Advanced Graph Algorithms Jan-Apr 2014

Lecture 19— 19 MARCH, 2014

Lecturer: Ashutosh Rai *Scribe: Aditya Potukuchi*

1 Overview

In the last lecture we saw a $\mathcal{O}(mr^\omega)$ algorithm that, given a linear matroid M, returns a subset of it's columns corresponding to its maximum parity and hinted at an algorithm with better running time.

To start off, we shall define a problem:

- *Matroid Parity Basis* Given a linear matroid M, does it have a Parity Basis?

In this lecture we show an $\mathcal{O}(m^\omega)$ algorithm that decides if a linear matroid M has a parity basis, i.e. there are $r/2$ pairs that are independent. One observation that can be made is that this algorithm works better than the previous one on matroids of large rank. Later, we shall reduce finding the maximum parity of a matroid to the Parity basis problem by an algorithm similar that that in *Lecture 12* given by Harvey.

2 Deciding Matroid Parity Basis

2.1 Some Preliminary Definitions

Using the previously defined notion of Parity Basis, we define some more terms for a Matroid $M = (E, \mathcal{I})$:

Definition 1 (Growable Set)**.** *A set $J \subset E$ is growable if it is part of some parity basis.*

We also use the notion of rank of a subset, i.e.

$$rank_M(U) = \{|I||I \subseteq U, I \in \mathcal{I}\}$$

and Contraction of a Matroid given by:

$$M|U = (E \setminus U, \mathcal{I}')I \in \mathcal{I}' iff I \cup U \in \mathcal{I}$$

The next lemma shows how the rank of a subset in the Contraction of a matroid compares to the rank in the original matroid.

Lemma 2. *For any 2 sets $A, B \subseteq E$, $B \in \mathcal{I}$, $rank_{M|B}(A) = rank_M(A \cup B) - |B|$*

Proof. Shall I leave this as an exercise? □

Next, we shall prove a slight generalization of *Lemma* 3 from *Lecture* 15 before we proceed.

Lemma 3. *Let Q and T be a skew symmetric matrices of size $N \times N$ such that T is a matrix on distinct variables. Let S be the set of rows/columns of Q/T. Then*

$$rank(Q+T) = max_{A \subseteq S}(rank(Q_{A,A}) + rank(T_{S \setminus A, S \setminus A}))$$

Proof. Let a set $A \subseteq S$ witness $max_{A \subseteq S}(rank(Q_{A,A}) + rank(T_{S \setminus A, S \setminus A}))$. By *Lemma number* from *Lecture number*, there are sets $A_1 \subseteq A$ and $A_2 \subseteq S \setminus A$, such that $|A_1| = rank(Q_{A,A})$, $|A_2| = rank(T_{S \setminus A, S \setminus A})$, and Q_{A_1,A_1} and T_{A_2,A_2} are both full rank. By *Lemma* 3 from *Lecture* 15, $Q_{A_1 \cup A_2, A_1 \cup A_2} + T_{A_1 \cup A_2, A_1 \cup A_2}$ is a full rank matrix. Moreover, it is a submatrix of $Q+T$. So, $rank(Q+T) \geq |A_1| + |A_2| = max_{A \subseteq S}(rank(Q_{A,A}) + rank(T_{S \setminus A, S \setminus A}))$.

Let k be the rank of $Q+T$. By *Lemma number* from *Lecture number*, there is a set $B \subseteq S$, such that $|B| = k$, and $(Q+T)_{B,B}$ is full rank. So, by *Lemma* 3 from *Lecture* 15, there must a partition of B into X and Y such that $Q_{X,X}$ and $T_{Y,Y}$ are both full rank. herefore $max_{A \subseteq S}(rank(Q_{A,A}) + rank(T_{S \setminus A, S \setminus A})) \leq |B| = k = rank(Q+T)$.

