Erratum: Maximum connected matching in bipartite graphs

1 Result

Recall the following definitions and results from [1].

Definition 2.3: For a given graph G, a connected matching in G is a matching M such that every two edges of M are connected by an edge of G. Let $\nu_c(G)$ denote the size of the maximum cardinality of a connected matching in G.

Theorem 3.1: Given a bipartite graph G with n vertices on each side, it is NP-hard to approximate $\nu_c(G)$ within a factor of $n^{1-\varepsilon}$ for any $\varepsilon > 0$ under a randomized polynomial time reduction.

Definition 3.2: Fix $n \in N$. A bipartite graph $HC_n = (A = \{u_1, \ldots, u_n\}, B = \{v_1, \ldots, v_n\}, E_H)$ is said to be a *bipartite half-cover of* K_n if (1) for every $\{i, j\} \subseteq [n]$, $(u_i, v_j) \in E_H$ or $(u_j, v_i) \in E_H$, and (2) for every $i \in [n]$, $(u_i, v_i) \notin E_H$.

Claim 3.3: There is an $O(n)$-time randomized algorithm that on input $n \in N$ outputs a graph HC_n, which is a bipartite half-cover of K_n such that $\nu_c(HC_n) \leq O(\log n)$ with probability $1 - o(1)$.

As remarked in [1] a deterministic polynomial time construction of such graphs would imply that the hardness result in Theorem 3.1 holds under a deterministic reduction (as opposed to the randomized reduction stated). Here we show how to modify the proof of Theorem 3.1 and get that a weaker hardness result holds under deterministic reduction. In particular, this gives that the problem of computing $\nu_c(G)$ for a given input bipartite graph G is APX-hard (and hence also NP-hard). Moreover, it is NP-hard to approximate this quantity up to any constant factor. This is stated in the following result.

Theorem 1.1. *Given a bipartite graph G with n vertices on each side, the problem of computing $\nu_c(G)$ up to any constant factor is NP-hard. In particular the problem is APX-hard.*

2 Proof

We need the fact that it is NP-hard to approximate the clique number of a graph even when this number is linear. This is well known, in particular we state the following result of Hastad [4] (see proof of Theorem 8.1 there).

Theorem 2.1 ([4])**.** *Given an n-vertex graph G, it is NP-hard to distinguish, for any fixed $\varepsilon > 0$, between the case that the size $w(G)$ of the maximum clique of G is at least $\frac{n}{4}(1 - \varepsilon)$ and the case that $w(G)$ is at most $\frac{n}{8}(1 + \varepsilon)$.*

Note that by a simple powering construction this implies that for any constant k it is NP-hard to distinguish between the case that the clique number is at least $\frac{n}{4^k}(1-\varepsilon)^k$ and the case it is at most $\frac{n}{8^k}(1+\varepsilon)^k$.

The main tool in the proof is the following (weak) derandomized version of Claim 3.3 above.

Theorem 2.2. *There is a deterministic polynomial time algorithm that on input $n \in N$ (of the form s^{2^k} for some integers s, k) outputs a graph HC_n, which is a bipartite half-cover of K_n such that*

$$\nu_c(HC_n) \leq \frac{n}{e^{\Omega(\sqrt{\log n})}}.$$

The proof of this theorem is based on two lemmas. The first is an efficient deterministic algorithm for constructing an initial relatively small half cover, and the second is an efficient procedure for squaring a half cover. The desired graph is obtained from the small initial graph by repeated squaring.

Lemma 2.3. *There is a deterministic algorithm that on input $n \in N$ outputs, in time polynomial in n, a graph H, which is a bipartite half-cover of K_m for $m = e^{\sqrt{\log n}}$ such that*

$$\nu_c(H) \leq O(\log m) = O(\sqrt{\log n}).$$

Proof (sketch): Apply the method of conditional expectations to the proof of Claim 3.3 given in [1]. (See, for example, [2], Section 16.1 for a similar argument.) The running time is $m^{O(\log m)} = n^{O(1)}$. $\square$

Lemma 2.4. *There is a deterministic polynomial time algorithm which, given as an input a bipartite half cover F of K_p with $\nu_c(F) \leq \varepsilon p$ where $0 < \varepsilon < 1/4$, outputs a bipartite half cover F' of K_{p^2} satisfying $\nu_c(F') \leq 4\varepsilon p^2$.*

