Online Motion Planning MA-INF 1314
Searching

Elmar Langetepe
University of Bonn
Rep.: Navigation

- Touch sensor, Target coordinates, Start $s$, Target $t$, Storage, Sojourner
- Actions:
  - Move toward the target
  - Move along the boundary
  - Sequence of Leave-Points $l_i$, Hit-Points $h_i$
Rep.: BUG1 Strategy: Lumelsky/Stepanov

Toward target, surround obstacle, best leave point, toward target!
Rep: Analysis BUG1 Strategy

- **Theorem** Strategy Bug1 is correct!

- **Theorem** Successful Bug1-path $\Pi_{\text{Bug1}}$ from start $s$ to target $t$:
  \[ |\Pi_{\text{Bug1}}| \leq D + \frac{3}{2} \sum_i \text{UP}_i. \]

- **Theorem** For any strategy $S$, for arbitrary large $K > 0$, there exists examples for any $D > 0$, such that for any arbitrarily small $\delta > 0$ we have: $|\Pi_S| \geq K \geq D + \sum \text{UP}_i - \delta$.

- **Korollar** Bug1 is $\frac{3}{2}$-competitive against any *online* strategy.
Rep. LB: $|\Pi_S| \geq K \geq D + \sum UP_i - \delta$

- Virtual horse-shoe, width $2W$, thickness $\epsilon \ll \delta$, length $L$, dist. $D$
- Virtual gets concrete by touch
- Roughly surround any obstacle, by any strategy!

$C = \sqrt{D^2 + W^2}$
Rep.: BUG2 Strategy

Line $G$ passing $st$, toward target, surround obstacle, shorter distance on $G$, toward target!
Rep.: Analysis BUG2 Strategy

- **Lemma** Let $n_i$ denote the number of intersection of $G$ with relevant obstacle $P_i$. Bug2 meets any point on $P_i$ at most $\frac{n_i}{2}$ times.
- **Corollary** Bug2 is correct!
- **Theorem** Bug2-path $\Pi_{Bug2}$ from $s$ to $t$. We have:
  \[ |\Pi_{Bug2}| \leq D + \sum_i \frac{n_i \cdot UP_i}{2}. \]
Rep.: Change I

Change I, use former Leave/Hit Points once for !

\[ |\Pi_{\text{Change I}}| \leq D + 2 \sum_i \text{UP}_i. \] This is a tight bound!

Theorem: Change I requires at most path length

Exercise!
Different models

- Sensor with range: Circle around current point
- Short-cut for BUG2: VisBug
- Many others
Searching for a goal!

- Coordinates of the target unknown: Searching vs. Navigation
- Polygonal environment
- Full sight: Visibility polygon

**Def.** Let $P$ be a simple polygon and $r$ a point with $s \in P$. The visibility polygon of $r$ w.r.t. $P$, $\text{Vis}_P(r)$, is the set of all points $q \in P$, such that the segment $\overline{rq}$ is fully inside $P$.

- Alg. Geom.: Compute in $O(n)$ time! Offline!
Corridors (without sight)

- 2-ray search: Find door along a wall!
- Compare to shortest path to the door, competitive?
- Reasonable strategy: Depth $x_1$ right, depth $x_2$ left and so on
- Start-situation: $2x_1 \geq C\epsilon$, for any $C > 0$ ex. $\epsilon$
- Additive constant or goal is at least step 1 away!
- Local worst-case, not visited at $d$, once back!
- Find strategy, such that: $\sum_{i=1}^{k+1} 2x_i + x_k \leq Cx_k$

![Diagram of corridors with distances labeled]
Corridors

- Worst-case, not visited at $d$, once back!
- Find strategy, such that: $\sum_{i=1}^{k+1} 2x_i + x_k \leq C x_k$
- Minimize: $\frac{\sum_{i=1}^{k+1} 2x_i + x_k}{x_k} = 1 + 2 \frac{\sum_{i=1}^{k+1} x_i}{x_k}$
- $x_i = 2^{i-1}$, gives ratio $C = 9$
- Proof: Blackboard!

