

Methoden der Offline Bewegungsplanung

Trapezzerlegung, BA-Themen

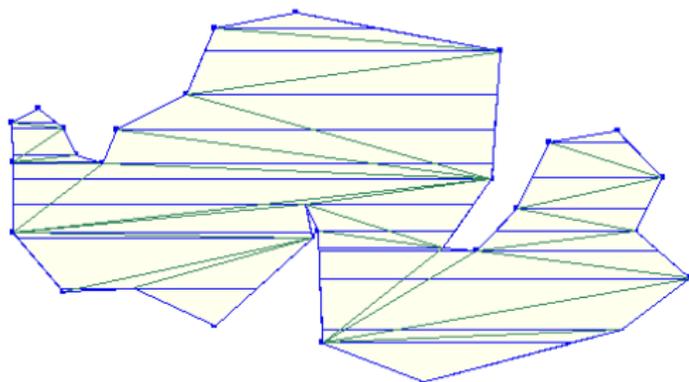
Elmar Langetepe

Universität Bonn, Institut für Informatik

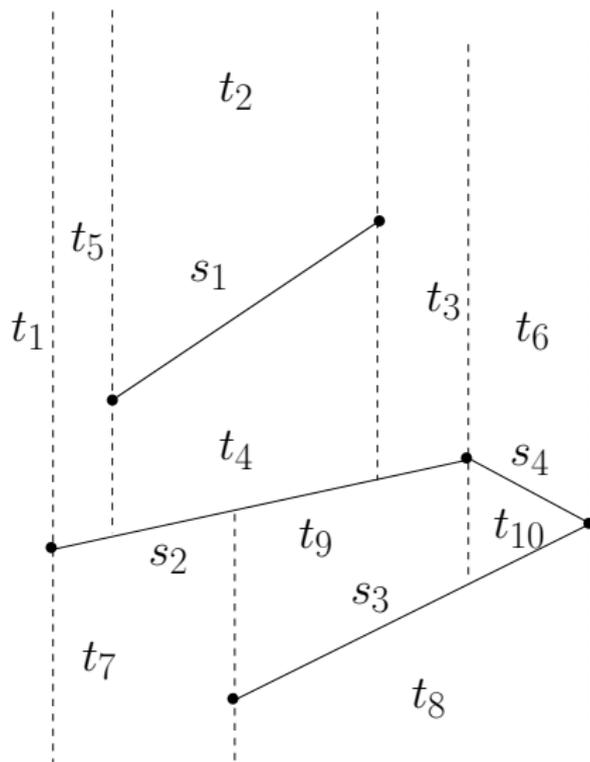
1.2.2017

Lokalisation, Datenstruktur: Trapezzerlegung nach Seidel

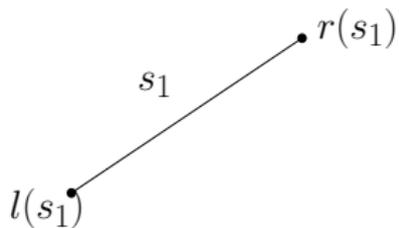
- Lokalisation Black Box
- Polygon: Dreieckszerlegung
- Anfrage: Für $p \in P$ finde Dreieck T mit $p \in T$
- Kürzeste Wege in Polygonen/auf Polyeder
- Aufbau: $O(n \log^* n)$, Query: $O(\log n)$
- Zerlegung in Trapeze, monotone Polygone, danach in Dreiecke
- Hier Trapezzerlegung!



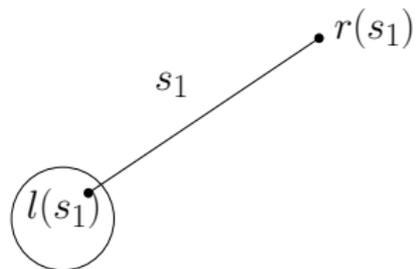
Trapezzerlegung nach Seidel



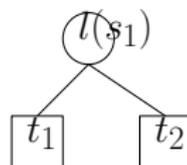
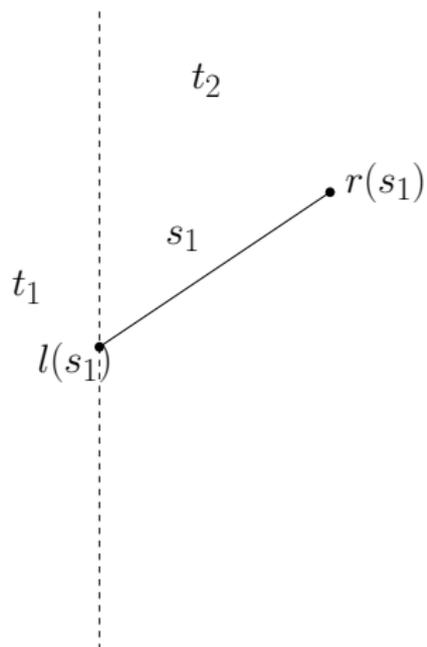
Datenstruktur: Trapezzerlegung nach Seidel



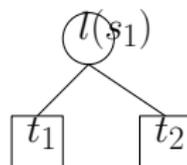
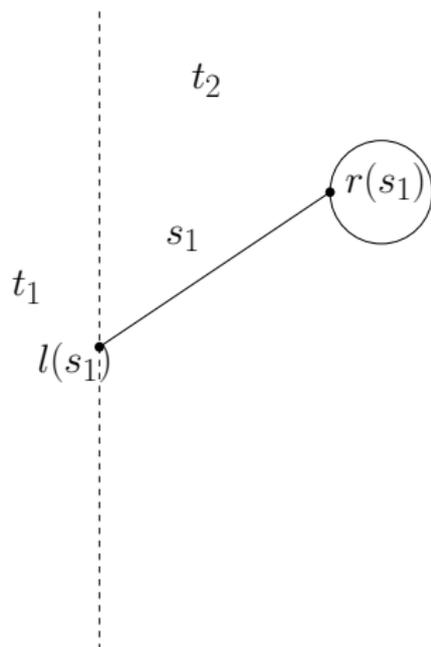
Datenstruktur: Trapezzerlegung nach Seidel



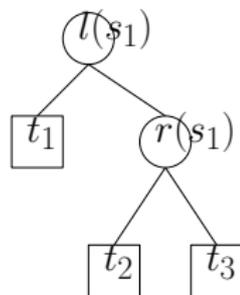
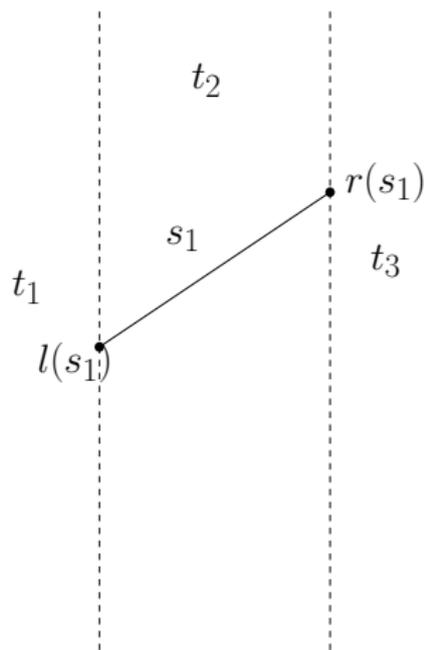
Datenstruktur: Trapezzerlegung nach Seidel



