

Studienarbeit

Routenplanung in der Handelsvertretung



Institut für Wirtschafts- und Verwaltungsinformatik
Arbeitsgruppe Troitzsch

vorgelegt von: Sascha Bonnemann

Matrikelnummer: 20421007

Betreuer: Dipl. Inf. Daniel Schmidt

Prüfer: Prof. Dr. Klaus G. Troitzsch

Oktober 2012

Hiermit erkläre ich, dass ich die beiliegende Arbeit selbst angefertigt
und
andere Hilfsmittel als die genannten nicht benutzt habe.

Koblenz, den

Inhaltsverzeichnis

1	Einleitung	4
1.1	Motivation	5
1.2	Ziel dieser Arbeit	5
1.3	Aufbau dieser Arbeit	6
2	Routenplanung	7
2.1	Definition Routenplanung	8
2.2	Routenplanung in der Handelsvertretung	8
2.3	Algorithmen zur Routenplanung	9
3	Traveling Salesman Problem	12
3.1	Definition Traveling Salesman Problem	13
4	Webbasierte Kartendienste	14
4.1	Google Maps	14
4.2	OpenStreetMap	16
5	Bestehende Softwarelösungen	19
6	Beispielimplementierung mit der Google Maps API	21
7	Fazit	31

1 Einleitung

In dieser Studienarbeit geht es um die theoretische Erweiterung von *conlabz adm* - einer Aussendienst-Management Software für Handelsvertretungen der *Conlabz GmbH*. [1] Für diese Software soll eine zusätzliche Komponente konzipiert werden. Diese Komponente soll die Tourenplanung unterstützen - konkret soll sie für eine Reihe von gegebenen Märkten eine möglichst kurze und optimale Route berechnen. Dies soll unter Anderem mit Hilfe von Geokoordinaten unterstützt werden. Jeder Markt, jeder Anfahrtspunkt für einen Vertreter besitzt eindeutige Geokoordinaten. Anhand dieser Koordinaten könnte man eine erste sinnvolle Analyse durchführen um bspw. herauszufinden, wie die Distanzen zwischen den einzelnen Punkten sind. Dieser Tourenplanung geht aber erst einmal eine möglichst sinnvolle Gebietsauslastungsplanung voraus auf die an dieser Stelle aber nicht weiter eingegangen werden soll. Diese Planung soll eine möglichst optimale Verteilung der Märkte auf die Mitarbeiter erreichen. Hierfür entscheidend sind natürlich zum Einen der Wohnort des Mitarbeiters, zum Anderen weitere Faktoren, wie Entfernung der Märkte untereinander, Verweildauer und Besuchsrhythmus pro Markt und die Provision. Anhand dieser Parameter sollte eine optimale Aufteilung der Gebiete möglich sein. Für die Abgrenzung der Gebiete untereinander werden optimalerweise die Postleitzahlengrenzen verwendet, da diese bereits geographische Begebenheiten wie Flüsse, Berge etc. berücksichtigen. Ein Teil dieser Arbeit wird aus Recherche und Analyse bereits vorhandener Software bestehen, die die oben genannten Aufgaben erfüllt und sich gegebenenfalls via Schnittstellen an *Conlabz adm* anpassen lässt.

Für die Tourenplanung sind 2 grosse Hauptpunkte zu nennen einmal die Clusterung, die angibt, welche Märkte zu einer Tour zusammengefasst werden und zum Anderen das Routing, welches die Reihenfolge der

Märkte innerhalb einer Tour definiert. Eine möglichst optimale Reihenfolge der Märkte zu bestimmen wird die Hauptaufgabe dieser Studienarbeit werden. Hierbei dürfen nicht nur die Geokoordinaten als Referenz herangezogen werden - es gibt eine Vielzahl von Faktoren die das Routing beeinflussen werden. Zudem müssen bzw. können die Märkte in unterschiedlichen Rhythmen besucht werden. Private Faktoren der einzelnen Vertreter, wie z.Bsp. Urlaubs-/Krankheitstage, Mittagspausen, etc. können den Routenverlauf ebenfalls beeinflussen.

1.1 Motivation

Die Motivation dieser Arbeit besteht darin, eine Recherche und Analyse bestehender Softwarelösungen zu erarbeiten und das Ganze im Umfeld einer nicht-fiktiven Aufgabe mit konkretem Praxisbezug. Routenplanung ist ein Thema, das gerade aufgrund der stark zunehmenden Verbreitung von Smartphones und Navigationsgeräten immer weiter an Wichtigkeit erlangt. Geräte wie das iPad bspw. bieten ganz neue Möglichkeiten für dynamische Routenplanungen und Anpassungsoptionen während der Arbeitszeit. Aktuell ist es so, dass Routen für die Handelsvertretung wochen- oder monatsweise erstellt werden und ab dann relativ statisch sind. Wünschenswert sind Lösungen die es einem Handelsvertreter ermöglichen auch unterwegs ideal auf unvorhergesehene Ereignisse zu reagieren und diese in Echtzeit in seine Route einarbeiten zu lassen. Im Rahmen dieser Arbeit wird allerdings nicht auf den dynamischen Betrieb mit bspw. Smartphones eingegangen.

1.2 Ziel dieser Arbeit

Ziel dieser Arbeit ist es, eine theoretische Lösung zu finden, mit deren Hilfe die *conlabz GmbH* ihr adm-Tool für die Handelsvertretung um ei-

ne Routenplanung erweitern kann. Vorausgesetzt wird hierbei, dass eine Menge M von Märkten gegeben ist, für die eine optimale Route berechnet werden soll. Die Marktzuzuweisung pro Mitarbeiter und damit die Gebietsauf-/verteilung wird im Rahmen einer anderen Studienarbeit behandelt und für diese Ausarbeitung als abgeschlossen betrachtet.

1.3 Aufbau dieser Arbeit

Diese Studienarbeit untersucht das Thema Routenplanung in diesem speziellen Fall mittels vier Hauptkategorien. Zuerst wird das Thema Routenplanung an sich genauer beleuchtet und untersucht. Hierbei lassen sich bereits wichtige Faktoren ermitteln, bspw. für die spätere Auswahl eines Online-Kartendienstes¹. Ein Hauptkernpunkt dieser Studienarbeit wird das Traveling Salesman Problem(TSP)² werden - es bekommt ebenfalls ein eigenes Kapitel, in welchem es genauer betrachtet wird. Ein wichtiger Teil dieser wissenschaftlichen Ausarbeitung ist die Analyse bereits vorhandener Software. Hier wurden aktuelle Programme/Dienstleister recherchiert und hinsichtlich der Lösung der hier behandelten Aufgabenstellung untersucht. Die webbasierten Kartendienste werden ebenfalls in einem eigenem Kapitel betrachtet - hierbei werden die aktuell zwei bekanntesten und vermeintlich besten Systeme genauer untersucht. Vor dem abschließenden Fazit wird anhand einer Beispielimplementierung gezeigt, wie eine Routenplanung mittels Google Maps gelöst werden kann.

¹ein Online-Kartendienst stellt in einem Webinterface Kartenmaterial von bestimmten Regionen, Ländern oder der ganzen Welt zur Verfügung. GoogleMaps bspw. bietet verschiedene Darstellungsmöglichkeiten: Karte, Satellit etc.

²Problem des Handelsreisenden: wie sieht die optimale Route aus, mit der ich in einer Tour bspw. 10 Städte besuchen kann.

2 Routenplanung

Wurden früher Reiserouten mit Hilfe von Karten am Küchentisch geplant, ist heute die computergestützte Routenplanung in weiten Teilen der Bevölkerung etabliert: Die beste Eisenbahnverbindung ermittelt man im Internet, für Routenplanung in Straßennetzen benutzt man häufig mobile Endgeräte. [2] Das Thema Routenplanung hat in der vergangenen Jahren vor allem durch die Präsenz von Navigationsgeräten in nahezu jedem Auto enorm an Wichtigkeit gewonnen.

Grundlegend geht es dabei um die Fragestellung "Wie komme ich am besten von einem Standort A zu einem Standort B". Diese scheinbar trivial zu beantwortende Frage birgt allerdings eine enorme Komplexität die es abzufangen gilt. Um die Strecke zwischen zwei Städten wie bspw. Berlin und München berechnen zu lassen muss auf verschiedene Faktoren geachtet werden:

- Art der Strecken, die der Nutzer fahren möchte/darf (Autobahn, Landstraße, ...)
- soll die kürzeste oder die schnellste Strecke gewählt werden (Zeit- vs. Spritkosten)
- wo/wie können bei mehrstündigen/mehrtägigen Fahrten Pausen eingebaut werden
- meide bestimmte Autobahnkreuze zu bestimmten Zeiten (Staugefahr)

Eine relativ naive Routenplanung [3] könnte bspw. folgendermaßen aussehen:

1. Suche nächste sinnvolle Autobahnauffahrt

2. Fahre auf der Autobahn möglichst nahe ans Ziel heran

3. Suche von der Autobahnausfahrt den Weg zum Ziel

Bei dieser Herangehensweise würden natürlich viele der kleinen aber sehr wichtigen Faktoren nicht berücksichtigt, weshalb eine Routenplanung nach diesem System in unserem Fall keinen wirklichen Sinn macht.

2.1 Definition Routenplanung

Routenplaner (Streckenplaner, Wegplaner, von französisch: route = Weg) sind Computerprogramme, mit deren Hilfe ein Weg zwischen einem Start- und einem Zielort gefunden werden kann. Meistens können auch ein oder mehrere Orte dazwischen ("via") angegeben werden. Es können meistens Wünsche angegeben werden, ob die schnellste, die kürzeste, die wirtschaftlichste (ökonomischste) oder manchmal auch die schönste Route gesucht werden soll. [4]

Diese Definition komprimiert sehr gut das Grundprinzip bzw. -problem der Routenplanung. Der Versuch, eine optimale Strecke zwischen A und B zu finden - eventuell auch mit sogenannten Zwischenpunkten.

2.2 Routenplanung in der Handelsvertretung

Die Routenplanung in der Handelsvertretung stellt einen Spezialfall dar, da es meist nicht nur um eine Strecke von A nach B (mit eventuellen Zwischenpunkten) geht, sondern in der Regel immer darum z.Bsp. an einem Tag N Kunden anzufahren und am Abend wieder nach Hause und somit zum Ausgangspunkt der Tour zurück zu kehren. Daraus kann man erkennen, dass ein wesentlicher Faktor bei der Berechnung die Gewichtung einer Strecke ist. Diese Gewichtung kann man sich als Entfernung (Kilometer oder Zeit) oder auch als benötigte Spritkosten vorstellen. In unserem Fall haben wir eine asymmetrische Routenplanung zu untersu-

chen, da die Strecke von A nach B immer ungleich der Strecke von B nach A ist. Gerade in der Handelsvertretung spielt eine perfekte Routenplanung eine entscheidene Rolle - nämlich dafür, wie viele Kunden ein Vertreter pro Tag schafft und somit, wieviel er der Firma einbringt und natürlich auch wieviel er selbst an Provision erhält. Die Provision ist hierbei ein entscheidender Faktor und muss vorallem bei der Marktzweisung und Gebietsaufteilung beachtet. Hierauf wird im Rahmen dieser Arbeit allerdings nicht weiter eingegangen.

2.3 Algorithmen zur Routenplanung

Bei der Routenplanung geht es darum in einem Transportnetzwerk die beste³ Verbindung zu finden. Hierzu existieren bereits viele Lösungen, die dieses Problem auch optimal lösen. Optimal bedeutet hierbei das eine Beweismöglichkeit dafür besteht, dass definitiv die beste Verbindung gefunden wurde. Abstrahiert bedeutet das: Suche alle möglichen Verbindungen zwischen bspw. den beiden Städten A und B. Die kürzeste (bezogen auf Zeit oder Kilometer) ist die beste. Das Problem hierbei ist allerdings die Rechenzeit die ein Algorithmus benötigt um alle möglichen Streckenoptionen zu berücksichtigen. Möchte man zum Beispiel in der Handelsvertretung eine möglichst dynamische Lösung entwickeln, die auch noch während der Vertreter sich auf der Straße befindet auf Gegebenheiten wie Stau, Pausen und Ähnliches eingeht, fallen alle Algorithmen/Lösungen weg, die mehrere Minuten für Neuberechnungen benötigen würden. Die optimale Route zu berechnen ist folglich die eine Seite - dass dies im Optimalfall innerhalb von (Milli-)Sekunden passieren soll die andere.

³die *beste* kann hier bei für die schnellste, kürzeste oder kostengünstigste Verbindung stehen.

Fast alle Algorithmen verfolgen erst einmal folgendes abstrahierende Grundprinzip, wenn es um die Abbildung bzw. die Berechnung von Problemen aus dem Bereich der Routenplanung geht:

- betrachte das Strassennetzwerk als einen Graphen $G = (V, E)$ ⁴
- bilde die Reisedauer pro Weg/Kante als Kantengewichte ab
- die kürzesten Wege im Graphen G entsprechen den schnellsten Verbindungen im jeweiligen Verkehrsnetzwerk

Eine Möglichkeit diese Berechnungen durchzuführen stellt der *Algorithmus von Dijkstra*⁵ dar. Der Dijkstra⁵-Algorithmus zählt zu den bekanntesten Algorithmen im Bereich der Routenplanung. Mit seiner Hilfe kann die Berechnung kürzester Wege in bewerteten Graphen durchgeführt werden. Der Algorithmus funktioniert folgendermaßen: [6]

1. Der Startknoten kommt in *opti*
2. Alle anderen Knoten kommen in *rest*
3. berechne alle Distanzen für die Knoten in *rest*
4. verschiebe den Knoten mit der kleinsten Distanz von *rest* nach *opti*
5. berechne die Distanzen für die Knoten in *rest* neu
6. wiederhole die Schritte 4. und 5. solange, bis *rest* leer ist

In Pseudocode geschrieben würde dieser Algorithmus daher wie folgt aussehen:

Knotenmenge s, k ; // $s =$ Startknoten

⁴ V bezeichnet hierbei eine Menge von Knoten und E eine Menge von Kanten [5]

⁵Edsger Wybe Dijkstra (* 11. Mai 1930 in Rotterdam; † 6. August 2002 in Nuenen, Niederlande) war ein niederländischer Informatiker

```

Knotenmenge  $opti = \{s\}$ ;
Knotenmenge  $rest = k \setminus \{s\}$ 
for ( $k$  aus  $rest$ ) do
     $D[k] = d[s, k]$  (es gibt Weg);
     $= \infty$  sonst;
done;
while ( $rest$  nicht leer) do;
    wähle  $k$  aus  $rest$  mit  $min(D[k])$ ;
     $opti + = \{k\}$ 
     $rest = rest - \{k\}$ ;
    for (alle Knoten  $n$  von  $k$ ) do;
         $D[n] = min(D[n], D[k] + d[k, n])$ ;
    done;
done;
```

Der Dijkstra-Algorithmus findet die kürzesten Distanzen zu allen anderen Knoten, wenn keine negativen Distanzen (Betrag) zugelassen sind. Aufgrund seiner Implementierung kann außerdem verifiziert nachgewiesen werden, dass die gefundene Distanz definitiv auch die kürzeste ist. Für eine praktische Anwendung in der Routenplanung ist dieser Algorithmus allerdings nicht geeignet, da er bei größeren Mengen (sprich Strecken und Anfahrtszielen) nicht schnell genug arbeitet. Hierfür gibt es verschiedene Erweiterungen die eine Geschwindigkeitsoptimierung bewirken sollen, allerdings ist hierbei noch keine perfekte Lösung gefunden worden. [6]

3 Traveling Salesman Problem

Das Problem der kürzesten Pfade in Netzen (in unserem Fall adaptiert auf die Routenplanung) beschäftigt seit vielen Jahrzehnten Informatiker und Mathematiker. Das Traveling Salesman Problem ist eines der bekanntesten Probleme der Informatik. Es tritt nicht nur in vielen Problemen des täglichen Lebens sondern auch als Unterproblem in anderen Fragestellungen auf. Durch die weite Verbreitung besitzt die effiziente Lösung des Traveling Salesman Problems einen hohen Stellenwert. [7]

Das *Traveling Salesman Problem* ist ein kombinatorisches Optimierungsproblem der theoretischen Informatik. [8] Bei diesem Optimierungsproblem geht es darum, die kürzeste Rundreise zwischen beliebig vielen Städten, die mit Straßen verbunden sind, zu finden. Ziel ist es eine Reihenfolge für den Besuch der Städte zu ermitteln, wobei außer dem Ausgangsort keine Stadt mehr als einmal besucht werden darf. Die Länge einer Tour wird hierbei durch die Summe der Längen der benutzten Straßen gebildet.

Das *Traveling Salesman Problem* ist allerdings nicht nur auf Orte und deren Entfernungen beschränkt, sondern wird in anderen Bereichen in denen es um Optimierungsprobleme geht, angewandt. Hierbei wird fast immer von Kosten gesprochen. Diese Kosten können verschiedene Parameter repräsentieren, wie bspw. die Entfernung, die Zeit oder die finanziellen Kosten, die für die Überwindung einer Strecke notwendig sind. Die Strecke wiederum ist keinesfalls nur auf Strassen und Wege in Verkehrsnetzen zu reduzieren, sondern kann ebenfalls für eine Reihe von verschiedensten Parametern stehen. Durch die Abstrahierungsmöglichkeit des *Traveling Salesman Problems* kann es in einer großen Bandbreite von Gebieten eingesetzt werden. Hierzu seien beispielhaft genannt:

- Tourenplanung
- Logistik von Postunternehmen und Speditionen
- Design⁶ von Mikrochips
- Layout von Leiterplatten

Bei den beiden zuletzt genannten Punkten werden die Kosten bspw. nicht als Distanz zwischen einzelnen Orten betrachtet, sondern vielmehr als die Signallaufzeit einer Strecke und der auf sie eventuell einwirkenden Störfaktoren. [7]

3.1 Definition Traveling Salesman Problem

Traveling Salesman ist das folgende Problem, als Sprache formuliert:

Traveling Salesman := $\{(G, g, k) \mid k \in \mathbb{N}, (G, g) \text{ ist ein vollständiger, bewerteter Graph mit } G = (V, E), \text{ und es existiert ein Hamilton-Kreis}^7 H = (v_0, \dots, v_n) \text{ in } G \text{ mit } \sum_{i=0}^{n-1} g((v_i, v_{i+1})) \leq k\}$

Als Hamilton-Kreis benutzt H jeden Knoten in V genau einmal, bis auf $v_0 = v_n$. (v_i, v_{i+1}) ist die Kante von v_i nach v_{i+1} . Insgesamt muß also eine Rundreise mit Gesamtkosten $\leq k$ gefunden werden. [9]

⁶Design bedeutet hierbei Strukturierung

⁷Namensgeber ist der irische Astronom und Mathematiker Sir William Rowan Hamilton (* 4. August 1805 in Dublin; † 2. September 1865 in Dunsink bei Dublin)

4 Webbasierte Kartendienste

Zum Zeitpunkt der Verfassung dieser Studienarbeit existiert eine Vielzahl von webbasierten Kartendiensten, die eventuell für eine Routenplanung in Frage kommen würden. Hierfür seien beispielhaft genannt:

- Bing Maps⁸
- Yahoo Maps⁹
- OpenLayers¹⁰
- Nokia Maps¹¹

Als besonders umfangreich und geeignet zeigen sich Google Maps und OpenStreetMap, weshalb diese beiden Dienste nun im Folgenden genauer analysiert werden.

4.1 Google Maps

Google Maps ist ein Dienst von Google Inc., der es ermöglicht Orte, Hotels und andere Objekte zu suchen und anschließend deren Position auf einer Karte oder auf einem Bild der Erdoberfläche (Satelliten- und Luftbilder) anzuzeigen. Der Dienst bietet eine Programmierschnittstelle (application programming interface - kurz API), mit der man Google Maps in seine eigenen Internetseiten einbinden kann. Das API bietet viele Möglichkeiten, die Karte auf der eigenen Seite zu konfigurieren - beispielsweise kann eine Zoomleiste eingeblendet werden oder es können Marker auf der Karte platziert werden. [12] Da mittles dieses API

⁸<http://www.bing.com/maps/>

⁹<http://de.maps.yahoo.com/>

¹⁰<http://openlayers.org/>

¹¹<http://maps.nokia.com/>

sehr viele Funktionen zur Verfügung stehen, wie zum Beispiel der Routenplaner, wird Google Maps für diese Studienarbeit genauer betrachtet.

Theoretisch kann Google Maps eine Lösungsmöglichkeit für viele im Rahmen dieser Studienarbeit auftretenden Probleme sein. Es bietet die Möglichkeit einer Kartenvisualisierung, das Setzen von mehreren Markern (in unserem Fall würden diese bspw. die einzelnen Märkte repräsentieren), das Einzeichnen von Gebietsgrenzen (eine Möglichkeit wäre die Unterteilung in Postleitzahlengebiete) und das Berechnen einer Route. Google Maps ist in der Lage verschiedene Routen von A nach B zu berechnen. Diese Routen werden straßengenau aufgeführt - als Ergebnis liefert das Programm eine Kilometeranzahl (bei mehreren Routenvorschlägen entsprechend verschieden) und eine ungefähre Abschätzung von Zeit und Spritverbrauch.

Google Maps kann Routen für verschiedene Transportmethoden mithilfe des *DirectionsService*-Objektes berechnen. Dieses Objekt kommuniziert mit dem Google Maps API-Routendienst, der Routenanfragen empfängt und berechnete Ergebnisse zurückgibt. Diese Routenergebnisse kann man nun entweder selbst weiter verarbeiten oder man verwendet das *DirectionsRenderer*-Objekt für die Wiedergabe der Ergebnisse. [13] In den Routenangaben können die Ursprungs- und Zielorte entweder in Form von Textzeichenfolgen angegeben werden, wie "Koblenz", "Köln, Deutschland" oder als LatLng-Werte (Längen- und Breitengrad). Der Routendienst kann auch eine Route zurückgeben, die über eine Reihe von Wegpunkten (die anschließend noch genauer erläutert werden) führen und somit aus mehreren Streckenabschnitten bestehen. Routen werden auf einer Karte als Polylinienzeichnung oder zusätzlich als Textbeschreibungen

(zum Beispiel “Nach 900 m rechts auf die A48 auffahren”) innerhalb eines `<div>`-Elements angezeigt.

Wegpunkte in GoogleMaps

Zusätzlich zu dieser einfachen Routenberechnung bietet die neueste Google Maps API in der Version 3 die Möglichkeit einer Route weitere Wegpunkte hinzuzufügen. Somit kann eine Route berechnet werden die außer Start- und Endpunkt über weitere Stationen führt. Ein *waypoint* besteht aus folgenden Feldern:

- *location* (erforderlich): Gibt die Adresse des Wegpunktes an.
- *stopover* (optional): Gibt an, ob dieser Wegpunkt ein Zwischenhalt auf der Route ist (*true*) oder nur ein Standort, über den die Route bevorzugt verlaufen soll (*false*). Für Zwischenhalte gilt standardmäßig der Wert *true*.

Standardmäßig berechnet der Routendienst eine Route so, dass sie in der Reihenfolge durch die angegebenen Wegpunkte führt, in der diese angegeben wurden. Optional kann man im *DirectionsRequest* den Wert *optimizeWaypoints: true* übergeben; dann optimiert der Routendienst die angegebene Route, indem er die Wegpunkte in eine effizientere Reihenfolge bringt. Diese Optimierung ist eine Anwendung des Problem des Handlungsreisenden. Damit der Routendienst eine Route optimieren kann, müssen alle ihre Wegpunkte Zwischenhalte sein. [14]

4.2 OpenStreetMap

OpenStreetMap (OSM) ist ein im Jahre 2004 gegründetes Projekt mit dem Ziel, eine freie Weltkarte zu erschaffen. OSM sammelt weltweit Da-

ten über Straßen, Eisenbahnen, Flüsse, Wälder, Häuser und alles andere, was gemeinhin auf Karten zu sehen ist. Weil die Daten selbst erhoben werden und nicht aus existierenden Karten abmalen, liegen auch alle Rechte an den Kartendaten- und Materialien beim Unternehmen selbst. Die OpenStreetMap-Daten darf jeder einsetzen und beliebig weiterverarbeiten. Hierbei fallen in keinsterweise Lizenzgebühren oder ähnliches an. Die meisten Mitglieder der OSM-Community beteiligen sich durch Mapping, also dadurch, dass sie mit einem GPS-Gerät Kartendaten sammeln und bei OpenStreetMap eingeben. Viele zeichnen auch dafür zugelassene Luftbilder ab und vervollständigen die Daten vor Ort. [10] OpenStreetMap selbst bietet die gesammelten Daten entweder in Rohform oder in Form vorberechneter Kartenbilder an. Zum Anschauen oder um schnell und unkompliziert eine Karte auf die eigene Webseite zu bringen, eignen sich die vorberechneten Kartenbilder am besten (auch die Karte auf OpenStreetMap.de arbeitet damit).

Wer eigene Berechnungen mit den Daten anstellen möchte oder Karten in einem selbst definierten Stil produzieren will, der benutzt statt fertiger Kartenbilder die Rohdaten. Die Rohdaten können als regelmäßig aktualisierte Datei heruntergeladen werden (das sog. „planet file“ mit den Daten der ganzen Welt). Weiterhin bieten verschiedene Firmen kostenlose Auszüge daraus: GeoFabrik.de, Cloudmade.com oder Metro Extracts. Für kleinere Ausschnitte (bis rund 1000 km², weniger in Gebieten mit hoher Datendichte) gibt es die Rohdaten auch direkt über das Web-Interface auf www.openstreetmap.org (Export-Tab).

Der größte Schwachpunkt von OpenStreetMap liegt in der Unvollständigkeit der Daten - laut Betreiber kann und will keinesfalls ein Anspruch auf Vollständigkeit erhoben werden. [11] Es fehlen noch viele Strassen, teilweise sogar komplette Orte. Dies ist mit der Hauptgrund, warum Open-

StreetMap für den professionellen Betrieb im Bereich der Routenplanung bei einer Handelsvertretung nicht relevant ist. Die bereits in der Einleitung erwähnte *conlabz GmbH* benötigt ein mindestens deutschlandweit komplett flächendeckendes Strassennetzwerk für die Routenplanung um Ihren Kunden eine sinnvolle Nutzung der Software garantieren zu können.

5 Bestehende Softwarelösungen

Der aktuelle¹² Softwaremarkt bietet bereits einige Komplettlösungen bzgl. Gebiets- und Routenplanung. Bei der Recherche wurden vier Angebote gefunden und genauer betrachtet, welche den Eindruck erweckten, die notwendigen Anforderungen erfüllen zu können. Das Problem hierbei war allerdings, die Software in vollem oder Demoumfang testen zu können. Trotz des Hinweis, dass die Anfragen im Rahmen einer Studienarbeit notwendig bzw. gewünscht sind, waren alle Softwarehersteller wenig kooperativ und eher nicht daran interessiert weitere Informationen zu kommunizieren, weshalb die Analyse der Software in diesem Kapitel wesentlich knapper als gewünscht ausfallen muss und die Angaben der jeweiligen Internetseiten als Grundlage der Betrachtungen dienen müssen.

Die vier Programme, die in die engere Auswahl kamen, sind:

- PTV SMarTour 1.1 [17]
- TransIT [18]
- RegioGraph [19]
- Business und Seele [20]

Alle vier dieser Programme versprechen die Möglichkeit mittels der bereit gestellten Software eine optimale Routen- und Gebietsplanung zu liefern. Oftmals geht dem eine Analyse des Ist-Zustand voraus, gefolgt von optimierten Vorschlägen für die zukünftige Vorgehensweise. *Regio-Graph Analyse* ist bspw. laut eigener Aussage ideal für die Darstellung und Analyse der Märkte, Kunden und Standorte auf Landkarten. *Business und Seele* bietet laut Herstellerangabe eine Auslastungsanalyse und

¹²Stand Sommer 2012

Tourenplanung die für Transparenz und die gezielte Steuerung jedes einzelnen ADM¹³ sorgen sollen. Desweiteren werden bereits einige wichtige Parameter wie Besuchsrhythmus, Besuchsdauer, Fahrzeiten innerhalb des Strassennetzes abgebildet. Die auf der Homepage sichtbaren Screenshots der Software lassen allerdings ein designtechnisch eher schwaches Produkt erwarten bei dem man vermutlich keine allzu große Übersichtlichkeit vorfinden wird.

Unabhängig davon, ob es um die Touren- oder Einsatzplanung geht, prophezeit *TransIT* eine Senkung der operativen Kosten um 10 bis 15 %. Weiterhin verspricht die Software eine Optimierung, die auch die schwierigen Anforderungen berücksichtigt, weit mehr als sonst marktüblich. Was genau hierbei besser als bei der Konkurrenz sein soll wird allerdings nicht eingehender erläutert. Ein großer Vorteil von *TransIT* sind die vorhandenen Schnittstellen die es ermöglichen die Software in nahezu jede andere vorhandene Umgebung einzubetten. *TransIT* nutzt das SOAP¹⁴-Protokoll, welches heute als Industrie-Standard für Web-Services angesehen werden kann. *TransIT*-Schnittstellen wurden bereits zu mehr als 30 verschiedenen Systemen implementiert. Beispielsweise zu SAP R/3 und Business One, zu CSB businessware© und zahlreichen anderen Softwaresystemen.

Wie bereits erwähnt, hat es sich als schwierig herausgestellt tiefergehende Informationen der jeweiligen Softwarehersteller zu bekommen, die für die Analyse wichtig und interessant gewesen wären.

¹³Außendienstmitarbeiter

¹⁴SOAP (ursprünglich für *Simple Object Access Protocol*) ist ein Netzwerkprotokoll, mit dessen Hilfe Daten zwischen Systemen ausgetauscht und Remote Procedure Calls durchgeführt werden können

6 Beispielimplementierung mit der Google Maps API

Google Maps ist ein am 8. Februar 2005 gestarteter Dienst von Google Inc., der es ermöglicht Orte, Hotels und andere Objekte zu suchen, um deren Position dann auf einer Karte oder auf einem Bild der Erdoberfläche (Satelliten- und Luftbilder) anzuzeigen. Der Anwender kann zwischen einer reinen Kartendarstellung, einem Luftbild und einer Ansicht wählen, die sowohl eine Karte als auch das Luftbild darstellt. Navigationselemente und eine Zoomfunktion erlauben es, sich auch ohne Suchbegriff auf dem Karten- bzw. Bildausschnitt zu bewegen. [12]

GoogleMaps bietet eine Vielzahl an APIs¹⁵ mit deren Hilfe der Nutzer die Dienste und Funktionen innerhalb seiner eigenen Seite nutzen kann. V3 ist die aktuelle¹⁶ Version des Javascript-API, die eine Einbettung von Google Maps in die eigene Webseite erlaubt. Das API bietet eine Reihe von Hilfsprogrammen, mit denen der Nutzer über verschiedene Dienste Karten bearbeiten (wie auf der Website <http://maps.google.com>) und Inhalte zu Karten hinzufügen kann. Das JavaScript Maps API V3 ist ein kostenloser Service und verfügbar für alle Websites, die für Besucher kostenlos sind.

Der folgende HTML- und Javascript-Code [16] zeigt ein Minimalbeispiel, wie man eine Google Maps Karte in die eigene Homepage einbinden kann. Hierbei wird die Karte auf Sydney in New South Wales, Australien fokussiert. Dies geschieht über die Angabe von den entsprechenden Längen- und Breitengraden:

```
<!DOCTYPE html>
<html>
```

¹⁵API=Application Programming Interface, deutsch: "Schnittstelle zur Anwendungsprogrammierung"

¹⁶Stand: Juli 2012

```

<head>
<meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
<style type="text/css">
  html { height: 100% }
  body { height: 100%; margin: 0px; padding: 0px }
  #map_canvas { height: 100% }
</style>
<script type="text/javascript"
  src="https://maps.google.com/maps/api/js?sensor=true">
</script>
<script type="text/javascript">
  function initialize() {
    var latlng = new google.maps.LatLng(-34.397, 150.644);
    var myOptions = {
      zoom: 8,
      center: latlng,
      mapTypeId: google.maps.MapTypeId.ROADMAP
    };
    var map = new google.maps.Map(document.getElementById("map_canvas"),
      myOptions);
  }

</script>
</head>
<body onload="initialize()">
  <div id="map_canvas" style="width:100%; height:100%"></div>
</body>
</html>

```

Als erstes muss das eigentliche Google Maps API geladen werden. Dies funktioniert, indem man die URL <http://maps.google.com/maps/api/js>

mittels `script`-Tag einbindet. Der Parameter *sensor* beschreibt ob die Position des Benutzers ermittelt werden kann - er muss daher auf *true* oder *false* gesetzt werden. Im `<body>`¹⁷-Bereich der Webseite muss ein `<div>`¹⁸-Element platziert werden in dem die Karte angezeigt werden soll. Das `<div>`-Element benötigt daher eine eindeutige Zuordnungsmöglichkeit, welche mittels der Vergabe einer ID geregelt wird. Diese eindeutige ID, in unserem Fall *map_canvas*, muss in der Funktion *initialize()* übergeben werden und schon wird die Google Maps Karte an entsprechender Stelle angezeigt. Bei unserem Beispiel geschieht das über die komplette Seite, da mittels CSS¹⁹ dem `<div>`-Element mit der ID *map_canvas* eine Breite und Höhe von jeweils 100% zugewiesen wurde.

Das für den Rahmen dieser Studienarbeit interessanteste API ist das *Google Directions API* [13], welches scheinbar alle notwendigen Funktionalitäten zur Routenberechnung bereitstellt. Beim *Google Directions API* handelt es sich um einen Dienst, der Routen zwischen Standorten mithilfe einer HTTP-Anforderung berechnet. In den Routen können Startorte, Zielorte und Wegpunkte entweder als Textstrings (z. B. "Koblenz, Köln") oder mithilfe von Breiten- und Längengradkoordinaten angegeben werden. Das Directions API kann mehrteilige Routen unter Verwendung einer Reihe von Wegpunkten zurückgeben. Man kann Routen für verschiedene Transportmethoden mithilfe des *DirectionsService*-Objektes berechnen. Dieses Objekt kommuniziert mit dem Google Maps API-Routendienst, der Routenanfragen empfängt und berechnete Ergebnisse

¹⁷Das HTML `body`-Tag kennzeichnet Beginn und Ende des Inhalts der Webseite, der im Browserfenster dargestellt wird

¹⁸Mit `<div>` leitet man ein allgemeines Block-Element ein, in das wiederum mehrere andere Block-Elemente eingeschlossen werden können

¹⁹CSS = Cascading Style Sheets, sind eine deklarative Sprache für Stilvorlagen von strukturierten Dokumenten. Sie wird vor allem zusammen mit HTML (HyperText-MarkupLanguage) eingesetzt

zurückgibt. Diese Routenergebnisse können entweder selbst verarbeitet oder an das `DirectionsRenderer`-Objekt für die Wiedergabe dieser Ergebnisse übergeben werden. [13]

Der Zugriff auf den Routendienst ist asynchron, da das Google Maps API einen externen Server aufrufen muss. Aus diesem Grund muss eine Callback-Methode übergeben werden, die nach der Durchführung der Anfrage ausgeführt wird. Mit dieser Callback-Methode werden die Ergebnisse verarbeitet. Hierbei ist zu beachten, dass der Routendienst mehrere mögliche Wegverläufe in Form eines Arrays mit separaten `routes[]` zurückgeben kann. Unter Verwendung der Google Maps Javascript API in der neuesten Version (V3), ruft man ein Objekt des Typs `DirectionsService` und anschließend die Funktion `DirectionsService.route()` auf, um eine Anfrage an den Routendienst zu initiieren. Dabei wird ein `DirectionsRequest`-Objektliteral übergeben, das die Eingabebedingungen und eine Callback-Methode enthält, die nach dem Empfang der Antwort ausgeführt werden soll. Das `DirectionsRequest`-Objektliteral enthält die folgenden Felder [13]:

```
{
  origin: LatLng | String,
  destination: LatLng | String,
  travelMode: TravelMode,
  unitSystem: UnitSystem,
  waypoints[]: DirectionsWaypoint,
  optimizeWaypoints: Boolean,
  provideRouteAlternatives: Boolean,
  avoidHighways: Boolean,
  avoidTolls: Boolean
  region: String
}
```

Im Folgenden werden die einzelnen Funktionsparameter [13] genauer erläutert:

- **origin** (*erforderlich*) – die Adresse oder der Breiten- und Längengrad als Textwert, von der/dem aus Routen berechnet werden sollen
- **destination** (*erforderlich*) – die Adresse oder der Breiten- und Längengrad als Textwert, zu der/dem Routen berechnet werden sollen
- **mode** (*optional, wird standardmäßig auf driving eingestellt*) – gibt an, welche Transportart bei der Berechnung von Routen verwendet werden soll.
- **waypoints** (*optional*) – gibt ein Array von Wegpunkten an. Durch Wegpunkte wird eine Route so geändert, dass sie über die angegebenen Standorte führt. Wegpunkte werden entweder als Koordinaten der geografischen Breite und Länge oder als Adresse angegeben, die dann geocodiert wird
- **alternatives** (*optional*) – gibt bei Einstellung von true an, dass das Directions API mehr als eine Routenalternative in der Antwort bereitstellen kann. Beachten Sie, dass die Bereitstellung von Routenalternativen die Antwortzeit des Servers erhöhen kann.
- **avoid** (*optional*) – legt fest, dass die berechneten Routen die angegebenen Elemente vermeiden sollen. Derzeit werden die beiden folgenden Argumente von diesem Parameter unterstützt:
 - **tolls** legt fest, dass die berechnete Route Mautstraßen und -brücken vermeiden soll.
 - **highways** legt fest, dass die berechnete Route Autobahnen vermeiden soll.

- **units** (*optional*) – gibt an, welches Maßeinheitensystem für die Anzeige der Ergebnisse verwendet werden soll
- **region** (*optional*): der Ländercode, der als zweistelliger ccTLD-Wert ("top-level domain") angegeben wird
- **language** (*optional*): die Sprache, in der die Ergebnisse zurückgegeben werden sollen
- **sensor** (*erforderlich*) – gibt an, ob die Routenanfrage von einem Gerät mit einem Standortsensor kommt. Dieser Wert muss entweder true oder false sein.

Für die Anforderungen dieser Studienarbeit ist es besonders interessant, dass das *Google Directions API* dazu in der Lage ist, Wegpunkte mit in die Berechnung einer Route einfließen zu lassen. Wegpunkte (in dem API als *waypoints* bezeichnet) ermöglichen es, Routen mithilfe von zusätzlichen Standorten zu berechnen, sodass die Route über diese angegebenen Wegpunkte verläuft. Damit wird genau das Traveling Salesman Problem angegangen, welches bei der Routenplanung mit mehreren Städten entsteht. Der Routendienst berechnet eine Route standardmäßig anhand der bereitgestellten Wegpunkte in ihrer angegebenen Reihenfolge. Optional können Sie `optimize:true` als erstes Argument innerhalb des `waypoints`-Parameters übergeben, um es dem Routendienst zu ermöglichen, die angegebene Route zu optimieren, indem die Wegpunkte in eine effizientere Reihenfolge gebracht werden. Diese Optimierung ist eine Anwendung des Problems des Handlungsreisenden.

Ein *waypoint* umfasst zwei Parameter [15]:

- **location** (*erforderlich*): Gibt die Adresse des Wegpunktes an

- **stopover** (*optional*): Gibt an, ob dieser Wegpunkt ein Zwischenhalt auf der Route ist (true) oder nur ein Standort, über den die Route bevorzugt verlaufen soll (false). Für Zwischenhalte gilt standardmäßig der Wert true.

Wenn man den Routendienst anweist, die Reihenfolge der Wegpunkte zu optimieren, wird die neue Reihenfolge im *optimized_waypoints_order*-Feld innerhalb des *DirectionsResult*-Objektes zurückgegeben.

