Acquisition Parameters of Graphs

Douglas B. West

Department of Mathematics
University of Illinois at Urbana-Champaign
west@math.uiuc.edu

Results with or by
Timothy D. LeSaulnier, Kevin G. Milans,
Noah Prince, Chris Stocker, Paul S. Wenger,
Leslie Wiglesworth, Pratik Worah

The Problem

Model: People start with one vote. Some are friends.

A person can give his or her votes to a friend
if the friend has at least as many votes.

The Problem

Model: People start with one vote. Some are friends.

A person can give his or her votes to a friend
if the friend has at least as many votes.

Can one person acquire all the votes?

The Problem

Votes can go to a friend having at least as many votes.

Can one person acquire all the votes?

Model: (Wenger) Each city starts with one regiment.

The troops in one city can withdraw to a neighboring city if that city already has at least as many troops.

The Problem

Votes can go to a friend having at least as many votes.

Can one person acquire all the votes?

Model: (Wenger) Each city starts with one regiment.

The troops in one city can withdraw to a neighboring city if that city already has at least as many troops.

Can the troops be gathered in a single city?

The Problem

Votes can go to a friend having at least as many votes.

Can one person acquire all the votes?

Model: (Wenger) Each city starts with one regiment.

The troops in one city can withdraw to a neighboring city if that city already has at least as many troops.

Can the troops be gathered in a single city?

Def. total aquisition move = transfer all weight from u to a neighbor v; allowed if currently $w(u) \leq w(v)$.

The Problem

Votes can go to a friend having at least as many votes.

Can one person acquire all the votes?

Model: (Wenger) Each city starts with one regiment.

The troops in one city can withdraw to a neighboring city if that city already has at least as many troops.

Can the troops be gathered in a single city?

Def. total aquisition move = transfer all weight from u to a neighbor v; allowed if currently $w(u) \leq w(v)$.

• Process ends when the vertices with $w(v) > 0$ form an independent set, after finitely many moves.

The Problem

Votes can go to a friend having at least as many votes.

Can one person acquire all the votes?

Model: (Wenger) Each city starts with one regiment.

The troops in one city can withdraw to a neighboring city if that city already has at least as many troops.

Can the troops be gathered in a single city?

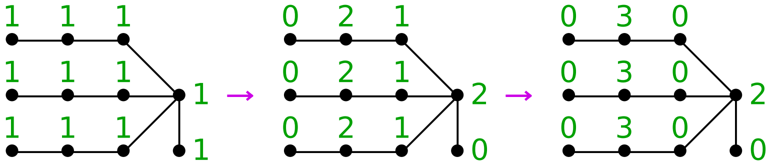
Def. total aquisition move = transfer all weight from u to a neighbor v; allowed if currently $w(u) \le w(v)$.

• Process ends when the vertices with $w(v) > 0$ form an independent set, after finitely many moves.

Def. total aquisition number $a_t(G)$ = min size of the final indep. set when each vertex starts with weight 1.

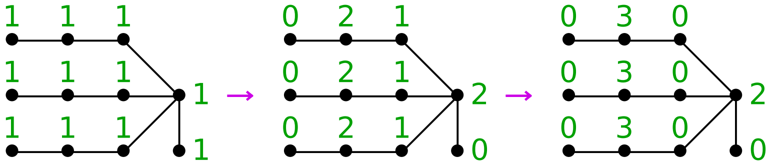
An Example

Ex. $a_t(T) = 4$, 11 vertices.



An Example

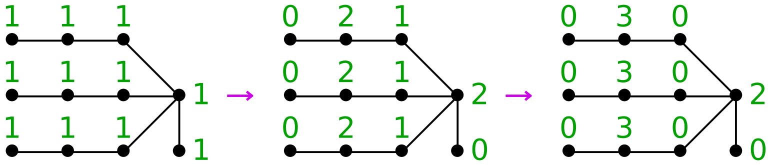
Ex. $a_t(T) = 4$, 11 vertices.



Alternative Models — Still start with weight 1 on all, and moving wt from u to v requires $w(v) \geq w(u)$.

An Example

Ex. $a_t(T) = 4$, 11 vertices.

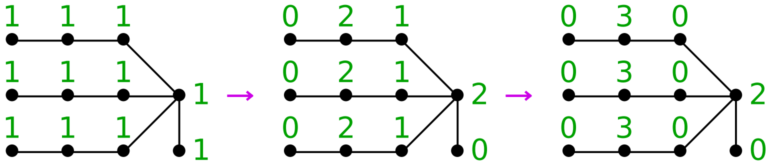


Alternative Models — Still start with weight 1 on all, and moving wt from u to v requires $w(v) \geq w(u)$.

unit acquisition: move any integer amount — $a_u(G)$

An Example

Ex. $a_t(T) = 4$, 11 vertices.



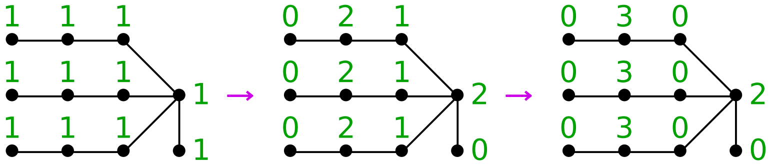
Alternative Models — Still start with weight 1 on all, and moving wt from u to v requires $w(v) \geq w(u)$.

unit acquisition: move any integer amount — $a_u(G)$

fractional acquisition: move any positive amt — $a_f(G)$

An Example

Ex. $a_t(T) = 4$, 11 vertices.



Alternative Models — Still start with weight 1 on all, and moving wt from u to v requires $w(v) \geq w(u)$.

unit acquisition: move any integer amount — $a_u(G)$

fractional acquisition: move any positive amt — $a_f(G)$

game acquisition: move all weight, but two players Min and Max alternate moves — $a_g(G)$

Total Acquisition Number — Extremal Problem

All our graphs have n vertices.

Thm. (Lampert–Slater [1995]) If G is connected and nontrivial, then $a_t(G) \le (n+1)/3$, and this is sharp.

Total Acquisition Number — Extremal Problem

All our graphs have n vertices.

Thm. (Lampert–Slater [1995]) If G is connected and nontrivial, then $a_t(G) \le (n+1)/3$, and this is sharp.

Pf. Since $H \subseteq G \Rightarrow a_t(H) \ge a_t(G)$,
it suffices to prove the bound for trees.

Total Acquisition Number — Extremal Problem

All our graphs have n vertices.

Thm. (Lampert–Slater [1995]) If G is connected and nontrivial, then $a_t(G) \le (n+1)/3$, and this is sharp.

Pf. Since $H \subseteq G \Rightarrow a_t(H) \ge a_t(G)$,
it suffices to prove the bound for trees.

Idea: Induction on n. Find a subtree T' with m vertices, $a_t(T') \le m/3$, and $T - V(T')$ connected.

Total Acquisition Number — Extremal Problem

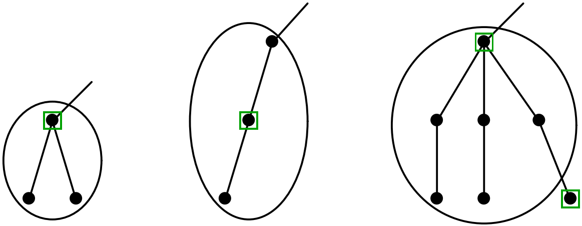
All our graphs have n vertices.

Thm. (Lampert–Slater [1995]) If G is connected and nontrivial, then $a_t(G) \leq (n+1)/3$, and this is sharp.

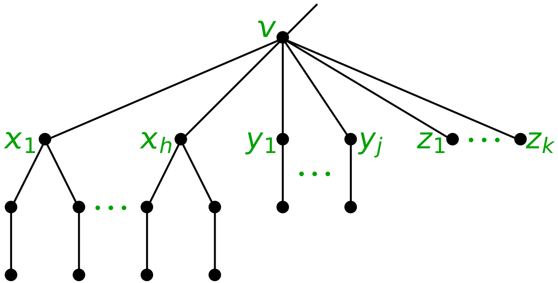
Pf. Since $H \subseteq G \Rightarrow a_t(H) \geq a_t(G)$,
it suffices to prove the bound for trees.

Idea: Induction on n. Find a subtree T' with m vertices, $a_t(T') \leq m/3$, and $T - V(T')$ connected.

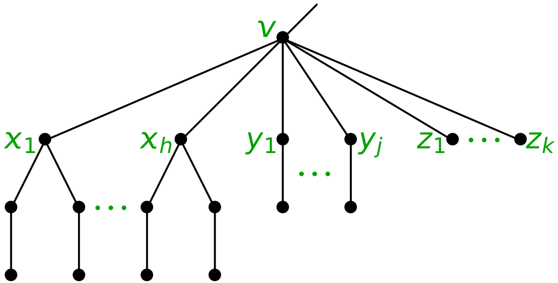
Easy cases:



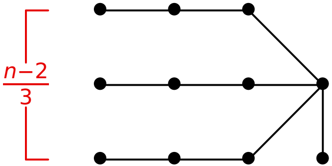
Hard Case:



Hard Case:



Sharpness:



Lower bound — An Obstruction

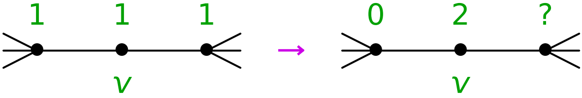
- View each initial unit of weight as a token or chip.

Lem. For total or unit acquisition, at most one chip can reach and leave a degree-2 vertex v, and it can only be the chip from a neighbor.

Lower bound — An Obstruction

- View each initial unit of weight as a token or chip.

Lem. For total or unit acquisition, at most one chip can reach and leave a degree-2 vertex v, and it can only be the chip from a neighbor.

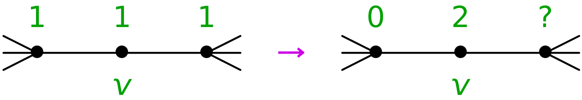


Pf. Such a chip must reach v before any departs v. The first to arrive must have been the only ($\therefore$ original) chip on a neighbor. That neighbor is now dead. ∎

Lower bound — An Obstruction

- View each initial unit of weight as a token or chip.

Lem. For total or unit acquisition, at most one chip can reach and leave a degree-2 vertex v, and it can only be the chip from a neighbor.



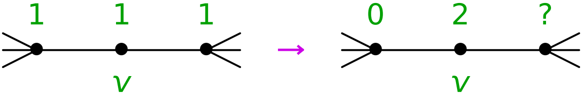
Pf. Such a chip must reach v before any departs v. The first to arrive must have been the only ($\therefore$ original) chip on a neighbor. That neighbor is now dead. ■

Cor. If x and y are vertices of a tree, and the x, y-path has an internal vertex of degree 2 adjacent to neither of them, then the chips from x and y can never combine.

Lower bound — An Obstruction

- View each initial unit of weight as a token or chip.

Lem. For total or unit acquisition, at most one chip can reach and leave a degree-2 vertex v, and it can only be the chip from a neighbor.



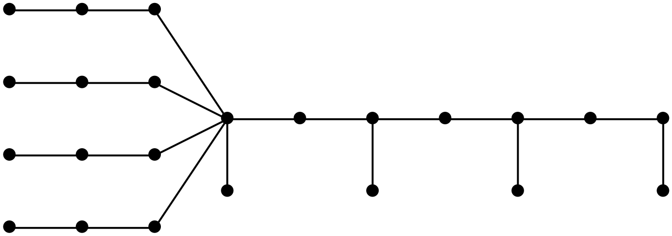
Pf. Such a chip must reach v before any departs v. The first to arrive must have been the only ($\therefore$ original) chip on a neighbor. That neighbor is now dead. ∎

Cor. If x and y are vertices of a tree, and the x,y-path has an internal vertex of degree 2 adjacent to neither of them, then the chips from x and y can never combine.

Thm. For $d \geq 3$ and $k \geq 6$, there is a tree T with $\Delta(T) = d$, $\mathrm{diam} T \geq k$, and $a_t(T) = (n+1)/3$.

Trees with $a_t(G)$ Large

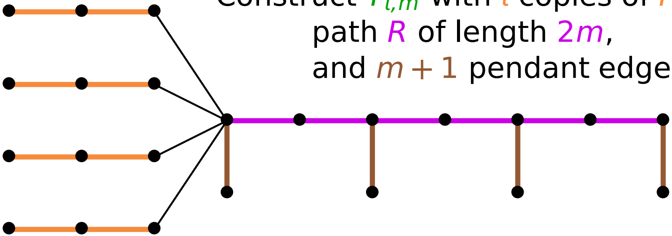
Ex. The tree $T_{4,3}$.