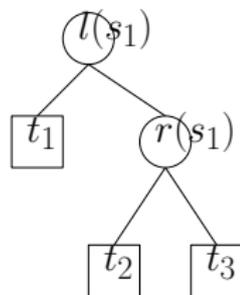
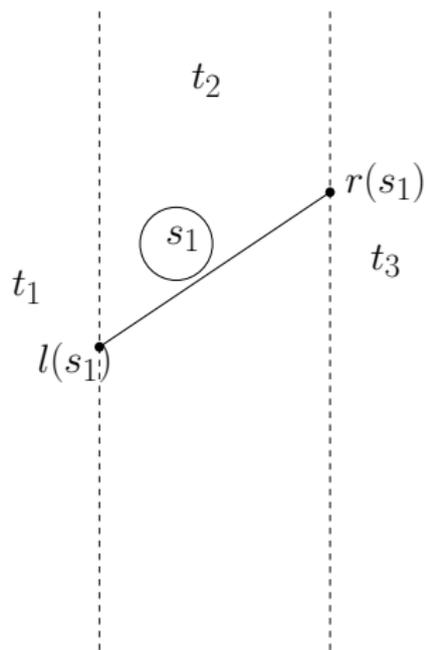
Datenstruktur: Trapezzerlegung nach Seidel



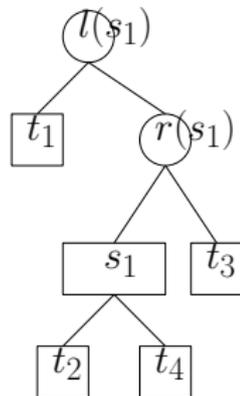
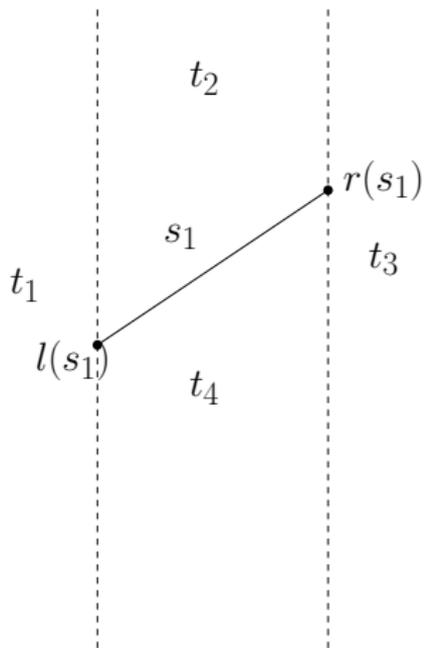
Datenstruktur: Trapezzerlegung nach Seidel



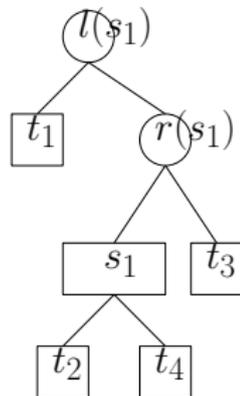
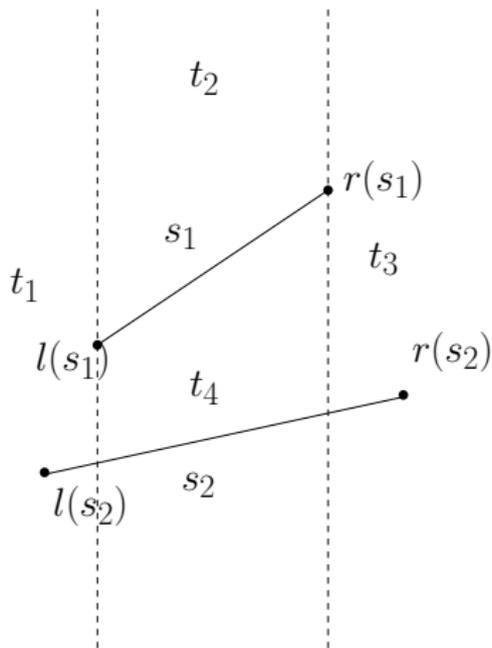
Datenstruktur: Trapezzerlegung nach Seidel



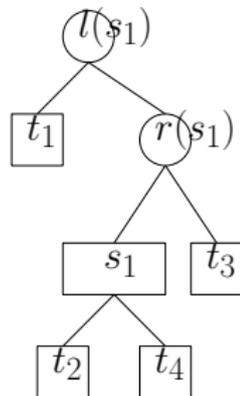
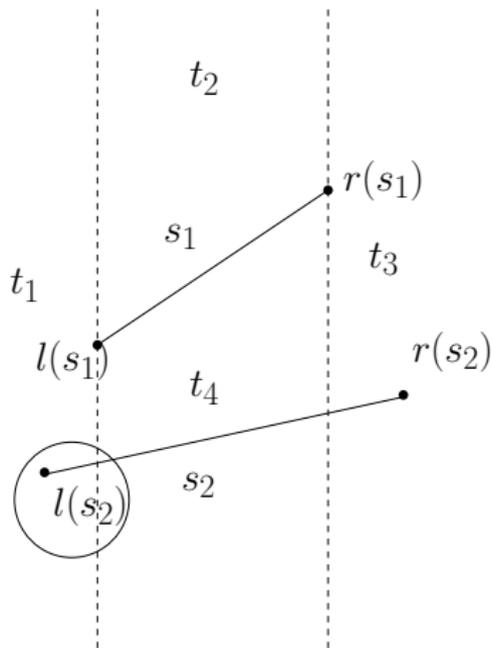
Datenstruktur: Trapezzerlegung nach Seidel



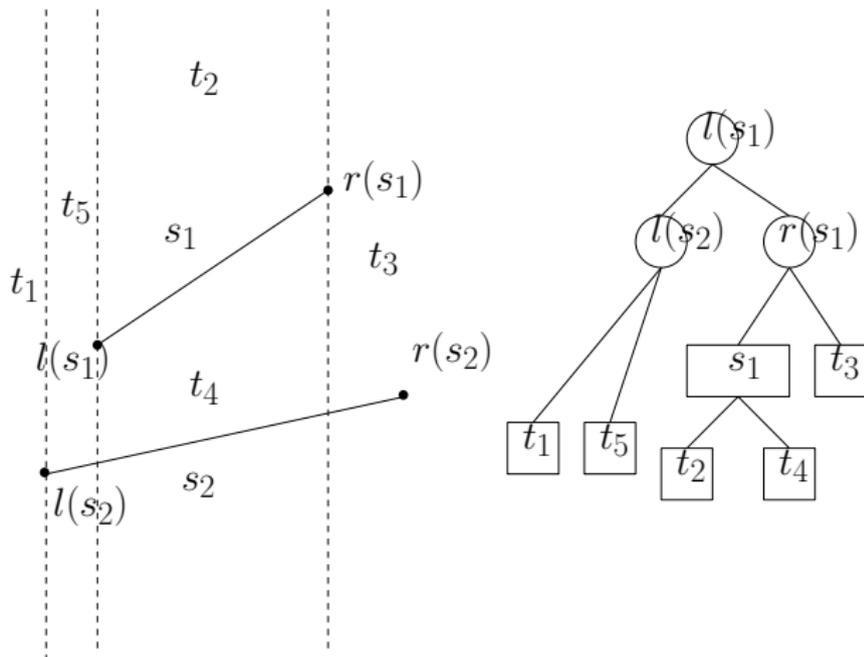
Datenstruktur: Trapezzerlegung nach Seidel



