University of Michigan–Ann Arbor

Department of Electrical Engineering and Computer Science

EECS 498 004 **Advanced Graph Algorithms**, Fall 2021

Lecture 14.0: The Cut-Matching Game: Deterministic Cut Player

October 12 & 14, 2021

Instructor: Thatchaphol Saranurak

Scribe: Yi Tang

The cut-matching game provides an *implicit* construction of expanders, where it "forces an adversary" to construct an expander.

1 The Game

The cut-matching game involves two players: the cut player and the matching player. The game starts with an empty graph G_0 with vertex set V; $|V| = n$. For each round $i = 1, 2, \ldots$:

- The cut player chooses a cut (A_i, B_i) (assuming $|A_i| \leq |V|/2$).
- The matching player gives a maximal matching M_i between A_i and B_i, with size $|A_i|$.
- Set $G_i = G_{i-1} \cup M_i$ (retaining parallel edges).

The game ends when G_i is an expander (say an $\Omega(1/\text{polylog}(n))$-expander). For simplicity, denote the total number of rounds by I and $G = G_I$.

In the cut-matching game, the objective of the cut player is to minimize the total number of rounds I by choosing the cuts cleverly, while the matching player behaves adversarially, trying to delay the process as much as possible.

In the following sections we will show strategies for the cut player so that no matter how badly the matching player behaves, the game yields an expander in $I = \text{polylog}(n)$ rounds.

2 Heuristics: Balanced Sparse Cuts

First of all we observe the following heuristics for the cut player:

- The chosen cut should be balanced, so that M_i is large and contains plenty of edges.
- The chosen cut should be sparse, or should contain a sparse cut inside, so that M_i makes progress in fixing sparse cuts, which is necessary for increasing the global conductance.

It turns out that these two heuristics are enough, while it is not trivial at all to formalize.

3 High-level Goal for Analysis: Random Walks that Embeds a Clique

In this section we show a sufficient condition for the game to end and yield an expander.

Consider the following random walk:

- Start from some vertex v_0.
- For each step/round $i \in [I]$:
  - If $v_{i-1} \notin M_i$ then stay put at $v_i = v_{i-1}$.
  - Otherwise if $(v_{i-1}, u) \in M_i$ for some vertex u then take one (lazy) random walk step according to M_i, i.e.,
    * with probability $1/2$, stay put at $v_i = v_{i-1}$;
    * with probability $1/2$, move to $v_i = u$.

Based on the random walk, define $p_i(u, v)$ to be the probability of reaching v at step i in the random walk starting from u.

Let $\mathbf{P}_i \in \mathbb{R}^{V \times V}$ be the matrix with value $p_i(u, v)$ at entry (u, v). During the random walk, $\mathbf{P}_i$ evolves as follows:

- $\mathbf{P}_0 = \mathbf{I}$.
- For $v \notin M_i$, the column satisfies $\mathbf{P}_i(\cdot, v) = \mathbf{P}_{i-1}(\cdot, v)$.
- For $(u, v) \in M_i$, the columns satisfy $\mathbf{P}_i(\cdot, u) = \mathbf{P}_i(\cdot, v) = (\mathbf{P}_{i-1}(\cdot, u) + \mathbf{P}_{i-1}(\cdot, v))/2$.

I.e., in each round $i \in I$, the columns of $\mathbf{P}_i$ are obtained by averaging the columns of $\mathbf{P}_{i-1}$ according to the matchings in M_i.

By induction, it is easy to see that the columns of $\mathbf{P}_i$ always sum to 1, i.e., $\sum_u \mathbf{P}_i(u, v) = 1$ for all v and i (as in each round the columns either remain the same or get averaged). (Cf. the rows always sum to 1 by definition.)

