Informatik
Hauptcampus

H O C H
S C H U L E
T R I E R

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

1

Einleitung und Problemstellung

Unter Pathfinding bzw. Wegfindung versteht man in der Informatik die algorithmengestütze Suche nach dem optimalen Weg(en) von einem Startpunkt zu einem oder mehreren Zielpunkten.[Wik22]
In diesem Paper werden wir verschiedene Pfadplanungsalgorithmen wie den Dijkstra-Algorithmus und seine Variante, die häufig in Verkehrleitsystemen wie etwa Google Maps verwendet werden.

Um die Geschwindigkeit des Dijkstra-Algorithmus bei Blindsuchen zu optimieren werden A* und seine Varianten als Stand der Technik-Algorithmen für den Einsatz in statischen Umgebungen vorgestellt [KSDS21]

Das Thema Funktionsprinzipien und Anwendungen von Algorithmen zur Pfadplanung hat damals, wie heute einen wichtigen Stellenwert. Ob im Bereich der Netzwerkroutenplanung oder bei KI-Spielern in Computerspielen Pfadsuchalgorithmen sind so relevant wie nie. Weitere Beispiele für die Anwendung von Pfadsuchalgroithmen sind Robotik (z.B. Pakete im Logistikbereich), Planung von öffentlichen Verkehrsmitteln und Routenplanung von Navigationssystemen. Viele Bereiche im Alltag verwenden im Hintergrund Pfadsuchalgorithmen um den (kosten)günstigsten Weg zu finden, dabei ist es wichtig, dass diese effizient, akkurat und schnell sein müssen, damit die Hauptsysteme noch genügen Ressourcen übrig haben um gewünscht zu funktionieren. [FGK+21]

Wir werden uns in diesem Paper die Funktionsweise der wichtigsten Pfadsuchalgorithmen, angefangen mit den uninformierten Suchalgorithmen wie der Breiten- oder Tiefensuche, welche einen ersten Einblick in die Thematik geben sollen und die Rahmenbedingungen und Problemumgebungen veranschaulichen sollen; Über den Bellman-Ford-Algorithmus, der intuitiv den kürzesten Weg liefert und bei dem auch negative Kantenlängen möglich sind.[MS20] Bis hin zu durch Heuristiken optimierte und informierte Pfadsuchalgorithmen wie Dijkstra oder A*, die häufig in Verkehrleitsystemen wie etwa Google Maps verwendet werden, und wie sie in heutiger Zeit sonst eingesetzt werden können, veranschaulichen und jeweils (Pseudo-)Code- und Anwendungsbeispiele geben.[RNRR10a]

2

Algorithmen zur Pfadplanung

2.1 Was ist Pfadplanung?

Die Pfadplanung ist ein nichtdeterministisches, polynomialzeitliches ("NP") schweres Problem mit der Aufgabe, einen kontinuierlichen Pfad zu finden, der ein System von einer Ausgangs- zu einer Endkonfiguration verbindet. Die Komplexität des Problems steigt mit zunehmender Anzahl der Freiheitsgrade des Systems. Der zu verfolgende Pfad (der optimale Pfad) wird auf der Grundlage von Einschränkungen und Bedingungen bestimmt, z. B. im Bereich mobiler Roboter unter Berücksichtigung des kürzesten Weges zwischen den Endpunkten oder der minimalen Fahrzeit ohne Kollisionen. Manchmal werden Einschränkungen und Ziele gemischt, z. B. um den Energieverbrauch zu minimieren, ohne dass die Fahrzeit einen bestimmten Schwellenwert überschreitet [1].

2.2 Uninformierter Ansatz

Breitensuche:
Die Breitensuche gehört zu den uninformierten Suchalgorithmen, diese werden auch „blind“ genannt, weil bei ihrer Suche auf keine zusätzlichen Informationen (wie z.B. Wichtungen) zurückgegriffen wird. Bei der Breitensuche wird zunächst vom Wurzelknoten aus betrachtet alle verbunden Knoten ersten Grades besucht und dies Ebene für Ebene im Baum wiederholt bis alle Knoten besucht wurden. Die Breitensuche findet weitestgehend in der Graphentheorie seine Anwendung.[RNRR10b]

Tiefensuche:
Die Tiefensuche gehört ebenfalls zu den uninformierten Suchalgorithmen. Im Gegensatz zu der Breitensuche werden nicht die Ebenen nacheinander abgesucht sondern je Nachfolger angefangen beim Wurzelknoten werden bis sie keine weiteren Nachfolger mehr haben besucht. Erst dann wird der nächste Nachbar in der ersten Ebene besucht bis keine unbesuchten Knoten mehr vorhanden sind.Die Tiefensuche ist indirekt an vielen komplexeren Algorithmen beteiligt. Unter anderem kann die Tiefensuche auch für das Ermitteln von Zusammenhangskomponenten oder für das Erzeugen eines Irrgartens verwendet werden. [RNRR10c]

3

Optimierungsstrategien

4

Anwendungen

5

Zusammenfassung und Ausblick

In den letzten Jahren hat die Pfadplanung immer mehr an Relevanz gewonnen, zum Beispiel im Bereich des autonomen Autofahrens wird immer mehr Forschung im Bereich der Pfadplanungsalgorithmen betrieben. [KSDS21]

