

**HAUSARBEIT**  
Benedikt Löhn

# **Anwendung von Partikel- schwarmoptimierung auf ein dynamisches Problem**

**FAKULTÄT TECHNIK UND INFORMATIK**  
Department Informatik

Faculty of Computer Science and Engineering  
Department Computer Science

Benedikt Löhn

# Anwendung von Partikelschwarmoptimierung auf ein dynamisches Problem

Hausarbeit eingereicht im Rahmen der Prüfungsleistung des Moduls Künstliche Intelligenz

im Studiengang *Bachelor of Science Angewandte Informatik*  
am Department Informatik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Peer Stelldinger  
Eingereicht am: 02. August 2024

**Benedikt Löhn**

**Thema der Arbeit**

Anwendung von Partikelschwarmoptimierung auf ein dynamisches Problem

**Stichworte**

Anwendung, Partikelschwarmoptimierung, dynamisch

**Kurzzusammenfassung**

Diese Arbeit beschäftigt sich mit der Anwendung eines Partikelschwarmoptimierungsalgorithmus auf ein dynamisches Problem. Hierfür wird eine zweidimensionale Landschaft generiert, auf welcher ein Partikelschwarm einen alleinigen Partikel verfolgt und versucht, einzuholen. Dabei wird die Zeit, die ein Partikel des Schwarms braucht, um sein Ziel zu erreichen, gemessen. Zusätzlich wird die Simulation mit verschiedenen Parametern durchgeführt, um zu untersuchen, wie sich das Verhalten der einzelnen Partikel ändert. Abschließend werden die Ergebnisse ausgewertet und zusammengefasst sowie eine Perspektive für die Zukunft gegeben.

# Inhaltsverzeichnis

<b>Abkürzungsverzeichnis.....</b>	<b>vi</b>
<b>1 Einleitung.....</b>	<b>1</b>
1.1 Problemstellung .....	1
1.2 Ziel der Arbeit.....	1
1.3 Aufbau der Arbeit.....	2
<b>2 Grundlagen.....</b>	<b>2</b>
2.1 Partikelschwarmoptimierung (PSO).....	2
2.1.1 Definition und Funktionsweise .....	2
2.1.2 Mathematisches Modell .....	3
2.2 Dynamische Ziele .....	3
2.2.1 Definition und Analogie.....	3
2.2.2 Herausforderungen bei der Optimierung.....	3
<b>3 Verwandte Arbeiten.....</b>	<b>4</b>
3.1 Überblick über bestehende Forschung.....	4
3.2 Relevante Studien .....	4
3.3 Vergleich verschiedener Ansätze.....	5
<b>4 Methodik.....</b>	<b>5</b>
4.1 Simulationsumgebung und Szenarien.....	5
4.1.1 Beschreibung der Testumgebung .....	5
4.1.2 Beschreibung des dynamischen Ziels.....	6
4.2 Anpassung der PSO an dynamische Ziele .....	7
4.2.1 Strategien zur Anpassung der Schwarmparameter.....	7
4.2.2 Mechanismen zur Erkennung und Reaktion auf Zieländerungen .....	7
<b>5 Experimente und Ergebnisse .....</b>	<b>7</b>

5.1	Durchführung der Experimente .....	7
5.1.1	Beschreibung der durchgeführten Tests .....	7
5.1.2	Parameter und Einstellungen .....	8
5.2	Ergebnisse .....	9
5.2.1	Darstellung der Resultate .....	9
5.2.2	Analyse der Ergebnisse .....	9
<b>6</b>	<b>Diskussion .....</b>	<b>10</b>
6.1	Interpretation der Ergebnisse .....	10
6.2	Stärken und Schwächen der angepassten PSO .....	10
<b>7</b>	<b>Schlussfolgerungen und Ausblick .....</b>	<b>11</b>
7.1	Zusammenfassung der wichtigsten Erkenntnisse .....	11
7.2	Mögliche zukünftige Forschungsrichtungen .....	11
	<b>Literaturverzeichnis .....</b>	<b>13</b>
<b>A</b>	<b>Anhang 1 .....</b>	<b>14</b>
A.1	GitHub-Repository .....	14

## Abkürzungsverzeichnis

<b>PSO</b>	Partikelschwarmoptimierung
<b>SPSO</b>	speciation PSO
<b>MPSO</b>	multi-swarm PSO

# 1 Einleitung

## 1.1 Problemstellung

Ein Großteil der praktischen Optimierungsprobleme der heutigen Welt sind dynamisch: die optimale Lösung ändert sich mit der Zeit. Dies stellt traditionelle Optimierungsmethoden vor große Herausforderungen, da sie meist für statische Ziele konzipiert sind. In der Vergangenheit wurden die besten Erfolge zur Bewältigung solcher Probleme mithilfe von modifizierten, auf das Problem angepassten evolutionären Algorithmen erzielt (Blackwell, et al., 2008, p. 193).

Partikelschwarmoptimierung (PSO) ähnelt den evolutionären Algorithmen in vielen Aspekten, liefert aber in ihrer ursprünglichen Form nur für statische Probleme verlässliche und gute Lösungen (Blackwell, et al., 2008, p. 193).