Trees with $a_t(G)$ Large

Ex. The tree $T_{4,3}$.

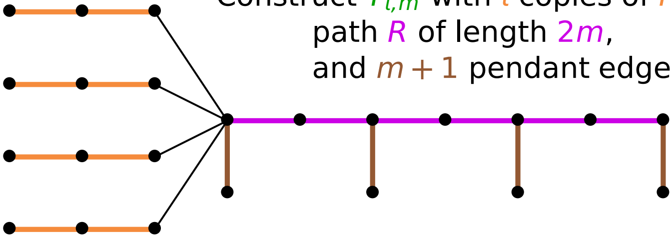
Construct $T_{l,m}$ with l copies of P_3,
path R of length $2m$,
and $m+1$ pendant edges.



Trees with $a_t(G)$ Large

Ex. The tree $T_{4,3}$.

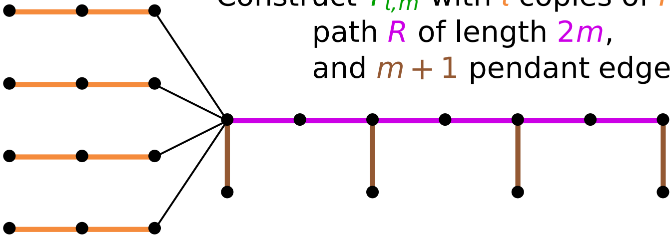
Construct $T_{l,m}$ with l copies of P_3,
path R of length $2m$,
and $m+1$ pendant edges.



#vertices = $3l+3m+2$, #leaves = $l+m+1$
diameter = $2m+4$, maxdegree = $l+2$.

Trees with $a_t(G)$ Large

Ex. The tree $T_{4,3}$.



Construct $T_{l,m}$ with l copies of P_3,
path R of length $2m$,
and $m+1$ pendant edges.

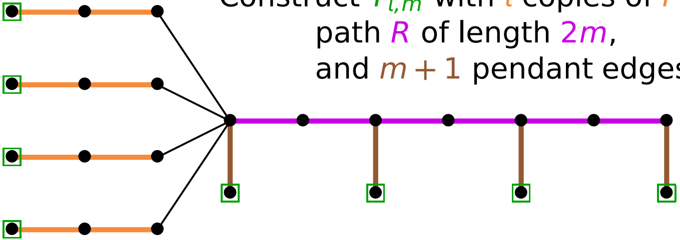
#vertices = $3l+3m+2$, #leaves = $l+m+1$
diameter = $2m+4$, maxdegree = $l+2$.

Prop. $a_t(T_{l,m}) = l+m+1 = (n+1)/3$.

Trees with $a_t(G)$ Large

Ex. The tree $T_{4,3}$.

Construct $T_{l,m}$ with l copies of P_3,
path R of length $2m$,
and $m+1$ pendant edges.



#vertices = $3l+3m+2$, #leaves = $l+m+1$
diameter = $2m+4$, maxdegree = $l+2$.

Prop. $a_t(T_{l,m}) = l+m+1 = (n+1)/3$.

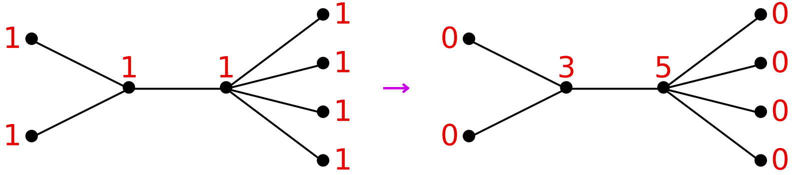
Pf. Chips from marked vertices can never combine. ■

Bounds for Trees

- For diameter 6 or higher, $\max a_t(T) = (n+1)/3$.

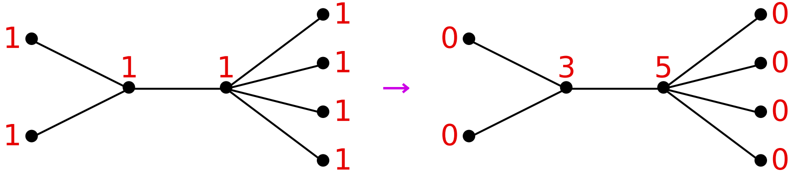
Bounds for Trees

- For diameter 6 or higher, $\max a_t(T) = (n+1)/3$.
- For diameter 2 or 3, $a_t(T) = 1$. "double-stars"



Bounds for Trees

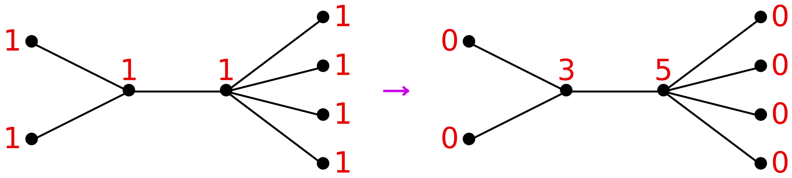
- For diameter 6 or higher, $\max a_t(T) = (n+1)/3$.
- For diameter 2 or 3, $a_t(T) = 1$. "double-stars"



What about diameter 4 and diameter 5?

Bounds for Trees

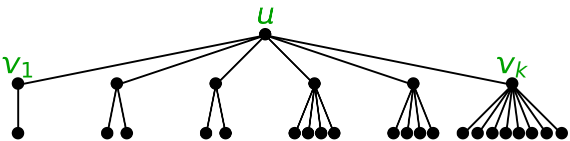
- For diameter 6 or higher, $\max a_t(T) = (n+1)/3$.
- For diameter 2 or 3, $a_t(T) = 1$. "double-stars"



What about diameter 4 and diameter 5?

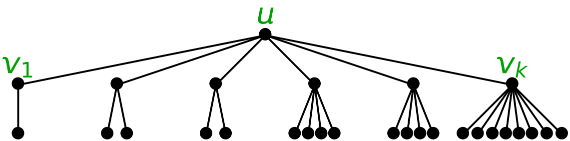
Thm. For diameter 4 or 5, $\max a_t(T) = \Theta(\sqrt{n \lg n})$.

Trees with Diameter 4, Upper Bound



Thm. $a_t(T) \leq 2\sqrt{n \lg(2n)}$.

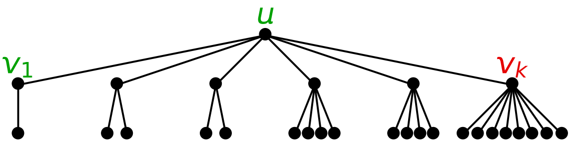
Trees with Diameter 4, Upper Bound



Thm. $a_t(T) \le 2\sqrt{n \lg(2n)}$.

Pf. Case 1: $k \le 2\sqrt{n \lg(2n)} \Rightarrow N(u)$ absorbs all.

Trees with Diameter 4, Upper Bound

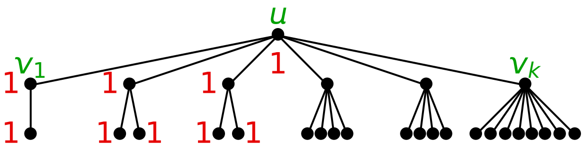


Thm. $a_t(T) \leq 2\sqrt{n \lg(2n)}$.

Pf. Case 1: $k \leq 2\sqrt{n \lg(2n)} \Rightarrow N(u)$ absorbs all.

Case 2: $d(v_k) \geq 2\sqrt{n} \Rightarrow a_t(T) \leq 1 + 2\sqrt{(n - 2\sqrt{n}) \lg(2n)}$.

Trees with Diameter 4, Upper Bound



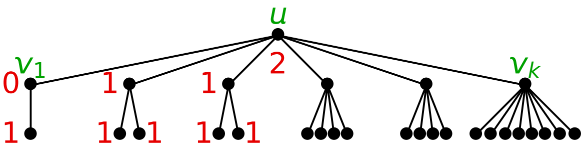
Thm. $a_t(T) \le 2\sqrt{n \lg(2n)}$.

Pf. Case 1: $k \le 2\sqrt{n \lg(2n)} \Rightarrow N(u)$ absorbs all.

Case 2: $d(v_k) \ge 2\sqrt{n} \Rightarrow a_t(T) \le 1 + 2\sqrt{(n-2\sqrt{n})\lg(2n)}$.

Case 3: let w_i = weight on u before processing v_i; algorithm gives u weight $\min\{w_i, d(v_i)\}$ from v_i.

Trees with Diameter 4, Upper Bound



Thm. $a_t(T) \le 2\sqrt{n \lg(2n)}$.

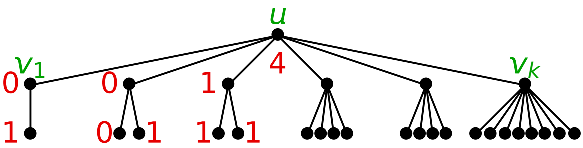
Pf. Case 1: $k \le 2\sqrt{n \lg(2n)} \Rightarrow N(u)$ absorbs all.

Case 2: $d(v_k) \ge 2\sqrt{n} \Rightarrow a_t(T) \le 1 + 2\sqrt{(n-2\sqrt{n}) \lg(2n)}$.

Case 3: let w_i = weight on u before processing v_i;
algorithm gives u weight $\min\{w_i, d(v_i)\}$ from v_i.

Let $S = \{i\colon d(v_i) > w_i\}$; weight doubles, so $|S| \le \lg n$.

Trees with Diameter 4, Upper Bound



Thm. $a_t(T) \le 2\sqrt{n \lg(2n)}$.

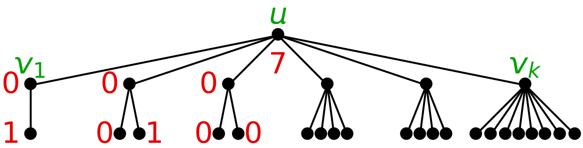
Pf. Case 1: $k \le 2\sqrt{n \lg(2n)} \Rightarrow N(u)$ absorbs all.

Case 2: $d(v_k) \ge 2\sqrt{n} \Rightarrow a_t(T) \le 1 + 2\sqrt{(n - 2\sqrt{n}) \lg(2n)}$.

Case 3: let w_i = weight on u before processing v_i;
algorithm gives u weight $\min\{w_i, d(v_i)\}$ from v_i.

Let $S = \{i\colon d(v_i) > w_i\}$; weight doubles, so $|S| \le \lg n$.
Some of $N(v_i)$ stays when $i \in S$. Let $m = \max S$.

Trees with Diameter 4, Upper Bound



Thm. $a_t(T) \le 2\sqrt{n \lg(2n)}$.

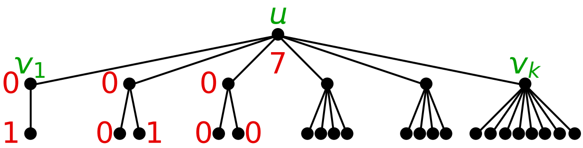
Pf. Case 1: $k \le 2\sqrt{n \lg(2n)} \Rightarrow N(u)$ absorbs all.

Case 2: $d(v_k) \ge 2\sqrt{n} \Rightarrow a_t(T) \le 1 + 2\sqrt{(n - 2\sqrt{n}) \lg(2n)}$.

Case 3: let w_i = weight on u before processing v_i;
algorithm gives u weight $\min\{w_i, d(v_i)\}$ from v_i.

Let $S = \{i \colon d(v_i) > w_i\}$; weight doubles, so $|S| \le \lg n$.
Some of $N(v_i)$ stays when $i \in S$. Let $m = \max S$.

Trees with Diameter 4, Upper Bound



Thm. $a_t(T) \le 2\sqrt{n \lg(2n)}$.

Pf. Case 1: $k \le 2\sqrt{n \lg(2n)} \Rightarrow N(u)$ absorbs all.

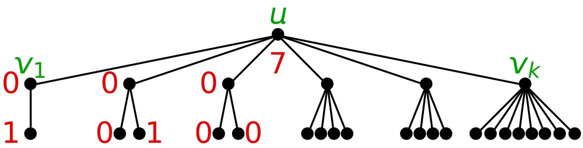
Case 2: $d(v_k) \ge 2\sqrt{n} \Rightarrow a_t(T) \le 1 + 2\sqrt{(n - 2\sqrt{n}) \lg(2n)}$.

Case 3: let w_i = weight on u before processing v_i;
algorithm gives u weight $\min\{w_i, d(v_i)\}$ from v_i.

Let $S = \{i : d(v_i) > w_i\}$; weight doubles, so $|S| \le \lg n$.
Some of $N(v_i)$ stays when $i \in S$. Let $m = \max S$.

$$a_t(T) < 1 + |S|\, d(v_m).$$

Trees with Diameter 4, Upper Bound



Thm. $a_t(T) \le 2\sqrt{n \lg(2n)}$.

Pf. Case 1: $k \le 2\sqrt{n \lg(2n)} \Rightarrow N(u)$ absorbs all.

Case 2: $d(v_k) \ge 2\sqrt{n} \Rightarrow a_t(T) \le 1 + 2\sqrt{(n-2\sqrt{n})\lg(2n)}$.

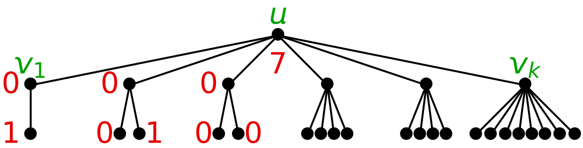
Case 3: let w_i = weight on u before processing v_i;
algorithm gives u weight $\min\{w_i, d(v_i)\}$ from v_i.

Let $S = \{i\colon d(v_i) > w_i\}$; weight doubles, so $|S| \le \lg n$.
Some of $N(v_i)$ stays when $i \in S$. Let $m = \max S$.

$$a_t(T) < 1 + |S|\, d(v_m).$$

$$m \le w_m < d(v_m) \le d(v_k) < 2\sqrt{n} < k/2.$$

Trees with Diameter 4, Upper Bound



Thm. $a_t(T) \le 2\sqrt{n \lg(2n)}$.

Pf. Case 1: $k \le 2\sqrt{n \lg(2n)} \Rightarrow N(u)$ absorbs all.

Case 2: $d(v_k) \ge 2\sqrt{n} \Rightarrow a_t(T) \le 1 + 2\sqrt{(n - 2\sqrt{n}) \lg(2n)}$.

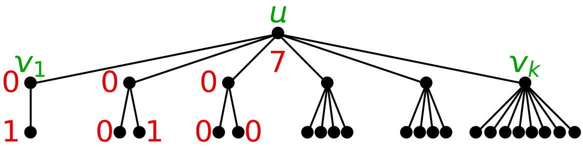
Case 3: let w_i = weight on u before processing v_i; algorithm gives u weight $\min\{w_i, d(v_i)\}$ from v_i.

Let $S = \{i\colon d(v_i) > w_i\}$; weight doubles, so $|S| \le \lg n$. Some of $N(v_i)$ stays when $i \in S$. Let $m = \max S$.

$$a_t(T) < 1 + |S|\, d(v_m).$$

$d(v_m)(k - m) < n$, so $d(v_m) < 2n/k$ (using $m < k/2$).

Trees with Diameter 4, Upper Bound



Thm. $a_t(T) \le 2\sqrt{n \lg(2n)}$.

Pf. Case 1: $k \le 2\sqrt{n \lg(2n)} \Rightarrow N(u)$ absorbs all.

Case 2: $d(v_k) \ge 2\sqrt{n} \Rightarrow a_t(T) \le 1 + 2\sqrt{(n-2\sqrt{n})\lg(2n)}$.

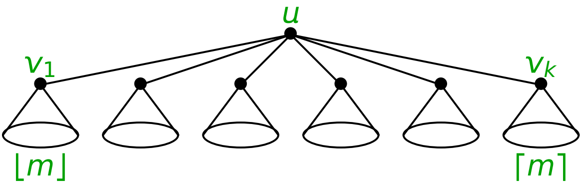
Case 3: let w_i = weight on u before processing v_i; algorithm gives u weight $\min\{w_i, d(v_i)\}$ from v_i.

Let $S = \{i\colon d(v_i) > w_i\}$; weight doubles, so $|S| \le \lg n$. Some of $N(v_i)$ stays when $i \in S$. Let $m = \max S$.

$$a_t(T) < 1 + |S|\, d(v_m).$$

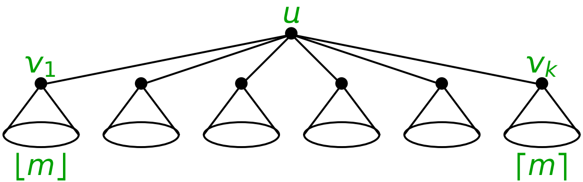
$d(v_m)(k-m) < n$, so $d(v_m) < 2n/k$ (using $m < k/2$). Hence $a_t(T) < 1 + |S|\,(2n/k) < 2\sqrt{n \lg(2n)}$. ■

Trees with Diameter 4, Lower Bound



Thm. $a_t(T) \geq .5\sqrt{n \lg n}$.

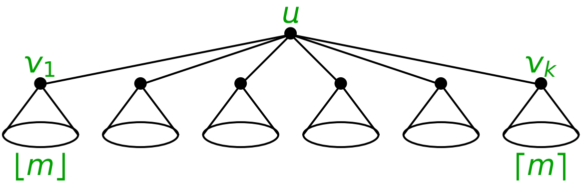
Trees with Diameter 4, Lower Bound



Thm. $a_t(T) \geq .5\sqrt{n \lg n}$.

Pf. Let $k = \sqrt{n \lg n}$ and $m = (n - k - 1)/k$.

Trees with Diameter 4, Lower Bound

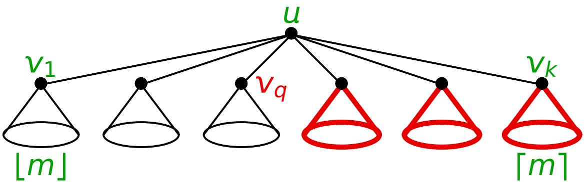


Thm. $a_t(T) \geq .5\sqrt{n \lg n}$.

Pf. Let $k = \sqrt{n \lg n}$ and $m = (n - k - 1)/k$.

Suppose u receives weight from q nbrs, say $v_1, \ldots, v_q$.

Trees with Diameter 4, Lower Bound



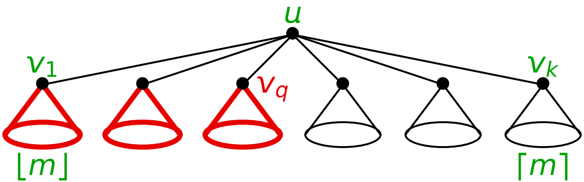
Thm. $a_t(T) \geq .5\sqrt{n \lg n}$.

Pf. Let $k = \sqrt{n \lg n}$ and $m = (n - k - 1)/k$.

Suppose u receives weight from q nbrs, say $v_1, \ldots, v_q$.

If $q \leq \lg n$, then weight remains in $k - \lg n$ subtrees.

Trees with Diameter 4, Lower Bound



Thm. $\alpha_t(T) \geq .5\sqrt{n \lg n}$.

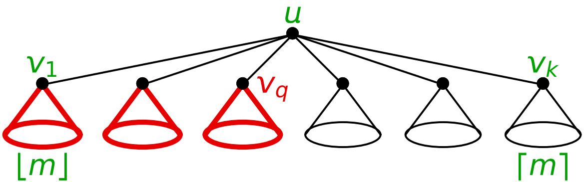
Pf. Let $k = \sqrt{n \lg n}$ and $m = (n - k - 1)/k$.

Suppose u receives weight from q nbrs, say $v_1, \ldots, v_q$.

If $q \leq \lg n$, then weight remains in $k - \lg n$ subtrees.

If $q > \lg n$, then at least $m/2$ leaves are stranded at least $\lg n$ times.

Trees with Diameter 4, Lower Bound



Thm. $a_t(T) \geq .5\sqrt{n \lg n}$.

Pf. Let $k = \sqrt{n \lg n}$ and $m = (n - k - 1)/k$.

Suppose u receives weight from q nbrs, say $v_1, \ldots, v_q$.

If $q \leq \lg n$, then weight remains in $k - \lg n$ subtrees.

If $q > \lg n$, then at least $m/2$ leaves are stranded at least $\lg n$ times.

Thus #leaves stranded $\geq \frac{n}{2\sqrt{n \lg n}} \lg n = \frac{1}{2}\sqrt{n \lg n}$. ∎

More on Trees

$\text{diam}(T) = 5$ similar; ignore central edge, use $\text{diam} = 4$.

More on Trees

$\mathrm{diam}(T) = 5$ similar; ignore central edge, use $\mathrm{diam} = 4$.

Def. rooted acquis. tree = all weight can move to root

More on Trees

$\mathrm{diam}(T) = 5$ similar; ignore central edge, use $\mathrm{diam} = 4$.

Def. rooted acquis. tree = all weight can move to root

Lem. A tree T rooted at r is r.a.t. $\iff$ $V(T) = \{r\}$ or $\exists$ edge rr' where both components of $T - rr'$ are r.a.t.s and the one containing r is biggest.

More on Trees

$\mathrm{diam}(T) = 5$ similar; ignore central edge, use $\mathrm{diam} = 4$.

Def. rooted acquis. tree = all weight can move to root

Lem. A tree T rooted at r is r.a.t. $\iff$ $V(T) = \{r\}$ or $\exists$ edge rr' where both components of $T - rr'$ are r.a.t.s and the one containing r is biggest.

This is the definition of union trees, characterized by Cai [1993], recognizable in quadratic time.

More on Trees

$\text{diam}(T) = 5$ similar; ignore central edge, use $\text{diam} = 4$.

Def. rooted acquis. tree = all weight can move to root

Lem. A tree T rooted at r is r.a.t. $\iff$ $V(T) = \{r\}$ or $\exists$ edge rr' where both components of $T - rr'$ are r.a.t.s and the one containing r is biggest.

This is the definition of union trees, characterized by Cai [1993], recognizable in quadratic time.

Thm. On trees, $a_t(T) \leq k$ is testable in time $O(n^{k+1})$.

Pf. Try all sets of $k - 1$ edges to delete and form k components. Test each for being a union tree. ■

More on Trees

$\text{diam}(T) = 5$ similar; ignore central edge, use $\text{diam} = 4$.

Def. rooted acquis. tree = all weight can move to root

Lem. A tree T rooted at r is r.a.t. $\Leftrightarrow$ $V(T) = \{r\}$ or $\exists$ edge rr' where both components of $T - rr'$ are r.a.t.s and the one containing r is biggest.

This is the definition of union trees, characterized by Cai [1993], recognizable in quadratic time.

Thm. On trees, $a_t(T) \le k$ is testable in time $O(n^{k+1})$.

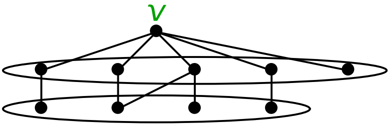
Pf. Try all sets of $k-1$ edges to delete and form k components. Test each for being a union tree. ∎

Thm. (Lampert–Slater [1995]) Testing $a_t(G) = 1$ on general graphs is NP-complete.

Sufficient Conditions for $\alpha_t(G) = 1$

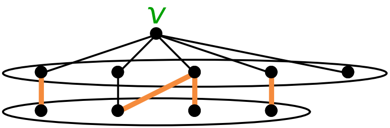
Sufficient Conditions for $a_t(G) = 1$

Lem. If the neighborhood of some vertex v with degree at least $n/2$ dominates G, then $a_t(G) = 1$.



Sufficient Conditions for $\alpha_t(G) = 1$

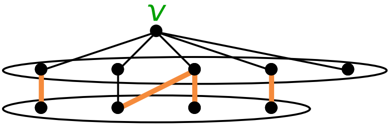
Lem. If the neighborhood of some vertex v with degree at least $n/2$ dominates G, then $\alpha_t(G) = 1$.



Pf. First move all weight from $V(G) - N[v]$ onto $N(v)$.

Sufficient Conditions for $\alpha_t(G) = 1$

Lem. If the neighborhood of some vertex v with degree at least $n/2$ dominates G, then $\alpha_t(G) = 1$.

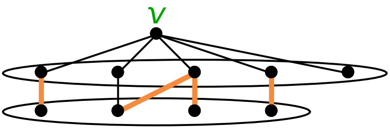


Pf. First move all weight from $V(G) - N[v]$ onto $N(v)$.

Since $d(v) > |V(G) - N[v]|$, #vertices of weight 1 left in $N(v)$ is at least the maximum wt added to any nbr of v.

Sufficient Conditions for $a_t(G) = 1$

Lem. If the neighborhood of some vertex v with degree at least $n/2$ dominates G, then $a_t(G) = 1$.