In diesem Paper haben wir aufeinander aufbauend Algorithmen vorgestellt, die in einem Programm mit bestimmten Eingaben und Umgebungen wie z.B Graphen verwendet werden können um von einem gegeben Start ein oder mehrere Ziele zu finden und dabei durch verschiedene Bewertungskriterien den besten Weg zu finden. Mit diesen Algorithmen kann ein Programm durch eine Sequenz von Aktionen sein Ziel erreichen. Diesen Prozess nennt man Suche. [RNRR10c]

- Bevor das Programm mit der Suche nach der besten Lösung beginnen kann, muss erst ein Ziel identifiziert und das Problem genau definiert werden.
- Suchalgorithmen behandeln Zustände und Aktionen atomar: Sie berücksichtigen keine interne Struktur, die sie besitzen könnten.
- Ein allgemeiner uninformierter TREE-SEARCH-Algorithmus berücksichtigt alle möglichen Wege, um eine Lösung zu finden, während ein informierter GRAPH-SEARCH-Algorithmus redundante Wege vermeidet.
- Uninformierte Pfadsuchalgorithmen haben nur Zugriff auf die Problemdefinition. Die grundlegenden Algorithmen sind wie folgt:
  - Die Breadth-first-Suche expandiert zuerst die flachsten Knoten; sie ist vollständig, optimal für einheitliche Pfadkosten, hat aber eine exponentielle Raumkomplexität.
  - Die Tiefensuche expandiert zuerst den tiefsten nicht expandierten Knoten. Sie ist weder vollständig noch optimal, hat aber eine lineare Raumkomplexität.
  - Die iterative Vertiefungssuche ist eine Wiederholung der Tiefensuche mit zunehmender Tiefenbegrenzung, bis ein Ziel gefunden wird. Sie ist vollständig, optimal für die Kosten pro Schritt, hat eine vergleichbare Zeitkomplexität wie die Breitensuche und eine lineare Raumkomplexität.
  - Der Greedy Dijkstra-Algorithmus der zwar bei der blinden Suche Zeit vergeudet aber dafür optimal ist und eine Trefferquote von 100% hat.
  - Die Optimierung durch eine bidirektionale Suche kann die Zeitkomplexität enorm reduzieren, ist aber nicht immer anwendbar und kann zu viel Speicherplatz beanspruchen.

- Informierte Suchmethoden können auf heuristische Funktionen zurückgreifen, die die Kosten einer Lösung schätzen
  - Der generische Best-First-Suchalgorithmus wählt einen Knoten für die Expansion gemäß einer Bewertungsfunktion aus.
  - Der Greedy best-first search expandiert Knoten mit minimalem heuristischem Funktionswert. Er ist nicht optimal, aber oft effizient.
  - A*-Suche expandiert Knoten mit minimalem Heuristischen Funktions- und Pfadkostenwerten. A* ist vollständig und optimal, vorausgesetzt die heuristische Funktion ist zulässig.
  - ALT-Algorithmen, die aufgebaut auf A* durch Preprocessing noch optimiertere Ergebnisse erzeugt.
  - Reach-based Pruning, welches den Dijkstra-Algorithmus um eine Metrik erweitert und dadurch optimiert.

[RNRR10c]

Alle diese Algorithmen haben diverse Vor- und Nachteile und daraus ergeben sich verschiedene Anwendungsbereiche auf welche wir in Kapitel 4 gesondert eingegangen sind.

Literaturverzeichnis

FGK+21. Foeada, Daniel, Alifio Ghifaria, Marchel Budi Kusumaa, Novita Hanafiahb und Eric Gunawanb: *A Systematic Literature Review of A* Pathfinding.* Elsevier B.V, 2021.

KSDS21. Karur, Karthik, Nitin Sharma, Chinmay Dharmatti und Joshua E. Siegel: *A Survey of Path Planning Algorithms for Mobile Robots.* Vehicles, 3(3):448–468, aug 2021.

MS20. Mukhlif, Fadhil und Abdu Saif: *Comparative Study On Bellman-Ford And Dijkstra Algorithms.* Int. Conf. Comm. Electric Comp. Net., 2020.

RNRR10a. Russell, Stuart J., Peter Norvig, Stuart Russell und Russell: *Artificial Intelligence: A Modern Approach*, Kapitel 3, Seite 64. Prentice Hall, Third Auflage, 2010.

RNRR10b. Russell, Stuart J., Peter Norvig, Stuart Russell und Russell: *Artificial Intelligence: A Modern Approach*, Kapitel 3.4.1, Seite 81. Prentice Hall, Third Auflage, 2010.

RNRR10c. Russell, Stuart J., Peter Norvig, Stuart Russell und Russell: *Artificial Intelligence: A Modern Approach*, Kapitel 3.4.3, Seiten 85,86. Prentice Hall, Third Auflage, 2010.

Wik22. Juli 2022. https://de.wikipedia.org/wiki/Pathfinding.

Sachverzeichnis

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:

- Kapitel 1: Einleitung und Problemstellung
- Kapitel 2.2: Uninformierter Ansatz
- Kapitel 5: Zusammenfassung und Ausblick

Teilnehmer 3:

Inhalte: