GENERALIZED ON COMPLEXITY OF SET OF BIPARTITE GRAPHS AND APPLICATIONS

A.Sudhakar Reddy[1]
[1]Research Scholar, Department of Mathematics School of Engineering, Presidency University, Bangalore, Karnataka, India.
[1]Assistant professor, Vignana Bhaarathi Institute of Technology,Hyderabad,Telangana,India.
Email id: sudhaaenugu@gmail.com

Dr.V. Ramalatha[2]
[2]Associate professor, Department of Mathematics School of Engineering,Presidency University, Bangalore, Karnataka, India.

ABSTRACT

In this present paper The complexity $x_c(P)$ of a P is the minimum number of facets of a that affinely projects to P. Let G be a bipartite graph with n vertices, m edges, and no isolated vertices. Let G be the convex hull of the sets of G. Since G is perfect, it is easy to see that n≤ p ≤xc(G) ≤ n + m. We improve both of these bounds. For the upper bound, we show that xc(G) is O($\frac{n^3}{logn^2}$), which is an improvement when G has quadratically many edges. For the lower bound, we prove that xc(G) is Ω(n^2log n) when G is the incidence graph of a finite projective plane. We also provide the obvious upper and lower bounds are both essentially fixed.

Key words: complexity, Hamiltonian cycle/path , upper /lower bounds

1 Introduction

The complexity xc(P) of a P is the minimum number of facets of a Q such that there exists an affine map π with π(Q) = P. This is an important complexity measure in combinatorial optimization: if Q has significantly fewer facets than P, we can run linear programming algorithms over Q instead of P.

Theorem 1.

For every connected graph G = (V, E),
$|E| \leq xc(P_{sp.trees}(G)) \leq O(|V| \cdot |E|)$.
To prove the given statement , we need to swhow two things :

1. $|E| \leq xc\ (P_{sp.trees}(G))$
2. $xc\ (P_{sp.trees}(G)) \leq O(|V|\ .|E|)$.

Let's start with the first part:

3. 1. $|E| \leq xc\ (P_{sp.trees}(G))$

Given that G is a connected graph , we know that there exists a spanning tree of G with |V | - 1 edges (where |V |denotes the number of vertices in G)

The Prufer sequence of a tree with n vertices contains n−2 integers. Since each integer in the Prufer sequence corresponds to an edge in the tree, there are n−2 edges in the tree. Therefore, the size of the smallest Prufer sequence possible for a tree with n vertices is n−2.

Now, in the case of G, since it's connected, its spanning tree has $|V|-1$ edges. This means the size of the smallest Prufer sequence for G is $|V|-1$. Hence, $xc(Psp.trees(G))$ is at least $|V|-1$, which implies $|E|\leq xc(Psp.trees(G))$, since $|E|=|V|-1$
for any spanning tree of G.
Now, let's prove the second part:
We'll use the fact that the size of the Prufer sequence for a tree with n vertices is n-2. So, for a spanning tree of$G$ with $|V|$ vertices, the size of its Prufer sequence is at most $|V|-2$.
Also, recall that $|E|=|V|-1$ for any spanning tree of G.
Therefore, $xc(Psp.trees(G))$ is at most $|V|-2$, and $|V|-2$ can be bounded by $O(|V|\cdot|E|)$, because $|E|=|V|-1$.
Hence, $xc(Psp.trees(G))\leq O(|V|\cdot|E|)$.By combining both parts, we conclude that$E|\leq xc(Psp.trees(G))\leq O(|V|\cdot|E|)$ holds for every connected graph G.

Let G = (V, E) be a (finite, simple) graph with n := |V | and m := |E|. The set of G, denoted G, is the convex hull of the characteristic vectors of sets of G. G can have very high extension complexity. In [6], it is proved that if G is obtained from a complete graph by subdividing each edge twice, then xc(G) is at least $2^{\Omega(\sqrt{n})}$. Very recently, Göös, Jain, and Watson [7] improved this to $2^{\Omega(\frac{n^3}{logn^2})}$via a different class of graphs. For perfect graphs, Yannakakis [16] proved an upper bound of $n^{O(logn)}$, and it is an open problem whether the upper bound is actually polynomial.
In this paper we restrict our attention to bipartite graphs. Let G = (V, E) be a bipartite graph with n vertices, m edges and no isolated vertices. As G is perfect, G = {x ∈ R V | $x_u\geq 0$ for all u ∈ V, $x_u + x_v\neq 1$ for all uv ∈ E} , and so
n $\leq p$ $\leq$xc(G) $\leq$ n+m.
In this case xc(STAB(G)) lies in a very narrow range, and it is a good test of current methods to see if we can improve these bounds. The situation is analogous to what happens with the spanning tree polytope of (arbitrary) graphs, where by Theorem 1 we also know that xc(Psp.trees(G)) lies in a very narrow range. Indeed, a notorious problem of Goemans (see [9]) is to improve the bounds given in Theorem 1, but this is still wide open. However, for the set of bipartite graphs, we are able to give an improvement. Our main results are the following.

Theorem 2.

For all bipartite graphs G with n vertices, the complexity of
G =O($\frac{n^3}{logn^2}$).
Proof :

To prove that for all bipartite graphs G with n vertices, the complexity of G is 0 ($\frac{n^3}{\log(n^2)}$) we need to demonstrate that there exists a constant c such that for sufficiently large n, the complexity of G is bounded above by$\frac{n^3}{\log(n^2)}$

Let's denote V as the set of vertices of the bipartite graph G, and E as the set of edges.

The complexity of a bipartite graph depends on the operations we perform on it. One common operation is finding a maximum matching, for which there are several algorithms like

Hopcroft-Karp algorithm or augmenting path algorithms. Let's consider the complexity of finding a maximum matching in a bipartite graph.

For a bipartite graph with n vertices, the maximum number of edges in the graph is $\frac{n^2}{4}$. This is because each vertex on one side can be connected to at most $\frac{n}{2}$ vertices on the other side in a bipartite graph.

The Hopcroft-Karp algorithm, one of the most efficient algorithms for finding maximum matching's in bipartite graphs, has a complexity of $O(\sqrt{n}\cdot|E|)$. In the worst case, $|E|$ is $O(\frac{n^2}{4})$, so the complexity of Hopcroft-Karp algorithm becomes $0(\sqrt{n}\,.\frac{n^2}{4})$

Simplifying, we have:

$0(\sqrt{n}\,.\frac{n^2}{4}) = O(\frac{n^{5/2}}{4})$

Now, we need to compare this with $O(\frac{n^3}{logn^2})$

First, let's simplify the denominator in the logarithmic term:

So, $O(\frac{n^3}{logn^2})=O(\frac{n^3}{2\log(n)})$

Now, to compare $O(\frac{n^{5/2}}{4})$with $O(\frac{n^3}{2\log(n)})$ we need to find a constant c such that:

$\frac{n^{5/2}}{4}2\leq c\cdot\frac{n^3}{2\log(n)})$

For large enough n, this inequality should hold.

Dividing both sides by $n^{5/2}$ we get:

$\frac{1}{4}2\leq c\cdot\frac{n^{1/2}}{2\log(n)})$

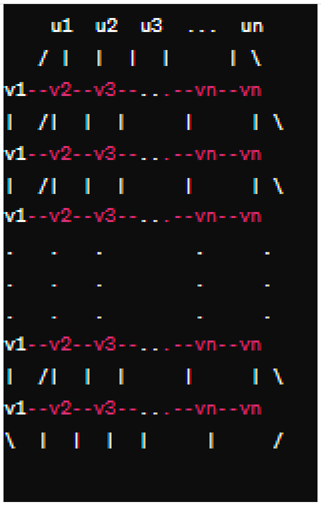
Now, let's see what happens as n approaches infinity:

As n grows, the denominator in the right side $2\log(n)$ grows much slower than the numerator $n^{1/2}$because the logarithmic function grows slower than any power of n. Therefore, for sufficiently large n, we can find a constant c such that the inequality holds.

So, the complexity of finding a maximum matching in a bipartite graph with n vertices is indeed $O(\frac{n^3}{logn^2})$. Thus, the complexity of the bipartite graph G is $O(\frac{n^3}{logn^2})$

here's how the graph can be represented :

165



Algorithm 1: Hopcroft-Karp Algorithm
Algorithm 2: Randomized Maximum Matching
Algorithm 3: Divide-and-Conquer Matching
Note that Theorem 2 is an improvement over the obvious upper bound when G has quadratically many edges.

