Rounding Techniques in Approximation Algorithms

Lecture 6: Randomized Pipage Rounding and k-ECSM

Lecturer: Nathan Klein

1 Randomized Pipage Rounding

1.1 Recap

In the last lecture, we discussed the *randomized pipage rounding* algorithm [AS04; CVZ10]. In particular, we (i) proved that the following is the convex hull of all integral spanning trees of a graph:

$$P_{\text{st}} = \begin{cases} x(E) = |V| - 1 & \\ x(E(S)) \leq |S| - 1 & \forall S \subseteq V \\ x_e \geq 0 & \forall e \in E. \end{cases}$$

and (ii) proved the following lemma:

Lemma 1.1. *Let $x \in P_{\text{st}}$. Then, there exists a direction $d \in \mathbb{R}^E$ with exactly two non-zero coordinates, one -1 and one $+1$, and an $\epsilon, \delta > 0$ so that $x + \epsilon d \in P_{\text{st}}$ and $x - \delta d \in P_{\text{st}}$, both points have a new tight constraint, and all previous tight constraints remain tight.*

This allowed us to define the randomized pipage rounding algorithm:

Randomized Pipage Rounding (for P_{st})

Given $x \in P_{\text{st}}$, apply Lemma 1.1 to find points $x + \epsilon d$ and $x - \delta d$ (where $d \in \mathbb{Z}^E$ has exactly two non-zero coordinates, one -1 and one 1). With probability $p = \frac{\delta}{\epsilon + \delta}$, move to $x + \epsilon d$. Otherwise, move to $x - \delta d$.

Repeat until x is integral and return the resulting tree T.

In the previous lecture, we argued that this algorithm runs in deterministic polynomial time. In this lecture, we will complete the proof of the following theorem by proving (1) and (2):

Theorem 1.2. *Let $x \in P_{\text{st}}$. Then, if T is the random tree produced by randomized pipage rounding, the following holds:*

1. *$\mathbb{P}[e \in T] = x_e$ for all $e \in E$.*
2. *$\mathbb{P}[S \subseteq T] \leq \prod_{e \in S} x_e$ for all $S \subseteq E$, i.e. the distribution over trees produced by pipage rounding is negatively correlated.*

Furthermore, randomized pipage rounding runs in deterministic polynomial time.

1.2 Expectation is Preserved

Proving that $\mathbb{P}[e \in T] = x_e$ for all $e \in E$ is almost obvious, since at every step, if x' is the new point produced, then by our choice of p we have $\mathbb{E}[x'] = x$. However, we will prove it formally to better set up the proof of negative correlation (2).

Let $x^1, x^2, \ldots, x^\ell$ be the sequence of points returned by pipage rounding with $x^1 = x$ and x^ℓ the integral tree. For convenience, we choose ℓ such that x^ℓ is an integral tree with probability 1. In this way, it may be that at some point i in the process, $x^i = x^{i+1} = \cdots = x^\ell$.

We will now prove that $\mathbb{P}[e \in T] = x_e$ for all $e \in E$, or equivalently $\mathbb{E}\left[x_e^\ell\right] = x_e$ for all $e \in E$. In particular, we will show that $\mathbb{E}\left[x_e^{i+1} \mid x^i\right] = x_e^i$ for all $1 \leq i \leq \ell - 1$. This implies $\mathbb{E}\left[x_e^{i+1}\right] = \mathbb{E}\left[x_e^i\right]$ by taking the expectation of both sides,[1] which gives the desired claim since $\mathbb{E}\left[x_e^\ell\right] = \mathbb{E}\left[x_e^{\ell-1}\right] = \cdots = \mathbb{E}\left[x_e^1\right] = x_e$. So it remains to prove the following.

Fact 1.3. *$\mathbb{E}\left[x_e^{i+1} \mid x^i\right] = x_e^i$ for all i.*

Proof. There are two cases, where notice that we may assume the algorithm deterministically chooses the values of d, ϵ, and δ given an input x^i.

1. Given that the output of step i is x^i, e is not updated in step $i+1$, i.e. $d_e = 0$. Then, $x_e^{i+1} = x_e^i$ and the claim is true.

