4.2 Eigenentwicklungen

4.2.1 Start-Nachbar-Verfahren

Das Problem des Nearest-Neighbour-Algorithmus ist, daß durch die lokale Betrachtungsweise nur der jeweils nächsten einzubindenden Stadt ohne Berücksichtigung des Gesamtaufbaus der Tour die letzten Kanten und insbesondere die Rückkehr zum Anfangspunkt sehr lang werden können. Das *Start-Nachbar(SN)-Verfahren* versucht dies dadurch zu vermindern, daß nicht nur die Entfernungen zum folgenden Ort klein, sondern auch der Abstand zum ersten Ort möglichst groß sein soll, was bewirkt, daß der Weg zuerst vom Start 'weggetrieben' wird und am Ende Punkte in der Nähe der Startstadt übrigbleiben. Realisiert wird dies, indem als nächster Ort derjenige ausgewählt wird, für den der Quotient

$$\frac{\text{ENTFTAB[aktueller Ort,nächster Ort]}}{\text{ENTFTAB[nächster Ort,Anfang]}} \tag{4.3}$$

minimal ist.

4.2.2 Optimiertes Start-Nachbar-Verfahren

Im Falle einer symmetrischen Entfernungsmatrix gibt es beim Start-Nachbar-Verfahren noch ein Problem: Bei der Bestimmung der ersten Kante ist der aktuelle Ort gleich dem Anfangspunkt, so daß der Quotient (4.3) für alle Städte gleich eins ist. Daher wird in Programm 4.10 in diesem Fall immer die in der Reihenfolge erste ausgewählt. Das *Optimierte Start-Nachbar(OSN)-Verfahren* geht deshalb im ersten Schritt wie beim Nearest-Neighbour-Algorithmus zum nächstgelegenen Ort und benutzt erst danach die Start-Nachbar-Vorgehensweise.

4.2.3 Start-Nachbar-Insertion-Varianten

Minimum-SN-Insertion, Maximum-SN-Insertion

So wie das Nearest-Insertion-Verfahren den nächsten einzufügenden Ort genauso wie der Nearest-Neighbour-Algorithmus auswählt, dann aber das Insertion-Prinzip anwendet, so kann man auch aus dem Start-Nachbar-Verfahren eine Insertion-Variante (*Minimum-SN-Insertion*) ableiten, indem man in Programm 4.6 ENTFTAB[Ort aus Tour,einzufügender Ort] durch Formel (4.3) ersetzt. Das entsprechende Gegenstück zu Farthest Insertion, ist *Maximum-SN-Insertion*.

Minimum-OSN-Insertion, Maximum-OSN-Insertion

Minimum-OSN-Insertion und *Maximum-OSN-Insertion* sind die analog zum OSN-Verfahren an symmetrische Entfernungsmatrizen angepaßten Versionen von Minimum-SN-Insertion und Maximum-SN-Insertion.

Minimum-Sum-Insertion, Maximum-Sum-Insertion

Bei den bisherigen Start-Nachbar-Insertion-Variationen wurde bei der Auswahl der nächsten einzufügenden Stadt jeweils eine der Größen 'Entfernung zum vorherigen Punkt' und 'Abstand zum Startpunkt' auf Maximalität und die andere auf Minimalität hin optimiert. Ersetzt man den Quotienten (4.3) jedoch durch eine Summe, so gilt für beide die gleiche Richtung. Das Ergebnis sind die Verfahren *Minimum-Sum-Insertion*, bei dem der Ort ausgewählt wird, für den die Summe minimal ist, und *Maximum-Sum-Insertion* für den entgegengesetzten Fall.

Minimum-Product-Insertion, Maximum-Product-Insertion

Minimum-Product-Insertion und *Maximum-Product-Insertion* verfolgen die gleiche Idee wie Minimum/Maximum-Sum-Insertion, wobei hier statt der Summe das Produkt

$$\text{ENTFTAB[aktueller Ort,nächster Ort]} \cdot \text{ENTFTAB[nächster Ort,Anfang]} \tag{4.4}$$

verwendet wird.

