Fakultät für Mathematik und Informatik
Lehrgebiet Kooperative Systeme

Diplomarbeit

Untere Schranken für das Erkundungsproblem

eingereicht am: 4.7.2019
von: Christian Edgar Lomp
geboren am 7. November 1969
in Düsseldorf
Matrikel-Nr. 4134850
Studiengang Informatik (Diplom II)

Betreuer: Prof Dr. Christian Icking

Erklärung

Name:	Christian Lomp
Matrikel-Nr.:	4134850
Fach:	Informatik
Modul:	Diplomarbeit (Diplom II)

Ich erkläre, dass ich die vorliegende Diplomarbeit mit dem Thema:

Untere Schranken für das Erkundungsproblem

selbständig und ohne unzulässige Inanspruchnahme Dritter verfasst habe. Ich habe dabei nur die angegebenen Quellen und Hilfsmittel verwendet und die aus diesen wörtlich, inhaltlich oder sinngemäß entnommenen Stellen als solche den wissenschaftlichen Anforderungen entsprechend kenntlich gemacht. Die Versicherung selbständiger Arbeit gilt auch für Zeichnungen, Skizzen oder graphische Darstellungen. Die Arbeit wurde bisher in gleicher oder ähnlicher Form weder derselben noch einer anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht. Mit der Abgabe der elektronischen Fassung der endgültigen Version der Arbeit nehme ich zur Kenntnis, dass diese mit Hilfe eines Plagiatserkennungsdienstes auf enthaltene Plagiate überprüft und ausschließlich für Prüfungszwecke gespeichert wird.

Datum: _____ **Unterschrift:** _____

Inhaltsverzeichnis

1 Einleitung **1**

2 Polygone mit einer Schwelle **6**

3 Polygone mit zwei Schwellen **11**

3.1 Schwellen mit uneingeschränkter Sicht 15

3.2 Schwellen ohne versteckte Taschen 23

3.3 Schwellen mit eingeschränkter Sicht 30

4 Polygone mit drei Schwellen **33**

4.1 Die Grundidee 33

4.2 Numerische Bestätigung der unteren Schranke 1.287 41

4.3 Leichte Verbesserung der unteren Schranke auf 1.2945 43

5 Schlußbemerkungen **47**

Literatur **48**

1 Einleitung

Das Kunstgalerieproblem

Beim Kunstgalerieproblem geht es um die Aufstellung einer minimalen Anzahl von Wächtern in einer Kunstgalerie, so dass jeder Ort der Galerie von mindestens einem der Wächter gesehen werden kann. In der algorithmischen Geometrie wird dieses Problem dadurch modelliert, dass man nach der minimalen Anzahl von Punkten (der Wächter) in einem einfachen, ebenen Polygon (der Galerie) sucht, so dass jeder Punkt des Inneren dieses Polygons von mindestens einem dieser ausgewählten Punkte "gesehen" werden kann, d.h. dass das Verbindungssegment zwischen beiden Punkten ganz im Inneren des Polygons oder auf dem Polygon selbst verläuft. Dieses Problem geht nach Václav Chvátal auf eine mündliche Frage Victor Klees zurück. Durch Triangulierung und Einfärbung der Triangulierungsecken kann gezeigt werden dass die obere Schranke der minimalen Anzahl von Wächtern durch $\lfloor n/3 \rfloor$ gegeben ist, wobei n die Anzahl der Ecken des Polygons ist (siehe [6, 10, 13, 15]).

Das Problem der optimalen Wächterroute

Im Gegensatz zum Kunstgalerieproblem, haben Chin und Ntafos in [5] vorgeschlagen, eine kürzeste Wächterroute zu finden, die beispielsweise ein Nachtwächter abgehen kann, um alle Orte der Galerie mindestens einmal gesehen zu haben. Damit ist nun ausgehend von einem Startpunkt ein geschlossener Weg gesucht, der nicht das Äußere des Polygons durchschneidet und von dem aus jeder innere Punkt des Polygons mindestens einmal gesehen werden kann. Durch Rückführung auf das Problem des Handlungsreisenden zeigen Chin und Ntafos, dass das Problem eine minimale Wächterroute zu finden NP-hart ist, falls das Polygon bekannt ist und "Löcher"hat, d.h. falls es nicht zusammenziehbar ist. Wenig später haben Tan und Hirata in [18] einen $O(n^2)$-Algorithmus angegeben, um eine minimale Wächterroute in einem bekannten zusammenziehbaren Polygon zu finden.

Das Erkundungsproblem

Im Fall eines unbekannten Polygons, sprechen wir von der Erkundung des Polygons und somit vom *Erkundungsproblem*. Ein Algorithmus, der das Erkundungsproblem löst, ist daher ein *online* - Algorithmus, also ein Programm, welches nur unvollständiges Wissen über das Polygon hat und dieses im Laufe seiner Ausführung erweitert. Im Gegensatz dazu nennt man einen Algorithmus, der vollständiges Wissen über das Polygon hat, einen *offline* - Algorithmus.

Es gibt sicherlich viele anwendungsbezogene Probleme, die als Erkundungsproblem interpretiert werden können, wie zum Beispiel die Erkundung eines ebenen Gebietes durch

einen autonomen mobilen Roboter mit ausgestattetem 360°-Sichtsystem. Dabei könnte es sich sowohl um eine Rettungsaktion in einem Katastrophengebiet, als auch um die Erforschung auf einem fremden Planeten handeln. Nach Erkunden des Gebietes würde der Roboter wieder an seinen Anfangspunkt zurückkehren. Verschiedene Einschränkungen werden wir in dieser Arbeit ausklammern, wie zum Beispiel die beschränkte Autonomie eines Roboters, der ansonsten in regelmäßigen Abständen zur Basis zurückkehren müsste (siehe [1, 3]). Oder Einschränkungen der 360° Sicht, die einen Roboter zwingen würden von Zeit zu Zeit anzuhalten, um eine Aufnahme zu machen (siehe [9]), wie auch Einschränkungen bei Richtungsänderungen, die gegebenenfalls zeitaufwendig wären (siehe [2]).

Die kompetitive Analyse

Die Güte eines *online*-Algorithmus bestimmen wir durch die *kompetitive Analyse*. Diese Idee geht auf eine Arbeit von Sleator und Tarjan zurück (siehe [17]), wobei die Laufzeitkosten eines *online* - Algorithmus und der eines optimalen *offline*-Algorithmus verglichen werden (siehe auch [4]):

Definition 1 (Sleator und Tarjan (siehe auch [13, Seite 330]))**.** *Sei* Π *ein Menge von Problemen und* S *eine Strategie, die jedes Problem* $P \in \Pi$ *korrekt löst. Seien ferner* $K_S(P)$ *die Kosten die* S *verursacht zur Lösung von Problem* P *und* $K_{opt}(P)$ *die Kosten einer optimalen Lösung. Man sagt, dass* S *kompetitiv mit Faktor* C *oder kurz* C*-kompetitiv ist, falls es eine reelle Zahl* A *gibt mit*

$$K_S(P) \leq CK_{opt}(P) + A, \qquad \forall P \in \Pi.$$

Wie in [13, 5.2.3] erklärt kann es für NP-harte Probleme effiziente approximative Lösungen geben, deren Kosten eben nur um einen (hoffentlich kleinen) Faktor C höher liegen als die optimale Lösung. Als Beispiel sei hier das Problem des Handlungsreisenden genannt, welches NP-hart ist, und bei dem es gilt den kürzesten Rundweg zu finden, der eine gewisse Menge von Punkten der Ebene einmal besucht. Als annähernde Lösung kann man einen Rundweg um einen minimalen Spannbaum dieser Punkte nehmen. Nach [13, Theorem 5.2.3.2] ist dieser Weg weniger als doppelt so lang wie ein optimaler Rundweg, d.h. diese Lösung ist C-kompetitiv, mit $C \leq 2$.

In dieser Arbeit werden wir uns nur auf online-Algorithmen S beschränken, die das Erkundungsproblem lösen. Bei der Analyse der C-Kompetitivität werden wir $A = 0$ verlangen und C einen Kompetitivitätsfaktor nennen, wenn für alle $P \in \Pi$, $\frac{K_S(P)}{K_{opt}(P)} \leq C$ gilt.

Ziel dieser Abschlussarbeit

Als Problemmenge Π betrachten wir die Menge der Paare (P, S), wobei P ein ebenes, einfaches Polygon und S ein Punkt, der auf P oder im Inneren des von P begrenzten Gebietes liegt. Genauer verstehen wir unter einem ebenen, einfachen Polygon P einen geschlossenen, sich nicht schneidenden endlichen Streckenzug in $\mathbb{R}^2$. Nach dem Jordanschen Kurvensatz teilt P die Ebene in zwei Mengen, deren Durchschnitt P ist, so dass genau eine der beiden Mengen beschränkt ist, welches das Inneren $\overline{P}$ des Polygons P genannt wird. Das (offene) Äußere von P ist demnach $\mathbb{R}^2 \setminus \overline{P}$. Insbesondere soll $S \in \overline{P}$ gelten.

Ein Algorithmus A löst das Erkundungsproblem, falls es zu (P, S) einen Weg W liefert, d.h. eine stückweise stetig differenzierbare Funktion $W : [0, 1] \to \overline{P}$, mit $W(0) = S = W(1)$ und der Eigenschaft, dass es für jeden Punkt $X \in \overline{P}$ ein $t \in [0, 1]$ gibt, so dass die Verbindungsstrecke $\overline{W(t)X}$ ganz in $\overline{P}$ liegt. Die euklidische Länge von W deuten wir als die Kosten $K_A((P, S))$, d.h.

$$K_A((P, S)) = \int_0^1 \sqrt{W_1(t)^2 + W_2(t)^2} dt$$

In dieser Arbeit gehen wir davon aus, dass der Weg durch einen Streckenzug gegeben ist und das sich daher $K_A((P, S))$ als Summe der Längen der den Streckenzug bildenden Segmente berechnen lässt.

Ein optimaler Wert $K_{opt}((P, S))$ kann durch einen geeigneten offline-Algorithmus berechnet werden. Ziel dieser Arbeit ist es, eine untere Schranke für die Kompetitivitätsfaktoren $K_A((P, S))/K_{opt}((P, S))$ des Erkundungsproblems zu finden. Ist ein kompetitiver online-Algorithmus A gegeben, so definieren wir

$$KF(A) := \sup_{(P,S)\in\Pi} \frac{K_A((P, S))}{K_{opt}((P, S))}.$$

Um eine untere Schranke des Erkundungsproblems zu bestimmen, müssen wir $KF(A)$, über alle kompetitiven online-Algorithmen A minimieren, d.h. wir wollen

$$\min \{KF(A) \mid A \text{ ein kompetitiver } \textit{online}\text{-Algorithmus für alle } (P, S) \in \Pi\} \tag{1}$$

bestimmen.

In der Arbeit [7] schlagen Deng et al. als eine der Ersten eine kompetitive Strategie für das Erkundungsproblem vor. Ausgehend von der Tatsache, dass ein Weg im Inneren eines Polygons genau dann alle Punkte sieht, wenn er alle Kanten, bzw. alle Ecken des Polygons sieht, betrachten sie die Liniensegmente, die die Polygonkanten ins Innere fortsetzen. Solche Liniensegmente, die man bei jeder Erkundung mindestens einmal überschreiten muss, nennen sie notwendige Schnitte. Ein notwendiger Schnitt dominiert einen anderen, falls es keinen Weg einer Erkundung gibt, der den ersten Schnitt ausspart, um den zweiten zu erkunden.

Die Schnitte, die von keinem anderen dominiert werden, nennen sie wesentliche Schnitte. Die Aufgabe besteht darin, einen kürzesten Rundweg über alle wesentlichen Schnitte zu finden. Allerdings sind dabei a priori weder die wesentlichen Schnitte im Voraus bekannt, noch ist es klar in welcher Reihenfolge diese besucht werden sollten. Für rechtwinklige Polygone lösen Deng et al. dieses Problem durch Angabe eines Greedy-Algorithmus, der online einen in der L_1-Norm optimalen Weg berechnet. In der euklidischen Norm hat ihr online-Algorithmus daher den Kompetitivitätsfaktor $\sqrt{2} \approx 1,414$ (siehe [8]). Für allgemeine Polygone geben sie ohne Details einen Algorithmus mit einem Kompetitivitätsfaktor von 2016 an. Dieser Kompetitivitätsfaktor wurde von Hoffmann at el. in [12] auf 26.5 mithilfe einer neue Datenstruktur, der sogenannten *angle-hull*, reduziert. Nachfolgende Abbildung 1 zeigt den kürzesten Erkundungsrundgang, der durch ein Java-Applet von M. Schwarz (siehe [16]) erzeugt worden ist.

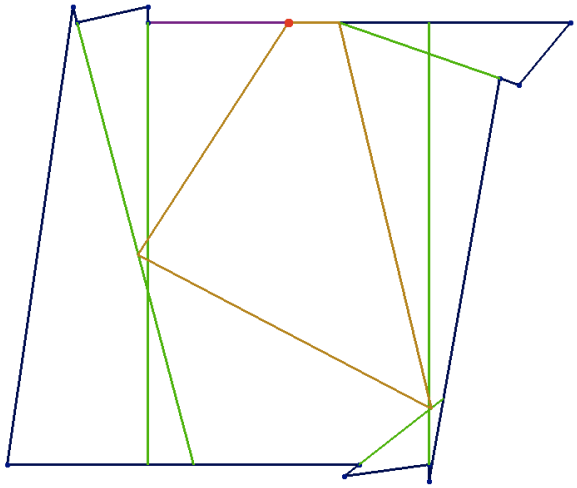


Abbildung 1: Beispiel eines kürzesten Erkundungsrundganges (gelbe Linie). Die grünen Linien stellen die notwendigen Schnitte dar, die es zu überqueren gilt, um das gesamte Polygon zu erkunden.

Ziel dieser Arbeit ist es, eine untere Schranke für den Kompetitivitätsfaktor einer (kompetitiven) Erkundungsstrategie eines Polygons anzugeben. Zentral bei unserer Betrachtung wird die Idee einer Schwelle sein, die jede Erkundung überqueren muss, um das Polygon vollständig zu erforschen (siehe Kapitel 2). Ausgehend von der einfachsten unteren Schranke 1.207 (siehe Abbildung 2 im nächsten Kapitel) werden wir versuchen, die Anzahl der Schwellen zu erhöhen, um entsprechend höhere untere Schranken zu erreichen. Für Fünfecke mit zwei Schwellen zeigen wir in Kapitel 3 eine untere Schranke von 1.2717. Im letzten Kapitel werden Polygone mit drei Schwellen betrachtet. Die untere Schranke 1.285 von Hagius et al. wird numerisch belegt. Darüber hinaus verbessern wir dieses Ergebnis auf 1.2945.

Danksagung

An dieser Stelle möchte ich die Gelegenheit nutzen, mich bei Herrn Professor Christian Icking für die herzliche und gute Betreuung meiner Abschlussarbeit und für das Korrekturlesen zu bedanken.

Ferner gilt mein Dank meiner Familie und insbesondere meiner Frau, die mich stets unterstützt und ermutigt hat, mein Diplom in Informatik zu beenden.

Mein Fernstudium begann im Wintersemester 1993/94 als Zweitstudium neben meinem Mathematikstudium - also vor über 25 Jahren! Das Fernstudium hat mir, wenn ich die Zeit dazu hatte, sehr viel Spaß gemacht. Die Qualität der Lehre und der Lehrenden habe ich stets als herausragend empfunden, weswegen mein besonderer Dank der Fernuniversität in Hagen gilt - als eine einzigartige Institution in Deutschland, die es Menschen allen Alters erlaubt auf höchstem Niveau zu studieren.

2 Polygone mit einer Schwelle

Eine der ersten unteren Schranken für den Kompetitivitätsfaktor einer Erkundungsstrategie von Polygonen ist $\frac{1+\sqrt{2}}{2} \cong 1.20710678118655$, welche nach [12] auf [7] zurückgeht. Als Begründung genügt es, ein rechtwinkliges, gleichschenkliges Dreieck mit Schenkellängen 1 zu betrachten, wie in Abbildung 2 angegeben:

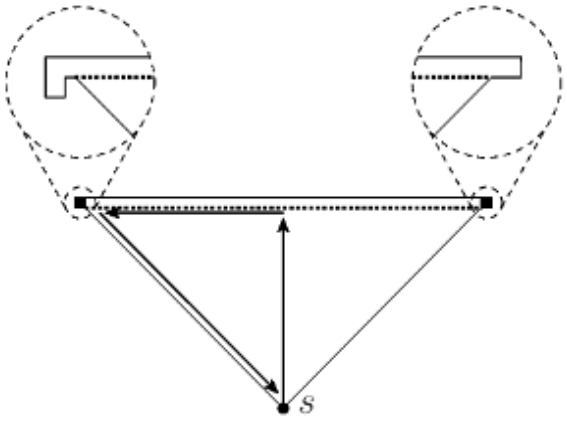


Abbildung 2: Untere Schranke 1.2071, Quelle:[11]

Der Startpunkt S einer Erkundung wird als die der Hypotenuse gegenüberliegende Ecke festgelegt. An einer der beiden anderen Ecken des Dreiecks ist eine verstecke Tasche angebracht. Die Hypotenuse selber dient als eine Schwelle, die ein jeder Erkundungsrundgang betreten muss, um eben diese Tasche zu finden. Weil das Polygon vom Startpunkt aus gesehen symmetrisch ist und weil die direkten (kürzesten) Wege vom Startpunkt zu den beiden Ecken gleich lang sind, kann man annehmen, dass es keinen Vorteil bringt, direkt eine Ecke auszusuchen, da sonst die Tasche an der anderen Ecke "platziert" werden würde. Die Erkundung muss daher den direkten Weg zur Mitte der Hypotenuse wählen, um daraufhin in eine der Ecken zu laufen. Der Gesamtlänge einer Erkundung beträgt somit mindestens $1 + \sqrt{2}$, während der optimale Weg zweimal die Schenkellänge, also 2, ist.

Wesentlich für diese Arbeit ist die Idee einer *Schwelle* in einem Polygon.

Definition 2. *Eine Schwelle in einem einfachen Polygon ist ein wesentlicher Schnitt, so dass es hinter diesem Schnitt höchstens einen weiteren wesentlichen Schnitt gibt.*

Hierbei soll "hinter" sich auf die Lage des Startpunktes einer Erkundung "vor"der zu betrachteten Schwelle beziehen. Ferner betrachten wir also nie mehr als zwei wesentliche Schnitte hintereinander, wenn wir von einer Schwelle sprechen.

