Problems and Results Motivated by Efficient Computation of the Independence Number

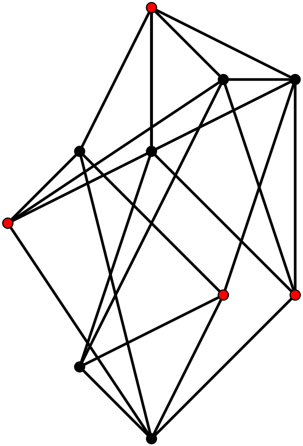
Craig Larson

Virginia Commonwealth University
Richmond, VA

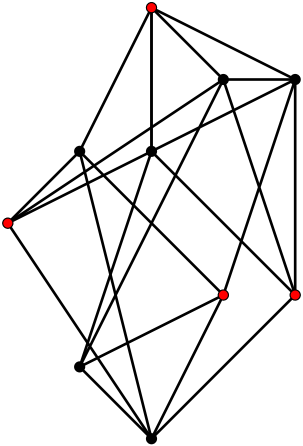
The University of Gent
Feb. 22, 2013

I. Introduction, Applications, Complexity

The Independence Number of a Graph

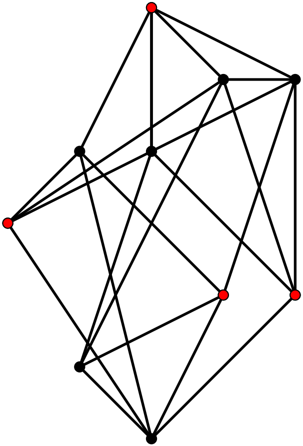


The Independence Number of a Graph



• The independence number α of a graph is the largest number of mutually non-adjacent vertices.

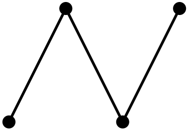
The Independence Number of a Graph



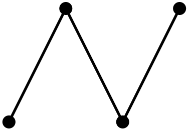
• The independence number α of a graph is the largest number of mutually non-adjacent vertices.

$$\alpha = 4.$$

Independent Sets and Chemical Properties



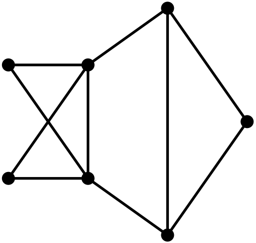
Independent Sets and Chemical Properties



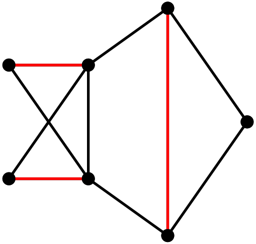
- The number of independent sets σ correlates both with alkane heats of formation and boiling points.

• R. Merrifield and H. Simmons, The Structure of Molecular Topological Spaces, *Theoretica Chimica Acta*, 1980.

The matching number of a graph

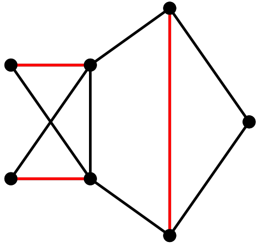


The matching number of a graph



Let M=red.

The matching number of a graph

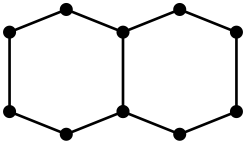


Let M=red.

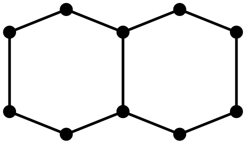
M is a maximum matching,

and $\mu = 3$.

Molecular Stability

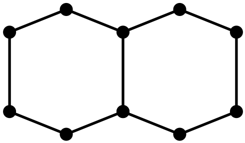


Molecular Stability



- Stable benzenoids maximize their matching number

Molecular Stability



- Stable benzenoids maximize their matching number
- and minimize their independence number.

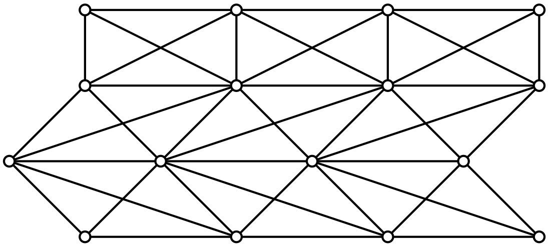
• R. Pepper, An upper bound on the independence number of benzenoid systems, *Discrete Applied Mathematics*, 2008.

Which Fullerene Isomers are Stable?

Atoms	Isomer	# of Isomers	α	Rank	Max	Min
60	C_{60}:1 (I_h)	1812	24	1	28	24
70	C_{70}:1 (D_{5h})	8149	29	1	33	29
76	C_{76}:1 (D_2)	19151	32	1	36	32
78	C_{78}:1 (D_3)	24109	33	1 (3)	37	33
	C_{78}:3 (C_{2v})		34	2		
	C_{78}:2 (C_{2v})		33	1 (3)		
84	C_{84}:22 (D_2)	51592	36	1 (17)	40	36
	C_{84}:23 (D_{2d})		36	1 (17)		

- S. Fajtlowicz, and C. E. Larson, Graph-theoretic Independence as a Predictor of Fullerene Stability, *Chemical Physics Letters*, 2003.

Shannon Capacity



- The zero-error capacity of a alphabet is lim $\sqrt[n]{\alpha(G^n)}$.

• C. Shannon, The zero error capacity of a noisy channel, *IRE Transactions on Information Theory*, 1956.

Optimal Communication Networks

• G. Brinkmann, S. Crevals, J. Frye, An independent set approach for the communication network of the GPS III system, *Discrete Applied Mathematics*, 2013.

Relations to Other Graph Invariants

Relations to Other Graph Invariants

- Domination number: $\gamma \leq \alpha$

Relations to Other Graph Invariants

- Domination number: $\gamma \leq \alpha$
- Clique Covering number: $\alpha \leq \bar{\omega}$

Relations to Other Graph Invariants

- Domination number: $\gamma \leq \alpha$
- Clique Covering number: $\alpha \leq \bar{\omega}$
- Chromatic number: $\alpha\chi \geq n$

Relations to Other Graph Invariants

- Domination number: $\gamma \leq \alpha$
- Clique Covering number: $\alpha \leq \bar{\omega}$
- Chromatic number: $\alpha\chi \geq n$
- Matching number: $n - 2\mu \leq \alpha \leq n - \mu$.

Relations to Other Graph Invariants

- Domination number: $\gamma \leq \alpha$
- Clique Covering number: $\alpha \leq \bar{\omega}$
- Chromatic number: $\alpha\chi \geq n$
- Matching number: $n - 2\mu \leq \alpha \leq n - \mu$.
- Clique Number: $\alpha(G) = \omega(\bar{G})$.

Relations to Other Graph Invariants

- Domination number: $\gamma \leq \alpha$
- Clique Covering number: $\alpha \leq \bar{\omega}$
- Chromatic number: $\alpha\chi \geq n$
- Matching number: $n - 2\mu \leq \alpha \leq n - \mu$.
- Clique Number: $\alpha(G) = \omega(\bar{G})$.
- Covering Number: $\alpha = n - \tau$.

Calculating the independence number of a graph

Calculating the independence number of a graph

- If a graph has n vertices, there are 2^n subsets to consider.

Calculating the independence number of a graph

- If a graph has n vertices, there are 2^n subsets to consider.
- Robson's algorithm is the fastest existing analyzed algorithm and runs in $\mathcal{O}(2^{.276n})$.

Calculating the independence number of a graph

- If a graph has n vertices, there are 2^n subsets to consider.
- Robson's algorithm is the fastest existing analyzed algorithm and runs in $\mathcal{O}(2^{.276n})$.
- Östergård's Cliquer algorithm is a simple, fast and popular practical general algorithm.

Calculating the independence number of a graph

- If a graph has n vertices, there are 2^n subsets to consider.
- Robson's algorithm is the fastest existing analyzed algorithm and runs in $\mathcal{O}(2^{.276n})$.
- Östergård's Cliquer algorithm is a simple, fast and popular practical general algorithm.
- New general algorithms may be faster - San Segundo's BBMC.

• J. M. Robson, Algorithms for Maximum Independent Sets, Journal of Algorithms 7 (1986) 425–440.
• P. Östergård, A fast algorithm for the maximum clique problem, Discrete Applied Mathematics 120 (2002) 197–207.
• P. San Segundo, An improved bit parallel exact maximum clique algorithm, *Optimization Letters*, 2011.

Independence number is NP-hard

The Independent Set Decision Problem:

Given a graph G and an integer k, does G have an independent set of size at least k?

Independence number is NP-hard

The Independent Set Decision Problem:

Given a graph G and an integer k, does G have an independent set of size at least k?

• R. M. Karp, Reducibility Among Combinatorial Problems, Complexity of Computer Computations, 1972, 85–103.
• M. Garey and D. Johnson, Computers and Intractability, W. H. Freeman and Company, New York, 1979.