2. Given that the output of step i is x^i, in step $i+1$ d_e is non-zero. Then,

$$\mathbb{E}\left[x_e^{i+1} \mid x^i\right] = \frac{\delta}{\epsilon + \delta}(x_e^i + \epsilon d_e) + \frac{\epsilon}{\epsilon + \delta}(x_e^i - \delta d_e) = x_e^i.$$

□

Therefore, property (1) in Theorem 1.2 is true.

1.3 Negative Correlation

We will prove property (2) in Theorem 1.2 in an analogous way. We need to show that for every set $S \subseteq E$, we have $\mathbb{E}\left[\prod_{e \in S} x_e^\ell\right] \leq \prod_{e \in S} x_e$. Thus, as before, we will do this inductively. We will show that for every i,

$$\mathbb{E}\left[\prod_{e \in S} x_e^{i+1} \mid x^i\right] \leq \prod_{e \in S} x_e^i \tag{1}$$

Similarly to above, by taking expectations of both sides and chaining the inequalities we prove the desired property. So it remains to show Eq. (1).

Fact 1.4. *Eq. (1) holds.*

Proof. Given x^i, the algorithm will pick two edges e, f and define $d_e = 1, d_f = -1$ and $d_g = 0$ for all $g \neq e, f$. Thus, for all variables x_g^{i+1} for $g \neq e, f$, $x_g^{i+1} = x_g^i$. Therefore, if $S \cap \{e, f\} = \emptyset$,

[1] In particular, we have $\mathbb{E}\left[x_e^{i+1}\right] = \sum_{x^i} \mathbb{P}\left[x^i\right] \mathbb{E}\left[x_e^{i+1} \mid x^i\right] = \sum_{x^i} \mathbb{P}\left[x^i\right] x_e^i = \mathbb{E}\left[x_e^i\right]$.

the claim trivially holds. If $|S \cap \{e, f\}| = 1$, then this holds with equality due to Fact 1.3. So, the interesting case is when $|S \cap \{e, f\}| = 2$. Then,

$$\mathbb{E}\left[\prod_{e \in S} x_e^{i+1} \mid x^i\right] = \mathbb{E}\left[x_e^{i+1} x_f^{i+1} \mid x^i\right] \prod_{g \neq e, f \in S} x_g^i$$

So to prove the claim it suffices to show that $\mathbb{E}\left[x_e^{i+1} x_f^{i+1} \mid x^i\right] \leq x_e^i x_f^i$. We can verify this easily:

$$\begin{aligned}\mathbb{E}\left[x_e^{i+1} x_f^{i+1} \mid x^i\right] &= \frac{\delta}{\epsilon + \delta}(x_e^i + \epsilon)(x_f^i - \epsilon) + \frac{\epsilon}{\epsilon + \delta}(x_e^i - \delta)(x_f^i + \delta) \\ &= x_e^i x_f^i - \epsilon\delta \leq x_e^i x_f^i\end{aligned}$$

as desired. □

Therefore, following the discussion in the previous subsection, (2) holds in Theorem 1.2.

1.4 Chernoff bounds

Perhaps the most important thing about negatively correlated distributions is that they imply exactly the same Chernoff bounds we have been using for independent Bernoullis. In this lecture we will use the following one:

Theorem 1.5 (Chernoff Lower Tail)**.** *Let $X_1, \ldots, X_n$ be negatively correlated Bernoulli random variables. Then if $X = \sum_{i=1}^n X_i$, $L \leq \mathbb{E}[X]$, and $0 \leq \delta \leq 1$, we have:*

$$\mathbb{P}[X \leq (1-\delta)L] \leq \exp(-L\delta^2/2)$$

2 Applying Pipage Rounding to k-ECSM

The k-ECSM problem is defined as follows. Given a graph $G = (V, E)$ with costs $c_e \geq 0$ on the edges, find the cheapest collection of edges that k-edge-connects the graph. Every edge can be used as many times as desired at the same cost. Recall the following polytope for k-edge-connectivity:

$$P_{k-EC} = \begin{cases} \sum_{e \in \delta_G(S)} x_e \geq k & \forall \emptyset \subsetneq S \subsetneq V \\ x_e \geq 0 & \forall e \in E \end{cases}$$

The first step of the approximation algorithm of [Kar+22] will be to solve the natural LP, i.e. find a point x in this polytope minimizing $\sum_{e \in E} c_e x_e$.

