

Vom Suchen und Finden

Als ich meine erste Vorlesung zu KI besuchte, hatte ich Bilder aus Science-Fiction-Filmen im Kopf. Aber schon in der ersten Sitzung gab sich der Dozent alle Mühe, diese Vorstellungen zu zerstören: »Künstliche Intelligenz ist ein Teilgebiet der Informatik, das sich sehr viel mit Suchen beschäftigt.«¹ Wenn der Titel der Vorlesung nicht »Methoden der KI« gewesen wäre, sondern »Suchalgorithmen«, wer weiß, ob ich sie überhaupt freiwillig belegt hätte.

Ein Beispiel für ein Suchproblem, das jeder kennt, ist die Routenplanung beim GPS im Auto oder auf der Homepage der Bahn. Sie wollen zum Beispiel von Berlin nach München fahren (weil es da schöner ist) und der Computer sucht für Sie den schnellsten Weg. Automatische Routenplanung ist heute so alltäglich, dass wir uns keine Gedanken darüber machen, wie sie funktioniert. Aber manch einer erinnert sich vielleicht noch, dass man früher, bevor man losfuhr, zuerst eine Landkarte aus Papier studierte. Eine Route zu planen, erfordert eine gewisse Intelligenz, wenn Menschen das tun. Daher kann man bei automatischer Routenplanung zurecht von KI sprechen. Tatsächlich sind die Suchalgorithmen, die dabei zum Einsatz kommen, klassische Methoden der KI, wie sie in jedem Einführungskurs zu KI vorkommen. Um KI wirklich zu begreifen, muss man zunächst einmal Suchalgorithmen verstehen.²

Stellen Sie sich vor, wir stehen am Weltraumbahnhof auf dem Planeten Alderaan und wollen zum Planeten Endor reisen. Leider gibt es keinen Direktflug. Aber am Terminal hängt ein Shuttleplan, auf dem alle Verbindungen zwischen den Planeten übersichtlich aufgeführt sind:

1 Liebe Grüße an Helmar Gust.

2 Das Standardlehrbuch zu KI stammt von Russell & Norvig (2009), das ich jedem empfehlen kann, der tiefer einsteigen möchte.

3 Alderaan-Bespin
 2 Alderaan-Felucia
 3 Bespin-Alderaan
 2 Bespin-Corellia
 1 Bespin-Geonosis
 2 Corellia-Bespin
 1 Corellia-Dagobah
 1 Corellia-Felucia
 1 Dagobah-Corellia
 1 Dagobah-Endor
 1 Endor-Dagobah
 2 Felucia-Alderaan
 1 Felucia-Corellia
 1 Geonosis-Bespin

Die Zahl am Anfang der Zeile zeigt jeweils an, wie viele Monate die Reise dauert. Neben dem Shuttleplan hängt auch noch eine Netzkarte (Abbildung 1, die Zahlen geben wieder die Reisezeit in Monaten an). Wie kommen wir jetzt am schnellsten von Alderaan nach Endor?

