

# **Traveling Salesman Problem (TSP)**

Einführung in das Traveling Salesman Problem  
und ausgewählter Lösungsansätze



## Inhaltsverzeichnis

Einführung.....	2
Gegenstand und Überblick der Arbeit.....	2
Historische Entwicklung.....	2
Charakterisierung des Traveling Salesman Problems.....	3
Allgemeine Definition und Problembeschreibung.....	3
Variationen & Familie der TSP.....	4
Klassifikation des Raumes.....	5
Das zweidimensionale euklidische sTSP.....	6
Mathematische Modelle.....	7
Als Problem ganzzahliger linearer Optimierung (LOP).....	7
Graphentheoretisches Modell.....	9
Ansätze zur Lösung.....	10
Exakte Verfahren.....	10
Vollständige Enumeration.....	10
Entscheidungsbaumverfahren (implizite Enumeration).....	11
Verkürzte Enumeration.....	11
Dynamische Optimierung.....	11
Branch & Bound.....	11
Heuristiken.....	13
Eröffnungsverfahren.....	13
Nearest Neighbor.....	13
Nearest Selection (Floyd-Warshall-Algorithmus).....	13
Minimum Spanning Tree (Prim-Algorithmus).....	14
Approximative Verfahren.....	14
Kantentausch (k-opt Verfahren).....	15
Metaheuristiken.....	15
Tabu Search.....	16
Simulated Annealing.....	16
Ameisenalgorithmen.....	17
Anhang I: Abbildungsverzeichnis.....	18
.....	18
Anhang II: Literaturverzeichnis.....	18

# 1. Einführung

## 1.1. Gegenstand und Überblick der Arbeit

Ziel und Gegenstand dieser Arbeit ist die knappe Analyse des „*Problem des Handlungsreisenden*“ (auch als Boten- oder Rundreiseproblem bezeichnet, eng. „*Traveling Salesman Problem*“, TSP) sowie verschiedener Verfahren zur optimalen Lösung bzw. einer guten Approximation derer. Dabei soll insbesondere die Hartnäckigkeit mit welcher sich das TSP einer Lösung widersetzt betont werden, da hier zweifelsohne ein Hauptgrund für das wissenschaftliche Interesse liegt. Schließlich kann das TSP als eine der bekanntesten und am meisten beachteten kombinatorischen Problemstellungen des 20. Jahrhunderts bezeichnet werden. 1991 wird es im Wissenschaftsteil der New York Times sogar als „*The mother of all Algorithmic Problems*“ bezeichnet. Auch entstand im Angloamerikanischen der Ausspruch „*it's so hard to solve like the Traveling Salesmanproblem*“, der die Ausichtslosigkeit zur Lösung eines Problems untermalen soll. Der jedoch vielleicht noch bedeutendere Aspekt ist das breite Anwendungsspektrum das im Laufe der Arbeit angedeutet werden soll.

Zunächst wird aber ein knapper Überblick über historische wissenschaftliche Arbeiten gegeben um zu einer raschen Begriffsbildung zu führen. Im zweiten Abschnitt „*Charakterisierung des Traveling Salesman Problem*“ wird dabei vorerst ein allgemeine Definition des Problems, sowie ein kurzer Einblick über die Vielzahl seiner möglichen Derivate gegeben. Desweiteren wird der zugrunde liegende Raum charakterisiert um schließlich einen exakten Definitionsbegriff des „*Euklidischen zweidimensionalen sTSP*“ zu erhalten, mit dem im weiteren Kontext gearbeitet werden kann. Im Abschnitt 3 „*Ansätze zur Lösung*“ werden nun schrittweise unter einer Auswahl eingehender und in der Literatur häufig verwendeter Algorithmen exakte, heuristische und deren Abstraktion durch metaheuristische Ansätze zur Lösung des Problems entwickelt, untersucht und verglichen.

## 1.2. Historische Entwicklung

Es ist bemerkenswert, dass die Ursprünge dieses Problems bzw. Die Frage, wann es erstmalig mit wissenschaftlicher Ernsthaftigkeit untersucht wurde, etwas unklar sind. Obgleich dem TSP ähnliche bzw. Identische Problemstellungen der Reihung von Objekten sicherlich seit Jahrhunderten bzw. Jahrtausenden im Alltag der Menschen aufgetreten sind, so wurden doch alle wesentlichen algorithmischen Ansätze zur Lösung des TSP erst in den letzten vier Dekaden eingeführt. Zwar können beispielsweise das „*Königsberger Brückenproblem*“ - von Leonard Euler (1736) formuliert – oder das „*Icosian Game*“ des irischen Mathematikers Hamilton aus dem 19. Jahrhundert als frühe Vorgänger des TSP angesehen werden - die erste konkrete Erwähnung einer sehr ähnlichen Problemstellung scheint jedoch auf *Karl Menger* (1932) zurückführbar zu sein:

*„Wir bezeichnen als Botenproblem (weil diese Frage in der Praxis von jedem Postboten, übrigens auch von vielen Reisenden zu lösen ist) die Aufgabe, für endlich viele Punkte, deren paarweise Abstände bekannt sind, den kürzesten die Punkte verbindenden Weg zu finden. Dieses Problem ist natürlich stets durch endlich viele Versuche lösbar. Regeln, welche die Anzahl der Versuche*

*unter die Anzahl der Permutationen der gegebenen Punkte herunterdrücken würden, sind nicht bekannt. Die Regel, man sollte vom Ausgangspunkt erst zum nächstgelegenen Punkt, dann zu dem diesen nächstgelegenen Punkt gehen usw., liefert im Allgemeinen nicht den kürzesten Weg.“*

Nach Flood (1956) – von Hoffman/Wolfe (1985) aufgrund seiner Publikationen und seines Wirkens hinsichtlich der Verbreitung der Problemstellung in der Operations-Research Gemeinde in den 40er und 50er Jahren als „*Botschafter des TSP*“ bezeichnen – wurde das TSP erstmalig in einem Seminargespräch von Whitney an der Princeton University im Jahre 1931 oder 1932 als „*Traveling Salesman Problem*“ erwähnt.

Ob die Wurzeln des TSP nun in Europa oder den USA zu identifizieren sind, mag dahingestellt sein – anhand der zunehmenden Veröffentlichungen in den 50er und 60er Jahren lässt sich unabhängig davon seine wachsende Popularität verfolgen. Besonderer Erwähnung bedürfen dabei die Beiträge von Dantzig/Folkerson/Johnson (1954) als einer der Ursprünge der polyedrischen Kombinatorik und des „*Branch & Bound*“ (siehe 3.1.2.3) Verfahrens sowie der Weiterentwicklung durch Little/Murty/Sweeney/Karel (1963)

Seit den 70er bis 80er Jahren entwickelt sich das TSP zu einem Musterbeispiel interdisziplinären wissenschaftlichen Arbeitens wobei sich in der Literatur Publikationen von Mathematikern, Informatikern aber auch Physikern, Biologen, Chemikern und anderer diverser Disziplinen finden. Dies führt zu einer gegenseitigen Befruchtung der Arbeit wie am Beispiel verschiedener metaheuristischer Verfahren wie „*Simulated Annealing*“ (siehe 3.3.2), „*Tabu Search*“ (siehe 3.3.1) oder „*Ameisensystemen*“ (siehe 3.3.3) offensichtlich wird.

