Theoretical Aspects of Intruder Search

Course Wintersemester 2015/16
Dynamic strategies on Trees

Elmar Langetepe

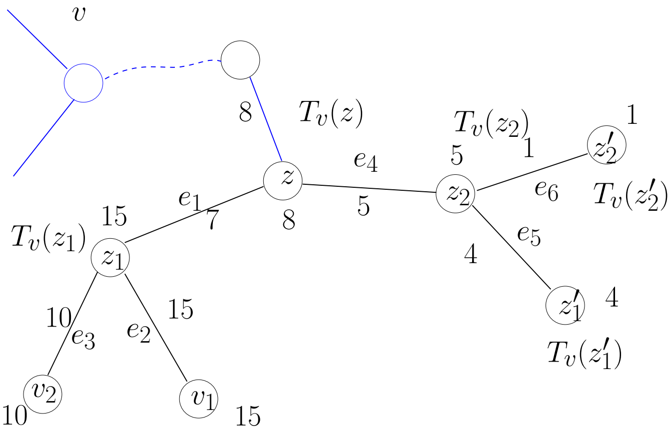
University of Bonn

November 10th, 2015

Design of a strategy: Example!

Startvertex v and order of the subtrees:

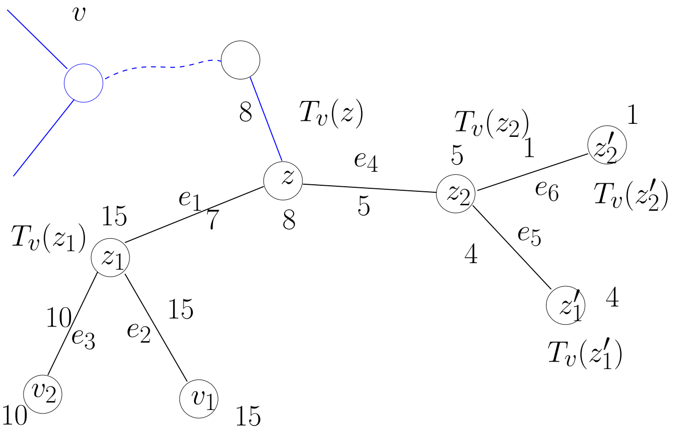
$$\mathrm{cs}(T_v(z)) = \max\{\mathrm{cs}(T_v(z_1)), \mathrm{cs}(T_v(z_2)) + w(z)\}$$



Design of a strategy: Example! Barriere et al. Flaw!

Startvertex v and order of the subtrees:

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



Lemma 23: Let $z_1, \ldots, z_d$ be the $d \geq 2$ children of a vertex z in T_v and assume that $\mathrm{cs}(T_v(z_i)) \geq \mathrm{cs}(T_v(z_{i+1}))$ for $i = 1, \ldots, d-1$. We have

$$\mathrm{cs}(T_v(z)) = \max\{\mathrm{cs}(T_v(z_1)), \mathrm{cs}(T_v(z_2)) + w(z)\} \qquad (1)$$

if the tree T is a tree with unit weights.

Proof:

- $\mathrm{cs}(T_v(z)) \geq \mathrm{cs}(T_v(z_1))$, order of cleaning
- Case 1: $\mathrm{cs}(T_v(z_1)) \geq \mathrm{cs}(T_v(z_2) + w(z)$
- Clear $T_v(z)$, set $w(z)$ on z, clear all $T_v(z_i)$ by $\mathrm{cs}(T_v(z_1)$ agents but $T_v(z_1)$ last
- Case 2: $\mathrm{cs}(T_v(z_1)) < \mathrm{cs}(T_v(z_2)) + w(z)$ is necessary!

Design of a strategy: Example! Barriere et al. Flaw!

