Theoretical Aspects of Intruder Search

Course Wintersemester 2015/16
Dynamic strategies on Trees

Elmar Langetepe

University of Bonn

November 3rd, 2015

Efficient Algorithm for Trees

Corollary 14: Computing a strategy for a tree T of size n that saves at least k vertices can be done in $O(n2^k k)$ time.

- Run above algorithm for $i = 1, \ldots, k$
- Sufficient!
- $\sum_{i=1}^{k} i2^i n \leq kn \sum_{i=1}^{k} 2^i = (2^{k+1} - 2)kn$

Subexponential bound:

Bound for k: Show $k \leq \sqrt{2n}$

Lemma 15: If a vertex at depth d is burning in an optimal strategy for an instance of the firefigther problem on trees, at least $\frac{1}{2}(d^2 + d)$ vertices are safe.

Proof:

- Optimal strategy, vertex v at depth d burning
- Guard at v_i in every depth $1, 2, \ldots, d$
- T_{v_i} has size $\geq d - i + 1$
- $\sum_{i=1}^{d}(d - i + 1) = \frac{1}{2}(d^2 + d)$

Subexponential bound:

Bound for k: Show $k \leq \sqrt{2n}$

Theorem 16: There is an $O\left(2^{\sqrt{2n}} n^{3/2}\right)$ algorithm for the firefigther problem on a tree of size n.

Proof:

- Run the algorithm for $k \leq \sqrt{2n}$: $(n \cdot 2^k \cdot k)$
- Above Lemma: Burning vertex at depth $\sqrt{2n}$, then $n + \sqrt{n/2} > n$ vertices safe? Contradiction!
- All vertices of depth $k = \sqrt{2n}$ has to be safe for an optimal strategy
- Suffices to use this bound!

Dynamic guards in Trees

- Stationary guards vs. dynamic guards!
- NP-hard for general graphs $\Rightarrow$ Trees
- Many different variants: Here *Clearing of edges!*
- Weights for the Corridors. Weights for the vertices.
- Recontamination, if weight is to small!
- Intruder has maximum speed.

Contiguous search strategy

Weighted Graphs $G = (V, E)$

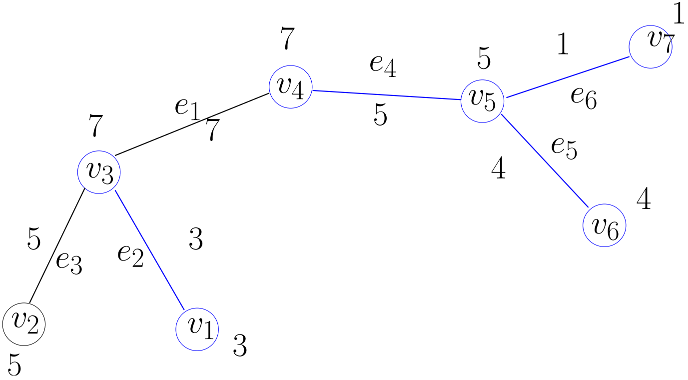
1. Place p guards on a vertex.
2. Move r guards along an edge.

The set of all *cleared* edges E_i after step i has to be connected!

- Edge weights $w(e)$, vertex weights $w(v)$ with $w(v) \geq w(e)$ for any $e = (v, u) \in E$
- Recontamination by non-protected paths
- Infinite speed for the Intruder
- Example: Blackboard

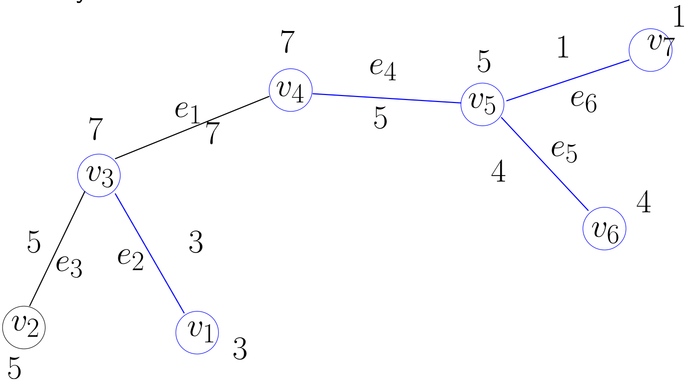
Optimal contiguous search strategy

Weighted Tree $T = (V, E)$, search number $\mathrm{cs}(T)$!



