THE PROBLEM OF WORM PROPAGATION/PREVENTION I.E. THE MINIMUM VERTEX COVER PROBLEM

Prof. Tiziana Calamoneri
Network Algorithms
A.y. 2024/25

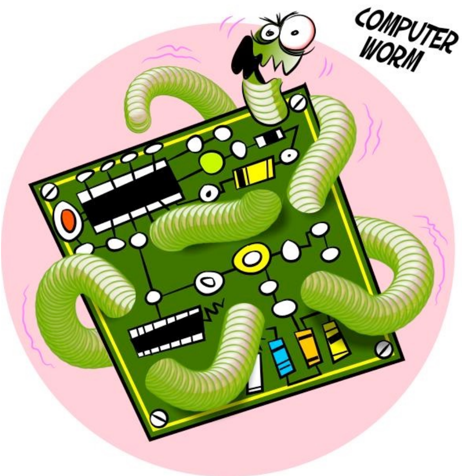
1

THE PROBLEM

2

WORMS (1)

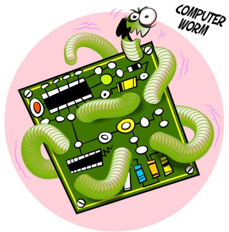
- A computer worm is a standalone malware (=malicious software, used or programmed by attackers to disrupt computer operation, gather sensitive information, or gain access to private computer systems) that replicates itself using a computer network in order to spread to other computers, relying on security failures on the target computer to access it.



WORMS (2)

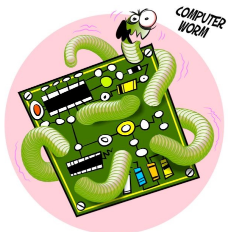
- Unlike a computer virus, it does not need to attach itself to an existing program.
- Worms almost always cause at least some harm to the network, even if only by consuming bandwidth, whereas viruses almost always corrupt or modify files on a targeted computer.

WORMS (3)



- One of the first worms was created by R. Morris in 1988 and called Internet Worm.
- It was able to affect between 4000 and 6000 machines, i.e. about the 4-6% of the computers connected to Internet at that time.

WORMS (4)



A worm tries to replicate itself in different ways:

- E-mails: it looks for e-mail addresses in the infected machine and generates additional e-mail messages contaning copies of itself.
- Social engineering techniques in order to induce people to open attachments containing the worm.
- Bugs of e-mail clients, in order to auto-execute themselves, once the message is simply visualized.

DAMAGES CAUSED BY WORMS (1)

We can roughly divide the harmful effects caused by a worm in two types:

- direct damages, resulting from the execution of the worm on the victim machine, and
- indirect damages, arising from the techniques used for the diffusion.

DAMAGES CAUSED BY WORMS (2)

Direct Damages:

Worms usually carry payloads that do considerable damage.

A payload of a worm is designed to do more than spread the worm; it might:

- delete files on a host system (e.g., the ExploreZip worm),
- encrypt files in a cryptoviral extortion attack,
- send documents via e-mail
- install a backdoor in the infected computer to allow the creation of a "zombie" computer under control of the worm author.

DAMAGES CAUSED BY WORMS (3)

Direct Damages (cntd):

- Simple worms, compound only by the instructions to replicate themselves, do not create serious direct damage beyond the waste of computational resources.
- Often, however, they interfere with the software designed to find them and to counteract the spread (antivirus and firewall) thus obstructing the normal operation of the host computer.
- …

DAMAGES CAUSED BY WORMS (4)

Direct Damages (cntd):

- …
- Very frequently a worm acts as a vehicle for automatic installation of backdoors or keyloggers, which can then be exploited by an attacker or another worm.
- They may also open TCP ports to create networks security holes for other applications.

DAMAGES CAUSED BY WORMS (5)

Undirect Damages:

- These are the side effects of infection by a worm of a large number of computers connected to the network.
- The e-mail messages sent by the worm to replicate increase the amount of junk e-mail, wasting valuable resources in terms of bandwidth and attention.
- The worms that exploit known vulnerabilities of some software cause desease of such programs, with consequences such as instability of the operating system and sometimes forced reboots and shutdowns.