Lemma 23: Let $z_1, \dots, z_d$ be the $d \geq 2$ children of a vertex z in T_v and assume that $\mathrm{cs}(T_v(z_i)) \geq \mathrm{cs}(T_v(z_{i+1}))$ for $i = 1, \dots, d-1$. We have

$$\mathrm{cs}(T_v(z)) = \max\{\mathrm{cs}(T_v(z_1)), \mathrm{cs}(T_v(z_2)) + w(z)\} \qquad (2)$$

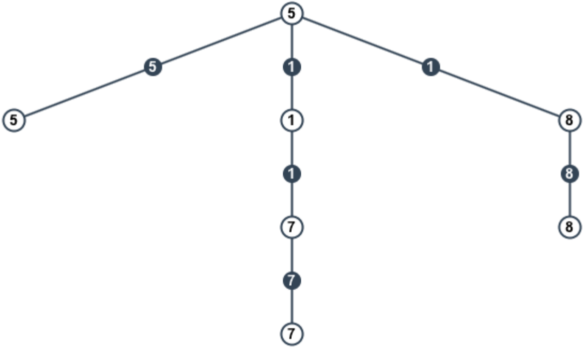
if the tree T is a tree with unit weights.

Case 2: $\mathrm{cs}(T_v(z_1)) < \mathrm{cs}(T_v(z_2)) + w(z)$
Show: $\mathrm{cs}(T_v(z_2)) + w(z) - 1$ not sufficient

1. $T_v(z_2)$ is cleared before $T_v(z_1)$: While $\mathrm{cs}(T_v(z_2))$ agents clear $T_v(z_2)$ there are only $w(z) - 1 = 0$ agents left for blocking a vertex in $T_v(z_1)$. Recontamination!
2. $T_v(z_1)$ is cleared before $T_v(z_2)$): While $\mathrm{cs}(T_v(z_1))$ agents clear $T_v(z_1)$ there are no more $w(z) - 1 = 0$ agents left for blocking a vertex in $T_v(z_2)$ (because $\mathrm{cs}(T_v(z_1)) = \mathrm{cs}(T_v(z_2))$). Recontamination!

Design of a strategy: Example! Barriere et al. Flaw!

$$\mathrm{cs}(T_v(z)) = \max\{\mathrm{cs}(T_v(z_1)), \mathrm{cs}(T_v(z_2)) + w(z)\} \qquad (3)$$



$\max\{\mathrm{cs}(T_x(z_1)), \mathrm{cs}(T_x(z_2)) + w(v)\} = \max\{8, 7+5\} = 12$
But 10 agents are also sufficient!

Direct consequence for unit weighted trees

Corollary 24: For a unit weighted tree T of size n and for a given starting vertex v we can compute the optimal monotone contiguous strategy starting at v in $O(n)$ time. An overall optimal contiguous strategy can be computed in $O(n^2)$.

Proof: For any root v compute the values $\mathtt{cs}(T_v(x))$ starting from the leafes. Do this for all $v \in T$.

Compute the information in one walkthrough!
Local recursive labeling: $\lambda_x(e)$ for the links $e = (x, y)$ adjacent to x.
Let $e = (x, y)$ be a link incident to x.

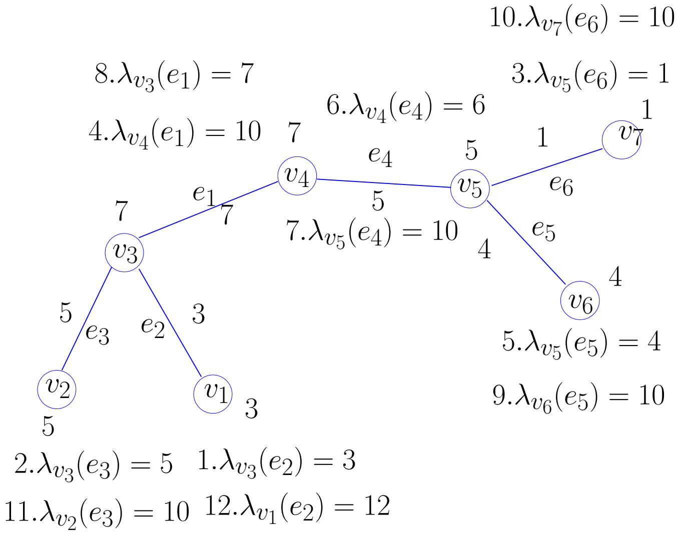
1. If y is a leaf, set $\lambda_x(e) = w(y)$.
2. Otherwise, let d be the degree of y and let $x_1, \ldots, x_{d-1}$ be the incident vertices of y different form x. Let $\lambda_y(y, x_i) =: l_i$ and $l_i \geq l_{i+1}$. Then,