Consider all together the random walks starting from all vertices, and regard the random walk as distributing masses, with initial mass 1 at each vertex at the beginning. Then the total mass going through edge $(u, v) \in M_i \subseteq G$ during the random walks is exactly

$$\underbrace{\frac{1}{2} \sum_a \mathbf{P}_{i-1}(a, u)}_{\text{from } u \text{ to } v} + \underbrace{\frac{1}{2} \sum_b \mathbf{P}_{i-1}(b, v)}_{\text{from } v \text{ to } u} = \frac{1}{2} + \frac{1}{2} = 1\,,$$

since the edge $(u, v) \in M_i$ is involved only during round i in the random walk (recall that G retains parallel edges).

Let K be the 1-product graph (i.e., clique with weight $1/n$ on every edge). The following lemma tells that a sufficient condition for the game to end and yield a graph that embeds a clique is to have p_i be well-mixed.

Lemma 3.1. *If $p_I(u, v) \geq 1/(2n)$ for all u, v, then $K/2 \preceq^{flow} G$ (and thus $K/2 \preceq^{cut} G$).*

Proof. Consider the flow that "follows the random walk," i.e., routes commodities in the same way as the random walk distributes masses (see Figure 1). Indeed, it suffices to construct a flow that routes at least $1/2n$ on each edge of K in G with congestion at most 1 since we could scale them with constant factors to make them satisfies the demands exactly. We construct the flow inductively by considering adding one matching M_t at a time and route $(p_t(i,j)+p_t(j,i))$ amount of flow between i and j in G_t, where $(p_I(i,j)+p_I(j,i)) \geq 1/n > \kappa_K(\{i,j\})$ by premise. For the base case when $t=0$ where we have no edges, $\forall(a,b) \in V \times V$, when $a \neq b$, we set $f^0_{a,b} := \mathbf{0}$ and $f^0_{a,a}$: $P^0_{a,a} \mapsto p_0(a,a) = 1$, where $P^0_{a,a} =< a >$. When a new edge $\{u,v\} \in M_{t+1}$ is added to G_t, $\forall a \in V$, the amount of flow that we need to route between a and u increases by $\left(\frac{1}{2}p_t(a,v) - \frac{1}{2}p_t(a,u)\right)$. In order to maintain the congestion yet satisfies the new demand, we let

$$f^{(t+1)}_{a,u} : \begin{cases} P^{(t)}_{a,u} \mapsto \frac{1}{2} f^{(t)}_{a,u}(P_{a,u}), & \text{for } P^{(t)}_{a,u} \text{ is a path from } a \text{ to } u \text{ in } G_t \\ P^{(t)}_{a,v}; < v,u > \mapsto \frac{1}{2} f^{(t)}_{a,v}(P_{a,v}), & \text{for } P^{(t)}_{a,v}; < v,u > \text{ is the concatenation of } P^{(t)}_{a,v} \text{ and } < v,u > \\ P^{(t+1)}_{a,u} \mapsto 0, & \text{o.w.} \end{cases}$$

as described in Figure 2. A Symmetric analysis and construction applies to $f^{(t+1)}_{a,v}$. In this way, the congestion on edges in G_t won't change and by the analysis above the lemma and inducting on t we know the total amount of flow going through $\{u,v\}$ is exactly 1, which means the congestion of the flow is also exactly 1. Hence $K/2 \preceq^{\text{flow}} G$. □

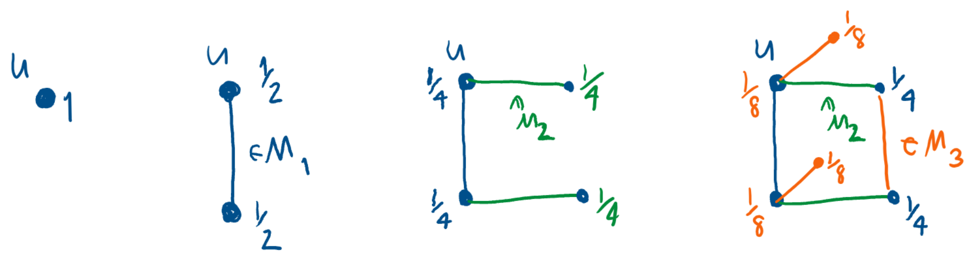

Figure 1: The distribution of masses from u to each vertex during the lazy random walk based on M_t can also be regarded as the amount of flow that we need to route from u to each vertex in the flow-embedding that we constructed at each inductive step. Namely, after M_t is added, we need that $\forall x \in V, ||f_{u,x}|| = p_t(u,x)$.