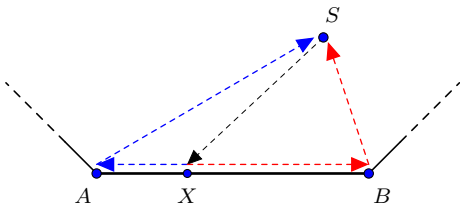


Abbildung 3: Erkundung einer Schwelle

In diesem Abschnitt wollen wir zeigen, dass die Situation in Abbildung 2 die größte untere Schranke für den Kompetitivitätsfaktor einer Erkundungsstrategie für ein Polygon mit nur einer Schwelle liefert. Wir nehmen nun an, dass wir ein Polygon mit genau einer Schwelle, sagen wir zwischen zwei Ecken A und B, ausgehend von einem Punkt S erkunden wollen. Nach Skalierung, Translation und gegebenenfalls Drehung können wir annehmen, dass $A = (0, 0)$, $B = (1, 0)$ und $S = (a, b)$ ein Punkt der oberen Halbebene ist, d.h. $b > 0$ (siehe Abbildung 3). Wir suchen also einen Weg von S zu einem Punkt $X = (x, 0)$ auf $\overline{AB}$, mit $x \in [0, 1]$, der entweder nach A oder B fortgesetzt wird, um dann wieder nach S zurückzukehren. Damit ergeben sich die Weglängen $W_0 = |SX| + |XA| + |AS|$, für den Fall, dass die versteckte Tasche in A liegt, und $W_1 = |SX| + |XB| + |BS|$, für den Fall, dass sie in B liegt. Der Abstand von S zu X ist durch $s(x) = |SX| = \sqrt{(x-a)^2 + b^2}$ gegeben, welche eine stetig differenzierbare positive Funktion ist, mit Minimum in $x = a$, falls $a \in [0, 1]$.[1] Die optimalen Weglängen sind $2|AS| = 2|S| = 2l$ für $l = \sqrt{a^2 + b^2}$ und $2|BS| = 2r$ für $r = \sqrt{(a-1)^2 + b^2} = \sqrt{a^2 + b^2 + 1 - 2a}$. Man hat also $l < r$ genau dann, wenn $a < \frac{1}{2}$ und $l = r$ genau dann wenn $a = \frac{1}{2}$.

Die relativen Weglängen sind ebenfalls Funktionen in x, d.h.

$$F_0(x) = \frac{W_0}{2l} = \frac{s(x) + x}{2l} + \frac{1}{2}, \qquad F_1(x) = \frac{W_1}{2r} = \frac{s(x) + 1 - x}{2r} + \frac{1}{2}$$

Während $F_0(x)$ monoton wachsend auf $[0, 1]$ ist, ist $F_1(x)$ monoton fallend. Somit gibt es ein x^* mit $F_0(x^*) = F_1(x^*)$, d.h. x^* erfüllt

$$(r - l)s(x^*) = l - (r + l)x^* \tag{2}$$

Der Punkt $X = (x^*, 0)$ ist damit der Landepunkt auf $\overline{AB}$, bei dem beide relativen Weglängen

[1] Ansonsten befindet sich das Minimum am Rand bei $x = 0$ oder $x = 1$.

gleich sind. Wegen der Monotonie ist für $x \neq x^*$ einer der Funktionswerte von $F_0(x)$ und $F_1(x)$ größer als $F_0(x^*)$. Falls $x^* \in [0,1]$ löst dieser Wert daher das Minimax-Problem

$$F_0(x^*) = F_1(x^*) = \min_{x\in[0,1]} \max\{F_0(x), F_1(x)\}$$

und beschränkt den Kompetitivitätsfaktor C einer Erkundungsstrategie nach unten, d.h. $F_0(x^*) \leq C$. Im Fall $a = \frac{1}{2}$ gilt $l = r = \sqrt{\frac{1}{4} + b^2}$ und somit $x^* = \frac{1}{2}$ und $s(x^*) = b$. Dann ist

$$F_0(x^*) = F_1(x^*) = \frac{b + \frac{1}{2}}{2\sqrt{\frac{1}{4} + b^2}} + \frac{1}{2} = \frac{2b+1}{2\sqrt{4b^2+1}} + \frac{1}{2} \leq \frac{\sqrt{2}+1}{2}, \tag{3}$$

da wegen $b > 0$ auch $2b + 1 = \sqrt{4b^2 + 4b + 1} \leq \sqrt{16b^2 + 4} = \sqrt{2}\sqrt{4b^2 + 1}$ gilt mit Gleichheit bei $b = \frac{1}{2}$. Im Fall $a \neq \frac{1}{2}$ gilt $l \neq r$ und somit ist die Gleichung (2) äquivalent zur Wurzelgleichung:

$$\sqrt{(x-a)^2 + b^2} = s(x) = \frac{l}{r-l} - \frac{r+l}{r-l}x = \left(\frac{l+r}{l-r}\right)(x-a) + \left(\frac{a(r+l)-l}{l-r}\right) \tag{4}$$

Das folgende Lemma gibt die Lösung einer solchen Wurzelgleichung an.

Lemma 3. *Seien u, v, w reele Zahlen mit $u \geq 0$ und $v^2 > 1$. Dann hat die Gleichung*

$$\sqrt{z^2 + u} = vz + w$$

die eindeutige Lösung

$$z^* = \frac{\epsilon\sqrt{w^2 + u(v^2-1)} - vw}{v^2 - 1}$$

wobei $\epsilon = \operatorname{sign}(v) \in \{1, -1\}$ das Vorzeichen von v ist.

Beweis. Da $u \geq 0$ und $v^2 > 1$ ist $w^2 + u(v^2-1) \geq 0$ und z^* wohldefiniert. Durch Einsetzen überprüfen wir, dass z^* eine Lösung ist.

$$\begin{aligned}
(z^*)^2 + u &= \frac{\left(\pm\sqrt{w^2 + u(v^2-1)} - vw\right)^2 + u(v^2-1)^2}{(v^2-1)^2} \\
&= \left(w^2 + u(v^2-1) \mp 2\sqrt{w^2 + u(v^2-1)}vw + v^2w^2 + uv^4 - 2uv^2 + u\right)(v^2-1)^{-2} \\
&= \left(v^2(w^2 + uv^2 - u) \mp 2v\sqrt{w^2 + u(v^2-1)}w + w^2\right)(v^2-1)^{-2} \\
&= \left(\pm v\sqrt{w^2 + u(v^2-1)} - w\right)^2 (v^2-1)^{-2} \\
&= \left(v\frac{\pm\sqrt{w^2 + u(v^2-1)}}{v^2-1} - \frac{w}{v^2-1}\right)^2 \\
&= \left(v\frac{\pm\sqrt{w^2 + u(v^2-1)} - vw}{v^2-1} + w\right)^2 = (vz^* + w)^2
\end{aligned}$$

Damit gilt

$$\sqrt{(z^*)^2 + u} = \sqrt{(vz^* + w)^2} = |vz^* + w|.$$

Ist $v > 1$ dann ist z^* mit $\epsilon = 1$ eine Lösung, denn

$$vz^* + w = \frac{v\sqrt{w^2 + u(v^2-1)} - v^2 w}{v^2 - 1} + \frac{wv^2 - w}{v^2 - 1} = \frac{v\sqrt{w^2 + u(v^2-1)} - w}{v^2 - 1} > 0.$$

Falls $v < -1$, dann ist z^* mit $\epsilon = -1$ eine Lösung, denn

$$vz^* + w = \frac{-v\sqrt{w^2 + u(v^2-1)} - v^2 w}{v^2 - 1} + \frac{wv^2 - w}{v^2 - 1} = \frac{|v|\sqrt{w^2 + u(v^2-1)} - w}{v^2 - 1} > 0.$$

□

Wir wenden nun Lemma 3 auf unsere Wurzelgleichung (4) an und setzen

$$v = \frac{l + r}{l - r} \qquad \text{und} \qquad w = \frac{a(r + l) - l}{l - r}.$$

Für $z = x - a$ wollen wir also $\sqrt{z^2 + b^2} = vz + w$ lösen. Es gelten

$$\begin{aligned} v^2 - 1 &= \frac{(l+r)^2 - (l-r)^2}{(l-r)^2} = \frac{4lr}{(l-r)^2} > 0 \\ w^2 + b^2(v^2 - 1) &= \frac{(l - a(l+r))^2 + 4b^2 lr}{(l-r)^2} \\ -vw &= \frac{(l+r)(l - a(l+r))}{(l-r)^2}. \end{aligned}$$

Wegen $v > 0$ genau dann wenn $l > r$ genau dann, wenn $a > 0.5$, ist die Lösung der Wurzelgleichung (4) nach Lemma 3:

$$x^* = a + \frac{(l-r)\sqrt{(l - a(l+r))^2 + 4b^2 lr} + (l+r)(l - a(l+r))}{4lr}. \tag{5}$$

falls $a > 0.5$ und für $a < 0.5$

$$x^* = a + \frac{(r-l)\sqrt{(l - a(l+r))^2 + 4b^2 lr} + (l+r)(l - a(l+r))}{4lr}. \tag{6}$$

Satz 4. *Jede kompetitivie Strategie zur Erkundung eines einfachen Polygons mit genau einer Schwelle hat einen Kompetitivitätsfaktor von mindestens* 1.207.

Beweis. Wie in der vorrausgehenden Diskussion gezeigt, können wir annehmen, dass die zu erkundende Schwelle das Segment von $(0, 0)$ nach $(1, 0)$ ist und dass der Startpunkt bei $S = (a, b)$ liegt mit $b > 0$. Im Falle $a = 0.5$ ist, wie in (3) gezeigt, $\frac{2b+1}{2\sqrt{4b^2+1}} + \frac{1}{2}$ eine untere Schranke für einen Kompetitivitätsfaktor mit einem Maximum von $\frac{\sqrt{2}+1}{2} \approx 1.207$ bei $b = 0.5$.

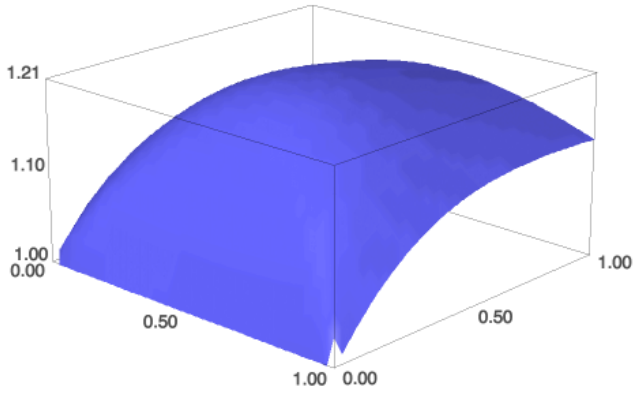


Abbildung 4: Graph von $(a, b) \to F_0(x^*)$

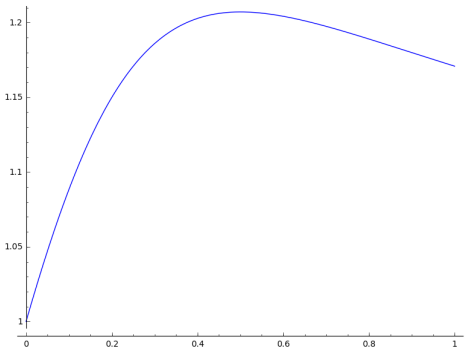


Abbildung 5: Querschnitt für $a = 0.5$

Im Fall $a \neq 0.5$ ist eine untere Schranke durch

$$F_0(x^*) = F_1(x^*) = \frac{\frac{l}{r-l} - \frac{r+l}{r-l}x^* + x^*}{2l} + \frac{1}{2} = \frac{2x^* - 1}{2(l-r)} + \frac{1}{2}$$

gegeben, wobei x^* die Gleichung (2) löst und somit $s(x^*) = \frac{l}{r-l} - \frac{r+l}{r-l}x^*$ gilt. Explizit gilt dabei

$$x^* = a \pm \frac{(l-r)\sqrt{(l-a(l+r))^2 + 4b^2lr} + (l+r)(l-a(l+r))}{4lr}. \tag{7}$$

je nachdem, ob $a < 0.5$ oder $a > 0.5$ ist. Abbildung 4 zeigt den Graphen $F_0(x^*)$ über dem Quadrat $(a, b) \in [0, 1]^2$. Für $a = 0.5$ zeigt Abbildung 5 den Querschnitt des Graphen in Abhängigkeit von $b \in [0, 1]$. Es ergibt sich ein Maximum bei $(a, b) = (0.5, 0.5)$ von $\frac{\sqrt{2}+1}{2}$. □

3 Erkundung von Polygonen mit zwei Schwellen

In diesem Kapitel werden wir Polygone mit zwei Schwellen betrachten, um eine höhere untere Schranke für den Kompetitivitätsfaktor einer Erkundungsstrategie zu erhalten. Eine solche Strategie muss demnach mindestens zwei Schwellen betreten, um möglicherweise versteckte Taschen ausfindig zu machen. Ferner wollen wir das Polygon symmetrisch aufbauen, so dass beide Schwellen im gleichen Abstand und gleichen Winkel zum Startpunkt stehen und es irrelevant ist, welche Schwelle eine Erkundung als erste auswählt.

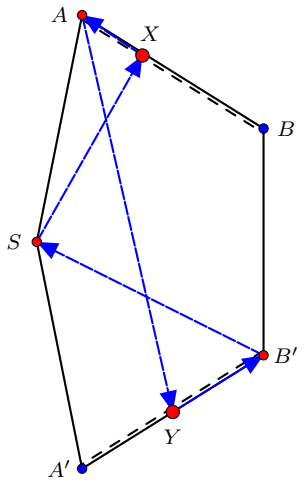


Abbildung 6: Erkundung zweier Schwellen

Das zu betrachtende Polygon besteht aus den fünf Ecken S, A, B, A' und B', wobei A' bzw. B' die gespiegelten Punkte A bzw. B sind, längst einer Geraden g auf der S liegt (siehe Abbildung 6). Nach geeigneter Drehung, Skalierung und Translation können wir annehmen, dass $S = (0, 0)$ der Ursprung des kartesischen Koordinatensystems und g die x-Achse ist und dass die Punkte A und B die Koordinaten $A = (a, b)$ bzw. $B = (1, c)$ haben, mit $a < 1$ und $b, c \geq 0$, so dass $A' = (a, -b)$ und $B' = (1, -c)$ gilt. Wir nehmen weiter an, dass B rechts vom Segment $\overline{SA}$ liegt, so dass die Segmente $\overline{AB}$ und $\overline{A'B'}$ vollständig von S aus sichtbar sind. Ein solches Polygon sei mit $P_{(a,b,c)}$ bezeichnet.

Wir legen ferner fest, dass der Startpunkt einer Erkundung stets der Punkt S sei. Die Segmente $\overline{AB}$ bzw. $\overline{A'B'}$ bilden unsere beiden Schwellen, die eine Erkundung betreten muss, um beide versteckte Taschen zu erkunden. Eine der beiden Taschen ist entweder in der Ecke A oder in der Ecke B versteckt, während die andere entweder in A' oder in B' versteckt ist. Wir nehmen an, dass die Ecken erst dann einsehbar sind, wenn die Schwellen $\overline{AB}$ bzw. $\overline{A'B'}$ erreicht worden sind, so dass jede Erkundung gezwungen ist, auf beiden Schwellen zu landen.

Aus Symmetriegründen können wir ohne Beschränkung der Allgemeinheit annehmen, dass eine Erkundung zuerst die Schwelle $\overline{AB}$ betritt und danach die Schwelle $\overline{A'B'}$. Abhängig von den möglichen Landepunkten X auf der Schwelle $\overline{AB}$ und Y auf der Schwelle $\overline{A'B'}$ errechnet sich die Weglänge einer Erkundung.

Die vier Paare $(A, A'), (A, B'), (B, A')$ und (B, B') stellen die vier mögliche Szenarien der versteckten Taschen dar. Diese werden durch ein Binärpaar (e_0, e_1) kodiert, wobei

$$(A, A') \leftrightarrow (0, 0), \qquad (A, B') \leftrightarrow (0, 1), \qquad (B, A') \leftrightarrow (1, 0), \qquad (B, B') \leftrightarrow (1, 1).$$

Manchmal werden wir ein solches Binärpaar (e_0, e_1) auch als Zahl $m = e_0 + 2 \cdot e_1$ interpretieren. Für jede dieser vier Szenarien berechnet sich die Weglänge W_m einer Erkundung als Funktion in Abhängigkeit der Landepunkte X und Y auf den Schwellen $\overline{AB}$ bzw. $\overline{A'B'}$ sowie natürlich in Abhängigkeit der Geometrie des Polygons, d.h. der Werte a, b, c. Tabelle 7 stellt die Wege und ihre Weglängen für alle vier Szenarien zusammen.

	Weg	Weglänge W_m	Optimum ω_m
0	SXAYA'S	$W_0 = \lvert X\rvert + \lvert XA\rvert + \mathrm{dist}(A, Y) + \lvert YA'\rvert + \lvert A'\rvert$	$2\lvert A\rvert + \mathrm{dist}(A, A')$
1	SXAYB'S	$W_1 = \lvert X\rvert + \lvert XA\rvert + \mathrm{dist}(A, Y) + \lvert YB'\rvert + \lvert B'\rvert$	$\lvert A\rvert + \mathrm{dist}(A, B') + \lvert B'\rvert$
2	SXBYA'S	$W_2 = \lvert X\rvert + \lvert XB\rvert + \mathrm{dist}(B, Y) + \lvert YA'\rvert + \lvert A'\rvert$	$\lvert B\rvert + \mathrm{dist}(B, A') + \lvert A'\rvert$
3	SXBYB'S	$W_3 = \lvert X\rvert + \lvert XB\rvert + \mathrm{dist}(B, Y) + \lvert YB'\rvert + \lvert B'\rvert$	$2\lvert B\rvert + \lvert BB'\rvert$

Abbildung 7: Weglängen der Erkundung mit zwei Schwellen

Je nach Konstruktion des Polygons kann es vorkommen, dass Y von A bzw. von B nicht sichtbar ist. Daher soll mit $\mathrm{dist}(-, -)$ der kürzeste Abstand innerhalb des Polygons gemeint sein. Beispielsweise ist in Abbildung 8, Y von A aus nicht sichtbar. Der kürzeste Weg folgt demnach über den Ursprung S, d.h. $\mathrm{dist}(A, Y) = |A| + |Y|$.

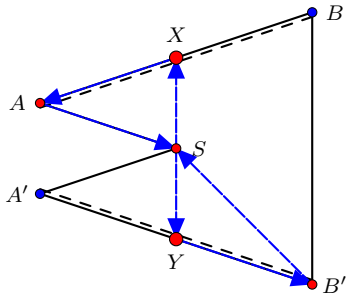


Abbildung 8: Y ist von A nicht sichtbar.

Die optimalen Weglängen seien mit ω_m bezeichnet und sind ebenfalls in Tabelle 7 aufgelistet. Aus Symmetriegründen ist $\mathrm{dist}(A, B') = \mathrm{dist}(A', B)$, sowie $|A| = |A'|$ und $|B| = |B'|$. Damit gilt $\omega_2 = \omega_3$. Die relativen Weglängen ergeben sich als $F_m := W_m/\omega_m$ für $m \in \{0, 1, 2, 3\}$.

Die Landepunkte X und Y sind eindeutig durch

$$X = A + x(B - A) \qquad \text{und} \qquad Y = A' + y(B' - A)$$

mittels Zahlen $x, y \in [0, 1]$ parametrisiert. Damit sind die Längen der Segmente $\overline{XA}, \overline{XB}, \overline{YA'}$ und $\overline{YB'}$, die in den Weglängen W_m auftreten durch

$$|XA| = \lambda x, \qquad |XB| = \lambda(1 - x), \qquad |YA'| = \lambda y, \qquad |YB'| = \lambda(1 - y),$$

gegeben, wobei $\lambda = |AB| = |A'B'|$ die Länge der Schwellen ist. Wenn die Polygonparameter (a, b, c) fest gewählt sind, sind die relativen Weglängen F_m Funktionen in x und y.

Wir haben also

$$\begin{aligned}
F_0(x, y) &= (|X| + \lambda x + \mathrm{dist}(A, Y) + \lambda y + |A|)/\omega_0 && (8)\\
F_1(x, y) &= (|X| + \lambda x + \mathrm{dist}(A, Y) + \lambda(1 - y) + |B|)/\omega_1 && (9)\\
F_2(x, y) &= (|X| + \lambda(1 - x) + \mathrm{dist}(B, Y) + \lambda y + |A|)/\omega_2 && (10)\\
F_3(x, y) &= (|X| + \lambda(1 - x) + \mathrm{dist}(B, Y) + \lambda(1 - y) + |B|)/\omega_3 && (11)
\end{aligned}$$

Insbesondere sind für festes $x \in [0, 1]$ die Funktionen $F_0(x, -)$ und $F_2(x, -)$ monoton wachsend, während $F_1(x, -)$ und $F_3(x, -)$ monoton fallend sind.

