

## 5.1 Klassische Verfahren

Bei den klassischen Optimierungsverfahren wird die Nachbarschaft einer Tour nach kürzeren Wegen abgesucht. Dabei kann man entweder unter allen kürzeren Wegen den mit der geringsten Länge oder direkt den ersten besseren, den man findet, auswählen und nun dessen Nachbarschaft untersuchen. Abgebrochen wird die Suche, wenn eine Tour gefunden wurde, deren Nachbarschaft keinen kürzeren Weg enthält. Anschaulich kann man sich einen Wanderer vorstellen, der von einem Berg ins Tal gelangen will und dazu sich umschaute, in welcher Richtung ein Weg nach unten führt. Entweder sieht er erst in alle Richtungen und wählt dann den steilsten Abstieg oder er nimmt sofort den ersten sich anbietenden abwärtsführenden Schritt.

### 5.1.1 2-Städtetausch

Beim *2-Städtetausch* ist die Nachbarschaft einer Tour als die Menge der Rundreisen definiert, die durch Austausch zweier Städte entstehen.

### 5.1.2 2-Kantentausch

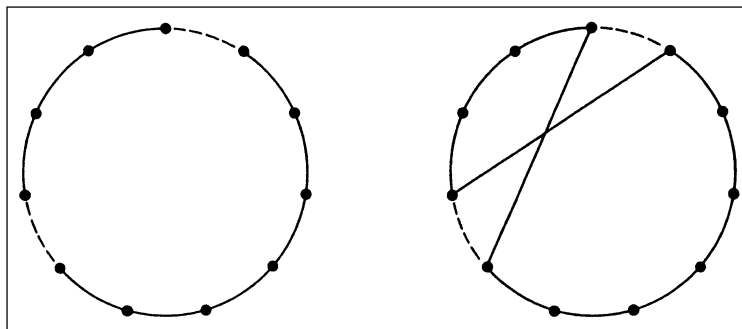


Abbildung 5.1: 2-Kantentausch <sup>1</sup>

Wie in Abbildung 5.1 dargestellt, entstehen die benachbarten Touren des *2-Kantentauschs* dadurch, daß man zwei Kanten 'aufbricht' und 'über Kreuz' wieder verbindet. Die Folge ist, daß ein Teil der Rundreise in umgekehrter Reihenfolge durchlaufen wird. Sind I und J die Schnittstellen und sei  $I < J$ , dann sieht dies z. B. folgendermaßen aus:

```

vorher:  1 2 3 | 4 5 6 | 7 8 9
           I       J
nachher:  1 2 3 | 6 5 4 | 7 8 9

```

Ist  $I > J$ , dann wird, da man die Tour als Kreis ohne Anfangs- und Endpunkt betrachtet, die Reihenfolge außerhalb des durch I und J eingeschlossenen Bereichs umgekehrt:

```

vorher:  1 2 3 | 4 5 6 | 7 8 9
           J       I
nachher:  9 8 7 | 4 5 6 | 3 2 1

```

Im Programm wird dies von der Funktion KTAUSCH (Zeile 4-16) durchgeführt. Den 'Kreischarakter' erhält man durch die Verwendung des MOD-Operators.

<sup>1</sup>M. M. Syslo: Discrete Optimization Algorithms, Prentice-Hall, 1983

### 5.1.3 3-Kantentausch

Im Gegensatz zum 2-Kantentausch werden hier drei Kanten aus der bestehenden Tour entfernt und die entstehenden Teilwege neu verknüpft (Abb. 5.2). Dabei gibt es die in Abbildung 5.3 dargestellten acht Möglichkeiten. Nr. 1 ist wieder die Ausgangstour, während 2-4 2-Kantentausche darstellen. 5-7 sind bis auf Drehung identisch, d. h. man kann 6 und 7 durch Umbenennen der Knoten in 5 erhalten, und brauchen deshalb nicht unterschieden zu werden. Es genügt also, Nr. 5 und die noch übriggebliebene Möglichkeit 8 zu untersuchen. Bei der ersten wird ein Teil der Tour in umgekehrter Richtung durchlaufen, weswegen diese Variante auch als *asymmetrisch* bezeichnet wird, während im anderen Fall die Richtungen erhalten bleiben ( $\rightarrow$ *symmetrisch*).