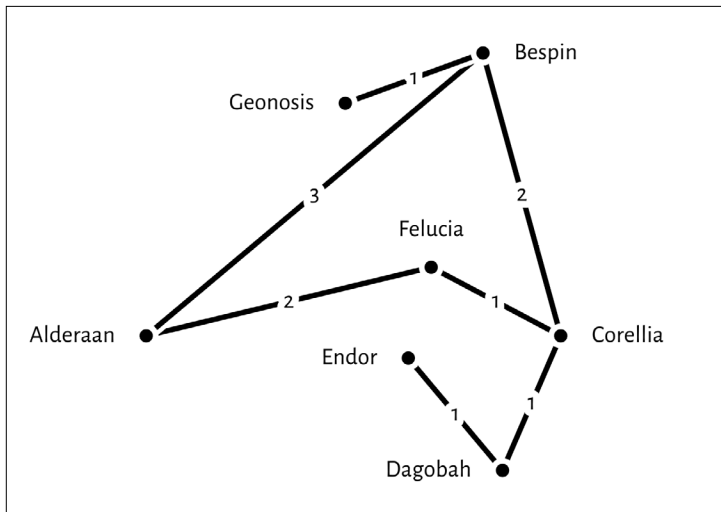


Abb. 1: Eine interplanetare Netzkarte

Wir müssen systematisch vorgehen

Durch Betrachten der Netzkarte finden Sie schnell die beste Route. Ein Computer, der keine Kamera hat, kann aber keine Karte lesen. Selbst mit Kamera ist Kartenlesen ein schwieriges KI-Problem. Wie wir in den vorausgehenden Kapiteln gesehen haben, kennt ein Computer nur Zeichen in seinem Speicher. Der Shuttleplan in Tabellenform, der voll von Zeichen ist, ist also erst einmal besser zur Verarbeitung in einem Computer geeignet. In dieser Tabelle gibt es zwei Shuttles, die in Alderaan abfliegen: eines nach Bespín und eines nach Felucia. Weil wir systematisch vorgehen wollen, schreiben wir beide Verbindungen – wie immer – auf ein Karopapier:

```
3 Alderaan-Bespín
2 Alderaan-Felucia
```

Der Suchalgorithmus geht jetzt nach einer einfachen Regel vor: Unter den möglichen Routen, die in den verschiedenen Zeilen stehen, wählt er die Zeile mit der bislang kürzesten Reisezeit aus und notiert auf dem Karopapier alle Weiterreisen, die von dort aus möglich sind (wobei wir nicht zurück an einen Ort reisen wollen, an dem wir auf der Route schon waren). Das macht der Algorithmus so lange, bis er die kürzeste Route zum Ziel gefunden hat. Für unsere Reiseplanung nach Endor heißt das: Bis Felucia waren es bisher zwei Monate und bis Bespín drei Monate, also macht der Algorithmus bei Felucia weiter, weil das bisher die kürzere Reiseroute ist. Von Felucia aus kann man nur nach Corellia reisen. Diese Reise dauert einen Monat. Wir verlängern die Reiseroute von Felucia entsprechend, zählen den einen Monat zu der bisherigen Länge der Reise hinzu ($2+1=3$) und notieren die neue Länge am Anfang der Zeile.

```
3 Alderaan-Bespín
3 Alderaan-Felucia-Corellia
```

Beide Routen, die wir bisher untersucht haben, sind jetzt gleich lang und keine ist schon in Endor angekommen. Von Bespín aus kann man auch nach Corellia fliegen oder aber nach Geonosis. Beide Möglichkeiten müssen im Blick behalten werden, deshalb notiert der Algorithmus beide.

- 5 Alderaan-Bespin-Coreellia
- 4 Alderaan-Bespin-Geonosis
- 3 Alderaan-Felucia-Coreellia

So macht der Algorithmus weiter, indem er immer die kürzeste Reiseroute verlängert und die, die in einer Sackgasse enden, entsprechend markiert.

- 5 Alderaan-Bespin-Coreellia-Alderaan-Bespin-Geonosis
- 5 Alderaan-Felucia-Coreellia-Bespin
- 5 Alderaan-Felucia-Coreellia-Dagobah-Endor

Der Algorithmus hat jetzt systematisch alle Routen untersucht, die man in höchstens fünf Monaten von Alderaan aus erreichen kann. Unter diesen haben wir eine Reiseroute nach Endor gefunden, die genau fünf Monate benötigt. Da wir unter den kürzeren Routen keine gefunden haben, die bis Endor kommt, ist die gefundene Route nach Endor die kürzeste.

Dieser Suchalgorithmus ist nicht besonders schlau. Sie haben wahrscheinlich anhand der Netzkarte intelligenter gesucht und zuerst einmal geprüft, wo man denn von Alderaan aus hinfliegen kann. Nämlich nach Bespin und nach Felucia. Aber dann haben Sie schnell gesehen, dass Bespin ja in der falschen Richtung liegt und dass eine Route über Felucia Sie viel näher an das Ziel Endor heranführt. Warum also, wie der Algorithmus, zuerst prüfen, wo man von Bespin aus hinfliegen kann, statt gleich auf die Felucia-Route zu setzen?

Mit Heuristiken geht es meistens schneller

Zur Verteidigung des Algorithmus muss man sagen, dass Sie durch die Netzkarte mehr Informationen besitzen als der Algorithmus, der nur den Shuttleplan in Tabellenform hat. In der Tabelle stehen nur die Reisezeiten zwischen den Planeten, zwischen denen es eine Shuttleverbindung gibt. Da der Algorithmus die Karte nicht sehen kann, kann er nicht wie Sie auf einen Blick erkennen, dass Felucia viel näher an Endor liegt als Bespin. Wie die Karte in Abbildung 2 zeigt, liegt Felucia nur 0,6 Monate Luftlinie von Endor entfernt, Bespin aber 2,1 Monate.