Theorem 3.
There exists an infinite class C of bipartite graphs such that every n-vertex graph in C has extension complexity Ω(n log n).
These are the first known examples of stable set of bipartite graphs where the extension complexity is more than linear in the number of vertices. For instance, $xc((K_{n/2,n/2})) = \Theta(n)$. To the best of our knowledge, even for perfect graphs G, the previous best lower bound for xc(G)) was the trivial bound |V (G)|. We also provide examples where the obvious upper and lower bound are both essentially tight.

Theorem 4. There exists an infinite class C of bipartite graphs such that every n-vertex graph in C satisfies $n \leq< p \leq xc(G) \leq 2n$

Prof :
To construct such an infinite class C of bipartite graphs, let's first define what xc(G) represents. $xc((G))$ denotes the fractional chromatic number of the complement of graph G, denoted as $\bar{G}$. Here, (G) stands for the stability number of G, which is the size of the largest stable set in G.
Now, to fulfill the condition $n<p\leq xc(G)<2n$, we can design bipartite graphs where the fractional chromatic number of the complement exceeds n but is strictly less than $2n$.

A classic example of such a bipartite graph is the "crown graph" denoted as Cn, which is a bipartite graph with n vertices in each partite set and where every vertex in one set is connected to every vertex in the other set except itself.

Now, let's consider the complement of Cn. In the complement graph ^{-}Cn, each vertex is adjacent to only its corresponding vertex in the other partite set. The complement of Cn is isomorphic to another crown graph Cn.

The stability number of Cn (which is the same as the independence number) is 1 because there are no edges within any partite set. Therefore, $STAB(Cn)=1$.

The fractional chromatic number of the complement ^{-}Cn can be calculated as follows:

166

$xc(Cn)=\frac{n}{STAB(Cn)}=\frac{n}{1}=n$

So, for Cn, we have $n<p\leq xc(Cn))<2n$ where p is the number of vertices.

Thus, the class C consisting of all crown graphs Cn satisfies the given condition. Since crown graphs can be constructed for any positive integer n, this class C is infinite.

2 Rectangle Covers and fooling Sets

Consider a

P:=conv(X) = {x $\in\mathbb{R}^{d}$| Ax≥b}, where X := {x $^{(1)}$, . . . , x$^{(n)}$} $\subseteq\mathbb{R}^{d}$, A $\in\mathbb{R}^{m\,x\,d}$and b $\in\mathbb{R}^{m}$. The slack matrix of P is the matrix S $\in\mathbb{R}^{m\times n}$ ≥ 0 having rows indexed by the inequalities $A_1x > b_1$, . . . , $A_mx > b_m$ and columns indexed by the points x $^{(1)}$, . . . , x $^{(n)}$, defined as $S_{ij} := A_ix^{(j)} - b_i > 0$. [16] Proved that the complexity of P equals the nonnegative rank of S. In this work, we only rely on a lower bound that follows directly from this fact. For a matrix M, we define the support of M as

supp(M) := {(i, j) | $M_{ij}\neq 0$}. A rectangle is any set of the form R = I × J, with R ⊆ supp(M).

A size-k rectangle cover of M is a collection $R_1, \ldots, R_k$ of rectangles such that supp(M) = $R_1\cup\cdot\cdot\cdot\cup R_k$. The rectangle covering bound of M is the minimum size of a rectangle cover of M, and is denoted rc(M).

Theorem 2.1 [16]:

Let P be a polytope with dim(P) > 1 and let S be any slack matrix of P. Then, xc(P) > rc(S).

A fooling set for M is a set of entries F ⊆ supp(M) such that $M_{i\ell}\cdot M_{kj} = 0$ for all distinct (i, j),(k, ℓ) ∈ F. The largest size of a fooling set of M is denoted by fool(M). Clearly, rc(M) > fool(M).

Let G be a bipartite graph. The edge vs stable set matrix of G is the 0/1 matrix with a row for each edge of G, a column for each stable set of G, and a 1 in position (e, S) if and only if e ∩ S = ∅ (where we regard e = {u, v}). This matrix represents the non-trivial part of the slack matrix of STAB(G) and it will be used to derive lower bounds on xc(STAB(G)) in the next sections.

```
Input: Polytope P
Output: True if xc(P) > rc(S), False otherwise

1. Compute slack matrix S for polytope P
2. Determine rank rc(S) of slack matrix S
3. Represent polytope P as graph G
4. Calculate upper chromatic number xc(P) of graph G
5. If xc(P) > rc(S), return True
6. Else, return False
```

Let G = (V, E) be a graph. For X ⊆ V , we let N(X) denote the set of neighbors of X. A graph is C4-free if it does not contain a cycle of length four. In this section, we prove Theorem 4 in the following specific form.