\[ \begin{array}{cccc}
32 & \hline & 8 & \hline & 2 & \hline & 16
\end{array} \]
Theorem Opt. of exponential solution: Gal 1980

- Strategy: Sequence $X = f_1, f_2, \ldots$
- Minimize functional $F_k(f_1, f_2, \ldots) := \sum_{i=1}^{k+1} \frac{f_i}{f_k}$ for all $k$
- More precisely $\inf_Y \sup_k F_k(Y) = C$ und $\sup_k F_k(X) = C$
- In general: Functional $F_k$ continuous/unimodal: Unimodal:
  $F_k(A \cdot X) = F_k(X)$ and $F_k(X + Y) \leq \max\{F_k(X), F_k(Y)\}$
- Some other helpful conditions!
- I.e.: $F_{k+1}(f_1, \ldots, f_{k+1}) \geq F_k(f_2, \ldots, f_{k+1})$
- **Theorem** Exponential function minimizes $F_k$:
  \[
  \sup_k F_k(X) \geq \inf_a \sup_k F_k(A_a)
  \]
  mit $A_a = a^0, a^1, a^2, \ldots$ und $a > 0$. 

Online Motion Planning Searching 23.5.2017 ©Elmar Langetepe
Example: Exponential function

- $F_k(f_1, f_2, \ldots) := \sum_{i=1}^{k+1} \frac{f_i}{f_k}$ for all $k$.
- Unimodal $F_k(A \cdot X) = F_k(X)$ and $F_k(X + Y) \leq \max\{F_k(X), F_k(Y)\}$?
- $\sum_{i=1}^{k+1} A \cdot f_i = \sum_{i=1}^{k+1} f_i$
- $F_k(X + Y) \leq \max\{F_k(X), F_k(Y)\}$?
- Follows from $\frac{a}{b} \geq \frac{c}{d} \iff \frac{a+c}{d+b} \leq \frac{a}{b}$
- Simple equivalence !
- Optimize: $f_k(a) := \sum_{i=1}^{k+1} \frac{a^i}{a^k}$
- Minimized by $a = 2$
Theorem Gal 1980

If functionaly $F_k$ has the following properties:

i) $F_k$ is continuous,

ii) $F_k$ is unimodal: $F_k(A \cdot X) = F_k(X)$ and $F_k(X + Y) \leq \max\{F_k(X), F_k(Y)\}$,

iii) $\lim \inf_{a \to \infty} F_k \left( \frac{1}{a^{k+i}}, \frac{1}{a^{k+i-1}}, \ldots, \frac{1}{a}, 1 \right) = \lim \inf_{\epsilon_{k+i}, \epsilon_{k+i-1}, \ldots, \epsilon_1 \to 0} F_k \left( \epsilon_{k+i}, \epsilon_{k+i-1}, \ldots, \epsilon_1, 1 \right)$,

iv) $\lim \inf_{a \to 0} F_k \left( 1, a, a^2, \ldots, a^{k+i} \right) = \lim \inf_{\epsilon_{k+i}, \epsilon_{k+i-1}, \ldots, \epsilon_1 \to 0} F_k \left( 1, \epsilon_1, \epsilon_2, \ldots, \epsilon_{k+i} \right)$,

v) $F_{k+1}(f_1, \ldots, f_{k+i+1}) \geq F_k(f_2, \ldots, f_{k+i+1})$.

Then: $\sup_k F_k(X) \geq \inf_a \sup_k F_k(A_a)$ with $A_a = a^0, a^1, a^2, \ldots$ and $a > 0$. 

---

Online Motion Planning Searching 23.5.2017 ©Elmar Langetepe
Application m-ray search

- Arbitrary $m$, not competitive, Fig.!
- $2m - 1$ vs. $1$
- Fixed $m$, infinite rays!
- Ass.: Rays in fixed order and increasing depth
- Tupel $(f_j, J_j)$: depth, next visit!
Applicatiot m-ray search

