

5 Optimierungungsverfahren

In den vorangegangenen Kapiteln wurde aus den Gesetzmäßigkeiten der Physik des Aufpralls eine Bewertungsfunktion abgeleitet. Sie dient dazu, jede Entscheidung des Auslösealgorithmus zu bewerten, indem sie die Schwere der Fehlentscheidung misst. Die Parameter des Auslösealgorithmus sind auf dieser Grundlage so zu optimieren, dass insgesamt das geringste Maß an Fehlentscheidungen erreicht wird. Dieses Kapitel soll eine kurze Einführung in die Begriffsmäßigkeiten und Besonderheiten der Optimierung geben und verschiedene Optimierungsverfahren vorstellen. Die Optimierungsverfahren werden in den folgenden zwei Kapiteln zur Einstellung der Parameter des Auslösealgorithmus herangezogen.

Optimierung bedeutet allgemein, die Zielfunktion zu maximieren bzw. zu minimieren und damit diejenige Einstellung zu finden, die das beste Ergebnis gewährleistet. In diesem Fall soll ein möglichst geringes Maß an Fehlentscheidungen sichergestellt werden. Daher steht im folgenden der Begriff Optimierung für Minimierung. Ob es bei der Optimierung um eine Minimierung oder eine Maximierung handelt, hängt meist nur von der Formulierung des Problems ab. Möglichst wirtschaftliches Autofahren, minimiert den Kraftstoffverbrauch und maximiert die Reichweite einer Tankfüllung. Die Anwendbarkeit der Optimierungsverfahren hängt nicht von der Richtung der Zielfunktion ab.

Bei der Optimierung sind außerdem folgende Klassifizierungen zu unterscheiden:

1. Parameteroptimierung und Funktionenoptimierung

Die Parameteroptimierung besteht darin, eine endliche Zahl von Stellschrauben, also kontinuierliche Parameter, und von Schaltern, also diskrete Parameter, möglichst gut einzustellen. Möchte man bei einer HiFi-Anlage den Hörgenuss seinen persönlichen Vorlieben anpassen, so stehen als Stellschrauben die Lautstärke und die Höhen- und Tiefenregler zur Verfügung. Dazu kommen Schalter, mit denen man zwischen verschiedenen Betriebszuständen wie Rauschunterdrückung (Dolby A, B, C) und Loudness wählen kann. Unter Funktionenoptimierung versteht man die Suche nach einer Funktion, die eine Aufgabe bestmöglich erfüllt. Ein Beispiel ist die Ideallinie eines Rennfahrers, um ein Bahnstück in kürzestmöglicher Zeit zu durchfahren. Diese Arbeit

befasst sich mit der Parameteroptimierung beim Auslösealgorithmus.

2. Statische und dynamische Optimierung.

Im vorliegenden Fall der Luftsackauslösung muss die Parametereinstellung vor der Herstellung des eingebetteten Systems, spätestens jedoch vor der Auslieferung und damit statisch erfolgen. Eine nachträgliche Anpassung ist nicht vorgesehen. Die dynamische Optimierung erfolgt während des Betriebs. Im Beispiel der Stereoanlage entspricht dies dem Lauterdrehen des Autoradios bei zunehmenden Fahrtgeräuschen. Sie kann auch automatisch erfolgen. Ist die Zielfunktion, also die Güte der Einstellung, durch das System selbst messbar, so kann sich das System selbstständig optimieren. Eine Entsprechung zwischen statischer und dynamischer Optimierung sind gewissermaßen Steuerung und Regelung. Sie unterscheiden sich darin, ob das Ergebnis im laufenden Betrieb rückgekoppelt wird und die Einstellungen angepasst werden.

Der Aufwand und die Zuverlässigkeit der Optimierung hängt von der Struktur der Zielfunktion ab. Dabei erleichtern bestimmte Eigenschaften das Finden des Optimums, beispielsweise ein monotoner Zusammenhang zwischen Stellwerten und Güte. Bei kontinuierlichen Parametern gibt es stetige und unstetige Funktionen, konvexe und alternierende. Ist die Anzahl der lokalen Tiefpunkte überschaubar, ist der globale Tiefpunkt meist leicht zu finden. Abbildung 30 illustriert vereinfacht verschiedene Zielfunktionen im zweidimensionalen Definitionsbereich. Weiss steht für die perfekte Einstellung, dunkle Punkte sind schlecht. Das Finden des hellsten Punktes gestaltet sich unterschiedlich schwer, wie leicht einzusehen ist. Außerdem ist zu erkennen, dass für verschiedene Probleme unterschiedliche Vorgehensweisen bei der Optimierung zum Erfolg führen.

Die gezeigten Zielfunktionen sind fiktiv. Abbildung 31 zeigt die Zielfunktion eines einfachen Auslösealgorithmus mit zwei Parametern, einer Filterbreite und einer Schwelle [D2]. Zielfunktionen von Algorithmen mit mehreren Parametern, können auf diese Weise nicht illustriert werden.