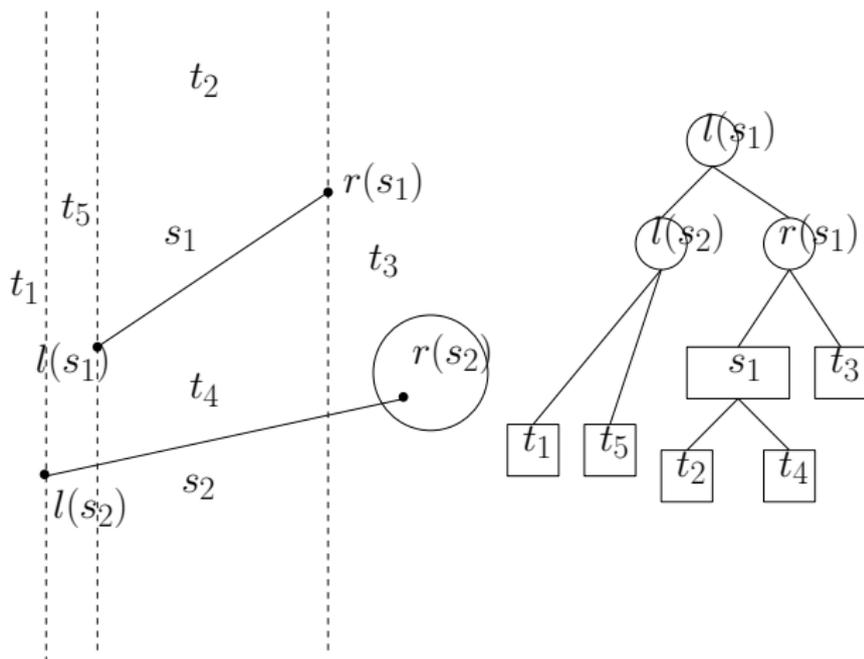
Datenstruktur: Trapezzerlegung nach Seidel



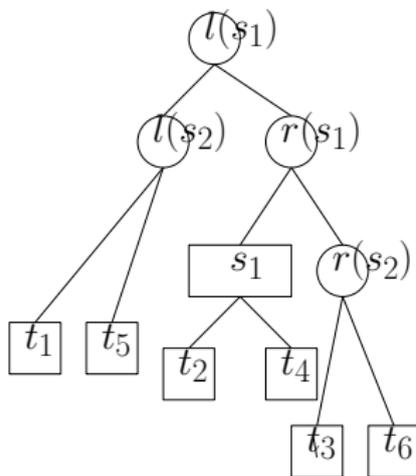
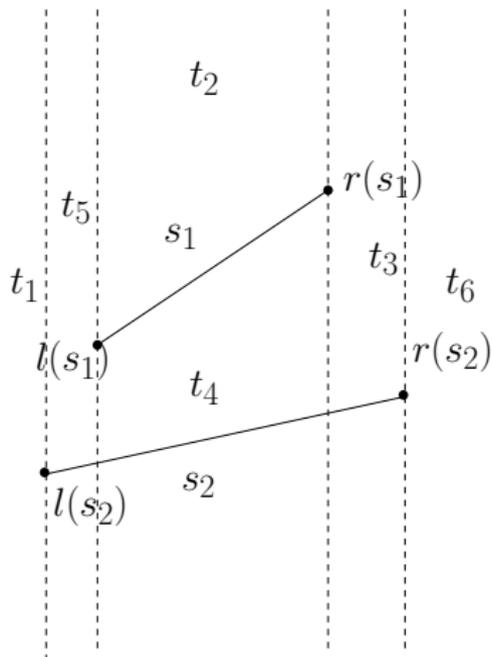
Datenstruktur: Trapezzerlegung nach Seidel



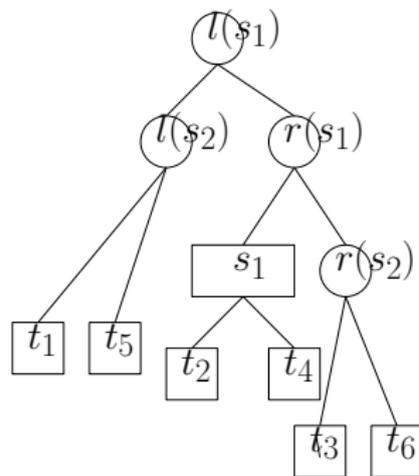
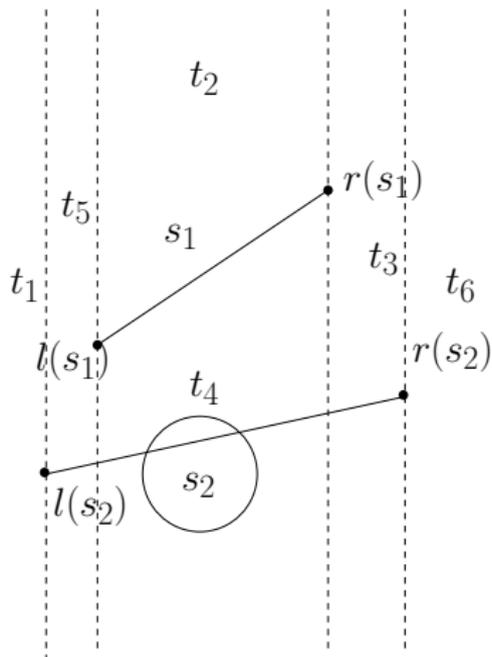
Datenstruktur: Trapezzerlegung nach Seidel



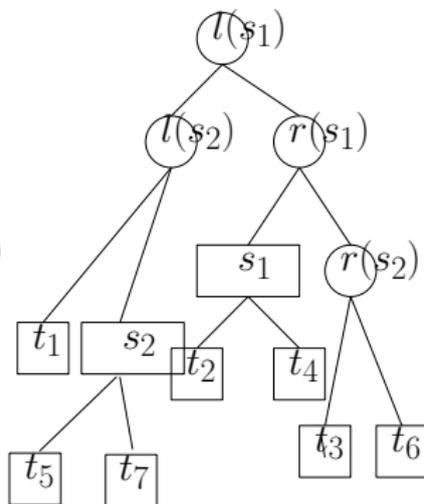
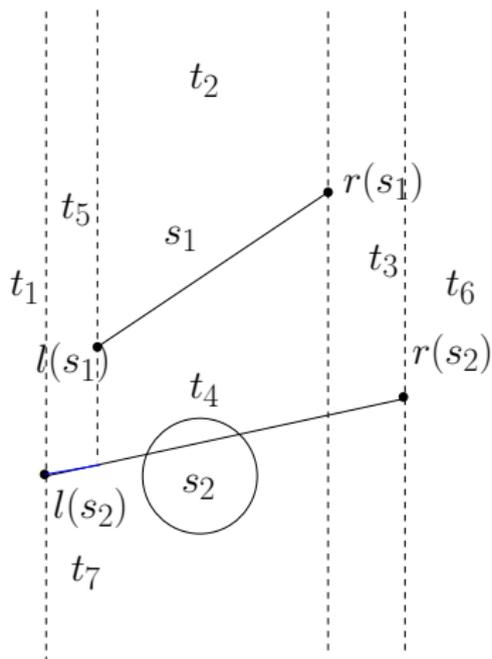
Datenstruktur: Trapezzerlegung nach Seidel