Im folgenden Beispiel werden Routen durch die USA mit verschiedenen Startpunkten, Endpunkten und Wegpunkten berechnet. Durch Strg+Klicken können mehrere Wegpunkte gleichzeitig in der Liste ausgewählt werden [15]:

```
var directionDisplay;
var directionsService = new google.maps.DirectionsService();
var map;

function initialize() {
    directionsDisplay = new google.maps.DirectionsRenderer();
    var chicago = new google.maps.LatLng(41.850033, -87.6500523);
    var myOptions = {
        zoom: 6,
        mapTypeId: google.maps.MapTypeId.ROADMAP,
        center: chicago
    }
    map = new google.maps.Map(document.getElementById("map_canvas"), myOptions);
    directionsDisplay.setMap(map);
}

function calcRoute() {
```

```

var end = document.getElementById("end").value;
var waypts = [];
var checkboxArray = document.getElementById("waypoints");
for (var i = 0; i < checkboxArray.length; i++) {
    if (checkboxArray.options[i].selected == true) {
        waypts.push({
            location:checkboxArray[i].value,
            stopover:true
        });
    }
}

var request = {
    origin: start,
    destination: end,
    waypoints: waypts,
    optimizeWaypoints: true,
    travelMode: google.maps.TravelMode.DRIVING
};

directionsService.route(request, function(response, status) {
    if (status == google.maps.DirectionsStatus.OK) {
        directionsDisplay.setDirections(response);
        var route = response.routes[0];
        var summaryPanel = document.getElementById("directions_panel");
        summaryPanel.innerHTML = "";
        // For each route, display summary information.
        for (var i = 0; i < route.legs.length; i++) {
            var routeSegment = i+1;
            summaryPanel.innerHTML += "<b>Route Segment: " + routeSegment + "</b><br />"
            summaryPanel.innerHTML += route.legs[i].start_address + " to ";

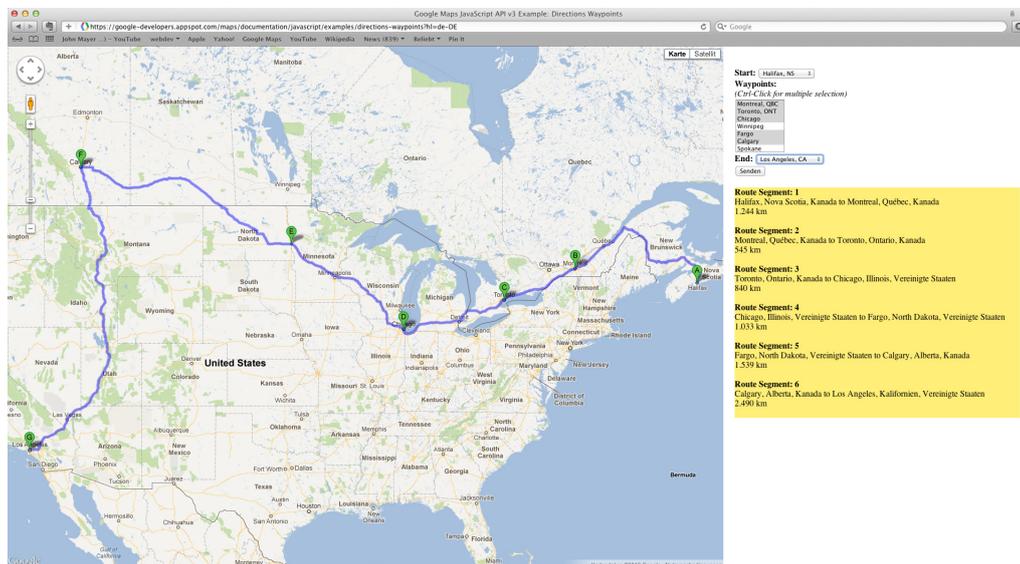
```

```

summaryPanel.innerHTML += route.legs[i].end_address + "<br />";
summaryPanel.innerHTML += route.legs[i].distance.text + "<br /><br />";
    }
}
});
}

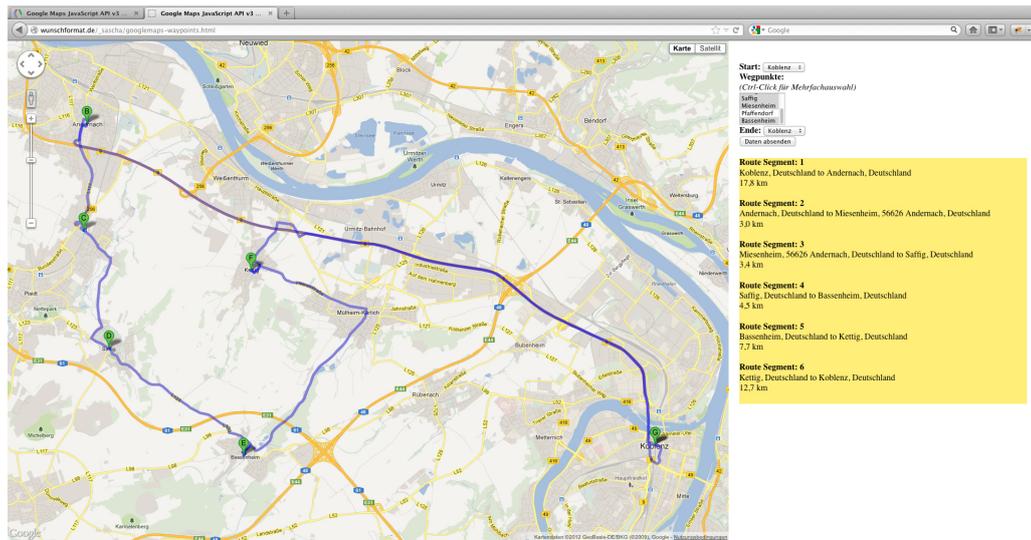
```

Das Ergebnis dieses Codes (eingebettet in eine Website mit einer Google Maps Karte) sieht folgendermaßen aus:



Um dieses Technik konkret auf unser Problem und die Handelsvertretung anzuwenden, wurden für das folgende Beispiel Orte im Raum Koblenz eingetragen. Koblenz wurde jeweils als Start- und Endpunkt angegeben um den Charakter eines Handelsreisenden widerzuspiegeln der bspw. von zu Hause aus seine Tour startet und anschließend wieder zurückkehrt. Der Quellcode ist nahezu identisch mit dem des vorherigen Beispiels - einzig die Städte wurden geändert und die Option von unterschiedlichem Start- und Endpunkt entfernt. Da im Javascript die Variable *waypoints* den

Wert *true* übergeben bekommen hat, errechnet das Skript automatisch die kürzeste Tour für die ausgewählten Orte. In unserem Fall startet die Strecke in Koblenz und führt über Andernach, Miesenheim, Saffig, Bassenheim und Kettig wieder zurück zum Ausgangsort Koblenz:



Hieran kann man sehen wie schnell und relativ einfach sich eine Routenplanung mittels der Google Maps API umsetzen lässt. Ist das Skript einmal geschrieben, benötigt der Nutzer im Prinzip nur noch die Eingabe der anzufahrenden Städte (Märkte, etc...) und kann die Berechnung starten. Die anzufahrenden Ziele liegen im besten Fall in Form von Geokoordinaten vor um eine bestmögliche Berechnung und Planung durchführen zu können. Die Google Maps API würde darüber hinaus auch erlauben alternative Routen anzubieten bzw. dem Nutzer die Möglichkeit zu geben selbst innerhalb der Karte die Route zu verschieben. Dies kann im Einzelfall bei der Routenplanung sinnvoll sein, wenn ein Handelsvertreter bestimmte Wege bspw. wegen Stau oder ähnlichem meiden will. Im Grunde sollte aber besser nicht so stark von der vorgegebenen Route abgewichen werden.