$$\lambda_x(e) := \max\{l_1, l_2 + w(y)\} .$$

Computed by message sending algorithm

1. Start with the leaves and for any leaf y and for $e = (x, y)$ send a message $l = w(y)$ to x. After receiving this messages, x sets $\lambda_x(e) = l$.
2. Consider a vertex y of degree d that has received at least $d - 1$ messages l_i from the incident certices $x_1, \ldots, x_{d-1}$ and let x be the remaining incident vertex. Let $l_i \geq l_{i+1}$. Send a message $l = \max\{l_1, l_2 + w(y)\}$ to x, after receiving the message x, set $\lambda_x((x, y)) = l$.

Example for general tree



Labeling by message sending!

Lemma 24: The links of a tree T can be labeled with labels λ_x by the above message sending algorithm by $O(n)$ messages in total.

Proof by construction!

Connection $\mathrm{cs}(T_x(y)) = \lambda_x(e)$

Lemma 26: For a unit weighted tree $T = (V, E)$ and an edge $e = (x, y) \in E$ we have $\mathrm{cs}(T_x(y)) = \lambda_x(e)$.

Proof: By induction!

- y leaf and $\lambda_x(e) = w(y)$ for $h(y) = 0$
- Statement holds for $0 \leq h(y) < k$ and consider $h(y) = k$
- $e = (x, y)$, $x_1, \ldots, x_d$ the $d \geq 1$ children of y in $T_x(y)$
- $T_y(x_i) = \lambda_y((y, x_i)$ by induction hypothesis, $T_y(x_i) = T_x(x_i)$ by definition
- $\mathrm{cs}(T_x(x_i)) \geq \mathrm{cs}(T_x(x_{i+1}))$ for $i = 1, \ldots, d - 1$.
- Recursion for $T_x(y)$ and $\lambda_x((x, y))$ identical!

Final computation!

Order all $\lambda_v((v, x_i)$ for all $i = 1, \dots, d$ incident edges (v, x_i) so that $\lambda_v((v, x_i)) \geq \lambda_v((v, x_{i+1}))$, compute

$$\mu(v) = \max\{\lambda_v((v, x_1)), \lambda_v((v, x_2)) + w(v)\}\,. \quad (4)$$

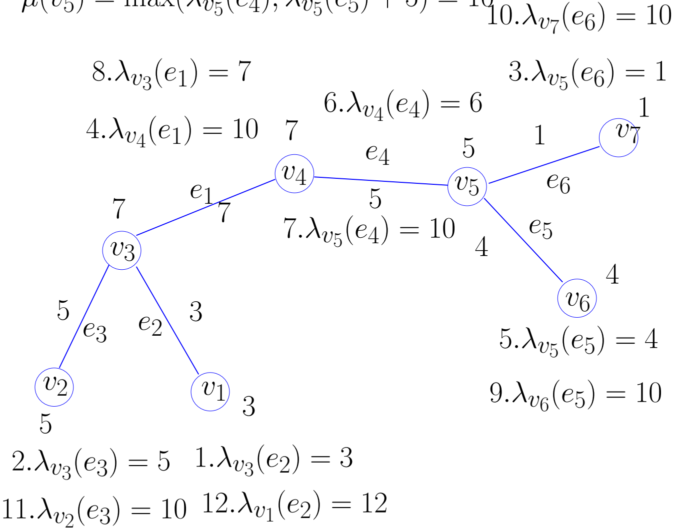
$\mu(v) = \mathrm{cs}(T_v)$ and $\min_{v \in V} \mu(v) = \mathrm{cs}(T)$.