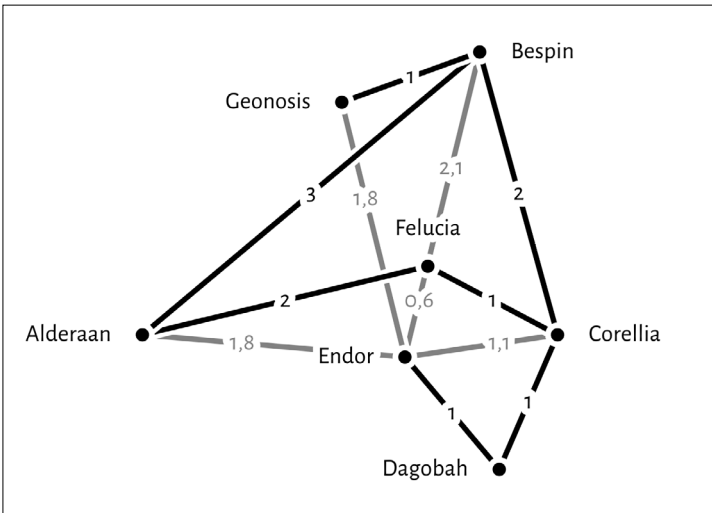


Abb. 2: Die interplanetare Netz Karte mit Luftlinien

Statt einem Bild der Karte stellen wir dem Algorithmus jetzt für jeden Planeten die Koordinaten zur Verfügung, damit er ausrechnen kann, wie weit ein Planet vom Zielplaneten Endor entfernt ist. Mit dieser Zusatzinformation kann der Algorithmus sich so schlau wie Sie verhalten. Er kann dann nämlich abschätzen, ob eine Route vielversprechend ist oder nicht. Das macht er, indem er für jede Route ausrechnet, wie lang sie im besten Fall sein wird. Für unsere Reiseplanung von Alderaan nach Endor heißt das: Von Alderaan aus kann man nach Bespin und Felucia fliegen, also notiert auch der verbesserte Algorithmus zunächst die zwei potenziellen Routen auf dem Karopapier.

3+2,1 Alderaan-Bespin

2+0,6 Alderaan-Felucia

Neben der bisherigen Reiselänge steht nun zusätzlich eine Abschätzung, wie lang die Restroute nach Endor bestenfalls wird, wenn wir über Bespin oder Felucia reisen. Von Alderaan nach Bespin sind es 3 Monate. Wenn wir von Bespin direkt nach Endor fliegen könnten, wären es von dort noch 2,1 Monate Reisezeit. Das kann der Algorithmus anhand der Koordinaten der Planeten leicht ausrechnen (und wir sehen das in der Netz Karte). Das heißt, dass die Route von Alderaan nach Endor über Bespin insgesamt mindestens 5,1 Monate dauert. Bestenfalls.

Weil man von Bepin aus nicht direkt fliegen kann, dauert es länger. Da die Reise von Alderaan nach Felucia 2 Monate dauert und Felucia 0,6 Monate Luftlinie von Endor entfernt ist, dauert diese Reiseroute mindestens 2,6 Monate.

Also sieht für den verbesserten Algorithmus die Reiseroute über Felucia vielversprechender aus. Daher untersucht er, wie es von Felucia aus weitergehen kann. Man kann von dort nur nach Corellia fliegen und das dauert einen Monat.

3+2,1 Alderaan-Bepin

3+1,1 Alderaan-Felucia-Corellia

Wir machen genauso weiter und verlängern immer die vielversprechendste Route, bis wir in Endor angekommen sind.

3+2,1 Alderaan-Bepin

5+2,1 Alderaan-Felucia-Corellia-Bepin

5+0,0 Alderaan-Felucia-Corellia-Dagobah-Endor

Da die so gefundene Route nach Endor insgesamt 5 Monate dauert und alle anderen möglichen Routen selbst im besten Fall länger dauern, ist das die kürzeste Route.