Does P=NP?

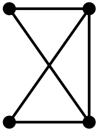
Does P=NP?

"My hunch is that P=NP, contrary to general belief."

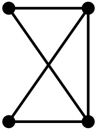
• B. Bollobás, The Future of Graph Theory, Quo Vadis, Graph Theory?, 1993, 5–11 .

II. A Structural Result

A König-Egervary graph (or KE graph) is a graph where $\alpha + \mu = n$.

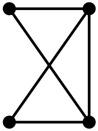


A König-Egervary graph (or KE graph) is a graph where $\alpha + \mu = n$.



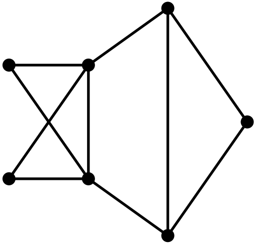
- $\alpha = 2$, $\mu = 2$, $n = 4$.

A König-Egervary graph (or KE graph) is a graph where $\alpha + \mu = n$.

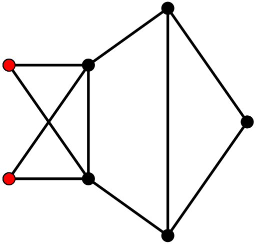


- $\alpha = 2$, $\mu = 2$, $n = 4$.
- $\alpha + \mu = n$.

The critical difference d is the maximum value of $|I| - |N(I)|$, for all independent sets I. An independent set I_c which realizes d is a critical independent set.

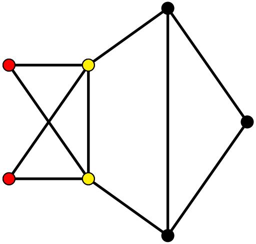


The critical difference d is the maximum value of $|I| - |N(I)|$, for all independent sets I. An independent set I_c which realizes d is a critical independent set.



Let I_c=red vertices,

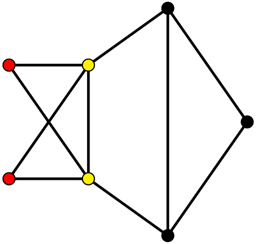
The critical difference d is the maximum value of $|I| - |N(I)|$, for all independent sets I. An independent set I_c which realizes d is a critical independent set.



Let I_c=red vertices,

then $N(I_c)$=yellow vertices,

The critical difference d is the maximum value of $|I| - |N(I)|$, for all independent sets I. An independent set I_c which realizes d is a critical independent set.

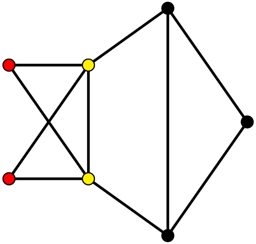


Let I_c=red vertices,

then $N(I_c)$=yellow vertices,

and $|I_c| - |N(I_c)| = 0$.

The critical difference d is the maximum value of $|I| - |N(I)|$, for all independent sets I. An independent set I_c which realizes d is a critical independent set.



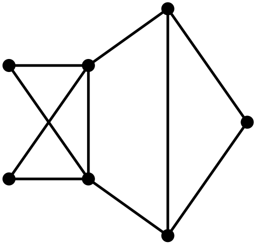
Let I_c=red vertices,

then $N(I_c)$=yellow vertices,

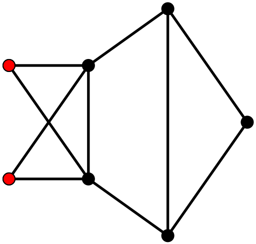
and $|I_c| - |N(I_c)| = 0$.

$d = 0$ and I_c is a critical independent set.

A maximum critical independent set is an independent set which realizes the critical difference d and has maximum cardinality.

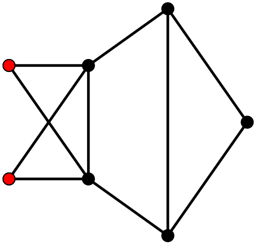


A maximum critical independent set is an independent set which realizes the critical difference d and has maximum cardinality.



Let I_c=red vertices,

A maximum critical independent set is an independent set which realizes the critical difference d and has maximum cardinality.



Let I_c=red vertices,

I_c is a maximum cardinality critical independent set.

An Independence Decomposition

Theorem: For any graph G, there is a unique set $X \subseteq V(G)$ such that

An Independence Decomposition

Theorem: For any graph G, there is a unique set $X \subseteq V(G)$ such that

1. $\alpha(G) = \alpha(G[X]) + \alpha(G[X^c])$,

An Independence Decomposition

Theorem: For any graph G, there is a unique set $X \subseteq V(G)$ such that

1. $\alpha(G) = \alpha(G[X]) + \alpha(G[X^c])$,
2. $G[X]$ is KE,

An Independence Decomposition

Theorem: For any graph G, there is a unique set $X \subseteq V(G)$ such that

1. $\alpha(G) = \alpha(G[X]) + \alpha(G[X^c])$,
2. $G[X]$ is KE,
3. $G[X^c]$ has the property that every non-empty independent set I has more than $|I|$ neighbors, and

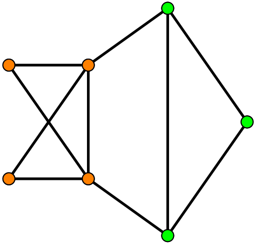
An Independence Decomposition

Theorem: For any graph G, there is a unique set $X \subseteq V(G)$ such that

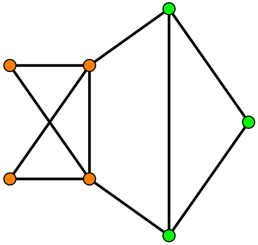
1. $\alpha(G) = \alpha(G[X]) + \alpha(G[X^c])$,
2. $G[X]$ is KE,
3. $G[X^c]$ has the property that every non-empty independent set I has more than $|I|$ neighbors, and
4. for every maximum critical independent set J_c of G, $X = J_c \cup N(J_c)$.

• L., The Critical Independence Number and an Independence Decomposition, *European Journal of Combinatorics*, 2011.

An Independence Decomposition

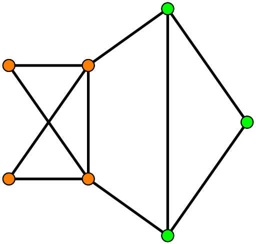


An Independence Decomposition



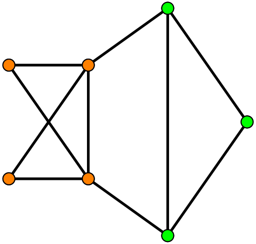
- X is orange, X^c is green,

An Independence Decomposition



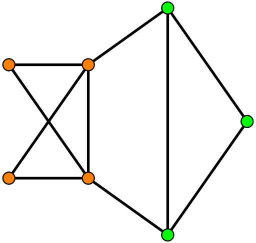
- X is orange, X^c is green,
- $G[X]$ is KE, and

An Independence Decomposition

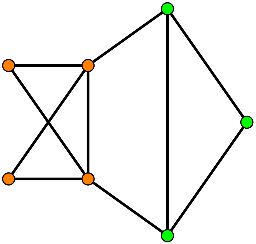


- X is orange, X^c is green,
- $G[X]$ is KE, and
- $G[X^c]$ has every non-empty independent set I has more than $|I|$ neighbors.

An Independence Decomposition

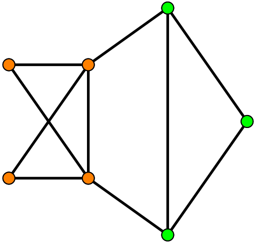


An Independence Decomposition



- $\alpha(G) = \alpha(G[X]) + \alpha(G[X^c]) = 3.$

An Independence Decomposition



- $\alpha(G) = \alpha(G[X]) + \alpha(G[X^c]) = 3$.
- Every graph decomposes into a KE graph and a graph where every independent set I has more than $|I|$ neighbors.

Integer Programming (VPIP)

For every v_i in $V = \{v_1, v_2, \ldots, v_n\}$, let $w(v_i) \in \{0, 1\}$,

Integer Programming (VPIP)

For every v_i in $V = \{v_1, v_2, \ldots, v_n\}$, let $w(v_i) \in \{0, 1\}$,

Constraints: $w(v_i) + w(v_j) \leq 1$ if v_i is adjacent to v_j

Integer Programming (VPIP)

For every v_i in $V = \{v_1, v_2, \ldots, v_n\}$, let $w(v_i) \in \{0, 1\}$,

Constraints: $w(v_i) + w(v_j) \leq 1$ if v_i is adjacent to v_j

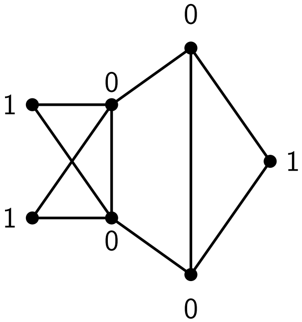
Objective: maximize $\sum w(v_i)$.