Lemma 2.1. *Given $x \in P_{k-EC}$,*

$$\left(1 - \frac{1}{n}\right) \frac{2}{k} x \in P_{\text{st}}$$

Proof. We first apply the parsimonious property [GB93] without proof, which says that we may assume all vertices have degree exactly k, i.e. $x(\delta(v)) = k$ for all $v \in V$. Now we simply verify the inequalities. Let $y = (1 - \frac{1}{n})\frac{2}{k}x$. First, $x(E) = \frac{kn}{2}$, so $y(E) = \frac{kn}{2}(1 - \frac{1}{n})\frac{2}{k} = n - 1$. Second, fix any cut $S \subset V$. Then,

$$\sum_{e \in E(S)} x_e = \frac{k|S| - x(\delta(S))}{2} \le \frac{k}{2}(|S| - 1)$$

Therefore after scaling down x by $\frac{2}{k}$, $y(E(S)) \le |S| - 1$ (note we did not even need the additional $1 - \frac{1}{n}$ factor). □

We can now define the algorithm.

Algorithm for k-ECSM

1. Solve the LP to obtain (parsimonious) $x \in P_{k-EC}$.
2. Let $y = \frac{2}{k}(1 - 1/n)x$. Sample $\frac{k}{2} + 1$ trees $T_1, \ldots, T_{k/2+1}$ independently using pipage rounding starting at $y \in P_{\text{st}}$ (which holds by Lemma 2.1). Let T^* be the multi-set union of these trees, i.e. counting edges with multiplicity.
3. Add $\sqrt{k \ln k}$ copies of the minimum spanning tree to T^*.
4. Initialize $R = \emptyset$. For each tree i in $T_1, \ldots, T_{k/2+1}$, check each of its cuts $S \subset V$ with $|\delta_{T_i}(S)| = 1$. If $|\delta_{T^*}(S)| < k$, put a copy of the unique edge $e \in \delta_{T_i}(S)$ in the set R.
5. Output $R \uplus T^*$.

Lemma 2.2. *The output of the algorithm, $R \uplus T^*$, is k-edge-connected (with probability 1).*

Proof. Consider any cut $S \subset V$. We will show it has at least k edges in the output. If $|\delta_{T^*}(S)| \ge k$, then there is nothing to prove. Otherwise, $|\delta_{T^*}(S)| < k$. In this case, in (4) of the above algorithm, every tree T_i such that $|\delta_{T_i}(S)| = 1$ will increase $|\delta_R(S)|$ by 1. In this way, every tree T_i contributes at least 2 edges to $|\delta_{R \uplus T^*}|$. The fact that there are at least $k/2$ trees completes the proof. □

We will now analyze the expected cost of the algorithm. Let X_e^i indicate if edge e is added to R as a result of tree T_i.

Lemma 2.3. $\mathbb{E}\left[X_e^i\right] \le 2k^{-3/2}x_e$.

Proof. X_e^i is 0 if $e \notin T_i$. Therefore it suffices to bound $\mathbb{E}\left[X_e^i \mid e \in T_i\right]$. Condition on $e \in T_i$ and any tree T_i. Then,

$$\mathbb{P}\left[X_e^i = 1 \mid T_i\right] \le \mathbb{P}\left[|\delta_{T^* \setminus T_i}(S)| < k\right]$$

In other words, $X_e^i = 0$ whenever the remaining $k/2$ trees have at least k edges across S (of course, it is sufficient that the remaining $k/2$ trees have at least $k - 1$ edges, but we use this for ease of notation).

Let $T = \biguplus_{j \ne i \in [k/2+1]} T_i$. By (3) and the above equation,

$$\mathbb{P}\left[X_e^i = 1 \mid T_i\right] \le \mathbb{P}\left[|\delta_T(S)| \le k - \sqrt{k \ln k}\right]$$

Using that $x(\delta(S)) \geq k$, we have $y(\delta(S)) \geq 2(1-1/n)$, so (ignoring the $(1-1/n)$ factor[2]) we obtain $\mathbb{E}[|\delta_T(S)|] \geq k$. Applying a Chernoff bound with $L = k$ and $\delta = \sqrt{\frac{\ln k}{k}}$, we obtain a bound of

$$\mathbb{P}\left[X_e^i = 1 \mid T_i\right] \leq e^{-\delta^2 k/2} = \frac{1}{\sqrt{k}}$$