Diesen Suchalgorithmus, der für jede Route abschätzt, wie lange die Reise wohl insgesamt dauern wird, nennt man in der KI-Forschung kurz und prägnant: A^* (A-Stern ausgesprochen). Die Abschätzung, wie lange die Reisezeit zum Zielplaneten noch ist, nennt man auch »Heuristik«. Was A^* macht, ist deshalb ein Beispiel für eine sogenannte heuristische Suche. In unserem Planetenbeispiel bleibt diese heuristische Suche mit der Route möglichst nah an der direkten Fluglinie zwischen Start- und Zielplanet.³

Ohne eine Heuristik können Suchalgorithmen nicht wissen, in welcher Richtung sie nach ihrem Ziel suchen sollen. Das ist wie beim Topf-schlagen. Dem Kind werden die Augen verbunden und es sucht innerhalb des Stuhlkreises blind nach dem Topf. Das Kind könnte den Kreis systematisch absuchen, aber das Spiel besteht gerade darin, dass die

3 Der A^* -Algorithmus wurde zuerst von Hart, Nilsson & Raphael (1968) beschrieben und findet sich in jedem Lehrbuch zu Künstlicher Intelligenz, z.B. bei Russell & Norvig (2009).

anderen Kinder im Stuhlkreis »wärmer!« rufen, sobald sich das Kind in der Mitte in Richtung Topf bewegt, und »kälter!« rufen, sobald es von der Ideallinie abweicht. Auf diese Art wird es den Topf wesentlich schneller finden. Genau so hilft eine Heuristik: Sie leitet den Suchalgorithmus, indem sie ihm eine Abschätzung darüber gibt, wo es wärmer und kälter wird.

Meistens erweist sich die Heuristik, mit der Route möglichst nah an der direkten Fluglinie zu bleiben, als recht hilfreich bei der Suche nach der kürzesten Reiseroute. Wenn Sie von Berlin nach München fahren wollen, sehen Sie sich auf der Karte ja auch nicht zuerst die Straßen Richtung Hamburg an (da wird's kälter). Diese Heuristik muss aber nicht immer hilfreich sein. Manchmal kann sie auch auf eine falsche Fährte führen. Zum Beispiel, wenn Sie von Corellia nach Endor reisen wollen. Dann sieht der Algorithmus, dass er von Corellia nach Bepin, Dagobah und nach Felucia reisen kann. Da Felucia aber näher an Endor dran ist als Dagobah und deshalb vielversprechender aussieht, wird der Algorithmus zunächst auf eine falsche Fährte nach Felucia geführt. Wenn er dann merkt, dass er von Felucia aus nur noch weiter von Endor wegfiegen kann, besinnt er sich und untersucht die Route über Dagobah. Aber zunächst ist er falsch abgebogen.

Das ist Ihnen so ähnlich sicher auch schon passiert. Sie wissen, Ihr Fahrziel ist nahe dem Kirchturm in der Innenstadt. Sie fahren direkt auf den Kirchturm zu und kommen an eine Weggabelung. Die eine Straße führt mehr in Richtung Kirchturm als die andere. Ihre Heuristik ist: Ich nehme die Straße, die mich wahrscheinlich näher an das Ziel heranführt. Bis Sie merken, dass die Straße auf der Sie fahren, Sie ohne Abbiegemöglichkeit am Kirchturm vorbeigeführt hat. Heuristiken können einen auf die falsche Fährte führen. Aber meistens helfen sie, das Ziel schneller zu finden.

Wenn man mit heuristischer Suche nur schnell Fahrtrouten finden könnte, wäre sie ziemlich uninteressant für KI im Allgemeinen. Tatsächlich kann man aber mit und anderen ähnlichen Suchalgorithmen eine beträchtliche Anzahl von Problemen lösen.

