

## Funktionsprinzipien und Anwendungen von Algorithmen zur Pfadplanung

Bearbeiter 1: Mohammed Salih Mezraoui

Bearbeiter 2: David Gruber

Bearbeiter 3: Marius Müller

Gruppe: WissArb22/Thema 2/Gruppe-7

Ausarbeitung zur Vorlesung Wissenschaftliches Arbeiten

Trier, 15.07.2022

---

## Kurzfassung

In der Kurzfassung soll in kurzer und prägnanter Weise der wesentliche Inhalt der Arbeit beschrieben werden. Dazu zählen vor allem eine kurze Aufgabenbeschreibung, der Lösungsansatz sowie die wesentlichen Ergebnisse der Arbeit. Ein häufiger Fehler für die Kurzfassung ist, dass lediglich die Aufgabenbeschreibung (d.h. das Problem) in Kurzform vorgelegt wird. Die Kurzfassung soll aber die gesamte Arbeit widerspiegeln. Deshalb sind vor allem die erzielten Ergebnisse darzustellen. Die Kurzfassung soll etwa eine halbe bis ganze DIN-A4-Seite umfassen.

Hinweis: Schreiben Sie die Kurzfassung am Ende der Arbeit, denn eventuell ist Ihnen beim Schreiben erst vollends klar geworden, was das Wesentliche der Arbeit ist bzw. welche Schwerpunkte Sie bei der Arbeit gesetzt haben. Andernfalls laufen Sie Gefahr, dass die Kurzfassung nicht zum Rest der Arbeit passt.

The same in english.

---

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung und Problemstellung</b>	<b>1</b>
<b>2</b>	<b>Algorithmen zur Pfadplanung</b>	<b>2</b>
2.1	Warum existieren unterschiedliche Konsistenzmodelle?	2
<b>3</b>	<b>Optimierungsstrategien</b>	<b>3</b>
3.1	Bidirektionale Suche	3
3.2	Informed Search	3
<b>4</b>	<b>Anwendungen</b>	<b>5</b>
4.1	Warum existieren unterschiedliche Konsistenzmodelle?	5
<b>5</b>	<b>Zusammenfassung und Ausblick</b>	<b>6</b>
	<b>Literaturverzeichnis</b>	<b>7</b>
	<b>Glossar</b>	<b>8</b>

## Einleitung und Problemstellung

Begonnen werden soll mit einer Einleitung zum Thema, also Hintergrund und Ziel erläutert werden.

Weiterhin wird das vorliegende Problem diskutiert: Was ist zu lösen, warum ist es wichtig, dass man dieses Problem löst und welche Lösungsansätze gibt es bereits. Der Bezug auf vorhandene oder eben bisher fehlende Lösungen begründet auch die Intention und Bedeutung dieser Arbeit. Dies können allgemeine Gesichtspunkte sein: Man liefert einen Beitrag für ein generelles Problem oder man hat eine spezielle Systemumgebung oder ein spezielles Produkt (z.B. in einem Unternehmen), woraus sich dieses noch zu lösende Problem ergibt.

Im weiteren Verlauf wird die Problemstellung konkret dargestellt: Was ist spezifisch zu lösen? Welche Randbedingungen sind gegeben und was ist die Zielsetzung? Letztere soll das beschreiben, was man mit dieser Arbeit (mindestens) erreichen möchte.

## Algorithmen zur Pfadplanung

In diesem Kapitel wird beschrieben, warum es unterschiedliche Konsistenzmodelle gibt. Außerdem werden die Unterschiede zwischen strengen Konsistenzmodellen (Linearisierbarkeit, sequentielle Konsistenz) und schwachen Konsistenzmodellen (schwache Konsistenz, Freigabekonsistenz) erläutert. Es wird geklärt, was Strenge und Kosten (billig, teuer) in Zusammenhang mit Konsistenzmodellen bedeuten.

### 2.1 Warum existieren unterschiedliche Konsistenzmodelle?

Nach [Mos93] kann die Performanzsteigerung der schwächeren Konsistenzmodelle wegen der Optimierung (Pufferung, Code-Scheduling, Pipelines) 10-40 Prozent betragen. Wenn man bedenkt, dass mit der Nutzung der vorhandenen Synchronisierungsmechanismen schwächere Konsistenzmodelle den Anforderungen der strengen Konsistenz genügen, stellt sich der höhere programmiertechnische Aufwand bei der Implementierung der schwächeren Konsistenzmodelle als ihr einziges Manko dar.

## Optimierungsstrategien

In diesem Kapitel wird beschrieben, wie bisher vorgestellten Algorithmen zur Pfadplanung optimiert werden können.