Figure 2: The figure shows how flows from a to u and a to v changes when $\{u,v\}$ was added. The amount of flow on $\{u,v\} \in M_{t+1}$ when it was added is $\frac{1}{2}\sum_{a\in V}(f^{(t)}_{a,v}(P_{a,v}) + f^{(t)}_{a,u}(P_{a,u})) = \frac{1}{2}\sum_{a\in V}(p_t(a,v)+p_t(a,u)) = 1$ by inducting on t, and the amount of flow on edges in G_t remains the same.

On the other hand, since $G = G_I$ has maximum degree I, we know $G \preceq^{\text{cut}} 2I \cdot K$. Then $K/2 \preceq^{\text{cut}} G \preceq^{\text{cut}} 2I \cdot K$, so K and G are $O(I)$−cut-equivalent. More precisely, the flow-embedding that embeds $K/2$ into G act as a certificate of G being an $\Omega(1/maxdeg(G))$-expander. Hence if $I = \text{polylog}(n)$ then $maxdeg(G) = \text{polylog}(n)$ and we obtain $\Omega(1/\text{polylog}(n))$-expanders as desired.

We will show two algorithms/strategies for the cut player: a randomized one and a deterministic one. The randomized one uses the sufficient condition established in this section, while the deterministic one uses something else yet similar. Nevertheless the deterministic one is actually more intuitive and will be presented first.

4 Notations

In this section we introduce some notations useful in the analysis.

Definition 4.1. A cut $S \subseteq V$ is *β-balanced* if $\min\{|S|, |V \setminus S|\} \geq \beta |V|$.

Definition 4.2. The *expansion* of cut $S \subseteq V$ in graph G is defined by

$$\Psi_G(S) = \frac{|E(S, V \setminus S)|}{\min\{|S|, |V \setminus S|\}} .$$

The expansion of the graph G is defined by $\Psi(G) = \min_{\emptyset \neq S \subsetneq V} \Psi_G(S)$.

Recall that this notion of expansion is the same as d-expansion with $d(v) = 1$ for all v.

Note that since $|S| \leq \mathrm{vol}_G(S) \leq d_{\max}(G) \cdot |S|$ (assuming unweighted graphs), we have $\Phi(G) \leq \Psi(G) \leq d_{\max}(G) \cdot \Phi(G)$, which means the expansion $\Psi(G)$ is a good proxy to the conductance $\Phi(G)$, especially when $d_{\max}(G) = \mathrm{polylog}(n)$.

5 Deterministic Exponential-Time Algorithm

Theorem 5.1 ([KKOV07])**.** *There is a deterministic exponential-time algorithm for the cut player so that after $I = O(\log n)$ rounds, $\Psi(G) = \Omega(1)$ (and thus $\Phi(G) = \Omega(1/\log n)$).*

Proof. The deterministic algorithm works as follows:

- If there exists a (1/4)-balanced sparse cut (A_i, B_i) with $\Psi_{G_{i-1}}(A_i) \leq 1/100$ then choose such a cut (A_i, B_i).
- Otherwise there must be a cut (A_i, B_i) with $|A_i| \leq n/4$ and $\Psi(G_{i-1}[B_i]) > 1/400$; then the cut player choose (A_i, B_i), and assert that it is the last cut to choose and leads to $\Psi(G_i) > 1/1200$.

Note that the (1/4)-balanced cut (A_i, B_i) with $\Psi_{G_{i-1}}(A_i) \leq 1/100$ really follows the heuristics of choosing a *balanced sparse cut*. The correctness and efficiency of the algorithm will be analyzed throughout this section. □

5.1 Entropy Potential

For vertex u and round i, define the *entropy potential* by

$$\Pi_i(u) = \mathrm{H}[p_i(u, \cdot)] = -\sum_v p_i(u, v) \log p_i(u, v) .$$

Also define the total entropy potential by $\Pi_i = \sum_u \Pi_i(u)$.