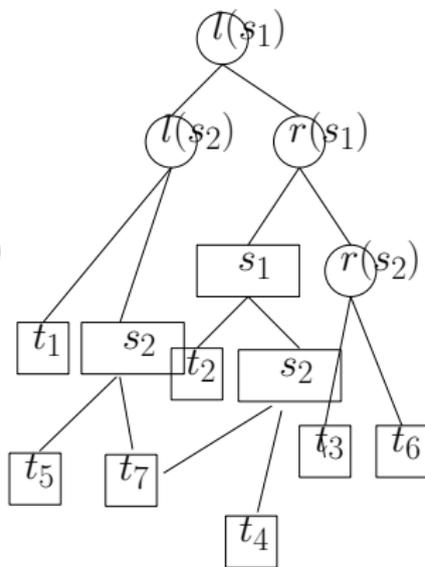
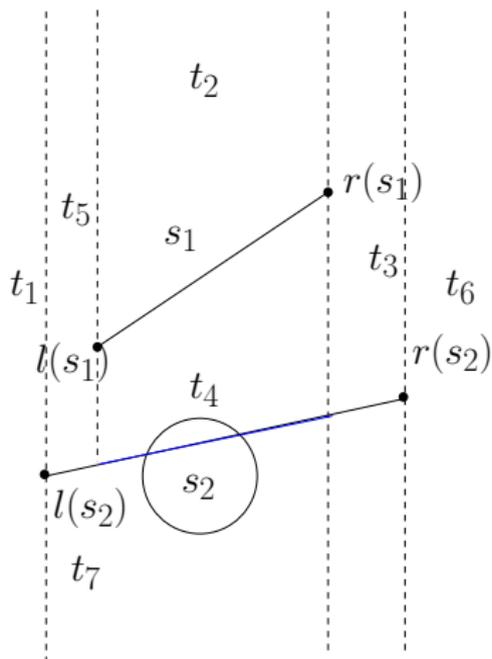
Datenstruktur: Trapezzerlegung nach Seidel



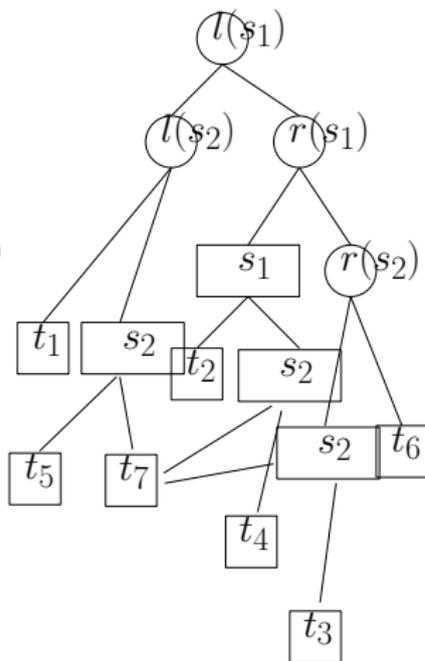
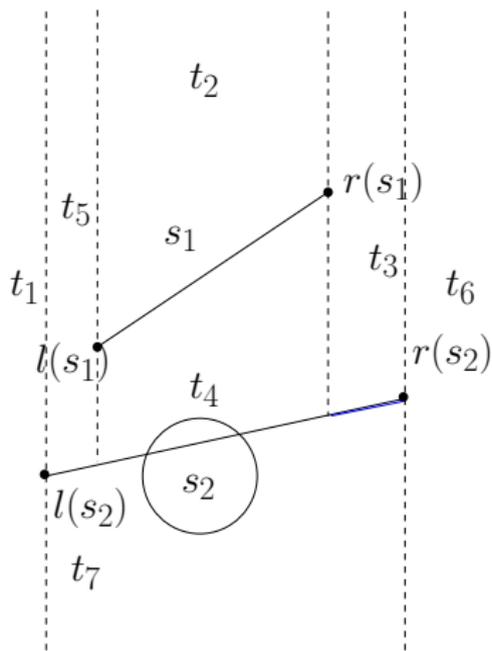
Datenstruktur: Trapezzerlegung nach Seidel



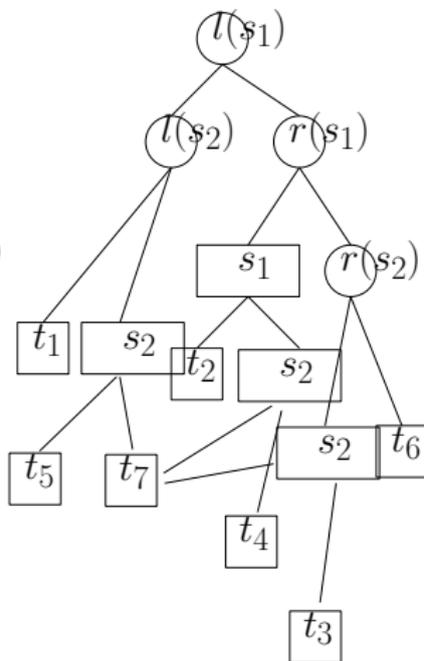
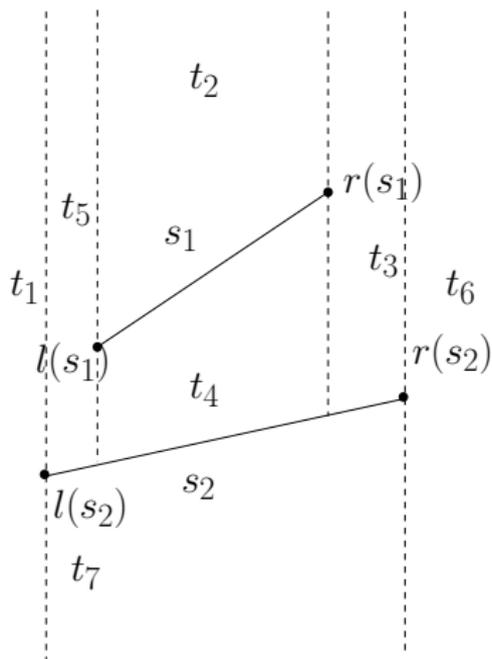
Datenstruktur: Trapezzerlegung nach Seidel



Datenstruktur: Trapezzerlegung nach Seidel



Datenstruktur: Trapezzerlegung nach Seidel



Vorgehensweise:

- Linken Endpunkt lokalisieren
- Rechten Endpunkt lokalisieren
- Strecke s_i durch Trapeze verfolgen
- Trapeze abspeichern mit Nachbarn und Links! $O(1)$
- Wesentlicher Aufwand: Lokalisation
- Query: Lokalisation eines Punktes
- Beweisskizze: Randomisierung! Einfache Analyse!

Datenstruktur: Trapezzerlegung nach Seidel

- Segmente s_1, s_2, \dots, s_n zufällig!
- $P_i = \{s_1, \dots, s_i\}$
- $D(P_i)$ Datenstruktur nach sukzessiven Einfügen
- $T(P_i)$ (eindeutige) Trapezzerlegung
- **Lokalisation der Endpunkte**
- Tracen der Segmente, insgesamt $O(n)$ wg. Komplexität

Lemma: Die erwartete Laufzeit bei zufälliger Einfügereihenfolge der Segmente zur Erstellung der Datenstruktur $D(P_n)$ liegt in $O(n \log n)$.

Beweis, Verbesserung durch Phasen, Tracing der polygonalen Kette!

Randomisierte Analyse: Lokalisation der Endpunkte

- Query Punkt q : Sei T_k Trapez von $T(P_k)$, das q enthält
- Betrachte Übergang von $D(P_{j-1})$ nach $D(P_j)$
- Annahme: q liegt in T_{j-1}
- Verändert sich T_{j-1} nicht, gilt $T_{j-1} = T_j$.
Hier ist kein weiterer Vergleich an dieser Stelle nötig
- Gilt $T_{j-1} \neq T_j$, hat T_j am Rand eines der vier Bestandteile von s_j .
- Jedes Segment hatte die Wahrscheinlichkeit $\frac{4}{j}$, einen Bestandteil zu liefern
- Wahrscheinlichkeit eines weiteren Vergleichs liegt in $O\left(\frac{1}{j}\right)$
- Erwartete Anzahl Vergleiche: $C \sum_{i=1}^n \frac{1}{i} \in O(\log n)$

Datenstruktur: Trapezzerlegung nach Seidel

Idee: Schnellere Lokalisation: Polygonale Kette in Phasen in akt. DS lokalisieren.