Nebenbedingungen zur Gültigkeit von Stellwerten erschweren meist die Optimierung zusätzlich. Stellwerte können unbeschränkt oder in eine oder beide Richtungen begrenzt sein. Häufig kommt es vor, dass die einstellbaren Werte gegenseitig voneinander abhängen. Solche Nebenbedingungen können in die Zielfunktion aufgenommen werden, indem man für ungültige Belegungen ei-

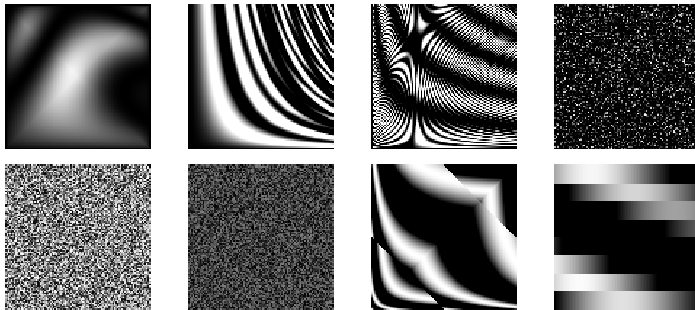


Abbildung 30: Verschiedene Zielfunktionen

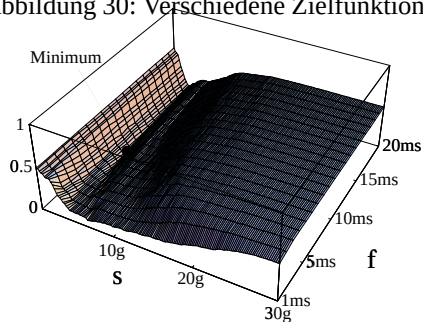


Abbildung 31: Zielfunktion eines einfachen Algorithmus

ne schlechte Bewertung festlegt, der für ein Optimum nicht in Frage kommt. Das beeinträchtigt die Optimierung dadurch, dass das allgemeine Verfahren häufig unzulässige Einstellungen ausprobiert. Es bietet sich daher an, die Nebenbedingungen von vorneherein in die Optimierung aufzunehmen. Häufig lassen sich die Nebenbedingungen auch durch Umformen und Ersetzen der Parameter vereinfachen.

Dazu zwei Beispiele: Ein chemischer Prozess, der optimiert werden soll, erfolgt in zwei Phasen. Parameter sind die Zeitpunkte des Phasenwechsels t und die Gesamtdauer T . Die Nebenbedingung lautet $T > t > 0$. Diese gegenseitige Abhängigkeit kann verhindert werden, durch Ersetzung mit den Phasendauern t_1 und t_2 . Die Nebenbedingung lautet dann nur noch $t_1, t_2 > 0$.

Gesucht wird ein bestimmter Punkt (x, y, z) im Raum. Die Nebenbedingung schreibt vor, dass er auf der Einheitskugel liegen soll, also $x^2 + y^2 + z^2 = 1$. Der Suchraum sei zweidimensional. In diesem Beispiel ist es sinnvoll, die drei Parameter x, y, z durch zwei zu ersetzen, beispielsweise durch Kugelkoordinaten mit Längengrad λ und Breitengrad ϕ . Der Radius ist 1, die Umformung lautet dann $(x, y, z) = (\cos(\phi) \cos(\lambda), \cos(\phi) \sin(\lambda), \sin(\phi))$. Nebenbedingung: $0 \leq \lambda < 360^\circ, -90^\circ \leq \phi \leq 90^\circ$

Die Nebenbedingungen in dieser Arbeit wurden so weit wie möglich auf die genannten Arten eliminiert, dass heisst die Auslösealgorithmen sind so parametrisiert, dass sie weitgehend orthogonal, also voneinander unabhängig sind.

Den dargestellten allgemeinen Schwierigkeiten bei der Optimierung wurde durch die Entwicklung verschiedener Optimierungsverfahren Rechnung getragen. Ihnen ist gemeinsam, dass das Optimum aufgrund einer festgelegten Systematik gesucht wird.

In der Regel arbeiten die Optimierungsverfahren mit Schrittweiten und Wahrscheinlichkeitsdichten, um den Zielraum nach dem Optimum abzusuchen. Nimmt man zur Berücksichtigung von Nebenbedingungen Parametertransformationen vor, wie sie oben dargestellt wurden, ist jedoch Vorsicht geboten, sobald es um Schrittweiten und Wahrscheinlichkeitsdichten geht. Dies soll in einem weiteren Beispiel veranschaulicht werden. Die Durchschnittstemperatur auf der Erde lasse sich folgendermaßen bestimmen: 1000 Punkte auf der Erde werden durch zufällige, also gleichverteilte Paare von Längen- und Breitengraden aus dem Definitionsbereich bestimmt. Die Messungen an diesen Punkten werden gemittelt. Dies führt jedoch systematisch zu einer zu niedrigen Temperatur, da die Dichte der Messpunkte pro Fläche über den Polarflächen höher als in der Nähe des Äquators ist.

Es gibt eine große Zahl an Optimierungsverfahren. Dabei sind die einzelnen Optimierungsverfahren jeweils nur für bestimmte Arten oder Strukturen von Zielfunktionen geeignet. Es gibt also kein bestes Verfahren für sämtliche Arten von Zielfunktionen. Im folgenden werden zunächst die Optimierungsverfahren beschrieben und deren Ergebnisse bei Anwendung auf die in dieser Arbeit verwendete Zielfunktion dargestellt und verglichen. Es handelt sich dabei nicht um „neueste Forschungsergebnisse“, sondern um klassische Verfahren, die sich bei ähnlichen Problemen bewährt haben. Die zugrundeliegende Auslösealgorithmen werden im folgenden Kapitel vorgestellt. Beispiele für die

Optimierungsverfahren werden zunächst jeweils separat dargestellt. In einem zweiten Schritt werden dann die Ergebnisse aus der kombinierten Anwendung von Optimierungsverfahren dargestellt. Die Abbildungen verdeutlichen jeweils das Verfahren in einem zweidimensionalen Definitionsbereich (links). Zur besseren bildlichen Veranschaulichung der Suchsystematik wurden für die Darstellungen in der jeweils linken Grafik vereinfachte Zielfunktionen ausgewählt. Der geometrischen Darstellung in der jeweils rechten Kurve und der Ergebnisdatei liegen jedoch die für die Auslöseoptimierung verwendete Zielfunktion zugrunde. Die rechte Kurve zeigt dabei jeweils die Zielfunktion Z bei 1000 Optimierungsschritten. Dargestellt ist jeweils nur das Ergebnis für den besten gefundenen Parametersatz. Verschlechterungen werden bei den meisten Optimierungsverfahren verworfen.

5.1 Zufälliges Raten

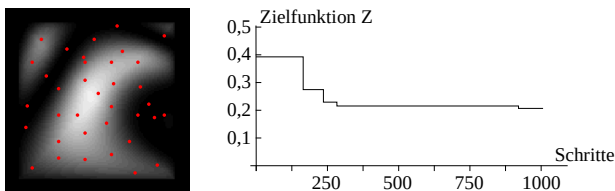


Abbildung 32: Zufälliges Raten

Das zufällige Raten ist kein eigenständiges Optimierungsverfahren. Es ist Teil anderer Optimierungsmethoden. Ausprobieren ermöglicht es, Startpunkte für ein Abstiegsverfahren finden, die erste Generation eines genetischen Verfahrens zu erzeugen oder Punkte für eine simulierte Auskühlung zu finden. Außerdem lassen sich Aussagen über den Mittelwert und die Varianz der Zielfunktion treffen. Dies ist später nützlich für die Bewertung des gefundenen Optimums. Dabei spielt die Art der Parametrisierung eine Rolle, wie das Beispiel der globalen Durchschnittstemperatur zeigt.

Abbildung 33 zeigt einen Ausschnitt aus der Ergebnisdatei des Optimierungsprogramms. In diesem Verfahren werden jeweils nur Verbesserungen gespeichert.

Zeit	Z	P[0]	P[1]	P[2]	P[3]	P[4]	P[5]	P[6]	P[7]	P[8]	P[9]
13:57	0.397	-10.00	1.00	0.00	0.00	1.00	-30.00	0.00	1.00	-30.00	0.00
13:57	0.282	-2.00	14.00	25.70	0.00	11.00	-3.95	3.00	17.00	-0.94	2.00
13:57	0.233	-10.00	10.00	27.36	2.00	2.00	-13.94	2.00	4.00	-21.25	2.00
13:57	0.221	-3.00	2.00	12.14	0.00	18.00	-16.36	0.00	5.00	-15.23	3.00
13:58	0.206	-6.00	1.00	5.56	0.00	4.00	-10.96	1.00	14.00	-4.96	2.00

Abbildung 33: Ergebnisdatei bei zufälligem Raten

chert. Daher ist die Folge der Bewertungen Z monoton fallend. Die Wahrscheinlichkeit, dass eine Verbesserung auftritt, nimmt exponentiell ab. $P[0] - P[9]$ stellen die zu optimierenden Parameter dar. Wie man sieht, besteht bei diesem Optimierungsverfahren kein unmittelbarer Zusammenhang zwischen aufeinanderfolgenden Zeilen, da jede Parameterbelegung zufällig und ohne Kenntnis der vorhergehenden erzeugt wird.

5.2 Vollständige Suche

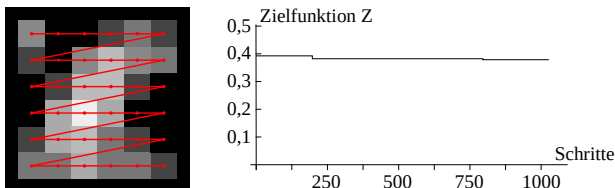


Abbildung 34: Vollständige Suche

Die vollständige Suche ist ein Optimierungsverfahren für Probleme mit endlichem, diskreten Definitionsbereich. Bei kontinuierlichen Variablen scheidet das Verfahren aus. Aber auch schon bei wenigen diskreten Variablen stößt das Verfahren meist an Grenzen des durchführbaren Aufwands.