### 3.1 Bidirektionale Suche

Bei der bidirektionalen Suche wird ein Suchalgorithmus simultan aus zwei Richtungen laufen gelassen – vom Startknoten zum Zielknoten und umgekehrt. Der Suchalgorithmus wird bei diesem Vorgehen so modifiziert, dass die Abbruchbedingung dann eintritt, wenn beide Suchen denselben Knoten expandieren. Somit betrachtet jede der beiden Suchen nur die Hälfte des Graphen, was in einer Reduktion der Zeitkomplexität resultiert [RN10b]. Es ist jedoch zu beachten, dass die Platzkomplexität stark ansteigt, da in beiden Suchen eigene Priority-Queues verwaltet werden müssen. Außerdem ist es für viele Problemstellungen keinesfalls trivial, eine Suche rückwärts durchzuführen, da eine Methode zur Berechnung des Vorgängers eines Knotens gegeben sein muss [RN10b].

Eine Implementierung der Bidirektionalen Suche ist der Bidirektionale Dijkstra Algorithmus von Vaira und Kurasova. [7] Hier wird der in Abschnitt ?? vorgestellte Dijkstra Algorithmus so modifiziert, dass eine bidirektionale Suche von zwei Prozessen auf einem Mehrkernprozessor parallelisiert durchgeführt wird. In mehreren experimentellen Analysen konnte gezeigt werden, dass die Laufzeit des parallelen bidirektionalen Dijkstra Algorithmus bis zu drei Mal kleiner ist als die des Standard Dijkstra Algorithmus.

### 3.2 Informed Search

Ein Ansatz, um effizienter Lösungen für das Shortest Path Problem zu finden, ist die Informed Search Strategie, bei der problemspezifisches Wissen, das über die Definition des Problems hinausgeht, bei der Lösungsfindung berücksichtigt wird. Der nächste zu expandierende Knoten auf dem Pfad zum Zielknoten wird auf Basis einer Bewertungsfunktion  $f(n)$  ausgewählt. Eine Komponente dieser Bewertungsfunktion ist eine heuristische Funktion  $h(n)$ , die die zu erwartenden Kosten des optimalen Pfades vom Knoten  $n$  zum Zielknoten berechnet [RN10a]. Im Falle des

Straßennetzes könnte hierzu die Länge der Luftlinie zwischen dem Knoten  $n$  und dem Zielknoten verwendet werden [HNR68].

Die einfachste Umsetzung dieser Strategie ist, nur die heuristische Funktion bei der Bewertung von Knoten heranzuziehen, sodass  $f(n) = h(n)$  gilt. Dieses Vorgehen wird auch Greedy Best-First Suche genannt, da in jedem Schritt versucht wird, so nahe wie möglich an den Zielknoten zu gelangen [RN10a]. Auf diese Weise werden die Suchkosten, also die Anzahl der expandierten Knoten zwar minimiert, es kann jedoch nicht garantiert werden, dass die gefundene Lösung optimal ist [HKH17].

Eine elaboriertere Umsetzung der Informed Search Strategie ist der A\*-Algorithmus zur Berechnung des kürzesten Pfades zwischen zwei Knoten [RN10a]. A\* basiert auf Dijkstra's Algorithmus und erweitert diesen um eine heuristische Funktion, um die Laufzeit zu reduzieren [PAG<sup>+</sup>20]. Die Bewertungsfunktion  $f(n)$  für den A\*-Algorithmus setzt sich zusammen aus den Kosten des optimalen Pfades vom Startknoten bis zum Knoten  $n$ ,  $g(n)$  und einer heuristischen Funktion  $h(n)$ , sodass gilt:

$$f(n) = g(n) + h(n) \tag{3.1}$$

Da verschiedene Heuristiken zur Konstruktion von  $h(n)$  gewählt werden können, stellt A\* streng genommen eine Familie von Algorithmen dar, wobei die Wahl einer Funktion  $h(n)$  einen spezifischen Algorithmus der Familie selektiert [HNR68]. Hart, Nilsson und Raphael, die in [HNR68] die A\*-Suche 1968 zum ersten Mal beschrieben haben, konnten nachweisen, dass A\* vollständig und optimal ist, wenn die gewählte Heuristik zulässig und konsistent ist. Das heißt, dass unter den angegebenen Voraussetzungen für die Heuristik, immer ein Pfad vom Start- zum Zielknoten gefunden wird (sofern dieser existiert) und dass dieser Pfad in jedem Fall optimal ist. Des Weiteren konnte gezeigt werden, dass A\* optimal effizient ist – es kann also keinen anderen optimalen Algorithmus geben, der garantiert weniger Knoten expandiert als A\* [RN10a].