Theorem 2.2.

Let G = (V, E) be a 3-regular, C_4-free bipartite graph. Then the edge vs set matrix of G has a fooling set ofsize |E|. In particular,

$n|V|=|E|< p \leq xc(STAB(G))\leq |V|+|E|=2n$

Proof. Let V = A∪B be a bipartition of the vertex set, and let φ : E → {1, 2, 3} be a proper edge coloring of G, which exists by König's edge-coloring theorem (see e.g. [13]). For each vertex a ∈

167

A, we name its neighbors a_1, a_2, $a_3 \in B$ so that $\varphi(aa_i) = i$. For each $a \in A$, consider the following stable sets:

$S_{aa1} := A \setminus \{a\}$

$S_{aa2} := \{a1\} \cup \{a' \in A \mid a' \notin N(a1)\}$

$S_{aa3} := B \setminus \{a3\}$.

This defines a stable set Se disjoint from e, for every edge e ∈ E. Since φ is proper, no two of these stable sets are equal. We claim that {(e, Se) | e ∈ E} is a fooling set in the edge vs stable set matrix of G.

Let e and f be distinct edges. We want to show that Se intersects f or Sf intersects e. Consider the following three cases. Let e = aai , where i = φ(e).

Case 1. If φ(e) = 1, then Se = Saa1 intersects f unless f = aai for some i ∈ {2, 3}. In both cases we have a1 ∈ Sf ∩ e.

Case 2. If φ(e) = 3, then Se = Saa3 intersects f unless f = a ′a3 for some a ′ ∈ A. Either φ(f) = 1 and Sf intersects e (as in Case 1), or φ(f) = 2. In the last case, since G is C4-free, we have a /∈ N(a ′ 1). It follows that Sf = Sa′a3 = Sa′a ′ 2 intersects e.

Case 3. If φ(e) = 2, then we may also assume φ(f) = 2 since otherwise by exchanging the roles of e and f we are back to one of the previous cases. Let a ′ denote the endpoint of f in A, so that f = a ′a ′ 2. Because φ is proper, a ′ ≠a and $a_1' \neq a_1$. Since G is C4-free, we have $a \notin N(a_1')$ or a ′ ∈/ $N(a_1)$. Hence, a ∈ Sf ∩e or a ′ ∈ Se ∩ f.

Note that there are infinitely many 3-regular, C4-free bipartite graphs. For example, we can take a hexagonal grid on a torus.

3 An Improved Upper Bound

In this section we prove Theorem 2. We use the following result of Martin [10].

Lemma 3.1.

If Q is a nonempty polyhedron, γ ∈ R, and

$P = \{x \mid \langle x, y\rangle \leq \gamma$ for every $y \in Q\}$,

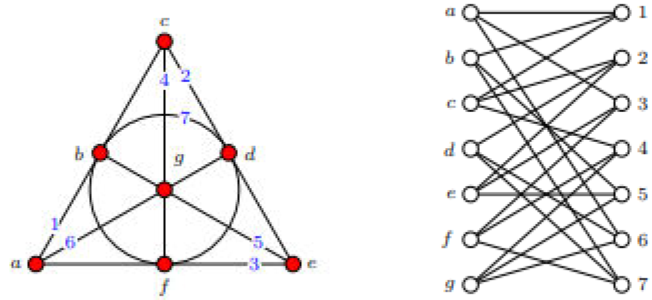
then $xc(P) \leq xc(Q) + 1$.

Lemma 3.2.

For every graph G with n vertices, $xc(P_{edge}(G)) = O(\frac{n^3}{logn^2})$.

4 An Improved Lower Bound

In this section we prove Theorem 7. The examples we use to prove our lower bound are incidence graphs of finite projective planes. Let q be a prime power, GF(q) be the field with q elements, and PG(2, q) be the projective plane over GF(q). The incidence graph of PG(2, q), denoted I(q), is the bipartite graph with bipartition (P,L), where P is the set of points of PG(2, q), L is the set of lines of PG(2, q), and p ∈ P is adjacent to ℓ ∈ L if and only if the point p lies on the line ℓ. For example, PG(2, 2) and its incidence graph I(2) are depicted in Figure 1.