Erinnern wir uns, dass wir als Begründung für die untere Schranke 1.2071 des Dreiecks aus Abbildung 2 angegeben haben, dass man annehmen kann, dass wenn ein Erkundungsrundweg die Schwelle nicht in der Mitte betritt, die Tasche an der vom Landepunkt entfernteren Ecke versteckt ist. Eine Erkundung mit minimaler relativer Weglänge ist daher gezwungen, auf der Mitte der Schwelle zu landen. Alternativ kann man aber auch argumentieren, dass bei einer der zwei möglichen Szenarien eine Erkundung keine bestmögliche Wegstrecke zurücklegt. Die Mitte der Schwelle ist daher der Landepunkt, bei dem die (relativen) Weglängen einer Erkundung beider Szenarien gleich sind.

Im Fall eines Fünfecks mit zwei Schwellen müssen wir ebenso die optimalen Landepunkte X^* und Y^* auf den beiden Schwellen finden, so dass jede Erkundung, die diese Punkte verfehlt, in mindestens einer der vier Szenarien eine größere, relative Weglänge zurücklegt. Hierbei unterscheiden wir zuerst, ob eine Erkundung von X^* über die Ecke A oder die Ecke B führt, so dass es zu jeden dieser beiden Fällen einen optimalen Landepunkt Y_l^* bzw. Y_r^* auf der Schwelle $\overline{A'B'}$ gibt. Wir teilen also die vier Funktionen in zwei Funktionenpaare (F_0, F_1) und (F_2, F_3) auf und untersuchen als erstes für einen festen Wert $x \in [0, 1]$ die Funktionen

$F_m(x,-):[0,1]\to\mathbb{R}$. Ziel ist es nun für jedes $x\in[0,1]$ Werte $y_l(x)$ und $y_r(x)$ zu finden, so dass $F_0(x,y_l(x))=F_1(x,y_l(x))$ und $F_2(x,y_r(x))=F_3(x,y_r(x))$ gilt. Wegen der Monotonie der Funktionen $F_m(x,-)$ gilt dann für alle $x\in[0,1]$:

$$\begin{aligned} F_0(x,y_l(x))=F_1(x,y_l(x)) &= \min_{y\in[0,1]}\max\{F_0(x,y),F_1(x,y)\}\\ F_2(x,y_r(x))=F_3(x,y_r(x)) &= \min_{y\in[0,1]}\max\{F_2(x,y),F_3(x,y)\}\end{aligned}$$

Für Polygone $P_{(a,b,c)}$ mit $a>0$ werden diese Funktionen y_l und y_r im nachfolgenden Abschnitt explizit angegeben. Falls $\omega_0=\omega_1$ gilt, dann ist y_l konstant. Ebenso ist y_r konstant, falls $\omega_2=\omega_3$ gilt. Dies sollte analog zum Fall in Abbildung 2 gesehen werden, bei dem stets der Mittelpunkt der Schwelle als Landepunkt ausgewählt wird. Es ist allerdings nicht möglich, dass alle optimalen Weglängen gleich sind. Für den Fall $a<0$, bei dem sich die Schwellen nicht vollständig sehen können, können wir y_l und y_r nicht mehr explizit angeben und werden ein numerisches Verfahren anwenden, um sie zu ermitteln.

Die Funktionen y_l und y_r definieren nun einstellige Funktion $f_0,f_1:[0,1]\to\mathbb{R}$ mit

$$f_0(x)=F_0(x,y_l(x))=\min_{y\in[0,1]}\max\{F_0(x,y),F_1(x,y))\tag{12}$$

und

$$f_2(x)=F_3(x,y_r(x))=\min_{y\in[0,1]}\max\{F_2(x,y),F_3(x,y))\tag{13}$$

Um den optimalen Landepunkt X^* auf der Schwelle $\overline{AB}$ zu finden suchen wir ein $x^*\in\mathbb{R}$, so dass $f_0(x^*)=f_1(x^*)$ gilt oder äquivalent:

$$F_0(x^*,y_l(x^*))=F_1(x^*,y_l(x^*))=F_2(x^*,y_r(x^*))=F_3(x,y_r(x^*)).\tag{14}$$

Aus der Monotonie von f_0 und f_1 ergibt sich

$$f_0(x^*)=\min_{x\in[0,1]}\max\{f_0(x),f_1(x)\}=\min_{x\in[0,1]}\max\{F_0(x,y_l(x)),F_2(x,y_r(x))\}.\tag{15}$$

Somit stellt $F_0(x^*,y_l(x^*))$ eine untere Schranke für das Minimax-Problem dar:

$$F_0(x^*,y_l(x^*))\le\min_{(x,y)\in[0,1]^2}\max\{F_0(x,y),F_1(x,y),F_2(x,y),F_3(x,y)\}.\tag{16}$$

Denn für alle $(x,y)\in[0,1]^2$ ist

$$\begin{aligned} F_0(x^*,y_l(x^*)) &= \min_{x\in[0,1]}\max\{F_0(x,y_l(x)),F_2(x,y_r(x))\}\\ &= \min_{x\in[0,1]}\max\left\{\min_{y\in[0,1]}\max\{F_0(x,y),F_1(x,y)\},\min_{y\in[0,1]}\max\{F_2(x,y),F_3(x,y)\}\right\}\\ &\le \min_{(x,y)\in[0,1]^2}\max\{F_0(x,y),F_1(x,y),F_2(x,y),F_3(x,y)\}.\end{aligned}$$

Das explizite (symbolische) Auffinden des Wertes x^* stellt sich als schwierig heraus und für konkrete Instanzen der Parameter a, b, c kann man das Newton Verfahren anwenden, um x^* hinreichen genau anzunähern. Durch die explizite Darstellung der Funktionen y_l und y_r und eine einfache Implementierung des Newton Verfahrens mittels Sage/Python kann x^* und $y_l(x^*)$ bzw. $y_r(x^*)$ schnell gefunden werden. Mehrere Beispiele werden im letzten Abschnitt aufgelistet.

Die Ungleichung (16) besagt insbesondere, dass die relative Weglänge eines Erkundungsrundganges für eine der Fälle (A, A'), (A, B'), (B, A') oder (B, B') größer gleich $F_0(x^*, y_l(x^*))$ ist. Damit ist der Kompetitivitätsfaktor einer Erkundungsstrategie für unser Polygon durch $F_0(x^*, y_l(x^*))$ nach unten beschränkt. Wir führen dafür folgende Notation ein:

Definition 5. *Für ein Polygon $P_{(a,b,c)}$ bezeichne $KF_{(a,b,c)} = F_0(x^*, y^*)$ eine untere Schranke des Erkundungsproblems für $P_{(a,b,c)}$, wenn $(x^*, y^*) \in [0,1]^2$ die Ungleichung (16) erfüllt.*

3.1 Schwellen mit uneingeschränkter Sicht

Wie im vorigen Abschnitt erklärt wollen wir jetzt die Funktionen y_l und y_r bestimmen und betrachten dazu Polygone $P_{(a,b,c)}$ mit $a > 0$[2]. Damit können sich die Schwellen gegenseitig sehen und die Abstandsberechnung in den Funktionen F_m vereinfacht sich zu

$$F_0(x,y) = (|X| + \lambda x + |AY| + \lambda y + |A|)/\omega_0 \quad (17)$$

$$F_1(x,y) = (|X| + \lambda x + |AY| + \lambda(1-y) + |B|)/\omega_1 \quad (18)$$

$$F_2(x,y) = (|X| + \lambda(1-x) + |BY| + \lambda y + |A|)/\omega_2 \quad (19)$$

$$F_3(x,y) = (|X| + \lambda(1-x) + |BY| + \lambda(1-y) + |B|)/\omega_3 \quad (20)$$

wobei $\lambda = |AB| = \sqrt{(1-a)^2 + (c-b)^2}$ die Länge einer Schwelle ist. Für $x, y \in [0,1]$ hatten wir die beiden Landepunkte $X = A + x(B - A)$ und $Y = A' + y(B' - A')$ betrachtet.

Wir nehmen an, dass der erste Landepunkt X, d.h. der Wert $x \in [0,1]$ fest gewählt wurde und dass sich die erste Tasche in der Ecke A befindet. Falls es ein $y^* \in [0,1]$ gibt, so dass $F_0(x, y^*) = F_1(x, y^*)$ gilt, dann folgt aus der Monotonie von $F_0(x, -)$ und $F_1(x, -)$ die Gültigkeit von $F_0(x, y^*) = \min_{y \in [0,1]} \max\{F_0(x,y), F_1(x,y)\}$. Betrachten wir daher die Gleichung $F_0(x,y) = F_1(x,y)$ mit festem x. Zur Vereinfachung sei $k_0 = |X| + \lambda x + |A|$ und $k_1 = |X| + \lambda x + |B|$. Dann gilt $F_0(x,y) = F_1(x,y)$ genau dann wenn, $\omega_1(|AY| + \lambda y + k_0) = \omega_0(|AY| + \lambda(1-y) + k_1)$ genau dann, wenn

$$(\omega_1 - \omega_0)|AY| = \omega_0(\lambda + k_1) - \omega_1 k_0 - \lambda(\omega_0 + \omega_1)y \quad (21)$$

[2]Nach unserer Annahme liegt $B = (c, 1)$ rechts von der Geraden, die den Ursprung mit $A = (a, b)$ verbindet, und es gilt $b \geq ac$.

Nehmen wir zuerst $\omega_1 \neq \omega_0$ an, dann folgt $|AY| = \frac{\omega_0(\lambda+k_1)-\omega_1 k_0}{\omega_1-\omega_0} + \lambda\frac{\omega_0+\omega_1}{\omega_0-\omega_1}y$. Setzen wir

$$C = \frac{\omega_0+\omega_1}{\omega_0-\omega_1} \qquad \text{und} \qquad g(x) = \frac{\omega_0(\lambda+k_1)-\omega_1 k_0}{\omega_1-\omega_0}, \tag{22}$$

dann ist also $F_0(x,y) = F_1(x,y)$ genau dann, wenn die Funktionalgleichung

$$|AY| = \lambda C y + g(x) \tag{23}$$

gilt. Da der Abstand von Y zu A durch $|AY| = \lambda\sqrt{\left(y - \frac{2b(b-c)}{\lambda^2}\right)^2 + \left(\frac{2b(1-a)}{\lambda^2}\right)^2}$ gegeben ist, kann Gleichung (23) zu

$$\sqrt{\left(y - \frac{2b(b-c)}{\lambda^2}\right)^2 + \left(\frac{2b(1-a)}{\lambda^2}\right)^2} = C\left(y - \frac{2b(b-c)}{\lambda^2}\right) + \frac{g(x)\lambda + 2b(b-c)C}{\lambda^2} \tag{24}$$

umgeformt werden. Wegen $C^2 - 1 = \frac{((\omega_0+\omega_1)^2-(\omega_0-\omega_1)^2}{(\omega_0-\omega_1)^2} = \frac{4\omega_0\omega_1}{(\omega_0-\omega_1)^2} > 0$ können wir Lemma 3 anwenden, welches uns folgende Lösung der Gleichung (23) liefert:

$$y_l(x) = \frac{2b(b-c)}{\lambda^2} + \frac{\epsilon\sqrt{w^2 + u(C^2-1)} - Cw}{C^2-1} \tag{25}$$

mit

$$u = \left(\frac{2b(1-a)}{\lambda^2}\right)^2 \qquad C = \frac{\omega_0+\omega_1}{\omega_0-\omega_1} \qquad w = \frac{g(x)}{\lambda} + \frac{2b(b-c)}{\lambda^2}C$$

und ϵ das Vorzeichen von $\omega_0 - \omega_1$. Durch Einsetzen erhalten wir

$$y_l(x) = \frac{2b(b-c)}{\lambda^2} + \frac{\epsilon\sqrt{\left(\frac{g(x)\lambda+2b(b-c)C}{\lambda^2}\right)^2 + \left(\frac{2b(1-a)}{\lambda^2}\right)^2(C^2-1)} - C\left(\frac{g(x)\lambda+2b(b-c)C}{\lambda^2}\right)}{C^2-1}. \tag{26}$$

Die Lösung kann folgendermaßen vereinfacht werden:

$$y_l(x) = \frac{\epsilon\sqrt{(g(x)\lambda + 2b(b-c)C)^2 + 4b^2(1-a)^2(C^2-1)} - g(x)C\lambda - 2b(b-c)}{\lambda^2(C^2-1)}, \tag{27}$$

Der Funktionswert von $F_1(x, y_l(x))$ ist dann wie folgt gegeben. Da $k_1 = |X| + \lambda x + |B|$ und die Funktionalgleichung (23) gilt, d.h. $|AY| = C\lambda y_l(x) + g(x)$ für die Lösungsfunktion y_l, erhalten wir im Fall $\omega_1 \neq \omega_0$:

$$\begin{aligned} F_1(x, y_l(x)) &= \frac{1}{\omega_1}(k_1 + C\lambda y_l(x) + g(x) + \lambda(1 - y_l(x))) && (28)\\ &= \frac{1}{\omega_1}(\lambda(C-1)y_l(x) + \frac{\omega_1}{\omega_1-\omega_0}(|B| - |A| + \lambda)) && (29)\\ &= \frac{1}{\omega_0-\omega_1}(2\lambda y_l(x) - |B| + |A| - \lambda) && (30) \end{aligned}$$

Falls $\omega_0 = \omega_1$, dann folgt aus (21),

$$y_l(x) = \frac{1}{2} + \frac{k_1 - k_0}{2\lambda} = \frac{1}{2} + \frac{|B| - |A|}{2\lambda} \tag{31}$$

Analog geht man bei der Betrachtung der relativen Weglängen bei festem Wert x und erster Tasche in B vor. Die Funktion $F_2 = (|X| + \lambda(1-x) + |BY| + \lambda y + |A|)/\omega_2$ ist monoton wachsend in y, während $F_3 = (|X| + \lambda(1-x) + |BY| + \lambda(1-y) + |B|)/\omega_3$ monoton fallend in y ist. Setzen wir wieder $k_2 = |X| + \lambda(1-x) + |A|$ und $k_3 = |X| + \lambda(1-x) + |B|$, dann ist $F_2(x,y) = F_3(x,y)$ genau dann, wenn

$$(\omega_3 - \omega_2)|BY| = \omega_2(\lambda + k_3) - \omega_3 k_2 - (\omega_2 + \omega_3)\lambda y \tag{32}$$

Falls $\omega_2 \neq \omega_3$ setzen wir

$$D = \frac{\omega_2 + \omega_3}{\omega_2 - \omega_3} \qquad \text{und} \qquad h(x) = \frac{\omega_2(\lambda + k_3) - \omega_3 k_2}{\omega_3 - \omega_2}. \tag{33}$$

Dann gilt $F_2(x,y) = F_3(x,y)$ genau dann, wenn die Funktionalgleichung

$$|BY| = D\lambda y + h(x) \tag{34}$$

erfüllt ist. Da der Abstand von B und Y durch

$$|BY| = \lambda\sqrt{\left(y - 1 - \frac{2c(b-c)}{\lambda^2}\right)^2 + \left(\frac{2c(1-a)}{\lambda^2}\right)^2} \tag{35}$$

gegeben ist, folgt aus Gleichung (34)

$$\sqrt{\left(y - 1 - \frac{2c(b-c)}{\lambda^2}\right)^2 + \left(\frac{2c(1-a)}{\lambda^2}\right)^2} = D\left(y - 1 - \frac{2c(b-c)}{\lambda^2}\right) + D + \frac{2c(b-c)D}{\lambda^2} + \frac{h(x)}{\lambda} \tag{36}$$

Wir wenden wieder Lemma 3 an. Wie zuvor ist $D^2 - 1 = \frac{4\omega_2\omega_3}{(\omega_2-\omega_3)^2} > 0$ und somit ist nach Lemma 3 die Lösung der Gleichung von (34) durch

$$y^* = 1 + \frac{2c(b-c)}{\lambda^2} + \frac{\epsilon\sqrt{w^2 + u(D^2-1)} - Dw}{D^2 - 1} \tag{37}$$

gegeben, mit ϵ dem Vorzeichen von $\omega_2 - \omega_3$, sowie

$$u = \left(\frac{2c(1-a)}{\lambda^2}\right)^2 \qquad \text{und} \qquad w = D + \frac{2c(b-c)D}{\lambda^2} + \frac{h(x)}{\lambda}.$$

Durch Einsetzen und Vereinfachen erhalten wir

$$y_r(x) = \frac{\epsilon\sqrt{(D\lambda^2 + 2c(b-c)D + \lambda h(x))^2 + 4c^2(1-a)^2(D^2-1)} - D\lambda h(x) - 2c(b-c) - 1}{\lambda^2(D^2-1)}. \tag{38}$$

Der Funktionswert von $F_2(x, y_r(x))$ ergibt sich unter Benutzung der Funktionalgleichung (34) und der Definition der Funktion h:

$$\begin{aligned} F_2(x, y_r(x)) &= \frac{1}{\omega_2}(k_2 + D\lambda y_r(x) + h(x) + \lambda y_r(x)) && (39)\\ &= \frac{1}{\omega_2}(\lambda(D+1)y_r(x) + \frac{\omega_2}{\omega_3 - \omega_2}(|B| - |A| + \lambda)) && (40)\\ &= \frac{1}{\omega_2 - \omega_3}(2\lambda y_r(x) - |B| + |A| - \lambda)) && (41) \end{aligned}$$

Im Falle von $\omega_2 = \omega_3$ folgert man aus (32) dass $F_2(x, y) = F_3(x, y)$ gilt genau dann, wenn

$$y_r(x) = \frac{1}{2} + \frac{|B| - |A|}{2\lambda} \tag{42}$$

Wir halten abschließend fest:

Satz 6. *Sei $P_{(a,b,c)}$ ein Polygon mit $a > 0$ und $b \geq ac$. Sind die optimalen Weglängen ω_0, ω_1 und ω_3 verschieden, dann ist*

$$KF_{(a,b,c)} = F_1(x^*, y_l(x^*)) = \frac{2\lambda y_l(x^*) - |B| + |A| - \lambda}{\omega_0 - \omega_1}$$

eine untere Schranke für das Erkundungsproblem, falls x^* eine Lösung der Gleichung

$$(\omega_2 - \omega_3)y_l(x) - (\omega_0 - \omega_1)y_r(x) = (2\omega_1 - \omega_0 - \omega_3)\,\frac{|B| - |A| + \lambda}{\lambda} \tag{43}$$

ist und $y_l(x)$ und $y_r(x)$ Werte in $[0, 1]$ haben, wobei y_l und y_r die Funktionen aus (27) und (38) sind.

Beweis. Im Falle $\omega_0 \neq \omega_1 = \omega_2 \neq \omega_3$ werden die Lösungsfunktionen y_l und y_r in (27) und (38) angegeben. Damit gilt $F_0(x, y_l(x)) = F_1(x, y_l(x))$ sowie $F_2(x, y_r(x)) = F_3(x, y_r(x))$ für alle $x \in [0, 1]$. Aus (28) und (39) folgt dann Gleichung (43) deren Lösung x^* auch $F_1(x^*, y_l(x^*)) = F_2(x^*, y_r(x^*))$ erfüllt, so dass $P_{(a,b,c)} = F_1(x^*, y_l(x^*))$ nach (16) eine untere Schranke des Erkundungsproblem ist. □

Es sei angemerkt, dass es vorkommen kann, dass die Funktionswerte von y_l ausserhalb des Intervals $[0, 1]$ liegen. Dies ist dann der Fall, wenn der Schnittpunkt von $F_0(x, -)$ und $F_1(x, -)$ nicht über dem Einheitsintervall liegt. In diesem Fall gilt $F_0(x, -) < F_1(x, -)$ oder $F_0(x, -) > F_1(x, -)$ auf $[0, 1]$. Die analoge Aussage gilt für y_r. Wir werden eine Beispiel sehen, bei dem $F_2(x, -)$ konstant und kleiner als $F_3(x, -)$ ist.