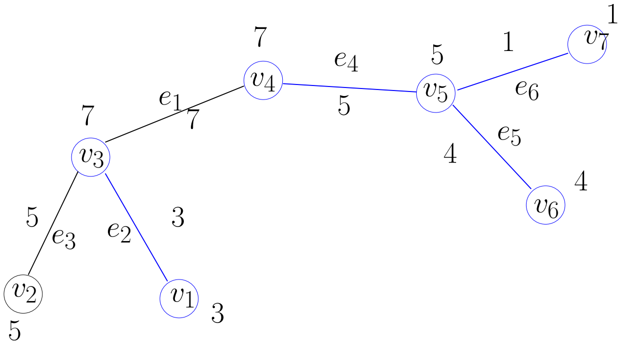
$(\emptyset, \{e_6\}, \{e_6, e_5\}, \{e_6, e_5, e_4\}, \{e_6, e_5, e_4, e_1\}, \{e_6, e_5, e_4, e_1, e_2\},$
$\{e_6, e_5, e_4, e_1, e_2, e_3\})$ 10 agents required! $\mathrm{cs}(T) = 10$!

Theorem 17: For any weighted tree T there is a monotone contiguous search strategy with $\mathtt{cs}(T)$ agents where all agents initially start at the same vertex b.



Boundary of edge subset!

- $X \subseteq E$
- *boundary vertices* $\delta(X)$:
  Vertices that have vertices incident to X and $E \setminus X$
- $w(X_i) := \sum_{v \in \delta(X_i)} w(v)$
- $w(\{e_4, e_5, e_6\}) = 7$ and $w(\{e_2\}) = 10$.



Optimal contiguous search strategy, Crusade definition

- $(X_0, X_1, \ldots, X_m)$ subsets $X_i \subseteq E$
- $X_0 = \emptyset$ and $X_m = E$
- $|X_i \setminus X_{i-1}| \leq 1$ for $1 \leq i \leq m$
- Connected if X_i connected for $1 \leq i \leq m$
- Frontier: $\max_{1 \leq i \leq m} w(X_i)$
- Progressive: $X_0 \subseteq X_1 \subseteq \cdots \subseteq X_m$ and $|X_i \setminus X_{i-1}| = 1$ for $1 \leq i \leq m$

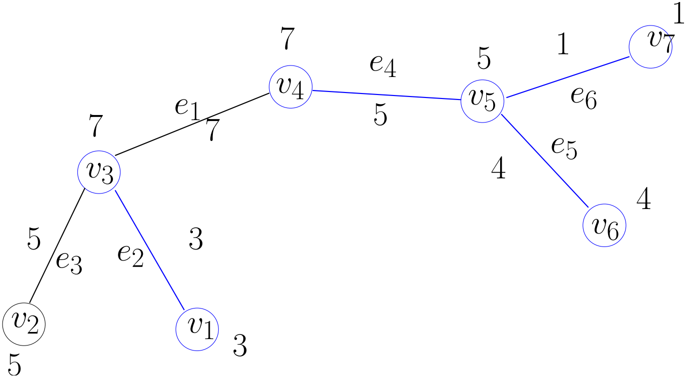
Contiguous search and connected crusade

- $\mathrm{cs}(T) \leq k$ and a contiguous search!
- X_i set after each step!
- Search step, at most one additional edge, means $|X_i \setminus X_{i-1}| \leq 1$
- X_i not destructed, means $w(X_i) \leq k$.
- X_i connected, because contiguous search
- $X_0 = \emptyset$ and $X_m = E$

Lemma 18: For $\mathrm{cs}(T) \leq k$ there is a connected crusade of frontier at most k.