Integer Programming (VPIP)

For every v_i in $V = \{v_1, v_2, \ldots, v_n\}$, let $w(v_i) \in \{0, 1\}$,

Constraints: $w(v_i) + w(v_j) \leq 1$ if v_i is adjacent to v_j

Objective: maximize $\sum w(v_i)$.

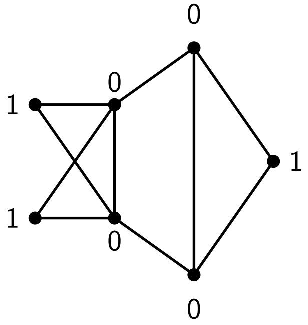


Integer Programming (VPIP)

For every v_i in $V = \{v_1, v_2, \ldots, v_n\}$, let $w(v_i) \in \{0, 1\}$,

Constraints: $w(v_i) + w(v_j) \leq 1$ if v_i is adjacent to v_j

Objective: maximize $\sum w(v_i)$.



$$\alpha = \max \sum w(v_i).$$

Linear Programming (VPLP)

For every v_i in $V = \{v_1, v_2, \ldots, v_n\}$, let $w(v_i) \in [0, 1]$,

Linear Programming (VPLP)

For every v_i in $V = \{v_1, v_2, \ldots, v_n\}$, let $w(v_i) \in [0, 1]$,

Constraints: $w(v_i) + w(v_j) \leq 1$ if v_i is adjacent to v_j.

Linear Programming (VPLP)

For every v_i in $V = \{v_1, v_2, \ldots, v_n\}$, let $w(v_i) \in [0, 1]$,

Constraints: $w(v_i) + w(v_j) \leq 1$ if v_i is adjacent to v_j.

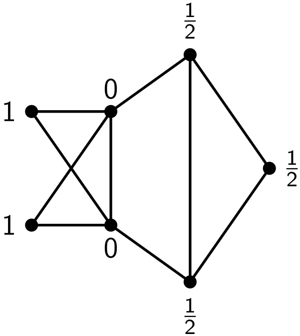
Objective: maximize $\sum w(v_i)$.

Linear Programming (VPLP)

For every v_i in $V = \{v_1, v_2, \ldots, v_n\}$, let $w(v_i) \in [0, 1]$,

Constraints: $w(v_i) + w(v_j) \leq 1$ if v_i is adjacent to v_j.

Objective: maximize $\sum w(v_i)$.

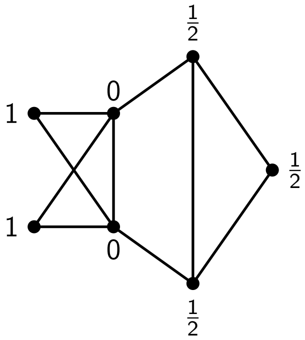


Linear Programming (VPLP)

For every v_i in $V = \{v_1, v_2, \ldots, v_n\}$, let $w(v_i) \in [0, 1]$,

Constraints: $w(v_i) + w(v_j) \le 1$ if v_i is adjacent to v_j.

Objective: maximize $\sum w(v_i)$.

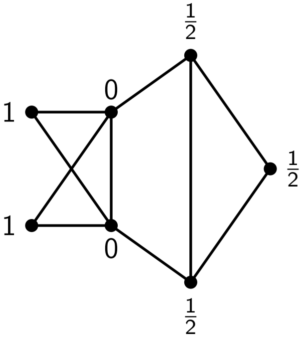


$$\alpha \le \max \sum w(v_i).$$

A Useful Result

Theorem

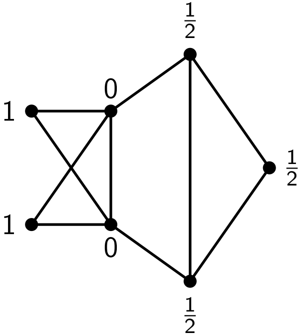
(Balinsky, 1965) There is an optimal solution to VPLP with weights $w(v_i) \in \{0, 1, \frac{1}{2}\}$.



A Useful Result

Theorem

(Balinsky, 1965) There is an optimal solution to VPLP with weights $w(v_i) \in \{0, 1, \frac{1}{2}\}$.



$$\max \sum w(v_i) = 3.5.$$

• G. L. Nemhauser and L. E. Trotter, "Properties of vertex packing and independence system polyhedra," in *Mathematical Programming*, 1974.

Picard-Queyranne Theorem

Theorem

(Picard, Queyranne) There are a unique maximal set of variables which are integral in optimal VPLP solutions.

Picard-Queyranne Theorem

Theorem

(Picard, Queyranne) There are a unique maximal set of variables which are integral in optimal VPLP solutions.

Theorem

(Picard, Queyranne) If V_0, V_1, $V_{\frac{1}{2}}$ and V'_0, V'_1, $V'_{\frac{1}{2}}$ are optimal solutions with a maximum number of integral variables, then $V_0 \cup V_1 = V'_0 \cup V'_1$.

• J-C. Picard, M. Queyranne, "On the Integer-Valued Variables in the Linear Vertex Packing Problem", *Mathematical Programming*, 1977.

Facts

For a optimal solution $V_0, V_1, V_{\frac{1}{2}}$ of VPLP, and a critical independent set I_c, . . .

Facts

For a optimal solution $V_0, V_1, V_{\frac{1}{2}}$ of VPLP, and a critical independent set I_c, ...

1. $|V_0| + |V_1| + |V_{\frac{1}{2}}| = |I_c| + |N(I_c)| + |(I_c \cup N(I_c))^c|$.

Facts

For a optimal solution $V_0, V_1, V_{\frac{1}{2}}$ of VPLP, and a critical independent set I_c, ...

1. $|V_0| + |V_1| + |V_{\frac{1}{2}}| = |I_c| + |N(I_c)| + |(I_c \cup N(I_c))^c|$.
2. $V_0 = N(V_1)$.

Facts

For a optimal solution $V_0, V_1, V_{\frac{1}{2}}$ of VPLP, and a critical independent set I_c, ...

1. $|V_0| + |V_1| + |V_{\frac{1}{2}}| = |I_c| + |N(I_c)| + |(I_c \cup N(I_c))^c|$.
2. $V_0 = N(V_1)$.
3. $|I_c| - |N(I_c)| \geq |V_1| - |V_0|$.

Optimal Solutions give Critical Independent Sets

Let $V_0, V_1, V_{\frac{1}{2}}$ be a feasible solution of VPLP with $N(V_1) = V_0$, and I_c be a critical independent set, ...

Theorem

$V_0, V_1, V_{\frac{1}{2}}$ is an optimal solution of VPLP if, and only if, V_1 is a critical independent set.

Picard-Queyranne Decomposition = Independence Decomposition

Theorem

$V_0, V_1, V_{\frac{1}{2}}$ is an optimal solution with a maximum number of integral variables if, and only if, V_1 is a maximum critical independent set.

Picard-Queyranne Decomposition = Independence Decomposition

Corollary

If $V_0, V_1, V_{\frac{1}{2}}$ *is an optimal solution with a maximum number of integral variables then* $V_0 \cup V_1 = X$*, from IDT.*

Picard-Queyranne Decomposition = Independence Decomposition

Corollary

If $V_0, V_1, V_{\frac{1}{2}}$ is an optimal solution with a maximum number of integral variables then $V_0 \cup V_1 = X$, from IDT.

Corollary

(Picard, Queyranne) There are a unique maximal set of variables which are integral in optimal VPLP solutions.

Picard-Queyranne Decomposition = Independence Decomposition

Corollary

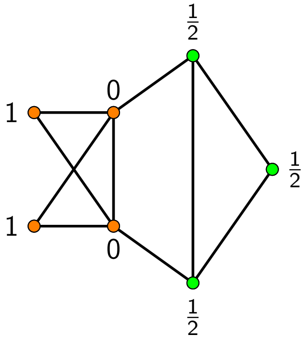
If $V_0, V_1, V_{\frac{1}{2}}$ is an optimal solution with a maximum number of integral variables then $V_0 \cup V_1 = X$, from IDT.

Corollary

(Picard, Queyranne) There are a unique maximal set of variables which are integral in optimal VPLP solutions.

Corollary

(Edmonds, L.) Picard-Queyranne Decomposition = Independence Decomposition


1
1
0
0
1/2
1/2
1/2

Separable Independent Sets

An independent set I is separable if

Separable Independent Sets

An independent set I is separable if

1. $X = I \cup N(I)$,

Separable Independent Sets

An independent set I is separable if

1. $X = I \cup N(I)$,
2. $X^c = V - X$, and

Separable Independent Sets

An independent set I is separable if

1. $X = I \cup N(I)$,
2. $X^c = V - X$, and
3. $\alpha(G) = |I| + \alpha(G[X^c])$.