Theorem: Die erwartete Laufzeit bei zufälliger Einfügereihenfolge der Segmente mit gelegentlicher Lokalisation der polygonalen Kette zur Erstellung der Datenstruktur $D(P_n)$ liegt in $O(n \log^* n)$. Die Zeit pro weiterer Anfrage liegt in $O(\log n)$.

- Shortest Watchman Route, achsenparallel, andere Zielfunktion!
- VC-Dimension: Art-Gallery Guarding, monoton, achsenparallel
- Feuerbekämpfung, Eindämmung, diskret, kontinuierlich!
- ...

Bewegungsplanung: Feuerbekämpfung

- Wichtiges Problem:
Schlagen von Feuerschneisen
- Vorhandene Schneisen
Neue Schneisen
- Einfaches Modell:
Gleichmäßige Ausbreitung
- Geschwindigkeit: $s \in [0, 1)$



Bewegungsplanung: Feuerbekämpfung

- Wichtiges Problem:
Schlagen von Feuerschneisen
- Vorhandene Schneisen
Neue Schneisen
- Einfaches Modell:
Gleichmäßige Ausbreitung
- Geschwindigkeit: $s \in [0, 1)$



Bewegungsplanung: Feuerbekämpfung

- Wichtiges Problem:
Schlagen von Feuerschneisen
- Vorhandene Schneisen
Neue Schneisen
- Einfaches Modell:
Gleichmäßige Ausbreitung
- Geschwindigkeit: $s \in [0, 1)$



- $s \approx 0.38 \in [0, 1)$
- $s = \cos(\alpha),$
 $\alpha \in (0, \pi/2]$

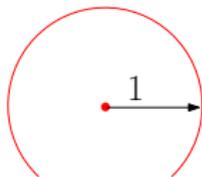
•

Bewegungsplanung: Feuerbekämpfung

- Wichtiges Problem:
Schlagen von Feuerschneisen
- Vorhandene Schneisen
Neue Schneisen
- Einfaches Modell:
Gleichmäßige Ausbreitung
- Geschwindigkeit: $s \in [0, 1)$



- $s \approx 0.38 \in [0, 1)$
- $s = \cos(\alpha)$,
 $\alpha \in (0, \pi/2]$

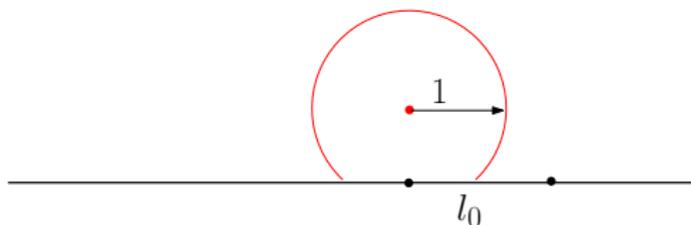


Bewegungsplanung: Feuerbekämpfung

- Wichtiges Problem:
Schlagen von Feuerschneisen
- Vorhandene Schneisen
Neue Schneisen
- Einfaches Modell:
Gleichmäßige Ausbreitung
- Geschwindigkeit: $s \in [0, 1)$



- $s \approx 0.38 \in [0, 1)$
- $s = \cos(\alpha)$,
 $\alpha \in (0, \pi/2]$

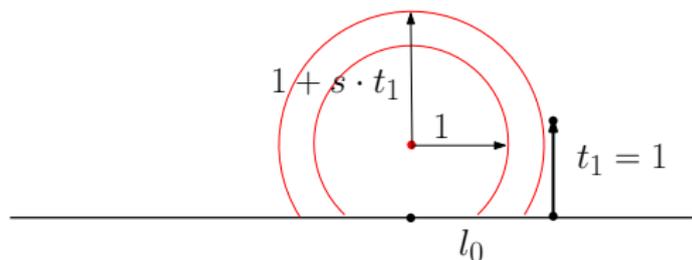


Bewegungsplanung: Feuerbekämpfung

- Wichtiges Problem:
Schlagen von Feuerschneisen
- Vorhandene Schneisen
Neue Schneisen
- Einfaches Modell:
Gleichmäßige Ausbreitung
- Geschwindigkeit: $s \in [0, 1)$



- $s \approx 0.38 \in [0, 1)$
- $s = \cos(\alpha)$,
 $\alpha \in (0, \pi/2]$

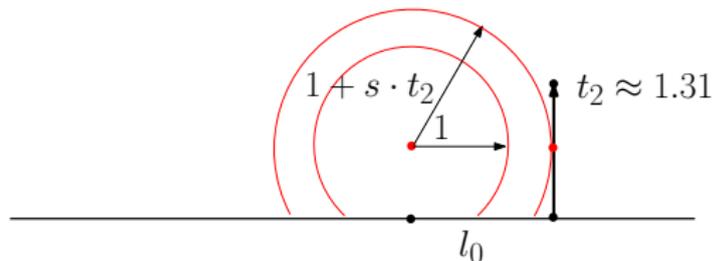


Bewegungsplanung: Feuerbekämpfung

- Wichtiges Problem:
Schlagen von Feuerschneisen
- Vorhandene Schneisen
Neue Schneisen
- Einfaches Modell:
Gleichmäßige Ausbreitung
- Geschwindigkeit: $s \in [0, 1)$



- $s \approx 0.38 \in [0, 1)$
- $s = \cos(\alpha)$,
 $\alpha \in (0, \pi/2]$



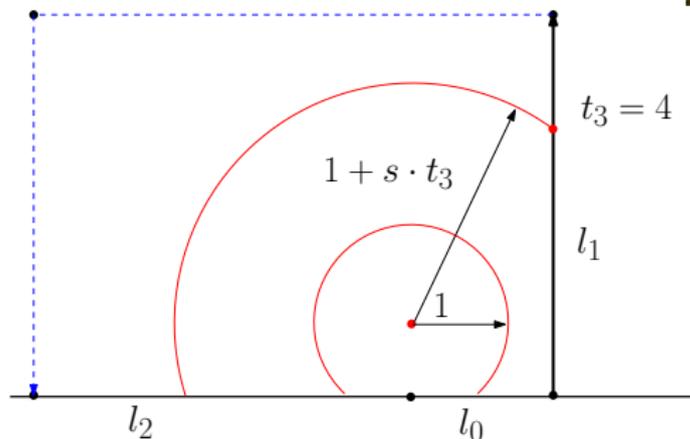
Bewegungsplanung: Feuerbekämpfung

- Wichtiges Problem:
Schlagen von Feuerschneisen
- Vorhandene Schneisen
Neue Schneisen
- Einfaches Modell:
Gleichmäßige Ausbreitung
- Geschwindigkeit: $s \in [0, 1)$



