Rough Notes on Bansal's Algorithm

Joel Spencer

A quarter century ago I proved that given any $S_1, \ldots, S_n \subseteq \{1, \ldots, n\}$ there was a coloring $\chi : \{1, \ldots, n\} \rightarrow \{-1, +1\}$ so that $\texttt{disc}(S_j) \leq 6\sqrt{n}$ for all $1 \leq j \leq n$ where we define

$$\chi(S) = \sum_{i \in S} \chi(i) \tag{1}$$

and $\texttt{disc}(S) = |\chi(S)|$. Here 6 is an absolute constant whose precise value will not here be relevant. The question remained: Is there a polynomial time algorithm to find a coloring χ with this property. I frequently conjectured that there was no such algorithm. But now Nikhil Bansal has found one! His result (which has much more) can be found on ArXiv at

http://arxiv.org/abs/1002.2259

Here I give a rough description, but emphasizing that the result is entirely his.

1 Semidefinite Programming

At its core, Bansal's algorithm uses **Semidefinite Programming**. Here is the context:

Let $S_1, \ldots, S_m \subseteq \{1, \ldots, n\}$. *Assume* that there *exists* a map

$$\chi : \{1, \ldots, n\} \rightarrow \{-1, 0, +1\}$$

(such χ are called partial colorings and i with $\chi(i) = 0$ are called uncolored) such that

1. At least $n\alpha$ of the i have $\chi(i) \neq 0$.
2. For $1 \leq j \leq m$
$$\texttt{disc}(S_j) \leq \beta_j \sqrt{n}$$

Consider the Semidefinite Program to find $\vec{v_1}, \ldots, \vec{v_n} \in R^n$ satisfying (so this is a *feasibility* program):

1. $\vec{v_i} \cdot \vec{v_i} \geq 1$ for $1 \leq i \leq n$.
2. $\sum_{i=1}^{n} \vec{v_i} \cdot \vec{v_i} \geq n\alpha$

3. For $1 \le j \le m$

$$[\sum_{i\in S_j} \vec{v_i}] \cdot [\sum_{i\in S_j} \vec{v_i}] \le [\beta_j\sqrt{n}]^2$$

The Semidefinite Program has a solution in R^1 by setting $\vec{v_i} = \chi(i)$. Therefore, Semidefinite Programming will *find* a solution (though probably not that one!) in time polynomial in n, m. Critically, we do *not* need to know the original χ. (Well, I'm lying a bit here. You need an ϵ of space – that the conditions hold with $n\alpha$ and $\beta_j\sqrt{n}$ replaced by $n\alpha(1+\epsilon)$ and $\beta_j\sqrt{n}(1-\epsilon)$ respectively for some fixed ϵ. This will always clearly hold.)

2 My Result

I need to state my result in more general form. For completeness I give the argument, far better than my original proof, due to Ravi Boppana. Let

$$H(x) = -x\log_2 x - (1-x)\log_2(1-x) \tag{2}$$

be the usual entropy function. For $\beta > 0$ define

$$\text{COST}(\beta) = \sum_{i=-\infty}^{+\infty} -p_i \log_2 p_i \tag{3}$$

with

$$p_i = \Pr[i - \frac{1}{2} \le \frac{N}{\beta} \le i + \frac{1}{2}],\ N \text{ standard normal}$$

That is, $\text{COST}(\beta)$ is the entropy of the roundoff of N to the nearest multiple of β.