WORM PROPAGATION (1)

To simplify, assume that the time of transmission of information in any given connection in the network is the same, equal to T.

If a worm has successfully infected a set of nodes C such that with a single step of spread all nodes can be infected, in time T, the entire network is infected ("first propagation step").

NOTE:

The real problem is more complex because all the networks of considerable size have dynamic connections.

WORM PROPAGATION (2)

- Knowing set C is the first step to protect the network from attack.
- The property that every edge is incident to a node in C is sufficient (although not necessary) to be sure to infect the network after the first step.
- From the point of view of the manager of the network, each filter to protect the network against attacks from worms of the first order slows down the communication and therefore it is necessary to minimize the number.

GRAPH MODEL OF THE PROBLEM

THE GRAPH MODEL (1)

- Def. Let $G=(V,E)$ be an undirected graph. The vertex cover is a subset V' of the nodes of the graph which contains at least one of the two endpoints of each edge.
- It is relevant to find the minimum vertex cover, i.e. the set V' of minimum cardinality.
- Obs. The minimum vertex cover is not unique:

THE GRAPH MODEL (2)

- Intuitively, every minimum vertex cover represents an excellent starting point for a worm.
- The computers to be protected are those that represent the nodes in the minimum vertex cover of the communication graph.
- If the graph has more than one minimum vertex cover, the computers in the intersection of all the covers need to be protected.

THE MINIMUM VERTEX COVER PROBLEM

17

MINIMUM VERTEX COVER (1)

- **Def.** Given *G=(V,E)*, *V'* subset of *V* is a vertex cover for *G* if ∀ {a,b}∈ *E*, *a* ∈ *V'* or *b* ∈ *V'*.
- **Obs.** Set *V* is trivially a vertex cover.
- Given *G=(V,E)*, the minimum Vertex Cover Problem is to find a vertex cover for G of minimum cardinality.
- **Obs.** There are 2^n possible subsets to check.

MINIMUM VERTEX COVER (2)

- Def. The Decisional version of the Minimum Vertex Cover Problem (VC) is to answer to the following question:

  given a graph G and an integer value k, is there a vertex cover for G of cardinality less than or equal to k?

MINIMUM VERTEX COVER (3)

VC is among the Karp's 21 NP-complete problems [Karp'72], a set of computational problems which have been proved to be NP-complete right after the Cook theorem ['71]

(first demonstrations that many natural computational problems occurring throughout computer science are computationally intractable; it drove interest in the study of NP-completeness and the "P versus NP" problem).

MINIMUM VERTEX COVER (4)

- The reduction is directly from 3-SAT or from MaxClique.
- VC is still NP-complete on cubic graphs [Garey, Johnson, Stockmeyer '74] and on planar graphs having degree at most 3 [Garey & Johnson '77].

MINIMUM VERTEX COVER (5)

ILP formulation for VC:

- We introduce the following n decision variables:

  for each i=1, 2, …, n, $x_i=1$ if the node i belongs to V' and $x_i=0$ otherwise.
- Objective: $\min \sum_{i=1}^{n} x_i$
- subject to constraints:

$$x_i + x_j \geq 1, \forall (i,j) \in E$$

$$x_i \in \{0,1\}, i = 1,2,\ldots,n$$

Note. Solving an ILP is in general NP-complete.

MINIMUM VERTEX COVER (5)

Summary:

- As already highlighted, VC is an NP-hard problem, so only superpolynomial agorithms are known.
- It is possible to approximate the solution in polynomial time.

MINIMUM VERTEX COVER (6)

Summary (cntd):

- In the following we will describe two naive algorithms that seem intuitively good but have, instead, bad approximation ratios.
- Then, we will describe a $O(n+m)$ time approximate algorithm that finds a vertex cover V' s.t. $|V'| \leq 2|V^*|$, where V^* is an optimal solution.
- Finally, we propose another 2-approximate algorithm exploiting the ILP formulation.