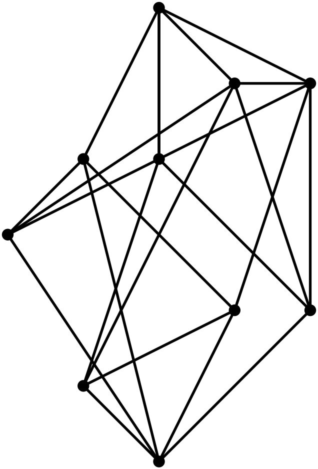
Separable Independent Sets

An independent set I is separable if

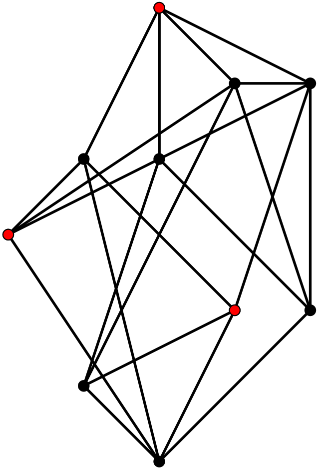
1. $X = I \cup N(I)$,
2. $X^c = V - X$, and
3. $\alpha(G) = |I| + \alpha(G[X^c])$.

Note: critical independent sets are separable independent sets.

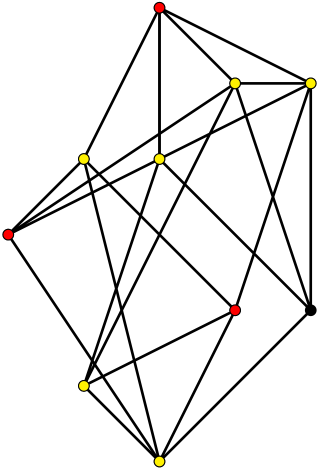
Separable Independent Sets



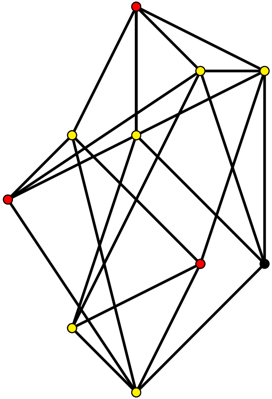
Separable Independent Sets



Separable Independent Sets

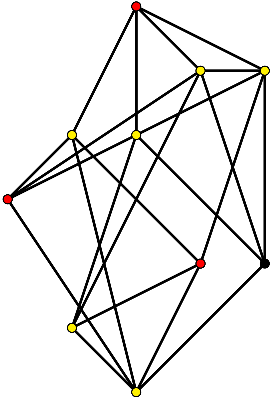


Separable Independent Sets



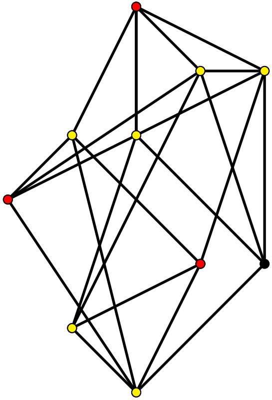
1. I=Red, $N(I)$=Yellow

Separable Independent Sets



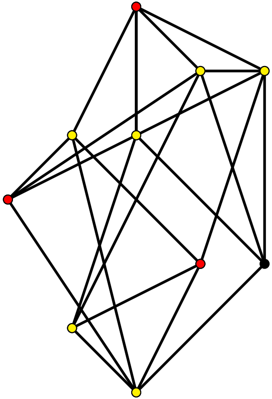
1. I=Red, $N(I)$=Yellow
2. $X = I \cup N(I)$,

Separable Independent Sets



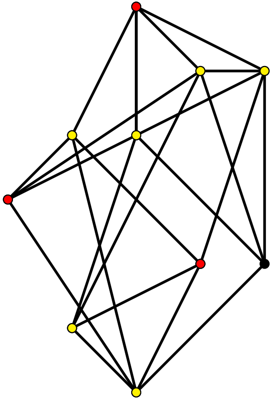
1. I=Red, $N(I)$=Yellow
2. $X = I \cup N(I)$,
3. X^c=Black, and

Separable Independent Sets



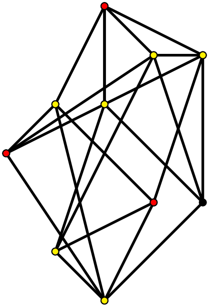
1. I=Red, $N(I)$=Yellow
2. $X = I \cup N(I)$,
3. X^c=Black, and
4. $4 = \alpha(G) = |I| + \alpha(G[X^c]) = 3 + 1$.

Separable Independent Sets

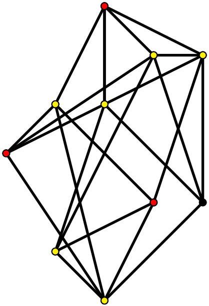


1. I=Red, $N(I)$=Yellow
2. $X = I \cup N(I)$,
3. X^c=Black, and
4. $4 = \alpha(G) = |I| + \alpha(G[X^c]) = 3 + 1$.
5. I is a separable independent set.

Problem 1

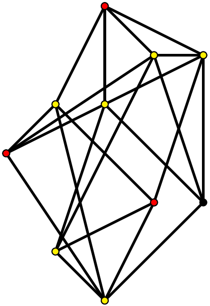


Problem 1



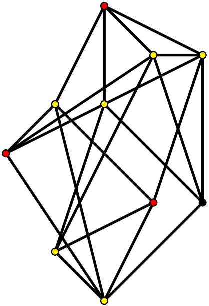
- How to Efficiently Identify Separable Independent Sets?

Problem 1



- How to Efficiently Identify Separable Independent Sets?
- What kinds of separable independent sets are there?

Problem 1

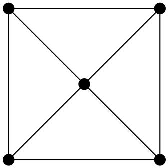


- How to Efficiently Identify Separable Independent Sets?
- What kinds of separable independent sets are there?
- Which kinds can be identified efficiently?

III. Efficient Computation of the Independence Number

When can independence number be computed efficiently?

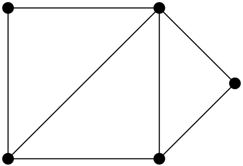
For claw-free graphs.



• G. Minty, On maximal independent sets of vertices in claw-free graphs, Journal of Combinatorial Theory. Series B, 28 (1980) 284–304.

When can independence number be computed efficiently?

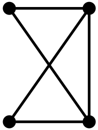
For perfect graphs.



• M. Chudnovsky, G. Cornuéjols, X. Liu, P. Seymour, K. Vušković, Recognizing Berge graphs, Combinatorica 25 (2005) 143–186.

When can independence number be computed efficiently?

For Bipartite and König-Egerváry graphs:



- König, Egerváry, 1931; Kuhn, 1955; Deming, Sterboul, 1979.

When can independence number be computed efficiently?

When can independence number be computed efficiently?

1. P_4-free

When can independence number be computed efficiently?

1. P_4-free
2. (C_4 ,$2K_2$)-free (Hertz, 1997)

When can independence number be computed efficiently?

1. P_4-free
2. (C_4 ,$2K_2$)-free (Hertz, 1997)
3. (Banner-P_8)-free (Gerber, Hertz, Lozin, 2004)

When can independence number be computed efficiently?

1. P_4-free
2. (C_4 ,2K_2)-free (Hertz, 1997)
3. (Banner-P_8)-free (Gerber, Hertz, Lozin, 2004)
4. Chair-free (Alekseev, 2004)

When can independence number be computed efficiently?

1. P_4-free
2. (C_4 ,$2K_2$)-free (Hertz, 1997)
3. (Banner-P_8)-free (Gerber, Hertz, Lozin, 2004)
4. Chair-free (Alekseev, 2004)
5. (P_5,A)-free (Lozin, Mosca, 2009)

When can independence number be computed efficiently?

1. P_4-free
2. (C_4 ,$2K_2$)-free (Hertz, 1997)
3. (Banner-P_8)-free (Gerber, Hertz, Lozin, 2004)
4. Chair-free (Alekseev, 2004)
5. (P_5,A)-free (Lozin, Mosca, 2009)
6. (P_5,$K_{3,3} - e$)-free (Lozin, Mosca, 2009)

When can independence number be computed efficiently?

1. P_4-free
2. $(C_4, 2K_2)$-free (Hertz, 1997)
3. (Banner-P_8)-free (Gerber, Hertz, Lozin, 2004)
4. Chair-free (Alekseev, 2004)
5. (P_5, A)-free (Lozin, Mosca, 2009)
6. $(P_5, K_{3,3} - e)$-free (Lozin, Mosca, 2009)
7. $(P, S_{2,2,2})$-free (Gerber, Lozin, 2003)

When can independence number be computed efficiently?