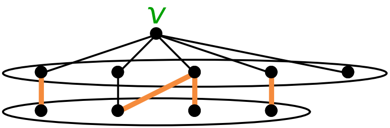
Pf. First move all weight from $V(G) - N[v]$ onto $N(v)$.

Since $d(v) > |V(G) - N[v]|$, #vertices of weight 1 left in $N(v)$ is at least the maximum wt added to any nbr of v.

Hence v can absorb the weight of neighbors with weight 1 and then absorb the remaining weight. ∎

Sufficient Conditions for $a_t(G) = 1$

Lem. If the neighborhood of some vertex v with degree at least $n/2$ dominates G, then $a_t(G) = 1$.



Pf. First move all weight from $V(G) - N[v]$ onto $N(v)$.

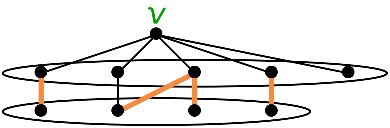
Since $d(v) > |V(G) - N[v]|$, #vertices of weight 1 left in $N(v)$ is at least the maximum wt added to any nbr of v.

Hence v can absorb the weight of neighbors with weight 1 and then absorb the remaining weight. ■

Lem. If G is $\frac{n-1}{2}$-regular and not C_5, then $a_t(G) = 1$.

Sufficient Conditions for $a_t(G) = 1$

Lem. If the neighborhood of some vertex v with degree at least $n/2$ dominates G, then $a_t(G) = 1$.



Pf. First move all weight from $V(G) - N[v]$ onto $N(v)$.

Since $d(v) > |V(G) - N[v]|$, #vertices of weight 1 left in $N(v)$ is at least the maximum wt added to any nbr of v.

Hence v can absorb the weight of neighbors with weight 1 and then absorb the remaining weight. ■

Lem. If G is $\frac{n-1}{2}$-regular and not C_5, then $a_t(G) = 1$.

Pf. Since nonadjacent vertices have a common nbr, $N(v)$ dominates. Previous argument almost works. ■

More on $\alpha_t(G) = 1$

Thm. If $G \neq C_5$, then $\alpha_t(G) = 1$ or $\alpha_t(\overline{G}) = 1$.

More on $a_t(G) = 1$

Thm. If $G \neq C_5$, then $a_t(G) = 1$ or $a_t(\overline{G}) = 1$.

Pf. $d_G(x, y) \geq 3 \Rightarrow \{x, y\}$ dominates $\overline{G} \Rightarrow a_t(\overline{G}) = 1$.

More on $\alpha_t(G) = 1$

Thm. If $G \neq C_5$, then $\alpha_t(G) = 1$ or $\alpha_t(\overline{G}) = 1$.

Pf. $d_G(x, y) \geq 3 \;\Rightarrow\; \{x, y\}$ dominates $\overline{G} \;\Rightarrow\; \alpha_t(\overline{G}) = 1$.

$\therefore\;$ $\mathrm{diam}\, G = \mathrm{diam}\, \overline{G} = 2$, and every nbhd dominates.

More on $a_t(G) = 1$

Thm. If $G \neq C_5$, then $a_t(G) = 1$ or $a_t(\overline{G}) = 1$.

Pf. $d_G(x, y) \geq 3 \Rightarrow \{x, y\}$ dominates $\overline{G} \Rightarrow a_t(\overline{G}) = 1$.

$\therefore$ $\mathrm{diam} G = \mathrm{diam} \overline{G} = 2$, and every nbhd dominates.

If $\max\{\Delta(G), \Delta(\overline{G})\} \geq n/2$, then first Lemma applies.

Else G and $\overline{G}$ are $\frac{n-1}{2}$-regular, and second applies. ■

More on $\alpha_t(G) = 1$

Thm. If $G \neq C_5$, then $\alpha_t(G) = 1$ or $\alpha_t(\overline{G}) = 1$.

Pf. $d_G(x, y) \geq 3 \;\Rightarrow\; \{x, y\}$ dominates $\overline{G} \;\Rightarrow\; \alpha_t(\overline{G}) = 1$.

$\therefore$ $\mathrm{diam} G = \mathrm{diam} \overline{G} = 2$, and every nbhd dominates.

If $\max\{\Delta(G), \Delta(\overline{G})\} \geq n/2$, then first Lemma applies.
Else G and $\overline{G}$ are $\frac{n-1}{2}$-regular, and second applies. ■

Thm. If $G \neq C_5$ and $d(u) + d(v) \geq n - 1$ whenever $uv \notin E(G)$, then $\alpha_t(G) = 1$.

More on $\alpha_t(G) = 1$

Thm. If $G \neq C_5$, then $\alpha_t(G) = 1$ or $\alpha_t(\overline{G}) = 1$.

Pf. $d_G(x, y) \geq 3 \Rightarrow \{x, y\}$ dominates $\overline{G} \Rightarrow \alpha_t(\overline{G}) = 1$.

$\therefore$ $\text{diam}\, G = \text{diam}\, \overline{G} = 2$, and every nbhd dominates.

If $\max\{\Delta(G), \Delta(\overline{G})\} \geq n/2$, then first Lemma applies.
Else G and $\overline{G}$ are $\frac{n-1}{2}$-regular, and second applies. ■

Thm. If $G \neq C_5$ and $d(u) + d(v) \geq n - 1$ whenever $uv \notin E(G)$, then $\alpha_t(G) = 1$.

Pf. The hypothesis gives a common nbr to nonadjacent vertices, so every neighborhood dominates.

Either $\Delta(G) \geq n/2$, or G is $(n-1)/2$-regular. ■

Results Not Presented

Random Graphs

Thm. If $p_n \geq \sqrt{3 \ln n / n}$, then almost always $a_t(G) = 1$.

Thm. If $p_n = c/n$, then almost always $a_t(G) = \Theta(n)$.

Thm. For almost all trees, $a_t(T) \geq .06n$.

Results Not Presented

Random Graphs

Thm. If $p_n \geq \sqrt{3 \ln n / n}$, then almost always $a_t(G) = 1$.

Thm. If $p_n = c/n$, then almost always $a_t(G) = \Theta(n)$.

Thm. For almost all trees, $a_t(T) \geq .06n$.

Edge-deletion

Thm. If e is an edge in G, then
$a_t(G - e) < a_t(G) + 7\sqrt{n}$.

Thm. There exists a tree T with an edge e such that
$a_t(T) = 1$ and $a_t(T - e) > \sqrt{n}/2$.

Results Not Presented

Random Graphs

Thm. If $p_n \geq \sqrt{3 \ln n / n}$, then almost always $a_t(G) = 1$.

Thm. If $p_n = c/n$, then almost always $a_t(G) = \Theta(n)$.

Thm. For almost all trees, $a_t(T) \geq .06n$.

Edge-deletion

Thm. If e is an edge in G, then
$a_t(G - e) < a_t(G) + 7\sqrt{n}$.

Thm. There exists a tree T with an edge e such that
$a_t(T) = 1$ and $a_t(T - e) > \sqrt{n}/2$.

Diameter 2

Thm. If $\mathrm{diam} G = 2$, then $a_t(G) \leq 250 \lg n \lg \lg n$.
If $\mathrm{diam} G = 2$ and $C_4 \not\subseteq G$ and $\Delta(G) \geq 8$, then $a_t(G) = 1$.

Open Problems (for Total Acquisition)

Conj. There is a constant c such that, if $\mathrm{diam}\, G = 2$, then $a_t(G) \le c$.

Perhaps $c = 2$. This suffices for Moore graphs, polarity graphs, and graphs without 4-cycles.

Open Problems (for Total Acquisition)

Conj. There is a constant c such that, if $\operatorname{diam} G = 2$, then $a_t(G) \le c$.

Perhaps $c = 2$. This suffices for Moore graphs, polarity graphs, and graphs without 4-cycles.

Ques. What is the maximum of $a_t(G)$ when G is a connected n-vertex graph with minimum degree k?

Open Problems (for Total Acquisition)

Conj. There is a constant c such that, if $\text{diam}G = 2$, then $a_t(G) \le c$.

Perhaps $c = 2$. This suffices for Moore graphs, polarity graphs, and graphs without 4-cycles.

Ques. What is the maximum of $a_t(G)$ when G is a connected n-vertex graph with minimum degree k?

Always $a_t(G) \le \frac{1+\ln(k+1)}{k+1}n$, since always $a_t(G) \le \gamma_t(G)$.

(Also bounded by #double-stars to cover $V(G)$.)

Open Problems (for Total Acquisition)

Conj. There is a constant c such that, if $\text{diam} G = 2$, then $a_t(G) \le c$.

Perhaps $c = 2$. This suffices for Moore graphs, polarity graphs, and graphs without 4-cycles.

Ques. What is the maximum of $a_t(G)$ when G is a connected n-vertex graph with minimum degree k?

Always $a_t(G) \le \frac{1+\ln(k+1)}{k+1}n$, since always $a_t(G) \le \gamma_t(G)$.

(Also bounded by #double-stars to cover $V(G)$.)

For $k = 2$, note that $a_t(C_n) = \lceil n/4 \rceil$,
but also $a_t(G) > (\frac{1}{4} + \frac{1}{1024})n$ is possible
(binary trees with triangles at the leaves).

Unit Acquisition — Always $\alpha_u(G) \leq \alpha_t(G)$

Lem. For total or unit acquisition, at most one chip can reach and leave a degree-2 vertex v, and it can only be the chip from a neighbor.

Unit Acquisition — Always $a_u(G) \le a_t(G)$

Lem. For total or unit acquisition, at most one chip can reach and leave a degree-2 vertex v, and it can only be the chip from a neighbor.

Cor. If x and y are vertices in G, and every x, y-path has an internal vertex of degree 2 adjacent to neither of them, then the chips from x and y can never combine.

Unit Acquisition — Always $a_u(G) \le a_t(G)$

Lem. For total or unit acquisition, at most one chip can reach and leave a degree-2 vertex v, and it can only be the chip from a neighbor.

Cor. If x and y are vertices in G, and every x,y-path has an internal vertex of degree 2 adjacent to neither of them, then the chips from x and y can never combine.

Thm. For $d \ge 3$ and $k \ge 6$, there is a tree T with $\Delta(T) = d$, $\mathrm{diam} T \ge k$, and $a_u(T) = a_t(T) = (n+1)/3$.

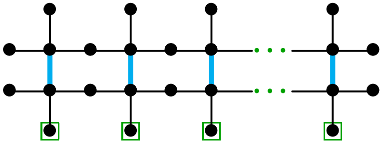
Unit Acquisition — Always $a_u(G) \le a_t(G)$

Lem. For total or unit acquisition, at most one chip can reach and leave a degree-2 vertex v, and it can only be the chip from a neighbor.

Cor. If x and y are vertices in G, and every x, y-path has an internal vertex of degree 2 adjacent to neither of them, then the chips from x and y can never combine.

Thm. For $d \ge 3$ and $k \ge 6$, there is a tree T with $\Delta(T) = d$, $\mathrm{diam} T \ge k$, and $a_u(T) = a_t(T) = (n+1)/3$.

Ex. $a_u(G) \le a_t(G) \le M$, where M is the least number of double-stars that cover $V(G)$, and this may be sharp.



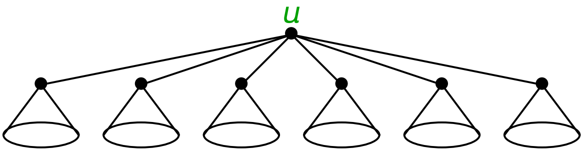
Diameter-4 Trees

- $\max a_t(T) = \Theta(\sqrt{n \lg n})$.

Diameter-4 Trees

- $\max a_t(T) = \Theta(\sqrt{n \lg n})$.

Prop. Among trees of diam 4, $\max a_u(T) = \lfloor \sqrt{n-1} \rfloor$.

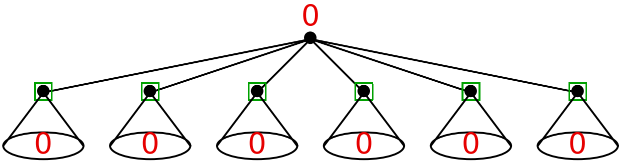


Diameter-4 Trees

- $\max a_t(T) = \Theta(\sqrt{n \lg n})$.

Prop. Among trees of diam 4, $\max a_u(T) = \left\lfloor \sqrt{n-1} \right\rfloor$.

Pf. If $d(u) \le \sqrt{n-1}$, then move all chips to $N(u)$.