Somit kann zusammenfassend konstatiert werden, dass das TSP nicht nur aufgrund seiner intellektuellen Herausforderung sowie seiner praktischen Relevanz ein attraktives Arbeitsgebiet darstellt – es bietet trotz umfangreicher Forschung und tausender Publikationen immer noch Möglichkeit neues Wissen zu erschließen und innovative interdisziplinäre Lösungsansätze zu entwickeln.

## 2. Charakterisierung des Traveling Salesman Problems

### 2.1. Allgemeine Definition und Problembeschreibung

Das klassische TSP ist seiner Natur nach ein knotenorientiertes Reihenfolgeproblem. Dabei beschreiben die „Knoten“ eine endliche Anzahl von gleichartigen paarweise bewerteten Objekten, die unter Berücksichtigung einer zu minimierenden Zielfunktion in einer zyklischen Sequenz<sup>1</sup> angeordnet werden. Hierbei wird der Betrag der Zielfunktion um den von zwei direkt folgenden Objekten zugeordneten Wert erhöht.

Im Kontext des Handlungsreisenden können die Objekte als Städte und deren Anordnung als Rundreise interpretiert werden. Die Erhöhung des Zielwertes  $d_{ij}$  entspricht der Entfernung, den Reisekosten oder der Reisezeit zweier direkt aufeinander folgender Städte. Gesucht ist nun diejenige Permutation  $p$  der Städte, welche die Zielfunktion

$$\left( \sum_{i=1}^{n-1} d(p_{(i)}, p_{(i+1)}) \right) + d(p_{(n)}, p_{(1)})$$

minimiert. Dabei werden im ersten Teil des Ausdruckes die aus einer Permutation

---

<sup>1</sup> Von einer zyklischen Sequenz spricht man, wenn die Sequenz zyklisch geschlossen durchlaufen werden kann.

resultierend angeordneten Objekte (Städte) paarweise bewertet (z.B. Distanz) aufsummiert und im zweiten Teil die durch die Abfolge bedingte Bewertung zwischen dem ersten und letzten Objekt hinzuaddiert um den geforderten Zyklus zu schließen.

Bei der Untersuchung dieser Problemstellung stellt sich dabei immer unmittelbar die Frage nach der Lösbarkeit, bzw der möglichen Effizienz von allgemeinen Lösungsansätzen. Daher werden die Problemstellungen in verschiedene Klassen eingeordnet. Nachgewiesenermaßen liegt das vorhergehend beschriebene TSP nach Karp (1972) in der Klasse der NP-Vollständigen („*non polynomial*“) Probleme. Dies bedeutet, dass bisher kein effizienter (polynomialer) Algorithmus zur Lösung dieses Problems bekannt ist.

## 2.2. Variationen & Familie der TSP

Das im vorhergehenden Abschnitt beschriebene TSP entspricht der klassischen Definition dieser Problemstellung, wie sie üblicherweise bei der Einführung der selben präsentiert wird. Im realweltlichen Kontext existiert aber eine nahezu unerschöpflich große Auswahl an möglicher und beliebig kombinierbarer Variationen des klassischen TSP, so dass faktisch eher von einer Familie der TSP als von einem TSP gesprochen werden muss.

Entsprechende Modifikationen des klassischen TSP können sowohl als Veränderung der grundlegenden Formulierung, als auch als zusätzliche Nebenbedingungen auftreten. Dabei lassen sich viele wieder auf ein klassisches TSP zurückführen. Da gezeigt werden kann, dass sich mit polynomialen Aufwand sämtliche NP-Vollständigen Probleme ineinander transformieren lassen und polynomial verlustfrei transformierte NP-Vollständige Probleme wiederum in der Klasse NP liegen existiert bisher auch für die auf das klassische TSP rückführbare Probleme kein effizienter Lösungsansatz.

Einige explizite und in der Literatur häufig verwendete Beispiele sind das „*Multiple Traveling Salesman Problem*“ (MTSP) oder das „*Single Vehicle Routing Problem*“ (VRP). Andere wiederum können als größtenteils unabhängige Varianten und eigenständige Mitglieder der Familie der TSP betrachtet werden, wie beispielsweise das „*Bottleneck TSP*“ (BTSP) das in der Klasse P (polynomial lösbar) liegt. Darüber hinaus reicht die Verwandtschaft bis hin zu völlig unabhängigen Problemstellungen und gedanklichen Konstrukten, wie etwa aus der Linearen Optimierung („*Lineare Programmierung*“, LP) (siehe 2.5.2)

### *Multiple Traveling Salesman Problem (MTSP)*

Eine Erweiterung des Problems des Handlungsreisenden ist das Multiple Traveling Salesman Problem. Dieses verallgemeinert das klassische TSP indem es auf die Einschränkung eines einzelnen Handlungsreisenden verzichtet. Dabei wird den  $n$  - entweder sämtlich („*nonlazy Salesman*“) oder nur zum Teil eingesetzten - Reisenden ein gemeinsamer Start- und Zielknoten (Depot) zugeordnet. Ziel ist es, dass alle Städte einmal von genau einem Handlungsreisenden besucht werden, und dabei die aufsummierten Distanzen (Reisekosten, Reisezeiten, ...) aller Handlungsreisenden zusammen einen minimalen Wert annehmen.

### *Single Vehicle Routing Problem (VRP)*

Das MTSP lässt sich zum Single Vehicle Routing Problem erweitern, indem Kapazitäten und Bestell umfänge hinzugefügt werden. Nachwievor gibt es genau ein Depot, in dem alle Touren starten und enden. Ein Kunde soll von genau einer Tour beliefert werden, auf der die Summe der ausgelieferten Bestellmengen nicht größer sein darf, als die Kapazität

des Transporters erlaubt. Außerdem darf eine Tour nicht länger sein als die maximale Fahrleistung, die ein Transporter aufbringen kann. Angestrebt wird dabei die Minimierung der aufsummierte Länge der Touren aller Transporter.

#### *Bottleneck TSP (BTSP)*

Einen effizient lösbaren Sonderfall beschreibt das Bottleneck TSP. Hier soll unter Veränderung der Zielfunktion nun nicht mehr die gesamte Rundreise, sondern nur die maximale in der Rundreise passierte Kante minimiert werden. Dies bewirkt eine möglichst gleichmäßige Verteilung bzw. Glättung der Distanzen (Reisekosten, Reisezeiten, ...). Eine denkbare Variante um Engpässe zu verhindern wäre die Maximierung der minimalen Kante.

### 2.3. Klassifikation des Raumes

Das klassische TSP lässt sich durch den zugrunde liegenden Raum der Bewertungen („Gewichtungen“) klassifizieren. Hierzu werden die Relationen zwischen den Objekten  $v_i$ ,  $v_j$  und ihrer paarweisen Bewertung  $d_{ij}$  – im weiteren Kontext als Distanz bezeichnet – untersucht. Als Voraussetzung wird dabei die Definition des klassischen TSP verwendet, und schrittweise mit Bedingungen an die Distanzen erweitert.

#### *Nichtnegativität*

Zu Beginn gehen wir davon aus, dass ein klassisches TSP mit einem einzelnen Handlungsreisenden (sTSP, single TSP) vorliegt. Um das Optimierungsproblem überhaupt lösen zu können fordern wir o.B.d.A<sup>2</sup>, dass alle Distanzen nicht negativ seien und die Distanz eines jeden Objektes in Bezug auf sich selbst neutral zur Addition ist..