Note that the entropy of a probability distribution over n elements is bounded between 0 and $\log n$, and is maximized by (and only by) the uniform distribution, thus measuring how well-spread/well-mixed the distribution is. This gives $\Pi_i(u) \leq \log n$ and $\Pi_i \leq n \log n$.

5.2 All Rounds except the Last

The following lemma shows that in the algorithm, the entropy potential Π_i increase by $\Omega(n)$ in each round, thus concluding that there are at most $I = O(\log n)$ rounds in the algorithm.

Lemma 5.2. *If there exists an $\Omega(1)$-balanced cut (A_i, B_i) with $\Psi_{G_{i-1}}(A_i) \leq 1 - \delta$ where $\delta = \Omega(1)$, then $\Pi_i - \Pi_{i-1} = \Omega(n)$.*

(Recall that the algorithm uses $(1/4)$-balanced cut with $\delta = 99/100$.)

Proof. Denote $p_i(S, T) = \sum_{u \in S, v \in T} p_i(u, v)$, which represents the total mass going from S to T in the first i rounds. Consider $p_{i-1}(A_i, B_i)$, the total mass going from A_i to B_i before round i. By the same analysis as in Section 3, this total mass is at most $\sum_{j<i} |M_j|/2 = |E_{G_{i-1}}(A_i, B_i)|/2$ (and the total mass going from B_i to A_i corresponds to the other half). Then since $\Psi_{G_{i-1}}(A_i) \leq 1 - \delta$, we know

$$p_{i-1}(A_i, B_i) \leq |E_{G_{i-1}}(A_i, B_i)|/2 \leq (1 - \delta)|A_i|/2 \; .$$

This implies that there are at least $\Omega(\delta |A_i|) = \Omega(n)$ vertices $a \in A_i$ where $p_{i-1}(a, B_i) \leq 1/2 - \delta/3$. We say such vertices to be *confined* (see Figure 3).

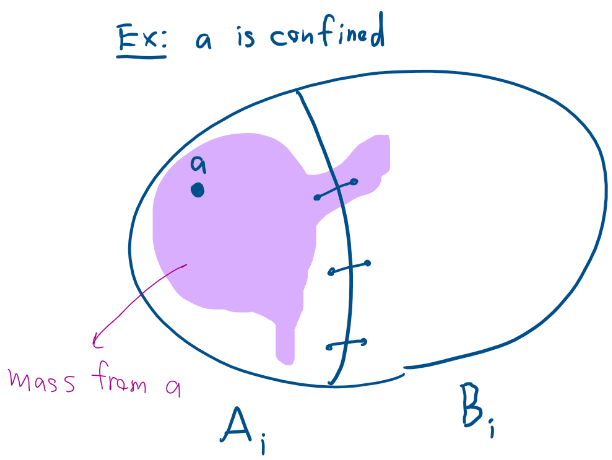

Figure 3: Illustration of a confined vertex a for cut (A_i, B_i).

For a confined vertex $a \in A_i$, intuitively, little mass at a goes into B_i and much of the mass remains in A_i, so there must be a large fraction of $(u, v) \in M_i$ such that $p_{i-1}(a, u) \gg p_{i-1}(a, v)$. Formally, we say an edge $(u, v) \in M_i$ to be *a-spreading* if $p_{i-1}(a, u) \geq (1 + \delta) \cdot p_{i-1}(a, v)$. For an a-spreading edge (u, v), since $p_{i-1}(a, u)$ and $p_{i-1}(a, v)$ differs notably, we expect it to contribute large entropy increase after averaging the probabilities according to M_i.