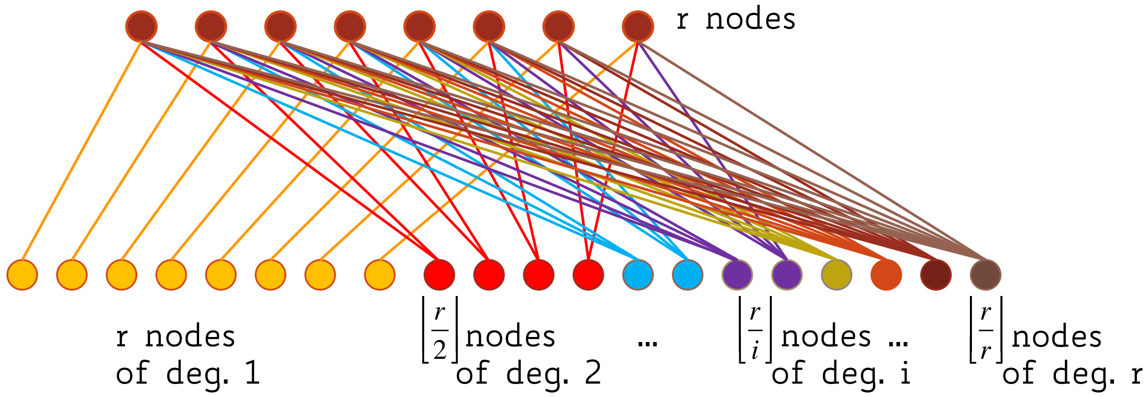
MINIMUM VERTEX COVER (7)

Algorithm Greedy1-VC(G)

- V'=empty set, E'=E
- While (E' is not empty) do
  - Select from E' an edge (i,j) and choose one of its endpoints i
  - Add i to V'
  - delete from E' all edges having i as an endpoint
- Return V'

MINIMUM VERTEX COVER (8)

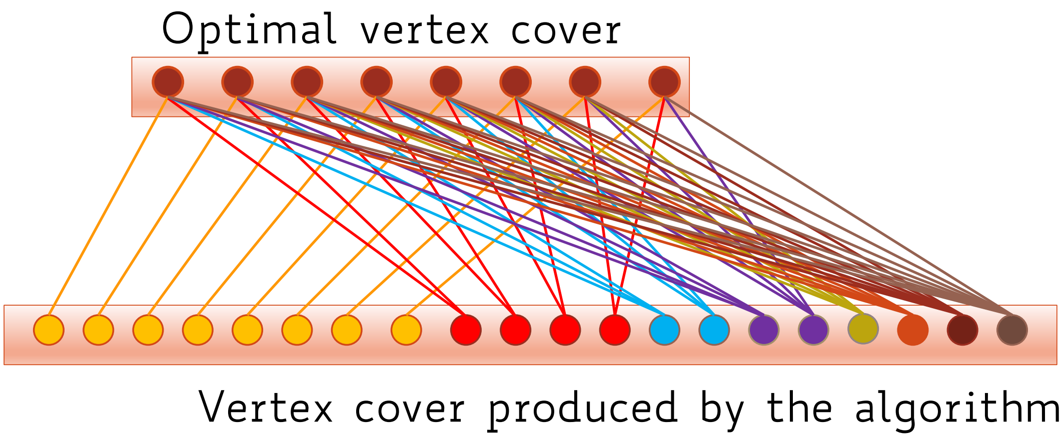
Unfortunately, Greedy1-VC could produce a vertex cover whose cardinality is very far from optimum:



$$n = r + \sum_{i=1}^{r} \left\lfloor \frac{r}{i} \right\rfloor \leq r + r\sum_{i=1}^{r} \frac{1}{i} = \Theta(r \log r)$$

Harmonic sum; can be approximated by ln r

MINIMUM VERTEX COVER (9)



Approximation ratio: $\Theta(\log r)$

Problem:
the algorithm could prefer small degree nodes instead of large degree nodes

MINIMUM VERTEX COVER (10)