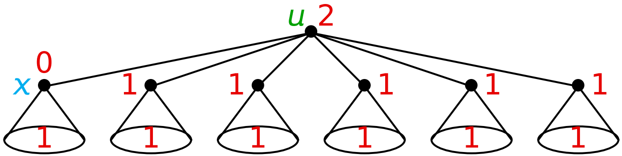


Diameter-4 Trees

- $\max a_t(T) = \Theta(\sqrt{n \lg n})$.

Prop. Among trees of diam 4, $\max a_u(T) = \left\lfloor \sqrt{n-1} \right\rfloor$.

Pf. If $d(u) \le \sqrt{n-1}$, then move all chips to $N(u)$.
Else $\exists\, x \in N(u)$ with $d(x) < \sqrt{n-1}$. Moving from x to u makes ascending tree that omits $< \sqrt{n-1}$ chips.

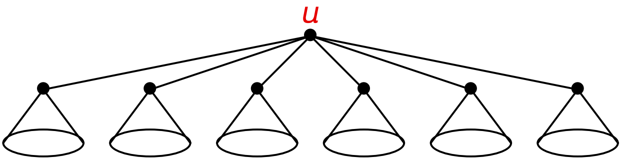


Diameter-4 Trees

- $\max a_t(T) = \Theta(\sqrt{n \lg n})$.

Prop. Among trees of diam 4, $\max a_u(T) = \left\lfloor \sqrt{n-1} \right\rfloor$.

Pf. If $d(u) \le \sqrt{n-1}$, then move all chips to $N(u)$.
Else $\exists\, x \in N(u)$ with $d(x) < \sqrt{n-1}$. Moving from x to u makes ascending tree that omits $< \sqrt{n-1}$ chips.



Construction: Make tree with $d(u) = m$ and $d(x) = m$ for $x \in N(u)$, so $n = m^2 + 1$. The first move involving u creates m components with positive weight.

A Tool for Unit Acquisition

Def. An ascending tree is a rooted tree such that each leaf has weight at most that of its neighbor, and other weights strictly increase along paths to the root.

A Tool for Unit Acquisition

Def. An ascending tree is a rooted tree such that each leaf has weight at most that of its neighbor, and other weights strictly increase along paths to the root.

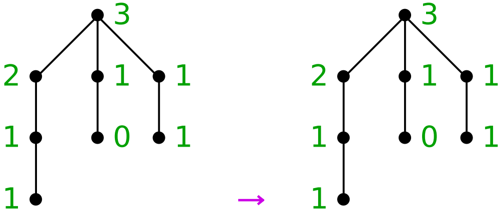
Lem. All weight in an ascending tree can be acquired to the root by unit acquisition moves.

A Tool for Unit Acquisition

Def. An ascending tree is a rooted tree such that each leaf has weight at most that of its neighbor, and other weights strictly increase along paths to the root.

Lem. All weight in an ascending tree can be acquired to the root by unit acquisition moves.

Pf. One unit can be moved from any leaf to the root. ■

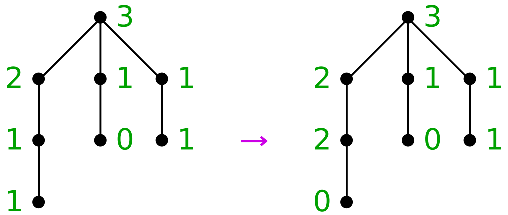


A Tool for Unit Acquisition

Def. An ascending tree is a rooted tree such that each leaf has weight at most that of its neighbor, and other weights strictly increase along paths to the root.

Lem. All weight in an ascending tree can be acquired to the root by unit acquisition moves.

Pf. One unit can be moved from any leaf to the root. ■

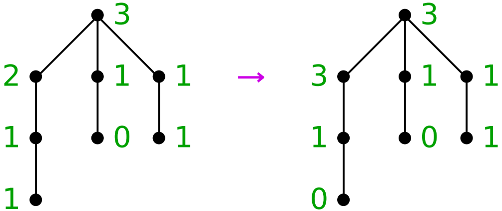


A Tool for Unit Acquisition

Def. An ascending tree is a rooted tree such that each leaf has weight at most that of its neighbor, and other weights strictly increase along paths to the root.

Lem. All weight in an ascending tree can be acquired to the root by unit acquisition moves.

Pf. One unit can be moved from any leaf to the root. ■

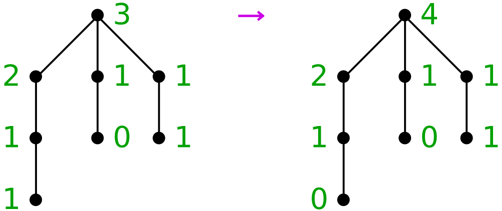


A Tool for Unit Acquisition

Def. An ascending tree is a rooted tree such that each leaf has weight at most that of its neighbor, and other weights strictly increase along paths to the root.

Lem. All weight in an ascending tree can be acquired to the root by unit acquisition moves.

Pf. One unit can be moved from any leaf to the root. ■

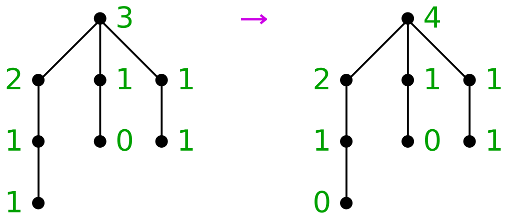


A Tool for Unit Acquisition

Def. An ascending tree is a rooted tree such that each leaf has weight at most that of its neighbor, and other weights strictly increase along paths to the root.

Lem. All weight in an ascending tree can be acquired to the root by unit acquisition moves.

Pf. One unit can be moved from any leaf to the root. ■



Thm. $a_u(T)=1 \Leftrightarrow$ an ascending tree can be produced.

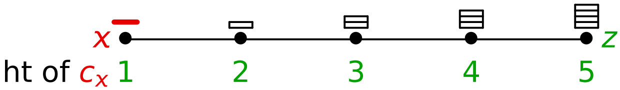
Weight as Chips at Heights

Idea: Weight on a vertex is a stack of chips.
As a chip moves, its height strictly increases.

Weight as Chips at Heights

Idea: Weight on a vertex is a stack of chips.
As a chip moves, its height strictly increases.

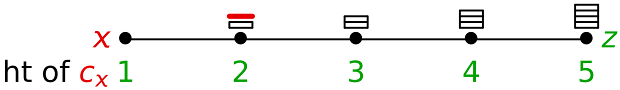
Lem. The (at least) $\sum_{i=1}^{k} i$ chips enabling chip c_x to move k steps (to height at least $k+1$) are distinct.



Weight as Chips at Heights

Idea: Weight on a vertex is a stack of chips.
As a chip moves, its height strictly increases.

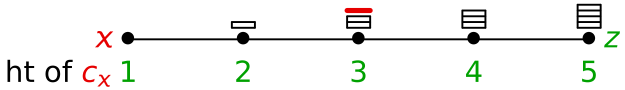
Lem. The (at least) $\sum_{i=1}^{k} i$ chips enabling chip c_x to move k steps (to height at least $k+1$) are distinct.



Weight as Chips at Heights

Idea: Weight on a vertex is a stack of chips.
As a chip moves, its height strictly increases.

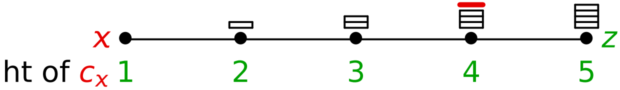
Lem. The (at least) $\sum_{i=1}^{k} i$ chips enabling chip c_x to move k steps (to height at least $k+1$) are distinct.



Weight as Chips at Heights

Idea: Weight on a vertex is a stack of chips.
As a chip moves, its height strictly increases.

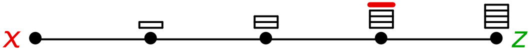
Lem. The (at least) $\sum_{i=1}^{k} i$ chips enabling chip c_x to move k steps (to height at least $k+1$) are distinct.



Weight as Chips at Heights

Idea: Weight on a vertex is a stack of chips.
As a chip moves, its height strictly increases.

Lem. The (at least) $\sum_{i=1}^{k} i$ chips enabling chip c_x to move k steps (to height at least $k+1$) are distinct.

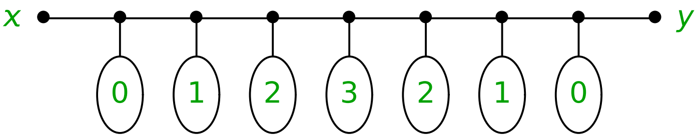


ht of c_x 1 2 3 4 5

- The chips under c_x can't be on the next vertex when c_x takes its ith step. Also, earlier chips supporting c_x can't be among the bottom i there.

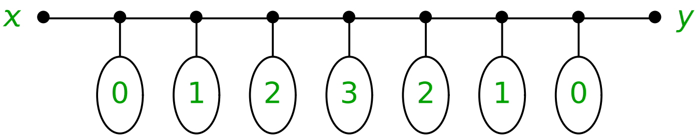
Merging Chips

Lem. If chips from leaves at distance $k+4$ reach a common vertex between them, then the subtrees at the central k vertices of the path together have at least $\sum_{i=1}^{k} \lceil i/2 \rceil$ vertices.



Merging Chips

Lem. If chips from leaves at distance $k+4$ reach a common vertex between them, then the subtrees at the central k vertices of the path together have at least $\sum_{i=1}^{k} \lceil i/2 \rceil$ vertices.



In trees, this can be used to establish lower bounds on a_u (like the degree-2 vertices earlier) or to seek a characterization of when $a_u(T) = 1$.

Necessary Condition for $a_u(T) = 1$

Cor. If $a_u(T) = 1$, then for any k-vertex path P in T that extends at least two steps farther at each end, the subtrees attached to vertices of P together have at least $\sum_{i=1}^{k} \lceil i/2 \rceil$ vertices.

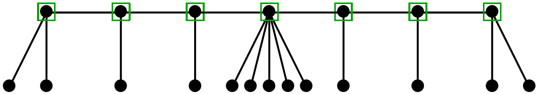
Pf. Consider where the root getting all the weight is. ∎

Necessary Condition for $a_u(T) = 1$

Cor. If $a_u(T) = 1$, then for any k-vertex path P in T that extends at least two steps farther at each end, the subtrees attached to vertices of P together have at least $\sum_{i=1}^{k} \lceil i/2 \rceil$ vertices.

Pf. Consider where the root getting all the weight is. ∎

Def. A caterpillar is a tree such that deleting all leaves produces a path (called the spine).

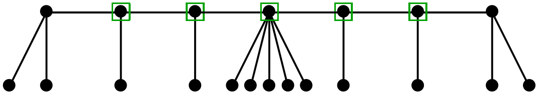


Necessary Condition for $a_u(T) = 1$

Cor. If $a_u(T) = 1$, then for any k-vertex path P in T that extends at least two steps farther at each end, the subtrees attached to vertices of P together have at least $\sum_{i=1}^{k} \lceil i/2 \rceil$ vertices.

Pf. Consider where the root getting all the weight is. ■

Def. A caterpillar is a tree such that deleting all leaves produces a path (called the spine).



Conj. If T is a caterpillar, then $a_u(T) = 1$ iff if every set S of k consecutive internal vertices along the spine has together at least $\sum_{i=1}^{k} \lceil i/2 \rceil$ leaf neighbors.

Degree Bounds

- If $\delta(G) = k$, then $\alpha_t(G) \leq \gamma_t(G) \leq \frac{1+\ln(k+1)}{k+1} n$.

Degree Bounds

- If $\delta(G) = k$, then $\alpha_t(G) \le \gamma_t(G) \le \frac{1+\ln(k+1)}{k+1} n$.

Thm. If $\delta(G) = k$, then $\alpha_u(G) \le \frac{1}{k}(n-1)$.

Degree Bounds

- If $\delta(G) = k$, then $\alpha_t(G) \le \gamma_t(G) \le \frac{1+\ln(k+1)}{k+1} n$.

Thm. If $\delta(G) = k$, then $\alpha_u(G) \le \frac{1}{k}(n-1)$.

Pf. Idea: Partition $V(G)$ into trees of diameter 4 centered at a maximal set with pairwise distance at least 3; acquire to $1/k$ of the vertices in each tree. ■

Degree Bounds

- If $\delta(G) = k$, then $\alpha_t(G) \le \gamma_t(G) \le \frac{1+\ln(k+1)}{k+1} n$.

Thm. If $\delta(G) = k$, then $\alpha_u(G) \le \frac{1}{k}(n-1)$.