*Nichtnegativitäts und Identitätsbedingung für Distanzen:*

$$d_{i,j} \in \begin{cases} \mathbb{R}_0^+, & \text{für } i \neq j \\ \{0\}, & \text{für } i = j \end{cases}$$

#### *Symmetrie*

Bei der weiteren Betrachtung der Distanz zweier Objekte kann zudem unterschieden werden, ob die Reihenfolge der beiden Objekte zur Bildung der Distanz beiträgt, oder vernachlässigt werden kann. Hierzu wird eine Matrix („Distanzmatrix“ od. „Gewichtmatrix“) betrachtet, deren Zeilen und Spalten jeweils durch alle betrachteten Objekte indiziert, die Distanzen zwischen diesen Objekten enthält. Die Neutralität der paarweisen Reihenfolge der Objekte zur Bildung der Distanz, kann damit leicht durch die Symmetrie der Matrix verifiziert werden. Dabei erweisen sich asymmetrische Probleme allerdings grundsätzlich als in symmetrische Probleme überführbar bzw. als solche lösbar.

*Symmetriebedingung für die Distanzmatrix:*

$$d_{i,j} = d_{j,i} \quad \forall i, j \in \{1, 2, \dots, n\}$$

Am Beispiel des Handlungsreisenden könnte hier der Abstand zweier Städte für die Neutralität der Reihenfolge herangezogen werden. Sollte jedoch die Reisezeit betrachtet werden kann es möglich sein dass sie von Stadt A nach Stadt B kürzer ist (z.B. bessere

---

2 Die Nichtnegativitätsbedingung kann gegebenenfalls durch eine Transformation der Distanzmatrix realisiert werden

Verkehrswege) als von Stadt B nach Stadt A und eine nicht symmetrische Distanzmatrix wäre die Folge.

#### *Erfüllung der Dreiecksungleichung*

Hinsichtlich symmetrischer Probleme kann weiterhin differenziert werden, ob die sog. „Dreiecksungleichung“ erfüllt ist. Die vorherige Betrachtung wird hierzu um ein zusätzliches drittes Objekt erweitert. Dabei wird die Distanz der ursprünglichen beiden Objekte  $v_i, v_j$  mit der aufsummierten Distanz über das dritte Objekt  $v_z$  verglichen.

*Dreiecksungleichung:*

$$d_{i,j} \leq d_{j,z} + d_{z,j} \quad \forall i, j, z \in \{1, 2, \dots, n\}$$

Veranschaulicht bedeutet dies, dass der Handlungsreisende bei einem Umweg über eine dritte Stadt stets eine größere Distanz zurücklegt als wenn er die direkte Verbindung zweier Städte wählt.

#### *Metrik*

Insofern die Distanzen unseres betrachteten TSP sämtlich den Nichtnegativitäts und Identitätsbedingung genügen, die Dreiecksungleichung erfüllen und eine symmetrische Distanzmatrix vorliegt, heißt diese bei der Existenz einer einheitlichen Berechnungsvorschrift zur Ermittlung der Distanzen, metrisch.

*Metrische Distanzmatrix (m - Dimensionen):*

$$d_{i,j} = f(\vec{x}_i, \vec{x}_j) \quad \forall i, j \in \{1, 2, \dots, n\}, \vec{x}_i, \vec{x}_j \in \mathbb{R}^m$$

Dabei werden die zugrunde liegenden Berechnungsvorschriften verschiedenen Metriken zugeordnet. Gängige in Verbindung mit dem TSP verwendete Metriken sind die euklidische Metrik, die City-Block-Metrik (auch „Manhattan-Metrik“ bzw. „Taxi-Metrik“) und die Maximum-Metrik (auch „Chebyshev-Metrik“). Diese sind Spezialfälle der sog.  $L_p$ -Norm („Minkowski-Metrik“)

*Minkowski Metrik (m - Dimensionen):*

$$d_{i,j} = \left[ \sum_{k=1}^m |x_{k,i} - x_{k,j}|^p \right]^{\frac{1}{p}} \quad \forall i, j \in \{1, 2, \dots, n\}, \vec{x}_i, \vec{x}_j \in \mathbb{R}^m$$

In der  $L_p$ -Norm hat  $p$  den Wert „1“ für die City-Block-Metrik, „2“ für die euklidische Metrik und geht gegen unendlich bei der Maximum-Metrik.

## 2.4. Das zweidimensionale euklidische sTSP

Im weiteren Verlauf der Arbeit und im Hinblick auf die Anwendbarkeit der im 3. Abschnitt vorgestellten Lösungsverfahren soll nun das zweidimensionale euklidische TSP für einen einzelnen Handlungsreisenden ohne zusätzliche Beschränkungen eingeführt werden.

Die zu berücksichtigenden Objekte liegen in einem zweidimensionalen Raum mit einem kartesischen Koordinatensystem. Bei letzterem stehen die zwei Achsen rechtwinklig aufeinander und sind bezüglich ihrer Einheiten linear äquidistant normiert. Die  $n$  Objekte des TSP verteilen sich auf den 1. Quadranten des Koordinatensystems, sind also

sämtlich über positive Koordinaten der Ordinate und der Abszisse (x,y) identifizierbar. Die Distanzen der Objekte untereinander werden nach der euklidischen Metrik bestimmt und ergeben sich bei der Festlegung ihrer Koordinaten.

$$d_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

Aus der Definition der euklidischen Metrik ergibt sich unmittelbar die Symmetrie der Distanzmatrix, sowie die Einhaltung der Dreiecksungleichung.

Auch das zweidimensionale euklidische TSP erweist sich nach Padimitriou/Steiglitz (1976) im allgemeinen als NP-vollständig. Mit polynomialem Aufwand lösbare Sonderfälle zeichnen sich zumeist durch eine spezielle Struktur der Lage der Objekte in der Ebene aus. Liegen diese z.B. sämtlich auf dem Rand der von ihnen erzeugten konvexen Hülle, so ergibt sich die optimale Lösung direkt aus der Folge der Objekte auf derselben. Daraus resultierend erscheinen jene Objekte, die auf dem Rand einer konvexen Hülle liegen in der optimalen Lösung wiederum genau in der Reihenfolge der Lage auf der konvexen Hülle (auch wenn zwischen ihnen noch weitere Städte platziert sein mögen). Eine weitere Folgerung ist dass sich die Verbindungen zwischen den Objekten in einer optimalen Rundreise nie kreuzen, denn wäre dem so, dann könnten die Objekte auf der konvexen Hülle nicht mehr in identischer Reihenfolge in ihr erscheinen.

Das zweidimensionale euklidische TSP scheint seit der Formulierung des Problems eine besondere Anziehungskraft auf die Menschen und Institutionen, die sich mit der Problemstellung auseinandersetzen ausgeübt zu haben. Dies liegt zweifelsohne in der geometrischen Anschaulichkeit die dem menschlichen Denken entgegenkommt. Darüber hinaus sind aber die meisten dafür entwickelten Verfahren über seine enge Verwandtschaft zu anders metrischen TSPs (z.B. „Chebyshev-TSP“ mit Maximum-Metrik) über die Minkowski Metrik auf diese leicht übertragbar.