1. P_4-free
2. (C_4 ,$2K_2$)-free (Hertz, 1997)
3. (Banner-P_8)-free (Gerber, Hertz, Lozin, 2004)
4. Chair-free (Alekseev, 2004)
5. (P_5,A)-free (Lozin, Mosca, 2009)
6. (P_5,$K_{3,3} - e$)-free (Lozin, Mosca, 2009)
7. (P,$S_{2,2,2}$)-free (Gerber, Lozin, 2003)
8. (P_5,diamond)-free (Arbib, Mosca, 2000)

When can independence number be computed efficiently?

1. P_4-free
2. (C_4 ,$2K_2$)-free (Hertz, 1997)
3. (Banner-P_8)-free (Gerber, Hertz, Lozin, 2004)
4. Chair-free (Alekseev, 2004)
5. (P_5,A)-free (Lozin, Mosca, 2009)
6. (P_5,$K_{3,3} - e$)-free (Lozin, Mosca, 2009)
7. (P,$S_{2,2,2}$)-free (Gerber, Lozin, 2003)
8. (P_5,diamond)-free (Arbib, Mosca, 2000)
9. (P_5,co-chair)-free (Brandstädt, Mosca, 2004)

When can independence number be computed efficiently?

1. P_4-free
2. (C_4 ,2K_2)-free (Hertz, 1997)
3. (Banner-P_8)-free (Gerber, Hertz, Lozin, 2004)
4. Chair-free (Alekseev, 2004)
5. (P_5,A)-free (Lozin, Mosca, 2009)
6. (P_5,$K_{3,3} - e$)-free (Lozin, Mosca, 2009)
7. (P,$S_{2,2,2}$)-free (Gerber, Lozin, 2003)
8. (P_5,diamond)-free (Arbib, Mosca, 2000)
9. (P_5,co-chair)-free (Brandstädt, Mosca, 2004)
10. (P_5,co-P)-free (Brandstädt, Mosca, 2004)

When can independence number be computed efficiently?

1. P_4-free
2. (C_4 ,$2K_2$)-free (Hertz, 1997)
3. (Banner-P_8)-free (Gerber, Hertz, Lozin, 2004)
4. Chair-free (Alekseev, 2004)
5. (P_5,A)-free (Lozin, Mosca, 2009)
6. (P_5,$K_{3,3} - e$)-free (Lozin, Mosca, 2009)
7. (P,$S_{2,2,2}$)-free (Gerber, Lozin, 2003)
8. (P_5,diamond)-free (Arbib, Mosca, 2000)
9. (P_5,co-chair)-free (Brandstädt, Mosca, 2004)
10. (P_5,co-P)-free (Brandstädt, Mosca, 2004)
11. (P_5,P)-free (Lozin, 2000; Brandstädt, Lozin, 2001)

When can independence number be computed efficiently?

1. P_4-free
2. (C_4 ,$2K_2$)-free (Hertz, 1997)
3. (Banner-P_8)-free (Gerber, Hertz, Lozin, 2004)
4. Chair-free (Alekseev, 2004)
5. (P_5,A)-free (Lozin, Mosca, 2009)
6. (P_5,$K_{3,3} - e$)-free (Lozin, Mosca, 2009)
7. (P,$S_{2,2,2}$)-free (Gerber, Lozin, 2003)
8. (P_5,diamond)-free (Arbib, Mosca, 2000)
9. (P_5,co-chair)-free (Brandstädt, Mosca, 2004)
10. (P_5,co-P)-free (Brandstädt, Mosca, 2004)
11. (P_5,P)-free (Lozin, 2000; Brandstädt, Lozin, 2001)
12. (P_5,bull)-free (De Simone, 1993)

When can independence number be computed efficiently?

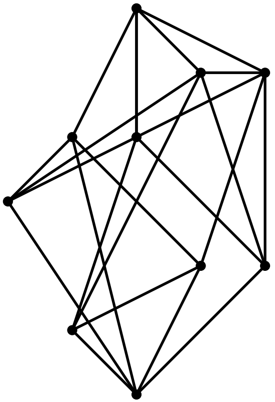
1. P_4-free
2. (C_4 ,2K_2)-free (Hertz, 1997)
3. (Banner-P_8)-free (Gerber, Hertz, Lozin, 2004)
4. Chair-free (Alekseev, 2004)
5. (P_5,A)-free (Lozin, Mosca, 2009)
6. (P_5,$K_{3,3} - e$)-free (Lozin, Mosca, 2009)
7. (P,$S_{2,2,2}$)-free (Gerber, Lozin, 2003)
8. (P_5,diamond)-free (Arbib, Mosca, 2000)
9. (P_5,co-chair)-free (Brandstädt, Mosca, 2004)
10. (P_5,co-P)-free (Brandstädt, Mosca, 2004)
11. (P_5,P)-free (Lozin, 2000; Brandstädt, Lozin, 2001)
12. (P_5,bull)-free (De Simone, 1993)
13. (P_5,house)-free (Hoang, 1983)

When can independence number be computed efficiently?

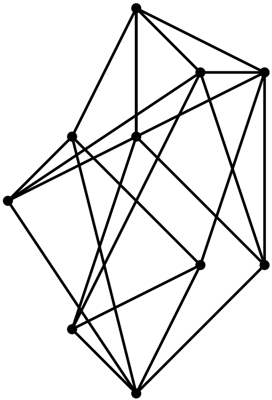
1. P_4-free
2. (C_4 ,$2K_2$)-free (Hertz, 1997)
3. (Banner-P_8)-free (Gerber, Hertz, Lozin, 2004)
4. Chair-free (Alekseev, 2004)
5. (P_5,A)-free (Lozin, Mosca, 2009)
6. (P_5,$K_{3,3} - e$)-free (Lozin, Mosca, 2009)
7. (P,$S_{2,2,2}$)-free (Gerber, Lozin, 2003)
8. (P_5,diamond)-free (Arbib, Mosca, 2000)
9. (P_5,co-chair)-free (Brandstädt, Mosca, 2004)
10. (P_5,co-P)-free (Brandstädt, Mosca, 2004)
11. (P_5,P)-free (Lozin, 2000; Brandstädt, Lozin, 2001)
12. (P_5,bull)-free (De Simone, 1993)
13. (P_5,house)-free (Hoang, 1983)
14. (P_5,gem)-free (Mosca, 1997; Brandstädt, Kratsch, 2001)

When can independence number be computed efficiently?

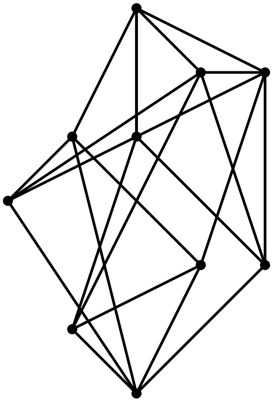
1. P_4-free
2. (C_4 ,$2K_2$)-free (Hertz, 1997)
3. (Banner-P_8)-free (Gerber, Hertz, Lozin, 2004)
4. Chair-free (Alekseev, 2004)
5. (P_5,A)-free (Lozin, Mosca, 2009)
6. (P_5,$K_{3,3} - e$)-free (Lozin, Mosca, 2009)
7. (P,$S_{2,2,2}$)-free (Gerber, Lozin, 2003)
8. (P_5,diamond)-free (Arbib, Mosca, 2000)
9. (P_5,co-chair)-free (Brandstädt, Mosca, 2004)
10. (P_5,co-P)-free (Brandstädt, Mosca, 2004)
11. (P_5,P)-free (Lozin, 2000; Brandstädt, Lozin, 2001)
12. (P_5,bull)-free (De Simone, 1993)
13. (P_5,house)-free (Hoang, 1983)
14. (P_5,gem)-free (Mosca, 1997; Brandstädt, Kratsch, 2001)
15. (P_5,$K_4 - e$)-free (Arbib, Mosca, 2002; Brandstädt, 2004)



Problem 2

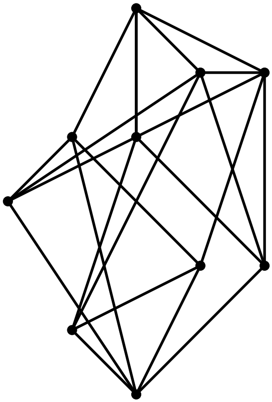


Problem 2



- Find New Forbidden Subgraph Characterizations

Problem 2



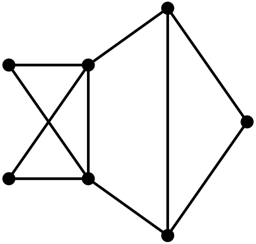
- Find New Forbidden Subgraph Characterizations
- Is it true that the independence number of P_5-free graphs can be computed efficiently?

When can independence number be computed efficiently?

When $l(G) = u(G)$, for an efficiently computable lower bound $l \leq \alpha$ and efficiently computable upper bound $\alpha \leq u$.