Using both the inequalities proved, $rank(Q+T) = max_{A \subseteq S}(rank(Q_{A,A}) + rank(T_{S \setminus A, S \setminus A}))$. □

Here also, we will relax the condition on T by saying that it is enough that it is enough that every nonsingular submatrix in T has a determinant that is a polynomial on a unique set of variables. The variables of T need not all be different.

2.2 Matrix Formulation

We define a matrix $Z(J)$, which is a variation of matrix Z in *Lecture 15* as follows:

$$Z(J) = \begin{bmatrix} 0 & M_{R,J} & M_{R,E \setminus J} \\ -M_{R,J}^T & 0 & 0 \\ -M_{R,E \setminus J^T} & 0 & T_J \end{bmatrix}$$

where R is the set of rows of the linear representation of the matroid M, and $T(J)$ is T (as defined in *Lecture* 15) restricted to $E \setminus J$.

We prove, using the following theorem that M has a parity basis iff Z_J is non-singular.

Theorem 4. *For any set $J \in \mathcal{I}$, $rank(Z_J) = 2v_{M|J} + 2m + |J|$, where $2m$ is the size of the ground set E of M, and $v_{M|J}$ is the size of the largest parity in $M|J$.*

Proof. Write Z_J in the folliwing manner:

$$\underbrace{\begin{bmatrix} 0 & M_{R,J} & M_{R,E \setminus J} \\ -M_{R,J}^T & 0 & 0 \\ -M_{R,E \setminus J^T} & 0 & T_J \end{bmatrix}}_{Z_J} = \underbrace{\begin{bmatrix} 0 & M_{R,J} & M_{R,E \setminus J} \\ -M_{R,J}^T & 0 & 0 \\ -M_{R,E \setminus J^T} & 0 & 0 \end{bmatrix}}_{Q} + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & T_J \end{bmatrix}}_{T}$$

Let S be the set of rows/columns of Z_J. $E \subseteq S$ corresponding to the columns of the matroid. Using *Lemma 3* proved above, $rank(Z_J) = max_{A \subseteq S}(rank(Q_{A,A}) + rank(T_{S \setminus A, S \setminus A}))$. Let $A' = A \cap E$.

We can assume that A' is a parity set, i.e. it contains columns that correspond to pairs in M. Suppose it is not, Let $B \supset A'$ be the minimal parity set that contains A'. $rank(Q_{B,B}) \geq rank(Q_{A',A'})$ and $rank(T_{S \setminus B, S \setminus B}) = rank(T_{S \setminus A', S \setminus A'})$, because for a column r to be non-zero in $T_{S \setminus A, S \setminus A}$, it must belong to $E \setminus J$, and both r and the column it is paired with, must be in $S \setminus A$ (similar to the case in *Theorem* 5 in *Lecture* 15). Hence, A' can be replaced by B.

Also, we may assume that $J \subseteq A'$. Suppose not, let $B = A' \cup J$. $rank(T_{S \setminus B, S \setminus B}) = rank(T_{S \setminus A', S \setminus A'})$ because T contains only 0 columns in J, and $rank(Q_{A',A'}) \leq rank(Q_{B,B})$, so A' can be replaced with B. So, $rank(T_{S \setminus A, S \setminus A}) = |S \setminus A| = |E| - |A'|$.

Also, note that $rank(Q_{A,A}) = 2rank(M_{R,A'})$, since $Mand - M^T$ do not have any rows/columns in Z_J. $rank(M_{R,A'}) = rank_M(M_{R,A'}) = rank_M(J) + rank_{M|J}(A' \setminus J)$.

So, putting $I = A' \setminus J$, we can write $rank(Z_J) = min_{I \subseteq V \setminus J}(2(rank_{M|J}(I) + |J|) + |E| - |I| - |J|) = min_{I \subseteq V \setminus J}(2rank_{M|J}(I) - |I|) + |V| + |J|$. Let some I maximize this quantity.