## 2.5. Mathematische Modelle

### 2.5.1. Als Problem ganzzahliger linearer Optimierung (LOP)

Wie in Abschnitt 2.2 angedeutet lässt sich ein TSP als ganzzahliges lineares Programm beschreiben. Dies ist durch den linearen Charakter der Zielfunktion und der Nebenbedingungen möglich, die sämtlich als lineare Gleichungen oder Ungleichungen fassbar sind. Werden die Verbindungen aller Städte zueinander zugelassen ergibt sich damit der folgende Ansatz:

*Unbeschränktes TSP als lineares Programm:*



$$\left( z = \sum_{i=1}^n \sum_{j=1}^n d_{i,j} b_{i,j} \right) \rightarrow \min \quad (a)$$

$$\sum_{j=1}^n b_{i,j} = 1 \quad \forall i \in \{1, \dots, n\} \quad (b)$$

$$\sum_{i=1}^n b_{i,j} = 1 \quad \forall j \in \{1, \dots, n\}$$

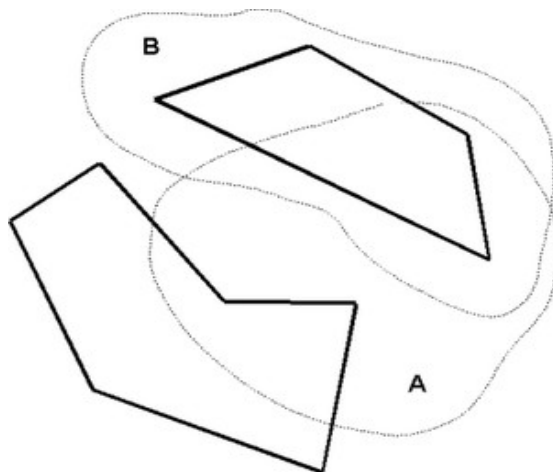
$$\sum_{i,j \in S} b_{i,j} \leq |S| - 1 \quad \forall S \subset \{1, \dots, n\} \text{ mit } 2 \leq |S| \leq n-2 \quad (c)$$

$$b_{i,j} \in \{0, 1\} \quad \forall i, j \in \{1, \dots, n\}, i \neq j \quad (d)$$

In der Zielfunktion (a) bezeichnet die Variable  $d_{i,j}$  wieder die Distanz der Objekte  $v_i$  und  $v_j$  und  $b_{i,j}$  eine binäre Variable, die wie aus (d) ersichtlich nur die Werte „0“ und „1“ annehmen kann. Sie gibt Information darüber, ob die Distanz  $d_{i,j}$  Bestandteil der zu ermittelnden optimalen Rundreise sein soll ( $b_{i,j} = 0$ ) oder nicht ( $b_{i,j} = 1$ ). Bei der Aufsummierung der Distanzen werden somit nur noch solche berücksichtigt, die tatsächlich in der Rundreise enthalten sind und bei einer optimalen Rundreise zu einer Minimumsbildung der Zielfunktion führen.

Im weiteren muss gesichert werden, dass eine zulässige Rundreise durch den Ansatz bestimmt wird. Die Bedingungen (b) legen in diesem Sinne zunächst fest, dass jedes Objekt  $v_i$  nur einmal als linker Parameter („Startpunkt“) und nur einmal als rechter Parameter („Endpunkt“) einer Distanz auftritt und somit genau über zwei Distanzen mit anderen Objekten verbunden ist.

Die bisherige Formulierung des Problems entspricht dem klassischen linearen Zuordnungsproblem. Jedoch wäre bei einer Beschränkung auf die bisherigen Bedingungen die Bildung von Subzyklen<sup>3</sup> erlaubt. Dies wird durch die sogenannte Zyklusbedingung (c) unterbunden. Zuerst werden die Indizes einer beliebigen Teilmenge unserer betrachteten Objekte fixiert und danach die Anzahl der fixierten Objekte mit der Anzahl der in der Rundreise von diesen Objekten bestimmten Distanzen verglichen.



$$S = A: \sum_{i,j \in A} b_{i,j} = 2 \leq |A| - 1 = 3$$

$$S = B: \sum_{i,j \in B} b_{i,j} = 4 > |B| - 1 = 3$$

Abbildung 1: Sybzyklen

<sup>3</sup> Unzusammenhängende bzw. isolierte geschlossene Rundreisen, welche jeweils nicht überlappende Teilmengen der Gesamtzahl der Objekte umfassen. (siehe Abbildung 1)

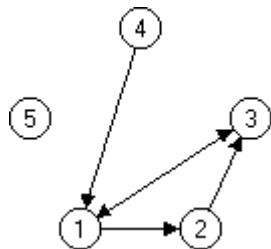
Sollte die Anzahl der Distanzen nicht echt kleiner als die Anzahl der Objekte sein, so enthalten die von unserer Menge bestimmten Objekte unter der Erfüllung der Voraussetzungen in (b) mindestens zwei Subzyklen und unsere Voraussetzung wird nicht erfüllt. Dieser Vergleich wird für alle Indexteilmengen durchgeführt, die mindestens zwei und höchstens  $n - 2$  Elemente enthalten.

Im Rahmen der obigen Formulierung des TSP als lineares Programm werden  $n \times (n - 1)$  binäre Variablen Variablen und  $2n$  Restriktionen des Typs (b) verwendet. Die Anzahl der Zyklusbedingungen (c) zeigt sich aber als nicht polynomial von der Größe des TSP abhängig – ihre Anzahl wächst mit steigendem  $n$  exponentiell.

## 2.5.2. Graphentheoretisches Modell

Ein (gerichteter) Graph ist ein Paar  $G = (V, E)$ , hierbei ist  $V$  eine endliche Menge von Knoten und  $E \subseteq V \times V$  eine Relation auf  $V$ , die Menge der Kanten. Die Knoten werden als Punkte oder Kreise gezeichnet, die Kanten als Pfeile, wobei ein Pfeil vom Knoten  $u \in V$  zum Knoten  $v \in V$  zeigt, wenn  $(u, v) \in E$  (siehe Abbildung 2). Ein ungerichteter Graph lässt sich als Spezialfall eines gerichteten Graphen auffassen, nämlich als ein gerichteter Graph, bei dem die Kanten stets in beide Richtungen verlaufen.

Der Außengrad  $o(u)$  eines Knotens  $u$  ist gleich der Anzahl der Knoten  $v$ , die von  $u$  direkt erreichbar sind, d.h.  $o(u) = |\{v \mid (u, v) \in E\}|$ . Der Innengrad  $i(u)$  eines Knotens  $u$  ist gleich der Anzahl der Knoten  $v$ , von denen er direkt erreichbar ist, d.h.  $i(u) = |\{v \mid (v, u) \in E\}|$ . Für einen ungerichteten Graphen  $G$  ist der Außengrad eines Knotens immer gleich seinem Innengrad und wird als Grad des Knotens bezeichnet.



$$\begin{aligned}
 G &= (V, E) \\
 V &= \{1, 2, 3, 4, 5\} \\
 E &= \{(1, 2), (1, 3), (2, 3), (3, 1), (4, 1), (5, 1)\}
 \end{aligned}$$

Abbildung 2: Darstellung des Graphen  $G$

Legt man ein graphentheoretische Terminologie für das TSP zugrunde, so können die Objekte als Knoten eines Graphen sowie die Distanzen von einem Objekt zu einem anderen als Bewertung der Kanten desselben aufgefasst werden.