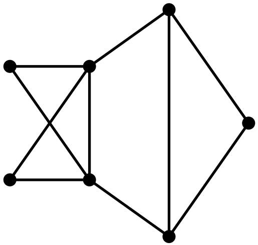
Residue Lower Bound

Given a graph G with degree sequence (d) the residue is the number of zeros at the result of the Havel-Hakimi process.



Residue Lower Bound

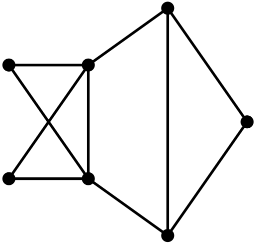
Given a graph G with degree sequence (d) the residue is the number of zeros at the result of the Havel-Hakimi process.



- $4, 4, 3, 3, 2, 2, 2.$

Residue Lower Bound

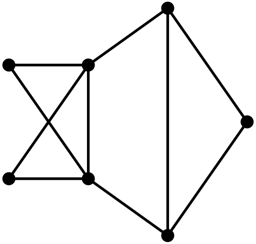
Given a graph G with degree sequence (d) the residue is the number of zeros at the result of the Havel-Hakimi process.



- $4, 4, 3, 3, 2, 2, 2.$
- $0, 0, 0.$

Residue Lower Bound

Given a graph G with degree sequence (d) the residue is the number of zeros at the result of the Havel-Hakimi process.



- $4, 4, 3, 3, 2, 2, 2.$
- $0, 0, 0.$
- $R = 3.$

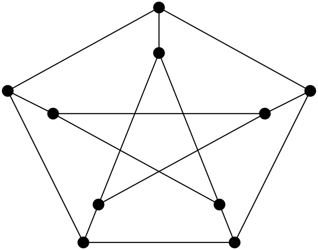
$R \leq \alpha$ Graffiti, 1988; Favaron, Maheo, Sacle, 1991; Griggs, Kleitman, 1994.

Cvetkovíc Eigenvalues Bound

- Cvetkovic bound: $\alpha \leq$ min$\{\#$ of non-negative eigenvalues, $\#$ of non-positive eigenvalues$\}$

Cvetkovíc Eigenvalues Bound

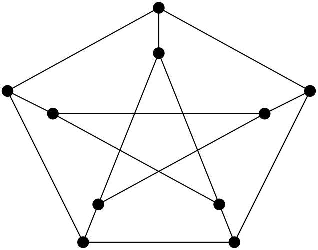
- Cvetkovic bound: $\alpha \leq \min\{\#$ of non-negative eigenvalues, $\#$ of non-positive eigenvalues$\}$



- Eigenvalues: 3, 1, 1, 1, 1, 1, -2, -2, -2, -2.

Cvetkovíc Eigenvalues Bound

• Cvetkovic bound: $\alpha \leq$ min$\{\#$ of non-negative eigenvalues, $\#$ of non-positive eigenvalues$\}$

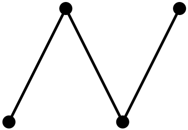


• Eigenvalues: 3, 1, 1, 1, 1, 1, -2, -2, -2, -2.

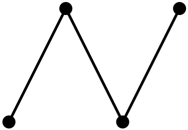
• $\alpha \leq 4$.

• D. Cvetković, M. Doob, and H. Sachs, Spectra of Graphs, 3rd ed., 1995.

When can independence number be computed efficiently?

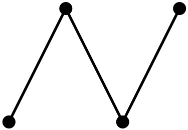


When can independence number be computed efficiently?



- $2 = Residue \leq \alpha \leq Cvetkovic = 2$

When can independence number be computed efficiently?



- $2 = \textit{Residue} \leq \alpha \leq \textit{Cvetkovic} = 2$
- Independence Number Theory implies $\alpha = 2$.

Residue is a good lower bound for α

Residue is a good lower bound for α

For connected graphs with minimum degree ≥ 3 and maximum degree $\leq n - 2$,

Residue is a good lower bound for α

For connected graphs with minimum degree ≥ 3 and maximum degree $\leq n-2$,

- Residue predicts α for 6 out of 8 graphs of order 6.

Residue is a good lower bound for α

For connected graphs with minimum degree ≥ 3 and maximum degree $\leq n-2$,

- Residue predicts α for 6 out of 8 graphs of order 6.
- Residue predicts α for 38 out of 88 graphs of order 7.

Residue is a good lower bound for α

For connected graphs with minimum degree ≥ 3 and maximum degree $\leq n-2$,

- Residue predicts α for 6 out of 8 graphs of order 6.
- Residue predicts α for 38 out of 88 graphs of order 7.
- Residue predicts α for 411 out of 2079 graphs of order 8.

Residue is a good lower bound for α

For connected graphs with minimum degree ≥ 3 and maximum degree $\leq n-2$,

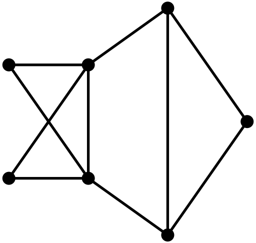
- Residue predicts α for 6 out of 8 graphs of order 6.
- Residue predicts α for 38 out of 88 graphs of order 7.
- Residue predicts α for 411 out of 2079 graphs of order 8.
- Residue predicts α for 11620 out of 76783 graphs of order 9.

Residue is a good lower bound for α

For connected graphs with minimum degree ≥ 3 and maximum degree $\leq n-2$,

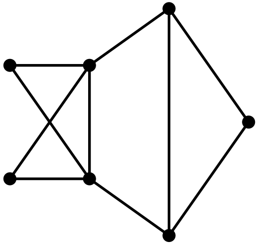
- Residue predicts α for 6 out of 8 graphs of order 6.
- Residue predicts α for 38 out of 88 graphs of order 7.
- Residue predicts α for 411 out of 2079 graphs of order 8.
- Residue predicts α for 11620 out of 76783 graphs of order 9.
- Residue predicts α for 501793 out of 5005243 graphs of order 10

Problem 3



$$\alpha = \textit{residue} = 3$$

Problem 3



$$\alpha = \textit{residue} = 3$$

- Characterize the Graphs where $\alpha = \textit{residue}$

Lovász Theta Function

- The Lovász number of a graph G is:

$$\vartheta(G) = \max[1 - \frac{\lambda_1(A)}{\lambda_n(A)}]$$

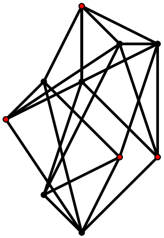
over all real matrices A with $a_{ij} = 0$ if $v_i \sim v_j$ in G, with eigenvalues $\lambda_1(A) \geq \ldots \lambda_n(A)$

Lovász Theta Function

- The Lovász number of a graph G is:

$$\vartheta(G) = \max[1 - \frac{\lambda_1(A)}{\lambda_n(A)}]$$

over all real matrices A with $a_{ij} = 0$ if $v_i \sim v_j$ in G, with eigenvalues $\lambda_1(A) \geq \ldots \lambda_n(A)$



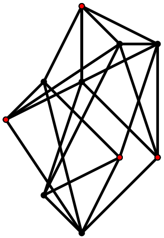
$\alpha \leq \vartheta = 4$

Lovász Theta Function

- The Lovász number of a graph G is:

$$\vartheta(G) = \max[1 - \frac{\lambda_1(A)}{\lambda_n(A)}]$$

over all real matrices A with $a_{ij} = 0$ if $v_i \sim v_j$ in G, with eigenvalues $\lambda_1(A) \geq \ldots \lambda_n(A)$



$\alpha \leq \vartheta = 4$

- L. Lovász, On the Shannon capacity of a graph, *IEEE Transactions on Information Theory*, 1979.
- D. Knuth, The sandwich theorem, Electronic Journal of Combinatorics 1 (1994).

Lovász Theta is a very good upper bound for α

For *all* simple graphs,

Lovász Theta is a very good upper bound for α

For *all* simple graphs,

- ϑ predicts α for 34 out of 34 graphs of order 5.

Lovász Theta is a very good upper bound for α

For *all* simple graphs,

- ϑ predicts α for 34 out of 34 graphs of order 5.
- ϑ predicts α for 156 out of 156 graphs of order 6.

Lovász Theta is a very good upper bound for α

For *all* simple graphs,

- ϑ predicts α for 34 out of 34 graphs of order 5.
- ϑ predicts α for 156 out of 156 graphs of order 6.
- ϑ predicts α for 1044 out of 1044 graphs of order 7.

Lovász Theta is a very good upper bound for α

For *all* simple graphs,

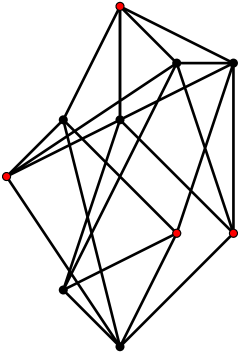
- ϑ predicts α for 34 out of 34 graphs of order 5.
- ϑ predicts α for 156 out of 156 graphs of order 6.
- ϑ predicts α for 1044 out of 1044 graphs of order 7.
- ϑ predicts α for 12346 out of 12346 graphs of order 8.