In konkreten Fällen kann man Gleichung 43 mit einem einfachen Newton Verfahren oder einem anderem numerischen Verfahren lösen, was wir im nächsten Abschnitt tun wollen.

Beispiel: Rechteck

Das einfachste Beispiel stellt das Rechteck dar mit den Parametern $(a, b, c) = (0, 1, 1)$ (siehe Abbildung 9). Die drei optimalen Weglängen haben die Werte $\omega_0 = 4$, $\omega_1 = \omega_2 = 1 + \sqrt{2} + \sqrt{5}$

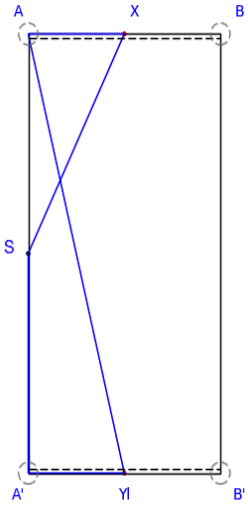


Abbildung 9: Ein Rechteck als entartetes Fünfeck

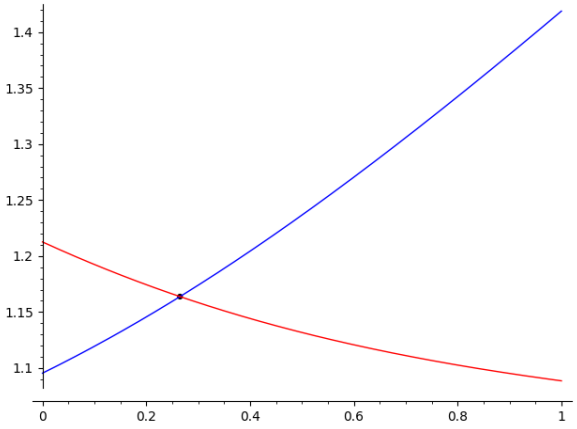


Abbildung 10: $F_0(x, y_l(x))$ (blau) und $F_3(x, y_r(x))$ (rot) im Fall des Rechtecks

und $\omega_3 = 2 + 2\sqrt{2}$. Die relativen Weglängen F_m sind durch folgende Funktionen gegeben:

$$\begin{aligned}
F_0(x,y) &= \frac{1}{4}\left(x + y + \sqrt{x^2+1} + \sqrt{y^2+4} + 1\right) \\
F_1(x,y) &= \frac{1}{\sqrt{5}+\sqrt{2}+1}\left(x - y + \sqrt{2} + \sqrt{x^2+1} + \sqrt{y^2+4} + 1\right) \\
F_2(x,y) &= \frac{-1}{\sqrt{5}+\sqrt{2}+1}\left(x - y - \sqrt{x^2+1} - \sqrt{(y+1)^2+4} - 2\right) \\
F_3(x,y) &= \frac{-1}{2+2\sqrt{2}}\left(x + y - \sqrt{2} - \sqrt{x^2+1} - \sqrt{(y+1)^2+4} - 2\right)
\end{aligned}$$

Zur besseren Lesbarkeit, werden wir die auftretenden Koeffizienten auf vier Stellen nach dem Komma runden. Die Lösungsfunktionen $y_l(x)$ und $y_r(x)$ sind somit wie folgt gegeben:

$$\begin{aligned}
y_l(x) &= 0.5821 - 0.0756(x + \sqrt{x^2+1}) - 0.0057\sqrt{\left(-x - \sqrt{x^2+1} + 7.6991\right)^2 + 703.8115} \\
y_r(x) &= 0.6561 + 0.0188(x - \sqrt{x^2+1}) - 0.0003\sqrt{\left(x - \sqrt{x^2+1} - 18.29127\right)^2 + 11320.2178}
\end{aligned}$$

Abbildung 10 zeigt die Graphen der Funktionen $x \mapsto F_0(x, y_l(x))$ (blau) und $x \mapsto F_3(x, y_r(x))$ (rot). Der Schnittpunkt liegt bei $x^* = 0.265024998658$, und es gilt ferner $y_l(x^*) = 0.328716242425$ und $y_r(x^*) = 0.603446136212$. Alle vier relativen Weglängen $F_m(x^*, y.(x^*))$ wurden mit Python berechnet und lieferten den selben Wert (Fehlerabweichung unter 10^{-15})

$$KF_{(0,1,1)} = F_0(x^*, y_l(x^*)) = 1.163774505531675,$$

der somit eine untere Schranke für den Kompetitivitätsfaktor darstellt.

Beispiel: Pentagon

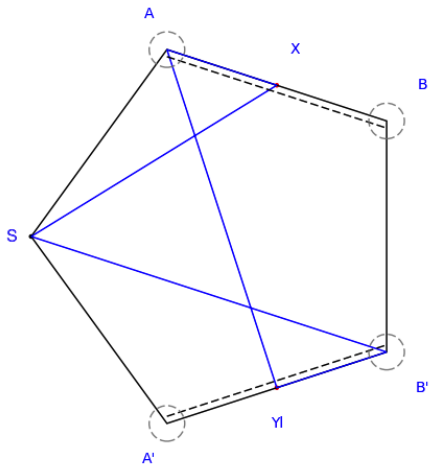


Abbildung 11: Das reguläre Fünfeck

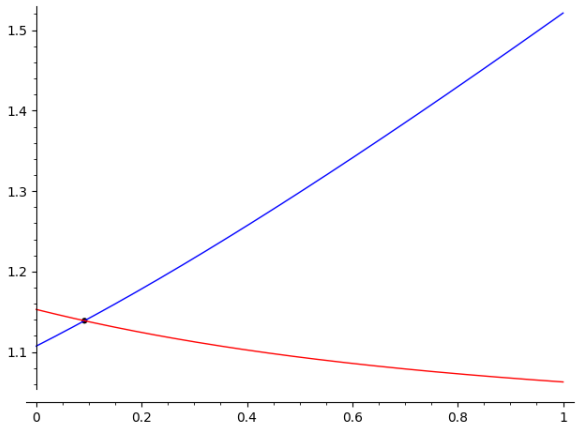


Abbildung 12: $F_0(x, y_l(x))$ (blau) und $F_3(x, y_r(x))$ (rot) für das reguläre Fünfeck.

Das reguläre Fünfeck (siehe Abbildung 11) kann durch folgende Parameter (a, b, c) erzeugt werden:

$$\begin{aligned} a &= \frac{1-\cos(2\pi/5)}{1-\cos(4\pi/5)} = \frac{5-\sqrt{5}}{5+\sqrt{5}} \cong 0.381966011250105 \\ b &= \frac{\sin(2\pi/5)}{1-\cos(4\pi/5)} = \frac{\sqrt{10+2\sqrt{5}}}{\sqrt{5}+5} \cong 0.525731112119134 \\ c &= \frac{\sin(4\pi/5)}{1-\cos(4\pi/5)} = \frac{\sqrt{10-2\sqrt{5}}}{\sqrt{5}+5} \cong 0.324919696232906 \end{aligned}$$

Die drei optimalen Weglängen haben die Werte

$$\omega_0 = 2.35114100916989 \qquad \text{und} \qquad \omega_2 = \omega_3 = 2.75276384094235.$$

Die relativen Weglängen $F_{\cdot,\cdot}$ sind:

$$\begin{aligned} F_0(x,y) &= 0.2764 + 0.2764(x+y) + 0.4253\left(\sqrt{0.4223(x^2+1)+0.261x} + \sqrt{0.4223(y^2-y)+1.1056}\right) \\ F_1(x,y) &= 0.6180 + 0.2361(x-y) + 0.3633\left(\sqrt{0.4223(x^2+1)+0.261x} + \sqrt{0.4223(y^2-y)+1.1056}\right) \\ F_2(x,y) &= 0.4721 - 0.2361(x-y) + 0.3633\left(\sqrt{0.4223(x^2+1)+0.261x} + \sqrt{0.4223y^2-1.1056(y-1)}\right) \\ F_3(x,y) &= 0.8541 - 0.2361(x+y) + 0.3633\left(\sqrt{0.4223(x^2+1)+0.261x} + \sqrt{0.4223y^2-1.1056(y-1)}\right) \end{aligned}$$

Das reguläre Fünfeck ist ein Polygon in dem die optimalen Längen ω_2 und ω_3 gleich sind. Für festes x gilt insbesondere $F_2(x, y) = F_3(x, y)$ genau dann, wenn $0.382 = 0.4721y$. Somit hat man die gerundete Lösung $y = 0.8091$ und die konstante Lösungsfunktion $y_r(x) = 0.8091$.

Die Lösung y_l der Gleichung $F_0(x, y_l(x)) = F_1(x, y_l(x))$ ist durch

$$\begin{aligned} y_l(x) &= 0.6677 - 0.0792x - 0.1218\sqrt{0.4223(x^2+1) + 0.261x} \\ &\quad -0.0062\sqrt{\left(2.118 - x - 1.5388\sqrt{0.4223(x^2+1) + 0.261x}\right)^2 + 380.0657} \end{aligned}$$

gegeben. Abbildung 12 zeigt die Graphen der Funktionen $x \mapsto F_0(x, y_l(x))$ (blau) und $x \mapsto F_3(x, y_r(x))$ (rot).

Der Schnittpunkt liegt bei $x^* = 0.091545477$ und Es gilt ferner $y_l(x^*) = 0.457043645$ und $y_r(x^*) = 0.808922165$. Alle vier relativen Weglängen $F_m(x^*, y_\cdot(x^*))$ wurden mit Python berechnet und lieferten denselben Wert (Fehlerabweichung unter 10^{-8})

$$KF_{(0.382,0.5257,0.325)} = F_0(x^*, y_l(x^*)) = 1.13875241,$$

der somit eine untere Schranke für den Kompetitivitätsfaktor darstellt.

Beispiel: Trapeze

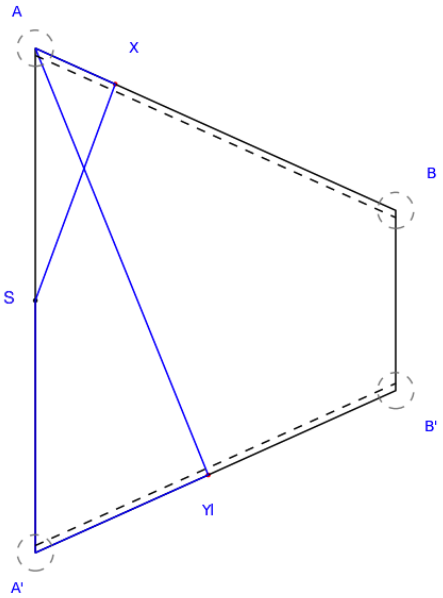


Abbildung 13: Ein Trapez

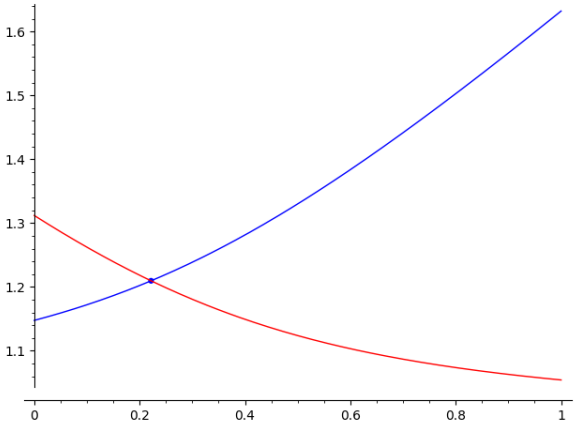


Abbildung 14: $F_0(x, y_l(x))$ (blau) und $F_3(x, y_r(x))$ (rot) für das Trapez.

Nach einigem Experimentieren hat sich gezeigt, dass Trapeze die besseren Kompetitivitätsfaktoren liefern. Betrachten wir zunächst das Trapez mit den Parametern $(a, b, c) = (0, 0.7, 0.25)$ (siehe Abbildung 13). Dieses hat die optimalen Weglängen $\omega_0 = 2.8$, $\omega_1 = \omega_2 = 3.11$ und $\omega_3 = 2.5615$. Die relativen Weglängen ergeben sich durch

$$
\begin{aligned}
F_0(x,y) &= 0.25 + 0.3916(x+y) + 0.3571\left(\sqrt{1.2025x^2 - 0.63x + 0.49} + \sqrt{1.2025y^2 - 1.26y + 1.96}\right)\\
F_1(x,y) &= 0.684 + 0.3526(x-y) + 0.3215\left(\sqrt{1.2025x^2 - 0.63x + 0.49} + \sqrt{1.2025y^2 - 1.26y + 1.96}\right)\\
F_2(x,y) &= 0.5776 - 0.3526(x-y) + 0.3215\left(\sqrt{1.2025x^2 - 0.63x + 0.49) + \sqrt{(1.2025(y-1)^2 - 0.45y + 0.7}\right)\\
F_3(x,y) &= 1.2586 - 0.4281(x+y) + 0.3903\left(\sqrt{1.2025x^2 - 0.63x + 0.49} + \sqrt{1.2025(y-1)^2 - 0.45y + 0.7}\right)
\end{aligned}
$$

Die Lösungen y_l und y_r sind wie folgt gegeben:

$$
\begin{aligned}
y_l(x) &= 0.5833 - 0.05261x - 0.048\sqrt{1.2025x^2 - 0.63x + 0.49}\\
&\quad -0.0027\sqrt{\left(1.1297 - x - 0.912\sqrt{1.2025x^2 - 0.63x + 0.49}\right)^2 + 491.029}\\
y_r(x) &= 0.5833 - 0.0526 * x - 0.04797\sqrt{(1.2025x^2 - 0.63x + 0.49)}\\
&\quad -0.0027\sqrt{\left(1.1297 - x - 0.912\sqrt{1.2025x^2 - 0.63x + 0.49}\right)^2 + 491.029}
\end{aligned}
$$

Abbildung 14 zeigt die Graphen der Funktionen $x \mapsto F_0(x, y_l(x))$ (blau) und $x \mapsto F_3(x, y_r(x))$ (rot). Der Schnittpunkt liegt bei $x^* = 0.22171898$, $y_l(x^*) = 0.47980461$, $y_r(x^*) = 0.95334344$ und der Kompetitivitätsfaktor errechnet sich durch

$$KF_{(0,0.7,0.25)} = F_0(x^*, y_l(x^*)) = 1.209555.$$

Mittels SageMath/Python habe ich die Berechnung des Kompetitivitätsfaktor automatisiert und ein Approximationsverfahren angewandt, um ein Polygon zu erhalten mit möglichst großem Kompetitivitätsfaktor. Ausgehend von einem Polygon mit Parameter (a_1, a_2, a_3) und einem Wert $\epsilon > 0$, werden in einer Iterationen des Verfahren die Kompetitivitätsfaktoren aller Polygone mit Parametern (a_1', a_2', a_3') berechnet für die $a_i = a_i'$ oder $|a_i - a_i'| = \epsilon$ gilt, d.h. es werden 27 Kompetitivitätsfaktoren für die Parameter im Nachbarschaftswürfel mit Seitenlänge 2ϵ und Zentrum (a_1, a_2, a_3) berechnet. Ein Parameter mit dem größten Kompetitivitätsfaktor wird ausgewählt und für die nächste Iteration als Parameter benutzt. Wenn der Parameter sich nicht ändert, wird der Wert ϵ durch den Faktor 10 geteilt. Das Verfahren wird dann solange fortgesetzt, bis der Faktor ϵ unter einem Wert, z.B. 10^{-6} fällt.

Es hat sich herausgestellt, dass ausgehend vom Parameter $(0.5, 1.2, 1)$, die "äußeren" Komponenten gegen Null streben und das Fünfeck zu einem Dreieck entartet. Nach hinreichenden Iterationen hat sich dabei der Parameter $(0, 0.7729444, 0)$ ergeben (siehe Abbildung 15). In diesem ausgearteten Fall fallen B und B' mit dem Punkt $(1, 0)$ zusammen. Daraus ergibt sich, dass die Funktion $F_2(x, y)$ konstant in y ist, denn dann ist $|BY| = \lambda(1 - y)$ und somit $W_2(x, y) = |X| + \lambda(1 - x) + |BY| + \lambda y + |A| = |X| - \lambda x + 2\lambda + |A|$. Insbesondere ist hier $F_2(x, y)$ stets kleiner als $F_3(x, y)$, was dazu führt, dass $y_r(x) = 1$ gewählt werden kann, da $F_3(x, 1)$ maximal ist.

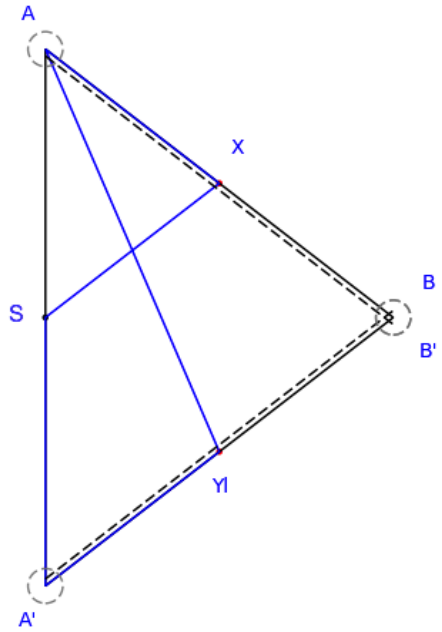


Abbildung 15: Optimales Dreieck

Für $x^* = 0.325880060$ und $y_l(x^*) = 0.61662497$, hat sich eine untere Schranke für den Kompetitivitätsfaktor von

$$KF_{(0,0.7729444,0)}) = 1.23329$$

ergeben. Explizit ergeben sich die relativen Weglängen:

$$\begin{aligned} F_0(0.325880060, 0.61662497) &= 1.2332961959428665 \\ F_1(0.325880060, 0.61662497) &= 1.233296196887088 \\ F_2(0.325880060, 1) &= 1.1536436711371543 \\ F_3(0.325880060, 1) &= 1.2332959561446062 \end{aligned}$$

3.2 Schwellen ohne versteckte Taschen

Obige Analyse betrachtet die relative Weglänge der Erkundungen von vier, vom Startpunkt aus gleich aussehenden, symmetrischen Polygonen. Nach der Landung auf einer Schwelle und dem Besuch einer der beiden Ecken kann man nicht mehr davon ausgehen, dass die Erkundung, die von der linken Ecke fortgesetzt wird, in der gleichen Situation ist, wie die Erkundung, die von der rechten Ecke aus fortgesetzt wird. Somit müssen die Fälle “Linke Ecke” und “Rechte Ecke” getrennt betrachten werden. Der Kompetitivitätsfaktor ist daher nicht

das Minimum des Maximums der vier Weglängen, d.h. die Lösung des Minimax-Problems

$$\min_{(x,y)\in[0,1]^2} \max\left(F_0(x,y), F_1(x,y), F_2(x,y), F_3(x,y)\right),$$

sondern die Lösung eines iterierten Minimax-Problems

$$KF = \min_{x\in[0,1]} \left(\max\left(\min_{y\in[0,1]} \max\left(F_0(x,y), F_1(x,y)\right), \min_{y\in[0,1]} \max\left(F_2(x,y), F_3(x,y)\right) \right) \right).$$

Man vergleiche dazu die Betrachtungen in (12), (13) und (15) auf Seite 14.