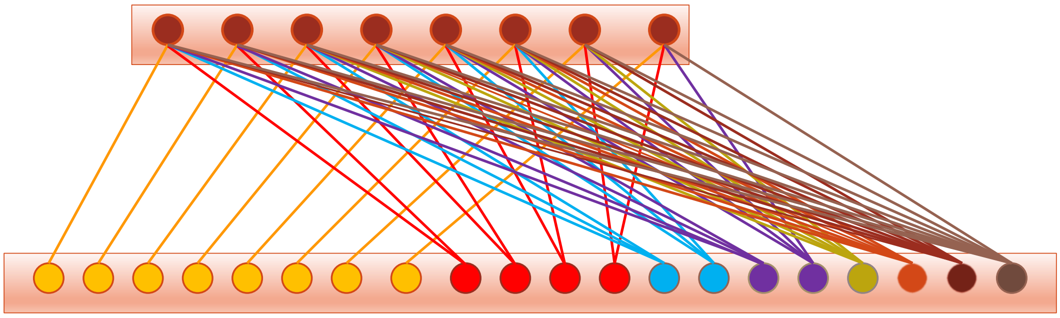
Let us try another approach:

Algorithm Greedy2-VC(G)

- V'=empty set, E'=E
- While (E' is not empty) do
  - Select a node v having max degree in the current graph
  - Add v to V'
  - Delete from E' all the edges having v as an endpoint
- Return V'

MINIMUM VERTEX COVER (11)

This second algorithm may produce this vertex cover…



…but even this, starting from the nodes to the right

So, the approximation ratio does not change!

MINIMUM VERTEX COVER (12)

A better algorithm:

Algorithm 2-Approx1-VC(G)

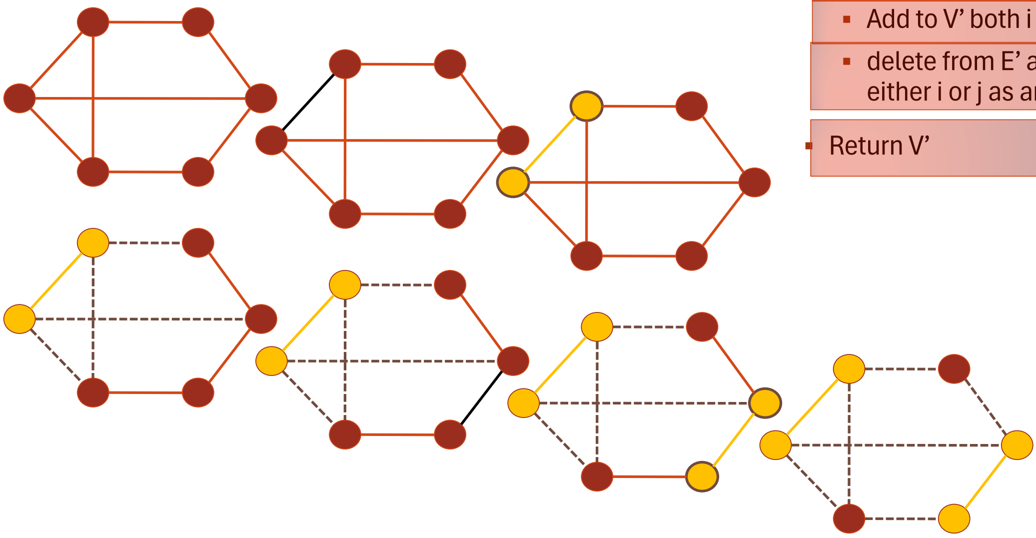
- V'=empty set
- E'=E
- While (E' is not empty) do
  - select from E' an edge (i,j)
  - Add to V' both i and j
  - Delete from E' all the edges having either i or j as an endpoint
- Return V'

Time complexity: O(n+m)

MINIMUM VERTEX COVER (13)

Algorithm 2-Approx-VC(G)

- V'=empty set
- E'=E
- While (E' is not empty) do
  - select from E' an edge (i,j)
  - Add to V' both i and j
  - delete from E' all the edges with either i or j as an endpoint
- Return V'



MINIMUM VERTEX COVER (14)

Th. Let V* be a minimum vertex cover. The set V' returned by 2-Approx1-VC is a vertex cover such that $|V'|\leq 2|V^*|$.