## Anwendungen

In diesem Kapitel wird beschrieben, warum es unterschiedliche Konsistenzmodelle gibt. Außerdem werden die Unterschiede zwischen strengen Konsistenzmodellen (Linearisierbarkeit, sequentielle Konsistenz) und schwachen Konsistenzmodellen (schwache Konsistenz, Freigabekonsistenz) erläutert. Es wird geklärt, was Strenge und Kosten (billig, teuer) in Zusammenhang mit Konsistenzmodellen bedeuten.

### 4.1 Warum existieren unterschiedliche Konsistenzmodelle?

Laut [Mal97] sind mit der Replikation von Daten immer zwei gegensätzliche Ziele verbunden: die Erhöhung der Verfügbarkeit und die Sicherung der Konsistenz der Daten. Die Form der Konsistenzsicherung bestimmt dabei, inwiefern das eine Kriterium erfüllt und das andere dementsprechend nicht erfüllt ist (Trade-off zwischen Verfügbarkeit und der Konsistenz der Daten). Stark konsistente Daten sind stabil, das heißt, falls mehrere Kopien der Daten existieren, dürfen keine Abweichungen auftreten. Die Verfügbarkeit der Daten ist hier jedoch stark eingeschränkt. Je schwächer die Konsistenz wird, desto mehr Abweichungen können zwischen verschiedenen Kopien einer Datei auftreten, wobei die Konsistenz nur an bestimmten Synchronisationspunkten gewährleistet wird. Dafür steigt aber die Verfügbarkeit der Daten, weil sie sich leichter replizieren lassen.



## Zusammenfassung und Ausblick

In diesem Kapitel soll die Arbeit noch einmal kurz zusammengefasst werden. Insbesondere sollen die wesentlichen Ergebnisse Ihrer Arbeit herausgehoben werden. Erfahrungen, die z.B. Benutzer mit der Mensch-Maschine-Schnittstelle gemacht haben oder Ergebnisse von Leistungsmessungen sollen an dieser Stelle präsentiert werden. Sie können in diesem Kapitel auch die Ergebnisse oder das Arbeitsumfeld Ihrer Arbeit kritisch bewerten. Wünschenswerte Erweiterungen sollen als Hinweise auf weiterführende Arbeiten erwähnt werden.

---

## Literaturverzeichnis

- HKH17. HEUSNER, M., T. KELLER und M. HELMERT: *Understanding the Search Behaviour of Greedy Best-First Search*. Tenth Annual Symposium on Combinatorial Search, 8(1), 2017.
- HNR68. HART, PETER, NILS NILSSON und BERTRAM RAPHAEL: *A Formal Basis for the Heuristic Determination of Minimum Cost Paths*. IEEE Transactions on Systems Science and Cybernetics, 4(2):100–107, 1968.
- Mal97. MALTE, PETER: *Replikation in Mobil Computing*. Seminar No 31/1997, Institut für Telematik der Universität Karlsruhe, Karlsruhe, 1997.  
<http://www.ubka.uni-karlsruhe.de/cgi-bin/psview?document=/ira/1997/31>.
- Mos93. MOSBERGER, DAVID: *Memory Consistency Models*. Technical Report 93/11, University of Arizona, November 1993.
- PAG<sup>+</sup>20. PERALTA, FEDERICO, MARIO ARZAMENDIA, DERLIS GREGOR, DANIEL G. REINA und SERGIO TORAL: *A Comparison of Local Path Planning Techniques of Autonomous Surface Vehicles for Monitoring Applications: The Ypacarai Lake Case-study*. Sensors, 20(5):1488, mar 2020.
- RN10a. RUSSEL, STUART J. und PETER NORVIG: *Artificial Intelligence - A Modern Approach*, Kapitel 3.5, Seiten 92–102. Pearson Education, Inc, 2010.
- RN10b. RUSSELL, STUART J. und PETER NORVIG: *Artificial Intelligence - A Modern Approach*, Kapitel 3.4.6, Seiten 90,91. Pearson Education, Inc, 2010.

# A

---

## Glossar

DisASter	DisASter (Distributed Algorithms Simulation Terrain), A platform for the Implementation of Distributed Algorithms
DSM	Distributed Shared Memory
AC	Linearisierbarkeit (atomic consistency)
SC	Sequentielle Konsistenz (sequential consistency)
WC	Schwache Konsistenz (weak consistency)
RC	Freigabekonsistenz (release consistency)

---

## Arbeitsverteilung

Bitte tragen Sie auf dieser Seite ein, welches Mitglied Ihrer Gruppe welche Inhalte formuliert hat, da die Notengebung für jeden Studenten individuell vorgenommen wird.

### **Teilnehmer 1:**

Inhalte:

### **Teilnehmer 2:**

Inhalte:

### **Teilnehmer 3:**

Inhalte: