Symposium on Foundations of Computer Science. Pages 586-594

(2) Bernd Grtner, Jiri Matousek. Approximation Algorithms and Semidefinite Programming.