Als Variante dieser Analyse betrachten wir nun den Fall, das auf einer Schwelle auch keine versteckte Tasche an einer Ecke zu finden ist. Da die Ecken vorher nicht einsehbar sind, muss jede Erkundung die Schwellen mindestens einmal besuchen. Wegen der Symmetrie ist es daher irrelevant, welche Schwelle zuerst besucht wird. Anstatt der beiden Fälle "Linke Ecke" und Rechte Ecke" muss jetzt auch noch der Fall "Keine Ecke" betrachtet werden, so dass es nun 9 relative Weglängen zu betrachten gilt.

Wir ändern unsere Notation und bezeichnen mit f_{ij} eine der relativen Weglängen, mit $i, j \in \{1, 2, 3\}$ und der Vereinbarung, dass $i = 1$ für "Linke Ecke", $i = 2$ für "Rechte Ecke" und $i = 3$ für "Keine Ecke" steht.

Unsere Analyse versucht nun das iterierte Minimax-Problem

$$\min_{x\in[0,1]} \left(\max\left(G_1(x), G_2(x), G_3(x)\right)\right) \tag{44}$$

zu lösen, wobei wir für $i \in \{1, 2, 3\}$ die Funktionen G_i durch

$$G_i(x) = \min_{y\in[0,1]} \left(\max\left(f_{i1}(x,y), f_{i2}(x,y), f_{i3}(x,y)\right)\right) \tag{45}$$

definieren. Wegen $F_0 = f_{11}$, $F_1 = f_{12}$, $F_2 = f_{21}$ und $F_3 = f_{22}$ müssen nur die noch fehlenden relativen Weglängen f_{13}, f_{23}, f_{31}, f_{32}, f_{33} berechnet werden.

Im Gegensatz zum vorherigen Abschnitt, werden wir nicht mehr explizite Lösung des iterierten Minimax-Problems anstreben, sondern setzen numerische Optimierungsverfahren aus dem Softwarepaket Python/SciPy ein. Dabei werden die Minimax-Probleme als gleichwertige nichtlineare Optimierungsprobleme mit durch Ungleichungen gegebenen Nebenbedingungen wie folgt umformuliert:

Lemma 7. *Seien $g_1, \ldots, g_n : [0,1]^m \to \mathbb{R}$ Funktionen. Dann ist*

$$z = \min_{x\in[0,1]^m} \max\left(g_1(x), \ldots, g_n(x)\right) \tag{46}$$

genau dann eine Lösung des Minimax-Problems, wenn z Lösung des Optimierungsproblems

$$\min_{(x,z)\in\mathbb{R}^{m+1}} z \tag{47}$$

mit den Nebenbedingungen $x \in [0,1]^m$ und $z \geq g_i(x)$, für alle $1 \leq i \leq n$, ist.

Beweis. Sei $\pi : \mathbb{R}^{m+1} \to \mathbb{R}$ die Projektion auf die letzte Komponente und

$$D = \{(x, z) \in \mathbb{R}^{m+1} \mid x \in [0,1]^m \text{ und } z \geq g_i(x), \quad \forall 1 \leq i \leq n\}$$

die Menge der Punkte (x, z), die die Nebenbedingung des Optimierungsproblem (47) erfüllen. Ferner betrachten wir $\varphi : [0,1]^n \to D$ mit $\varphi(x) = (x, \max(g_1(x), \ldots, g_n(x)))$. Dann gilt für alle $(x, z) \in D$, wegen $z \geq g_i(x)$, für alle i, auch

$$z = \pi((x, z)) \geq \max(g_1(x), \ldots, g_n(x)) = \pi(\varphi(x)).$$

Somit erhalten wir

$$\min_{x \in [0,1]^m} \max(g_1(x), \ldots, g_n(x)) = \min_{x \in [0,1]^m} \pi(\varphi(x)) = \min_{(x,z) \in D} z.$$

□

Bestimmung der relativen Weglängen

Die relativen Weglängen $f_{ij}(x, y)$ mit $1 \leq i, j \leq 2$ sind die oben angegebenen berechneten Funktionen $F_0(x, y), F_1(x, y), F_2(x, y), F_3(x, y)$. Mit $W_{ij}(x, y)$ seien die absoluten Weglängen einer Erkundung mit Landepunkten X und Y (bzw. $x, y \in [0, 1]$) der Fälle (i, j) bezeichnet, wobei das Paar (i, j) für linke(1)/rechte(2)/keine(3) (obere, untere) Ecke steht, also z.B. W_{13} steht für die Weglänge einer Erkundung, die auf der oberen Schwelle zur linken Ecke geht und auf der unteren Schwelle keine Ecke besucht. Dann gilt:

$$W_{13} = |X| + \lambda x + |AY| + |Y| \tag{48}$$

$$W_{23} = |X| + \lambda(1 - x) + |BY| + |Y| \tag{49}$$

$$W_{31} = |X| + |XY| + \lambda y + |A| \tag{50}$$

$$W_{32} = |X| + |XY| + \lambda(1 - y) + |B| \tag{51}$$

$$W_{33} = |X| + |XY| + |Y| \tag{52}$$

Hierbei ist wieder $\lambda = |AB|$. Abbildung 16 zeigt die fünf zusätzlichen Polygone.

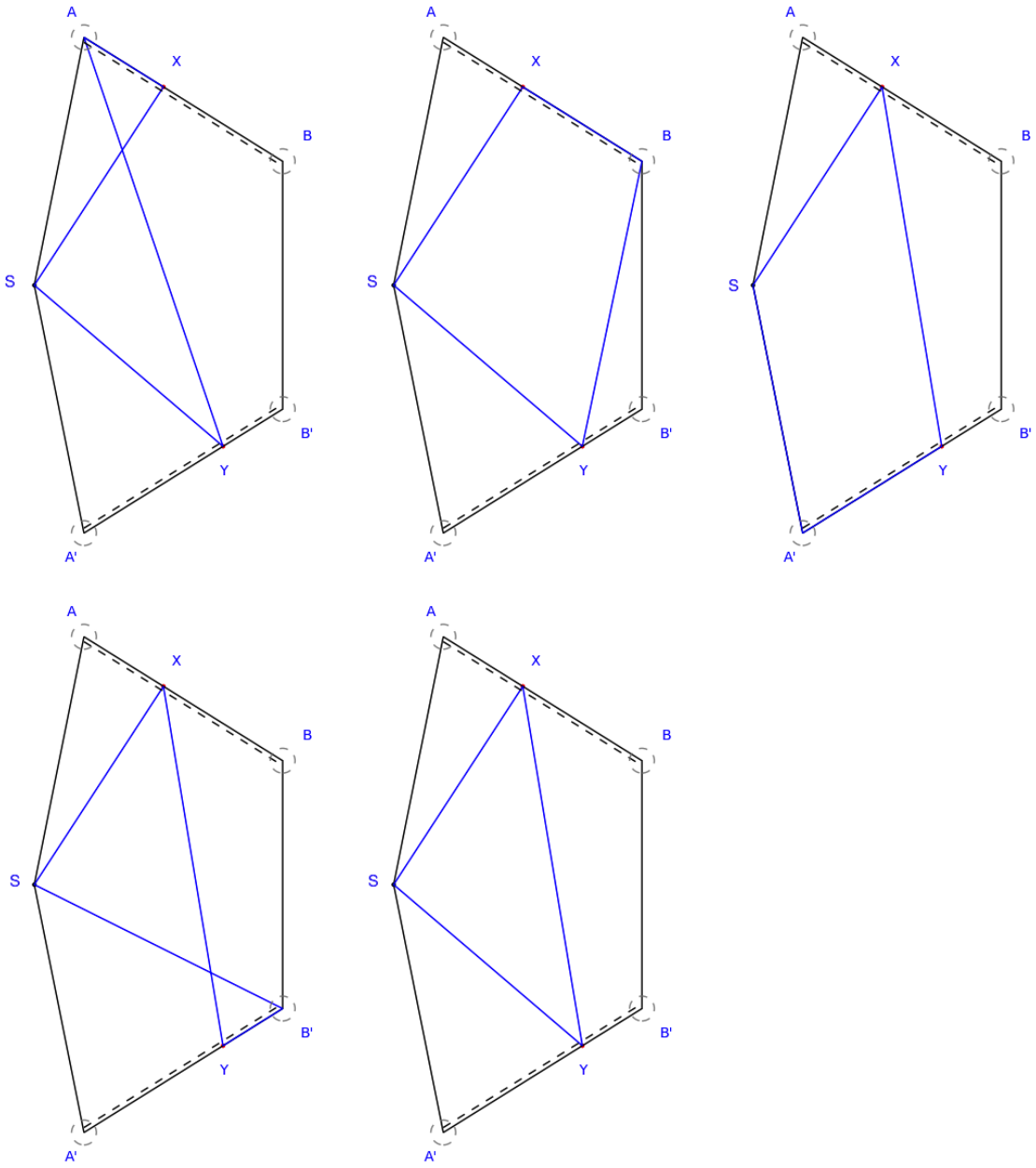


Abbildung 16: Fünf zusätzliche mögliche Polygone.

Mit ω_{ij} seien die optimale Weglänge für den Fall (i, j) bezeichnet. Die optimale Weglänge ω_{13} ist die Länge des Weges $SAYS$ für einen Punkt Y auf $\overline{A'B'}$, so dass AYS minimal ist. Wir müssen also den kürzesten Weg von A nach S suchen, der über das Segment $\overline{A'B'}$ geht. Der Punkt Y ist dabei der Schnittpunkt der durch S und dem an der durch $\overline{A'B'}$ verlaufenden Geraden gespiegelten Punktes A, sagen wir $\tilde{A}$. Insbesondere ist die Weglänge AYS dann gleich dem Abstand von S und $\tilde{A}$. Allgemein gilt, dass der kürzeste Weg von einem Punkt R zu einem Punkt S über ein Segment $\overline{PQ}$ durch den Abstand von S zum gespiegelten Punkt $\tilde{R}$ gegeben ist (siehe Abbildung 17). Ist v ein zu dem Verbindungsvektor $Q - P$ orthogonaler Vektor, dann gilt $\tilde{R} = R + 2\frac{(Q-R)\cdot v}{||v||^2}v$.

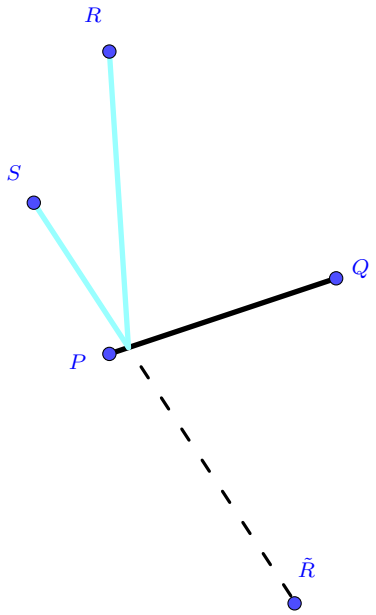


Abbildung 17: Kürzeste Weg von R nach S über PQ

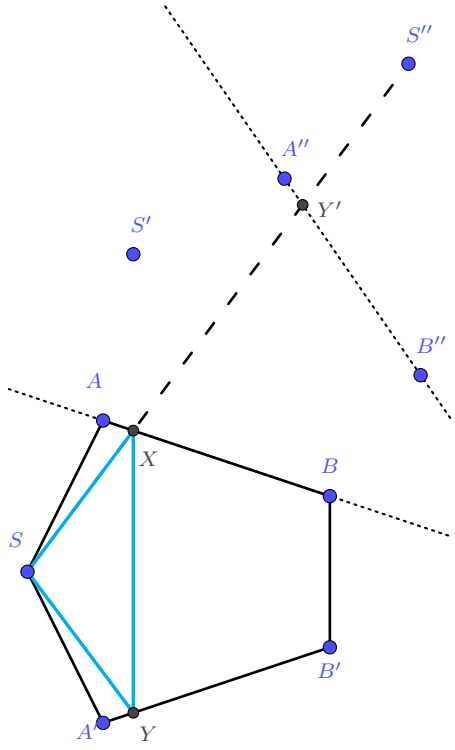


Abbildung 18: Kürzeste Weg über zwei Seiten

In unserem konkreten Fall ist $R = A$, $P = A'$, $Q = B'$, $v = (b - c, a - 1)$ und $B' - A = (1 - a, -b - c)$. Aus $(B' - A) \cdot v = 2b(1 - a)$ und $||v||^2 = \lambda^2$ erhalten wir:

$$\tilde{A} = A + 4b(1 - a)\lambda^{-2}v = (a + 4\lambda^{-2}b(1 - a)(b - c), b - 4\lambda^{-2}b(1 - a)^2) \tag{53}$$

Damit ist die optimale Weglänge im Fall $(1, 3)$ durch $\omega_{13} = |A| + |\tilde{A}S|$ gegeben. Analog berechnet man ebenso $\omega_{23} = |B| + |\tilde{B}S|$, wobei $\tilde{B}$ der an $\overline{A'B'}$ gespiegelter Punkt B ist. Aus

Symmetriegründen gilt $\omega_{31} = \omega_{13}$ und $\omega_{32} = \omega_{23}$. Bleibt nur noch ω_{33} zu berechnen, bei dem es einen kürzesten Weg zu finden gilt, der beide Schwellen $\overline{AB}$ und $\overline{A'B'}$ besucht und zum Startpunkt zurückkehrt (siehe das letzte Polygon in Abbildung 16). In diesem Fall müssen die Punkte S, A' und B' am Segment $\overline{AB}$ gespiegelt werden, so dass man Punkte S', A'' und B'' erhält. Danach wird S' am Segment $\overline{A''B''}$ gespiegelt und man erhält einen Punkt S'' (siehe Abbildung 18). Der Abstand von S zu S'' stellt dann den kürzesten Rundgang dar, der beide Schwellen besucht, d.h. $\omega_{33} = |SS''|$.

Bestimmung einer unteren Schranke für einen Kompetitivitätsfaktor

Die Bestimmung des Kompetitivitätsfaktors geschieht durch die Funktion *minimize* des Softwarepakets SciPy (siehe [14]). Für ein gegebenes Polygon $P_{(a,b,c)}$ werden zuerst die optimalen Wege wie oben beschrieben bestimmt. Um das iterierte Minimax-Problem zu lösen benutzen wir zweimal Lemma 7 und rufen die SciPy-Funktion *minimize* rekursiv auf. Dabei wird *minimize* verwendet, um die Projektion $(x, z) \to z$ zu minimieren unter der Nebenbedingung das $z - G_i(x) \geq 0$ und $x \in [0, 1]$ (siehe (45) für die Definition der Funktionen G_i).[3] Die Berechnung der Funktionen $G_i(x)$ geschieht in einer Unterfunktion, die das Minimax-Problem für die Funktionen f_{i1}, f_{i2}, f_{i3} wiederum mit *minimize* löst.

Die berechneten Kompetitivitätsfaktoren sind etwas besser als im vorherigen Abschnitt und es hat sich wiederum gezeigt, dass man bei einem zu einem Dreieck $P_{(0,b,0)}$ degradierten Fünfeck die besten unteren Schranken erhält. Hier fallen B und B' mit $(1, 0)$ zusammen. Für die optimalen Weglängen heißt dies: $\omega_{11} = 4|A| = 4b$. Da A' mit dem an $\overline{A'B}$ gespiegelte Punkt A zusammenfällt, gilt $\omega_{13} = \omega_{31} = |A| + |AA'| + |A| = 4b$. Ferner, da $B = B' = (1, 0)$ haben wir $\omega_{22} = 2|B| = 2$, und da B mit seinem Spiegelpunkt zusammenfällt, gilt ebenso $\omega_{23} = \omega_{32} = 2$. Desweiteren haben wir $\omega_{12} = \omega_{21} = |A| + |AB| + |B| = b + 1 + \sqrt{1 + b^2}$. Um ω_{33} zu ermitteln spiegeln wir $S = (0, 0)$ und $A' = (0, -b)$ am Segment $\overline{AB}$ und erhalten $S' = \frac{1}{b^2+1}(2b^2, 2b)$ und $A'' = \frac{1}{b^2+1}(4b^2, 3b - b^3)$. Abschließend müssen wir S' noch am Segment $\overline{A''B}$ spiegeln, was uns $S'' = \frac{4b}{(b^2+1)^2}(2b, 1 - b^2)$ liefert. Der Abstand zum Ursprung ist somit $\omega_{33} = \frac{4b}{(b^2+1)}$. Die Weglängen für Polygone $P_{(0,b,0)}$ können auf elementare Weise in Python

[3] Als Optimierungsmethode stehen uns verschiedene Verfahren zu Verfügung und meistens wurde der *Constrained optimization by linear approximation (COBYLA)* Verfahren verwendet, bei dem die Gradienten nicht bekannt sein müssen.

implementiert werden und lauten:

$$\begin{aligned}
f_{11}(x,y) &= (|X| + \lambda x + |AY| + \lambda y + b)/\omega_{11} \\
f_{12}(x,y) &= (|X| + \lambda x + |AY| + \lambda(1-y) + 1)/\omega_{12} \\
f_{13}(x,y) &= (|X| + \lambda x + |AY| + |Y|)/\omega_{13} \\
f_{21}(x,y) &= (|X| + \lambda(1-x) + \lambda + b)/\omega_{21} \\
f_{22}(x,y) &= (|X| + \lambda(1-x) + 2\lambda(1-y) + 1)/\omega_{22} \\
f_{23}(x,y) &= (|X| + \lambda(1-x) + \lambda(1-y) + |Y|)/\omega_{23} \\
f_{31}(x,y) &= (|X| + |XY| + \lambda y + b)/\omega_{31} \\
f_{32}(x,y) &= (|X| + |XY| + \lambda(1-y) + 1)/\omega_{32} \\
f_{33}(x,y) &= (|X| + |XY| + |Y|)/\omega_{33}
\end{aligned}$$

mit den Abständen $\lambda = |AB| = \sqrt{1+b^2}$, $|X| = \sqrt{x^2 + (b(1-x))^2}$, $|AY| = \sqrt{y^2 + (b(2-y))^2}$, $|Y| = \sqrt{y^2 + (b(y-1))^2}$ und $|XY| = \sqrt{(y-x)^2 + (b(x+y-2))^2}$.

Für das Polygon $P_{(0,0.4655055,0)}$ erhält man 1.2497235909098847 als den Wert des iterierten Minimax-Problems (44) , d.h.

$$KF_{(0,0.4655055,0)} = 1.2497235909098847.$$

Als Nachweis wurden die Funktionen f_{ij} in Python implementiert und es ergab sich der Grenzwert $x^* = 0.1971722$ für die obere Schwelle als Stelle an der

$$\min_{x\in[0,1]} \max\{G_1(x), G_2(x), G_3(x)\}$$

erfüllt wird. Dabei wird die Lösung des ersten inneren Minimax-Problems

$$G_1(x^*) = \min_{y\in[0,1]} \left(\max\left(f_{i1}(x,y), f_{i2}(x,y), f_{i3}(x,y)\right)\right)$$

bei $y_l = 0.34204389$ erreicht. Die Lösung des inneren Minimax-Problems $G_2(x^*)$ wird bei $y_r = 1$ erreicht und die Lösung des letzten inneren Teilproblems $G_3(x^*)$ findet sich bei bei $y_k = 0.66411935$. Die nachstehende Tabelle listet die 9 relativen Weglängen auf:

$f_{11}(x^*, y_l) = 1.249723621420$	$f_{21}(x^*, 1) = 1.119949350516$	$f_{31}(x^*, y_k) = 1.249723590309$
$f_{12}(x^*, y_l) = 1.249723621465$	$f_{22}(x^*, 1) = 1.154047631381$	$f_{32}(x^*, y_k) = 1.249723590909$
$f_{13}(x^*, y_l) = 1.043678451016$	$f_{23}(x^*, 1) = 1.154047631381$	$f_{33}(x^*, y_k) = 1.183509213527$

Anmerkung

Eine weitere Variante wäre, auch solche Polygone zu betrachten, die an beiden Seiten einer Schwelle versteckte Taschen haben könnten. Dies führt zur Betrachtung von 16 Polygonen.