Theorem 2.1 *Let $S_1, \ldots, S_m \subseteq \{1, \ldots, n\}$. Let $\beta_1, \ldots, \beta_m$ be such that*

$$\sum_{j=1}^{m} COST(\beta_j) \le cn \tag{4}$$

where $c < 1$. Write $c = 1 - H(\alpha)$ with $\alpha \in (0, \frac{1}{2})$. Then there is a $\chi : \{1, \ldots, n\} \to \{-1, 0, 1\}$ with

1. *At least $2\alpha n$ of the i have $\chi(i) \ne 0$.*
2. *For $1 \le j \le m$*

$$\texttt{disc}(S_j) \le \beta_j\sqrt{n}$$

For completeness, here is Boppana's argument. For any $\chi : \{1, \ldots, n\} \to \{-1, +1\}$ set

$$\Lambda(\chi) = (b_1, \ldots, b_m) \tag{5}$$

where b_j is the roundoff of $\chi(S_j)$ to the nearest multiple of $\beta_j\sqrt{n}$. For $\chi : \{1, \ldots, n\} \to \{-1, +1\}$ uniform random $\chi(S_j)$ is the sum of $|S_j|$ random ± 1 which is basically (minor hand waving here) normal with mean zero and variance $|S_j| \le n$ so that b_j (fixed j, χ random) will have entropy at most $\text{COST}(\beta_j)$. By subadditivity $\Lambda(\chi)$ has entropy at most $\sum \text{COST}(\beta_j) =$

$(1 - H(\alpha))n$. Thus some value of $\Lambda(\chi)$ occurs with probability at least $2^{-(1-H(\alpha))n}$, that is, for at least $2^{H(\alpha)n}$ different χ. A theorem of Dan Kleitman (weaker methods suffice if the best constants are not needed) says that the subset of the Hamming Cube of a given size with minimal diameter is basically a ball – so that the diameter is at least $2n\alpha$. Take χ_1, χ_2 with $\Lambda(\chi_1) = \Lambda(\chi_2)$ that differ in at least $2n\alpha$ places. Finally, set

$$\chi(i) = \frac{\chi_1(i) - \chi_2(i)}{2} \tag{6}$$

Then $\chi(i) \in \{-1, 0, +1\}$ tautologically and it is nonzero for at least $2n\alpha$ values i. Critically, for every j, $\chi(S_j) = [\chi_1(S_j) - \chi_2(S_j)]/2$ and as $\chi_1(S_j), \chi_2(S_j)$ have the same roundoff to the nearest multiple of $\beta_j\sqrt{n}$, $|\chi(S_j)| \leq \beta_j\sqrt{n}$ as desired.

`Remark:` The jump from the "small" entropy to $2^{H(\alpha)n}$ different χ with the same $\Lambda(\chi)$ is basically the pigeonhole principle and led me to the incorrect conjecture that this argument could not be made algorithmic.

For our purposes (as the constants do not concern us here) we need only rough upper bounds on the function $\mathrm{COST}(\beta)$. We note that

$$\mathrm{COST}(\beta) = \Theta(\ln \beta^{-1}) \text{ as } \beta \to 0^+ \tag{7}$$

as, basically, usually $|N| \leq 10$ and so the roundoff of $N\beta$ usually takes on one of $20\beta^{-1}$ values. Further

$$\mathrm{COST}(\beta) = \Theta(\beta e^{-\beta^2/8}) \text{ as } \beta \to +\infty \tag{8}$$

as, basically, we consider either $|N| \leq \beta/2$ or not, and the not occurs with probability $\Theta(\beta^{-1}e^{-\beta^2/8})$. These are far finer than we need, which is only that $\lim_{\beta\to+\infty} \mathrm{COST}(\beta) = 0$ and that $\mathrm{COST}(\beta)$ has very slow growth as $\beta \to 0^+$. In application we like to think of $\mathrm{COST}(\beta_j)$ as the cost of the j-th discrepency condition and (4) as our `cost equation`. The total cost is basically at most n. This allows us to have a small number of quite tight (that is, β_j small) discrepency conditions.