Pf. Idea: Partition $V(G)$ into trees of diameter 4 centered at a maximal set with pairwise distance at least 3; acquire to $1/k$ of the vertices in each tree. ■

- (Lampert–Slater [1995]) $\alpha_t(G) \ge n/2^{\Delta(G)}$,
since a vertex v can acquire at most $2^{d(v)}$ weight.

Degree Bounds

- If $\delta(G) = k$, then $a_t(G) \le \gamma_t(G) \le \frac{1+\ln(k+1)}{k+1} n$.

Thm. If $\delta(G) = k$, then $a_u(G) \le \frac{1}{k}(n-1)$.

Pf. **Idea:** Partition $V(G)$ into trees of diameter 4 centered at a maximal set with pairwise distance at least 3; acquire to $1/k$ of the vertices in each tree. ■

- (Lampert–Slater [1995]) $a_t(G) \ge n/2^{\Delta(G)}$, since a vertex v can acquire at most $2^{d(v)}$ weight.

- Max vertex weight attainable under unit acquisition.

$\Delta(G) \rightarrow$	1	2	3	4	5
max wt $\rightarrow$	1	4	10	239	???

Degree Bounds

- If $\delta(G) = k$, then $\alpha_t(G) \le \gamma_t(G) \le \frac{1+\ln(k+1)}{k+1}n$.

Thm. If $\delta(G) = k$, then $\alpha_u(G) \le \frac{1}{k}(n-1)$.

Pf. Idea: Partition $V(G)$ into trees of diameter 4 centered at a maximal set with pairwise distance at least 3; acquire to $1/k$ of the vertices in each tree. ∎

- (Lampert–Slater [1995]) $\alpha_t(G) \ge n/2^{\Delta(G)}$, since a vertex v can acquire at most $2^{d(v)}$ weight.

- Max vertex weight attainable under unit acquisition.

$\Delta(G) \rightarrow$	1	2	3	4	5
max wt $\rightarrow$	1	4	10	239	???

Thm. (Wenger) There is an infinite family of trees with maximum degree 5 and unit acquisition number 1.

Degree Bounds

- If $\delta(G) = k$, then $a_t(G) \le \gamma_t(G) \le \frac{1+\ln(k+1)}{k+1} n$.

Thm. If $\delta(G) = k$, then $a_u(G) \le \frac{1}{k}(n-1)$.

Pf. Idea: Partition $V(G)$ into trees of diameter 4 centered at a maximal set with pairwise distance at least 3; acquire to $1/k$ of the vertices in each tree. ■

- (Lampert–Slater [1995]) $a_t(G) \ge n/2^{\Delta(G)}$, since a vertex v can acquire at most $2^{d(v)}$ weight.

- Max vertex weight attainable under unit acquisition.

$\Delta(G) \to$	1	2	3	4	5
max wt $\to$	1	4	10	239	???

Thm. (Wenger) There is an infinite family of trees with maximum degree 5 and unit acquisition number 1.

Pf. (Idea) Complicated inductive construction. ■

Diameter 2

- **Conj:** $a_t(G) \leq 2$ when G has diameter 2.

Diameter 2

- **Conj:** $a_t(G) \le 2$ when G has diameter 2.

Thm. (Prince–Wenger) If $\operatorname{diam} G = 2$ and G is not C_5 or the Petersen graph, then $a_u(G) = 1$.

Diameter 2

- **Conj:** $a_t(G) \le 2$ when G has diameter 2.

Thm. (Prince–Wenger) If $\operatorname{diam} G = 2$ and G is not C_5 or the Petersen graph, then $a_u(G) = 1$.

Pf. **(Idea)** In each of several cases depending on neighborhoods within a largest clique, a few moves create an ascending tree on the remaining vertices. ∎

Diameter 2

- **Conj:** $a_t(G) \le 2$ when G has diameter 2.

Thm. (Prince–Wenger) If $\operatorname{diam} G = 2$ and G is not C_5 or the Petersen graph, then $a_u(G) = 1$.

Pf. **(Idea)** In each of several cases depending on neighborhoods within a largest clique, a few moves create an ascending tree on the remaining vertices. ■

Open Problems:

- Find general lower bounds for $a_u(G)$ in terms of other parameters of G.

- Characterize the trees with $a_u(T) = 1$.

- What is the complexity of recognizing $a_u(G) = 1$?

Fractional Acquisition

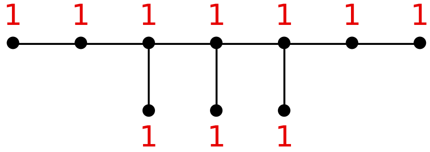
- Always $a_f(G) \leq a_u(G) \leq a_t(G)$
Fractional moves are more flexible and may do better.

Fractional Acquisition

- Always $a_f(G) \le a_u(G) \le a_t(G)$

Fractional moves are more flexible and may do better.

Ex. A caterpillar with $a_u(G) = a_t(G) = 2$
(no ascending tree can be produced).

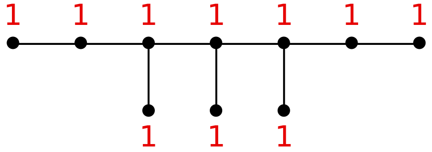


Fractional Acquisition

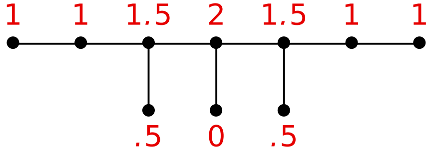
- Always $a_f(G) \le a_u(G) \le a_t(G)$

Fractional moves are more flexible and may do better.

Ex. A caterpillar with $a_u(G) = a_t(G) = 2$
(no ascending tree can be produced).

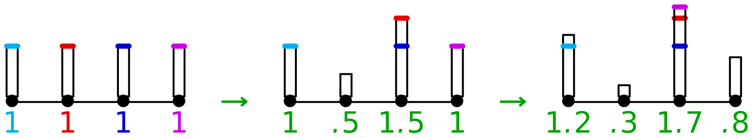


Move .5 twice to create an ascending tree with min difference .5: $a_f(G) = 1$.



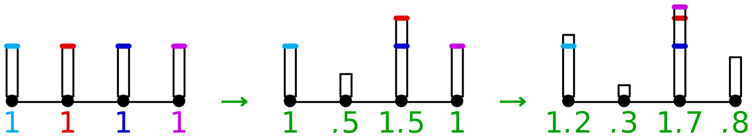
Tool for Lower Bounds

Def. Represent the weight at a vertex by a tower of that height. The tops of the initial intervals are tips. When weight α is moved from u to v, the amount α is removed from the top of the tower at u and placed on top of the tower at v, along with any tips it contains.



Tool for Lower Bounds

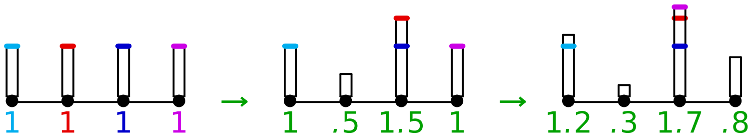
Def. Represent the weight at a vertex by a tower of that height. The tops of the initial intervals are tips. When weight α is moved from u to v, the amount α is removed from the top of the tower at u and placed on top of the tower at v, along with any tips it contains.



Obs. The height of a tip never decreases.

Tool for Lower Bounds

Def. Represent the weight at a vertex by a tower of that height. The tops of the initial intervals are tips. When weight α is moved from u to v, the amount α is removed from the top of the tower at u and placed on top of the tower at v, along with any tips it contains.



Obs. The height of a tip never decreases.

To make $\alpha_f(G)$ small, tips must reach common towers.

Paths and Cycles

Lem. If fractional moves on a path P produce v with $w(v) < 1$ between vertices x and y that have tips, then the tips from x and y can never reach one vertex.

Paths and Cycles

Lem. If fractional moves on a path P produce v with $w(v) < 1$ between vertices x and y that have tips, then the tips from x and y can never reach one vertex.

Pf. Let $w(v) = c < 1$. We claim that always a vertex of weight < 1 remains between the tips of x and y. If not, then v gains weight, first from a neighbor u with $w(u) \leq c$. Since heights of tips don't decrease, u does not have the tip of x or y. Hence u is between them, and $w(u) < 1$. ∎

Paths and Cycles

Lem. If fractional moves on a path P produce v with $w(v) < 1$ between vertices x and y that have tips, then the tips from x and y can never reach one vertex.

Pf. Let $w(v) = c < 1$. We claim that always a vertex of weight < 1 remains between the tips of x and y. If not, then v gains weight, first from a neighbor u with $w(u) \leq c$. Since heights of tips don't decrease, u does not have the tip of x or y. Hence u is between them, and $w(u) < 1$. ■

Thm. $\alpha_f(P_n) = \lceil n/4 \rceil$.

Pf. For any five vertices, the first move involving the middle one leaves a vertex with weight less than 1 between the tips of the end ones. Hence no vertex acquires five tips. Upper bound: $\alpha_t(P_4) = 1$. ■

Paths and Cycles

Lem. If fractional moves on a path P produce v with $w(v) < 1$ between vertices x and y that have tips, then the tips from x and y can never reach one vertex.

Pf. Let $w(v) = c < 1$. We claim that always a vertex of weight < 1 remains between the tips of x and y. If not, then v gains weight, first from a neighbor u with $w(u) \le c$. Since heights of tips don't decrease, u does not have the tip of x or y. Hence u is between them, and $w(u) < 1$. ■

Thm. $a_f(P_n) = \lceil n/4 \rceil$.

Pf. For any five vertices, the first move involving the middle one leaves a vertex with weight less than 1 between the tips of the end ones. Hence no vertex acquires five tips. Upper bound: $a_t(P_4) = 1$. ■

- Similarly, $a_f(C_n) = \lceil n/4 \rceil$.

Surprise!

- If G is a path or cycle, then $\alpha_f(G) = \lceil n/4 \rceil$.

Surprise!

- If G is a path or cycle, then $\alpha_f(G) = \lceil n/4 \rceil$.
- Otherwise, $\Delta(G) \geq 3$.

Surprise!

- If G is a path or cycle, then $\alpha_f(G) = \lceil n/4 \rceil$.
- Otherwise, $\Delta(G) \geq 3$.

Thm. (Wenger) $\Delta(G) \geq 3$ and connected

Surprise!

- If G is a path or cycle, then $\alpha_f(G) = \lceil n/4 \rceil$.
- Otherwise, $\Delta(G) \geq 3$.

Thm. (Wenger) $\Delta(G) \geq 3$ and connected $\Rightarrow \alpha_f(G) = 1$.

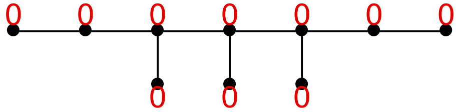
Surprise!

- If G is a path or cycle, then $\alpha_f(G) = \lceil n/4 \rceil$.
- Otherwise, $\Delta(G) \geq 3$.

Thm. (Wenger) $\Delta(G) \geq 3$ and connected $\Rightarrow \alpha_f(G) = 1$.

Main Steps:

1) New model: make all initial weights 0, require integer moves, and allow negative weights.



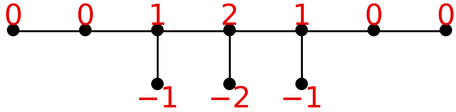
Surprise!

- If G is a path or cycle, then $\alpha_f(G) = \lceil n/4 \rceil$.
- Otherwise, $\Delta(G) \geq 3$.

Thm. (Wenger) $\Delta(G) \geq 3$ and connected $\Rightarrow \alpha_f(G) = 1$.

Main Steps:

1) New model: make all initial weights 0, require integer moves, and allow negative weights.



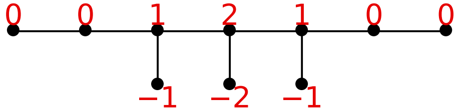
Surprise!

- If G is a path or cycle, then $\alpha_f(G) = \lceil n/4 \rceil$.
- Otherwise, $\Delta(G) \geq 3$.

Thm. (Wenger) $\Delta(G) \geq 3$ and connected $\Rightarrow \alpha_f(G) = 1$.

Main Steps:

1) New model: make all initial weights 0, require integer moves, and allow negative weights.



2) An ascending tree can be normalized: divide by the largest magnitude of weight ever used in the process and add 1. Now the corresponding fractional moves in the original problem produce an ascending tree.