Bei der Analyse dieser 16 Polygone müsste man nun auch unterscheiden, ob ein Erkundungsrundgang im Fall von zwei verstecken Taschen an einer Schwelle zuerst die linke und dann die rechte oder zuerst die rechte und dann die linke Ecke besucht. Dadurch entstehen 25 mögliche Familien von Erkundungen (abhängig von den Landepunkten x und y auf den Schwellen) die mit 16 optimalen Wegen verglichen werden müssen. Durch Hinzufügen der jeweiligen Weglängenfunktionen $f_{??}$ kann man den vorherigen Ansatz leicht abändern. Dadurch hat sich jedoch keine Verbesserung der Kompetitivitätsfaktoren ergeben und die genaue Analyse führen wir hier nicht weiter aus.

3.3 Schwellen mit eingeschränkter Sicht

In diesem Abschnitt wollen wir auch Parameter $a < 0$ bei unseren Polygonen zulassen. Dabei wird die Sicht zwischen den Schwellen $\overline{AB}$ und $\overline{A'B'}$ durch den vorspringenden Startpunkt S behindert.

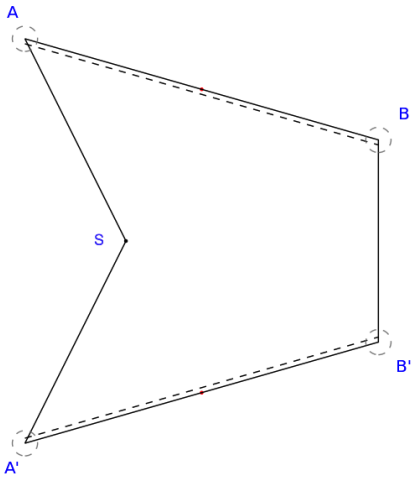


Abbildung 19: Fünfeck mit $a < 0$

Vom Ursprung S aus können dann immer noch alle Punkte auf den Schwellen ungehindert angesteuert werden. Andererseits sind einige Punkte der Schwelle $\overline{A'B'}$ von $\overline{A'B'}$ nicht mehr sichtbar. Wie in (8) erklärt, müssen die Wege von A bzw. B zum Punkt Y auf $\overline{A'B'}$ gegebenenfalls über S geführt werden. Der direkte Weg von einem Punkt X auf $\overline{AB}$ nach Y, der ganz im Polygon verläuft, ist genau dann das Verbindungssegment XY, wenn der Ursprung S rechts von der durch X nach Y verlaufenden Geraden liegt. Andererseits, wenn S links von dieser Geraden liegt, besteht der direkte Weg aus den Segmenten XS und SY. Bei der Berechnung der relativen Weglänge muss also die Bestimmung des Abstandes zwischen X und Y entsprechend abgeändert werden. Die Entscheidung, ob ein Punkt R links von einer durch zwei Punkte P und Q verlaufenden Geraden liegt, wird durch das Kreuzprodukt bestimmt: Seien $P = (p_1, p_2)$, $Q = (q_1, q_2)$ und $R = (r_1, r_2)$. Am Vorzeichen der dritten Komponente des Kreuzprodukt $\vec{u} \times \vec{v}$ der Vektoren $\vec{u} = (q_1 - p_1, q_2 - p_2, 0)^t$ und $\vec{v} = (r_1 - p_1, r_2 - p_2, 0)^t$ in $\mathbb{R}^3$ kann man erkennen, ob R links von der Geraden durch P und Q liegt. Es gilt

$$\vec{u} \times \vec{v} = (0, 0, (q_1 - p_1)(r_2 - p_2) - (r_1 - p_1)(q_2 - p_2))^t. \tag{54}$$

Somit liegt R genau dann links von der Geraden von P nach Q, wenn

$$(q_1 - p_1)(r_2 - p_2) - (r_1 - p_1)(q_2 - p_2) > 0. \tag{55}$$

Die benötigten Änderungen des Python-Programmcodes im Vergleich zur vorherigen Lösung bestehen nun darin, bei der Berechnung des Abstandes eines Punktes auf $\overline{AB}$ und einem Punkt auf $\overline{A'B'}$ abzuprüfen, ob der Startpunkt S links oder rechts vom Verbindungssegment liegt. Dazu wurde eine Funktion *IstLinks(P,Q,R)* eingefügt, die *True* liefert genau dann, wenn R links von der Geraden von P zu Q liegt. Ferner wurde die Funktion *dist(P,Q)* definiert, die unter Benutzung der Funktion *IstLinks* die Weglänge von P auf $\overline{AB}$ zu Q auf $\overline{A'B'}$ berechnet.

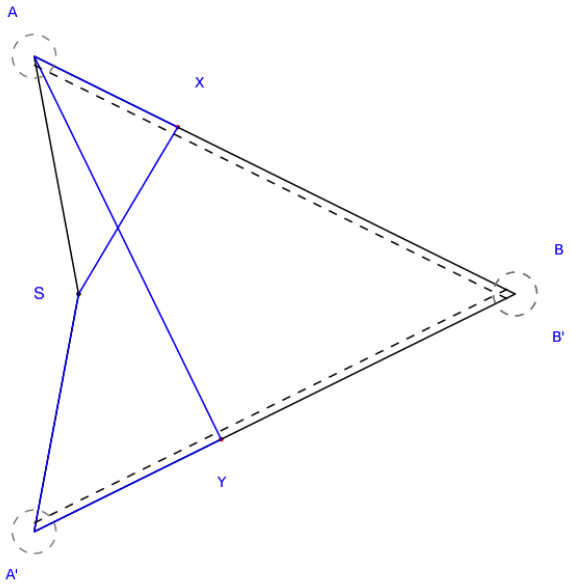
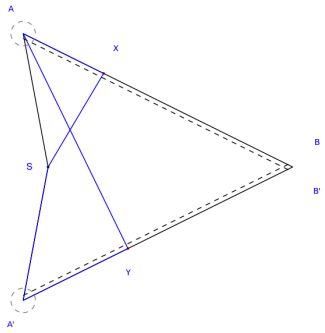


Abbildung 20: Fünfeck mit bestem Kompetitivitätsfaktor, wobei der eingezeichnete Weg dem Fall entspricht, bei dem beide versteckten Taschen sich links befinden. Hier $x = 0.29845943$ und $y = 0.38867476$ sind die Entscheidungspunkte auf der oberen bzw. unteren Schwelle.

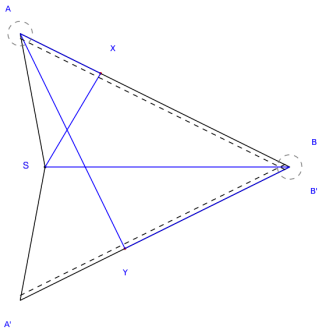
Die Berechnung einer unteren Schranke eines Kompetitivitätsfaktors geschieht wie zuvor durch Lösen eines iterierten Minimax-Problems. Nach einigem Experimentieren hat sich für das Polygon $P_{(a,b,0)}$ aus Abbildung (20) mit $a = -0.101871739$ und $b = 0.545301534$ folgende untere Schranke des Kompetitivitätsfaktors ergeben:

$$KF_{(-0.101871,0.545301,0)} = 1.2717534953305767.$$

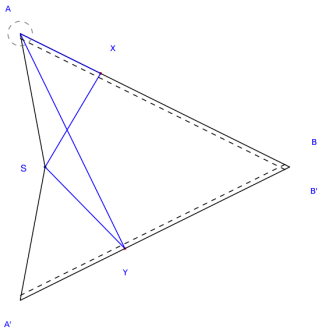
Hierbei löst $x = 0.298492176147$ das äußere Minimax-Problem $\min_x \max\{G_1(x), G_2(x), G_3(x)\}$, während die inneren Minimax-Probleme bei $y_l = 0.388749155$, $y_r = 1$ und $y_k = 0.672650946$ ihre Lösungen haben.



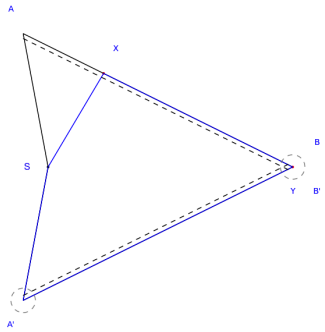
$F_{11} = 1.2717536919406138$



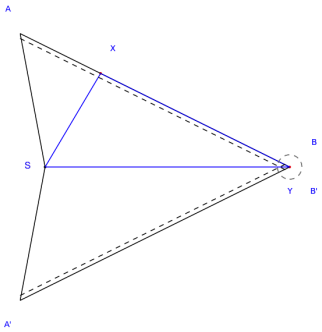
$F_{12} = 1.2717536402273737$



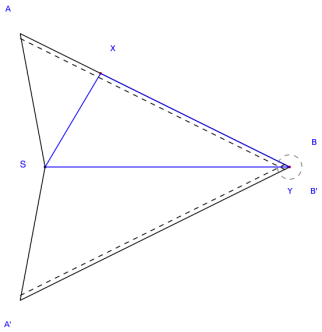
$F_{13} = 1.1371980640901642$



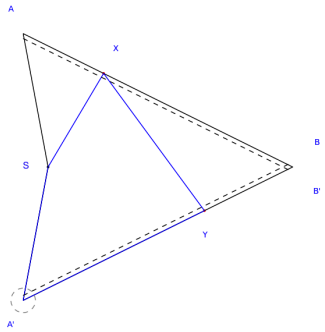
$F_{21} = 1.1103663910916470$



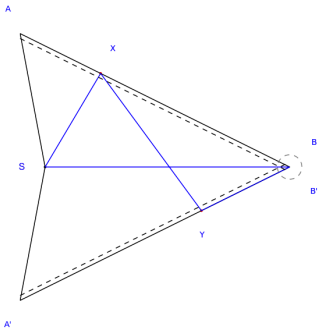
$F_{22} = 1.1536386323715513$



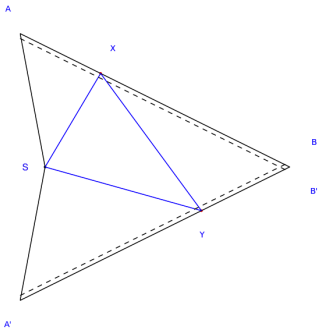
$F_{23} = 1.1536386332979884$



$F_{31} = 1.2717534194165379$



$F_{32} = 1.2717534193632134$



$F_{33} = 1.1350253432430895$

Abbildung 21: Degenerierter Fall mit $|B - B'| = 10^{-8}$.

4 Polygone mit drei Schwellen

In diesem letzten Kapitel setzen wir die Suche nach einer größtmöglichen unteren Schranke für den Kompetitivitätsfaktor des Erkundungsproblems fort, indem wir die Anzahl der Schwellen auf drei erhöhen. Allerdings ändern wir das Verhalten der Schwellen und fordern nur, dass es auf den ersten zwei Schwellen höchstens an der linken Seite (von der Erkundung aus gesehen) eine Tasche geben könnte, während sich an genau einer der beiden Seiten der letzten Schwelle eine Tasche befindet.

4.1 Die Grundidee

Hagius et al. haben in [11] ein Polygon wie in Abbildung 22 betrachtet, das fast wie ein gleichseitiges Dreieck aussieht. Allerdings besteht die rechte untere Ecke aus zwei nahe beieinander liegenden Ecken, sagen wir A_1 und A_2, von denen es weitere Kanten zu einem fünften Punkt S im Dreiecksinneren gibt. Insgesamt sind bis zu drei Taschen an den Ecken A_1, A_2, B, C

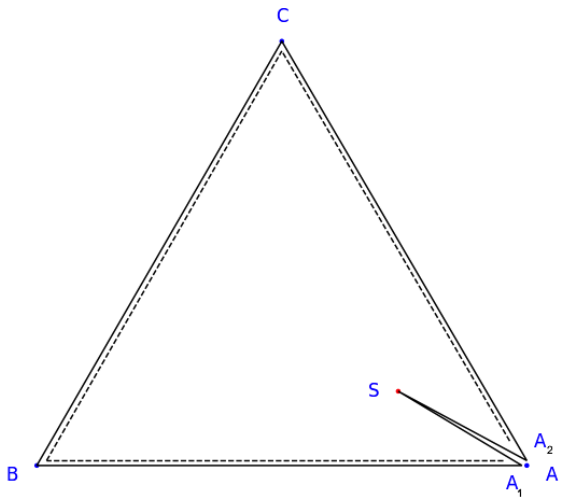


Abbildung 22: Ein Polygon, das einem gleichseitigem Dreieck ähnlich ist.

versteckt.

- In A_1 kann sich eine Tasche befinden, was nur bei Erkundung von $\overline{A_1B}$ sichtbar wird.
- In B kann sich eine Tasche befinden, was nur bei Erkundung von $\overline{BC}$ sichtbar wird.
- In genau einer Ecke C oder A_2 befindet sich eine Tasche, was nur auf $\overline{CA_2}$ erkennbar ist.

Die Seiten $\overline{A_1B}$, $\overline{BC}$ und $\overline{CA_2}$ müssen daher mindestens einmal besucht werden, um alle Taschen zu entdecken. Die acht möglichen Szenarien, d.h. die acht möglichen Polygone, werden durch ein Element $(e_1, e_2, e_3) \in \{0,1\}^3$ parametrisiert und mit $P_{e_1e_2e_3}$ bezeichnet, wobei

- $e_1 = 1$ genau dann, wenn sich in A_1 eine Tasche befindet.
- $e_2 = 1$ genau dann, wenn sich in B eine Tasche befindet.
- $e_3 = 0$ genau dann, wenn sich in C eine Tasche befindet und in A_2 nicht.
- $e_3 = 1$ genau dann, wenn sich in A_2 eine Tasche befindet und in C nicht.

Wir machen folgende Annahmen:

1. Der Abstand zwischen A_1 und A_2 wird ignoriert, und beide Punkte werden zu einem Punkt A zusammengefasst, so dass man sich das Segment $\overline{SA}$ als Wand vorstellen kann.
2. Jede Erkundung startet im Punkt S.
3. S liegt auf der Winkelhalbierenden des Winkels $\measuredangle(C, A, B)$.
4. $\overline{AB}$ und $\overline{AC}$ sind gleich lang.
5. Die Schwellen werden in der Reihenfolge $\overline{AB}$, $\overline{BC}$, $\overline{CA}$ besucht.

Durch die Bedingungen 3. und 4. sieht das zu betrachtende Polygon vom Standpunkt S völlig symmetrisch aus.

Unter Benutzung von Cabri Geometry haben Hagius et al. experimentell gezeigt, dass der Kompetitivitätsfaktor mindestens 1.2825 sein muss (siehe [11]). Ziel dieses Kapitels ist diese untere Schranke numerisch zu belegen (und sogar leicht zu verbessern).

Da ein Erkundungsrundgang jede Schwelle einmal besuchen muss, sind alle Erkundungen eindeutig durch ein Element $(x, y, z) \in [0,1]^3$ bestimmt, welches die Landepunkte auf den drei Schwellen wie folgt angibt:

$$X = A + x(B - A), \qquad Y = B + y(C - B), \qquad Z = C + z(A - C).$$

Die absoluten Weglängen der Erkundungen der Polygone $P_{e_1e_2e_3}$ in Abhängigkeit der

Landepunkte X, Y, Z sind wie folgt gegeben:

$$\begin{aligned}
W_{000} &= |SX| + \mathrm{dist}(X,Y) + \mathrm{dist}(Y,Z) + |ZC| + |CS| \\
W_{001} &= |SX| + \mathrm{dist}(X,Y) + \mathrm{dist}(Y,Z) + |ZA| + |AS| \\
W_{010} &= |SX| + \mathrm{dist}(X,Y) + |YB| + \mathrm{dist}(B,Z) + |ZC| + |CS| \\
W_{011} &= |SX| + \mathrm{dist}(X,Y) + |YB| + \mathrm{dist}(B,Z) + |ZA| + |AS| \\
W_{100} &= |SX| + |XA| + \mathrm{dist}(A,Y) + \mathrm{dist}(Y,Z) + |ZC| + |CS| \\
W_{101} &= |SX| + |XA| + \mathrm{dist}(A,Y) + \mathrm{dist}(Y,Z) + |ZA| + |AS| \\
W_{110} &= |SX| + |XA| + \mathrm{dist}(A,Y) + |YB| + \mathrm{dist}(B,Z) + |ZC| + |CS| \\
W_{111} &= |SX| + |XA| + \mathrm{dist}(A,Y) + |YB| + \mathrm{dist}(B,Z) + |ZA| + |AS|
\end{aligned}$$

Hierbei bezeichne $|PQ|$ den (direkten) euklidischen Abstand von P nach Q und $\mathrm{dist}(P,Q)$ den kürzesten Weg von P nach Q im Inneren des zu betrachtenden Polygons. Da wir voraussetzen, dass eine Erkundung im Uhrzeigersinn stattfindet, ist der Abstand $\mathrm{dist}(P,Q)$ von einem Punkt P zu einem Punkt Q auf der nachfolgenden Seite genau dann gleich $|PQ|$, wenn der Punkt S rechts vom Strahl $\overrightarrow{PQ}$ liegt. Ferner gilt $\mathrm{dist}(P,Q) = |PS| + |SQ|$ genau dann, wenn S links vom Strahl $\overrightarrow{PQ}$ liegt.[4] Wenn ein Erkundungsrundgang den Punkt X auf der ersten Schwelle erreicht, wird erkennbar, ob man sich in einem Polygon vom Typ $P_{0e_2e_3}$ oder vom Typ $P_{1e_2e_3}$ befindet. Der Rundgang wird dann entweder direkt oder über den Punkt A zum Punkt Y fortgesetzt. Dort wird erkennbar, ob man sich in einer der vier Klassen von Polygonen P_{00e_3}, P_{01e_3} oder P_{10e_3}, P_{11e_3} befindet, bevor dann der Weg entweder direkt oder über den Punkt B zum Punkt Z fortgesetzt wird. Abbildung 23 zeigt die möglichen Wege in den acht Polygonen.

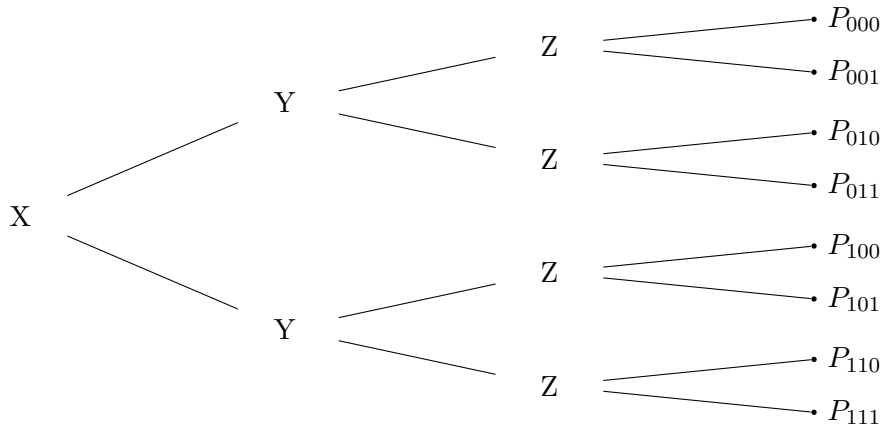


Abbildung 23: Entscheidungswege zu den acht Polygonen

[4] Wenn S auf dem Strahl von P nach Q liegt, dann sind beide Ausdrücke gleich.