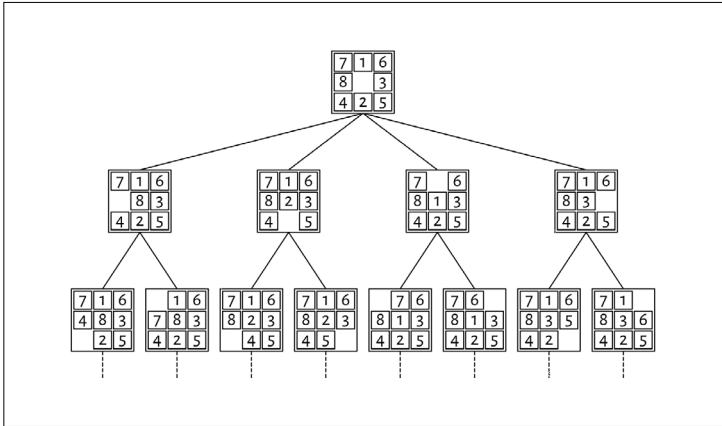


Abb. 3: Suchraum für das Schiebepuzzle

Viele Probleme sind schwere Suchprobleme

Ein weiteres Beispiel sind Schiebepuzzles. Das sind diese quadratischen Plastikdinge, in die acht quadratische Teile eingelassen sind, bei denen immer genau eine Lücke bleibt, in die man die nebenliegenden Teile hineinschieben kann, wodurch am Ursprungsort eine Lücke entsteht. Die Aufgabe lautet, alle acht Teile in eine bestimmte Ordnung zu bringen. In Abbildung 3 sieht man solche Schiebepuzzles, bei denen die Teile durchnummeriert sind. Die Zahlen sind durcheinander: Im Schiebepuzzle ganz oben steht links oben die 7, rechts daneben die 1, und so weiter. Den Zustand des Puzzles am Anfang des Spiels, bevor wir probieren die Puzzle-Teile in die richtige Reihenfolge zu bringen, nennt man »Startzustand«. Das Ziel des Spiels ist es, die 1 nach links oben zu bringen, rechts daneben die 2, rechts oben die 3, in der mittleren Reihe ganz links die 4, und so weiter, mit der Lücke rechts unten. Diesen Zustand des Puzzles nennt man »Zielzustand«. Ein Zug bringt uns von einem Zustand des Problems in einen anderen Zustand. Schieben wir zum Beispiel im Startzustand die 8 nach rechts, erreichen wir den Zustand in der Abbildung links unter dem Startzustand. Abbildung 3 zeigt alle Zustände, die wir in zwei Zügen vom Startzustand aus erreichen können. Die Menge aller Problemzustände nennt man den »Suchraum« eines Problems. Statt im Weltraum von Planet zu Planet zu reisen, reisen wir jetzt im Suchraum von Problemzustand zu Problemzustand. Und wie vorher suchen wir einen Weg vom Start zum Ziel. Am

besten den kürzesten Weg. Suchalgorithmen können uns helfen, eine Handlungsabfolge zu finden, die uns vom Startzustand zum Zielzustand bringt. Eine solche Handlungsabfolge nennt man einen ›Plan‹.

Im Vergleich zu unserem kleinen interplanetaren Reiseproblem ist es in Schiebepuzzles deutlich schwieriger, den optimalen Reiseplan zu finden. Meist braucht man etwa zwanzig Züge, um so ein Puzzle zu lösen. Von jedem Zustand aus gibt es mindestens zwei mögliche Züge, wenn wir das Zurücknehmen eines Zuges mitzählen. Und wenn die Lücke sich nicht gerade in einer Ecke befindet, sind es drei mögliche Züge, oder sogar vier in der Mitte. Da es also bei jedem der zwanzig Züge mindestens zwei mögliche Handlungen gibt, gibt es mindestens 2^{20} , also ungefähr eine Million, mögliche Pläne. Der Suchraum aller Problemzustände ist etwas kleiner. Die Lücke kann an 9 verschiedenen Orten sein, die 8 nur noch an den 8 übrigen, die 7 an den dann noch übrigen 7 Orten, und so weiter. Die Anzahl der möglichen Zustände ist also $9 \cdot 8 \cdot 7 \cdot \dots \cdot 1$, was mit $9!$ (ausgesprochen: neun Fakultät) abgekürzt wird. Ausgerechnet sind das ungefähr 350.000 verschiedene Problemzustände. Wenn man statt einem 3×3 -Puzzle mit 8 Teilen, ein 4×4 -Puzzle mit 15 Teilen lösen will, dann braucht man üblicherweise etwa 50 Züge. Das entspricht mindestens 2^{50} , also etwa einer Billion möglicher Pläne und $16!$, ungefähr 20 Billionen, verschiedenen Zuständen.⁴