3 Floating Colors

During the algorithm there for each vertex $1 \leq i \leq n$ there will be a value $x_i \in [-1, +1]$ which will move around. Initially all $x_i = 0$ (though the method works for any initial x_i) and at the end all $x_i \in \{-1, +1\}$. We call i *floating* if

$$|x_i| < 1 - \lg^{-1} n$$

Otherwise we call it *frozen*, and we call it *final* when $x_i \in \{-1, +1\}$. We *never* have $|x_i| > 1$. Once x_i is fixed it does not move until the very end, when we have the (relatively easy) `Final Roundoff` to the final values.

Suppose that during some period of the algorithm the values move from x_i to some x_i'. We let

$$\Delta(S_j) = \sum_{i \in S_j} x_i' - x_i$$

denote the change in $\chi(S_j)$ during that period.

The algorithm splits into `Phases`. We will number them Phase t for $t = 0, 1, \ldots$. Phase t will begin with $x_1, \ldots, x_n$ with at most $n2^{-t}$ floating. It will end with $x'_1, \ldots, x'_n$ with at most $n2^{-t-1}$ floating. (If it happens to start with fewer than $n2^{-t-1}$ floating it does nothing.) Phase 0 will have

$$|\Delta(S_j)| \leq K\sqrt{n}, 1 \leq j \leq n \tag{9}$$

and Phase $t \geq 1$ will have

$$|\Delta(S_j)| \leq K\sqrt{t}\sqrt{n2^{-t}}, 1 \leq j \leq n \tag{10}$$

We'll end when the number of floating variables goes below $n \ln^{-1} n$, and then do the Final Roundup. I'll emphasize Phase 0 here and make a few remarks about the general case.

4 The Easy Final Roundup

Suppose that the phases have worked. From (9,10) the total absolute value change in each $\chi(S_j)$ is at most the sum of the absolute values of the changes and as $\sum_{t=1}^{\infty} \sqrt{t2^{-t}}$ converges this is all $K_1\sqrt{n}$. Now we have at most $c\frac{n}{\ln n}$ floating x_i and the rest frozen. For the Final Roundup for each i we replace x_i with $+1$ with probability $\frac{1+x_i}{2}$ and with -1 with probability $\frac{1-x_i}{2}$. This keeps the expectation the same. Each $\mathtt{disc}(S_j)$ is changed by $\sum_{i \in S_i} Y_i$ where $\Pr[Y_i = 1 - x_i] = \frac{1+x_i}{2}$ and $\Pr[Y_i = -1 - x_i] = \frac{1-x_i}{2}$. Here Y_i has mean zero and variance $1 - x_i^2$. This is at most 1 for the at most $cn \ln^{-1} n$ floating i and at most $c \ln^{-1} n$ for the at most n frozen i so the total variance is at most $cn \ln^{-1} n$. Basic Chernoff bounds now give that the probability that $\mathtt{disc}(S_j)$ is changed by more than $K_2\sqrt{n}$ is at most $n^{-c'}$. Adjusting the constants we make this $o(n^{-1})$. So the Final Roundoff also changes each $\mathtt{disc}(S_j)$ by $O(\sqrt{n})$ and, putting everything together, the total change overall in each $\mathtt{disc}(S_j)$ is at most $K_3\sqrt{n}$ as desired.

5 Phase 0

We start with floating $x_1, \ldots, x_n$ Actually they are all zero but we needn't assume that and indeed its nicer not too as this isn't the case with later phases. We proceed is *steps*. At each step x_i will move a small (more on that soon) amount. When an x_i becomes frozen it no longer moves. When the number of floating variables reaches $n/2$ we terminate Phase 0. We are successful if (9) holds for all S_j. Our procedure may abort, we shall only show that it succeeds with probability bounded away from zero. Then if it aborts we simply repeat it and get a success with an expected bounded number of attempts.