Lovász Theta is a very good upper bound for α

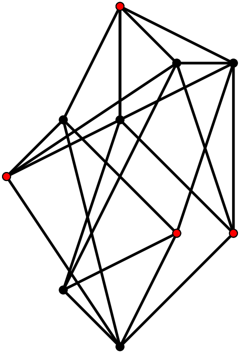
For *all* simple graphs,

- ϑ predicts α for 34 out of 34 graphs of order 5.
- ϑ predicts α for 156 out of 156 graphs of order 6.
- ϑ predicts α for 1044 out of 1044 graphs of order 7.
- ϑ predicts α for 12346 out of 12346 graphs of order 8.
- ϑ predicts α for 274668 out of 274668 graphs of order 9.

Problem 4



Problem 4



- Characterize Graphs Where $\alpha =$ Lovász Theta.

Efficiently Computable Bounds for the Independence Number

Efficiently Computable Bounds for the Independence Number

1. Residue Lower Bound

Efficiently Computable Bounds for the Independence Number

1. Residue Lower Bound
2. Cvetkovic Upper Bound

Efficiently Computable Bounds for the Independence Number

1. Residue Lower Bound
2. Cvetkovic Upper Bound
3. Lovász Theta Upper Bound

Efficiently Computable Bounds for the Independence Number

1. Residue Lower Bound
2. Cvetkovic Upper Bound
3. Lovász Theta Upper Bound
4. Even minus Even Horizontal Lower Bound

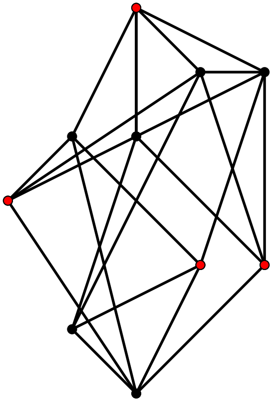
Efficiently Computable Bounds for the Independence Number

1. Residue Lower Bound
2. Cvetkovic Upper Bound
3. Lovász Theta Upper Bound
4. Even minus Even Horizontal Lower Bound
5. Fractional Independence Upper Bound

Efficiently Computable Bounds for the Independence Number

1. Residue Lower Bound
2. Cvetkovic Upper Bound
3. Lovász Theta Upper Bound
4. Even minus Even Horizontal Lower Bound
5. Fractional Independence Upper Bound
6. 50 Efficiently Computable Bounds are Known

Problem 5



- Find More Efficiently Computable Bounds for α.

When can independence number be computed efficiently?

When can independence number be computed efficiently?

- When the graph has a vertex v of degree $n-1$, and α of $G - N[v]$ can be computed efficiently. (So $\alpha(G) = \alpha(G - N[v]))$

When can independence number be computed efficiently?

- When the graph has a vertex v of degree $n-1$, and α of $G - N[v]$ can be computed efficiently. (So $\alpha(G) = \alpha(G - N[v])$)
- When the graph has twin vertices v and w (that is, $N[v] = N[w]$) and α of $G - v$ can be computed efficiently. So $\alpha(G) = \alpha(G - v)$).

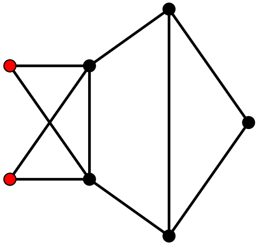
When can independence number be computed efficiently?

- When the graph has a vertex v of degree $n-1$, and α of $G - N[v]$ can be computed efficiently. (So $\alpha(G) = \alpha(G - N[v]))$

- When the graph has twin vertices v and w (that is, $N[v] = N[w]$) and α of $G - v$ can be computed efficiently. So $\alpha(G) = \alpha(G - v))$.

- When the graph has a simplicial vertex (that is, a vertex v, where $N[v]$ is complete. So $\alpha(G) = \alpha(G - v))$.

When can independence number be computed efficiently?



When G has a non-empty critical independent set, and α of $G[X^c]$ can be computed efficiently.

• L., A note on critical independence reductions, Bulletin of the ICA 51 (2007) 34–46.

What is an α-reduction?

What is an α-reduction?

G is α-reducible if it is possible to efficiently find a smaller order graph G' such that $\alpha(G)$ can be computed in terms of $\alpha(G')$.

What is an α-reduction?

G is α-reducible if it is possible to efficiently find a smaller order graph G' such that $\alpha(G)$ can be computed in terms of $\alpha(G')$.

1. Is disconnected.

What is an α-reduction?

G is α-reducible if it is possible to efficiently find a smaller order graph G' such that $\alpha(G)$ can be computed in terms of $\alpha(G')$.

1. Is disconnected.
2. Maximum degree $= n - 1$.

What is an α-reduction?

G is α-reducible if it is possible to efficiently find a smaller order graph G' such that $\alpha(G)$ can be computed in terms of $\alpha(G')$.

1. Is disconnected.
2. Maximum degree $= n - 1$.
3. Has twin vertices.

What is an α-reduction?

G is α-reducible if it is possible to efficiently find a smaller order graph G' such that $\alpha(G)$ can be computed in terms of $\alpha(G')$.

1. Is disconnected.
2. Maximum degree $= n - 1$.
3. Has twin vertices.
4. Has a simplicial vertex.

What is an α-reduction?

G is α-reducible if it is possible to efficiently find a smaller order graph G' such that $\alpha(G)$ can be computed in terms of $\alpha(G')$.

1. Is disconnected.
2. Maximum degree $= n - 1$.
3. Has twin vertices.
4. Has a simplicial vertex.
5. Has a non-empty critical independent set.

What is an α-reduction?

G is α-reducible if it is possible to efficiently find a smaller order graph G' such that $\alpha(G)$ can be computed in terms of $\alpha(G')$.

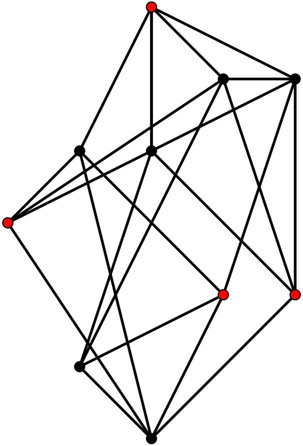
1. Is disconnected.
2. Maximum degree $= n - 1$.
3. Has twin vertices.
4. Has a simplicial vertex.
5. Has a non-empty critical independent set.
6. Has a foldable vertex (Fomin, Grandoni, Kratsch, 2006).

What is an α-reduction?

G is α-reducible if it is possible to efficiently find a smaller order graph G' such that $\alpha(G)$ can be computed in terms of $\alpha(G')$.

1. Is disconnected.
2. Maximum degree $= n - 1$.
3. Has twin vertices.
4. Has a simplicial vertex.
5. Has a non-empty critical independent set.
6. Has a foldable vertex (Fomin, Grandoni, Kratsch, 2006).
7. Has a magnet (Leveque, de Werra, 2012).

Problem 6



- Find new α-reductions.

IV. The Independence Number Project

Joint Work with Patrick Gaskill

What is the Independence Number Project?

What is the Independence Number Project?

The main idea is to find the smallest graphs whose independence number cannot be efficiently computed (according to existing Independence Number Theory) and use these graphs to help extend the theory.

What is the Independence Number Project?

The main idea is to find the smallest graphs whose independence number cannot be efficiently computed (according to existing Independence Number Theory) and use these graphs to help extend the theory.

- Connected graphs with $n \leq 4$ either have a degree $n-1$ vertex, or a foldable vertex.

What is the Independence Number Project?

The main idea is to find the smallest graphs whose independence number cannot be efficiently computed (according to existing Independence Number Theory) and use these graphs to help extend the theory.

- Connected graphs with $n \leq 4$ either have a degree $n-1$ vertex, or a foldable vertex.
- Then we generated all graphs with $n = 5$, and checked if:

What is the Independence Number Project?

The main idea is to find the smallest graphs whose independence number cannot be efficiently computed (according to existing Independence Number Theory) and use these graphs to help extend the theory.

- Connected graphs with $n \leq 4$ either have a degree $n-1$ vertex, or a foldable vertex.
- Then we generated all graphs with $n = 5$, and checked if:
  - they were reducible (and α could be computed in terms of the independence number of a graph with $n < 5$),

What is the Independence Number Project?

The main idea is to find the smallest graphs whose independence number cannot be efficiently computed (according to existing Independence Number Theory) and use these graphs to help extend the theory.

- Connected graphs with $n \leq 4$ either have a degree $n - 1$ vertex, or a foldable vertex.
- Then we generated all graphs with $n = 5$, and checked if:
  - they were reducible (and α could be computed in terms of the independence number of a graph with $n < 5$),
  - they had an α-property, or

What is the Independence Number Project?