Optimal contiguous search strategy

Weighted Tree $T = (V, E)$, search number $\mathrm{cs}(T)$!



$(\emptyset, \{e_6\}, \{e_6, e_5\}, \{e_6, e_5, e_4\}, \{e_6, e_5, e_4, e_1\}, \{e_6, e_5, e_4, e_1, e_2\},$
$\{e_6, e_5, e_4, e_1, e_2, e_3\})$ 10 agents required! $\mathrm{cs}(T) = 10$!

Lemma 19: For $\mathtt{cs}(T) \leq k$ there is a *progressive* connected crusade of frontier at most k.

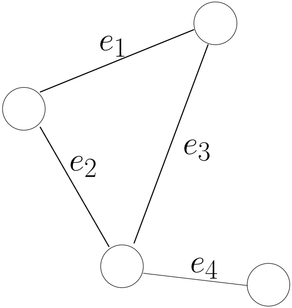
Connected crusades $C = (X_0, X_1, \ldots, X_m)$ of frontier at most k
Choose one with:

1. $\sum_{i=0}^{m}(w(X_i)+1)$ is minimum.
2. Amog all crusade satisfying condition 1. choose one with: $\sum_{i=0}^{m}|X_i|$ is minimum.

Has to exist, show that this is progressive:
$|X_i \setminus X_{i-1}| = 1$ for $1 \leq i \leq m$

Difference: Connected Crusade, Progessive Crusade

Connected crusade:
$(\emptyset, \{e_1\}, \{e_1, e_2\}, \{e_2\}, \{e_2, e_3\}, \{e_1, e_2, e_3\}, \{e_3, e_4\}, \{e_1, e_3, e_4\},$
$\{e_1, e_2, e_3, e_4\})$: $|X_i \setminus X_{i-1}| \leq 1$ for $1 \leq i \leq m$
Pogressive con. crusade:
$(\emptyset, \{e_1\}, \{e_1, e_2\}, \{e_1, e_2, e_3\}, \{e_1, e_2, e_3, e_4\})$,
$|X_i \setminus X_{i-1}| = 1$ for $1 \leq i \leq m$, $X_0 \subseteq X_1 \subseteq \cdots \subseteq X_m$



Contiguous search and connected crusade

1. $\sum_{i=0}^{m}(w(X_i)+1)$ is minimum.
2. Amog all crusade satisfying condition 1. choose one with: $\sum_{i=0}^{m}|X_i|$ is minimum.

- Assume: $C=(X_0,X_1,\ldots,X_m)$ with $|X_i \setminus X_{i-1}|=0$
- Take: $C'=(X_0,\ldots,X_{i-1},X_{i+1},\ldots,X_m)$, Condition 1.!
- This means $X_i \subseteq X_{i-1}$.
- $|X_{i+1} \setminus X_{i-1}| \leq 1$ from $|X_{i+1} \setminus X_i| \leq 1$ and $X_i \subseteq X_{i-1}$,
- Connected!
- Can assume: $|X_i \setminus X_{i-1}|=1$ for $1 \leq i \leq m$!

Progressive connected crusade, frontier at most k

1. $\sum_{i=0}^{m}(w(X_i)+1)$ is minimum.
2. Amog all crusade satisfying condition 1. choose one with: $\sum_{i=0}^{m}|X_i|$ is minimum.