Proof. By construction, V' is a vertex cover.

Let A be the set of the edges selected from E'. For each edge (i,j) in A, i and j are added to V' so:

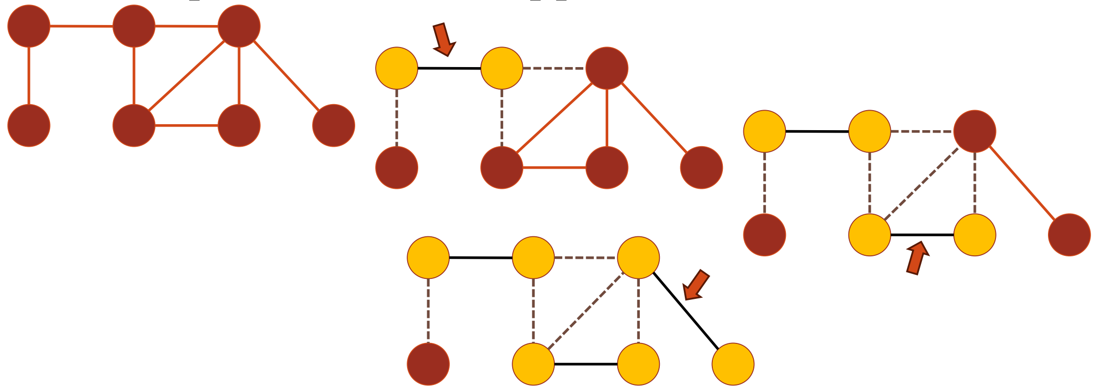
$$|V'|=2|A|.$$

Moreover, all the edges having either i or j as an endpoint are deleted from E', so edges in A cannot be incident and must be covered by any optimal solution, i.e. $|A|\leq|V^*|$.

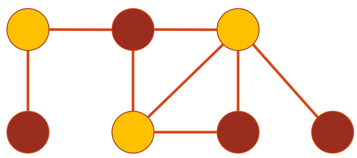
Putting together: $|V'|=2|A|\leq 2|V^*|$. ■

MINIMUM VERTEX COVER (15)

An example where the upper bound is reached:



optimal solution:



MINIMUM VERTEX COVER (16)

An algorithm based on the ILP formulation:

Algorithm 2-Approx2-VC(G)

- V'=empty set
- Relax the ILP formulation by eliminating the constraint that x_i must be integer.
- Invoke a polynom. time LP solver to get a solution $x_1, \ldots, x_n$
- For i=1 to n do
  - if $x_i \geq ½$ then
    - add to V' node i
- Return V'

Time complexity: O(n+m)

MINIMUM VERTEX COVER (17)

Th. The node set V' returned by 2-Approx2-VC is a vertex cover.

Proof. We know from our constraints that for each edge (i,j), $x_i+x_j \geq 1$. Therefore, at least one of x_i or $x_j \geq ½$ and so at least one of the nodes i,j from the edge (i,j) must belong to V'. ■

MINIMUM VERTEX COVER (18)

Th. Let V^* be a minimum vertex cover. The vertex cover V' returned by 2-Approx2-VC is such that:

$|V'| \leq 2|V^*|$.

Proof. Let $Z^*=x_1+\ldots+x_n$ the "cost" of the optimal solution. (This is the sum of real numbers and not the size of any set.)

Since $x_1, \ldots, x_n$ is optimal for the LP, $Z^* \leq |V^*|$.

Let $x'_1, \ldots, x'_n$ the int. solution obtained from $x_1, \ldots, x_n$.