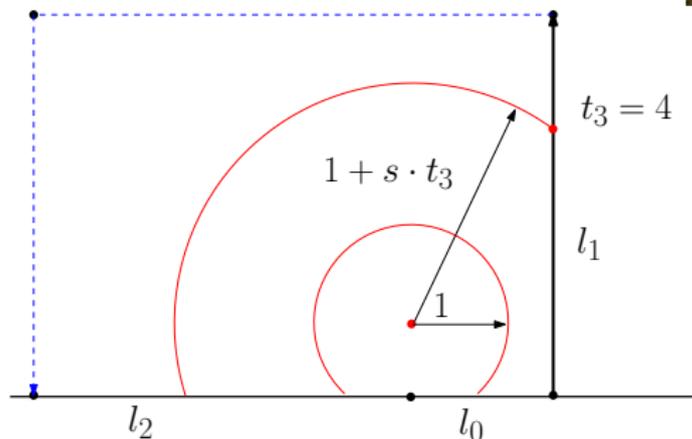
- $s \approx 0.38 \in [0, 1)$

- $s = \cos(\alpha),$
 $\alpha \in (0, \pi/2]$



Bewegungsplanung: Feuerbekämpfung

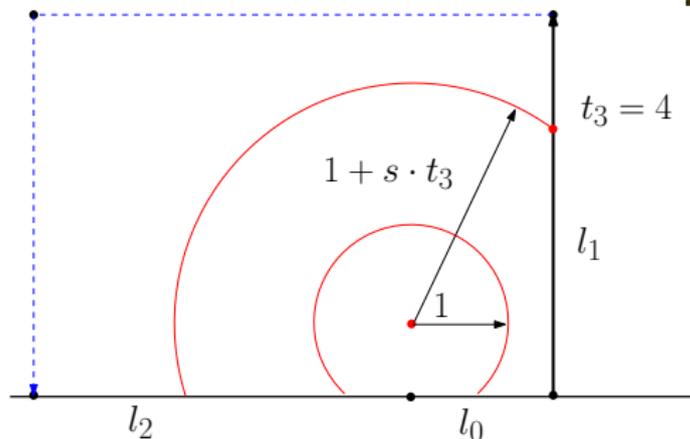
- Wichtiges Problem:
Schlagen von Feuerschneisen
- Vorhandene Schneisen
Neue Schneisen
- Einfaches Modell:
Gleichmäßige Ausbreitung
- Geschwindigkeit: $s \in [0, 1)$



- $s \approx 0.38 \in [0, 1)$
- $s = \cos(\alpha)$,
 $\alpha \in (0, \pi/2]$
- Feuer schnell
umschließen
(Zeitminimierung!)

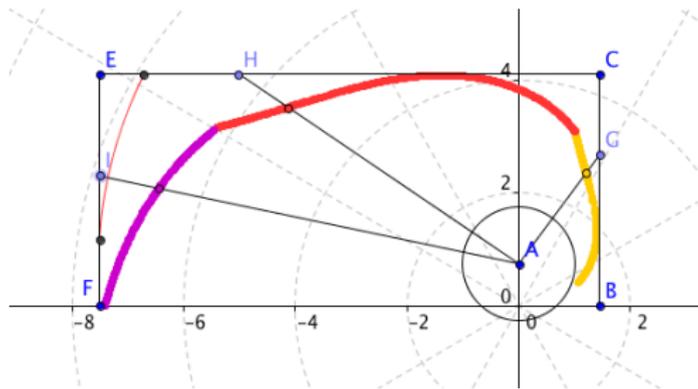
Bewegungsplanung: Feuerbekämpfung

- Wichtiges Problem:
Schlagen von Feuerschneisen
- Vorhandene Schneisen
Neue Schneisen
- Einfaches Modell:
Gleichmäßige Ausbreitung
- Geschwindigkeit: $s \in [0, 1)$



- $s \approx 0.38 \in [0, 1)$
- $s = \cos(\alpha)$,
 $\alpha \in (0, \pi/2]$
- Feuer schnell
umschließen
(Zeitminimierung!)
- l_0, l_1, l_2 gefahrlos
minimieren!

Bewegungsplanung: Feuerbekämpfung



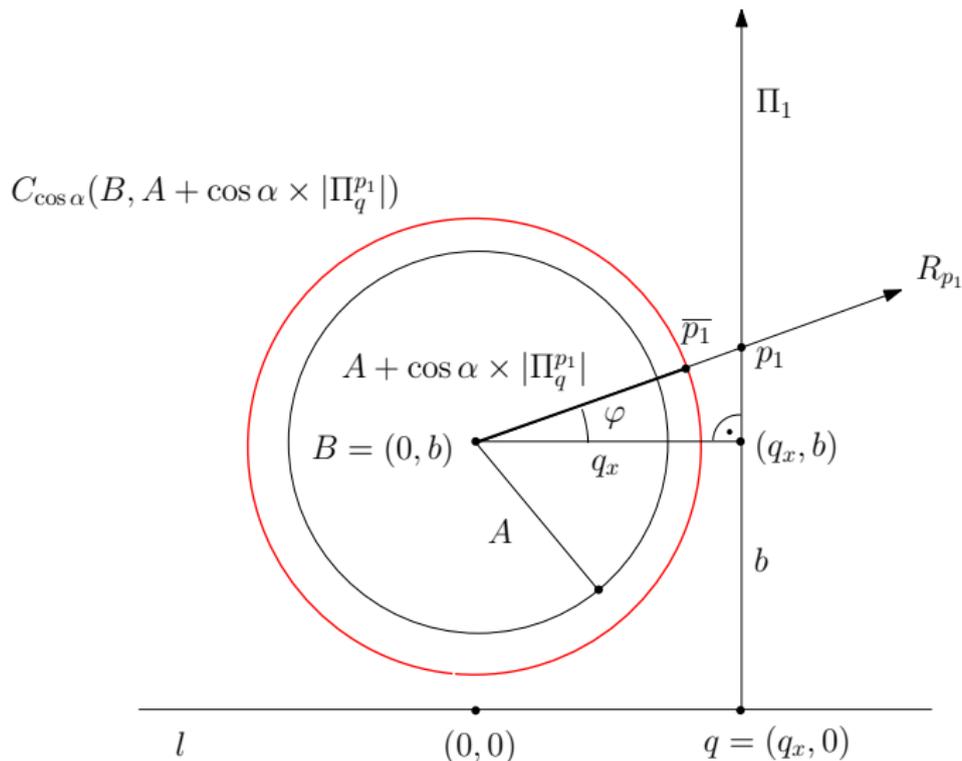
- $s \approx 0.38 = \cos(1.18)$
Optimale Lösung!
- Minimieren der Werte
- *Passieren* des Feuers
- Zeit *und* Fläche!

Theorem: There exists a unique area and completion time optimal axis-parallel firebreak for all $\alpha \in (\pi/4, \pi/2]$. For $\alpha \in (0, \pi/4]$ no solution exists.

Beweis!

Bewegungsplanung: Feuerbekämpfung

Situation erster Teil!

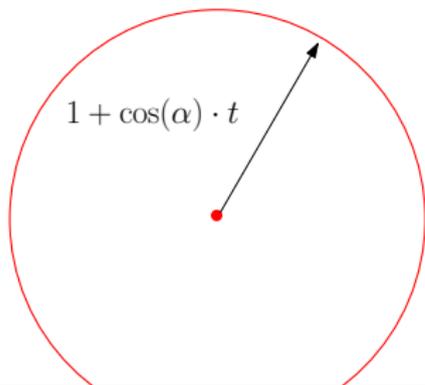


Bewegungsplanung: Feuerbekämpfung

- Für $\alpha \in (0, \pi/4]$ keine Lösung!
- Geht das, wenn die Firefigther beliebig graben dürfen?
- Idee: Starte nah am Feuer und bleibe dran!
- Geschwindigkeit $\cos(\alpha)$, $\alpha \in (0, \pi/2]$
- Kontinuierlich lokal in Richtung α !