The main idea is to find the smallest graphs whose independence number cannot be efficiently computed (according to existing Independence Number Theory) and use these graphs to help extend the theory.

- Connected graphs with $n \leq 4$ either have a degree $n-1$ vertex, or a foldable vertex.
- Then we generated all graphs with $n = 5$, and checked if:
  - they were reducible (and α could be computed in terms of the independence number of a graph with $n < 5$),
  - they had an α-property, or
  - the best upper bound equals the best lower bound.

What is the Independence Number Project?

The main idea is to find the smallest graphs whose independence number cannot be efficiently computed (according to existing Independence Number Theory) and use these graphs to help extend the theory.

- Connected graphs with $n \leq 4$ either have a degree $n-1$ vertex, or a foldable vertex.
- Then we generated all graphs with $n = 5$, and checked if:
  - they were reducible (and α could be computed in terms of the independence number of a graph with $n < 5$),
  - they had an α-property, or
  - the best upper bound equals the best lower bound.
- Then we generated all graphs with $n = 6$, ...

What is the Independence Number Project?

The main idea is to find the smallest graphs whose independence number cannot be efficiently computed (according to existing Independence Number Theory) and use these graphs to help extend the theory.

- Connected graphs with $n \leq 4$ either have a degree $n-1$ vertex, or a foldable vertex.
- Then we generated all graphs with $n = 5$, and checked if:
  - they were reducible (and α could be computed in terms of the independence number of a graph with $n < 5$),
  - they had an α-property, or
  - the best upper bound equals the best lower bound.
- Then we generated all graphs with $n = 6$, ...
- Then we generated all graphs with $n = 7$, ...

What is the Independence Number Project?

The main idea is to find the smallest graphs whose independence number cannot be efficiently computed (according to existing Independence Number Theory) and use these graphs to help extend the theory.

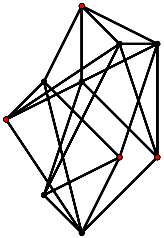
- Connected graphs with $n \leq 4$ either have a degree $n-1$ vertex, or a foldable vertex.
- Then we generated all graphs with $n = 5$, and checked if:
  - they were reducible (and α could be computed in terms of the independence number of a graph with $n < 5$),
  - they had an α-property, or
  - the best upper bound equals the best lower bound.
- Then we generated all graphs with $n = 6$, ...
- Then we generated all graphs with $n = 7$, ...
- Then we generated all graphs with $n = 8$, ...

What is the Independence Number Project?

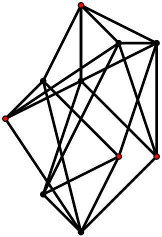
The main idea is to find the smallest graphs whose independence number cannot be efficiently computed (according to existing Independence Number Theory) and use these graphs to help extend the theory.

- Connected graphs with $n \leq 4$ either have a degree $n-1$ vertex, or a foldable vertex.
- Then we generated all graphs with $n = 5$, and checked if:
  - they were reducible (and α could be computed in terms of the independence number of a graph with $n < 5$),
  - they had an α-property, or
  - the best upper bound equals the best lower bound.
- Then we generated all graphs with $n = 6$, ...
- Then we generated all graphs with $n = 7$, ...
- Then we generated all graphs with $n = 8$, ...
- Then we generated all graphs with $n = 9$, ...

What is the Independence Number Project?

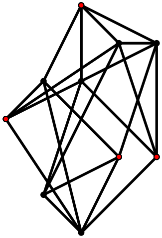


What is the Independence Number Project?



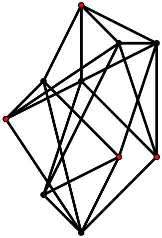
- But we got stuck at $n = 10$.

What is the Independence Number Project?



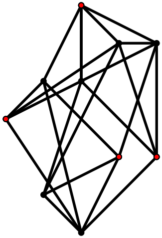
- But we got stuck at $n = 10$.
- For this graph (I?bbrr[ko), $\alpha = 4$ but the best lower bound $=$ 3,

What is the Independence Number Project?



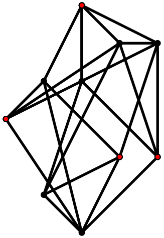
- But we got stuck at $n = 10$.
- For this graph (I?bbrr[ko), $\alpha = 4$ but the best lower bound = 3,
- It has no $n-1$ vertices, no foldable vertices, no simplicial vertices, no magnets, or any other α-reductions,

What is the Independence Number Project?



- But we got stuck at $n = 10$.
- For this graph (I?bbrr[ko), $\alpha = 4$ but the best lower bound $=$ 3,
- It has no $n-1$ vertices, no foldable vertices, no simplicial vertices, no magnets, or any other α-reductions,
- It has a claw, a bull, a chair, a co-chair, a house, a P_5, a P, a co-P, a gem, and a diamond.

What is the Independence Number Project?



- But we got stuck at $n = 10$.
- For this graph (I?bbrr[ko), $\alpha = 4$ but the best lower bound $=$ 3,
- It has no $n - 1$ vertices, no foldable vertices, no simplicial vertices, no magnets, or any other α-reductions,
- It has a claw, a bull, a chair, a co-chair, a house, a P_5, a P, a co-P, a gem, and a diamond.
- It demands new theory.

Problems

Problems

1. Characterize of and Efficiently Identify Separable Independent Sets

Problems

1. Characterize of and Efficiently Identify Separable Independent Sets

2. Find new forbidden subgraph characterizations, especially one that applies to I?bbrr[ko. Find an efficient algorithm for P_5-free graphs.

Problems

1. Characterize of and Efficiently Identify Separable Independent Sets
2. Find new forbidden subgraph characterizations, especially one that applies to I?bbrr[ko. Find an efficient algorithm for P_5-free graphs.
3. Efficiently characterize graphs where $\alpha = \mathit{residue}$.

Problems

1. Characterize of and Efficiently Identify Separable Independent Sets
2. Find new forbidden subgraph characterizations, especially one that applies to I?bbrr[ko. Find an efficient algorithm for P_5-free graphs.
3. Efficiently characterize graphs where $\alpha = \mathit{residue}$.
4. Efficiently characterize graphs where $\alpha = \vartheta$.

Problems

1. Characterize of and Efficiently Identify Separable Independent Sets
2. Find new forbidden subgraph characterizations, especially one that applies to I?bbrr[ko. Find an efficient algorithm for P_5-free graphs.
3. Efficiently characterize graphs where $\alpha = \textit{residue}$.
4. Efficiently characterize graphs where $\alpha = \vartheta$.
5. Find new efficiently computable bounds for α, especially lower bounds, and especially one that gives equality for I?bbrr[ko.

Problems

1. Characterize of and Efficiently Identify Separable Independent Sets
2. Find new forbidden subgraph characterizations, especially one that applies to I?bbrr[ko. Find an efficient algorithm for P_5-free graphs.
3. Efficiently characterize graphs where $\alpha = \textit{residue}$.
4. Efficiently characterize graphs where $\alpha = \vartheta$.
5. Find new efficiently computable bounds for α, especially lower bounds, and especially one that gives equality for I?bbrr[ko.
6. Find new α-reductions, especially one that applies to I?bbrr[ko.

Tools

Tools

- Sage: `sagemath.org`

Tools

- Sage: `sagemath.org`
- Python: `python.org`

Tools

- Sage: `sagemath.org`
- Python: `python.org`
- nauty: `cs.anu.edu.au/~bdm/nauty`

Tools

- Sage: `sagemath.org`
- Python: `python.org`
- nauty: `cs.anu.edu.au/~bdm/nauty`
- Grinvin: `grinvin.org`

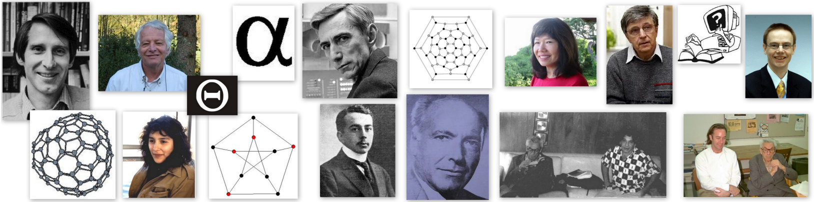
Tools

- Sage: `sagemath.org`
- Python: `python.org`
- nauty: `cs.anu.edu.au/~bdm/nauty`
- Grinvin: `grinvin.org`
- Github: `github.com`

Tools

- Sage: `sagemath.org`
- Python: `python.org`
- nauty: `cs.anu.edu.au/~bdm/nauty`
- Grinvin: `grinvin.org`
- Github: `github.com`
- Wordpress: `wordpress.com`

Thank You!



The Independence Number Project:

`independencenumber.wordpress.com`

clarson@vcu.edu