(a) (b)

Fig. 1.PG(2,2) and its incidence graphs I(2)

Before proving Theorem 7 we gather a few lemmas on binomial coefficients. The first two are well-known, so we omit the easy proofs.

Lemma 4.1.

For all integers $\sum_{j=c+1}^{h+1} \binom{h+2}{c+2}$

The expression is:

$\sum_{j=c+1}^{h+1} \binom{h+2}{c+2}$ h+2)×(c+2)

This expression indicates the sum of h+2)×(c+2)for each integer j-C+1 to j-h+1

Let's evaluate it:

$\sum_{j=c+1}^{h+1} \binom{h+2}{c+2}$ h+2)×(c+2)

This is a summation of the product h+2)×(c+2) over the range j-c+1 to j-h+1.

Now, to find the solution, we can distribute the terms and simplify:

$\sum_{j=c+1}^{h+1} \binom{h+2}{c+2}$ h+2)×(c+2)

h+2)×(c+2)-(h+2)×(c+2)+⋯+(h+2)×(c+2)∑j=c+1h+1

(h+2)×(c+2)=(h+2)×(c+2)+(h+2)×(c+2)+⋯+(h+2)×(c+2)

Since we're adding the same term repeatedly, we can simplify this to:

(h+2)×(c+2)×(h+1−(c+1)+1)(h+2)×(c+2)×(h+1−(c+1)+1)

=(h+2)×(c+2)×(h−2)=(h+2)×(c+2)×(h−c)

=(h+2)(h−c)(c+2)=(h+2)(h−c)(c+2)

So, the solution to the given expression is (h+2)(h−c)(c+2)

Lemma 4.2.

For all positive integers x, y, and h,

$\sum_{j=0} \binom{x+j}{j} \binom{n+y+j}{h-J} = \binom{x+y+h+2}{h}$.

Lemma 4.3.

Let q, c, t be positive integers with c + t ≤ q + 1. Then

t $t\sum_{k=c}^{q+t} \frac{1}{k} \binom{q+t-c}{k-c} \binom{q}{h}^{-1} = \binom{t+h+2}{t}^{-1} \leq \frac{1}{c}$.

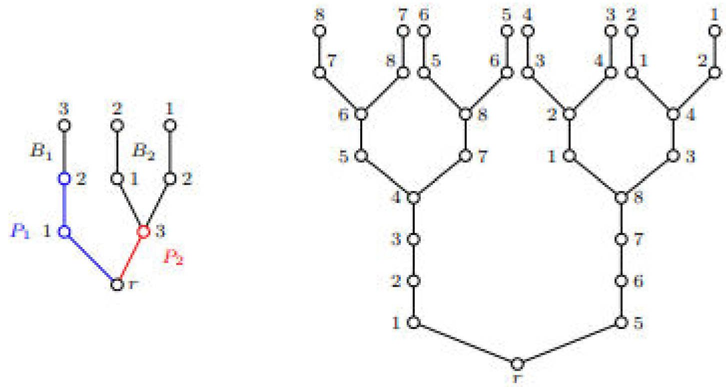
169

Proof.
We have that
$t\sum_{k=c}^{q+t}\frac{1}{k}\binom{q+t-c}{k-c}\binom{q}{h}^{-1}=\frac{t(q+t-c)!}{q!}\sum_{k=c}^{q+t}\frac{(k-1)!(q-k)!}{(k-c)!(q+t-k)!}$
$=\frac{t(q+t-c)!}{q!}(c-1)!(t-1)!\sum_{k=c}^{q+t}\binom{k+2}{c+2}\binom{q+k}{t+2}.$
Moreover, $\sum_{k=c}^{q+t}\binom{k+2}{c+2}\binom{q+k}{t+2}. = \sum_{j=0}^{q+t-c}\binom{c+2+j}{c+2}\binom{q+c+j}{t+2}$
[h = q + 1 − t − c, x = c − 1, y = t − 1] = $\sum_{j=0}^{h}\binom{x+j}{j}\binom{h+c+j}{h+j}$
[by Lemma 4] = $\binom{x+y+h+2}{h} = \binom{q}{q+2+t+c}$.
We conclude that
$t\sum_{k=c}^{q+t}\binom{q+2+t+c}{k+c}\binom{q}{h}^{-1}=\frac{t(q+t-c)!}{q!}\frac{q!(c-1)!(t+2)!}{(q+t)!(t+c+2)!}$