- Prove $X_i \subseteq X_{i-1}$!
- Case 1.: $w(X_{i-1} \cup X_i) < w(X_i)$
- $C' = (X_0, \ldots, X_{i-1}, X_{i-1} \cup X_i, X_{i+1}, \ldots, X_m)$, Cond. 1.!
- X_i and X_{i-1} connected, $X_{i-1} \cup X_i$ is connected since $|X_i \setminus X_{i-1}| = 1$
- $|X_{i+1} \setminus (X_{i-1} \cup X_i)| \leq 1$ since $|X_{i+1} \setminus X_i| = 1$.
  If $|X_{i+1} \setminus (X_{i-1} \cup X_i)| = 0$ go back to former case!
- Case 2.: $w(X_{i-1} \cup X_i) \geq w(X_i)$

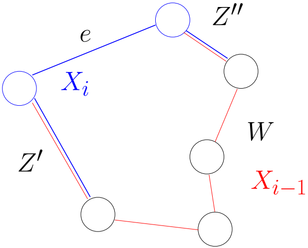
Progressive connected crusade, frontier at most k

1. $\sum_{i=0}^{m}(w(X_i)+1)$ is minimum.
2. Amog all crusade satisfying condition 1. choose one with: $\sum_{i=0}^{m}|X_i|$ is minimum.

- Prove $X_i \subseteq X_{i-1}$!
- Case 2.: $w(X_{i-1} \cup X_i) \geq w(X_i)$
- Exercise: $w(A \cup B) + w(A \cap B) \leq w(A) + w(B)$ link sets A, B
- $w(X_{i-1} \cap X_i) \leq w(X_i)$ for $1 \leq i \leq m$
- $C'' = (X_0, \ldots, X_{i-2}, X_{i-1} \cap X_i, X_{i+1}, \ldots, X_m)$
- Cond. 2.! $|X_{i-1} \cap X_i| \geq |X_{i-1}|$ which gives $X_{i-1} \subseteq X_i$
- $|X_i \setminus (X_i \cap X_{i-1})| = |X_i \setminus X_{i-1}| = 1$ and $|(X_i \cap X_{i-1}) \setminus X_{i-2}| \leq |X_{i-1} \setminus X_{i-2}| \leq 1$
- Show that C'' is connected!!

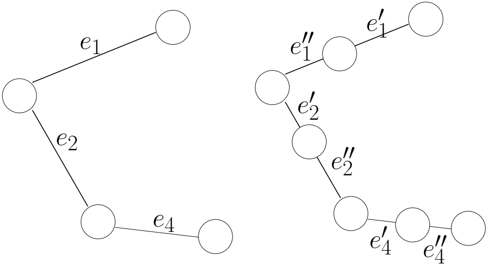
Progressive connected crusade, frontier at most k

- $C'' = (X_0, \dots, X_{i-2}, X_{i-1} \cap X_i, X_{i+1}, \dots, X_m)$ connected?
- Ass. $X_{i-1} \cap X_i$ not connected!
- $\{e\} = X_i \setminus X_{i-1}$ and $W = X_{i-1} \setminus X_i$ and $Z = X_{i-1} \cap X_i$. By assumption $Z = Z' \cup Z''$ where Z' and Z'' do not share a vertex.
- Contrad. T is a tree, $X_{i-1} \cap X_i$ is also connected.



Lemma 19: For $\texttt{cs}(T) \leq k$ there is a *progressive* connected crusade with frontier at most k.

- Build strategy from *progressive* connected crusade frontier at most k!
- First, double the edges T, T'!



Lemma 20: Let T' be a tree so that every link has at least one vertex of degree 2. If there is a progressive connected crusade of frontier $\leq k$ in T', there is a monotone contiguous search strategy using $\leq k$ guards in T' and the guards can be initially placed at a single vertex v_1.

Proof: Inductive argument!