7 Fazit

Der Softwaremarkt bietet bereits einige fertige Lösungen für die Routenplanung. Einige davon können nur innerhalb einer bestimmten Umgebung laufen, andere wiederum bieten verschiedene Schnittstellen und ermöglichen somit eine Integration des Dienstes in bereits vorhandene Systeme. Aufgrund des schlechten und eher spärlichen Feedbacks der in Kapitel 5 (bestehende Softwarelösungen) genannten Produkthersteller ist es nicht möglich innerhalb dieser Studienarbeit eins der Produkte als besonders geeignet zu bezeichnen. Basierend auf den Herstellerbeschreibungen kann man nur mutmaßen, dass scheinbar alle der aufgezählten Softwareprodukte prinzipiell dazu in der Lage sein sollten, den gegebenen Anforderungen der Routenplanung in der Handelsvertretung gerecht zu werden. Ob und wie gut das nun aber wirklich der Fall ist, konnte leider nicht endgültig überprüft werden.

Deshalb wurde der Fokus auf verfügbare, kostenfreie Online-Dienste gerichtet, hierbei insbesondere auf GoogleMaps. GoogleMaps bietet derzeit das beste und flächendeckendste Kartenmaterial und zudem die umfangreichsten und ausgereiftesten APIs. Insbesondere durch die Verwendung von *waypoints*, was nichts anderes als eine Implementierung des *Traveling Salesman Problems* ist, eignet sich Google Maps hervorragend für die Verwendung im Bereich der Tourenplanung. Ein weiterer Vorteil ist, dass die Karten von Google Maps, deren Erscheinungsbild und deren Bedienung vielen Menschen schon von anderen Internetseiten oder eben Google Maps selbst bekannt sind. Dies kann sicherlich für einen hohen Prozentteil der Benutzer einen schnellen, weil intuitiven Einstieg gewährleisten.

GoogleMaps und die zahlreichen APIs bieten alle Möglichkeiten, die man für die Routenplanung braucht. Es stehen genügend wichtige Parameter

zur Verfügung, man kann wählen ob man zu Fuß, mit dem Rad oder dem Auto unterwegs ist und ob man bspw. Autobahnen in den vorgeschlagenen Routen vermeiden möchte. Außerdem ist es möglich sich mehrere Alternativrouten vorschlagen zu lassen - insofern es sinnvoll bzw. notwendig ist.

Nachteile bei Google Maps sind, dass man maximal acht *waypoints* (zusätzlich zu Start- und Endpunkt) einsetzen kann. Das wird vermutlich bei vielen Handelsvertretern für eine Tagestour genügen, ist allerdings keine perfekte Grundlage für größere Projekte. Desweiteren ist die Google Lizenz nicht mehr kostenfrei und an andere Lizenzen gebunden, wenn man die APIs und Karten kommerziell in einem nicht-öffentlich zugänglichen Bereich verwendet. Dies ist im Fall der Routenplanung einer Handelsvertretung natürlich der Fall und müsste im speziellen Fall von der *conlabz GmbH* mit Google und dem jeweiligen Kunden abgeklärt werden. Ob es möglich ist hier Google Maps einzusetzen hängt dann im Endeffekt von rein wirtschaftlichen Faktoren ab und müsste jeweils im Einzelfall geprüft und ausgehandelt werden.

OpenStreetMap könnte in der Zukunft ein Konkurrent werden, wenn die entsprechende Community schneller und weiter wächst und noch mehr und besseres Kartenmaterial liefert. Wie bereits weiter oben erwähnt, ist dies aktuell genau die größte Schwachstelle des Projekts. Google hat hier natürlich durch den finanziellen Background ganz andere Möglichkeiten und Mittel um seinen Kartendienst weiterhin an der Spitze zu halten und baut diese Position durch weitere Angebote wie bspw. *Google Street View* etc. immer weiter aus.

Literatur

- [1] ein Produkt der Conlabz GmbH Koblenz
<http://www.conlabzadm.de/>
- [2] Wagner, D. & Sanders, P. (2012) *Algorithm Engineering für Routenplanung - Vorlesung der Uni Karlsruhe - Institut für Technologie* Online verfügbar unter <http://i11www.iti.uni-karlsruhe.de/teaching/winter2012/algorithmengineeringpraktikum/index> (Stand: 11.10.2012)
- [3] Sanders, P. & Schultes, D. (2005). *Schnelle und genaue Routenplanung - Tag der Informatik / Institut für Theoretische Informatik, Universität Karlsruhe* Online verfügbar unter <http://algo2.iti.kit.edu/schultes/hwy/tagderinf.pdf> (Stand 03.05.2012)
- [4] <http://de.wikipedia.org/wiki/Routenplanung> (Stand 14.06.2012)
- [5] Schwabe, F. (2009). *Graphen - Seminar Bioinformatik*. Vortrag an der TU Dortmund. Online verfügbar unter www.statistik.tu-dortmund.de/fileadmin/user_upload/Lehrstuehle/Genetik/BI09/Schwabe_Vortrag.pdf (Stand 14.07.2012)
- [6] Hinzmann, J.(2008) *Der Algorithmus von Dijkstra (Berechnung kürzester Wege in bewerteten Graphen)* Vortrag an der Leibniz Universität Hannover. Online verfügbar unter <http://janhinzmann.de/uni/master/sem2/gisPraxis2/vortrag.pdf> (Stand 03.06.2012)
- [7] Werner, M. & Proksch S. (2008) *Traveling Salesman Problem* Online verfügbar unter http://storage.kuldisch.de/3.2-tsp_ausarbeitung.pdf (Stand 18.05.2012)

- [8] Wikimedia Foundation Inc.
http://de.wikipedia.org/wiki/Problem_des_Handlungsreisenden
(Stand: 03.04.2012)
- [9] Lutz, P., Erk, K.(2000) *Theoretische Informatik - eine umfassende Einführung* 2. Auflage, Seite 453
- [10] OpenStreetMap.de - *OpenStreetMap.de - ein Projekt des FOSSGIS e.V.* Online verfügbar unter <http://www.openstreetmap.de> (Stand 02.03.2012)
- [11] OpenStreetMap.de - *http://www.openstreetmap.de/faq.html#was_ist_osm*
(Stand:11.10.2012)
- [12] Google Inc. (2005), *http://de.wikipedia.org/wiki/Google_Maps*
(Stand 23.07.2012)
- [13] Google Inc.,
<https://developers.google.com/maps/documentation/directions/?hl=de-DE> (Stand: 24.07.2012)
- [14] Google Inc.,
<https://developers.google.com/maps/documentation/directions/#Waypoints>
(Stand: 12.10.2012)
- [15] Google Inc.,
<https://developers.google.com/maps/documentation/javascript/directions>
(Stand: 12.10.2012)
- [16] Google Inc.,
<https://developers.google.com/maps/documentation/javascript/tutorial?hl=de>
(Stand: 12.10.2012)

- [17] PTV Planung Transport Verkehr AG (2012),
http://smartour.ptvgroup.com/LP/DE/lp_ptvsmartour/index.html
(Stand: 15.06.2012)
- [18] GTS Systems and Consulting GmbH (2012),
<http://www.gts-systems.de/index.php/de/tourenplanung>
(Stand: 15.06.2012)
- [19] GfK GeoMarketing GmbH (2012),
<http://www.regiograph.de> (Stand: 15.06.2012)
- [20] Business und Seele, Harald Knoll (2012),
<http://businessundseele.de/50> (Stand: 15.06.2012)