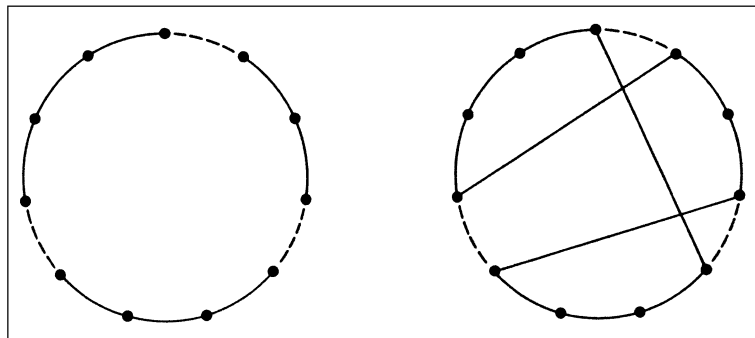


Abbildung 5.2: 3-Kantentausch <sup>1</sup>

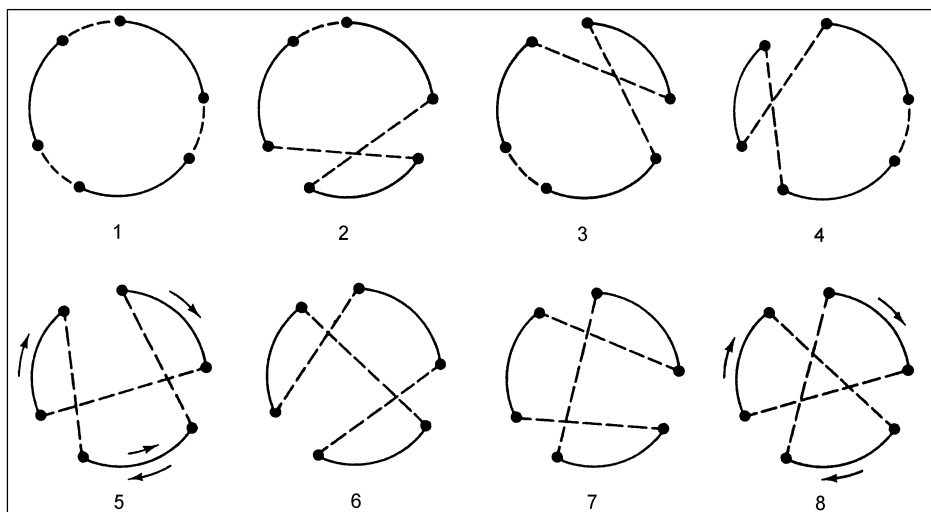


Abbildung 5.3: Verknüpfungsmöglichkeiten beim 3-Kantentausch <sup>1</sup>

Zuerst werden die drei potentiell auszutauschenden Kanten bestimmt und für jede der beiden Möglichkeiten die sich ergebende Verkürzung berechnet, die günstigere davon wird ausgewählt. Dies geschieht für alle in Frage kommenden Kantentriplets der aktuellen Tour. Ist mit der besten sich daraus ergebenden Änderung eine Verkürzung zu erzielen, so wird diese tatsächlich durchgeführt und die Routine wiederholt, ansonsten hat man ein (lokales) Optimum erreicht und der Algorithmus wird beendet.

<sup>1</sup>M. M. Syslo: Discrete Optimization Algorithms (Prentice-Hall, 1983)

In Programm 5.5, das auf einer entsprechenden Prozedur aus *M. M. Syslo: Discrete Optimization Algorithms (Prentice-Hall, 1983)* beruht, erfolgt die Auswahl der beiden Möglichkeiten durch die Prozedur TAUSCHTEST, die das Ergebnis in einem record vom Typ TAUSCHREC übergibt. Dieser enthält die auszutauschenden Kanten, die Änderung der Weglänge, die sich dadurch ergibt, und die Art des Tauschs (true: asymmetrisch, false: symmetrisch). Um die Berechnung der Längenänderungen einfach zu halten, ist sie nur für symmetrische Kostenmatrizen ausgelegt; das Programm eignet sich also nicht für asymmetrische TSPs. Wurde der asymmetrische Kantentausch gewählt, so kehrt UMKEHREN die Durchlaufrichtung des betroffenen Teilstücks um.

3-Kantentausch liefert im gesamten bessere Ergebnisse als die 2-Städte-/2-Kantentausch-Verfahren, allerdings auch mit einem höheren Rechenaufwand.

#### 5.1.4 2+3-Kantentausch

Das Programm DKTAUSCH.PAS des vorigen Abschnittes betrachtet nur die 'echten' 3-Kantentausche, nicht aber die 'entarteten' Fälle, welche 2-Kantentauschen entsprechen. Diese lassen sich in die Berechnung einbeziehen, wenn man eine Kombination mit den 2-Kantentausch-Routinen aus Abschnitt 5.1.2 vornimmt: Programm 5.6 ruft abwechselnd die Prozeduren KTAUSCH1 (2-Kantentausch/steilster Abstieg) und DKTAUSCH (3-Kantentausch) auf, bis eine der beiden keine Verbesserung mehr bewirkt; eine Fortsetzung würde bedeuten, daß die jeweils andere mit ihrem eigenen Ergebnis als Starttour begänne. Die Initialisierung von M mit MAXENTFERNUNG in Zeile 116 bewirkt, daß DKTAUSCH mindestens einmal aufgerufen wird, selbst wenn KTAUSCH1 bereits im ersten Durchgang keine bessere Tour liefert.

Statt des 2-Kantentauschs mit steilstem Abstieg kann man auch die Erste-Schritt-Methode verwenden, indem die KTAUSCH1-Prozedur durch KTAUSCH2 aus Programm 5.4 ersetzt wird.

#### 5.1.5 n-Kantentausch

Das Prinzip des Kantentauschs läßt sich auch auf vier und mehr Kanten ausweiten, wobei aber der damit verbundene Aufwand entsprechend wächst. Nach verschiedenen Untersuchungen liefert der 3-Kantentausch merklich bessere Resultate als der 2-Kantentausch, während der 4-Kantentausch keine bedeutenden Verbesserungen mehr bringt.