Of course, $x'_i \leq 2x_i$ for each i=1, …, n, so

$|V'|=x'_1+ \ldots + x'_n \leq 2(x_1+ \ldots + x_n)=2Z^* \leq 2|V^*|$. ■

MINIMUM VERTEX COVER (19)

- Even if these two latter algorithms are very easy, it is impossible to do much better, indeed:
- VC is not approximable in less than 1.1666 [Håstad '97] and then in less than 1.3606 [Dinur & Safra '05]
- The best known approximation ratios are:

$$2-\frac{\log\log|V|}{2\log|V|}$$ [Monien& Speckenmeyer '85]

$$2-\frac{\log\log|V|}{\log|V|}$$ [Bar-Yehuda, Even '85]

$$2-\frac{\ln\ln|V|}{\ln|V|}(1-o(1))$$ [Halperin '00]

$$2-\Theta\left(\frac{1}{\sqrt{\log|V|}}\right)$$ [Karakostas '04]

PROPERTIES OF THE MIN VERTEX COVER (1)

Def. An independent set of G=(V,E) is a set of nodes of V, no two of which are adjacent.

Th. A set of nodes V' is a vertex cover if and only if its complement V-V' is an independent set.

Proof.

V' VC =>V-V' IS

- By contradiction. If in V-V' there exist two adjacent nodes, then the corresponding edge is not covered. A contradiction.

V-V' IS => V' VC

- By contradiction. If there exists an edge e that is not covered by any node in V', the nodes incident to e are adjacent in V-V'. A contradiction.

PROPERTIES OF THE MIN VERTEX COVER (2)

- **Cor.** The number of nodes of a graph is equal to the size of its min vertex cover plus the size of a maximum independent set [Gallai '59].
- Nevertheless, these two problems are not equivalent, from an approximation point of view: IS cannot be approximated by any constant [Håstad '99].

PROPERTIES OF THE MIN VERTEX COVER (3)

Def. A matching of $G=(V,E)$ is a subset M of E without common nodes.

Th. Let M be a matching of G and C a vertex cover for G. Then $|M|\leq|C|$.

Proof. C is a vertex cover, so it must cover all edges in M.

From the other side, by definition of matching, for each edge in M, at least one of its endpoints must be in C.

So $|M|\leq|C|$. ■

PROPERTIES OF THE MIN VERTEX COVER (4)

Cor. Let M be a matching of G and C a vertex cover for G. If|M|=|C| then M is a maximum matching and C is a minimum vertex cover.

It is polynomial to compute a max matching.

Could we think to solve the min vertex cover passing through the max matching problem?

No, because:

Fact: The reverse of the previous corollary is false.

Anyway, an algorithm based on this property has been proposed: ...

PROPERTIES OF THE MIN VERTEX COVER (5)

Algorithm New-Approx-VC(G) [Gavril '79]

- Compute a max matching M
- V'= empty set
- For each e in M
  - Insert in V' both the endpoints of e
- Return V'

Time complexity:

It depends on the computation of the max matching:

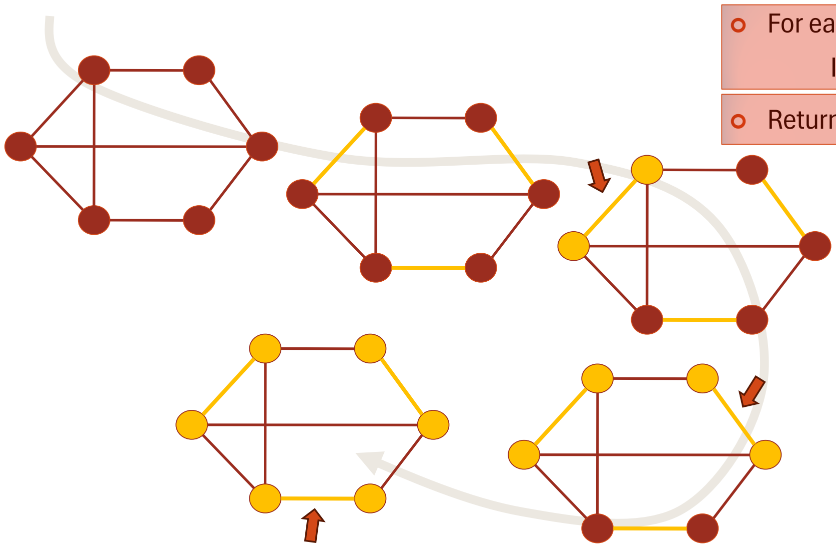
$O(n^4)$ [Edmonds '65]

$O(m\sqrt{n})$ [Micali & Vazirani '80]

PROPERTIES OF THE MIN VERTEX COVER (6)

Algorithm New-Approx-VC(G) [Gavril '79]

- Compute a maximum matching M
- V'= empty set
- For each e in M
  - Insert in V' both the endpoints of e
- Return V'



PROPERTIES OF THE MIN VERTEX COVER (7)

Th. The set V' returned by New-Approx-VC is a vertex cover for G such that $|V'| \leq 2\,|V^*|$.