4.2.4 Kürzeste-Wege-Rundreisen

Während bei den bisherigen Algorithmen meist der metrische Fall im Mittelpunkt stand, sind die *Kürzeste-Wege-Rundreise(KWR)-Verfahren* speziell für nichtmetrische Entfernungsmatrizen ausgelegt. Wie der Name schon sagt, benutzen sie die Bestimmung von kürzesten Wegen als Basis, und falls die Dreiecksungleichung gilt, ist die direkte Verbindung immer der kürzeste Weg. Gemeinsam ist allen Algorithmen die folgende Vorgehensweise (der Rundweg wird in W aufgebaut):

```
1: W:=Startort;
2: solange W nicht alle Städte enthält
3:  berechne kürzeste Wege vom letzten Ort in W zu allen noch nicht in W
    enthaltenen Orten;
4:  wähle einen dieser Wege aus und füge ihn an W an;
5: ende{solange}
6: W:=W+Startort;
```

Der Unterschied zwischen den verschiedenen Varianten liegt darin, wie in Schritt 4 der anzufügende Weg ausgewählt wird.

Für die Ermittlung der kürzesten Wege wird eine modifizierte Version der DIJKSTRA-Prozedur aus Programm 3.1 verwendet, die kürzeste Wege vom Endpunkt eines übergebenen Weges ohne die sonstigen darin enthaltenen Städte berechnet. Dazu wird in den Zeilen 14-22 von DIJKSTRA_W ein INDEX-Feld aufgebaut, das die Nummern der noch nicht besuchten Orte enthält und über das im eigentlichen Dijkstra-Algorithmus z. B. auf die Entfernungstabelle und die Längen zugegriffen wird.

Kürzeste-Wege-Rundreise 1

Beim ersten KWR-Algorithmus wird versucht, mit jedem hinzuzufügenden Weg möglichst viele Städte zu erfassen, d. h. es wird derjenige Weg ausgewählt, der die meisten Orte enthält. Im metrischen Fall besteht jeder natürlich nur aus einer Stadt, so daß alle gleichberechtigt sind und quasi zufällig, nämlich anhand der Reihenfolge, in der sie von DIJKSTRA_W übergeben werden, einer bestimmt wird. Entsprechend schlecht sind in diesem Fall die Resultate.

Kürzeste-Wege-Rundreise 2

Hier wird der mittlere Abstand zwischen den Städten der kürzesten Wege, also

$$\frac{\text{Weglänge}}{\text{Anzahl der Städte auf diesem Weg}}, \tag{4.5}$$

der möglichst gering sein soll, als Entscheidungskriterium verwendet. Im metrischen Fall ergibt sich daraus das Nearest-Neighbour-Verfahren.

Kürzeste-Wege-Rundreise 3, Kürzeste-Wege-Rundreise 4

Die dritte Variante versucht, mit den Endpunkten der kürzesten Wege möglichst nahe am Startpunkt zu bleiben, d. h.

$$\text{ENTFTAB[letzter Ort des Weges,Startort]} \tag{4.6}$$

soll minimal sein, während *KWR 4* in der Tradition von Farthest Insertion und dem Start-Nachbar-Verfahren das Gegenteil anstrebt und den Weg auswählt, für den der Abstand vom Wegende zum Startort maximal ist und damit die besseren Ergebnisse erzielt.

Kürzeste-Wege-Rundreise 5

Kürzeste-Wege-Rundreise 5 verbindet KWR 1 und KWR 4, indem sowohl die Anzahl der Städte des jeweiligen Weges als auch der Abstand des Endpunktes zum Startort möglichst groß sein sollen. Als Auswahlkriterium wird daher das Produkt

$$\text{(Anzahl der Städte auf dem Weg)} \cdot \text{ENTFTAB[letzter Ort des Weges,Startort]} \tag{4.7}$$

dieser beiden Größen verwendet und damit tatsächlich die Resultate von KWR 1-4 für die nichtmetrischen Fälle übertroffen.

Kürzeste-Wege-Rundreise 6

Die letzte Variante ist schließlich eine Kombination von KWR 2 und KWR 4, bei der der Abstand vom Wegendpunkt zum Tourstartpunkt maximal, der mittlere Abstand der Städte im Weg jedoch minimal sein soll. Es wird daher der Weg gesucht, für den der Quotient

$$\frac{\text{ENTFTAB[letzter Ort des Weges,Startort]}}{\text{mittlerer Stadtabstand}} \tag{4.8}$$

maximal ist. Für die nichtmetrischen Fälle erhält man ähnliche Ergebnisse wie bei KWR 5, im metrischen Fall ist der Algorithmus identisch zum Start-Nachbar-Verfahren.