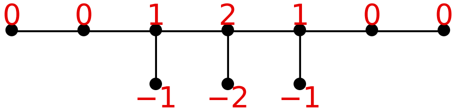
Surprise!

- If G is a path or cycle, then $\alpha_f(G) = \lceil n/4 \rceil$.
- Otherwise, $\Delta(G) \geq 3$.

Thm. (Wenger) $\Delta(G) \geq 3$ and connected $\Rightarrow \alpha_f(G) = 1$.

Main Steps:

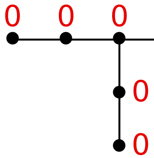
1) New model: make all initial weights 0, require integer moves, and allow negative weights.



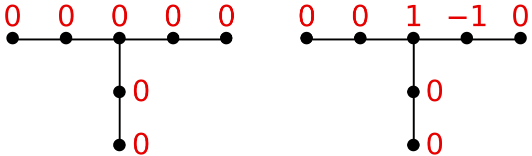
2) An ascending tree can be normalized: divide by the largest magnitude of weight ever used in the process and add 1. Now the corresponding fractional moves in the original problem produce an ascending tree.

3) Inductively produce an ascending tree in this model.

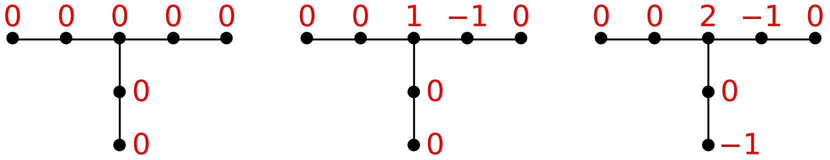
An Example



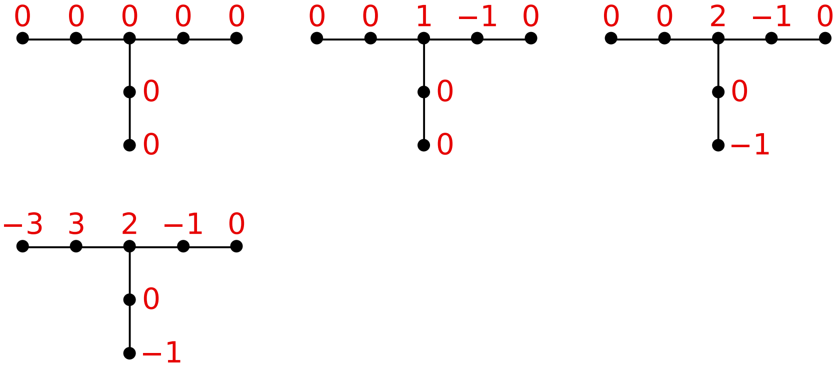
An Example



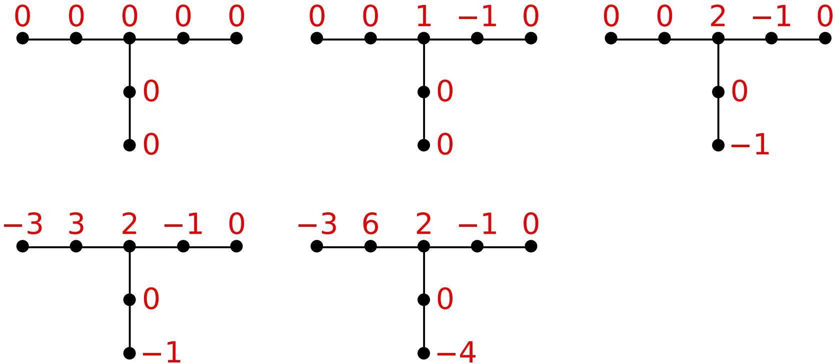
An Example



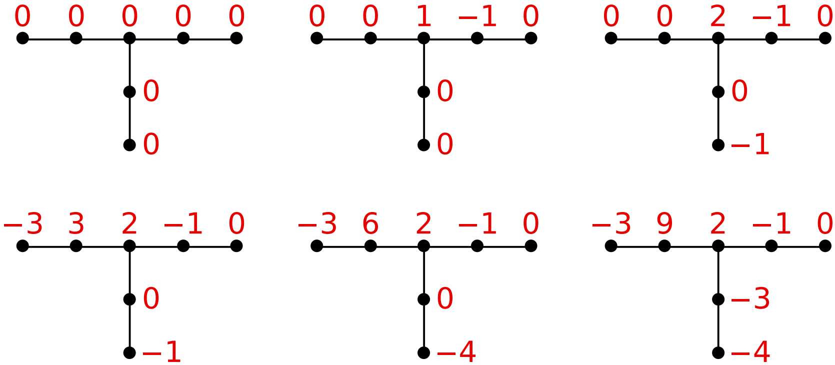
An Example



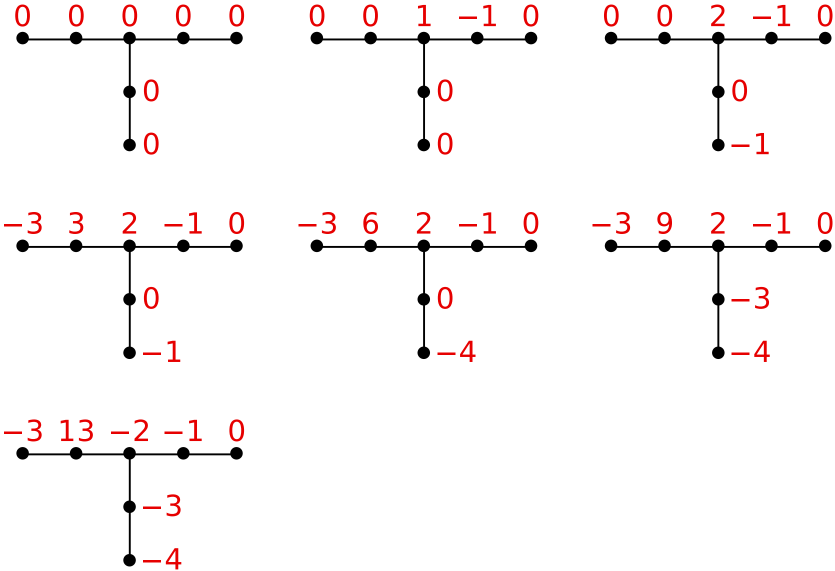
An Example



An Example

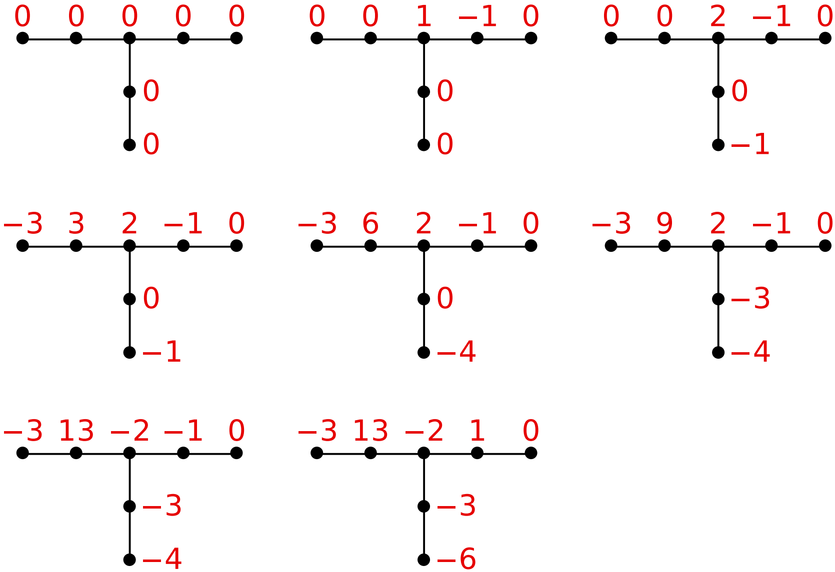


An Example

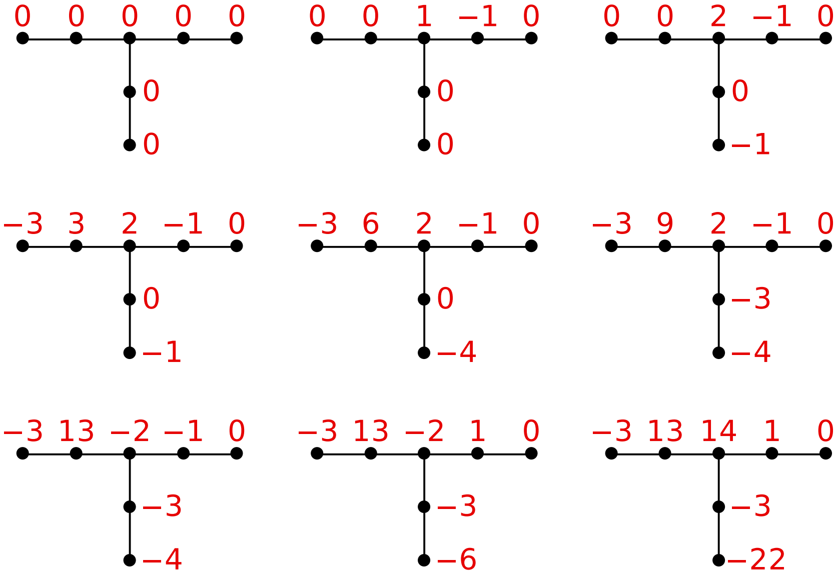


0 0 0 0 0
0
0
0 0 1 −1 0
0
0
0 0 2 −1 0
0
−1
−3 3 2 −1 0
0
−1
−3 6 2 −1 0
0
−4
−3 9 2 −1 0
−3
−4
−3 13 −2 −1 0
−3
−4

An Example


0 0 0 0 0
0
0
0 0 1 −1 0
0
0
0 0 2 −1 0
0
−1
−3 3 2 −1 0
0
−1
−3 6 2 −1 0
0
−4
−3 9 2 −1 0
−3
−4
−3 13 −2 −1 0
−3
−4
−3 13 −2 1 0
−3
−6

An Example


0 0 0 0 0 0 0
0 0 1 −1 0 0 0
0 0 2 −1 0 0 −1
−3 3 2 −1 0 0 −1
−3 6 2 −1 0 0 −4
−3 9 2 −1 0 −3 −4
−3 13 −2 −1 0 −3 −4
−3 13 −2 1 0 −3 −6
−3 13 14 1 0 −3 −22

Game Acquisition

Def. (Slater–Wang [2004]) Minimizer and Maximizer alternate total acquisition moves, aiming to min or max the final independent set. The game acquisition number(s) $a_g(G)$ and $\hat{a}_g(G)$ are the result of best play when Min or Max plays first, respectively.

Game Acquisition

Def. (Slater–Wang [2004]) Minimizer and Maximizer alternate total acquisition moves, aiming to min or max the final independent set. The game acquisition number(s) $a_g(G)$ and $\hat{a}_g(G)$ are the result of best play when Min or Max plays first, respectively.

Thm. (Slater–Wang [2004] $a_g(P_n) = 2n/5$.

Game Acquisition

Def. (Slater–Wang [2004]) Minimizer and Maximizer alternate total acquisition moves, aiming to min or max the final independent set. The game acquisition number(s) $a_g(G)$ and $\hat{a}_g(G)$ are the result of best play when Min or Max plays first, respectively.

Thm. (Slater–Wang [2004] $a_g(P_n) = 2n/5$.

Ex. Who moves first?: $a_g(K_{1,q}) = 1$, but $\hat{a}_g(K_{1,q}) = q$.

Game Acquisition

Def. (Slater–Wang [2004]) Minimizer and Maximizer alternate total acquisition moves, aiming to min or max the final independent set. The game acquisition number(s) $a_g(G)$ and $\hat{a}_g(G)$ are the result of best play when Min or Max plays first, respectively.

Thm. (Slater–Wang [2004] $a_g(P_n) = 2n/5$.

Ex. Who moves first?: $a_g(K_{1,q}) = 1$, but $\hat{a}_g(K_{1,q}) = q$.
Double-star $S_{p,q}$ with $p \le q$: $a_g(S_{p,q}) = p$, but $\hat{a}_g(S_{p,q}) = q$.

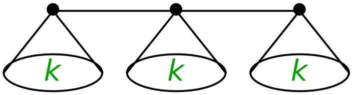
Game Acquisition

Def. (Slater–Wang [2004]) Minimizer and Maximizer alternate total acquisition moves, aiming to min or max the final independent set. The game acquisition number(s) $a_g(G)$ and $\hat{a}_g(G)$ are the result of best play when Min or Max plays first, respectively.

Thm. (Slater–Wang [2004] $a_g(P_n) = 2n/5$.

Ex. Who moves first?: $a_g(K_{1,q}) = 1$, but $\hat{a}_g(K_{1,q}) = q$.
Double-star $S_{p,q}$ with $p \le q$: $a_g(S_{p,q}) = p$, but $\hat{a}_g(S_{p,q}) = q$.

Ex. For the tree T below, $a_g(T) \approx 2n/3$. Max kills one end of the spine in the first round and combines the two remaining spine vertices in the second round.



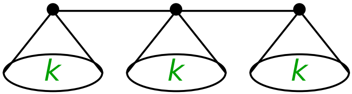
Game Acquisition

Def. (Slater–Wang [2004]) Minimizer and Maximizer alternate total acquisition moves, aiming to min or max the final independent set. The game acquisition number(s) $a_g(G)$ and $\hat{a}_g(G)$ are the result of best play when Min or Max plays first, respectively.

Thm. (Slater–Wang [2004] $a_g(P_n) = 2n/5$.

Ex. Who moves first?: $a_g(K_{1,q}) = 1$, but $\hat{a}_g(K_{1,q}) = q$.
Double-star $S_{p,q}$ with $p \le q$: $a_g(S_{p,q}) = p$, but $\hat{a}_g(S_{p,q}) = q$.

Ex. For the tree T below, $a_g(T) \approx 2n/3$. Max kills one end of the spine in the first round and combines the two remaining spine vertices in the second round.



Ques. What is $\max a_g(T)$ among n-vertex trees?

Results

- We study a_g and $\hat{a}_g$ for complete bipartite $K_{m,n}$.

Fix $1 \leq m \leq n$, partite sets X, Y with $|X| = m$, $|Y| = n$.

Results

- We study α_g and $\hat{\alpha}_g$ for complete bipartite $K_{m,n}$.

Fix $1 \leq m \leq n$, partite sets X, Y with $|X| = m$, $|Y| = n$.

Upper bd = Min strategy; Lower bd = Max strategy.

Results

- We study α_g and $\hat{\alpha}_g$ for complete bipartite $K_{m,n}$.

Fix $1 \le m \le n$, partite sets X, Y with $|X| = m$, $|Y| = n$.

Upper bd = Min strategy; Lower bd = Max strategy.

Thm. $\hat{\alpha}_g(K_{m,n}) = n - m + 1$.

Results

- We study a_g and $\hat{a}_g$ for complete bipartite $K_{m,n}$.

Fix $1 \le m \le n$, partite sets X, Y with $|X| = m$, $|Y| = n$.

Upper bd = Min strategy; Lower bd = Max strategy.

Thm. $\hat{a}_g(K_{m,n}) = n - m + 1$.

Thm. $a_g(K_{m,n}) \le n - m + 2$.

Results

- We study a_g and $\hat{a}_g$ for complete bipartite $K_{m,n}$.

Fix $1 \le m \le n$, partite sets X, Y with $|X| = m$, $|Y| = n$.

Upper bd = Min strategy; Lower bd = Max strategy.

Thm. $\hat{a}_g(K_{m,n}) = n - m + 1$.

Thm. $a_g(K_{m,n}) \le n - m + 2$.

Thm. $a_g(K_{n,n}) = 2$.

Results

- We study α_g and $\hat{\alpha}_g$ for complete bipartite $K_{m,n}$.

Fix $1 \le m \le n$, partite sets X, Y with $|X| = m$, $|Y| = n$.

Upper bd = Min strategy; Lower bd = Max strategy.

Thm. $\hat{\alpha}_g(K_{m,n}) = n - m + 1$.

Thm. $\alpha_g(K_{m,n}) \le n - m + 2$.

Thm. $\alpha_g(K_{n,n}) = 2$.

Thm. If $n \ge 2m + 2$, then $\alpha_g(K_{m,n}) \le \lfloor m/2 \rfloor + 2$.

Results

- We study a_g and $\hat{a}_g$ for complete bipartite $K_{m,n}$.

Fix $1 \le m \le n$, partite sets X, Y with $|X| = m$, $|Y| = n$.

Upper bd = Min strategy; Lower bd = Max strategy.

Thm. $\hat{a}_g(K_{m,n}) = n - m + 1$.

Thm. $a_g(K_{m,n}) \le n - m + 2$.

Thm. $a_g(K_{n,n}) = 2$.

Thm. If $n \ge 2m + 2$, then $a_g(K_{m,n}) \le \lfloor m/2 \rfloor + 2$.

Conj. $a_g(K_{m,n}) = 2 + \min\{\left\lfloor \frac{n-m}{3} \right\rfloor, \left\lfloor \frac{m}{2} \right\rfloor\}$.

Results

- We study a_g and $\hat{a}_g$ for complete bipartite $K_{m,n}$.

Fix $1 \le m \le n$, partite sets X, Y with $|X| = m$, $|Y| = n$.

Upper bd = Min strategy; Lower bd = Max strategy.

Thm. $\hat{a}_g(K_{m,n}) = n - m + 1$.

Thm. $a_g(K_{m,n}) \le n - m + 2$.

Thm. $a_g(K_{n,n}) = 2$.

Thm. If $n \ge 2m + 2$, then $a_g(K_{m,n}) \le \lfloor m/2 \rfloor + 2$.

Conj. $a_g(K_{m,n}) = 2 + \min\{\lfloor \frac{n-m}{3} \rfloor, \lfloor \frac{m}{2} \rfloor\}$.

Ques. Note that $a_g(K_{n,n}) = 2$ but $\hat{a}_g(K_{n,n}) = 1$.
When does $a_g(G) > \hat{a}_g(G)$ hold? How much difference?

Strategy for Max

Def. Live vertices are serfs (wt $= 1$) or kings (wt > 1).

Thm. $\hat{\alpha}_g(K_{m,n}) \geq n - m + 1$.

Strategy for Max

Def. Live vertices are serfs (wt $= 1$) or kings (wt > 1).

Thm. $\hat{\alpha}_g(K_{m,n}) \geq n - m + 1$.

Pf. Max initially combines two serfs into a king in Y.
Now X has r serfs and no kings, and Y has s serfs and t kings, where $t \geq 1$ and $s - r \geq n - m$.

Strategy for Max

Def. Live vertices are serfs (wt $= 1$) or kings (wt > 1).

Thm. $\hat{a}_g(K_{m,n}) \ge n - m + 1$.

Pf. Max initially combines two serfs into a king in Y.
Now X has r serfs and no kings, and Y has s serfs and t kings, where $t \ge 1$ and $s - r \ge n - m$.

In any such situation, if Min moves next, then Max restores these conditions: still at least $n - m$ excess serfs and one king in Y, no kings in X.

Strategy for Max

Def. Live vertices are serfs (wt $= 1$) or kings (wt > 1).

Thm. $\hat{a}_g(K_{m,n}) \geq n - m + 1$.

Pf. Max initially combines two serfs into a king in Y. Now X has r serfs and no kings, and Y has s serfs and t kings, where $t \geq 1$ and $s - r \geq n - m$.

In any such situation, if Min moves next, then Max restores these conditions: still at least $n - m$ excess serfs and one king in Y, no kings in X.

If Min makes a new X-king, Max eats it with a Y-king. Otherwise, Max eats an X-serf with a Y-king (if X has no serf, the game is over). Since X has no king, Min has no other way to disturb the conditions. ∎

Strategy for Min

Thm. $\hat{\alpha}_g(K_{m,n}) \leq n - m + 1$.

Strategy for Min

Thm. $\hat{a}_g(K_{m,n}) \le n - m + 1$.

Pf. The first move by Max makes a king; Min makes a king on the other side (if this is impossible, then the game is over or Max played badly). Now each side has one king, and the excess of serfs for Y over X is $n - m$.

Strategy for Min

Thm. $\hat{a}_g(K_{m,n}) \le n - m + 1$.

Pf. The first move by Max makes a king; Min makes a king on the other side (if this is impossible, then the game is over or Max played badly). Now each side has one king, and the excess of serfs for Y over X is $n - m$.

In any such situation, if Max moves next, then Min maintains these conditions: one king on each side, and excess of at most $n - m$ serfs for Y.

Strategy for Min

Thm. $\hat{a}_g(K_{m,n}) \le n - m + 1$.

Pf. The first move by Max makes a king; Min makes a king on the other side (if this is impossible, then the game is over or Max played badly). Now each side has one king, and the excess of serfs for Y over X is $n - m$.

In any such situation, if Max moves next, then Min maintains these conditions: one king on each side, and excess of at most $n - m$ serfs for Y.

If Max eats a serf with a king, Min eats a serf with the other king; excess unchanged. If Max makes a new king, Min eats it with the opposite king; both sides lose one serf. If Max eats a king, then Min replaces it (the game is over if this can't be done). ∎

Strategy for Min going first

Thm. $a_g(K_{m,n}) \le n - m + 2$.

Strategy for Min going first

Thm. $a_g(K_{m,n}) \le n - m + 2$.

Pf. Min first makes a king in X; excess remains $n - m$. Thereafter, Min always leaves at most three kings, with at least half on the smaller side. Also the numbers of serfs differ by at most $n - m$.

Strategy for Min going first

Thm. $a_g(K_{m,n}) \le n - m + 2$.

Pf. Min first makes a king in X; excess remains $n - m$. Thereafter, Min always leaves at most three kings, with at least half on the smaller side. Also the numbers of serfs differ by at most $n - m$.

If Max makes a king, then Min eats it, unless Max left two kings in X and none in Y; Min then makes a king in Y (if impossible, then the two kings in X eat the serfs in Y to end with 2). If Max eats a king, then Min restores it. Otherwise, Max eats a serf with a king; Min maintains the imbalance by eating a serf on the other side unless no king is available. In that case, Max ate a serf from Y and reduced the imbalance; Min makes a king in Y. ∎

Sharpness for $m = n$

Thm. $\alpha_g(K_{m,n}) \geq 2$ if $m \geq 2$.

Sharpness for $m = n$

Thm. $\alpha_g(K_{m,n}) \geq 2$ if $m \geq 2$.

Pf. Min must first make a king; Max makes another on that side. Max then eats kings made on side 2, keeping at least two kings on side 1 and none on side 2. The game ends when the serfs are eaten on side 2. ∎

Sharpness for $m = n$

Thm. $\alpha_g(K_{m,n}) \geq 2$ if $m \geq 2$.

Pf. Min must first make a king; Max makes another on that side. Max then eats kings made on side 2, keeping at least two kings on side 1 and none on side 2. The game ends when the serfs are eaten on side 2. ∎

When n is large, Min can make a King in X and then eat all kings Max makes in Y (eating Y-serfs when Max makes no king there). This yields $\alpha_g(K_{m,n}) \leq m$ for all n. Surprisingly, Min can do better.

Sharpness for $m = n$

Thm. $\alpha_g(K_{m,n}) \geq 2$ if $m \geq 2$.

Pf. Min must first make a king; Max makes another on that side. Max then eats kings made on side 2, keeping at least two kings on side 1 and none on side 2. The game ends when the serfs are eaten on side 2. ∎

When n is large, Min can make a King in X and then eat all kings Max makes in Y (eating Y-serfs when Max makes no king there). This yields $\alpha_g(K_{m,n}) \leq m$ for all n. Surprisingly, Min can do better.

Thm. If $n \geq 2m + 2$, then $\alpha_g(K_{m,n}) \leq \lfloor m/2 \rfloor + 2$.

Pf. A complicated three-stage strategy for Min. ∎

Sharpness for $m = n$

Thm. $\alpha_g(K_{m,n}) \geq 2$ if $m \geq 2$.

Pf. Min must first make a king; Max makes another on that side. Max then eats kings made on side 2, keeping at least two kings on side 1 and none on side 2. The game ends when the serfs are eaten on side 2. ∎

When n is large, Min can make a King in X and then eat all kings Max makes in Y (eating Y-serfs when Max makes no king there). This yields $\alpha_g(K_{m,n}) \leq m$ for all n. Surprisingly, Min can do better.

Thm. If $n \geq 2m + 2$, then $\alpha_g(K_{m,n}) \leq \lfloor m/2 \rfloor + 2$.

Pf. A complicated three-stage strategy for Min. ∎

The conjecture includes the statement that this bound holds for all n and is optimal when $n > 5m/2$.

Conj. $\alpha_g(K_{m,n}) = 2 + \min\{\left\lfloor \frac{n-m}{3} \right\rfloor, \left\lfloor \frac{m}{2} \right\rfloor\}$.