1 Gegenbeispiel Algorithmus von Helvig et. al.

Dieses Kapitel dient temporär als Gegenbeispiel für den Algorithmus Helvig et. al. Input:

- Ziele $Z = \{(-933, 13), (-883, -6), (-203, -12), (756, 8)\}$
  (Die Zahlen der Ziele sind zufällig generiert, also nicht wundern)
- Verfolger $\kappa = (0, 15)$

Damit befinden sich drei Ziele auf der linken und eines auf der rechten Seite des Ursprungs. TODO: Wenn Gegenbsp korrekt, erstelle Grafik
Mit dem Algorithmus von Helvig et. al. würden nun folgende 6 Zustände erstellt werden:

$$
\begin{aligned}
&A_0\\
&A_1 = \{(-203, -12), (756, 8)\}\\
&A_2 = \{(756, 8), (-203, -12)\}\\
&A_3 = \{(-883, -6), (756, 8)\}\\
&A_4 = \{(756, 8), (-883, -6)\}\\
&A_5 = \{(-933, 13), (756, 8)\}\\
&A_6 = \{(756, 8), (-933, 13)\}\\
&A_{final}
\end{aligned}
$$

Nun wird durch jeden dieser Zustände in chronologischer Reihenfolge iteriert. Dabei ergeben sich für die jeweiligen Zustände folgende Zeiten:

Iteration 0: $t = [0.0, 67.67, 108.0, \infty, \infty, \infty, \infty, \infty]$
Iteration 1: $t = [0.0, 67.67, 108.0, 98.11, 398.0, \infty, \infty, \infty]$
Iteration 2: $t = [0.0, 67.67, 108.0, 98.11, 398.0, \infty, \infty, 2079.33]$
Iteration 3: $t = [0.0, 67.67, 108.0, 98.11, 398.0, 1005.17, 528.48, 2079.33]$
Iteration 4: $t = [0.0, 67.67, 108.0, 98.11, 398.0, 1005.17, 528.48, 1737.78]$
Iteration 5: $t = [0.0, 67.67, 108.0, 98.11, 398.0, 1005.17, 528.485, 1737.78]$

Bemerke, dass keine Iteration 7 nötig ist, da nach A_{final} keine Ziele mehr abgefangen werden müssen. Mit der Iteration 6 gab es keine weiteren Änderungen, kann also auch weggelassen werden. Somit hat die vermeintlich optimale Tour eine Dauer von 1737.78-Zeiteinheiten. Dabei werden die Ziele in folgender Reihenfolge abgefangen:

$$
\begin{aligned}
(0, 0),&\ \text{Abfangzeit} : 0.0\\
(-933, 13),&\ \text{Abfangzeit} : 33.32\\
(-203, -12),&\ \text{Abfangzeit} : 67.66\\
(756, 8),&\ \text{Abfangzeit} : 398.0\\
(-883, -6),&\ \text{Abfangzeit} : 1199, 22\\
(0, 0),&\ \text{Abfangzeit} : 1727, 78
\end{aligned}
$$

Wendet man nun den Brute-Force-Algorithmus erhält man folgende Reihenfolge der Ziele:

$$\begin{aligned}(0,0),&\text{ Abfangzeit : } 0.0\\(-933,13),&\text{ Abfangzeit : } 33.32\\(-203,-12),&\text{ Abfangzeit : } 67.66\\(-883,-6),&\text{ Abfangzeit : } 98.11\\(756,8),&\text{ Abfangzeit : } 528.48\\(0,0),&\text{ Abfangzeit : } 860.73\end{aligned}$$

Zunächst würde man vermuten, dass der Algorithmus von Helvig et. al. nicht korrekt implementiert wurde. Dies kann man anhand dieses Beispiels wiederlegen. Nach dem Algorithmus wird der Graph G in topologischer Reihenfolge erstellt. Dabei gibt es nur Übergänge in Zustände mit einem Summenwert der Indizes, der genau ein höher ist, als der Summenwert des jetzigen Zustandes. Betrachten nun das nicht so ganz offensichtliche Problemziel $Z_P = (-993, 13)$. Dieses liegt in den Zuständen A_6 und A_7, welche sich damit in G an letzter Stelle vor A_{final} befinden. In der optimalen Tour liegt der erste Wendepunkt bei dem Ziel $(-883, -6)$ und fängt dabei $(-993, 13)$ und $(-203, -12)$ ab. Der eindimensionalen Algorithmus überprüft hingegen nicht auf Zustände bzw. Ziele, welche auf dem Weg zum Wendepunkt liegen. Diese werden erst im *Postprocessing*-Schritt hinzugefügt. Somit wird Z_P im Graphen erst ganz am Ende berechnet. Dabei dauern alle Pfade zu lange, sodass sich Z_P inzwischen sehr weit entfernt vom Ursprung befindet.

Dies lässt sich relativ simpel lösen, indem eben genau diese Zustände berechnet werden, bei denen das jeweilige s_k zwischen Wendepunkten eingeholt wird. Diese Zustände werden markiert und wenn eine Verbindung zu diesem Zustand berechnet wurde, werden die maximal 2 nachfolgenden Zustände berechnet. Befindet sich die Iteration bei einem markierten Zustand, wird dieser übersprungen. Zwar erhöhen sich die Berechnungen für einen Zustand ggf., die Gesamtberechnungen bleiben wiederum gleich. Dies lässt sich algorithmisch allerdings nur in $\mathcal{O}(n^3)$ lösen. Mit Sicherheit gibt es eine elegantere Lösung, bei der Laufzeitkomplexität von $\mathcal{O}(n^2)$ bleibt der Algorithmus damit vermutlich nicht mehr.