The Decycling Number of Graphs

Hung-Lin Fu
Department of Applied Mathematics
National Chiao Tung University
Hsin Chu, Taiwan

Joint work with Huilan Chang, Hong-Bin Chen,
Min-Yun Lien and Chih-Huai Shih

Definition and Applications

Definition (Decycling Problem)

Given a directed/undirected graph $G = (V, E)$, find a minimum set $D \subset V$ such that $G \setminus D$ is acyclic.

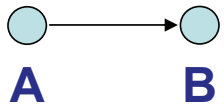
- Has applications in
  - Deadlock prevention in operating systems (Wang et al. 1985; Silberschatz et al. 2003)

Definition and Applications

Definition (Decycling Problem)

Given a directed/undirected graph $G = (V, E)$, find a minimum set $D \subset V$ such that $G \setminus D$ is acyclic.

- Has applications in
  - Deadlock prevention in operating systems (Wang et al. 1985; Silberschatz et al. 2003)
    Example: An operating system schedules different processes.



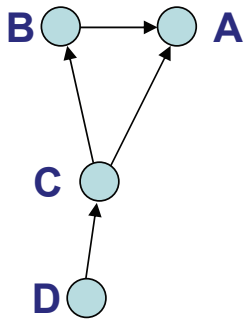
Process A is waiting for the resource on Process B so it can't release its own resource.

Definition and Applications

Definition (Decycling Problem)

Given a directed/undirected graph $G = (V, E)$, find a minimum set $D \subset V$ such that $G \setminus D$ is acyclic.

- Has applications in
  - Deadlock prevention in operating systems (Wang et al. 1985; Silberschatz et al. 2003)
    Example: An operating system schedules different processes.



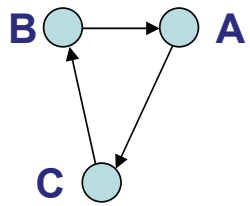
Process A is waiting for no resource so it can release its resource.

Definition and Applications

Definition (Decycling Problem)

Given a directed/undirected graph $G = (V, E)$, find a minimum set $D \subset V$ such that $G \setminus D$ is acyclic.

- Has applications in
  - Deadlock prevention in operating systems (Wang et al. 1985; Silberschatz et al. 2003)
    Example: An operating system schedules different processes.



Deadlock:
Competing actions are each waiting for the other to finish, and thus neither ever does.
Solution:
Remove some processes to **break such cycles** and put them in a waiting queue.

Applications

- Monopolies in synchronous distributed systems (Peleg 1998; Peleg 2002)

Applications

- Monopolies in synchronous distributed systems (Peleg 1998; Peleg 2002)
  - Vertices are colored YES or NO.

Applications

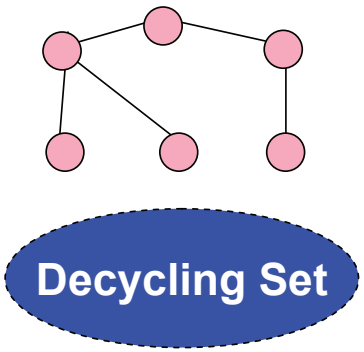
- Monopolies in synchronous distributed systems (Peleg 1998; Peleg 2002)
  - Vertices are colored YES or NO.
  - Monotone synchronous system: at each step a NO vertex changes to YES if more than half of its neighbors are YES.

Applications

- Monopolies in synchronous distributed systems (Peleg 1998; Peleg 2002)
  - Vertices are colored YES or NO.
  - Monotone synchronous system: at each step a NO vertex changes to YES if more than half of its neighbors are YES.
  - Problem: set the minimum number of vertices YES at beginning such that all vertices become YES eventually.

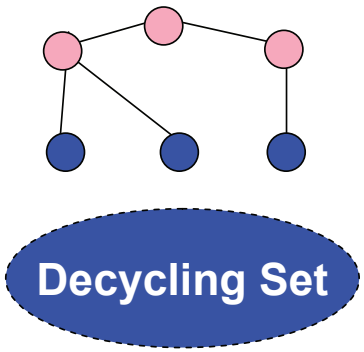
Applications

- Monopolies in synchronous distributed systems (Peleg 1998; Peleg 2002)
  - Vertices are colored YES or NO.
  - Monotone synchronous system: at each step a NO vertex changes to YES if more than half of its neighbors are YES.
  - Problem: set the minimum number of vertices YES at beginning such that all vertices become YES eventually.
  - A decycling set could be the only choice!
    Example: A 4-regular graph (such as toroidal mesh network).



