Simulated Annealing

Praktikumsbericht - Methoden der Höheren Physik
Betreuer: Prof. Pikovsky

Alexander Vogel, Sönke Beier

April 2022

Innerhalb dieses Projektes wird die Methode „Simulated Annealing“[Pre+92] verwendet, um den kürzesten Weg einer Städterundreise zu finden.

1 Theoretischer Hintergrund und Methode

Um dieses Optimierungsproblem zu lösen, wird der **Metropolis Algorithmus** angewendet, welcher ursprünglich für thermodynamische Systeme entwickelt wurde[Met+53]. Nötig wird dies, da es zu viel Zeit in Anspruch nehmen würde alle Konfigurationen einmal auszuprobieren und es bei dieser Art von Problem mehrere lokale Minimas gibt, welche schwer zu finden sind. Grundprinzip dieses Algorithmus ist es bei jeder Iteration eine zufällige Konfiguration der Städte zu erzeugen und mithilfe der Boltzmann Verteilung

$$p \propto e^{-\frac{\Delta E}{k_b T}}$$

zu entscheiden, ob diese neue Konfiguration angenommen wird oder nicht. Hierbei ist die Energiedifferenz $\Delta E = E_1 - E_2$ des urprünglichen Algorithmus äquivalent zu dem Weglängenunterschied der beiden Konfigurationen 1 und 2. Da bei $E_2 < E_1$ die Wahrscheinlichkeit 1 ist und somit die neue Konfiguration immer angenommen wird, wird der Weg der Konfiguration meist geringer und steigt nur manchmal wieder an[Pre+92]. Der Kontrollparameter T, welcher analog zur Temperatur verwendet wird, sinkt mit Fortlaufen der Simulation. Dadurch wird die Wahrscheinlichkeit geringer aus einem kleinem lokalen Minimum in ein viel größeres zu springen. Dies sorgt dafür, dass mit der Zeit immer kleinere lokale Minimas gefunden werden.
Dabei ist entscheidend, wie schnell die Temperatur abkühlt. Kühlt sie zu schnell ab, „verfängt“ sich das System in einem erst besten Minimum und bleibt hier stecken. Kühlt diese Temperatur zu langsam ab, springt das System aus schon guten Minimas wieder heraus und so dauert die Berechnung sehr lang. Zur Verringerung von T folgen wir hierbei[Pre+92] und senken die Temperatur nach $100N$ Rekombinationen oder $10N$ erfolgreichen Rekombinationen (also bei Rekombinationen mit niedriger Weglänge) um einen Faktor c (Abkühlungsparameter), wobei N für die Anzahl der Städte steht.
In diesem Versuch werden wir zwei zufällige **Vertauschungsverfahren** verwenden. Beim **„swapping“** Verfahren werden je Iterationsschritt zwei zufällig ausgewählte Städte vertauscht (bspw. Stadt 3 und 6 werden ausgewählt und der Weg ist nach der Vertauschung 1,2,6,4,5,3,7,..). Beim **„revert part“** werden zwei Städte zufällig ausgewählt und der Weg zwischen diesen beiden Städten wird umgekehrt (bspw. Stadt 3 und 6 werden ausgewählt und nach der Vertauschung ist der Weg 1,2,6,5,4,3,7..).

2 Ergebnisse und Auswertung

2.1 Anfangsbedingungen und Daten

Die Städte zwischen denen die kürzeste Route gefunden werden soll sind mitsamt ihren geografischen Koordinaten im CSV Format gespeichert. Dabei wurde eine Liste mit insgesamt 40 Städten benutzt, siehe Anhang Kapitel 3. Diese Städte mit ihren Koordinaten wurden zu Beginn in das Programm geladen. Dann wurde eine Lookuptabelle erstellt, die die Entfernung zwischen jedem vorhandenen Städtepaar enthält um im späteren Verlauf des Programms nicht bei jedem Schritt die Entfernung zwischen den einzelnen Städten neu berechnen zu müssen.