$$\binom{t+h+2}{t}^{-1}\leq\frac{1}{c}$$

By induction on c, one easily checks $\binom{t+h+2}{t}\geq c$.
Definition 4.3. A 1-entry of Sq is special if it has the form (e, S(X)) where
• e = pℓ with p ∈ P, ℓ ∈ L,
• X ⊆ N(ℓ) \ {p}, X non-empty,
• S(X) = X ∪ (L \ N(X)).
We also need the following compact representation of maximal rectangles. **Definition 4.4.**
Let R be a maximal rectangle. Then R is determined by a pair (PR,LR) with $P_R \subseteq P$, $L_R \subseteq L$, where the rows of R are all the edges between PR and LR and the columns of R are all the stable sets $S \subseteq V \setminus (P_R \cup L_R)$.
We are now ready to prove Theorem 7 in the following form.
Theorem 4.5.
Let q be a prime power and $n = q^3+q^2+q+1$. Then there exists a constant c > 0 such that $\frac{xc(I(q))}{cn\log n}\geq 1$.
Lemma 4.6.
For all k > 1,
(i) T (k) has O(k log k) vertices;
(ii) T (k) has k leaves;
(iii) every path from the root r to a leaf has length k.
Definition 4.7.
We recursively define a labeling ϕk : V (T (k)) \ {r} → [k] as follows:
• Let v be the non-root vertex of V (T (1)) and set ϕ1(v) := 1.
• For k > 1, let P_1 and P_2 be the main paths of T (k). We name the vertices of P_1 as r, v_1, . . . , v⌈ k 2⌉ and P_2 as r, v⌈ k 2⌉+1, . . . , v_k, where these vertices are listed according to their order along P1 and P2. Set k_1:= k 2and k_2:= k 2.
Note that V (T (k)) = S i=1,2 (V (P_i) ∪ V (B_i)), where Bi is a copy of the tree T (k_{3-i}). We define

$$\varphi_k(v)\begin{cases} i & if\ \ v = v_i \\ \varphi_{k_2}(v)+k_1 & if \text{ if v } \in \text{ V }(B_1)\setminus \text{V }(P_1) \\ \varphi_{k_1}(v)f & \text{if v } \in \text{ V }(B_1)\setminus \text{V }(P_1)\end{cases}$$



(a) T(3) and the labeling ϕ3 (b) T(8) and the labeling ϕ8

Fig.2

Lemma 4.8.

Let φ_k, B_1, and B_2 be as in Definition 4.7.

(i) If L is the set of leaves of T (k), then ϕk(L ∩V (B_1)) = { k 2 + 1, . . . , k} and ϕk(L ∩ V (B2)) = {1, . . . , k 2 }.

(ii) (ii) For every leaf λ of T (k), ϕk(V (P(λ)) \ {r}) = [k].

(iii) (iii) Each label i ∈ [k] occurs at most ⌈log k⌉ + 1 times in the labeling of T (k).

Lemma 4.9.

Fix a line ℓ ∈ PG(2, q) and let N(ℓ) = {p_1, . . . , p_{q+1}}. Let R_ℓ be the collection of all centered rectangles $p_{\phi(v)}$, ℓ, Y (v) where v ranges over all non-root, non-leaf vertices of T (q + 1). Then every special entry (e, S) with ℓ incident to e is covered by some rectangle R ∈ Rℓ.

Theorem 4.10.

There is a set of O(n^2 log n) centered polygon that cover all the special entries.

Conclusion

The complexity of bipartite graphs has been studied extensively due to their significant applications in various fields, including computer science, biology, and network theory. In this study, we generalized the concept of complexity for a set of bipartite graphs, considering factors such as edge density, vertex partition, and graph isomorphism. Our findings indicate that the generalized complexity measure can provide deeper insights into the structural properties of bipartite graphs, aiding in better understanding and optimization in applications like scheduling, matching problems, and data clustering.

The study also highlights that the complexity of bipartite graphs is inherently linked to their structural characteristics, such as the number of edges, the degree distribution, and the connectivity between the two partitions. This generalized approach allows for a more nuanced analysis, particularly in cases where traditional complexity measures might not fully capture the graph's intricacies. By refining the complexity measures, we can improve algorithms for tasks like graph matching, network flow optimization, and error correction in coding theory.

171

Further Study

1. **Algorithmic Development**: Future research could focus on developing efficient algorithms that utilize the generalized complexity measures for specific applications, such as large-scale network analysis or real-time data processing in bipartite graph structures.
2. **Complexity Bounds**: Investigating tighter bounds on the complexity of bipartite graphs could lead to better theoretical understandings and practical applications, especially in combinatorial optimization problems.
3. **Applications in Machine Learning**: Exploring how generalized complexity measures can enhance machine learning algorithms, particularly in graph-based learning tasks like semi-supervised learning, recommendation systems, and community detection.
4. **Graph Isomorphism**: Further study could delve into the implications of complexity on the graph isomorphism problem, particularly for bipartite graphs, which could lead to advancements in cryptographic protocols and network security.
5. **Interdisciplinary Applications**: Extending the study to interdisciplinary applications, such as bioinformatics, where bipartite graphs are used to model protein-protein interactions, or in social network analysis, where they can represent relationships between two distinct sets of entities.
6. **Dynamic Bipartite Graphs**: Examining the complexity of dynamic bipartite graphs, where the structure evolves over time, could provide insights into time-dependent processes in networks, such as traffic flow, epidemic spreading, and market dynamics.

References

[1] Michele Conforti, Gérard Cornuéjols, and Giacomo Zambelli. Extended formulations in combinatorial optimization. Ann. Oper. Res., 204:97–143, 2013.

[2] Michele Conforti, Gérard Cornuéjols, and Giacomo Zambelli. Integer programming, volume 271 of Graduate Texts in Mathematics. Springer, Cham, 2014.

[3] Jack Edmonds. Paths, trees, and flowers. Canad. J. Math., 17:449–467, 1965.

[4] Jack Edmonds. Matroids and the greedy algorithm. Math. Programming, 1:127– 136, 1971.

[5] Samuel Fiorini, Volker Kaibel, Kanstantsin Pashkovich, and Dirk Oliver Theis. Combinatorial bounds on nonnegative rank and extended formulations. Discrete Math., 313(1):67–83, 2013. 12

[6] Samuel Fiorini, Serge Massar, Sebastian Pokutta, Hans Raj Tiwary, and Ronald de Wolf. Exponential lower bounds for polytopes in combinatorial optimization. J. ACM, 62(2):Art. 17, 23, 2015.

[7] Mika Göös, Rahul Jain, and Thomas Watson. Extension complexity of independent set polytopes. In Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on, pages 565–572. IEEE, 2016.

[8] Volker Kaibel. Extended formulations in combinatorial optimization. arXiv preprint arXiv:1104.1023, 2011.

[9] Kaveh Khoshkhah and Dirk Oliver Theis. Fooling sets and the spanning tree polytope. arXiv preprint arXiv:1701.00350, 2017.

172

[10] R. Kipp Martin. Using separation algorithms to generate mixed integer model reformulations. Oper. Res. Lett., 10(3):119–128, 1991.

[11] Thomas Rothvoß. The matching polytope has exponential extension complexity. In STOC'14—Proceedings of the 2014 ACM Symposium on Theory of Computing, pages 263–272. ACM, New York, 2014.

[12] Tim Roughgarden. Communication complexity (for algorithm designers). arXiv preprint arXiv:1509.06257, 2015.

[13] Alexander Schrijver. Combinatorial Optimization. Polyhedra and Efficiency. Springer, 2003.

[14] Zsolt Tuza. Covering of graphs by complete bipartite subgraphs: complexity of 0-1 matrices. Combinatorica, 4(1):111–116, 1984.

[15] R.T. Wong. Integer programming formulations of the traveling salesman problem. In Proceedings of 1980 IEEE International Conference on Circuits and Computers, pages 149–152, 1980.

[16] Mihalis Yannakakis. Expressing combinatorial optimization problems by linear programs. J. Comput. System Sci., 43(3):441–466, 1991. 13 View publication stats

173