Strategy: By the increasing order of the values λ_x at vertex x!

Final computation! General example!

$$\mu(v_3) = \max(\lambda_{v_3}(e_1), \lambda_{v_3}(e_3) + 7) = 12$$

$$\mu(v_5) = \max(\lambda_{v_5}(e_4), \lambda_{v_5}(e_5) + 5) = 10$$



Final result for unit weighted trees!

Theorem 27: On optimal contiguous strategy for a unit weighted tree $T = (V, E)$ can be computed in $O(n)$ time and space.

Proof:

- Calc. messages an μ values in $O(n)$ time
- Register only three greatest values for every vertex

Example: Applet!

Lower and upper bounds for the contiguous search

Theorem 28: For unit weights and for any number of vertices n, we have $\lfloor \log_2 n \rfloor - 1 \leq \mathrm{cs}(n) \leq \lfloor \log_2 n \rfloor$.

Two directions!

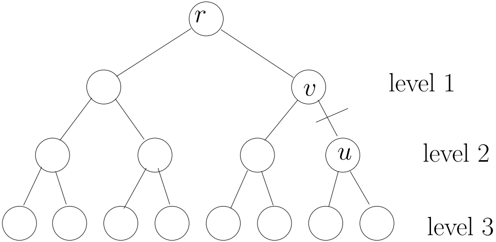
Lemma 29: For every $n \geq 1$ we find trees T_n with $\mathrm{cs}(T_n) \geq \lfloor \log_2(\frac{2}{3}(n+1)) \rfloor \geq \lfloor \log_2 n \rfloor - 1$.

Proof:

- Case 1: n equals $2^k - 1$
- Choose complete binary tree
- $\mathrm{cs}(T_n) = k - 1 = \log_2(n+1) - 1 \geq \log_2 \lfloor (\frac{2}{3}(n+1)) \rfloor$

- Case 1: n equals $2^k - 1$
- $\mathsf{cs}(T_n) = k - 1 = \log_2(n+1) - 1 \geq \log_2\lfloor(\frac{2}{3}(n+1))\rfloor$

$k = 4$ and $n = 2^k - 1$



$\lambda_v((v,u)) = k - \text{level}(u)$

$\lambda_u((v,u)) = k - 1$

$\mu(r) = k$ and $\mu(u \neq r) = k - 1$

Lower and upper bounds for the contiguous search

Lemma 29: For every $n \geq 1$ we find trees T_n with $\mathtt{cs}(T_n) \geq \lfloor \log_2(\frac{2}{3}(n+1)) \rfloor \geq \lfloor \log_2 n \rfloor - 1$.

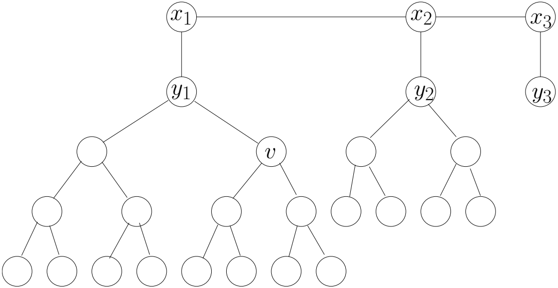
Proof:

- Case 1: n equals $2^k - 1$
- Case 2: n does not equal $2^k - 1$
- $n = \sum_{i=1}^{r} 2^{\alpha_i}$ with $\alpha_1 > \alpha_2 > \cdots > \alpha_r$.
- $n = 11010$ in binary representation with $\alpha_1 = 4$,$\alpha_2 = 3$, $\alpha_3 = 2$.
- Chain of vertices $x_1, x_2, \ldots, x_r$
- For any x_i connect complete binary tree T_{α_i} of size $2^{\alpha_i} - 1$
- $2^{\alpha_1} - 1 < n < 2^{\alpha_1 + 1} - 1$ and require $\mathtt{cs}(T_n) = \alpha_1 \geq \log_2(n+1) - 1 \geq \log_2 \lfloor (\frac{2}{3}(n+1)) \rfloor$