Partition A_i into $S_a = \{u \in A_i : (u, v) \in M_i \text{ is } a\text{-spreading}\}$ and $N_a = \{u \in A_i : (u, v) \in M_i \text{ is } a\text{-non-spreading}\}$, and let $N_a^* = \{v \in B_i : (u, v) \in M_i, u \in N_a\}$. Note that $\sum_{u \in A_i} p_{i-1}(a, u) = p_{i-1}(a, A_i) = 1 - p_{i-1}(a, B_i)$, while for the a-non-spreading edges,

$$\sum_{u \in N_a} p_{i-1}(a, u) \leq (1 + \delta) \sum_{v \in N_a^*} p_{i-1}(a, v) \leq (1 + \delta) \cdot p_{i-1}(a, B_i) \; .$$

Therefore, for the a-spreading edges,

$$\sum_{u \in S_a} p_{i-1}(a, u) \geq 1 - (2 + \delta) \cdot p_{i-1}(a, B_i) \geq \delta/6 + \delta^2/3 = \Omega(1) \; .$$

Observe that for $1 \geq p \geq (1+\Theta(1)) \cdot q \geq 0$,

$$-(p+q)\log\frac{p+q}{2} + p\log p + q\log q = \Omega(p)\ .$$

(This follows by noting that the expression on the left hand side is decreasing in q.) Hence adding an a-spreading edge $(u,v) \in M_i$ then averaging the probabilities according to the random walk process, $\Pi_{i-1}(a)$ increases by $\Omega(p_{i-1}(a,u))$. Summing over all a-spreading edges and all confined vertices a, the total entropy increases at least by $\Omega(n) \cdot \sum_{u \in S_a} \Omega(p_{i-1}(a,u)) = \Omega(n)$, as desired. □

5.3 Last Round

The following lemmas show that the assertions made in the algorithm are always valid.

Lemma 5.3. *If there is no $(1/4)$-balanced cut B with $\Psi_{G_{i-1}}(B) \leq \psi$, then there exists cut (A_i, B_i) such that $|A_i| \leq n/4$ and $\Psi(G_{i-1}[B_i]) > \psi/3$.*

(Recall that the algorithm uses $\psi = 1/100$.)

Proof. Consider the following procedure:

- Start with $H \leftarrow G_{i-1}$, $A \leftarrow \emptyset$.
- While there exists a sparse cut S with $\Psi_H(S) \leq \psi/3$ and $|S| \leq |V_H|/2$, update $H \leftarrow H \setminus S$, $A \leftarrow A \cup S$.
- Return H, A.

Obviously the procedure returns H with $\Psi(H) > \psi/3$. Then it suffices to show that the procedure also returns A with $|A| \leq n/4$, so that setting $A_i = A$ (and hence $G_{i-1}[B_i] = H$) satisfies all the desired properties.

Suppose towards contradiction that $|A| > n/4$. Since A is initialized to be $\emptyset$, it means there is a certain round where $|A| \leq n/4$ while $|A \cup S| > n/4$. Note that $|S| \leq |V_H|/2 \leq n/2$. Hence $A \cup S$ is a $(1/4)$-balanced cut. Moreover, $A \cup S$ is the disjoint union of a sequence of cuts $\{S_j\}_{j \in [J]}$ with $\Psi_H(S_j) \leq \psi/3$, so the cut size of $A \cup S$ in G_{i-1} is at most $(\psi/3) \cdot |A \cup S| \leq (\psi/3) \cdot (3n/4) = \psi n/4$ (as every edge cut by $A \cup S$ in G_{i-1} must have been cut by some S_j in the corresponding iterate of H). This implies $\Psi_{G_{i-1}}(A \cup S) \leq (\psi n/4)/(n/4) = \psi$, which contradicts the given condition. □

Lemma 5.4. *For any cut (A_i, B_i) (assuming $|A_i| \leq n/2$), $\Psi(G_i) > \min\{\Psi(G_{i-1}[B_i])/2, 1\}$.*

(Recall that the algorithm uses $|A_i| \leq n/4$ and $\Psi(G_{i-1}[B_i]) > 1/400$.)