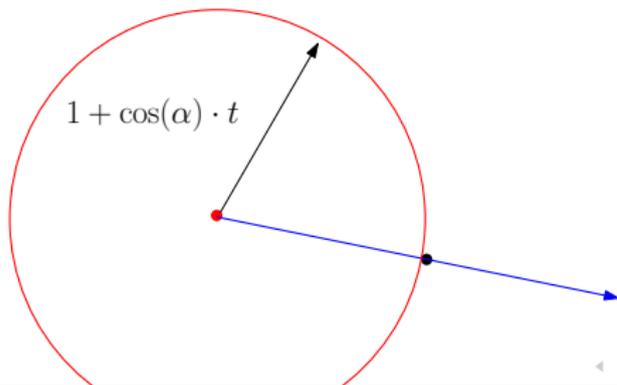
Bewegungsplanung: Feuerbekämpfung

- Für $\alpha \in (0, \pi/4]$ keine Lösung!
- Geht das, wenn die Firefigther beliebig graben dürfen?
- Idee: Starte nah am Feuer und bleibe dran!
- Geschwindigkeit $\cos(\alpha)$, $\alpha \in (0, \pi/2]$
- Kontinuierlich lokal in Richtung α !



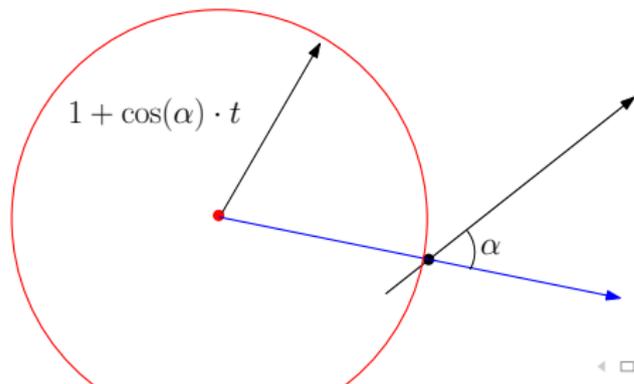
Bewegungsplanung: Feuerbekämpfung

- Für $\alpha \in (0, \pi/4]$ keine Lösung!
- Geht das, wenn die Firefigther beliebig graben dürfen?
- Idee: Starte nah am Feuer und bleibe dran!
- Geschwindigkeit $\cos(\alpha)$, $\alpha \in (0, \pi/2]$
- Kontinuierlich lokal in Richtung α !



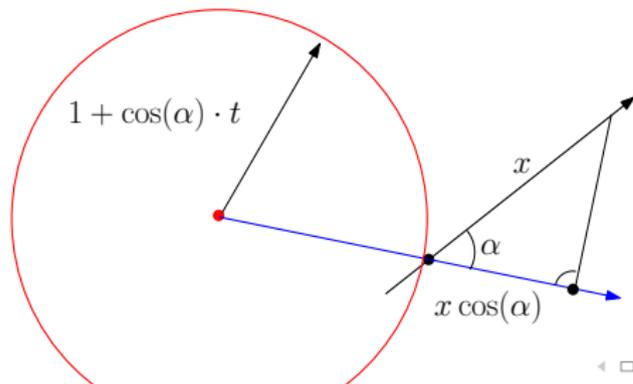
Bewegungsplanung: Feuerbekämpfung

- Für $\alpha \in (0, \pi/4]$ keine Lösung!
- Geht das, wenn die Firefigther beliebig graben dürfen?
- Idee: Starte nah am Feuer und bleibe dran!
- Geschwindigkeit $\cos(\alpha)$, $\alpha \in (0, \pi/2]$
- Kontinuierlich lokal in Richtung α !



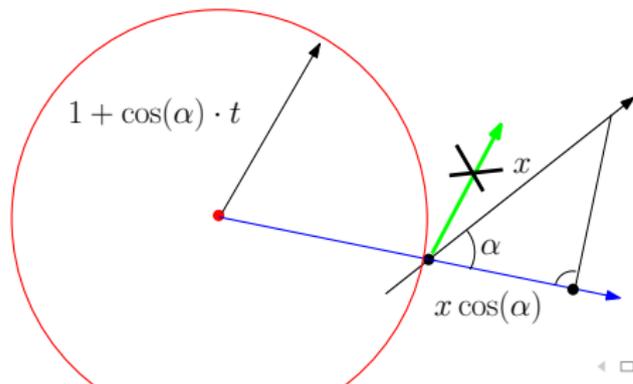
Bewegungsplanung: Feuerbekämpfung

- Für $\alpha \in (0, \pi/4]$ keine Lösung!
- Geht das, wenn die Firefigther beliebig graben dürfen?
- Idee: Starte nah am Feuer und bleibe dran!
- Geschwindigkeit $\cos(\alpha)$, $\alpha \in (0, \pi/2]$
- Kontinuierlich lokal in Richtung α !



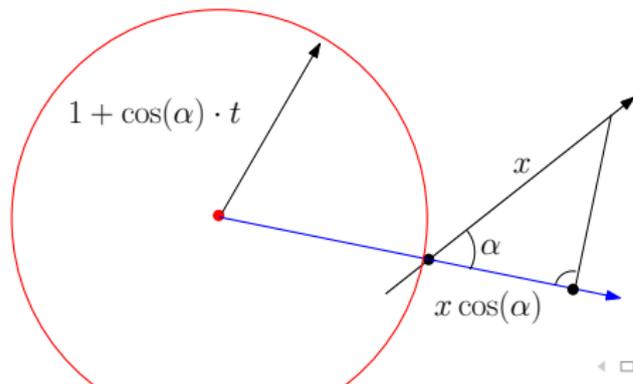
Bewegungsplanung: Feuerbekämpfung

- Für $\alpha \in (0, \pi/4]$ keine Lösung!
- Geht das, wenn die Firefigther beliebig graben dürfen?
- Idee: Starte nah am Feuer und bleibe dran!
- Geschwindigkeit $\cos(\alpha)$, $\alpha \in (0, \pi/2]$
- Kontinuierlich lokal in Richtung α !



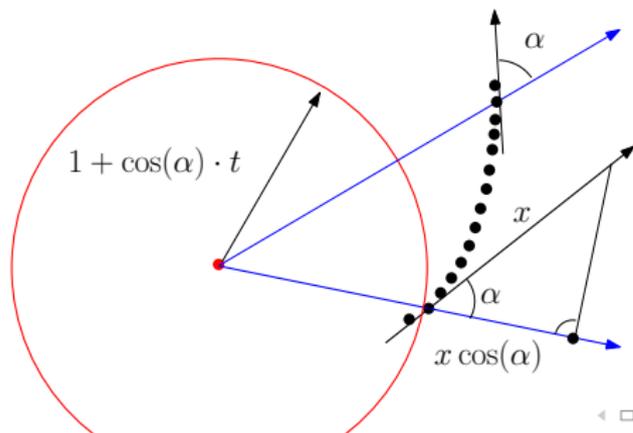
Bewegungsplanung: Feuerbekämpfung

- Für $\alpha \in (0, \pi/4]$ keine Lösung!
- Geht das, wenn die Firefigther beliebig graben dürfen?
- Idee: Starte nah am Feuer und bleibe dran!
- Geschwindigkeit $\cos(\alpha)$, $\alpha \in (0, \pi/2]$
- Kontinuierlich lokal in Richtung α !



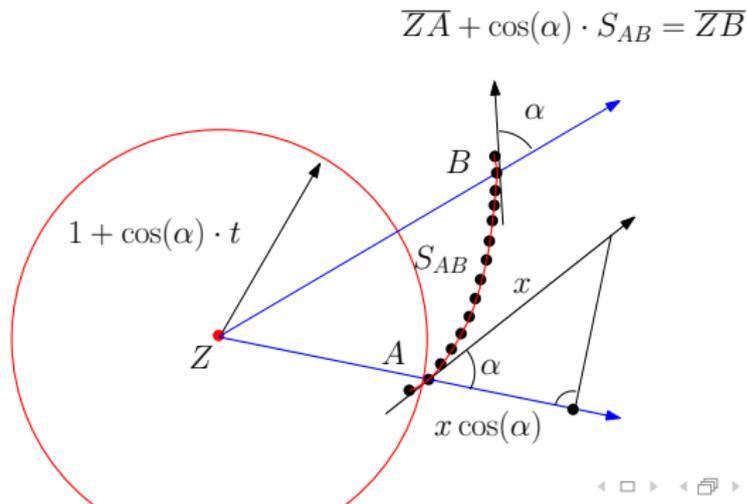
Bewegungsplanung: Feuerbekämpfung

- Für $\alpha \in (0, \pi/4]$ keine Lösung!
- Geht das, wenn die Firefigther beliebig graben dürfen?
- Idee: Starte nah am Feuer und bleibe dran!
- Geschwindigkeit $\cos(\alpha)$, $\alpha \in (0, \pi/2]$
- Kontinuierlich lokal in Richtung α !