Proof: Let the vertex classes of F be $A = \{a_1, a_2, \ldots, a_p\}$ and $B = \{b_1, b_2, \ldots, b_p\}$ where for every i $a_i b_i$ is not an edge. We construct the graph F' by blowing up H as follows. Replace each vertex a_i by a set U_i of p vertices and each vertex b_j by a set V_j of p vertices, where all these sets are pairwise disjoint. The vertex classes of F' are $U = \cup_i U_i$ and $V = \cup_j V_j$. For each $1 \leq i \leq p$, the bipartite graph between U_i and V_i is a copy of F. For every $1 \leq i \neq j \leq p$, the bipartite graph between U_i and V_j is complete if $a_i b_j$ is an edge of F, otherwise it is edgeless.

It is easy to see that the constructed graph F' is a bipartite half cover of K_{p^2}. In order to complete the proof we show that

$$\nu_c(F') \leq 4\varepsilon p^2. \tag{1}$$

Indeed, let M be a maximum connected matching in F'. Construct an auxiliary graph F" on the classes of vertices A, B by letting $a_i b_j$ be an edge if and only if $i \neq j$ and there is at least one edge of M connecting a vertex of U_i and a vertex of V_j. Any matching in F" can be partitioned into three disjoint matchings, where none of these three matchings saturates both a_i and b_i.

Note that each of these three matchings in F" must be a connected matching in F, and hence its size is at most εp. This shows that the size of the maximum matching in F" is at most $3\varepsilon p$. By König's Theorem this means that F" has a vertex cover of size at most $3\varepsilon p$, implying that all edges of M that do not connect a vertex of U_i with one of V_i (for some i, with the same index i) are covered by the vertices in at most $3\varepsilon p$ of the blocks U_i, V_j. This gives a total of at most $3\varepsilon p^2$ edges of M. In addition, for each fixed i the connected matching M can contain at most εp edges connecting a vertex in U_i and one in V_i, adding at most εp^2 additional edges and establishing (1). This completes the proof of the lemma. $\Box$

Starting with the graph $H_1 = H$ in Lemma 2.3 apply Lemma 2.4 repeatedly k times, where $2^k = \sqrt{\log n}$. In each application $2p$ is the number of vertices of the bipartite graph F to which the lemma is applied, and ε is the ratio between $\nu_c(F)$ and p. This process shows that there is a deterministic algorithm that on input $n \in N$ (of the form s^{2^k} for some integers s, k) outputs, in time polynomial in n, a graph HC, which is a bipartite half-cover of K_n such that

$$\nu_c(HC) \le 4^k \frac{O(\sqrt{\log n})}{e^{\sqrt{\log n}}} n = \frac{n}{e^{\Omega(\sqrt{\log n})}}.$$

This establishes the statement of Theorem 2.2. The assumption that n is of the form s^{2^k} is not essential, as it is possible to remove, if needed, an appropriate number of vertices of the resulting half-cover in each of the applications of the lemma.

The assertion of Theorem 1.1 follows from Theorem 2.2 and the paragraph following it, together with Theorem 2.1, by following the argument given in [1] for proving Theorem 3.1, based on Claim 3.4 in [1]. We omit the details.

Acknowledgment We thank Cyriac Antony and Daniel Paulusma [3] for pointing out a flaw in a previous proof we suggested in [1] for a weaker version of Theorem 1.1. The result here is stronger than the one claimed in [1] as it shows that the problem of computing $\nu_c(G)$ for a given bipartite input graph G is not only NP-hard but is also APX-hard. In fact, even the problem of approximating $\nu_c(G)$ up to any constant factor is NP-hard. A full derandomization of Claim 3.3 may lead to a version of Theorem 3.1 which holds under a deterministic polynomial reduction.

References

[1] N. Alon, J. D. Cohen, P. Manurangsi, D. Reichman, I. Shinkar, T. Wagner and A. Yu, Multitasking capacity: hardness results and improved constructions, SIAM J. Discrete Math. 34 (2020), no. 1, 885–903.

[2] N. Alon and J.H. Spencer, The Probabilistic Method, Fourth Edition, Wiley, 2016.

[3] C. Antony and D. Paulusma, private communication, 2022.

[4] J. Hastad, Some optimal inapproximability results, J. ACM, 48 (2001), 798–859.