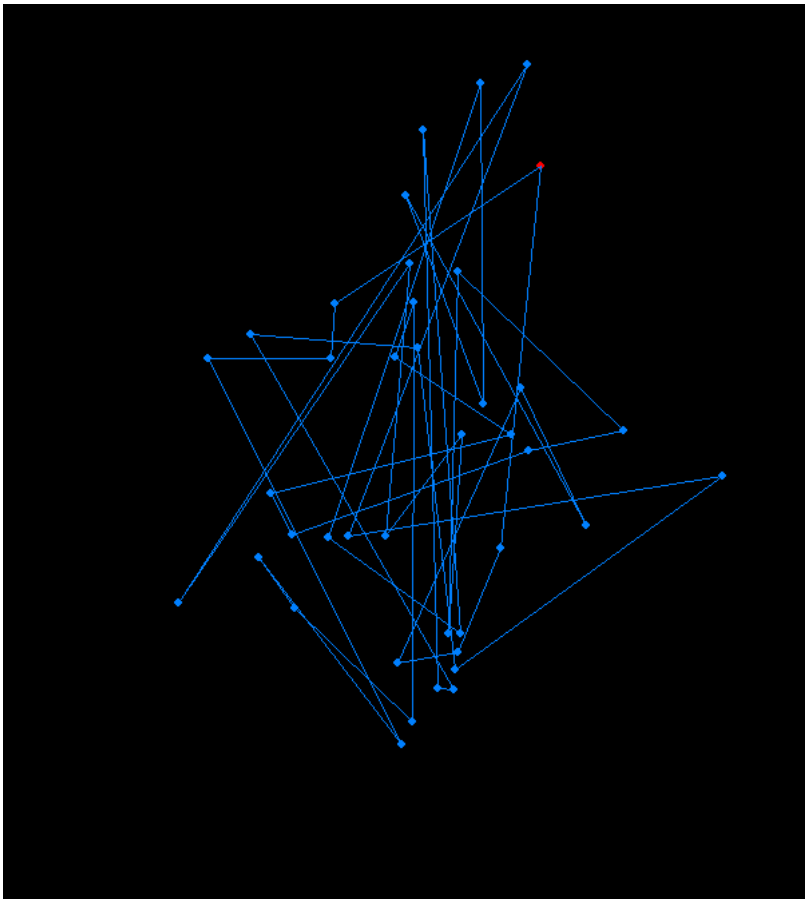

Figure 1: Startroute (Tabelle 3) für alle folgenden Berechnungen. Gesamtlänge 10974km. Die rote Markierung ist Berlin.

Als Startroute (Abbildung 1) wurde die Reihenfolge der Städte gewählt so wie sie in der CSV Datei gespeichert sind. Die Routenlänge beträgt: 10974 km

2.2 Abbruchbedingung

Als Abbruchbedingung der Simulation verwendeten wir

$$e^{-10km/T} < p_{abbr}$$

Dabei gilt die Simulation als abgeschlossen, sobald die Wahrscheinlichkeit von einer Konfiguration in eine 10 km größeren Konfiguration zu springen unter einer Abbruchwahrscheinlichkeit p_{abbr} liegt. Bei einer geringen Abbruchwahrscheinlichkeit ist es ab diesem Zeitpunkt also sehr unwahrscheinlich, dass sich die Länge der Konfiguration noch signifikant ändern würde. Eine passende Abbruchwahrscheinlichkeit ermittelten wir, indem wir mehrere Verläufe bei unterschiedlichen Abkühlungsparametern c ausprobierten (siehe 2). So waren bei dem verwendeten $p_{abbr} = 0,05$ kaum noch Veränderungen in der Weglänge feststellbar. Die Länge der letzten Konfiguration wurde als endgültige Länge der Simulation verwendet, welche für die Berechnung der Wahrscheinlichkeitsdichten in den Abbildungen 3, 4, 5 verwendet wurden.