Das ist ein wahnsinnig großer Suchraum! Zum Vergleich: In der Milchstraße gibt es geschätzt 250 Milliarden Sterne. Der Suchraum für das 4×4 -Schiebepuzzle ist also 80-mal größer, als eine interstellare Netzkarte für Shuttlereisen in unserer Galaxie wäre. Wie findet man unter all diesen möglichen Zuständen den kürzesten Weg zum Zielzustand? Wie findet man überhaupt zum Ziel? Das ist, als würde man die sprichwörtliche Nadel im Heuhaufen suchen. Sucht man systematisch alle Zustände ab, muss man schlimmstenfalls 20 Billionen Zustände untersuchen. Genau wie bei der Routenplanung benutzt man also besser eine Heuristik, um die Suche zielorientierter und damit intelligenter zu machen, denn man möchte keine Rechenleistung dafür verplempern, unnötige Züge zu untersuchen, die einen vom Ziel wegführen. Glücklicherweise können wir auch für das Schiebepuzzle den aktuellen Abstand zur Lösung mit einer Heuristik abschätzen, die unsere Suche leiten kann: Wir zählen, wie viele Teile noch nicht am richtigen Platz

4 Korf & Schultze (2005) haben alle möglichen Zustände des 4×4 -Schiebepuzzles berechnet.

sind. Wenn sie jeweils nur ein Feld davon entfernt wären, ist das die Anzahl der Züge, die wir bestenfalls noch brauchen. Das ist unsere Luftlinie für das Schiebepuzzle. Mit dieser Heuristik kann man auch für 4x4-Schiebepuzzles mit ihrem wahnsinnig großem Suchraum leicht die kürzeste Lösung finden.

Wie lösen Sie solche Puzzles? Jedenfalls nicht, indem Sie wie bei unserer Weltraumreise die ganze Route im Voraus planen. Ein Reiseplan mit allen Zügen wäre bei einem 4x4-Schiebepuzzle üblicherweise etwa 50 Züge lang. So weit können Sie ohne Computer – oder zumindest ohne Papier und Bleistift und viel Geduld – gar nicht im Voraus planen. Vielmehr werden Sie die Sache wahrscheinlich nur mit einem groben Plan angehen. Erst bringen Sie die 1 nach links oben, danach die 2 daneben, und so weiter. Sie folgen der Heuristik, dass ein Zustand näher am Zielzustand ist, je mehr Teile schon am richtigen Platz sind. Deshalb schieben Sie auch ungern Teile, die schon am richtigen Platz sind, wieder weg. Und falls Sie das machen müssen, dann reparieren Sie das schnell wieder. Diese Züge planen Sie wahrscheinlich im Kopf im Voraus, und sehen so, dass Sie sich manchmal erst vom Ziel entfernen müssen, um ihm im Ergebnis näherzukommen.

Wenn Sie nicht, bevor Sie anfangen, alle Züge bis zum Ende durchdacht haben, können Sie nicht wissen, ob ein vielversprechender Zug, den Sie machen, auf dem kürzesten Weg liegt, oder ob er zu einem Umweg führt (zum Beispiel, wenn Sie von Corellia nach Endor wollen und nach Felucia fliegen, weil Felucia so nah an Endor dran ist). Je weiter Sie in die Zukunft planen, umso besser wird Ihre Lösung sein.

Genauso ist es auch beim Schach. Auch Schach ist ein Suchproblem – mit der Anfangsstellung als Startzustand und mehreren möglichen Zielzuständen. Der Suchraum ist jedoch ungleich größer als bei einem 4x4 Puzzle! Aber auch hier suchen die Spieler eine Abfolge von Zügen, die sie zum Ziel führen. Und das Ziel ist, den Gegner matt zu setzen. Beim Schachspielen habe ich zumindest einen groben Plan, und beim Endspiel bemühe ich mich, möglichst viele Züge im Voraus zu bedenken. Manchmal gehen meine Pläne auch auf (ich liebe es, wenn ein Plan funktioniert). Meistens geht aber nichts nach Plan. Denn der Erfolg des Plans hängt davon ab, wie sich der Gegner verhält. Damit ein Plan gut ist, muss er auch alle möglichen Züge des Gegners berücksichtigen. Das kriegt mein Schachcomputer immer besser hin als ich.

Dass ein Plan schiefgeht, und ich umplanen muss, passiert nicht nur bei Spielen, in denen sich mein Gegner nicht an meinen Plan hält.