Als Verfahren für die teilweise Optimierung ist es jedoch nützlich. So kommt es vor, dass nur ein oder zwei Parameter optimiert werden sollen. Dazu werden die anderen festgehalten, also die Intervallgrenzen zusammengeschoben, und die freien Parameter systematisch durchprobiert. Auch eine grobe Optimierung lässt sich durchführen. Dazu stellt man für kontinuierliche Parameter eine große Schrittweite ein. Der Lösungsraum wird in einem groben Raster

systematisch abgesucht. Abbildung 35 zeigt den Anfang der Ergebnisdatei.

Zeit	Z	P[0]	P[1]	P[2]	P[3]	P[4]	P[5]	P[6]	P[7]	P[8]	P[9]
13:59	0.394	-10.00	1.00	0.00	0.00	1.00	-30.00	0.00	1.00	-30.00	0.00
13:59	0.390	-10.00	1.00	0.00	0.00	1.00	-30.00	0.00	1.00	-30.00	1.00
13:02	0.377	-10.00	1.00	0.00	0.00	1.00	-30.00	0.00	1.00	-30.00	2.00
13:02	0.377	-10.00	1.00	0.00	0.00	1.00	-30.00	0.00	1.00	-30.00	3.00
13:04	0.377	-10.00	1.00	0.00	0.00	1.00	-30.00	0.00	1.00	-29.99	3.00

Abbildung 35: Ergebnisdatei bei vollständiger Suche

Alle Belegungen werden nach dem „Kilometerzählerprinzip“ durchprobiert. Das heisst, der rechte Parameter (hier P[9]) wird zyklisch in jedem Schritt weitergezählt. Die anderen Parameter werden ebenfalls zyklisch gezählt, jedoch immer nur nach einem Zyklus des jeweils rechten Nachbarn. Bei einer Verbesserung wird der Parametersatz eingetragen.

5.3 Zufälliges Suchen

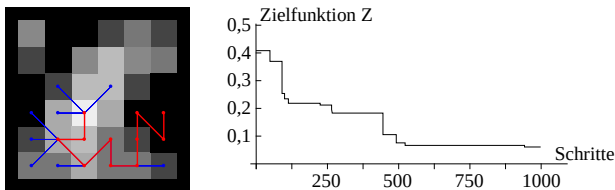
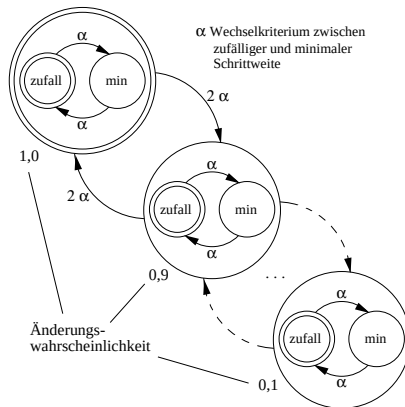


Abbildung 36: Zufälliges Suchen

Die einfachste Art, eine Einstellung zu verbessern, ist der „Random Walk“, also das blinde Umhertapsen im Lösungsraum. Man geht von einem Startpunkt aus einen zufälligen Schritt in eine beliebige Richtung und prüft die neue Einstellung. Verschlechterungen werden verworfen, bei einer Verbesserung wird der neue Punkt als Ausgangspunkt gewählt. Dies wird wiederholt, bis sich eine zufriedenstellende Lösung einstellt oder an einer Stelle keine Verbesserung mehr eintritt. Das Verfahren stellt keinerlei Anforderungen an die Zielfunktion. Eine erfolgreiche Verbesserung tritt jedoch nur ein, wenn in der näheren Umgebung des Punktes weitere gute Einstellungen zu finden sind. Ein lokaler



Verfahrens. α gibt die Anzahl der zulässigen Fehlversuche an. Findet in α keine Verbesserung statt, so wechselt der Optimierer von der Strategie beliebiger Schrittweite (zufall) zur kleinsten Schrittweite (min) und umgekehrt. Versagen beide Strategien, das heißt über 2α Schritte wird keine Verbesserung gefunden, wechselt der Optimierer die Wahrscheinlichkeit, mit der ein Parameter geändert wird. Durch dieses Vorgehen findet abwechselnd globale und lokale Suche statt. Im Zustand der Änderungswahrscheinlichkeit 1,0 und Modus zufall entspricht das Verfahren dem zufälligen Raten. In der Ergebnisdatei dieses

Zeit	Opt	Z	P[0]	P[1]	P[2]	P[3]	P[4]	P[5]	P[6]	P[7]	P[8]	P[9]
09:32	moP	0.108922	0.00	17.00	11.40	0.00	4.00	-13.42	2.00	14.00	-10.62	1.00
09:32	moP	0.108922	-1.00	17.00	11.40	0.00	4.00	-13.42	2.00	14.00	-10.62	1.00
09:32	moP	0.108922	-1.00	18.00	11.39	0.00	4.00	-13.42	2.00	14.00	-10.62	2.00
09:33	moP	0.107092	-1.00	18.00	11.39	1.00	4.00	-13.42	1.00	14.00	-10.62	2.00
09:33	moP	0.107092	-2.00	18.00	11.40	1.00	4.00	-13.43	1.00	14.00	-10.63	2.00
09:34	moP	0.100776	-1.00	18.00	11.41	2.00	3.00	-13.44	1.00	13.00	-10.63	2.00

Abbildung 38: Ergebnisdatei beim „Random Walk“

Verfahrens (Abbildung 38) ist zu erkennen, dass sich benachbarte Zeilen nur wenig unterscheiden.

5.4 Methode des steilsten Abstiegs

Wie beschrieben, ermöglicht blindes Umhertapsen, lokale Tiefpunkte zu finden. Die schnellste Methode ist das jedoch nicht. Ein Bach im Schwarzwald sucht sich den schnellsten Weg ins Tal, indem er immer der Neigungslinie folgt. Der Mathematiker bezeichnet den Neigungsvektor als Gradient. Er steht senkrecht auf der zugehörigen Höhenlinie.

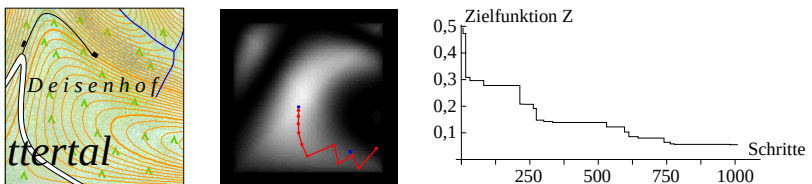


Abbildung 39: Steilster Abstieg

Daher liest man auch „Gradientenabstieg“ oder, auf der Suche nach einem Hochpunkt, „Hill-Climbing“, also Bergsteigen. Unter diesen Begriffen sind eine ganze Reihe verschiedener Optimierungsverfahren zusammengefasst. Alle versuchen, mit möglichst wenig Schritten zu einem Tiefpunkt zu kommen. Zwei Hauptmerkmale charakterisieren die Verfahren: Die Bestimmung der Richtung und der Schrittweite.

Zeit	Z	P[0]	P[1]	P[2]	P[3]	P[4]	P[5]	P[6]	P[7]	P[8]	P[9]
14:03	0.203	-10.00	16.00	15.49	3.00	19.00	-20.20	2.00	7.00	-16.73	3.00
14:03	0.194	-10.00	16.00	15.49	3.00	19.00	-20.20	2.00	8.00	-16.73	3.00
14:03	0.181	-10.00	16.00	15.49	3.00	19.00	-20.20	2.00	9.00	-16.73	3.00
14:03	0.169	-10.00	16.00	15.49	3.00	19.00	-20.20	2.00	10.00	-16.73	3.00
14:03	0.167	-10.00	16.00	15.49	3.00	19.00	-20.20	3.00	10.00	-16.73	3.00
14:03	0.167	-10.00	16.00	15.49	3.00	18.00	-20.20	3.00	10.00	-16.73	3.00

Abbildung 40: Ergebnisdatei bei Gradientenabstieg

5.4.1 Newton-Raphson

Am einfachsten arbeitet das Verfahren, wenn die Zielfunktion mathematisch dargestellt wird und zweimal ableitbar ist. So wird die Richtung des steilsten Abstiegs, also der Gradientenvektor, über die Ableitung bestimmt. Eine recht gute Schrittweite ergibt sich durch Hinzunahme der zweiten Ableitung, also der Krümmung der Kurve. Dieses Verfahren entspricht dem Newtonverfahren zur numerischen Bestimmung von Nullstellen. Es nennt sich daher Newton-Raphson-Methode. Sind die weiteren Ableitungen einer Funktion Null, man spricht dann von einer quadratischen Funktion, so findet man den Tiefpunkt in einem Schritt. Da die Zielfunktionen der meisten technischen Anwendungen jedoch nicht quadratisch sind, sondern ihre Potenzreihe unendlich ist, tritt dieser Fall selten auf. Das Verfahren konvergiert dann immerhin noch quadratisch. Für die hier zu bestimmende Zielfunktion, die weder als mathematische Formel darstellbar, noch ableitbar ist, kann diese Methode nicht eingesetzt werden.

5.4.2 Hooke-Jeeves

Die Hooke-Jeeves-Methode arbeitet nach zwei Grundsätzen, Tasten und Voranschreiten. Das Tasten ersetzt die Ableitung. Es wird in jeder Richtung ein kleiner Tastschritt unternommen, dessen Verbesserung oder Verschlechterung als partielle Ableitung dient. Beim Tasten wird also jeweils nur ein Parameter geändert. Anschließend erfolgt das Voranschreiten, wie beim Newton-Raphson-Verfahren. Vorteil dieses Verfahrens ist, dass die Ableitung nicht bekannt sein muss. Nachteil ist das aufwendige Tasten in alle Richtungen. Bei 100 Variablen kann die ungerichtete Suche in derselben Zeit bereits 100 Schritte durchführen.

5.5 Genetische Optimierung

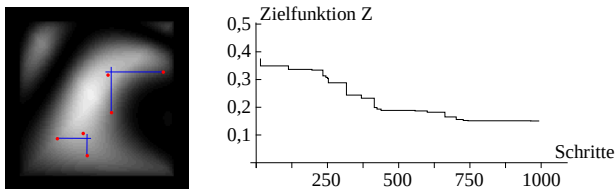


Abbildung 41: Genetische Optimierung

Die genetische Optimierung ahmt die biologische Evolution nach. Sie wurde Ende der siebziger Jahre entwickelt [25]. Die Idee ist folgende: Die Entwicklung der Lebewesen erfolgte durch Kreuzen und Verändern (mutieren) des Erbguts. Missgeburten und Lebewesen, die nicht an den Lebensraum angepasst sind, sterben aus, ohne sich fortzupflanzen. Die stärksten Exemplare zeugen auch die meisten Nachkommen. Dieser Vorgang gewährleistet, dass jede Generation mindestens so gut wie die vorhergehende ist.

In der Optimierung entspricht jedes Exemplar einem Parametersatz. Die Güte der Anpassung an den Lebensraum ist durch die Zielfunktion gegeben. Zwei Parametersätze werden gekreuzt, indem man zufällig oder nach bestimmten Mustern jeden Parameter mit dem Wert eines der beiden Sätze belegt. Abbil-

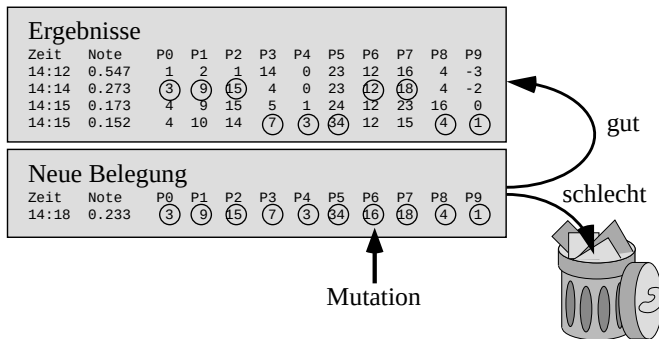


Abbildung 42: Genetische Optimierung

Abbildung 42 verdeutlicht das Verfahren. Mutiert wird durch zufälliges Ändern einer oder mehrerer Parameter. Die Wahrscheinlichkeit einer Änderung und die Stärke der Änderung sind wichtige Stellwerte des Verfahrens. Für diese Anwendung wurde mit verschiedenen Mutationswahrscheinlichkeiten, Schrittweiten und asymmetrischer Kreuzung experimentiert. Die Wahrscheinlichkeit einer Mutation von 10% und einer Varianz der minimalen Schrittweite zeigten die besten Ergebnisse. Eine asymmetrische Kreuzung, also einer Wahrscheinlichkeit ungleich 50% für die Übernahme des Vater- oder Mutter-Gens, brachte keine erkennbare Verbesserung des Verfahrens.

Zeit	Z	P[0]	P[1]	P[2]	P[3]	P[4]	P[5]	P[6]	P[7]	P[8]	P[9]
13:42	0.164	-8.00	9.00	0.00	0.00	1.00	-19.65	3.00	6.00	-14.37	1.00
13:43	0.164	-1.00	9.00	23.91	1.00	1.00	-19.65	3.00	6.00	-21.89	1.00
13:43	0.164	-8.00	9.00	0.00	0.00	1.00	-19.65	3.00	6.00	-14.37	1.00
13:43	0.164	-9.00	1.00	0.00	0.00	1.00	-19.65	3.00	7.00	-19.71	1.00
13:43	0.132	-1.00	9.00	0.00	0.00	1.00	-18.42	3.00	14.00	-14.37	1.00
13:44	0.129	-1.00	9.00	0.00	0.00	1.00	-18.23	3.00	14.00	-14.37	1.00

Abbildung 43: Ergebnisdatei bei genetischer Optimierung

In Abbildung 43 ist zu erkennen, dass Parameter (hier -14.37), die in einer Belegung nicht auftreten, in einer späteren Generation wieder erscheinen.

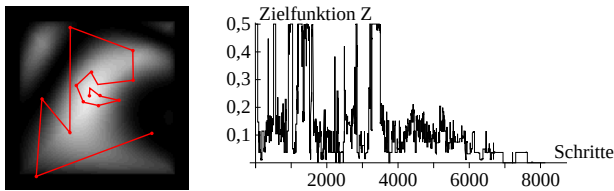


Abbildung 44: Simuliertes Auskühlen

5.6 Simuliertes Auskühlen

Simuliertes Auskühlen (Simulated Annealing) lässt sich vergleichen mit einem Verfahren der Werkstoffphysik. Beim Erstarren eines Metalls, das aus der Schmelze genommen wird, also dem Übergang vom flüssigen in den festen Zustand, bilden die Atome Kristalle. Beim vollkommenen Kristall sind die Atome mit geringster Energie angeordnet. Kühlt ein Metall zu schnell ab, so erreichen die Kristalle nicht die energieärmste Konfiguration. Es bilden sich viele Kristallisationskerne, von denen die Aushärtung ausgeht. An den Grenzen der verschiedenen Kristallisationsbereiche bilden sich Unregelmäßigkeiten.

Kühlt das Metall bei gleichzeitiger Wärmezufuhr schrittweise langsam ab, so erhalten die Kristalle genügend Bewegungsenergie, um eine schlechte Position zu verlassen und eine günstigere Stelle einzunehmen.

Dieses Verfahren lässt sich übertragen, um Tief- oder Hochpunkte von Funktionen in mehreren Variablen zu finden. Seit den 80er Jahren befassen sich einige Wissenschaftler mit diesen Verfahren [22].

Ausgehend von der Zielfunktion und einer zufällig gewählten Parameterbelegung untersucht man die benachbarten Belegungen. Ob diese Belegung akzeptiert wird, hängt von der Güte und einem Kontrollparameter ab. Der Kontrollparameter ist ein abnehmender Wert zwischen 0 und 1, der die Wahrscheinlichkeit für die Annahme einer schlechteren Belegung angibt.

So werden am Anfang des Verfahren noch fast wahllos die neuen Konfigurationen akzeptiert, da die Wahrscheinlichkeit, mit der eine schlechtere Variante akzeptiert wird, noch sehr hoch ist. Mit abnehmender Wahrscheinlichkeit nimmt dann auch die Bereitschaft ab, eine schlechtere Konfiguration zu ak-

zeptieren. Geht diese Wahrscheinlichkeit gegen 0, so werden nur noch kleinere Variationen zugelassen, analog zu der Metallschmelze, die immer zähflüssiger wird und sich die Sprünge zu ungünstigeren energetischen Zuständen immer mehr reduzieren. Je geringer die Wahrscheinlichkeit ist, dass ein schlechterer Zustand akzeptiert wird, desto starrer wird die Suche.

Damit das Verfahren immer wieder einen Ausweg aus lokalen Minima findet, muss die Abkühlung, das heißt das Sinken der Wahrscheinlichkeit, entsprechend langsam stattfinden. Setzt man die Wahrscheinlichkeit von Anfang an auf 0, entspricht das Verfahren dem „Random Walk“.

Der Kontrollparameter wird durch das sogenannte Metropolis-Kriterium festgelegt. Ist Z die Zielfunktion und M die Temperaturstufe, so ist die Wahrscheinlichkeit für die Annahme einer schlechteren Belegung:

$$Prob = e^{\frac{Z(P_{neu}) - Z(P_{alt})}{M}}$$

Bei dieser Anwendung beginnt der Kontrollparameter M bei 0,1 und sinkt mit jeder Abkühlung um 0,001. Der Optimierungsvorgang endet bei 0,001. Wie lange das System auf einer Temperaturstufe verweilt, hängt von der Anzahl der Parameter und der Art ab. Es zeigte sich, dass bereits nach 10000 Schritten ein gutes Ergebnis erzielt wird.

5.7 Zusätzliche Eingriffe

Einige Verfahren lassen sich verbessern, wenn während der Optimierung die Ergebnisdatei von Hand oder automatisch bereinigt wird. Das hängt mit der Kombination der Verfahren zusammen. Findet an einem Parametersatz eine lokale Optimierung statt, so werden schnell einige ähnliche bzw. identische Parametersätze erzeugt. Dies führt dazu, dass bei der genetischen Optimierung mit hoher Wahrscheinlichkeit eng verwandte Generationen miteinander gekreuzt werden, also gewissermaßen Inzucht. Dies lässt sich beheben, indem man von ähnlichen Parametersätzen jeweils nur den Besten behält und die schlechteren löscht. Ein einfaches Skript, das nach vorgegebener Ähnlichkeits-Metrik die Daten vergleicht, liefert gute Ergebnisse. Als Ähnlichkeits-Metrik wurde der euklidische Abstand mit der Schwelle S verwendet.

$$\sum_{i=0}^{\#P} (P[i]_1 - P[i]_2)^2 \geq S$$

Ein weiterer Vorteil dieser Eingriffe ist die Übersichtlichkeit der Ergebnisd-
tei.

5.8 Parallele Optimierung

Für diese Arbeit wurden die genannten Verfahren implementiert. Alle, außer der simulierten Auskühlung, eignen sich für eine parallele Ausführung. Hier wurde die Parallelität durch vernetzte Rechner verwirklicht. Das Optimierungsprogramm wurde in C geschrieben. Es kann auf verschiedenen Rechnern und Betriebssystemen kompiliert werden. Für diese Anwendung standen Linux, UNIX, Windows und BeOS zur Verfügung.

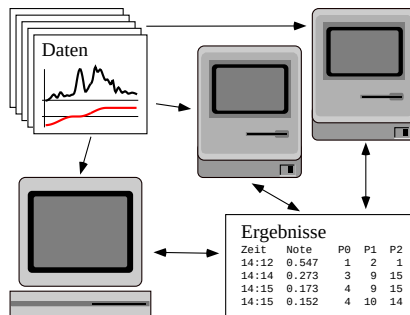


Abbildung 45: Versuchsaufbau

Die verschiedenen Rechner können auf eigenen Dateien und über NFS (Network File System) auf gemeinsamen Dateien arbeiten. Abbildung 45 zeigt den Versuchsaufbau. Die Beschleunigungsdaten der Kollisionsversuche und die zugehörigen Bewertungsfunktionen werden gemeinsam genutzt. Das ausführbare Optimierungsprogramm befindet sich jeweils auf dem lokalen Rechner. Die Optimierungsergebnisse werden in einer gemeinsamen Datei abgelegt. Jeder „Rechenknecht“ kann während der Optimierung die gemeinsamen Ergebnisse lesen. Hängt er in einem lokalen Minimum fest, kann er ein Ergebnis eines anderen Prozesses nutzen. Er liest dazu die Ergebnisdatei, Abbildung 46. Die Spalte „Opt“ in der Ergebnisdatei gibt an, welches Optimierungsverfah-

ren den jeweiligen Parametersatz gefunden hat. G: Genetische Optimierung, P: Probabilistischer Random Walk, R: Zufälliges Raten, D: Gradientenabstieg.

Zeit	Opt	Z	P[0]	P[1]	P[2]	P[3]	P[4]	P[5]	P[6]	P[7]	P[8]	P[9]
10:58	P	0.500	-10.00	1.00	0.00	0.00	1.00	-30.00	0.00	1.00	-30.00	0.00
10:58	P	0.548	-8.00	17.00	21.61	1.00	7.00	-10.48	3.00	2.00	-9.64	1.00
10:58	R	0.445	-4.00	1.00	23.59	3.00	4.00	-13.97	2.00	14.00	-16.56	3.00
10:58	G	0.105	-5.00	1.00	29.11	1.00	5.00	-10.34	1.00	16.00	-14.04	2.00
10:58	R	0.470	-8.00	9.00	5.26	2.00	15.00	-9.65	1.00	19.00	-19.15	1.00
10:58	D	0.277	-1.00	17.00	1.01	2.00	3.00	-21.67	3.00	10.00	-19.61	1.00
10:58	G	0.105	-5.00	1.00	29.11	1.00	5.00	-10.34	1.00	16.00	-14.04	2.00
10:58	G	0.105	-5.00	1.00	29.11	1.00	5.00	-10.34	1.00	16.00	-14.04	2.00
10:58	G	0.105	-5.00	1.00	29.11	1.00	5.00	-10.34	1.00	16.00	-14.04	2.00
10:58	G	0.105	-5.00	1.00	29.11	1.00	5.00	-10.34	1.00	16.00	-14.04	2.00
10:59	G	0.105	-5.00	1.00	29.11	1.00	5.00	-10.34	1.00	16.00	-14.04	2.00
10:59	G	0.105	-5.00	1.00	29.11	1.00	5.00	-10.34	2.00	16.00	-14.04	2.00

Abbildung 46: Ergebnisdatei bei paralleler Optimierung

Während eines Schreibzugriffs wird die Ergebnisdatei gesperrt (Semaphore-Konzept). Werden in schneller Folge von verschiedenen Rechnern Verbesserungen erzielt, kann es geschehen, dass sie sich gegenseitig überschreiben. Dadurch geht ab und zu ein Parametersatz verloren. Mit zunehmender Güte verringert sich die Frequenz der Verbesserungen jedoch deutlich. Werden neue Ergebnisse nur noch im Sekunden- oder sogar Minutenrhythmus gefunden, ist die Wahrscheinlichkeit des gegenseitigen Überschreibens gering. Die zeitliche Leistungseinbuße liegt unter einem Prozent. Ein anderes Kommunikationskonzept wäre aufwendiger und langsamer.

Ein einzelner Aufprall besteht aus etwa 100 Beschleunigungswerten im Millisekundenrhythmus. Jeder wird in einem 4-Byte-Datentyp (Float) gespeichert. Ein Datensatz benötigt daher bei sauberer Programmierung weniger als ein halbes Kilobyte. Ebensoviel benötigt die Bewertungsfunktion. Ein Datensatz von 1000 Ereignissen findet daher in weniger als einem Megabyte Hauptspeicher Platz. Derzeitige Rechner verfügen über ein Vielfaches davon. Daher waren die Optimierungsverfahren sehr schnell. Die schnellste Maschine, ein PowerPC 601e mit 350 MHz prüft eine Parameterbelegung an 1000 Crashes in einer Zehntel Sekunde. Lässt man die Optimierungsprogramme auf zehn Rechnern einen Tag laufen, so werden über 8 Millionen Belegungen geprüft, wobei meist sehr gute Bewertungen erzielt wurden. Man kann daher davon ausgehen, dass die Optimierung hinreichende Ergebnisse erbrachte.

Für eine leistungsstarke Optimierung ist es sinnvoll, möglichst viele Rechner parallel arbeiten zu lassen. Die Optimierungsverfahren sollten anfänglich zufällige Belegungen prüfen und eine vollständige Suche im groben Raster durchzuführen. Sind eine Reihe von Startwerten erzeugt, erweisen sich nur noch die genetische Optimierung, der Gradientenabstieg und die zufällige Suche als sinnvoll.