Ein Pfad in einem Graphen ist eine endliche Folge von Kanten  $p = (u_0, v_0), \dots, (u_{m-1}, v_{m-1})$  mit  $m \in \mathbb{N}_0$  und  $v_{i-1} = u_i$  für alle  $i \in \{1, \dots, m-1\}$ . Hierbei ist  $m$  die Länge des Pfades. Wir identifizieren Kanten und Pfade der Länge 1. Der Pfad der Länge 0 heißt der leere Pfad. Der Knoten  $u_0$  heißt Anfangsknoten von  $p$ , der Knoten  $v_{m-1}$  heißt Endknoten. Alle anderen Knoten von  $p$  heißen innere Knoten.  $p$  heißt Zyklus, wenn er geschlossen ist, d.h. wenn  $v_{m-1} = u_0$  gilt.

Eine Rundreise als mögliche (jedoch nicht unbedingt optimale) Lösung des TSP ist dann ein zyklischer, zusammenhängender Pfad, der jeden Knoten des Graphen genau einmal berührt – jeder Knoten weist damit genau zwei ihn berührende Kanten und damit den Grad zwei auf. Eine solche Kantenfolge wird auch als „Hamiltonischer Zyklus“ bezeichnet. Die optimale Lösung des TSP entspricht damit dem bewertungsminimalen Hamiltonischen Zyklus, d.h. jenem mit der minimal bewerteten Kantenfolge.

Wird ein vollständiger, ungerichteter Graph zugrundegelegt, in welchem Verbindungen zwischen beliebigen Objekten richtungsunabhängig zugelassen sind, so stellt die Identifikation eines Hamiltonischen Zyklus kein Problem dar: Jede mögliche Permutation bzw. Folge der Knoten entspricht einem solchen. Da damit sichergestellt ist, dass für den Graphen stets mindestens ein hamiltonischer Zyklus existiert, wird er auch als „*hamiltonisch*“ bezeichnet - Ist ein Graph jedoch unvollständig in dem Sinne, dass die Folge bestimmter Knoten unzulässig ist (also keine Kante zwischen diesen existiert) oder gerichtet in dem Sinne, dass die Folge bestimmter Knoten nur richtungsabhängig zulässig ist, so ist es im Allgemeinen ein äußerst schwieriges Problem („*Hamiltonkreisproblem*“) nachzuweisen, ob der Graph hamiltonisch ist oder nicht.

Ein „*Hamiltonischer Pfad*“ unterscheidet sich vom Hamiltonischen Zyklus dadurch, dass keine geschlossene, zirkuläre Kantenfolge verlangt wird. Vielmehr genügt für die Identifikation desselben das Auffinden einer Kantenfolge, die sämtliche Knoten des Graphen einmal berührt. Widerum muss jeder Knoten – ausgenommen natürlich Anfangsknoten und Endknoten – dabei den Grad zwei aufweisen. Die Suche nach einem bewertungsminimalen hamiltonischen Pfad kann nunmehr unmittelbar in ein TSP überführt werden, indem eine den ersten und letzten Knoten verbindende fiktive Kante eingeführt wird, welche mit einer hinreichend kleinen Distanz zu bewerten ist.

### 3. Ansätze zur Lösung

Zur allgemeinen Lösung einer bestimmten Art von Problemen werden mehr oder weniger genau definierte Handlungsvorschriften (Algorithmen) erstellt.

Die Anzahl der Schritte, die ein Algorithmus benötigt, wird als die Laufzeit des Algorithmus bezeichnet. Der Begriff Schritt bezieht sich auf ein bestimmtes zugrundegelegtes Maschinenmodell. Die Maschine muss in der Lage sein, einen einzelnen Schritt in konstanter Zeit auszuführen.

Um Algorithmen unabhängig von den Details der Implementation bewerten zu können, wird die Zeitkomplexität mithilfe der  $O$ -Notation angegeben. Die  $O$ -Notation gibt nur die Größenordnung der Komplexität wieder, d.h. ob es sich z.B. um eine linear, quadratisch oder exponentiell wachsende Funktion handelt. Die Größenordnung wird in Form einer Komplexitätsklasse angegeben.

Um eine Komplexitätsklasse zu bezeichnen, gibt man immer die einfachste Funktion an, die geeignet ist, die jeweilige Komplexitätsklasse zu repräsentieren. Tatsächlich handelt es sich bei  $O(2n^2 + 7n - 10)$  und  $O(n^2)$  um dieselben Mengen, man wählt aber  $O(n^2)$  als Bezeichnung. Die Menge  $O(n^2)$  ist die Komplexitätsklasse aller höchstens quadratisch wachsenden Funktionen. Entsprechend ist  $O(n)$  die Menge der höchstens linear wachsenden Funktionen,  $O(\log(n))$  die Menge der höchstens logarithmisch wachsenden Funktionen,  $O(1)$  sind die durch eine Konstante beschränkten Funktionen,  $O(n^k)$  die höchstens polynomiell wachsenden Funktionen,  $O(2^n)$  die höchstens exponentiell zur Basis 2 wachsenden Funktionen.

#### 3.1. Exakte Verfahren

##### 3.1.1. Vollständige Enumeration

Bei der Suche nach einer optimalen Lösung eines solchen Problems ist eine der

grundlegendsten Fragen, wie viele mögliche Lösungen ein TSP überhaupt hat. Im Falle endlich vieler Lösungen und der Möglichkeit diese zu bewerten ist es somit rein theoretisch möglich die bzw. eine optimale auszuwählen.

Bei der vollständigen Enumeration (lat: „*enumerare*“: aufzählen, ausrechnen) werden alle mögliche Lösungen bestimmt, bewertet und verglichen. Bei großen TSP ist dieses Verfahren jedoch gänzlich ungeeignet, da sich für vollständige ungerichtete Graphen  $\frac{(n-1)!}{2}$  mögliche Lösungen für n Knoten ergeben!

Es stellt sich daher die Frage nach approximativen Verfahren, die zwar nicht unbedingt die beste Lösung finden, aber dieser sehr nahekommen. (*siehe 3.2*)

### 3.1.2. Entscheidungsbaumverfahren (implizite Enumeration)

Entscheidungsbäume beginnen mit einem Stamm, an dessen Ende sich eine Verzweigung befindet, die in mehrere - mit Wahrscheinlichkeiten versehene - wiederum verzweigte Äste führt. Jeder Endpunkt des Baums ist durch einen eindeutigen Weg erreichbar.

Bei Entscheidungsbaumverfahren wird im Allgemeinen ein Problem als eine Reihe von Teilproblemen betrachtet, die sich durch Entscheidungen verzweigen. Die Anzahl nacheinander getroffener Entscheidungen heißt die Ebene einer Entscheidung.

#### 3.1.2.1. Verkürzte Enumeration

Ergänzend zur vollständigen Enumeration wird hier bei der Berechnung einer Rundreise abgebrochen, sobald ein schlechteres Ergebnis als bei einer zuvor berechneten Rundreise erzielt wird. Wird nicht abgebrochen, so entspricht die neue Bewertung einer oberen Schranke für weitere Berechnungen. Zur Optimierung wird am Anfang ein heuristisches Eröffnungsverfahren (*siehe 3.2.1*) verwendet.