Proof. V' is a vertex cover indeed any (u,v) in G is:

- either in M and hence both its endpoints are in V'
- or is in E\M, and at least one of its endpoint is in V' (otherwise it could be added to M, that is maximum).

By construction $|V'|=2|M|$.

Notice that each $(u,v) \in M$ must have at least one of its endpoints in any min. vertex cover: $|M| \leq |V^*|$

Putting together the two relations: $|V'| = 2\,|M| \leq 2\,|V^*|$ ■

PROPERTIES OF THE MIN VERTEX COVER (8)

If G is bipartite, a stronger relation holds between min vertex cover and max matching, so deducing that:

VC is polynomially solvable on bipartite graphs.

In particular, the previous algorithm can be modified in order to produce an optimal solution (time complexity: at least $O(m\sqrt{n})$ – it depends on the computation of the maximum matching).

PROPERTIES OF THE MIN VERTEX COVER (9)

König's Th. ['31] (Egervàry ['31]): In any bipartite graph, the number of edges in a maximum matching equals the number of vertices in a minimum vertex cover.

Proof. Omitted…

ANOTHER APPLICATION (1)

brain connectivity networks [Candemir & Akram '23]

The human brain contains a highly complex network structure consisting of billions of neurons and the synaptic connections these neurons form with each other.

Brain connectivity refers to the intricate network of structural and functional connections between different regions of the brain.

ANOTHER APPLICATION (2)

Structural connectivity plays a fundamental role in understanding the anatomical organization with physical connections of the brain, especially between the neighboring regions.

Functional connectivity points out the relations and interactions of distinct regions in the human brain.

ANOTHER APPLICATION (3)

Studying brain connectivity is crucial for understanding how different brain regions work together to support various cognitive processes, (perception, attention, memory…)

It provides insights into the functional integration of brain regions, the formation of specialized networks, and the underlying mechanisms of brain function.

Moreover, disruptions in brain connectivity are related with neurological and psychiatric disorders.

ANOTHER APPLICATION (4)

Brain connectivity networks can be analyzed using graph theory.

<u>Brain connectivity networks</u> can be represented as <u>graphs</u>, where:

- brain regions are nodes,
- the connections between them are edges.

By studying the graph parameters of brain connectivity networks, researchers can gain insights into the underlying architecture and dynamics of the human brain.

ANOTHER APPLICATION (5)

In the context of brain connectivity networks, the min vertex cover problem can be used to identify a minimal set of brain regions that can adequately represent the entire network, allowing for efficient analysis, feature selection, and the identification of biomarkers associated with neurological disorders.

The changes of the min vertex cover can indicate the functional connectivity changes of the brain in healthy aging.

A RELATED PROBLEM: EVC (1)

Dynamic network security / fault-tolerance model:
Given a network, deploy a min set of guards at the nodes, so that if there is an attack (or fault) on a single link at any time, a guard is available at the end of the link, and move across the link to defend (or repair) the attack (or fault). Simultaneously, the remaining guards reconfigure themselves, possibly by repositioning on an adjacent node, so that any later attack (or fault) can also be protected. Thus, the model guarantees protection against single link attacks/failures ad-infinitum.

A RELATED PROBLEM: EVC (2)

Eternal vertex cover:

- the network is modeled as a graph;
- at most one defender is located at each node;
- an attacker can attack edges;
- a defender protects all the edges incident to the nodes where gards are located; a guard must move along the attacked edge to defend it;
- all the other guards can move traversing a single edge.

A RELATED PROBLEM: EVC (3)

Given the subset of nodes with guards on them at a certain instant, if it is not a vertex cover, then the attacker can target any of the uncovered edges to win the game.