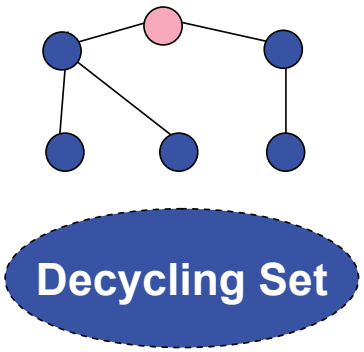
Applications

- Monopolies in synchronous distributed systems (Peleg 1998; Peleg 2002)
  - Vertices are colored YES or NO.
  - Monotone synchronous system: at each step a NO vertex changes to YES if more than half of its neighbors are YES.
  - Problem: set the minimum number of vertices YES at beginning such that all vertices become YES eventually.
  - A decycling set could be the only choice!
    Example: A 4-regular graph (such as toroidal mesh network).



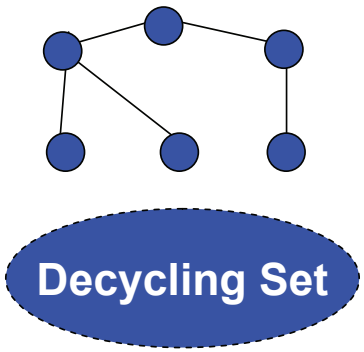
Applications

- Monopolies in synchronous distributed systems (Peleg 1998; Peleg 2002)
  - Vertices are colored YES or NO.
  - Monotone synchronous system: at each step a NO vertex changes to YES if more than half of its neighbors are YES.
  - Problem: set the minimum number of vertices YES at beginning such that all vertices become YES eventually.
  - A decycling set could be the only choice!
    Example: A 4-regular graph (such as toroidal mesh network).



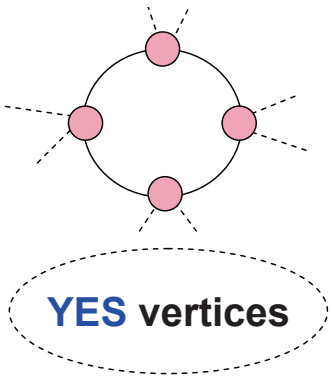
Applications

- Monopolies in synchronous distributed systems (Peleg 1998; Peleg 2002)
  - Vertices are colored YES or NO.
  - Monotone synchronous system: at each step a NO vertex changes to YES if more than half of its neighbors are YES.
  - Problem: set the minimum number of vertices YES at beginning such that all vertices become YES eventually.
  - A decycling set could be the only choice!
    Example: A 4-regular graph (such as toroidal mesh network).



Applications

- Monopolies in synchronous distributed systems (Peleg 1998; Peleg 2002)
  - Vertices are colored YES or NO.
  - Monotone synchronous system: at each step a NO vertex changes to YES if more than half of its neighbors are YES.
  - Problem: set the minimum number of vertices YES at beginning such that all vertices become YES eventually.
  - A decycling set could be the only choice!
    Example: A 4-regular graph (such as toroidal mesh network).



NO vertices on the cycle remain NO.
Failed!

Complexity

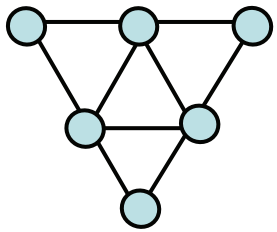
- Has been extensively studied.
  - NP-hard
    Reduction from VERTEX COVER (R. Karp 1972).
    Even for planar graphs.

Related Problems

- Is related or equivalent to
  - Feedback vertex set problem (Wang et al. 1985).
  - Hitting cycle problem
  - Maximum induced forest problem
    (Erdös, Saks and Sós 1986 Maximum induced trees in graphs).
    At least $\frac{71n+72}{128}$ for triangle-free n-vertex planar graphs (Kowalik et al. 2010).