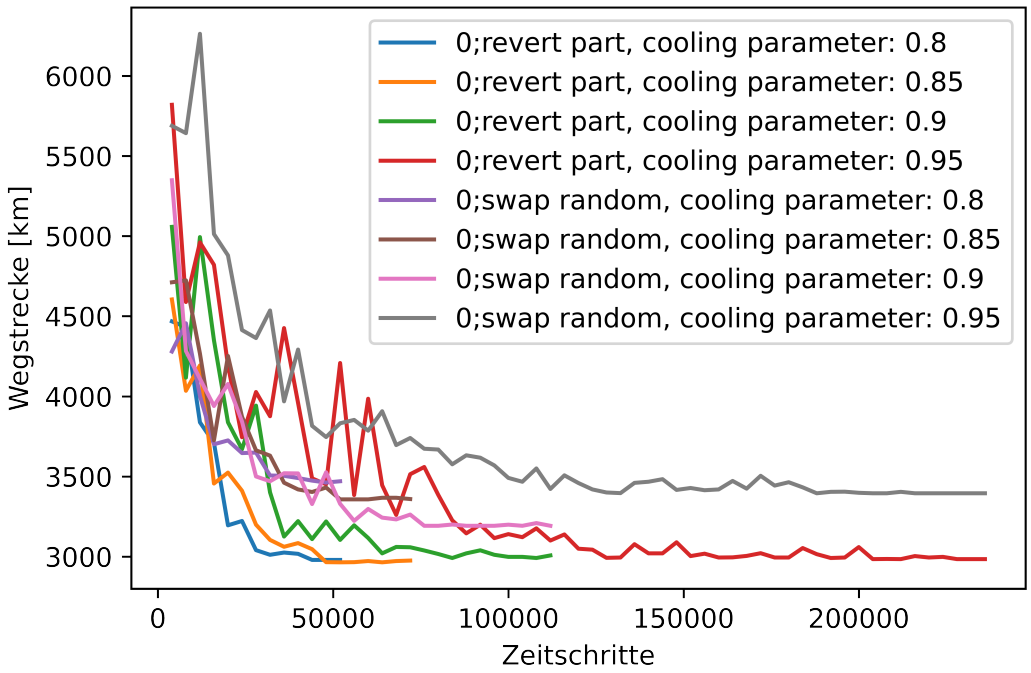

Figure 2: Verlauf der Simulationen mit verschiedenen Vertauschungsmethoden bei unterschiedlichen Abkühlungsparameter c und Abbruchwahrscheinlichkeit p_{abbr}
. Die Zeitschritte entsprechen der Anzahl an Iterationen des Metropolis Algorithmus

2.3 Verteilung der gefundenen Routenlänge

Um die Wahrscheinlichkeitsdichte der gefundenen Route zu bestimmen wurden je 200 Durchläufe der verschiedenen Vertauschungsfunktionen mit verschiedenen Parametern simuliert. Die Ergebnisse wurden dann nach ihrer Länge in ein Histogramm übertragen und normiert und so die Wahrscheinlichkeitsdichte ermittelt. Als Maß der genutzten CPU Zeit bis zur Abbruchbedingung (Kapitel 2.2) haben wir die Anzahl der Vertauschungen gezählt, da eine bloße Messung der Computerzeit abhängig von der Leistung der unterschiedlichen genutzten Computern gewesen wäre. In Kapitel 2.4 wird dann nochmal betrachtet, wie groß der Zeitunterschied eines Iterationsschritt der „swapping"Methode zu einem Schritt der „revert part" Methode ist.

Zuerst betrachten wir hierfür die „swapping" Methode, nach welcher in jeder Iteration zwei zufällige Städte vertauscht werden. Man kann in Abbildung 3 erkennen, dass diese Methode eine breite Verteilung hat und dass die Ergebnisse mit zunehmendem Abkühlungsparameter besser wird.

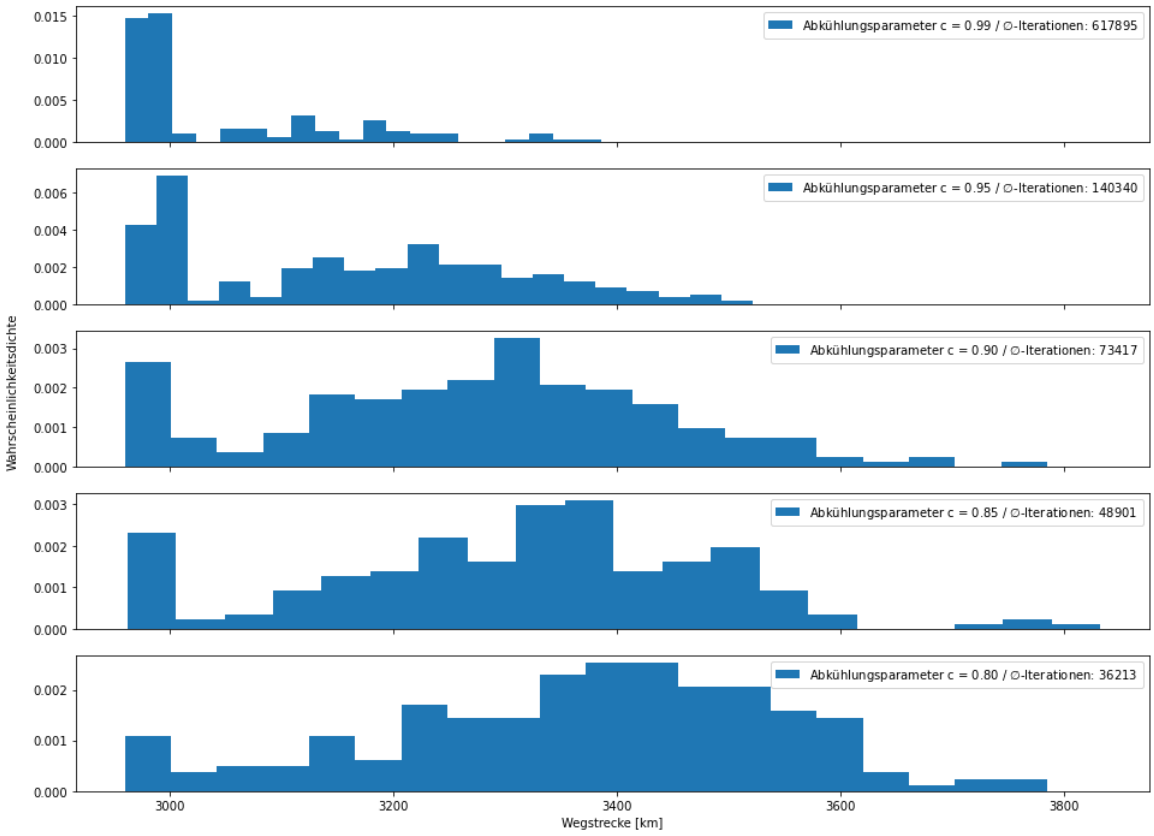

Figure 3: Die Wahrscheinlichkeitsdichte für die Vertauschungsmethode "swapping" in Abhängigkeit verschiedener Abkühlungsparameter. (Histogramm 20 Bins.)

Man kann in Abbildung 4 erkennen, dass das Vertauschungsverfahren **„part revert"** im Vergleich zum **„swapping"** Verfahren unabhängiger vom Abkühlungsparameter ist. Dies könnte mglw. daran liegen, dass bei diesem Verfahren viel mehr Änderungen der Reihenfolge durchgeführt werden und somit hier viel weniger Iterationen benötigt werden, um ein nahezu optimales Minimum zu erreichen.

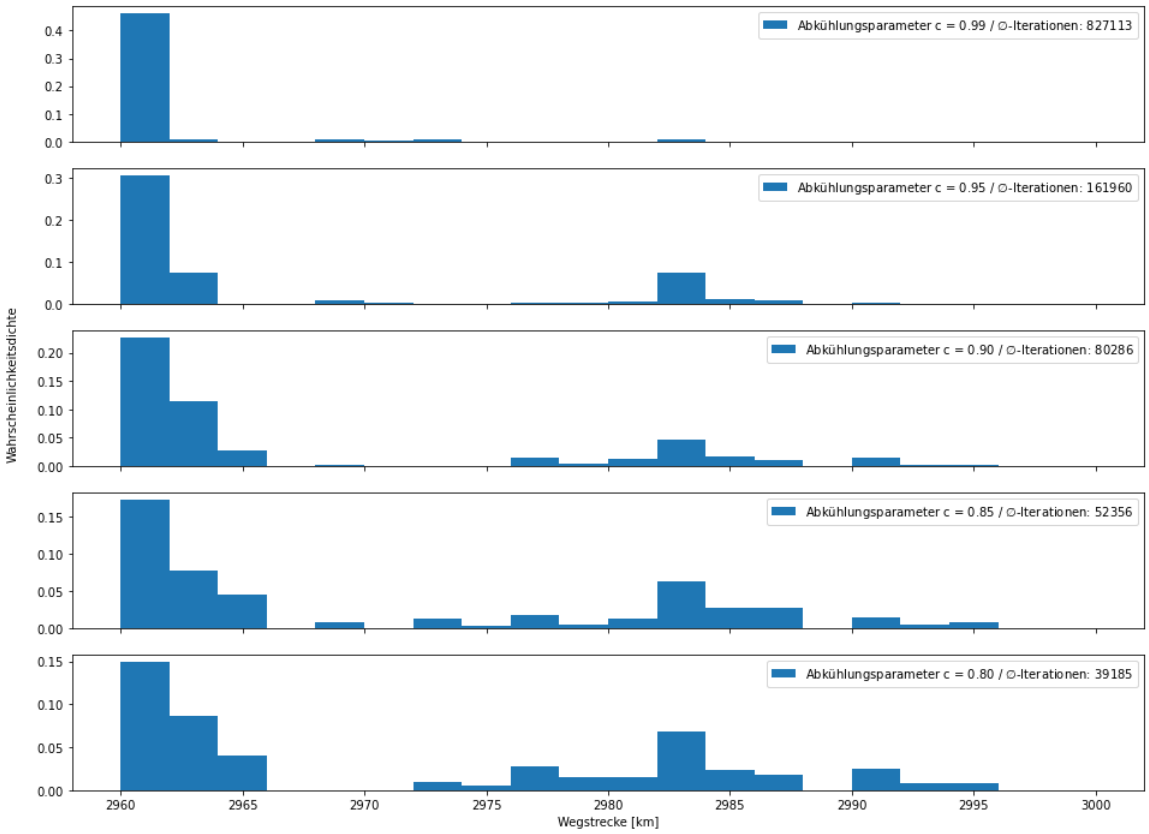

Figure 4: Die Wahrscheinlichkeitsdichte für die Vertauschung **„part revert“** in Abhängigkeit verschiedener Abkühlungsparameter. (Histogramm 20 Bins.)

Vergleichen wir nun die Verteilungen der beiden Vertauschungsverfahren (Abbildung 5), so können wir erkennen, dass das „revert part“ Verfahren für die betrachteten Abkühlungsparameter $c = 0,99$ mit einer höheren Wahrscheinlichkeit einen kürzeren Weg findet, als das „swapping“ Verfahren.