Therefore, the defender must always "reconfigure" one vertex cover into another in response to any attack.

So, if $\boldsymbol{\alpha}(G)$ is the cardinality of a min VC and $\boldsymbol{\alpha}^{\infty}(G)$ the cardinality of a min eternal VC:

$$\boldsymbol{\alpha}(G) \leq \boldsymbol{\alpha}^{\infty}(G)$$

A RELATED PROBLEM: EVC (4)

Theorem. Let G be a connected graph and let V′ be a vertex cover inducing a <u>connected</u> subgraph. Then $\alpha^{\infty}(G) \le |V'|+1$.

Sketch of Proof.

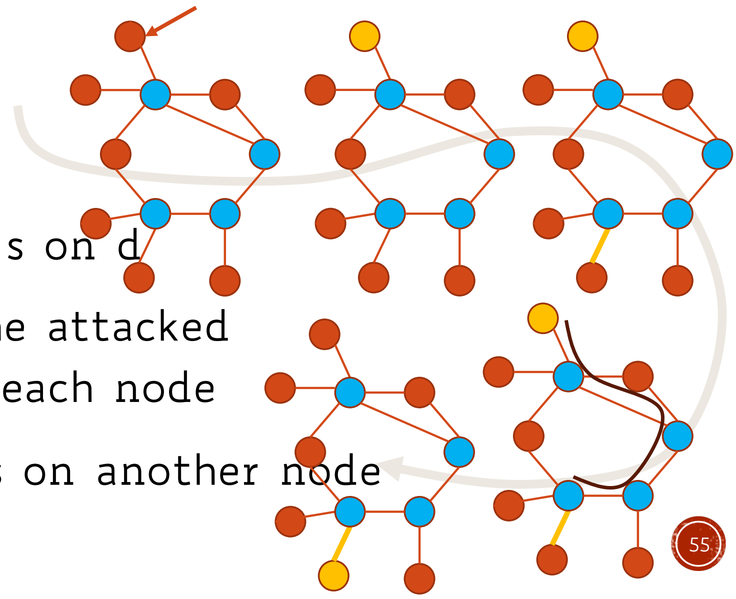
Select d in V\V′.

Put a shadow guard s on d

P=path from d to the attacked edge with gards on each node

The shadow guard is on another node

Repeat…



A RELATED PROBLEM: EVC (5)

Corollary. Let G be a connected graph and let V′ be a vertex cover inducing a subgraph. with k connected components. Then $\alpha^{\infty}(G) \le |V'|+k$.

Since V′ can induce at most |V′| connected components, $\alpha^{\infty}(G) \le 2|V'|$.

It follows that:

$$\alpha^{\infty}(G) \le 2\alpha(G)$$

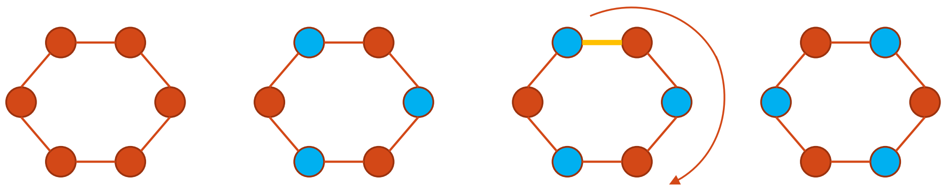
A RELATED PROBLEM: EVC (6)

Theorem. Let G be a connected graph. Then:

$$\alpha(G) \leq \alpha^{\infty}(G) \leq 2\alpha(G).$$

Theorem. For any $n \geq 3$, $\alpha^{\infty}(C_n) = \alpha(C_n) = \lceil n/2 \rceil$.

Proof.



A RELATED PROBLEM: EVC (7)

Theorem. For any $n \geq 1$, $\alpha^{\infty}(P_n) = n-1$.

Sketch of proof. By contradiction, if two nodes are outside the EVC, it is possible to design an attack strategy that move them until they become endpoints of the same edge and the attacker wins.

☞ Some students' lessons and master theses are available on this topic