Es passiert in der wirklichen Welt auch bei der Reiseplanung. Fahre ich mit der Bahn und muss umsteigen, kann es passieren, dass ich wegen einer Verspätung meinen Anschluss verpasse. Wenn ich weiß, dass der Zug oft Verspätung hat, plane ich vielleicht erst gar nicht damit, dass ich den knappen Anschluss erreiche, oder ich habe einen Plan B, oder ich muss halt umplanen, falls der Anschluss tatsächlich weg ist. All dieses intelligente Verhalten können Computer dank Suchalgorithmen auch zeigen.

Unabhängig davon, ob ein Mensch oder ein Computer ein Problem lösen soll, ein Problem besteht immer aus einem Startzustand, der beschreibt, wie die Welt ist, und einem Zielzustand, der beschreibt, wie wir die Welt gerne hätten. Wir sind in Berlin und wollen nach München. Die Schachfiguren stehen in der Anfangsstellung und wir wollen eine Stellung erreichen, in der der Gegner matt gesetzt ist. Die Einzelteile eines Autos liegen in den Lagern der Zulieferer über ganz Europa verteilt und sollen alle zu einer bestimmten Zeit im Werk zur Endmontage sein. Wir wiegen zu viel und wollen fünf Kilo Gewicht verlieren. Wir produzieren als Gesellschaft viel zu viel CO₂ und wollen unseren CO₂-Ausstoß auf null reduzieren. Wenn wir also längst Suchalgorithmen haben, die Probleme für uns lösen können und uns bei vielen Problemen sogar überlegen sind, warum werden nicht alle unsere Probleme längst von Computern gelöst?

Viele Probleme sind ungenügend definiert

Ein Grund ist, dass wir Suchalgorithmen nur anwenden können, falls sowohl der Startzustand des Problems, als auch der Zielzustand des Problems sowie die Handlungsmöglichkeiten so klar definierbar sind, dass wir sie als Zeichen auf das Karopapier einer unserer Papier- und Bleistift-Maschinen schreiben können. Außerdem haben schon scheinbar einfache Probleme, wie das Schiebepuzzle, einen so großen Suchraum, dass sie nur durch den Einsatz von Heuristiken effizient zu lösen sind. Und diese Heuristiken müssen im Normalfall dem Computer auch erst einprogrammiert werden. Ohne Programmierer lösen Computer keine Probleme für uns.

Bei unserer Reise von Alderaan nach Endor war es leicht, den Suchraum computerverständlich zu beschreiben und dem Suchalgorithmus eine Heuristik mitzugeben. Das war auch leicht bei den Schiebepuz-

zles. Bei anderen Problemen kann es aber schon ziemlich mühselig sein, den Startzustand zu beschreiben. Zur Planung der Energiewende muss man erst mal wissen, wo eigentlich welche Mengen an CO₂ anfallen. Wie viel CO₂ wird bei der Verstromung von Braunkohle ausgestoßen? Wie viel beim Heizen? Wie viel im Verkehr? Von diesen Informationen hängt maßgeblich ab, ob man mehr Anstrengungen in den Kohleausstieg, die Gebäudedämmung oder den Ausbau des Bahnverkehrs stecken sollte. Auch ist oft unklar, ob überhaupt schon alle Handlungsmöglichkeiten auf dem Tisch liegen. Gibt es nicht vielleicht zur Einführung von CO₂-Steuer oder Zertifikatehandel andere Handlungsmöglichkeiten, die die Politik bisher nicht bedacht hat? Und wie genau wirken sich diese Maßnahmen auf den Zustand der Welt aus? Ohne all diese Informationen kann weder ein Mensch noch ein Computer einen vernünftigen Plan für die Energiewende machen.

Im Fall der Energiewende scheitert die Planung aber nicht unbedingt an mangelnder Information über den Zustand der Welt und die verfügbaren Handlungsmöglichkeiten, denn zu all diesen relevanten Fragen gibt es gute Forschung. Und tatsächlich können KI-Methoden zur Planung der Energiewende gut genutzt werden. Der Zielzustand lautet also: Wir wollen CO₂ reduzieren! KI-Algorithmen finden schnell eine einfache Lösung: Schalte alle Kohlekraftwerke und Ölheizungen ab und lass alle Autos stehen. Aber das wollen wir dann doch nicht. Es stellt sich leider heraus, dass wir nicht nur das eine Ziel haben, sondern mehrere gegensätzliche Ziele: Wir wollen CO₂ reduzieren, aber es darf nicht viel Geld und keine Arbeitsplätze kosten. Die Stromversorgung wollen wir auch behalten, und unsere Autos sowieso. Das wahre Problem bei der Planung der Energiewende sind solche Zielkonflikte. Damit uns eine KI bei der Planung der Energiewende helfen kann, müssen wir dem Suchalgorithmus erst mitteilen, was unser angestrebter Zielzustand ist, und was uns die Lösung kosten darf. Wir müssten wissen, was wir eigentlich wollen. Und oft tun wir das nicht.⁵