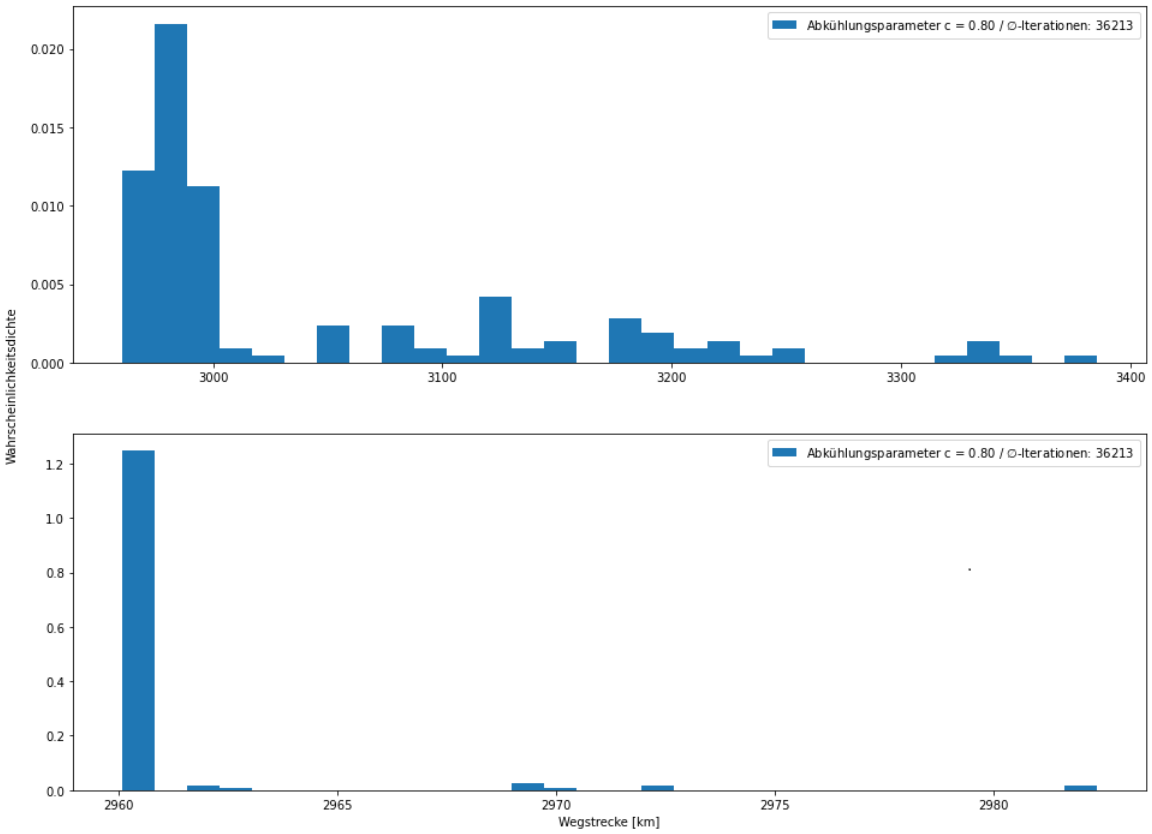

Figure 5: Die Wahrscheinlichkeitsdichte der beiden Vertauschungsverfahren im Vergleich. Dabei lag die Anfangstemperatur bei $T_0 = 70$ und der Abkühlungsparameter bei $c = 0,99$. (Histogramm 30 Bins.)

2.4 Zeiteffizienz

Da wir in unseren Auswertungen nur die Anzahl der Iterationen zur Bestimmung der benötigten Rechenzeit heranziehen, überprüfen wir im Folgenden, ob und wenn ja um welchen Faktor sich die verschiedenen Vertauschungstypen auf die benötigte Simulationszeit auswirken. Dafür wurden zwei identische Simulationen mit denselben Parametern mit dem Vertauschungsverfahren „swapping“, sowie „part revert“ durchgeführt und die Laufzeit gemessen.

Table 1: Bestimmung der Effizienz der Vertauschungsverfahren

Verfahren	Iterationen Gesamt	Zeit Gesamt	∅ Zeit / Iteration
Swap	1208000	143 Sekunden	$1,1838 \cdot 10^{-4} s$
part revert	1208000	145 Sekunden	$1,2003 \cdot 10^{-4} s$
		Faktor	1,014

Es ist zu erkennen, dass die Auswahl des Vertauschungsverfahrens keinen signifikanten Einfluss auf die ∅ Zeit pro Iteration hat. Somit ist es möglich die Anzahl der Iterationen als Maß für die Rechenintensität zum Vergleich der verschiedenen Simulationen heranzuziehen.