Sei $\omega_{e_1e_2e_3}$ die optimale Weglänge eines Erkundungsrundganges im Polygon $P_{e_1e_2e_3}$. Die relative Weglänge sei definiert durch $F_{e_1e_2e_3} := W_{e_1e_2e_3}/\omega_{e_1e_2e_3}$. Um den Kompetitivitätsfaktor zu ermitteln, muss wieder ein iteriertes Minimax-Problem gelöst werden. Für alle $a, b \in \{0,1\}^2$ definieren wir folgende Funktionen:

$$G_{ab}(x,y) := \min_{z\in[0,1]} \max\left(F_{ab0}(x,y,z), F_{ab1}(x,y,z)\right) \tag{56}$$

$$H_a(x) := \min_{y\in[0,1]} \max\left(G_{a0}(x,y), G_{a1}(x,y)\right) \tag{57}$$

Das zu lösende Minimax-Problem ist dann

$$\begin{aligned}
KF &= \min_{x\in[0,1]} \max\left(H_0(x), H_1(x)\right) \\
&= \min_{x\in[0,1]} \max\left(\min_{y\in[0,1]} \max\left(G_{00}(x,y), G_{01}(x,y)\right), \min_{y\in[0,1]} \max\left(G_{10}(x,y), G_{11}(x,y)\right)\right) \\
&= \min_{x\in[0,1]} \max\Bigg(\\
&\quad \min_{y\in[0,1]} \max\left(\min_{z\in[0,1]} \max\left(F_{000}(x,y,z), F_{001}(x,y,z)\right), \min_{z\in[0,1]} \max\left(F_{010}(x,y,z), F_{011}(x,y,z)\right)\right), \\
&\quad \min_{y\in[0,1]} \max\left(\min_{z\in[0,1]} \max\left(F_{100}(x,y,z), F_{101}(x,y,z)\right), \min_{z\in[0,1]} \max\left(F_{110}(x,y,z), F_{111}(x,y,z)\right)\right) \\
&\quad \Bigg)
\end{aligned}$$

Optimale Weglängen

Die optimalen Weglängen ergeben sich wie folgt. Dabei sei angemerkt, dass A_2 von B nicht direkt angesteuert werden kann und der kürzeste Weg daher über S verläuft:

Weglänge	Taschen in	kürzester Rundweg
$\omega_{111} = 3\|SA\| + \|AB\| + \|SB\|$	A_1, B und A_2	$SABSAS$
$\omega_{110} = \|SA\| + \|AB\| + \|BC\| + \|CS\|$	A_1, B und C	$SABCS$
$\omega_{101} = 2\|SA\| + \|AA'\|$	A_1 und A_2	siehe unten
$\omega_{100} = 2\|SA\| + 2\|SC\|$	A_1 und C	$SASCS$
$\omega_{011} = 2\|SB\| + 2\|SA\|$	B und A_2	$SBSAS$
$\omega_{010} = \|SB\| + \|BC\| + \|CS\|$	B und C	$SBCS$
$\omega_{001} = \|SS''\| + 2\|AS\|$	A_2	siehe unten
$\omega_{000} = \|SC'\| + \|SC\|$	C	siehe unten

Im Falle $(1,0,1)$, bei dem sich Taschen in A_1 und A_2 befinden, muss zuerst von S nach A_1 gegangen und dann der kürzeste Weg von A_1 nach A_2 über die Schwelle $\overline{BC}$ gefunden werden. Wie auf Seite 27 erklärt, muss dazu A an $\overline{BC}$ gespiegelt werden, so dass wir einen Punkt A'

erhalten. Die Weglänge von A_1 nach A_2 ist daher durch den Abstand $|AA'|$ gegeben. Von A_2 muss dann wieder nach S gegangen werden. Wir erhalten daher die optimale Weglänge $\omega_{101} = 2|SA| + |AA'|$.

Im Falle $(0, 0, 1)$ gibt es nur eine versteckte Tasche, nämlich in A_2. Somit muss man einen Weg von S nach A_2 finden, der über die Schwellen $\overline{AB}$ und $\overline{BC}$ führt. Da S auf der Winkelhalbierenden liegt, reicht es einen kürzesten Weg von S über $\overline{AB}$ und $\overline{BC}$ zu finden und dann zweimal den Abstand $|SA|$ zu berechnen. Wie auf Seite 27 erklärt (siehe auch Abbildung 18), kann man S und C an $\overline{AB}$ spiegeln, wobei man Punkte S' und C' erhält. Spiegelt man danach S' an $\overline{BC'}$ erhält man den Punkt S'', so dass der Abstand $|SS''|$ die Länge des kürzesten Rundweges von S über $\overline{AB}$ und $\overline{BC}$ berechnet. Es ergibt sich somit $\omega_{001} = |SS''| + 2|AS|$

Im Fall $(0, 0, 0)$ gibt es nur eine versteckte Tasche, nämlich in C. Der kürzeste Rundweg führt also von S nach C über $\overline{AB}$ und dann nach S zurück. Da die Länge des kürzesten Weges von S nach C über $\overline{AB}$ der Abstand $|SC'|$ ist, mit C' dem an $\overline{AB}$ gespiegeltem Punkt C, so erhält man $\omega_{000} = |SC'| + |SC|$. Die optimalen Wege der acht Szenarien sind in Abbildungen 24 bis 31 abgebildet. Die versteckten Taschen sind symbolisch als blaue Rauten eingezeichnet.

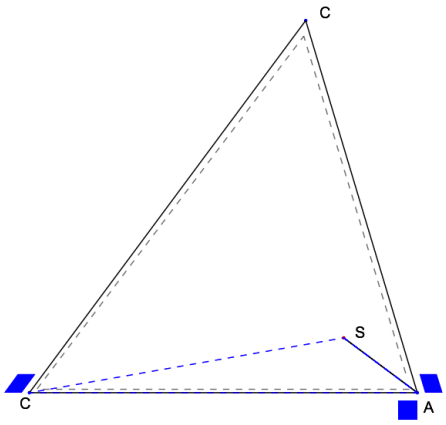


Abbildung 24: Der optimale Wege für das Szenario (111)

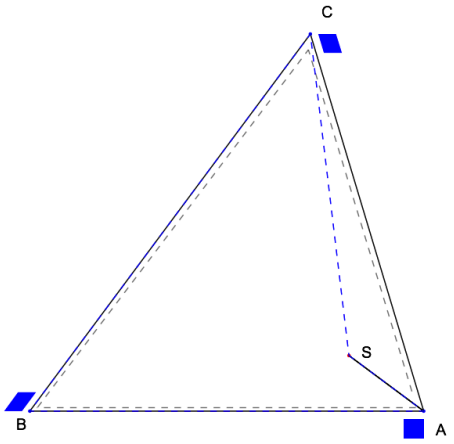


Abbildung 25: Der optimale Wege für das Szenario (110)

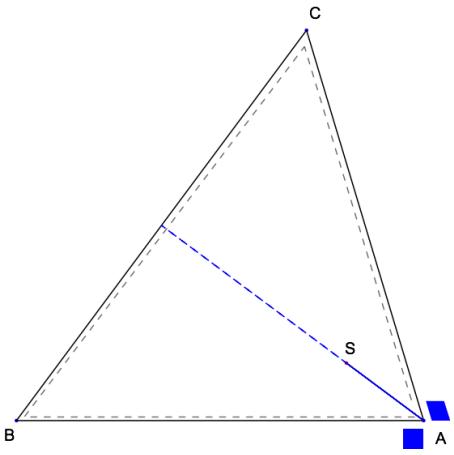


Abbildung 26: Der optimale Wege für das Szenario $(1, 0, 1)$

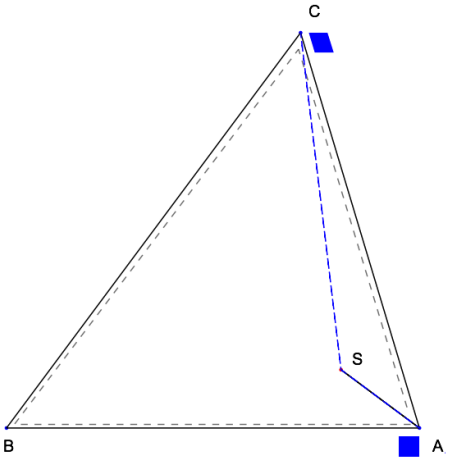


Abbildung 27: Der optimale Wege für das Szenario (100)

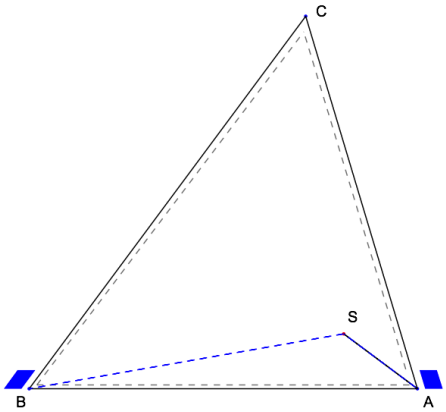


Abbildung 28: Der optimale Wege für das Szenario (011)

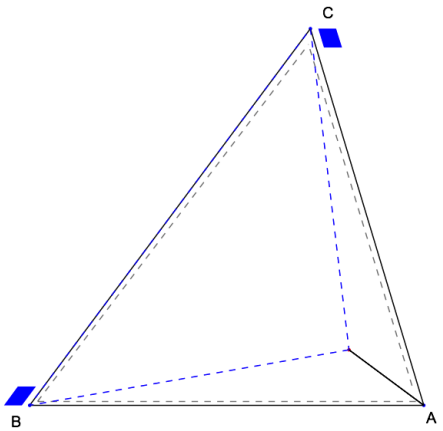


Abbildung 29: Der optimale Wege für das Szenario (010)

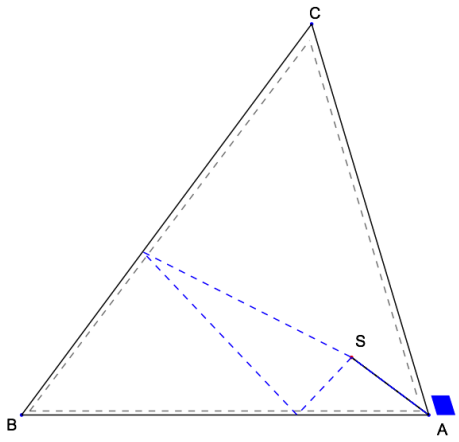


Abbildung 30: Der optimale Wege für das Szenario $(0, 0, 1)$

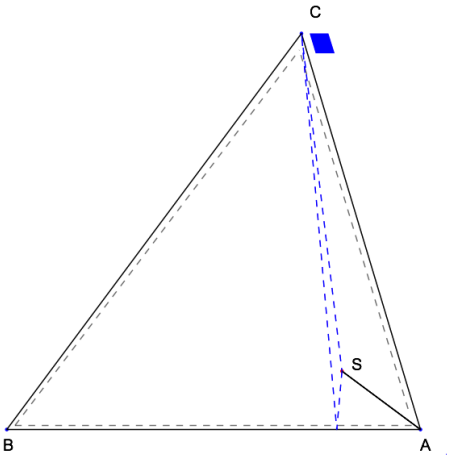


Abbildung 31: Der optimale Wege für das Szenario (000)

Bestimmung einer unteren Schranke mit Python/SciPy

Abhängig von den Punkten A, B, C, S wurden zuerst alle acht Funktionen $W_{e_1e_2e_3}(x, y, z)$ durch eine Python-Funktion $f_{e_1e_2e_3}(A, B, C, S, x, y, z)$ implementiert. Ferner wurden die optimalen Weglängen in ein dreidimensionales Array $Optima[e_1][e_2][e_3]$ abgelegt. Nach Lemma 7 ist das Minimax-Problem

$$\min_{x\in[0,1]} \max\left(H_0(x), H_1(x)\right) \tag{58}$$

gleichwertig zu dem Optimierungsproblem

$$\min_{(x,z)\in\mathbb{R}^2} z$$

$$z - H_0(x) \geq 0, \qquad z - H_1(x) \geq 0, \qquad 0 \leq x \leq 1$$

was wieder durch die Funktion *minimize* berechnet wird:

```
f = lambda x: x[1]
cons=(
  {  'type':'ineq',
     'fun': lambda x: x[1]-H(x[0],A,B,C,S,f000,f001,f010,f011,Optima[0])
  },
  {  'type':'ineq',
     'fun': lambda x: x[1]-H(x[0],A,B,C,S,f100,f101,f110,f111,Optima[1])
  },
```

```
  {  'type':'ineq',   'fun': lambda x:x[0]     },
  {  'type':'ineq',   'fun': lambda x:1-x[0]   }
);
result=minimize(f,(0.5,0),method="COBYLA", constraints=cons);
```

Der Funktion H werden dabei die entsprechenden Weglängenfunktionen und die zugehörigen optimalen Weglängen übergeben. Die erste Komponente ist der zu ermittelnde x-Wert bei dem ein Minimum erreicht werden soll. Die Deklaration der Funktion H lautet:

```
def H(xWert,A,B,C,S,g00,g01,g10,g11)
```

Die Funktion H berechnet nun das Minimax-Problem

$$\min_{y\in[0,1]} \max\left(G_{e_0 0}(x,y), G_{e_0 1}(x,y)\right) \tag{59}$$

in Abhängigkeit des übergebenen x-Wertes und der Funktionen $g00, g01, g10, g11$, was durch einen erneuten Aufruf der Funktion *minimize* geschieht:

```
f = lambda x: x[1]
cons=(
 {   'type':'ineq',
     'fun': lambda x: x[1]-G(xWert,x[0],A,B,C,S,g00,g01,Optima[0])
 },
 {   'type':'ineq',
     'fun': lambda x: x[1]-G(xWert,x[0],A,B,C,S,g10,g11,Optima[1])
  }
);
result=minimize(f,(0.5,0),method="COBYLA", constraints=cons);
```

Der Funktion G werden wieder die entsprechenden Weglängenfunktionen und optimalen Weglängen übergeben, sowie die derzeitigen Werte für x und y. Die Deklaration der Funktion G lautet:

```
def G(xWert,yWert,A,B,C,S,h0,h1,Optima)
```

Wiederum muss ein Minimax-Problem gelöst werden, welches diesmal

$$\min_{z\in[0,1]} \max\left(F_{e_0 e_1 0}(x,y,z), F_{e_0 e_1 1}(x,y,z)\right) \tag{60}$$

lautet und durch folgenden Aufruf der Funktion *minimize* realisiert wird:

```
f = lambda x: x[1]
cons=(
 {   'type':'ineq',
     'fun': lambda x: x[1]-h0(A,B,C,S,xWert,yWert,x[0])/Optima[0]
 },
 {   'type':'ineq',
     'fun': lambda x: x[1]-h1(A,B,C,S,xWert,yWert,x[0])/Optima[1]
 }
);
result=minimize(f,(0.5,0),method="COBYLA", constraints=cons);
```

Der Aufruf der gesamten Prozedur, bei der auch die optimalen Weglängen berechnet werden wurde in einer Python-Funktion

```
def KF(A,B,C,S):
```

implementiert und liefert die Lösung des iterierten Minimax-Problems und somit eine untere Schranke $KF_{(A,B,C,S)}$ für das Erkundungsproblem.

4.2 Numerische Bestätigung der unteren Schranke 1.287

Mithilfe der Python-Funktion KF können wir für konkrete Dreiecke A, B, C mit Startpunkt S das iterierte Minimax-Problem lösen. Ohne Einschränkung nehmen wir $B = (0, 0)$ und $A = (1, 0)$ an. Nach Annahme soll $\overline{AC}$ die Länge 1 haben, so dass C auf dem Kreisbogen mit Mittelpunkt A und Radius 1 liegt. Wir haben somit nur noch die Möglichkeit den Winkel $\measuredangle(C, A, B)$ zu ändern. Da der Startpunkt S auf der Winkelhalbierenden des Winkels $\measuredangle(C, A, B)$, zwischen A und dem Mittelpunkt von $\overline{BC}$, liegt, gibt es ein $t \in]0, 1[$ mit

$$S = A + \frac{t}{2}(C + B - 2A) = A + \frac{t}{2}(C - 2A).$$

Wir wollen zuerst annehmen, dass auch $\overline{BC}$ die Länge 1 hat, d.h. $C = \frac{1}{2}(1, \sqrt{3})$ und $\measuredangle(C, A, B) = 60°$. Wegen der Symmetrie sehen alle möglichen acht Polygone $P_{e_1e_2e_3}$ von S gesehen gleich aus. Ferner gilt

$$S = \frac{1}{4}(4 - 3t, \sqrt{3}t), \qquad \text{für ein } t \in]0, 1[.$$

Um die untere Schranke für den Kompetitivitätsfaktor zu maximieren betrachten wir die Funktion

$$\kappa(t) := \mathrm{KF}\left((1, 0), (0, 0), (1/2, \sqrt{3}/2, ((4 - 3t)/4, \sqrt{3}t/4)\right), \tag{61}$$

bei der wir unsere Python-Funktion KF benutzen. Durch SciPy's Funktion *minimize_scalar* können wir $\kappa(t)$ maximieren, indem wir $-\kappa(t)$ minimieren. Dabei ergibt sich ein maximaler Kompetitivitätsfaktor von

$$KF_{(A,B,C,S)} = 1.2874883652798921.$$

bei $t = 0.34990307454116104$, d.h. für den Startpunkt $S = (0.737572694094, 0.151512475707)$. Dieser Wert bestätigt damit den in [11] experimentell ermittelten Wert von 1.2825 durch ein numerisches Verfahren. Abbildung 32 zeigt dieses Dreieck, mit den kritischen Punkten auf den drei Seiten. Auf der unteren Seite $\overline{BA}$ ergab sich die kritische Stelle $x^* = 0.1971722$, bei dem

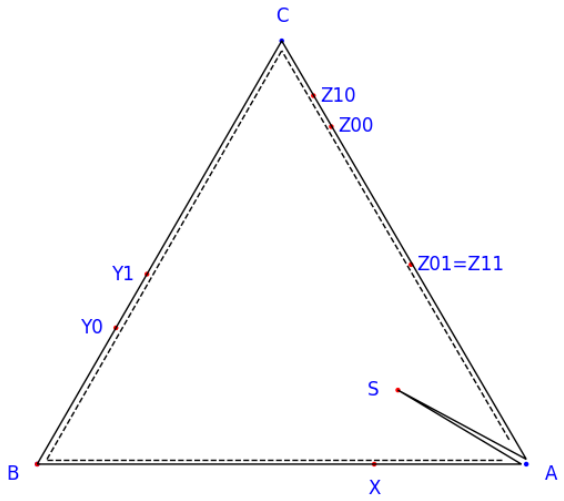


Abbildung 32: Nachweis der unteren Schrank 1.287

$H_0(x^*)$ oder $H_1(x^*)$ das äußere Minimax-Problem $\min_x \max\{H_0(x), H_1(x)\}$ löst. Die Lösung des mittleren Minimax-Problems $H_0(x^*) = \min_y \max\{G_{00}(x^*, y), G_{01}(x^*, y)\}$ wird an der kritischen Stelle $y_0^* = 0.3219905136876557$ erreicht, wogegen die kritische Stelle für $H_1(x^*) = \min_y \max\{G_{10}(x^*, y), G_{11}(x^*, y)\}$ bei $y_1^* = 0.4488542442169342$ erreicht wird. Die kritischen Stellen $z^*_{e_0e_1}$ der inneren Minimax-Probleme $\min_z \max\{F_{e_0e_10}(x^*, y^*_{e_0}, z), F_{e_0e_11}(x^*, y^*_{e_0}, z), \}$ liegen bei

$$z_{00}^* = 0.202694728, \qquad z_{01}^* = 0.528625383, \qquad z_{10}^* = 0.129474835, \qquad z_{11}^* = 0.528625358$$