#### 3.1.2.2. Dynamische Optimierung

Dynamische Programmierung kann dann erfolgreich eingesetzt werden, wenn das Optimierungsproblem aus vielen gleichartigen Teilproblemen besteht, und eine optimale Lösung des Problems sich aus optimalen Lösungen der Teilprobleme zusammensetzt. Das Verfahren der dynamischen Programmierung besteht grundsätzlich darin, zuerst die optimalen Lösungen der kleinsten Teilprobleme zu berechnen, und diese dann geeignet zu einer Lösung eines nächstgrößeren Teilproblems zusammenzusetzen, und so weiter.

Anstatt eine Rundreise komplett zu berechnen werden hier in Ebenen alle möglichen Kanten von einem Knoten aus parallel verarbeitet, gespeichert und bewertet. Dies ermöglicht das vorzeitige Ausschließen von Entscheidungszweigen. Es gilt hier, bei der Lösung kostspielige Rekursionen durch Wiederverwendung schon berechneter Zwischenlösungen zu vermeiden. Einmal berechnete Teilergebnisse werden in einer Tabelle gespeichert, um später auf sie zurückgreifen zu können.

#### 3.1.2.3. Branch & Bound

Der Branch-and-Bound Algorithmus besteht aus zwei Teilen: dem Branching ("Verzweigen") und dem Bounding ("Beschränken"). Im ungünstigsten Fall müssen alle möglichen Belegungen enumeriert werden. Es wird versucht, den zu untersuchenden

Lösungsraum möglichst klein zu halten, indem Zweige im aufgespannten Entscheidungsbaum als suboptimal identifiziert werden und aus der weiteren Betrachtung herausfallen.

### *Branching*

Der *Branching*-Schritt dient dazu, das vorliegende Problem in zwei oder mehr Teilprobleme aufzuteilen, die eine Vereinfachung des ursprünglichen Problems darstellen. Durch rekursives Ausführen des Branching-Schritts für die erhaltenen Teilprobleme entsteht eine Baum-Struktur ("tree structure"). Es gibt verschiedene Auswahlverfahren für die Wahl des nächsten zu bearbeitenden Knotens im Branching-Baum, die unterschiedliche Zielsetzungen haben. Im Folgenden werden drei häufig verwendete Verfahren beschrieben:

*Tiefensuche* ("depth first search"): Von den noch nicht bearbeiteten Teilproblemen wird das gewählt, welches als letztes eingefügt wurde. Mit dieser Auswahlregel arbeitet sich das Verfahren im Baum möglichst schnell in die Tiefe, so dass sehr schnell eine zulässige Lösung erlangt wird, über deren Qualität jedoch nichts ausgesagt werden kann.

*Breitensuche* ("breadth first search"): Von den noch nicht bearbeiteten Teilproblemen wird das gewählt, welches als erstes in den Baum eingefügt wurde. Bei Verwendung dieser Auswahlregel werden die Knoten im Baum pro Ebene abgearbeitet, bevor tiefer gelegene Knoten betrachtet werden. Eine zulässige Lösung wird in der Regel erst relativ spät erhalten, hat aber tendenziell einen guten Lösungswert.

*Bestensuche* ("best first search"): Von den noch nicht bearbeiteten Teilproblemen wird das gewählt, welches die beste untere Schranke<sup>4</sup> vorweist. Durch diese qualitativ arbeitende Auswahlregel wird versucht, direkt einen sehr guten Lösungswert als erste Lösung zu erhalten.

### *Bounding*

Der *Bounding*-Schritt hat die Aufgabe, bestimmte Zweige des Baumes abzuschneiden, d.h. in der weiteren Berechnung nicht mehr zu betrachten, um so den Rechenaufwand zu begrenzen. Dies erreicht der Algorithmus durch Berechnung und Vergleich zweier Schranken. Der Weg von der Wurzel des Baumes zu einem Teilproblem bildet die untere Schranke ("lower bound") dieses Teilproblems – auf welche Weise sich auch der weitere Weg durch den Baum gestaltet, der Weg zum Teilproblem ist festgelegt. Als obere Schranke ("upper bound") dient eine vorerst optimale zulässige Lösung. Diese erhält man durch Berechnung eines kompletten Zweiges (von der Wurzel zu einem Blatt des Entscheidungsbaumes) – ebenso ist es aber auch möglich, eine Heuristik vor dem eigentlichen Branch-and-Bound Verfahren zu berechnen und die erhaltene Lösung als obere Schranke zu verwenden.

Übersteigt nun die untere Schranke in einem Teilproblem die obere Schranke, so kann der aktuell betrachtete Teilbaum "abgeschnitten" werden, da es mindestens eine bessere zulässige Lösung gibt, die nicht mehr erreicht werden kann. Sollte hingegen eine Komplettlösung berechnet werden, die besser als die obere Schranke ist, so wird diese ersetzt und als neue optimale Lösung vorgehalten.

---

4 Als Schranke wird ein Wert bezeichnet, der durch ein Optimum keines Falls unter - bzw. überschritten werden kann

### 3.2. Heuristiken

Der Begriff Heuristik ist nach dem griechischen Ausruf „Heureka“ (gr.: „*Ich habe es gefunden!*“) benannt und bezeichnet eine Strategie, die das Streben nach Erkenntnis und das Finden von Wegen zum Ziel planvoll gestaltet.

Bezüglich dem TSP steuern Heuristiken beim Suchen, in welcher Reihenfolge der Suchraum bzw. Suchbaum durchquert wird. Dies geschieht durch eine geschickt gewählte Bewertungsfunktion, welche die Entfernung der zur Auswahl stehenden Wege zum Ziel schätzt.

#### 3.2.1. Eröffnungsverfahren

##### 3.2.1.1. Nearest Neighbor

Dies ist das die wahrscheinlich erste intuitive Methode die bei einer Annäherung der Lösung als sinnvoll erscheint und entspräche bestimmt dem tatsächlichen Vorgehen eines Handlungsreisenden.

Von einem Knoten als Startpunkt ausgehend wird die minimalgewichete benachbarte Kante zum nächsten Knoten gewählt. Dieses wird sukzessive fortgesetzt, bis alle Knoten zu einem Hamiltonischen Pfad zusammengefasst wurden. Dieser wird schließlich durch die Verbindung von Startknoten und Endknoten zyklisch geschlossen. (*siehe Abbildung 3*) Wie bereits durch Karl Menger angedeutet liefert dieses Verfahren meistens nicht die beste Lösung. Dies liegt daran, dass der Startknoten und der Endknoten zu keinem Zeitpunkt berücksichtigt werden und anstatt dessen eine mögliche große Distanz zwischen ihnen in Kauf genommen wird.

Indem iterativ jeder einzelne Knoten einmal als Startknoten zur Ermittlung der Gewichtung der jeweilig entstehenden Rundreise gewählt wird und diese abschließend miteinander verglichen werden wird das Verfahren besser. Jedoch entspricht dieses sogenannte „*All Nearest Neighbor*“ Verfahren bereits der Komplexitätsklasse  $O(n^2)$ .