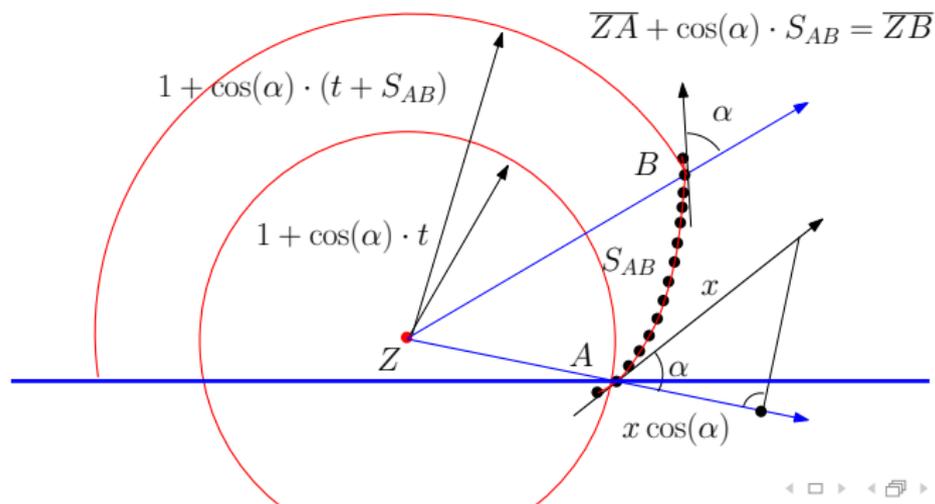
Bewegungsplanung: Feuerbekämpfung

- Für $\alpha \in (0, \pi/4]$ keine Lösung!
- Geht das, wenn die Firefigther beliebig graben dürfen?
- Idee: Starte nah am Feuer und bleibe dran!
- Geschwindigkeit $\cos(\alpha)$, $\alpha \in (0, \pi/2]$
- Kontinuierlich lokal in Richtung α !



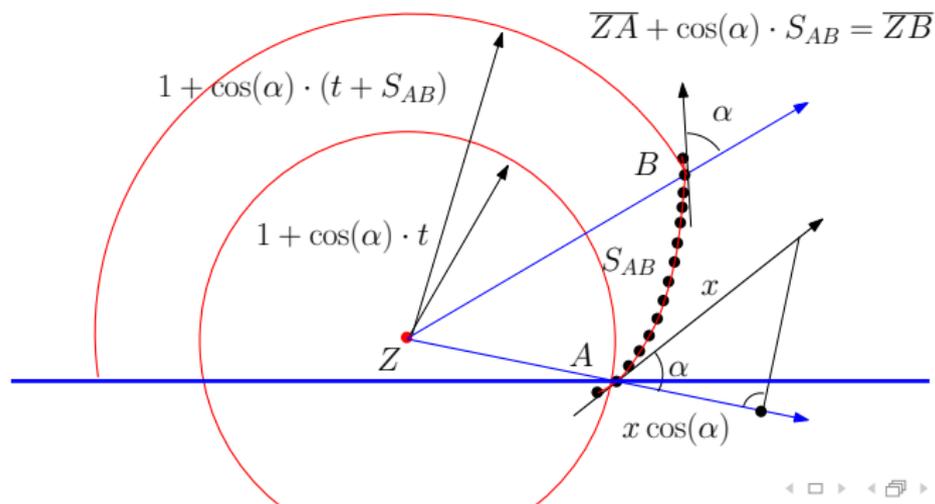
Bewegungsplanung: Feuerbekämpfung

- Für $\alpha \in (0, \pi/4]$ keine Lösung!
- Geht das, wenn die Firefigther beliebig graben dürfen?
- Idee: Starte nah am Feuer und bleibe dran!
- Geschwindigkeit $\cos(\alpha)$, $\alpha \in (0, \pi/2]$
- Kontinuierlich lokal in Richtung α !



Bewegungsplanung: Feuerbekämpfung

- Für $\alpha \in (0, \pi/4]$ keine Lösung!
- Geht das, wenn die Firefigther beliebig graben dürfen?
- Idee: Starte nah am Feuer und bleibe dran!
- Geschwindigkeit $\cos(\alpha)$, $\alpha \in (0, \pi/2]$
- Kontinuierlich lokal in Richtung α !
- Weg: Logarithmische Spirale mit Exzentrizität α !

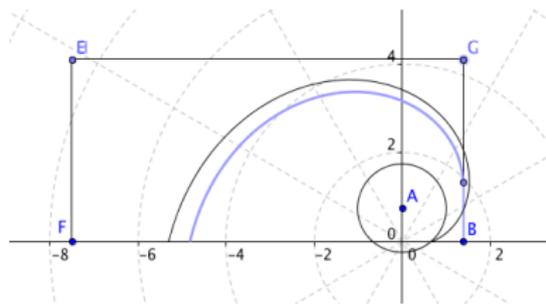


Bewegungsplanung: Feuerbekämpfung

- Mit Spiralen eine bessere/optimale Lösung erstellen!
- Verschiedene Kandidaten!
- Optimale Lösung: Zeitminimierung für jedes α !!!

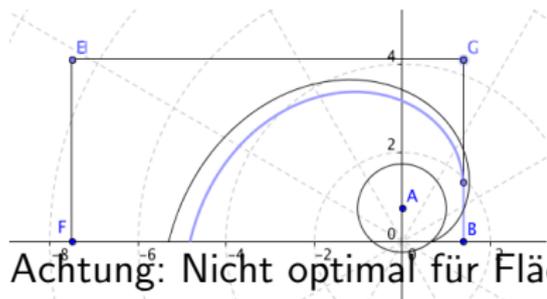
Bewegungsplanung: Feuerbekämpfung

- Mit Spiralen eine bessere/optimale Lösung erstellen!
- Verschiedene Kandidaten!
- Optimale Lösung: Zeitminimierung für jedes α !!!
- Begründung! (Beweis!) Gilt für alle $\alpha \in (0, \pi/2]$



Bewegungsplanung: Feuerbekämpfung

- Mit Spiralen eine bessere/optimale Lösung erstellen!
- Verschiedene Kandidaten!
- Optimale Lösung: Zeitminimierung für jedes α !!!
- Begründung! (Beweis!) Gilt für alle $\alpha \in (0, \pi/2]$



Bewegungsplanung: Feuerbekämpfung

- Waldbrände, Grundsätze
- Ausbreitungsgeschwindigkeit klein: 0.5 km/h
- Graben mit Winkel
- Veränderte Ausbreitungskurven
- Modellerweiterungen



Mögliche Aufgabenstellungen

- Kontinuierliches Modell
- Beschränkung der verbrannten Fläche
- Wege mit ein/zwei Agenten, Wege mit wenig Knicken
- Diskretes Modell
- Optimale Vorabschneise plus Bekämpfung
- Grid-Graphen oder allgemeine Graphen