Proof. Intuitively, for any cut S in G_i, if a large fraction of S is in A_i, then the expansion of S is large as the edges in M_i goes from $S \cap A_i$ to B_i; while if a large fraction of S is in B_i, then the expansion of S is large as well since $G_{i-1}[B_i]$ has large expansion.

Let $\psi = \Psi(G_{i-1}[B_i])$. Consider an arbitrary cut S in G_i, with $S = S_A \cup S_B$ where $S_A = S \cap A_i$ and $S_B = S \cap B_i$. Suppose without loss of generality that $|S_B| \leq |B_i|/2$, and let S_B^* be the matching of S_B under M_i, i.e., $S_B^* = \{u \in A_i : (u,v) \in M_i, v \in S_B\}$ (clearly $|S_B^*| \leq |S_B|$ as $|A_i| \leq n/2$). Then by direct calculations without case analysis,

$$\begin{aligned}
\Psi(G_i) &\geq \frac{|\partial_{G_{i-1}[B_i]} S_B| + |S_A \oplus S_B^*|}{|S_B| + |S_A|} \\
&\geq \frac{\psi|S_B| + |S_A \setminus S_B^*|}{|S_B| + |S_A|} \\
&= \frac{\psi|S_B| + |S_A \setminus S_B^*|}{|S_B| + |S_A \cap S_B^*| + |S_A \setminus S_B^*|} \\
&\geq \frac{\psi|S_B| + |S_A \setminus S_B^*|}{2|S_B| + |S_A \setminus S_B^*|} \geq \min\{\psi/2, 1\} \ .
\end{aligned}$$

(Here $\oplus$ denotes symmetric difference of sets.) □

5.4 Why Exponential Time?

The algorithm costs exponential time because of finding sparse cuts; specifically, the algorithm involves the task of finding either a $(1/4)$-balanced cut S with $\Psi_{G_{i-1}}(S) \leq 1/100$ or a subgraph $G_{i-1}[B_i]$ of size $|B_i| \geq 3n/4$ with $\Psi(G_{i-1}[B_i]) > 1/400$, and the latter implies, say, all $(3/8)$-balanced cuts in G_{i-1} have expansions at least $1/3200$. So the algorithm decides the expansion of a graph (with regard to only the $(3/8)$-balanced cuts though) up to a constant factor. However, it is NP-hard to decide whether or not $\Psi(G) \leq \psi$ given (G, ψ) [MS90] (and unique-game-hard to decide up to any constant factor [CKK$^+$06, KV15]).

This complexity issue can be resolved by using polylog(n)-approximate sparse cuts, which can be computed efficiently. In particular, by Lemma 5.4, finding $\Psi(G_{i-1}[B_i]) \geq 1/\text{polylog}(n)$ instead of $\Psi(G_{i-1}[B_i]) \geq \Omega(1)$ is still good enough for obtaining $\Psi(G) \geq 1/\text{polylog}(n)$.

References

[CKK$^+$06] Shuchi Chawla, Robert Krauthgamer, Ravi Kumar, Yuval Rabani, and D. Sivakumar. On the hardness of approximating multicut and sparsest-cut. *Comput. Complexity*, 15(2):94–114, 2006. Preliminary version in CCC 2005.

[KKOV07] Rohit Khandekar, Subhash A. Khot, Lorenzo Orecchia, and Nisheeth K. Vishnoi. On a cut-matching game for the sparsest cut problem. Technical Report UCB/EECS-2007-177, EECS Department, University of California, Berkeley, Dec 2007.

[KV15] Subhash A. Khot and Nisheeth K. Vishnoi. The unique games conjecture, integrability gap for cut problems and embeddability of negative-type metrics into ℓ_1. *J. ACM*, 62(1):Art. 8, 39, 2015. Preliminary version in FOCS 2005.

[MS90] David W. Matula and Farhad Shahrokhi. Sparsest cuts and bottlenecks in graphs. volume 27, pages 113–123. 1990. Computational algorithms, operations research and computer science (Burnaby, BC, 1987).