Relations with Cycle Packing Number

- Is often compared with the following graph parameter.
  - Cycle Packing Number $c(G)$: the maximum number of vertex-disjoint cycles of G.



$c(G)=1$

  - **Definition:** $\nabla(G) :=$ the decycling number (minimum size of decycling set) of G.

Relations with Cycle Packing Number

- Dirac and Gallai had interest in the relations between $c(G)$ and $\nabla(G)$.
- It is clear that $c(G) \leq \nabla G)$.

Relations with Cycle Packing Number

- Dirac and Gallai had interest in the relations between $c(G)$ and $\nabla(G)$.
- It is clear that $c(G) \leq \nabla G)$.

- **Definition:** $\nabla(k) := \max\{\nabla(G) : c(G) = k\}$.

Relations with Cycle Packing Number

- Dirac and Gallai had interest in the relations between $c(G)$ and $\nabla(G)$.
- It is clear that $c(G) \leq \nabla G)$.

- **Definition:** $\nabla(k) := \max\{\nabla(G) : c(G) = k\}$.
- $\nabla(1) = 3$; $c(K_5) = 1$ and $\nabla(K_5) = 3$ (Bollobás 1964).
- $\nabla(2) = 6$ and $9 \leq \nabla(3) \leq 12$ (Voss 1968).

Relations with Cycle Packing Number

- Dirac and Gallai had interest in the relations between $c(G)$ and $\nabla(G)$.
- It is clear that $c(G) \leq \nabla G)$.

- **Definition:** $\nabla(k) := \max\{\nabla(G) : c(G) = k\}$.
- $\nabla(1) = 3$; $c(K_5) = 1$ and $\nabla(K_5) = 3$ (Bollobás 1964).
- $\nabla(2) = 6$ and $9 \leq \nabla(3) \leq 12$ (Voss 1968).
- $c_1 k \log k \leq \nabla(k) \leq c_2 k \log k$ for some constants c_1 and c_2 (Erdös and Pósa 1964).

Relations with Cycle Packing Number

- Consider planar graphs:

Relations with Cycle Packing Number

- Consider planar graphs:

Jones' Conjecture (Kloks, Lee and Liu 2002)

For every planar graph G, $\nabla(G) \leq 2c(G)$.

Relations with Cycle Packing Number

- Consider planar graphs:

Jones' Conjecture (Kloks, Lee and Liu 2002)

For every planar graph G, $\nabla(G) \leq 2c(G)$.

Theorem (Chen, Fu and Shih 2010)

For every planar graph G, $\nabla(G) \leq 3c(G)$.

- Lemma

Every 2-edge-connected triangle-free planar graph G with minimum degree 3 has either a C_4 containing a 3-vertex or a C_5 containing at least four 3-vertices.

Proof. By discharging method.

Decycling number of outerplanar graphs

Consider outerplanar graphs:

Theorem (Kloks, Lee and Liu 2002)

For every outerplanar graph G, $\nabla(G) \leq 2c(G)$.

- An outerplanar graph G is called *lower-extremal* if $\nabla(G) = c(G)$ and *upper-extremal* if $\nabla(G) = 2c(G)$.

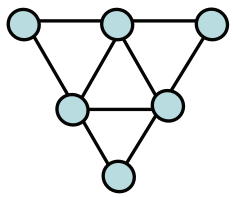
Decycling number of outerplanar graphs

Consider outerplanar graphs:

Theorem (Kloks, Lee and Liu 2002)

For every outerplanar graph G, $\nabla(G) \leq 2c(G)$.

- An outerplanar graph G is called *lower-extremal* if $\nabla(G) = c(G)$ and *upper-extremal* if $\nabla(G) = 2c(G)$.
- Upper-Extremal Results:
  - We define a *sun graph* S_3 as follows.



  - $\nabla(S_3) = 2 = 2c(S_3)$.

Decycling number of outerplanar graphs

Theorem (Chang, Fu, Lien, 2011)

An outerplanar graph G is upper-extremal if and only if G is an S_3-tree.