Lower and upper bounds for the contiguous search

- Case 2: n does not equal $2^k - 1$
- $\mathtt{cs}(T_n) = \alpha_1 \geq \log_2(n+1) - 1 \geq \log_2\lfloor(\frac{2}{3}(n+1))\rfloor$

$$n = 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 11010$$



$$\lambda_{y_1}((v, y_1)) = \alpha_1 - 1$$
$$\lambda_{y_1}((x_1, y_1)) = \alpha_2 + 1 = \alpha_1$$

Lower and upper bounds for the contiguous search

Lemma 30: For every $n \geq 1$ and unit weights, $\lfloor \log_2 n \rfloor$ agents are sufficient for a contiguous search strategy.

Proof: Arbitrary tree T_r with root r, $\mathrm{cs}(T)$, construct T'_r

1. For a node x and its $d > 2$ children $x_1, x_2, \ldots, x_d$ ordered by $\mathrm{cs}(T_r(x_i)) \geq \mathrm{cs}(T_r(x_{i+1}))$ remove all $T_r(x_i)$ for $i > 2$.
2. For a node x with two children x_1 and x_2 and $\mathrm{cs}(T_r(x_1)) > \mathrm{cs}(T_r(x_2))$, remove $T_r(x_2)$.
3. For a node $x \neq r$ with only one child x_1, remove x and connect x_1 to the parent of x.
4. If there are more than two vertices left, and r has only one child x_1, remove x_1 and connect the children of x_1 to r.

Lemma 30: For every $n \geq 1$ and unit weights, $\lfloor \log_2 n \rfloor$ agents are sufficient for a contiguous search strategy.

Proof:

- Agents required for T and T_r are the same, computation of $\mu(r)$ in T_r use the same values.
- Weights restricted to one, rule 2. is correct by $\mathtt{cs}(T_r(x_1)) \geq \mathtt{cs}(T_r(x_2)) + 1$.
- Complete binary tree? 1. Binary! 2. Complete

Lower and upper bounds for the contiguous search

1. Binary: Any inner vertex has no more than 2 chidren! Rule 1 and 2!

Rule three deletes internal nodes with one child except for the root. Rule 4 make the root have 2 or 0 children.

1. For a node x and its $d > 2$ children $x_1, x_2, \ldots, x_d$ ordered by $\mathtt{cs}(T_r(x_i)) \geq \mathtt{cs}(T_r(x_{i+1}))$ remove all $T_r(x_i)$ for $i > 2$.
2. For a node x with two children x_1 and x_2 and $\mathtt{cs}(T_r(x_1)) > \mathtt{cs}(T_r(x_2))$, remove $T_r(x_2)$.
3. For a node $x \neq r$ with only one child x_1, remove x and connect x_1 to the parent of x.
4. If there are more than two vertices left, and r has only one child x_1, remove x_1 and connect the children of x_1 to r.

1. Complete: T'_x not complete and no subtree in T'_x incomplete

1. For a node x and its $d > 2$ children $x_1, x_2, \ldots, x_d$ ordered by $\mathtt{cs}(T_r(x_i)) \geq \mathtt{cs}(T_r(x_{i+1}))$ remove all $T_r(x_i)$ for $i > 2$.
2. For a node x with two children x_1 and x_2 and $\mathtt{cs}(T_r(x_1)) > \mathtt{cs}(T_r(x_2))$, remove $T_r(x_2)$.
3. For a node $x \neq r$ with only one child x_1, remove x and connect x_1 to the parent of x.
4. If there are more than two vertices left, and r has only one child x_1, remove x_1 and connect the children of x_1 to r.