Im Kleinen taucht diese Schwierigkeit auch schon bei der Anwendung von KI bei der Routenplanung der Bahn auf. Auf den ersten Blick lautet das Ziel, von Berlin nach München zu kommen. Wenn wir die Route genau so planen, wie wir das in unserem Planetenbeispiel gemacht haben, kann ein Suchalgorithmus für uns die schnellste Route

5 Eine Liste mit Vorschlägen, wie KI helfen kann, den Klimawandel zu bekämpfen, findet sich bei Rolnick et al. (2019).

finden. Aber vielleicht wollen wir stattdessen die kürzeste oder die billigste Route. Dann muss der Suchalgorithmus mit der Strecke oder dem Preis statt mit der Reisezeit rechnen. Aus Sicht des Algorithmus macht es keinen Unterschied, ob die ›Kosten‹, die minimiert werden sollen, in Euros, in Minuten oder in Kilometern gemessen werden. Meist kann man dafür irgendwo ein Häkchen setzen. Meine eigenen Kriterien für die optimale Route sind allerdings stimmungsabhängig und deutlich komplexer: Der Sprinter ist zwar der schnellste Zug, aber ich finde ihn unverhältnismäßig teuer, weshalb mir eine etwas langsamere Verbindung lieber ist, aber nur, falls ich nicht mehr als einmal umsteigen muss, und die Umsteigezeit nicht zu knapp ist, um die Wahrscheinlichkeit möglichst gering zu halten, dass ich den Anschluss verpasse. Habe ich es eilig oder möchte ausschlafen, nehme ich aber trotzdem den Sprinter. Es ist oft nicht leicht zu wissen, welche Ziele man eigentlich hat. Und nicht selten ändern sich unsere Ziele sprunghaft. Wie soll ein KI-System es uns da recht machen?

Halten wir fest: Suchalgorithmen bilden den Kern jedes KI-Systems, das Probleme lösen kann, sei es zur Routenplanung, zum Schachspielen oder zur Planung der Energiewende. Das machen die Suchalgorithmen aber nicht selbständig. Jemand muss dem Algorithmus den Startzustand, den Zielzustand, die Handlungsmöglichkeiten, die Heuristik und die Kosten, die minimiert werden sollen, einprogrammieren. Das praktische Problem dabei ist, dass es oft viele gegensätzliche Ziele gibt.

Die Interessen der Hersteller von KI-Systemen sind nicht unbedingt auch meine. Die Bahn hat vielleicht mehr Interesse daran, dass die Hochgeschwindigkeitszüge ausgelastet sind, als dass ich die günstigste Route finde. Facebook hat vielleicht mehr Interesse daran, viele Klicks zu generieren, als dass ein KI-Programm die Falschmeldungen aus meinem Newsfeed herausfiltert. Und der Hersteller eines selbstfahrenden Autos hat vielleicht ein größeres Interesse daran, bei einem drohenden Unfall das Leben seiner Kunden zu schützen, als das Leben der anderen Verkehrsteilnehmer.

Die Suchalgorithmen haben selbst hingegen keine eigenen Ziele! Obwohl das – wie wir aus Science-Fiction wissen – durchaus vorstellbar ist, so muss man sich beim aktuellen Stand der KI-Forschung noch keine ernsthaften Sorgen machen, dass Algorithmen bald eigene Wünsche und Ziele entwickeln. Klassische KI-Systeme, wie sie lange zum Beispiel beim Schach eingesetzt wurden, können ihre erstaunlichen Leistungen nur deshalb erbringen, weil Menschen die Probleme vorher

entsprechend für die KI-Systeme aufbereitet haben. Wenn ein KI-System ein Problem löst, hat vorher ein Mensch das Problem in ein Suchproblem übersetzt, das der Computer verarbeiten kann. Man muss sich daher immer fragen, wer hinter einem KI-System steht.