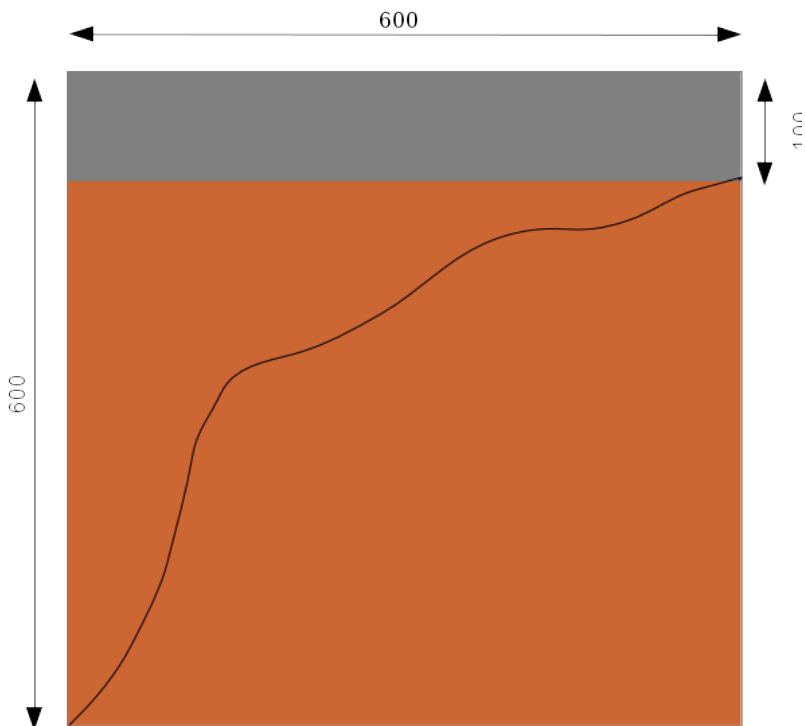


## Am Strand

### Wer findet den günstigsten Weg?



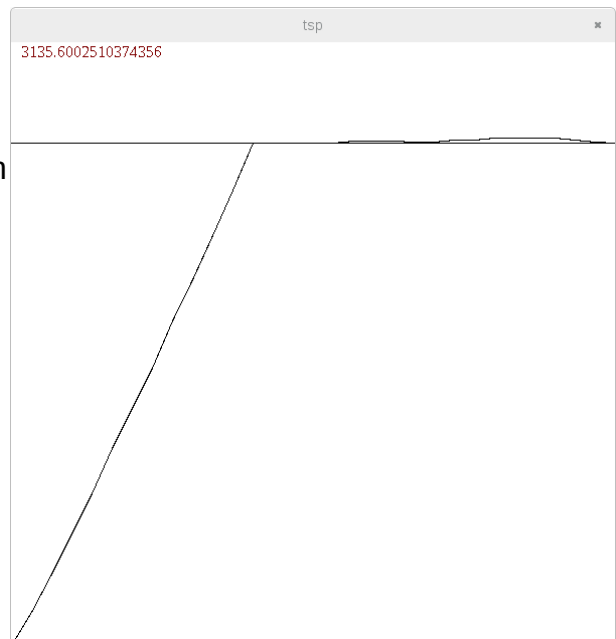
Im Sand am Strand bewegt man sich deutlich langsamer als auf dem befestigten Teil der Strandpromenade. Wie muss man dann eigentlich gehen, wenn man möglichst günstig vom eigenen Standort zum Kiosk an der Promenadenkante gehen will?

Die Abmessungen (in cm) sind hier klein gewählt. Die betrachtete Fläche hat eine Breite von 6 m und wir befinden uns links unten in der Ecke 5 m von der Promenadenkante entfernt. Von der Promenade sind 1 m Breite eingezeichnet. Die Verzögerung berücksichtigen wir durch einen Faktor mit dem die Weglänge im Sand multipliziert wird.

### Warum Genetische Algorithmen?

Ihre Anwendung auf dieses Beispiel ist nur zu Demonstrationszwecken geeignet. Das liegt daran, dass es zu diesem "einfachen" Problem eine einfache mit den Mitteln der Analysis zu berechnende Lösung gibt und dann macht man das analytisch und nicht mit Suchverfahren. Allerdings zeigt die Anwendung der GA mit grafischer Darstellung sehr schön, wie sie arbeiten, wie sie eine gute Lösung erzielen und wieso manchmal eben auch nicht.

Das rechts dargestellte Ergebnis (für  $v=5$ ) ist also nicht schlecht, insbesondere, wenn man die Ausgangswerte des Prozesses betrachtet, aber auch noch nicht das Beste zu erzielende. Interessant ist es mit verschiedenen Mutationen und Parameterwerten zu experimentieren.



## **Überlegungen zur Modellierung**

### **Datenstruktur und Pool**

Für die Datenstruktur gibt es einen naheliegenden Ansatz: Die Punkte werden als Zahlenpaare der Koordinaten gespeichert, "wie üblich" x-Koordinaten nach rechts und y nach unten.

Zu einer möglichen Vereinfachung kommt man bei den Überlegungen zum Generieren des Pools und damit der Zwischenpunkte der Wege.

Eine erste Variante generiert diese rein zufällig. Das ist aber offensichtlich keine gute Grundmodellierung, da dies zu rückwärts führenden Wegabschnitten und Überschneidungen führt. Man könnte das vermeiden, wenn man die zufällig generierten Zwischenpunkte in den Wegen jeweils zunächst nach ihren x-Koordinaten sortiert.