At the start of any step let FL denote the set of floating variables. Let $\Delta(S_j)$ refer to the absolute value of the change in $\chi(S_j)$ up to this point. We call j *dangerous* if $\Delta(S_j) \geq \frac{K}{2}\sqrt{n}$ where our goal (9) is that $\Delta(S_j) \leq K\sqrt{n}$ at the end of Phase 0. More generally (a rather annoying technical part) we'll call j Level l dangerous if $\Delta(S_j) \geq K(1-2^{-l})\sqrt{n}$. For technical reasons, we'll call the level of danger for j the highest Level l dangerous that has been reached during Phase 0 up to this point. Let $DANG(l)$ denote those j at level of danger l and $SAFE$ denote the other j. We'll want to be more careful with the dangerous j by tightening the conditions. While an infinite number of danger levels is technically necessary it somewhat obscures the main line of the argument and the reader can profitably think of a small proportion of the S_j becoming dangerous and having tighter conditions on them.

At the start of the step we solve the Semidefinite Program

1. $\vec{v_i} \cdot \vec{v_i} \geq 1$ for $i \in FL$.
2. $\sum_{i=1}^{n} \vec{v_i} \cdot \vec{v_i} \geq n/4$
3. For $j \in SAFE$
$$[\sum_{i \in S_j} \vec{v_i}] \cdot [\sum_{i \in S_j} \vec{v_i}] \leq [K_1\sqrt{n}]^2 \tag{11}$$
4. For $j \in DANG(l)$
$$[\sum_{i \in S_j} \vec{v_i}] \cdot [\sum_{i \in S_j} \vec{v_i}] \leq [K_1\kappa_l\sqrt{n}]^2 \tag{12}$$

We shall choose K_1 large and then K much larger by (24), both absolute constants. Here the κ_l will be chosen by (25) to go to zero at an appropriate rate to be discussed later. It may be that the Semidefinite Program above does not find a solution. In this case we say that we *abort* the attempt at Phase 1. Since Semidefinite Programming (allowing some ϵ leeway) will find some solution if it exists this will only occur if the Semidefinite Program does not have a solution. More on this later.

Central Step: We let $g_1, \ldots, g_n$ be i.i.d. standard normals and set $\vec{G} = (g_1, \ldots, g_n)$. We set
$$\delta_i = \epsilon \vec{v_i} \cdot \vec{G} \tag{13}$$
with ϵ chosen below (15). We update:
$$x_i \leftarrow x_i + \delta_i \tag{14}$$
for $1 \leq i \leq n$.

As $\vec{G}$ is distributed symmetrically, δ_i will be normally distributed with mean zero and standard deviation $\epsilon|\vec{v_i}| \leq \epsilon$ We set
$$\epsilon = c\ln^{-3/2} n \tag{15}$$
Then the probability that $|\delta_i| \geq \ln^{-1} n$ is polynomially small and it will not occur throughout the procedure. Therefore the x_i will not jump over the values -1 or $+1$. (The algorithm would work for smaller ϵ but smaller ϵ would require more steps as indicated below.)

Claim 5.0.1 *The probability that this process goes S steps (or more) is at most $4\epsilon^{-2}S^{-1}$.*

Proof: Define W_t to be the value of $\sum_{i=1}^n x_i^2$ if the process is still going. Letting $x_1, \ldots, x_n$ be the values after $t-1$ steps,

$$E[W_t - W_{t-1}] = \sum_{i=1}^{n} E[(x_i + \delta_i)^2 - x_i^2] \tag{16}$$

But as δ_i is normal with mean zero and variance $\epsilon^2 |\vec{v_i}|^2$ we get

$$E[W_t - W_{t-1}] = \sum_{i=1}^{n} E[\delta_i^2] = \sum_{i=1}^{n} \epsilon^2 |\vec{v_i}|^2 \geq \frac{n}{4}\epsilon^2 \tag{17}$$

If the process has already stopped we simply set $W_t = W_{t-1} + \frac{n}{4}\epsilon^2$. With this technical trick, $W_0, \ldots, W_S$ are defined and

$$E[W_S] \geq S\epsilon^2 \frac{n}{4} \tag{18}$$

If the process does stop then at the time t that it does start $W_t \leq n$ tautologically and hence the final

$$W_S \leq n + (S-t)\epsilon^2 \frac{n}{4} \leq n + S\epsilon^2 \frac{n}{4} \tag{19}$$

If the process has not stopped by time S then all $|x_i| \leq 1$ and

$$W_S \leq n \tag{20}$$

Let p be the probability that the process has not stopped by time S. Then

$$S\epsilon^2 \frac{n}{4} \leq E[W_S] \leq pn + (1-p)[n + S\epsilon^2 \frac{n}{4}] \tag{21}$$

from which the claim follows.

Now we simply set

$$S = 40\epsilon^{-2} \tag{22}$$

so that with 90% chance the process has stopped in S stops. If the process has not stopped it is considered a failure. Henceforth we consider the process continuing for at most S steps. We now need "only" show that, with the appropriate values of κ_l, the Semidefinite Programs will all have solutions with probability, say, 50%.

Consider a single S_j and the movement of $\texttt{disc}(S_j)$ throughout Phase 0. At each step this is changed by the sum of the changes of the x_i, $i \in S_j$, which is precisely

$$\epsilon \vec{G} \cdot [\sum_{i \in S_j} \vec{v_i}] \tag{23}$$

The change is normally distributed with mean zero and standard deviation ϵ times the norm of $\sum_{i \in S_j} \vec{v_i}$. From (11,12) we have:

1. While S_j is safe the change in $\texttt{disc}(S_j)$ is normally distributed with mean zero and standard deviation at most $K_1 \epsilon \sqrt{n}$.

2. While S_j is at danger level l the change in $\texttt{disc}(S_j)$ is normally distributed with mean zero and standard deviation at most $K_1 \kappa_l \epsilon \sqrt{n}$.

Now we can bound the probability that S_j becomes dangerous. There are at most $S = 40\epsilon^{-2}$ steps and at each step the change is normal with mean zero and standard deviation at most $K_1\epsilon\sqrt{n}$. The values of $\texttt{disc}(S_j)$ become a martingale. The ϵ scales out and the probability of reaching $\frac{K}{2}\sqrt{n}$ is at most $2\exp[-K^2/(160K_1^2)]$. We'll pick K so large, for definiteness we will set

$$K = 20K_1 \tag{24}$$

so that this probability is quite small, less than $2\exp[-2K_1^2]$.

Lets look at the probability that S_j becomes $l+1$-dangerous. Once it is l-dangerous each step is normal with mean zero and standard deviation at most $K_1\kappa_l\epsilon\sqrt{n}$. It has to move at least $K2^{-l-1}\sqrt{n}$. Even it has all S steps (so we are giving some ground here, indeed, ignoring the probability that it becomes l-dangerous) the probability is at most $2\exp[-K^2 2^{-2l-2}\kappa_l^{-2}/160K_1^2]$. For definiteness, we set

$$\kappa_l = 4^{-l} \tag{25}$$

so that this probability goes rapidly to zero, less than $2\exp[-2K_1^2 2^{2l-2}]$.

We *cannot* assume that the walks for the different S_j are independent. If, for example, $S_j = S_{j'}$ then the two walks would be precisely the same. Nonetheless, the above probabilities are so small that we can effectively use Markov's Inequality. The probability that the number of j for which S_j becomes $l+1$-dangerous is greater than $2^{l+3}\exp[-2K_1^2 2^{2l-2}]$. is at most 2^{-l-2}. Adding over $l \geq 0$, with probability at least 50% this does not occur for any $l \geq 0$.

This combines with our previous at least 90% chance that the process does not go to S steps. Thus with a positive, here 40%, chance, Phase I is successful.

Or is it? We must check that the Semidefinite Program will be feasible. To do that we must check that the cost equation (4) is satisfied with $c = 1 - H(1/8)$. Here there are at most n conditions of cost $\mathrm{COST}[K_1]$ and for each $l \geq 0$ at most $n2^{l+3}\exp[-2K_1^2 2^{2l-2}]$ conditions of cost $\mathrm{COST}[K_1 4^{1-l}]$. That is, the cost equation becomes

$$n\mathrm{COST}[K_1] + \sum_{l\geq 0} n2^{l+3}\exp[-2K_1^2 2^{2l-2}]\mathrm{COST}[K_1 4^{1-l}] \leq cn \tag{26}$$

The factors of n scale out and this becomes

$$\mathrm{COST}[K_1] + \sum_{l\geq 0} 2^{l+3}\exp[-2K_1^2 2^{2l-2}]\mathrm{COST}[K_1 4^{1-l}] \leq c \tag{27}$$

We could easily give an explicit K_1 here but let us instead show that (replacing K_1 by x)

$$\lim_{x\to\infty} \mathrm{COST}[x] + \sum_{l\geq 0} 2^{l+3}\exp[-2x^2 2^{2l-2}]\mathrm{COST}[x4^{1-l}] = 0 \tag{28}$$

Clearly $\lim_{x\to\infty}\mathrm{COST}[x] = 0$. The only difficulty is that for any fixed x, $\mathrm{COST}[x4^{1-l}]$ is unbounded. But its growth rate is slow. With $x \geq 4$ we have from (7)

$$\text{COST}[x4^{1-l}] \leq \text{COST}[4^{-l}] \leq c_1(l+1) \tag{29}$$

for some constant c_1, uniformly over $l \geq 0$. Thus it suffices that

$$\lim_{x\to\infty} \sum_{l\geq 0} (l+1)2^{l+3} \exp[-2x^2 2^{2l-2}] = 0 \tag{30}$$

which is clearly true. That is, there is a constant K_1 with (26) holding. For this K_1, Phase 1 has at least a 40% chance of succeeding.

`Remark:` A worry throughout is that we start with n sets and so some of them will get into deep trouble. The cost equation (4) allows some equations to be in trouble. But the cost of being level l dangerous only goes up linearly in l and the proportion of sets reaching that level drops hypergeometrically in l and so the cost equation remains satisfied.

6 A few words on later Phases

If we had $n2^{-t}$ floating variables and $n2^{-t}$ sets we would only have a change of parameter and our algorithm for Phase t would give all $\texttt{disc}(S_j) \leq K_1(n2^{-t})^{1/2}$. The technical issue is that we still have n sets and this leads to an additional $\sqrt{t}$ term. But there is plenty of room, we could even replace the $\sqrt{t}$ by, say, $2^{t/6}$, finding the weaker $\texttt{disc}(S_j) \leq K_1\sqrt{n}2^{-t/3}$ but still getting the final discrepancy as $O(\sqrt{n})$.

In the safe region with n sets and $n2^{-t}$ floating vertices we replace condition (11) with

$$[\sum_{i\in S_j} \vec{v_i}] \cdot [\sum_{i\in S_j} \vec{v_i}] \leq [K_1\sqrt{t}\sqrt{n2^{-t}}]^2 \tag{31}$$

Now in the cost equation (4) has $m = n$ sets and all $\beta_i = K_1\sqrt{t}$. From (8) $\text{COST}(K_1\sqrt{t}) = \exp[-\Theta(tK_1^2)]$. For K_1 appropriately large this is much smaller than 2^{-t} and so

$$\sum_{i=1}^{n} \text{COST}(\beta_i) \leq cn \tag{32}$$

For the full argument we must again consider S_j to be dangerous at level l when $\texttt{disc}(S_j)$ reaches $K(1-2^{-l})\sqrt{t}\sqrt{n2^{-t}}$ at which point the bound on $|\vec{v_j}|$ is tightened by a factor of κ_l. The details are very similar to the Phase 0 case and I omit them.