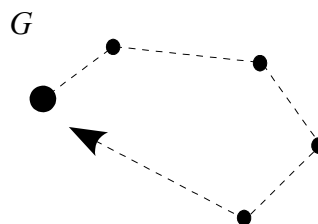


Abbildung 3: "Nearest Neighbor" Verfahren

##### 3.2.1.2. Nearest Selection (Floyd-Warshall-Algorithmus)

Gegeben sei ein gerichteter Graph  $G = (V, E)$  mit  $V = \{0, \dots, n-1\}$ ,  $n \in \mathbb{N}$ . Gefragt ist für alle Paare von Knoten  $(i, j)$ , ob es in  $G$  einen Weg von  $i$  nach  $j$  gibt. Der Graph  $G^+$  wird aus  $G$  entwickelt, indem schrittweise neue Kanten hinzugenommen werden. Im ersten Schritt kommt eine Kante  $(i, j)$  hinzu, wenn sich aus zwei Kanten ein Weg von  $i$  nach  $j$  bilden lässt, der über den Knoten 0 führt (d.h. wenn  $(i, 0)$  und  $(0, j)$  Kanten sind). Im zweiten Schritt kommt eine Kante  $(i, j)$  hinzu, wenn sich aus zwei Kanten ein Weg von  $i$

nach  $j$  bilden lässt, der über den Knoten 1 führt; hierbei werden die in Schritt 1 neugefundenen Kanten mit berücksichtigt.

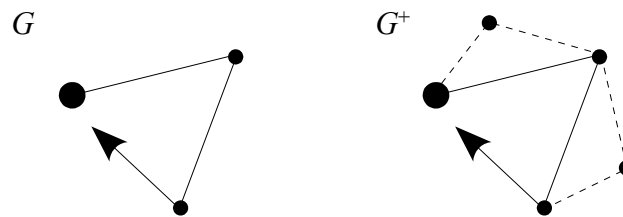


Abbildung 4: Transitive Hülle durch "Nearest Selection"

Jedesmal, wenn der Algorithmus einen kürzeren Weg von  $i$  nach  $j$  als den bisher bekannten findet, wird  $G^+$  aktualisiert (siehe Abbildung 4). Dieses Verfahren wird bis zum  $n$ -ten Knoten fortgesetzt und entspricht der Komplexitätsklasse  $O(n^3)$ .

### 3.2.1.3. Minimum Spanning Tree (Prim-Algorithmus)

Ein gerichteter Graph  $T = (V, E)$  ist ein Baum, wenn er keine Zyklen und genau einen Knoten mit Eingangsgrad 0 enthält; dieser Knoten ist die Wurzel des Baumes, sowie alle anderen Knoten den Eingangsgrad 1 haben.  $T$  heißt Spannbaum des Graphen  $G$ , wenn er ein Baum ist und aus allen Knoten von  $G$  besteht. Gesucht ist der Spannbaum  $T$  des Graphen  $G$ , der ein minimales Kantengewicht hat.

Die Idee des Verfahrens von Prim zur Berechnung eines minimalen Spannbaums ist, den Baum ausgehend von einem Startknoten  $s$  Schritt für Schritt aufzubauen. Zu jedem Zeitpunkt besteht die Menge aller Knoten  $V$  aus drei disjunkten Mengen:

(A) aus der Menge der Knoten  $T$ , die schon zum Baum dazugehören, (B) aus einer Menge von Kandidaten  $K$ , die zu einem Baumknoten benachbart sind, aber noch nicht selbst zum Baum gehören und (C) aus den restlichen, noch unberücksichtigten Knoten.

Die Strategie ist nun, nach einem Kandidaten  $v$  zu suchen, der durch eine Kante minimalen Gewichts mit  $T$  verbunden ist. Diese Kante wird zum Baum hinzugenommen, der Kandidat  $v$  wird zu  $T$  hinzugenommen und aus  $K$  entfernt. Anschließend wird die Menge der Kandidaten  $K$  aktualisiert, indem die zu  $v$  benachbarten Knoten hinzugenommen werden, die noch nicht zu  $T$  gehören.

Durch Verdopplung der Kanten wird der Zyklus geschlossen. Da diese Lösung allerdings unzulässig ist, werden nach und nach von den ursprünglichen Enden des minimal spannenden Baumes umgekehrt zum Nearest Selection Verfahren die Kanten nach und nach abgetragen, bis eine zulässige Lösung entsteht.

### 3.2.2. Approximative Verfahren

Approximative Optimierungsverfahren versuchen meist, ausgehend von einer vorhandenen Lösung durch eine geeignete Abänderung zu einer besseren Lösung zu gelangen. Führt keine solche mögliche Abänderung mehr zu einer Verbesserung, so ist ein Optimum gefunden.

### 3.2.2.1. Kantentausch (k-opt Verfahren)

Die Verbesserung der Lösung beim TSP besteht darin, eine durch Zufall oder Heuristiken gefundene Rundreise so abzuwandeln, dass die Summe der Kantenbewertungen entlang der Tour reduziert wird. Das k-Opt-Verfahren zeichnet sich dadurch aus, dass es eine bestimmte Menge von Kanten aus der aktuellen Lösung entfernt, um danach neue Kanten einzufügen, die die entstandenen Lücken schließen.

Die k-Opt-Heuristiken verwenden das Verfahren der lokalen Suche in Nachbarschaften. Die *Nachbarschaft* zu einer gegebenen Lösung  $f$  ist definiert als die Menge aller Lösungen, die man durch gewisse festgelegte Veränderungen an  $f$  erhält. Im Falle der k-Opt-Heuristik wird eine *k-Tausch-Nachbarschaft* definiert als Menge aller gültigen Lösungen, die entstehen, wenn man  $k$  Kanten aus  $f$  entfernt und danach  $k$  neue Kanten einfügt. Um die Gültigkeit der Lösung zu gewährleisten müssen die neuen Kanten die Rundreise schließen (siehe *Abbildung 5*).

Der wohl bekannteste k-Opt Algorithmus ist der von S. Lin und B. W. Kernighan (1973). Er arbeitet in der Praxis sehr effizient, kann aber in Einzelfällen exponentielle Zeitkomplexität aufweisen. Das besondere am Lin-Kernighan-Algorithmus ist, dass er mit einem variablen  $k$ -Wert arbeitet, der in jeder Iteration neu bestimmt wird.

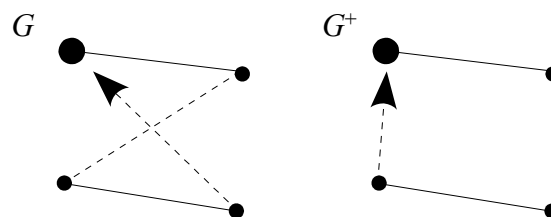


Abbildung 5: Kantentausch mit zwei Kanten (2-opt Verfahren)

### 3.3. Metaheuristiken

Problematisch bei approximativen Verfahren, wird die Tatsache, dass bei einer vollständigen Durchführung der Aufwand im Vergleich zur vollständigen Enumeration nicht gesenkt würde. Deshalb werden diese Approximationsschritte in einzelne Bereiche geteilt. Bei der Auffindung eines Optimums ist es nun jedoch möglich, dass es sich um ein lokales Optimum handeln kann, welches wesentlich schlechter ist als das globale Optimum.