Wie bei dem Vergleich der durchschnittlichen CPU-Time in den Abbildungen 3 und 4 zu erkennen ist, erreichen Simulationen mit der „revert part“ Methode die Abbruchbedingung bei einem höheren Zeitschritt (Abschnitt 2.2). Dies könnte daran liegen, dass die Temperatur schneller fällt, wenn bei mehreren Schritte hintereinander ein kürzerer Weg gefunden wird. Da beim „swapping“ nur sehr kleine Schritte möglich sind und damit die Weglänge langsamer fällt, ist die Wahrscheinlichkeit, eine erfolgreiche Rekombination zu erhalten erhöht, da die Anzahl an kürzeren Kombinationen noch höher ist. Der zusätzliche zeitliche Aufwand der „revert part“ Methode im Vergleich zur „swapping“ Methode steigt mit einem größeren Abkühlungsparameter. In Tabelle 2 ist das Verhältnis des zeitlichen Aufwandes bei gleicher Abbruchbedingung aufgelistet.

Abkühlungsparameter	∅Iterationen part revert	∅Iterationen swapping	Verhältnis
0,8	39185	36213	1,08
0,85	52356	48901	1,07
0,9	80286	73417	1,09
0,95	161960	140340	1,15
0,99	827113	617895	1,34

Table 2: Durchschnittlicher Zeitlicher Aufwand bei der beschriebenen Abbruchbedingung 2.2. Angegeben ist das Verhältnis Zeitschritte beim part revert Verfahren zum swapping verfahren. Die dazugehörigen Wahrscheinlichkeitsdichten sind in Abbildung 4 und 3 zu finden.

2.5 Die kürzeste gefundene Wegstrecke

Der kürzeste Weg den wir während unserer Simulationen gefunden haben beträgt 2960km und ist in Abbildung 6 zu sehen. Im Vergleich zur Route zum Beginn der Simulation, mit einer Gesamtlänge von 10975km, konnte die Weglänge um das ca. 3,7-fache reduziert werden.

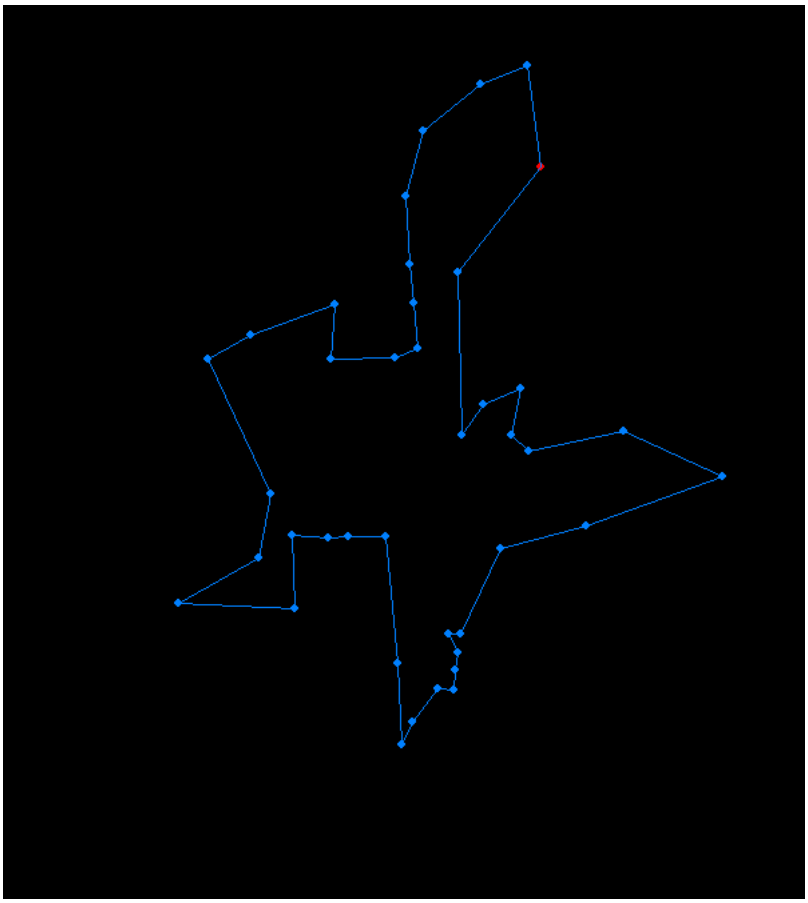

Figure 6: Der vom Programm ermittelte kürzeste Weg mit 2960km (Tabelle 4).