Einfacher wird das ganze, wenn man vorher die Anzahl der Zwischenabschnitte für alle Wege des Pools vorgibt und diese gleich breit macht, so dass man die x-Koordinaten nicht mehr zufällig generiert, sondern gleichmäßig aufsteigend vorgibt. Dann könnte man die x-Koordinaten sogar jeweils weglassen und bräuchte dann nur noch die y-Koordinaten in einer Liste ablegen, da sich die x-Koordinate aus der Position des Wertes in der Liste berechnen ließe.

In meiner Lösung habe ich das vermieden, da man sie dann jeweils für die Ausgabe neu berechnen müsste. Herauszufinden, welche Variante besser wäre, könnte eine interessante Aufgabe sein.

Bei einer Breite von 600 im Beispiel und 20 Wegabschnitten wären sie dann jeweils 30 breit, so dass die Liste dann die Form

[ (0,600) (30,<Zufallswert>) (60,<Zufallswert>) ... (570,<Zufallswert>) (600,100) ] hat.

### **Crossing-over (kreuzen)**

Das erleichtert die Modellierung des Kreuzens und deren Umsetzung im Programm stark. Man muss nun nur noch die beiden Positionen für den jeweils zwischen zwei Wegen auszutauschenden Wegabschnitt zufällig bestimmen. Die zugehörigen Wegausschnitte werden dann zwischen den beiden Wegen ausgetauscht.

### **Mutationen**

Die eigentlich interessanten Überlegungen betreffen die Mutation. Als Beispiele sind drei Varianten im Projekt umgesetzt:

- Einzelne Orte werden zufällig ausgewählt und deren y-Koordinaten zufällig neu erzeugt. Dabei können sie alle Werte zwischen 0 und 600 einnehmen. Hierin liegt auch der Grund, dass die befestigte Promenade mit einer Breite von 100 mit in die Modellumgebung aufgenommen worden ist, da sonst die Wahrscheinlichkeit zu klein wäre, den Rand zu erreichen.
- Einzelne Orte werden zufällig ausgewählt und deren y-Koordinaten zufällig um einen Wert verändert. Beispielsweise kann man sie bei einer Bandbreite von 100 jeweils maximal 50 nach oben oder 50 nach unten verändern. Dabei kann man noch mit der Bandbreite und deren Schwerpunktsetzung experimentieren.
- Es kann auch erfolgreicher sein, einen ganzen Wegabschnitt so zu modifizieren. Dazu müssen dann die beiden y-Koordinaten zweier benachbarter Punkte um den gleichen Betrag verändert werden.

Die letzte Variante bringt auf die Idee, ob es nicht auch interessant sein könnte, als einen möglichen Prozess die y-Koordinaten zweier benachbarter Punkte auszutauschen.

### **Experimentieren**

Das reizvolle an GA ist das Experimentieren mit fertigen Lösungen, einerseits wie oben angegeben mit der Art der Modellierung von Prozessen, aber auch mit der Wahl der Parameterwerte.

Dies sind zunächst die Grundparameter Größe des Pools und Anzahl der Schritte (Generationenfolge) und die Steuerung der Parameter für Wahrscheinlichkeit, mit der Kreuzen und Mutation auftreten sollen. Bei der hier vorliegenden Modellierung hat die Anzahl der Zwischenpunkte zusätzlich auf die Bestimmung einer Lösung Einfluss.

### **Selektion**

Auch für die Umsetzung der Selektion gibt es nicht "die richtige" Lösung. Im Beispielprojekt werden jeweils zwei Wege aus dem durch Kreuzen und Mutation vergrößerten Pool solange ausgewählt und von ihnen nur die bessere in den neuen Pool übernommen, bis der wieder die ursprüngliche Größe erreicht.

### **Nebenmaxima des Lösungsraums vermeiden**

Dabei ist ganz wichtig, dass unser Verfahren nicht zu strikt arbeitet. Es bleibt dann häufig in einem Zustand hängen, der einem Nebenmaximum des Suchraums entspricht. Um das zu vermeiden, kann es sinnvoll sein, die Lösungen zwischendurch einmal zu "heizen", also bewusst stärker zu verändern als im normalen Verfahren, damit im Pool wieder Wege auftauchen, die dieses Nebenmaximum vermeiden. Eine andere Möglichkeit ist, die in der Selektion beschriebene Paarbildung zur Auslese nicht strikt durchzuführen, sondern einzelne Individuen (Wege) des Pools einfach so zu übernehmen.

## Anhang

### Analytische Berechnung der Lösung zum Vergleich mit dem Ergebnis der Berechnung mit GA

Dazu wird angenommen, dass die Abschnitte in den beiden Bereichen jeweils gradlinig sein müssen, von der linken unteren Ecke zum um  $x$  von der linken Seitenkante entfernten Punkt an der Promenadenkante und von dort ebenso gradlinig zum Kiosk am rechten Rand.

Die Länge im Sand ist dann  $s = \sqrt{500^2 + x^2}$ , die auf der Promenade  $p = 600 - x$ .

Die erste Länge müssen wir mit dem Verzögerungsfaktor  $v$  multiplizieren. Der Aufwand ist dann:

$$a = v * \sqrt{500^2 + x^2} + 600 - x$$

Die optimal anzusteuern Position erhält man dann an der Nullstelle der Ableitung

$$a'(x) = v * \frac{2x}{2 * \sqrt{500^2 + x^2}} - 1, \text{ zu } x = \frac{500}{\sqrt{v^2 - 1}}$$

Für einen Verzögerungsfaktor von 5 erhält man  $\frac{500}{\sqrt{24}} \approx 102$  und eine minimale gewichtete Länge von etwa 3050.