- Ass.: \((f_j, J_j), J_j = j + m, f_j \geq f_{j-1}\)
- Visit rays in fixed order, increasing depth
- \(F_k(f_1, f_2, \ldots) := \frac{f_k + 2 \sum_{i=1}^{k+m-1} f_i}{f_k}\) for all \(k\)
- (Gal) Exp.-function minimizes \(F_k\):
  \[ \sup_k F_k(X) \geq \inf_a \sup_k F_k(A_a) \]
  with \(A_a = a^0, a^1, a^2, \ldots\) and \(a > 1\), optimal \(a = \frac{m}{m-1}\)
- Ratio: \(C = 1 + 2m \left( \frac{m}{m-1} \right)^{m-1}\) opt.
m-ray search

- **Lemma** There is an optimal m-ray search strategy \((f_1, f_2, \ldots)\) that visits the rays in a fixed order and with increasing depth.
- periodic and monotone: \((f_j, J_j), J_j = j + m, f_j \geq f_{j-1}\)
- Second part: Proof blackboard! Change strategy! Conditions!

![Diagram of m-ray search with rays labeled from f1 to f10]
Other approach: Optimality for equations!

- Reasonable strategy, ratio: \[ \sum_{i=1}^{k+1} \frac{2x_i + x_k}{x_k} = 1 + 2 \sum_{i=1}^{k+1} \frac{x_i}{x_k} \]

- Ass.: \( C \) optimal, \[ \sum_{i=1}^{k+1} \frac{x_i}{x_k} \leq \frac{(C-1)}{2} \]

- There is strategy \((x'_1, x'_2, x'_3 \ldots)\) s. th. \[ \sum_{i=1}^{k+1} \frac{x'_i}{x'_k} = \frac{(C-1)}{2} \] for all \( k \)

- Monotonically increasing in \( x'_j \) (\( j \neq k \)), decreasing in \( x'_k \)

- First \( k \) with: \[ \sum_{i=1}^{k+1} \frac{x_i}{x_k} < \frac{(C-1)}{2} \], decrease \( x_k \)

- \[ \sum_{i=1}^{k} \frac{x_i}{x_{k-1}} < \frac{(C-1)}{2} \], \( x_{k-1} \) decrease etc., monotonically decreasing sequence, bounded, converges! Non-constructive!
Other approach: Optimality for equations!

- Set: \( \sum_{i=1}^{k+1} x_i' = \frac{(C-1)}{2} \) for all \( k \)
- \( \sum_{i=1}^{k+1} x_i' - \sum_{i=1}^{k} x_i' = \frac{(C-1)}{2} (x_k' - x_{k-1}') \)
- Thus: \( C' (x_k' - x_{k-1}') = x_{k+1}' \), Recurrence!
- Solve a recurrence! Analytically! Blackboard!
- Characteristic polynom: No solution \( C' < 4 \)
- \( x_i' = (i + 1)2^i \) with \( C' = 4 \) is a solution! Blackboard! Optimal!

\[
\begin{align*}
x_1 & \quad x_2 & \quad x_3 \\
x_4 & \quad x_5 & \quad x_6 \\
& \quad & \\
& \quad & \\
& \quad & d
\end{align*}
\]
2-ray search, restricted distance

- Assume goal is no more than dist. \( \leq D \) away
- Exactly \( D \)! Simple ratio 3!
- Find optimal strategy, minimize \( C \)
- Vice-versa: \( C \) is given! Find the largest distance \( D \) (reach \( R \)) that still allows \( C \) competitive search.
- One side with \( f_{\text{Ende}} = R \), the other side arbitrarily large!
2-ray search, maximal reach $R$

- $C$ given, optimal reach $R$!
- **Theorem** The strategy with equality in any step maximizes the reach $R$!
- Strategy: \[ \sum_{i=1}^{k+1} \frac{x_i}{x_k} = \frac{(C-1)}{2}, \text{first step: } x_1 = \frac{(C-1)}{2} \]
- Recurrence: $x_0 = 1, x_{-1} = 0, x_{k+1} = \frac{(C-1)}{2}(x_k - x_{k-1})$
- Strategy is optimal! By means of the Comp. Geom. lecture!
2-ray search, maximal reach $R$

- $f(C) :=$ maximal reach depending on $C$
- Bends are more steps!
2-ray search, given distance $R$

- $f(C) :=$ maximal reach depending on $C$
- Rotate, $R$ given, binary search!