3 Anhang

Den Programmcode und die verwendeten Städte (cities40.dat) finden Sie unter: https://github.com/alvogel/simann

References

[Met+53] Nicholas Metropolis et al. "Equation of State Calculations by Fast Computing Machines". In: *The Journal of Chemical Physics* 21.6 (1953), pp. 1087–1092. DOI: 10.1063/1.1699114. eprint: https://doi.org/10.1063/1.1699114. URL: https://doi.org/10.1063/1.1699114.

[Pre+92] William H. Press et al. *Numerical Recipes in C (2nd Ed.): The Art of Scientific Computing*. USA: Cambridge University Press, 1992. ISBN: 0521431085.

Table 3: Startroute mit Einzelstrecken

Von	Nach	Strecke [km]
Aachen	Augsburg	439
Augsburg	Bamberg	169
Bamberg	Bayreuth	50
Bayreuth	Berlin	311
Berlin	Bielefeld	330
Bielefeld	Bochum	109
Bochum	Bonn	84
Bonn	Braunschweig	292
Braunschweig	Bremen	147
Bremen	Chemnitz	377
Chemnitz	Clausthal-Zellerfeld	210
Clausthal-Zellerfeld	Cottbus	276
Cottbus	Darmstadt	452
Darmstadt	Dortmund	201
Dortmund	Dresden	440
Dresden	Düsseldorf	487
Düsseldorf	Duisburg	24
Duisburg	Eichstätt	425
Eichstätt	Erfurt	232
Erfurt	Essen	285
Essen	Flensburg	404
Flensburg	Frankfurt/Main	522
Frankfurt/Main	Frankfurt/Oder	479
Frankfurt/Oder	Freiburg	680
Freiburg	Gera	444
Gera	Giessen	243
Giessen	Göttingen	137
Göttingen	Hagen	173
Hagen	Halle	314
Halle	Hamburg	266
Hamburg	Hannover	132
Hannover	Heidelberg	337
Heidelberg	Heilbronn	48
Heilbronn	Hildesheim	339
Hildesheim	Ilmenau	176
Ilmenau	Jena	54
Jena	Jülich	368
Jülich	Kaiserslautern	193
Kaiserslautern	Karlsruhe	67
Karlsruhe	Aachen	258
	Gesamt	10974

Table 4: kürzeste Route mit Einzelstrecken

Von	Nach	Strecke [km]
Dortmund	Bielefeld	93
Bielefeld	Bremen	119
Bremen	Flensburg	194
Flensburg	Hamburg	142
Hamburg	Hannover	132
Hannover	Hildesheim	28
Hildesheim	Braunschweig	41
Braunschweig	Clausthal-Zellerfeld	53
Clausthal-Zellerfeld	Göttingen	41
Göttingen	Halle	141
Halle	Berlin	147
Berlin	Frankfurt/Oder	88
Frankfurt/Oder	Cottbus	67
Cottbus	Dresden	89
Dresden	Chemnitz	62
Chemnitz	Gera	59
Gera	Jena	35
Jena	Erfurt	40
Erfurt	Ilmenau	32
Ilmenau	Bayreuth	96
Bayreuth	Bamberg	50
Bamberg	Eichstätt	113
Eichstätt	Augsburg	62
Augsburg	Freiburg	230
Freiburg	Karlsruhe	119
Karlsruhe	Heilbronn	61
Heilbronn	Heidelberg	48
Heidelberg	Kaiserslautern	68
Kaiserslautern	Darmstadt	79
Darmstadt	Frankfurt/Main	27
Frankfurt/Main	Giessen	53
Giessen	Bonn	113
Bonn	Aachen	72
Aachen	Jülich	25
Jülich	Düsseldorf	45
Düsseldorf	Duisburg	24
Duisburg	Essen	17
Essen	Bochum	15
Bochum	Hagen	22
Hagen	Aachen	117
	Gesamt	2960