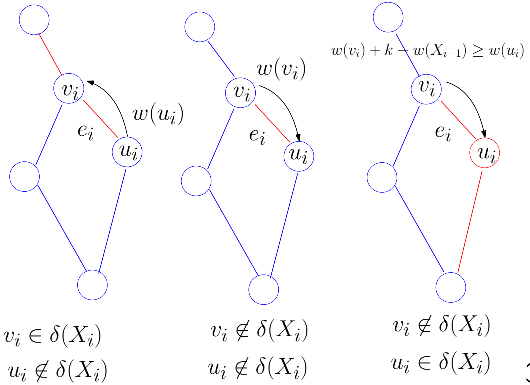
- pcc. $C = (X_0, X_1, \ldots, X_m)$ frontier $\leq k$
- $e_i = (v_i, u_i) := X_i \setminus X_{i-1}$, this order
- Start with k guards at v_i
- $w(X_1) = w(v_1) + w(u_1) \leq k$, $w(e_1) \leq w(u_1)$
- move $w(u_1)$ searchers along w_1

Lemma 20: T' (every link has vertex of degree 2) and progressive connected crusade of frontier $\leq k$. Monotone contiguous strategy with the same bound!

Proof:

- $e_1, \ldots, e_{i-1}$ without recontaminations
- $e_i = (v_i, u_i)$ incident to X_{i-1}, $v_i \in \delta(X_{i-1})$
- Case 1: $w(X_{i-1}) + w(u_i) \leq k$:
  Clear link e_i by $w(u_i)$ agents move from v_i to u_i.
- Case 2: $w(X_{i-1}) + w(u_i) > k$
- Not both vertices v_i, u_i in $\delta(X_i)$
- $v_i \in \delta(X_{i-1})$. Assume $v_i \in \delta(X_i)$
- $\deg(v_i) > 2$ and $\deg(u_i) = 2$
- $u_i \in \delta(X_i)$ implies link $f_i \neq e_i$ containing u_i has to be contaminated and $u_i \notin \delta(X_{i-1})$
- $w(X_i) = w(X_{i-1}) + w(u_i)$ Contradiction!

Case 2: $w(X_{i-1}) + w(u_i) > k$ and not both vertices v_i, u_i in $\delta(X_i)$



3.

$w(X_i) = w(X_{i-1}) - w(v_i) + w(u_i)$ and at least $w(v_i)$ guards at v_i.

Move all $k - w(X_{i-1})$ free guards to v_i.

$$w(v_i) + k - w(X_{i-1}) \geq w(v_i) + w(X_i) - w(X_{i-1}) \geq w(u_i)$$

Lemma 21: Any contiguous monotone strategy for T' can be translated to a contiguous monotone strategy for T with the same number k of agents.

Proof:
Let $e' = (x, y)$ and $e'' = (y, z)$ links stemming extension e.

If q guards move from x to y or z to y, they stay in place in T.

If q guards move from y to x or from y to z, they move from z to x or from x to z in T, respectively.

Lemma 22: Any contiguous monotone strategy for T with k agents can be translated to a contiguous monotone strategy for T' with the same number k of agents.

Proof:
A move along an edge $e = (u, v)$ in T is splitted into two moves along e' and e''.

u is kept safe: If the move clears $e = (u, v)$, then $q \geq w(e)$ have traversed e.

From the construction q searchers are also enough for $w(e) = w(e') = w(e'')$ and the weight $w(e)$ of the intermediate vertex.

Theorem 17: For any weighted tree T there is a monotone contiguous search strategy with $\mathtt{cs}(T)$ agents where all agents initially start at the same vertex b.

- $\mathtt{cs}(T') \leq \mathtt{cs}(T)$ (Theorem 22)
- Connected crusade of frontier $\mathtt{cs}(T')$ in T' (Lemma 18)
- Monotone contiguous strategy for $\mathtt{cs}(T')$ in T' with start vertex b (Lemma 20)
- Monotone contiguous strategy for $\mathtt{cs}(T') = \mathtt{cs}(T)$ in T with start vertex b (Lemma 21)

Design of a strategy: Example!

Startvertex v and order of the subtrees:

$$\mathrm{cs}(T_v(z)) = \max\{\mathrm{cs}(T_v(z_1)), \mathrm{cs}(T_v(z_2)) + w(z)\}$$