Now we can bound:

$$\mathbb{E}\left[X_e^i\right] = \sum_T \mathbb{P}[T]\,\mathbb{I}\{e \in T_i\}\,\mathbb{P}\left[X_e^i = 1 \mid T_i\right] \leq \frac{1}{\sqrt{k}} \sum_T \mathbb{P}[T]\,\mathbb{I}\{e \in T_i\} \leq 2k^{-3/2} x_e$$

where we used that $\mathbb{P}[e \in T_i] \leq \frac{2}{k} x_e$. □

Theorem 2.4. *This algorithm is a* $1 + O\left(\sqrt{\frac{\log k}{k}}\right)$ *approximation for k-ECSM.*

Proof. First we bound the expected cost of sampling the first $k/2+1$ trees. This is at most $\frac{2}{k} \cdot (\frac{k}{2}+1) \cdot c(x) = (1+\frac{2}{k})c(x) \leq (1+\frac{2}{k}) \cdot OPT$ in expectation.

Next we bound the cost of the additional $\sqrt{k \ln k}$ trees. Since the spanning tree polytope is integral, $c(MST) \leq c(y)$, so this costs at most $\sqrt{k \ln k} \cdot \frac{2}{k} \cdot OPT = 2\sqrt{\frac{\ln k}{k}} \cdot OPT$ in expectation.

Finally, we bound the expected cost of R. Let $X_e = \sum_{i=1}^{k/2+1} X_e^i$. Then, $c(R) = \sum_{e \in E} X_e c_e$. But,

$$\mathbb{E}[X_e] = \left(\frac{k}{2}+1\right)\mathbb{E}\left[X_e^i\right] \leq \left(\frac{k}{2}+1\right) 2k^{-3/2} x_e \leq \frac{3}{\sqrt{k}} x_e$$

Where we use a very loose bound for the last inequality for simplicity. Therefore by linearity of expectation,

$$\mathbb{E}[c(R)] \leq \sum_{e \in E} \frac{3}{\sqrt{k}} c_e x_e \leq \frac{3}{\sqrt{k}} \cdot OPT$$

Summing the three terms proves the theorem. □

We note that [Kar+22] proves a stronger guarantee of $1 + O(1/\sqrt{k})$ for a variant of this algorithm using max entropy sampling instead of pipage rounding. This was improved to $1 + O(1/k)$ by [HKZ24], who also demonstrated that no asymptotically better approximation exists unless P=NP.

References

[AS04] A. A. Ageev and M. I. Sviridenko. "Pipage Rounding: A New Method of Constructing Algorithms with Proven Performance Guarantee". In: *Journal of Combinatorial Optimization* 8.3 (2004), pp. 307–328. ISSN: 1573-2886. DOI: 10.1023/B:JOCO.0000038913.96607.c2 (cit. on p. 1).

[CVZ10] Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. "Dependent Randomized Rounding via Exchange Properties of Combinatorial Structures". In: *FOCS*. 2010, pp. 575–584 (cit. on p. 1).

[2] We can increase the expectation of $|\delta_T(S)|$ to at least k by sampling every edge independently with probability $\frac{2}{kn} x_e$ for each tree T_i

[GB93] Michel X. Goemans and Dimitris Bertsimas. “Survivable networks, linear programming relaxations and the parsimonious property”. In: *Math. Program.* 60 (1993), pp. 145–166 (cit. on p. 3).

[HKZ24] D. Ellis Hershkowitz, Nathan Klein, and Rico Zenklusen. “Ghost Value Augmentation for k-Edge-Connectivity”. In: *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*. STOC 2024. Vancouver, BC, Canada: Association for Computing Machinery, 2024, 1853–1864. isbn: 9798400703836. doi: 10.1145/3618260.3649715 (cit. on p. 5).

[Kar+22] Anna R. Karlin, Nathan Klein, Shayan Oveis Gharan, and Xinzhi Zhang. “An improved approximation algorithm for the minimum k-edge connected multi-subgraph problem”. In: *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*. STOC 2022. New York, NY, USA: Association for Computing Machinery, 2022, 1612–1620 (cit. on pp. 3, 5).