Am Beispiel eines Minimierungsproblems wird die Situation dargestellt (siehe *Abbildung 6*). Die Ausgangslösung ist  $A$ , durch schrittweise Verbesserung ist das Verfahren zur Lösung  $B$  gelangt. Alle Schritte, die von  $B$  aus möglich sind, führen zu einer Verschlechterung der Lösung,  $B$  ist ein lokales Minimum. Das globale Minimum liegt jedoch bei  $D$ . Das Verfahren kann das globale Minimum jetzt nur noch finden, wenn es für eine begrenzte Anzahl von Schritten auch eine Verschlechterung der Lösung in Kauf nimmt, so dass es über den Berg  $C$  gelangen kann.



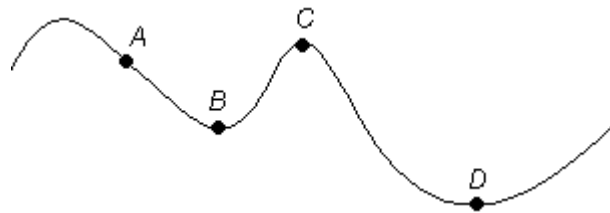


Abbildung 6: Lokales und globales Minimum

### 3.3.1. Tabu Search

Eine bereits bestehende Rundreise wird durch ein Approximationsverfahren (z.B. Kantenaustausch) lokal solange verbessert bis dadurch keine weitere relative Verbesserung erzielt wird. Um zu verhindern, dass das absolute Optimum nicht erreicht wurde, wird als nächsten Schritt eine gezielte minimale Verschlechterung erwirkt und das Approximationsverfahren wieder aufgenommen. Dabei wird der vorhergehend als optimale festgestellte Zyklus mit einem Tabu belegt.

### 3.3.2. Simmulated Annealing

Der Begriff Annealing stammt aus der Metallurgie und bezeichnet das Tempern von Metallen. Ein Metall ist umso härter, je besser sich eine geordnete Kristallstruktur ausgebildet hat. Beim Erstarren aus der Schmelze bilden sich an vielen Stellen Kristallisationszentren, und das Ergebnis ist ein Metall, bei dem viele kleine Kristalle kreuz und quer durcheinanderliegen. Beim Tempern wird nun das Metall nochmals bis kurz vor den Schmelzpunkt erhitzt und dann langsam wieder abgekühlt, so daß sich die Kristallstruktur des Metalls besser ausbilden kann; es bilden sich größere Kristalle, da die Metallatome innerhalb der Kristallstruktur eine niedrigere potentielle Energie haben als außerhalb der Kristallstruktur.

Durch das Tempern wird den Metallatomen noch einmal die Möglichkeit gegeben, sich zu bewegen, so daß sie beim Abkühlen einen energetisch günstigeren Platz finden können. Das Tempern muß sehr vorsichtig geschehen: wird das Metall zu stark erhitzt, kommen alle Atome wieder durcheinander und die vorher schon gefundene Ordnung wird wieder zerstört. Wird es zu schwach erhitzt, lösen sich die Atome nicht aus ihrer alten Ordnung und es wird keine bessere Ordnung gefunden.

Indem dieser Prozess des Temporns zur Herstellung eines Zustandes möglichst niedriger Energie simuliert wird - daher die Bezeichnung Simulated Annealing - erhält man ein Optimierungsverfahren. Die Temperatur entspricht hierbei dem oben erwähnten Parameter  $\sigma$ . Das „Simmulated Annealing“ Verfahren (Simmuliertes Ausglühen) ahmt einen natürlichen Prozess des aushärtens von Kristallstrukturen nach. Dabei wird eine vorhandene Rundreise stochastisch approximiert (z.B. Kantenaustausch), wobei abhängig von der Optimierungsstufe (Temperatur) auch Verschlechterungen (Anfangs auch stärkere, später nur leichte) akzeptiert werden, um ein möglichst optimale Rundreise zu erhalten.

Um das Travelling Salesman Problem mit Simulated Annealing zu lösen, wird wie folgt vorgegangen: Gestartet wird mit einer Rundreise, die die Städte in einer völlig zufälligen Reihenfolge durchläuft. Nun wird versucht, diese Lösung zu verbessern. Hierzu werden zwei Städte  $s_i$  und  $s_j$  zufällig ausgewählt, und der Weg von  $s_i$  nach  $s_j$  wird in umgekehrter Richtung durchlaufen.

Ist die neue Rundreise kürzer als die alte Rundreise, so wird mit dieser verbesserten

Lösung fortgefahren. Ist sie länger, so wird sie nur dann akzeptiert, wenn sie um höchstens *sigma* länger ist als die alte Rundreise. Ansonsten wird sie verworfen, und es wird mit der alten Lösung fortgefahren. Nach jeweils einer bestimmten Anzahl von Iterationsschritten wird der Parameter *sigma* verringert, bis er den Wert 0 erreicht, so daß zum Schluß keine Verschlechterung der Lösung mehr zugelassen wird.

### 3.3.3. Ameisenalgorithmen

Eine weitere Heuristik basiert auf so genannten Ameisenalgorithmen, bei denen das natürliche Verhalten von Ameisen bei der Wegsuche modelliert wird.

Bei der Futtersuche scheiden einzelne Ameisen einen Duftstoff (sog. Pheromon) aus, der andere Ameisen anzieht. Gibt es zwischen Nest und Futterquelle mehrere mögliche Wege, so wird mit der Zeit auf dem kürzesten Pfad eine höhere Pheromonkonzentration als auf den anderen vorherrschen, so dass die Ameisen bevorzugt diesen Weg wählen werden: sie scheinen entlang einer Straße zu laufen, eine Ameisenstraße ist entstanden. Ähnliches gibt es auch beim Menschen. Z.B. wurde auf einem Campus beobachtet, wie die Fußspuren durch den Schnee verlaufen und im Frühjahr wurden danach die Wege (Verlauf und Breite) gestaltet.

Dieses Vorgehen ist eine Form von Schwarmintelligenz: eine Leistung (hier: die Lösung eines Optimierungsproblems) wird durch das Zusammenspiel einer Vielzahl von wenig intelligenten (und damit leicht nachprogrammierbaren) Einzelakteuren (den Ameisen) hervorgebracht.

Ameisenalgorithmen zählen zu Multiagentensystemen. Ameisenalgorithmen zählen zu den lokalen Suchverfahren, die durch iterative Optimierung einer zufällig gefundenen Lösung gekennzeichnet sind. Neben dieser iterativen Optimierung sind Ameisenalgorithmen durch Metaheuristiken gekennzeichnet, die in diesem Fall aus einem natürlichen Verhalten von Ameisen gewonnen werden. Die zugehörigen Optimierungsverfahren sind auch als Ant Colony Optimization (ACO) bekannt.

## Anhang I: Abbildungsverzeichnis

Sybyklen.....	8
Darstellung des Graphen G.....	9
"Nearest Neighbor" Verfahren.....	13
Transitive Hülle durch "Nearest Selection".....	14
Kantentausch mit zwei Kanten (2-opt Verfahren).....	15
Lokales und globales Minimum.....	16

## Anhang II: Literaturverzeichnis

Jarre, Stoer, „Optimierung“, Springer  
Domschke, Drexl, „Einführung in Operations Research“, 6. Aufl., Springer  
H.W. Lang, „Algorithmen“, Oldenbourg  
W. Schmitting, „Das Traveling Salesman Problem“  
D. Jungnickel: „Graphen, Netzwerke und Algorithmen“, B.I. Wiss.-Verl.  
<http://de.wikipedia.org/wiki/Kategorie:Graphentheorie>