We may assume that I is an independent set. Suppose not, let $I' \subseteq I$ be the maximum size independent set in I. $2rank_{M|J}(I') - |I'| \geq 2rank_{M|J}(I) - |I|$. We may further assume that I is a parity set. Suppose not, let $I' \subset I$ be the maximal parity subset of I. It is easy to verify that $rank_{M|J}(I') - |I'| = rank_{M|J}(I) - |I|$, so we may replace I with I'. Given that I is an independent set and I is a parity set, the expression $2rank_{M|J}(I) - |I|$ is maximized when I is the maximum parity set in $M|J$.

Therefore $rank(Z_J) = 2v_{M|J} + 2m + |J|$. □

Corollary 5. *Z_J is full rank iff J is growable.*

The proof follows from the observation that M has a parity basis containing J if and only if $2v_{M|J} = r - |J|$.

3 An m^ω Algorithm for checking Parity Basis

This is a divide and conquer algorithm that is similar to *5. Algorithm* in *Lecture 12*. The idea here is that we maintain a growable set and keep updating the matrix Z_T. Note that all the updates for the matrix are rank 2 updates, so we can use *Theorem* 1 from *Lecture* 12 due to Sherman-Morrison-Woodbury, to check efficiently, whether or not the update gives a singular matrix.

Let $J' = J \cup 2i - 1, 2i$, i.e, J' contains one more pair than J. Let $N = Z_J^{-1}$. Note that N is skew symetric. We know that the matrix $Z_{J'}$ is singular iff

$$det\left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & t_i \\ -t_i & 0 \end{bmatrix} \begin{bmatrix} 0 & n_i \\ -n_i & 0 \end{bmatrix}\right) = det\left(\begin{bmatrix} 1 + t_i n_i & 0 \\ 0 & 1 + t_i n_i \end{bmatrix}\right) \neq 0$$

Or $(1 + t_i n_i)^2 \neq 0$ or $t_i n_i \neq -1$

We write the algorithm as follows

MatroidParity(M)
1 Construct Z and substitute t_is with numbers randomly chosen from a large enough field
2 $N \leftarrow Z^{-1}$
3 **return** BulidParity(S,N,$\emptyset$);

BuildParity(S,N,J)
1 **if** $|S| = 2$ **then**
2 Let $S = \{2i - 1, 2i\}$
3 **if** $1 + N_{2i-1,2i} t_i \neq 0$ **then**
4 **return** {2i-1,2i}
end
5 **else**
return $\emptyset$
end
end
6 **else**
7 Partition S into two equal size sets S_1 and S_2
8 $J_1 \leftarrow$ BuildParity(S_1,N,J)
9 $N' \leftarrow Z(J \cup J_1)^{-1}_{S_2,S_2}$
10 $J_2 \leftarrow$ Buildparity(S_2,N,$J \cup J_1$)
end
11 **return** $J_1 \cup J_2$;

Algorithm 1: An m^ω algorithm for checking if a matroid has a parity basis

The correctness of the algorithm follows from the correctness of the two invariants we maintain, i.e., J is a growable set, and $N' = Z(J_1 \cup J_2)^{-1}$.

3.1 Running Time

To compute running time, we notice that along with calling the BulidParity routine on S_1 and S_2, we are computing N', which takes time $\mathcal{O}(|S_2|^\omega)$ using *Theorem* 1 from *Lecture* 12. So, may write the following recurrence

$$T(n) = 2T(n/2) + \mathcal{O}(n^\omega)$$

which gives us an $\mathcal{O}(m^\omega)$ time algorithm.

4 Further Improvements

We shall see in the next lecture that even this time can be improved. The rough idea is that the routine *Buildparity* routine can be run on smaller blocks which are obtained by computing the

inverse of $Z(J)$ efficiently and also computing Y efficiently instead of adding m $r \times r$ matrices.

References

[1] Ho Yee Cheung, *Algebraic algorithms in combinatorial optimization*,Master Thesis,The Chinese University of Hong Kong, 2011.