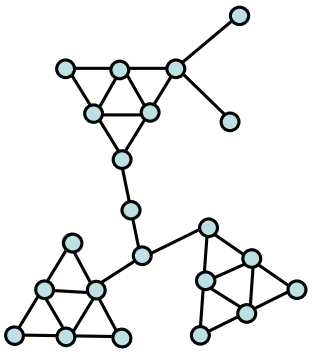
- A graph is an *S_3-tree of order t* if it has exactly t vertex-disjoint S_3-subdivisions and every edge not on these S_3-subdivisions belongs to no cycle.

Decycling number of outerplanar graphs

Theorem (Chang, Fu, Lien, 2011)

An outerplanar graph G is upper-extremal if and only if G is an S_3-tree.

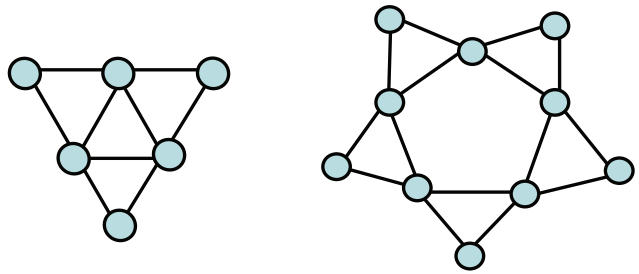
- A graph is an *S_3-tree of order t* if it has exactly t vertex-disjoint S_3-subdivisions and every edge not on these S_3-subdivisions belongs to no cycle.
- Example:



An S_3-tree G of order 3, where $\tau(G) = 6 = 2c(G)$.

Decycling number of outerplanar graphs

- Lower-Extremal Results:
  - The following graphs are NOT lower-extremal ($\nabla(G) \neq c(G)$):
    Sun graphs S_k with odd number k:



S_3 S_5

  - $\nabla(S_k) = \lceil \frac{k}{2} \rceil$ and $c(S_k) = \lfloor \frac{k}{2} \rfloor$.

Decycling number of outerplanar graphs

Theorem (Chang, Fu, Lien, 2011)

For an outerplanar graph G, if G has no S_k-subdivision for all odd number k, then G is lower-extremal.

Decycling number of outerplanar graphs

Theorem (Chang, Fu, Lien, 2011)

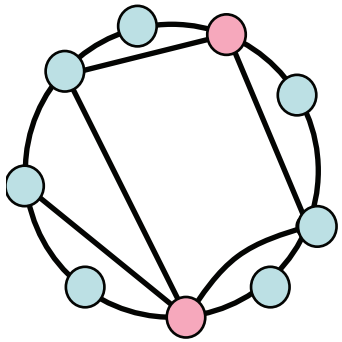
For an outerplanar graph G, if G has no S_k-subdivision for all odd number k, then G is lower-extremal.

Lemma

If G is a 2-connected outerplanar graph with no S_k-subdivision for all odd number k, then G is lower-extremal.

(Proved by induction on $|E(G)|$.)

- Example:



References

- Albertson MO, Berman DM (1979) A conjecture on planar graphs, Bondy JA, Murty USR, Graph theory and related topics 357.
- Erdös P, Saks M, Sós VT (1986) Maximum induced trees in graphs. J Combin Theory Ser B 41:61-79.
- Bau S, Beineke LW, Vandell RC (1998) Decycling snakes. Congr Numer 134:79-87.
- Bodlaender HL (1994) On disjoint cycles. Int J Found Comput Sci 5:59-68.

- Erdös P, Saks M, Sós VT (1986) Maximum induced trees in graphs. J Combin Theory Ser B 41:61-79.


Festa P, Pardalos PM, Resende MGC (2000) Feedback set problems, Handbook of Combinatorial Optimization, Du D-Z, Pardalos PM, Eds, Kluwer Academic Publishers, Supplement A, pp 209-259.


Kloks T, Lee C-M, Liu J (2002) New algorithms for k-face cover, k-feedback vertex set, and k-disjoint cycles on plane and planar graphs. in Proceedings of the 28th International Workshop on Graph-Theoretic Concepts in Computer Science (WG 2002) Springer-Verlag, 2573:282-295.


F. R. Madelaine and I. A. Stewart, Improved upper and lower bounds on the feedback vertex numbers of grids and butterflies, Discrete Math., 308 (2008) 4144-4164.


D. A. Pike and Y. Zou, Decycling Cartesian products of two cycles, SIAM J. Discrete Math., 19 (2005) 651-663.