Die acht relativen Weglängen berechnen sich wie folgt:

$F_{000}(x^*, y_0^*, z_{00}^*)$	$= 1.2874721258235022$	$F_{001}(x^*, y_0^*, z_{00}^*)$	$= 1.2874721022305646$
$F_{010}(x^*, y_0^*, z_{01}^*)$	$= 1.2874767268835747$	$F_{011}(x^*, y_0^*, z_{01}^*)$	$= 1.2874776689875138$
$F_{100}(x^*, y_1^*, z_{10}^*)$	$= 1.2874883652798921$	$F_{101}(x^*, y_1^*, z_{10}^*)$	$= 1.2874867389180298$
$F_{110}(x^*, y_1^*, z_{11}^*)$	$= 1.2874711208878942$	$F_{111}(x^*, y_1^*, z_{11}^*)$	$= 1.2874710577682535$

4.3 Leichte Verbesserung der unteren Schranke auf 1.2945

Wir wollen jetzt die Einschränkung $\alpha = \measuredangle(C, A, B) = 60°$ fallen lassen und auch den Winkel variieren. Wieder nehmen wir an, dass die Schwellen $\overline{AB}$ und $\overline{CA}$ gleich lang sind und dass der Startpunkt S auf der Winkelhalbierenden des Winkels $\measuredangle(C, A, B)$ liegt, d.h.

$$S = A + \frac{t}{2}\left(C + B - 2A\right), \qquad t \in]0, 1[.$$

Wegen der Symmetrie sehen wieder alle möglichen acht Polygone von S gesehen gleich aus. Unter der Annahme $A = (1, 0)$ und $B = (0, 0)$ gibt es daher ein $\alpha \in]0, \pi[$ und ein $t \in]0, 1[$, so dass

$$C = (1 - \cos(\alpha), \sin(\alpha)), \qquad S = \left(1 - \frac{t(\cos(\alpha) + 1)}{2}, \frac{t\sin(\alpha)}{2}\right).$$

Um die untere Schranke für den Kompetitivitätsfaktor zu maximieren, betrachten wir die Funktion

$$\kappa(\alpha, t) := \mathrm{KF}\left((1, 0), (0, 0), (1 - \cos(\alpha), \sin(\alpha)), (1 - t(\cos(\alpha) + 1)/2, t\sin(\alpha)/2)\right). \tag{62}$$

Das zweidimensionale Optimierungsproblem lässt sich dann wieder mit *minimize* lösen durch den Aufruf:[5]

```
minimize(lambda x: (-1)*kappa(x),[pi/3,0.3499030745411],method='Nelder-Mead')
```

Hierbei ist die Variable x ein zweidimensionaler Vektor $x = [x[0], x[1]]$ der für das Argument (α, t) steht. Als Startwerte habe ich die im vorherigen Abschnitt ermittelten optimalen Werte für C und S angegeben, d.h. einen Winkel $\alpha = \pi/3$ für ein gleichseitiges Dreieck und den obigen Wert für t.

Als Ergebnis erhalten wir ein Optimum bei $\alpha = 1.28056317$ und $t = 0.29414688$ ergeben, d.h. $\alpha \approx 73.3°$. Die Punkte C und S sind dabei

$$C = (0.713824361598, 0.958177177762), \;\; S = (0.810837724416, 0.140922413663)$$

und die Funktion KF ergibt das beste Ergebnis unserer Arbeit:

$$KF_{(A,B,C,S)} = 1.2945872183711380.$$

Abbildung 33 zeigt das ermittelte gleichschenklige Dreieck mit den kritischen Punkten auf den Seiten.

[5] Ohne Angabe der Optimisierungsmethode verwendet *minimize* die BFGS-Methode, die in meinem Fall zu einem Überschreiten der Rechengenauigkeit geführt hat. Die Nelder-Mead Methode liefert mir dagegen den besten Wert.

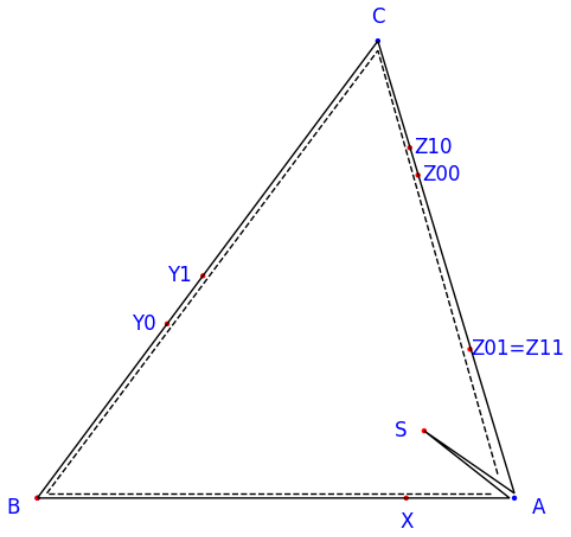


Abbildung 33: Das gleichschenklige Dreieck mit dem besten KF

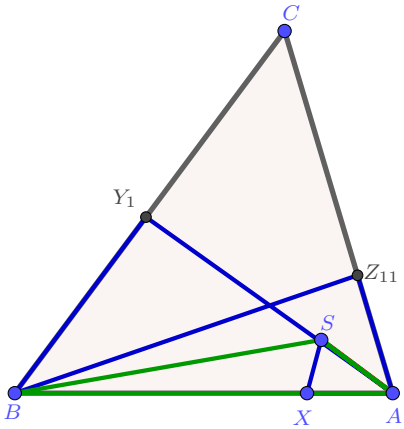
Auf der unteren Seite $\overline{BA}$ ergab sich die kritische Stelle $x^* = 0.2268533877908267$, bei der $H_0(x^*)$ oder $H_1(x^*)$ das äußere Minimax-Problem löst. Die Lösung des mittleren Minimax-Problems wird an der kritischen Stelle $y_0^* = 0.3810496569529814$ und $y_1^* = 0.4861416253872443$ erreicht. Die kritischen Stellen $z^*_{e_0e_1}$ der inneren Minimax-Probleme liegen bei

$$z_{00}^* = 0.293593312, \qquad z_{01}^* = 0.674510618, \qquad z_{10}^* = 0.233716094, \qquad z_{11}^* = 0.674521258$$

und die acht relativen Weglängen berechnen sich wie folgt:

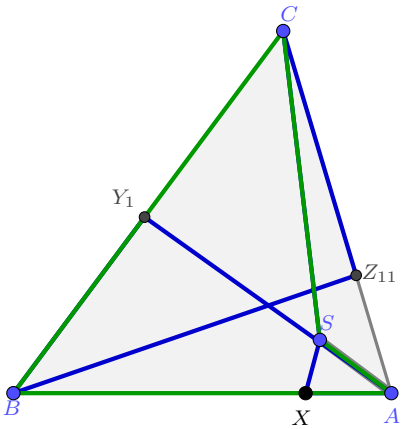
$F_{000}(x^*, y_0^*, z_{00}^*) = 1.2945684066579768$		$F_{001}(x^*, y_0^*, z_{00}^*) = 1.2945684067328900$	
$F_{010}(x^*, y_0^*, z_{01}^*) = 1.2945872183711380$		$F_{011}(x^*, y_0^*, z_{01}^*) = 1.2945647332079770$	
$F_{100}(x^*, y_1^*, z_{10}^*) = 1.2945665324647275$		$F_{101}(x^*, y_1^*, z_{10}^*) = 1.2945665323425639$	
$F_{110}(x^*, y_1^*, z_{11}^*) = 1.2945759308923610$		$F_{111}(x^*, y_1^*, z_{11}^*) = 1.2945454801188201$	

Abbildungen 34 bis 41 wurden mit GeoGebra 5 erstellt. Die Längen der grün eingezeichneten optimalen Wege und der blau eingezeichneten Wege der Erkundungen für alle acht Szenarien (bzgl. der oben angegebenen Landepunkte $X, Y_0, Y_1, Z_{00}, Z_{01}, Z_{10}$ und Z_{11}) wurden mit GeoGebra berechnet und bestätigen obige Werte des Python-Programms.



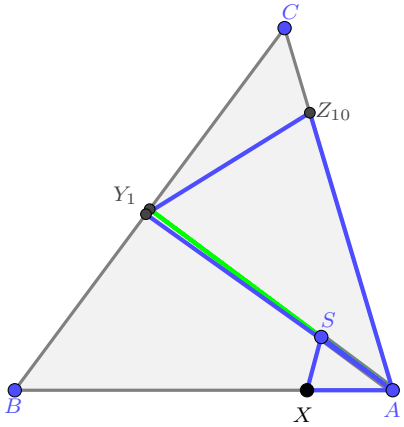
$W_{111} = 3.276036532318527$
$\omega_{111} = 2.530646147858676$
$F_{111} = 1.294545480050843$

Abbildung 34: Fall $(1, 1, 1)$



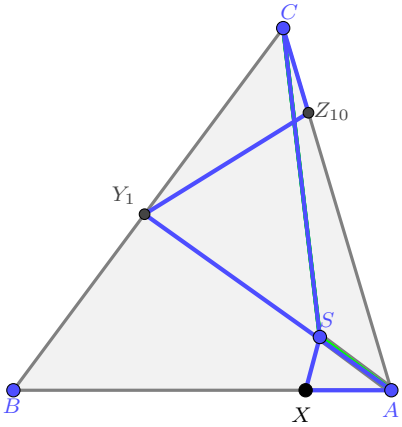
$W_{110} = 4.212187231478909$
$\omega_{110} = 3.253719717144619$
$F_{110} = 1.294575930828921$

Abbildung 35: Fall $(1, 1, 0)$



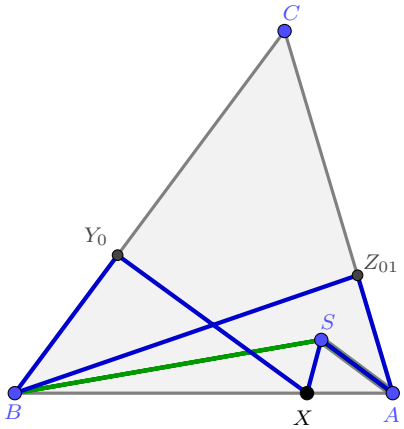
$W_{101} = 2.687033509769902$
$\omega_{101} = 2.075624111967202$
$F_{101} = 1.294566532676877$

Abbildung 36: Fall $(1, 0, 1)$



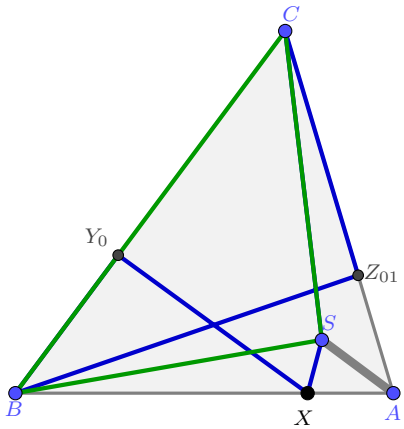
$W_{100} = 2.741573880930285$
$\omega_{100} = 2.117754331019059$
$F_{100} = 1.294566532469819$

Abbildung 37: Fall $(1, 0, 0)$



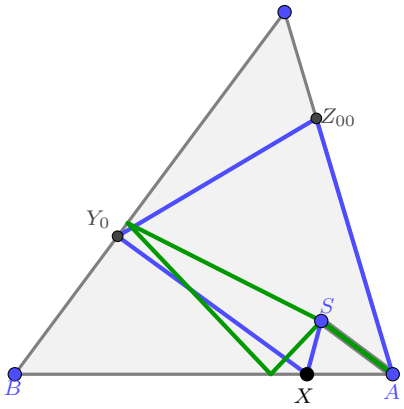
$W_{011} = 2.741570070228575$
$\omega_{011} = 2.117754331019059$
$F_{011} = 1.294564733062941$

Abbildung 38: Fall $(0, 1, 1)$



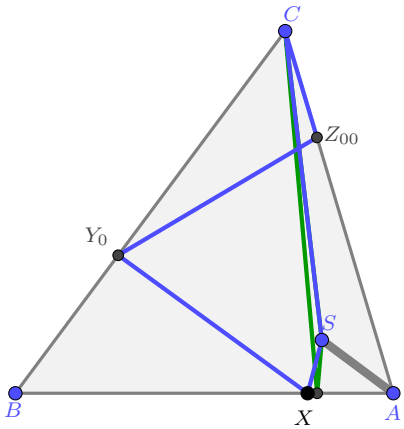
$W_{010} = 3.677699489388957$
$\omega_{010} = 2.840827900305001$
$F_{010} = 1.294587218392957$

Abbildung 39: Fall $(0, 1, 0)$



$W_{001} = 2.319517057150503$
$\omega_{001} = 1.294568406535016$
$F_{001} = 1.294568406535016$

Abbildung 40: Fall $(0, 0, 1)$



$W_{000} = 2.493811864310886$
$\omega_{000} = 1.92636546013219$
$F_{000} = 1.294568406630255$

Abbildung 41: Fall $(0, 0, 0)$

5 Schlußbemerkungen

Regelmäßige Polygone

Man kann unsere Vorgehensweise verallgemeinern und Polygone mit n Schwellen betrachten. Als Versuch habe ich beispielsweise regelmäßige Polygone betrachtet, bei denen der Startpunkt S im Ursprung liegt und deren Ecken auf dem Einheitskreis verteilt sind, d.h. $P[k] = (\cos(2\pi k/n), \sin(2\pi k/n)$ für $k \in \{0, \ldots, n-1\}$. Wir nehmen an, dass eine Erkundung alle Seiten bis auf die letzte Seite gegen den Uhrzeigersinn besucht (in Abbildung 42 sind die Schwellen rot-gestrichelt eingezeichnet). Auf jeder der ersten $n-2$ Seiten findet sich höchstens eine versteckte Tasche am Anfangspunkt des Segmentes, d.h. aus Sicht der Erkundung an der rechten Ecke des Segments. Auf der vorletzten Seite gibt es eine Tasche an genau einer seiner Endpunkte. Das iterierte Minimax-Problem hat nun $n-1$ Rekursionsstufen, welches ohne große Probleme in Python implementiert werden kann, obwohl die Ausführung bei $n > 4$ deutlich mehr Zeit benötigt. Dabei war meine Beobachtung, dass bei wachsendem n, die Abschätzung für den Kompetitivitätsfaktor fällt anstatt zu steigen.

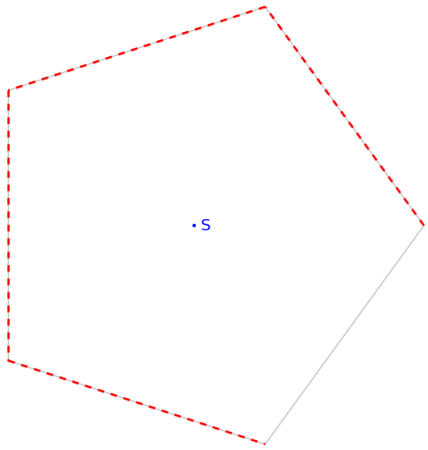


Abbildung 42: Pentagon, KF=1.1927

Zusammenfassung der Ergebnisse

In dieser Arbeit haben wir versucht, den Kompetitivitätsfaktor einer (kompetitiven) Erkundungsstrategie eines Polygons nach unten abzuschätzen. Zentral bei unserer Betrachtung war die Idee einer Schwelle, die jeder Erkundungsrundgang betreten muss, um das Polygon vollständig zu erforschen. Ausgehend von der einfachsten unteren Schranke, die durch das Dreieck aus Abbildung (6) ermittelt werden kann und den Kompetitivitätsfaktor nach unten mit 1.207 abschätzt, haben wir versucht, die Anzahl der Schwellen zu erhöhen. Für Fünfecke mit zwei Schwellen hat sich in Kapitel 3 eine untere Schranke von 1.2717 ergeben. In Kapitel 4 wurden Polygone mit drei Schwellen betrachtet und die untere Schranke 1.285 von Hagius et al. numerisch belegt. Darüber hinaus wurde dieses Ergebnis durch Angabe eines bestimmten Polygons auf 1.2945 verbessert.

Literatur

[1] Albers, Kursawe, and Schuierer, *Exploring unknown environments with obstacles*, Algorithmica **32** (2002), no. 1, 123–143.

[2] Spyros Angelopoulos, Diogo Arsénio, and Christoph Dürr, *Infinite linear programming and online searching with turn cost*, Theoret. Comput. Sci. **670** (2017), 11–22.

[3] Margrit Betke, Ronald L. Rivest, and Mona Singh, *Piecemeal learning of an unknown environment*, Machine Learning **18** (1995), no. 2-3, 231–254.

[4] Allan Borodin and Ran El-Yaniv, *Online computation and competitive analysis*, Cambridge University Press, New York, 1998.

[5] Wei-Pang Chin and Simeon Ntafos, *Optimum watchman routes*, Inform. Process. Lett. **28** (1988), no. 1, 39–44.

[6] V. Chvátal, *A combinatorial theorem in plane geometry*, J. Combinatorial Theory Ser. B **18** (1975), 39–41.

[7] Xiaotie Deng, Tiko Kameda, and Christos Papadimitriou, *How to Learn an Unknown Environment (Extended Abstract)*, 32nd Annual Symposium on Foundations of Computer Science, San Juan, Puerto Rico, 1-4 October 1991, 1991, pp. 298–303.

[8] ———, *How to learn an unknown environment. I. The rectilinear case*, J. ACM **45** (1998), no. 2, 215–245.

[9] Sándor P. Fekete and Christiane Schmidt, *Polygon exploration with time-discrete vision*, Comput. Geom. **43** (2010), no. 2, 148–168.

[10] S Fisk, *A short proof of Chvalal's watchman theorem*, Combin. Theory Ser. B **24** (1978), 374.

[11] Roland Hagius, Christian Icking, and Elmar Langetepe, *Lower bounds for the polygon exploration problem (Extended abstract)*, 20th European Workshop on Computational Geometry, Seville, Spain, March 24-26 (2004), 5pp.

[12] Frank Hoffmann, Christian Icking, Rolf Klein, and Klaus Kriegel, *The polygon exploration problem*, SIAM J. Comput. **31** (2001), no. 2, 577–600.

[13] Christian Icking and Rolf Klein, *Algorithmische Geometrie*, Kurs 1840, FernUniversität in Hagen, 2019.

[14] Eric Jones, Travis Oliphant, Pearu Peterson, et al., *SciPy: Open source scientific tools for Python* (2001), `http://www.scipy.org/`, [Online; accessed 27. Juni 2019].

[15] Joseph O'Rourke, *Art gallery theorems and algorithms*, International Series of Monographs on Computer Science, The Clarendon Press, Oxford University Press, New York, 1987.

[16] Markus Schwarz, *Shortest watchman routes und eine Implementierung in Java*, Diplomarbeit, FernUniversität in Hagen, 2000.

[17] Daniel D. Sleator and Robert E. Tarjan, *Amortized efficiency of list update and paging rules*, Comm. ACM **28** (1985), no. 2, 202–208.

[18] Xue-Hou Tan and Tomio Hirata, *Constructing shortest watchman routes by divide-and-conquer*, Algorithms and computation (Hong Kong, 1993), Lecture Notes in Comput. Sci., vol. 762, Springer, Berlin, 1993, pp. 68–77.