-Bäckerei Süpke hat wie so häufig Frühstücksbestellungen gesammelt, nachdem Frühstücksexpress ab 21.09 repariert wurde und unterwegs war

-Express: Sömmerda, Erfurt, Weimar und Jena

-möglich schnelle Route, um alle Stationen abzufahren und wieder in ihrer Bäckerei anzukommen

-manche merken es schon, manche wussten es noch

-> in diesem Vortrag werde ich euch das Problem des Handlungsreisenden vorstellen,

das sich genau damit beschäftigt

1 Erläuterung des Problems des Handlungsreisenden

2 Mögliche Lösungsverfahren

2.1 Greedy

2.2 Genetischer

2.3 MST in Kombination mit DFS

3 2-Opt-Verfahren zur Optimierung

4 Vorführung des Programmes

1 Erklärung des Problems des Handlungsreisenden

-Geschäftsmann gegebene Anzahl von Städten mit ihren Lagekoordinaten besuchen

-Gesamtdistanz der Route möglichst gering und am Ende Route am Startpunkt enden

-sollte bekannt sein

-> hier Beispiel mit Frühstücksexpress:

-wenn Startpunkt Sömmerda, dann könnte Frühstücksexpress bspw. hier lang fahren

-NP-schweres Problem -> kein optimaler Algorithmus, der die beste Route schnell und effizient findet (ohne halt alle Routen auszuprobieren – aber, dass das nicht effizient)

-> viele unterschiedliche Lösungsalgorithmen

-drei Stück in diesem Vortrag

2 Erläuterung der Lösungsverfahren

2.1 Greedy

-jeder Entscheidung die augenblicklich beste Wahl

-> Punkt mit geringster nächster Entfernung aus

-effizient -> nicht beste Lösung

-Beispiel:

-immer nächster Punkt

-Pseudocode

-> Abstandsbestimmung implementierte ich einfach über den SdP, hier auch nochmal abgebildet

-> vermutlich einfachste Algorithmus

2.2 Genetischer Algorithmus

-Name nicht ohne Grund: Evolution inspiriert

-Pseudocode

-Population: zufällige Routen (Basis)

-verschiedene Generationen

-in jeder Generation:

-Fitness (Fitness = Länge der Route, höhere Fitness -> kürzere Länge)

-mit höchster Fitness ausgewählt

-Kreuzung ausführen (Austausch von Punkten zwischen Routen)

-Mutation (zufällig Punkte innerhalb einer Route vertauscht)

-Beispiel:

-gegebene Punktemenge

-Generierung zufälliger Ausgangsrouten (eine exemplarisch)

-Bewertung Fitness (besten zwei behalten)

-Vermischung beim Crossover

-zufälliges Vertauschen zweier Punkte bei Mutation

2.3 Nutzung eines MSTs und der Tiefensuche

-Idee: zuerst MST in Graphen (also die möglichst kurze Verbindung aller Punkte)

-> Ermittlung durch Kruskal / Prim

-MST muss keinen Kreis enthalten -> eher „gerüstartig“

-> DFS, um Pfad zu finden, der jeden Punkt einmal besucht

-Beispiel:

-gegeben folgende Graph

-> MST schon eingezeichnet

-DFS durchgeführt (Knoten wird zur Route hinzugefügt, wenn zweite Mal besucht)

-wenn Punkt F -> zurück zu C -> E

-Städte durch Kanten, die nacheinander entdeckt

3 2-Opt-Verfahren

-Verbesserungsverfahren

-zwei Knoten zufällig rausgenommen vertauscht, sodass sich Kanten verändern

-Abfolge, die kürzer sein kann

-Beispiel: Ausschnitt einer Route

-blau: ursprüngliche Route, rot: neue Route

4 Vorführung des Programmes

-Vorstellung das Problem des Handlungsreisenden näherbringen

-> vielleicht nutzt die Bäckerei Süpke ja auch einen dieser Algorithmen, um die kürzeste Route zu berechnen