Um also für dynamische Probleme gute Lösungen mithilfe von PSO zu finden, muss das Verfahren, wie auch bei den evolutionären Algorithmen, modifiziert und an das Problem angepasst werden.

## 1.2 Ziel der Arbeit

Das Ziel dieser Arbeit ist es, die Anwendung der PSO auf dynamische Ziele zu untersuchen sowie Strategien zur Anpassung des Algorithmus an sich ändernde Bedingungen zu entwickeln und zu evaluieren.

### 1.3 Aufbau der Arbeit

Zunächst werden in Kapitel 2 die Grundlagen der Partikelschwarmoptimierung vorgestellt, indem die Definition und Funktionsweise des Algorithmus sowie das zugrundeliegende mathematische Modell erläutert werden. Anschließend wird in Kapitel 3 ein Überblick über bestehende Arbeiten zu verwandten Themen geschaffen. In Kapitel 4 werden die verwendete Methodik sowie die Simulationsumgebung und -szenarien beschrieben. Anschließend werden in Kapitel 5 verschiedene durchgeführte Experimente beschrieben und die Ergebnisse dokumentiert, bevor diese in Kapitel 6 interpretiert und diskutiert werden. Abschließend wird in Kapitel 7 eine Schlussfolgerung gezogen und ein Ausblick in die Zukunft gewährt.

## 2 Grundlagen

### 2.1 Partikelschwarmoptimierung (PSO)

#### 2.1.1 Definition und Funktionsweise

(Kennedy & Eberhart, 1995) PSO ist ein Algorithmus, der unter die naturanalogen Optimierungsverfahren fällt und auf Schwarmverhalten basiert, das in der Natur auftritt, wie etwa das von Vögeln oder Fischen. Zu Beginn wird die Population mit zufälligen Werten bzw. Lösungen initialisiert. Zusätzlich wird jeder Lösung der Population (im Nachfolgenden Partikel genannt) ein zufälliger Bewegungsvektor zugewiesen, bevor die Partikel durch die Simulationsumgebung geschickt werden. Jeder Partikel merkt sich hierbei seine eigene bisherige Position mit der besten Lösung  $pBest$  sowie die globale beste Position  $gBest$ , also die beste Position des besten Partikels.



Nach jeder Iteration, also jedem Bewegungsschritt der Partikel, wird der Bewegungsvektor jedes Partikels in Richtung von  $pBest$  und  $gBest$  angepasst.

### 2.1.2 Mathematisches Modell

Die Simulationsumgebung, auf die sich diese Arbeit bezieht (nachfolgend Terrain genannt), ist ein zweidimensionaler Raum mit einer Größe von  $1000 \times 1000$  Pixeln. Alle Partikel starten unabhängig voneinander auf einer zufälligen Position innerhalb des Terrains und mit einem Bewegungsvektor mit zufällig ermittelten x- und y-Werten zwischen -1 und 1.

Der Einfluss von  $pBest$  und  $gBest$  wird jeweils mit einem Zufallswert gewichtet (Kennedy & Eberhart, 1995, p. 40), um die Individualität jedes Partikels zu gewährleisten.

Um realitätsnäher an der Analogie (siehe 2.2.1) zu sein, werden zusätzlich eine bestimmte Anzahl an Hindernissen in Form von Vektoren, mit zufälliger Position und Länge, auf dem Terrain gesetzt, die von den Partikeln nicht durchquert werden können.

## 2.2 Dynamische Ziele

### 2.2.1 Definition und Analogie

Ein dynamisches Problem hat keine feste Lösung; sie ändert sich mit der Zeit. Ein Algorithmus, der dieses Problem löst, muss also die sich ändernden Bedingungen beobachten und das Optimum stets neu berechnen (Blackwell, et al., 2008, p. 193).

Die Analogie, die hierfür in dieser Arbeit verwendet wird, ist die Hetzjagd eines Wolfrudels auf eine Beute, die vor den Wölfen flieht.

### 2.2.2 Herausforderungen bei der Optimierung

In dynamischen Umgebungen tritt das Problem der Änderungserkennung auf, d.h. die Partikel müssen eine Änderung am Optimum erfassen (Blackwell, et al., 2008, p. 195). Da die Beute stets auf der Flucht ist und nie stoppt (bis sie gefangen wird), kann dieses Problem umgangen werden, indem das momentane  $gBest$  nach jeder Bewegung jedes Partikels aktualisiert wird.

Des Weiteren ist das *pBest* eines Wolfs nach der nächsten Bewegung nicht mehr optimal, da sich in Richtung der Beute bewegt wird. Aufgrund dessen wird das *pBest* nach jeder Bewegung auf die neue Position des Wolfs und *gBest* auf das Maximum aller individuellen *pBest* gesetzt (Blackwell, et al., 2008, p. 196).

## 3 Verwandte Arbeiten

### 3.1 Überblick über bestehende Forschung

PSO in seiner ursprünglichen Form wurde erstmals von James Kennedy und Russell Eberhart vorgestellt (Kennedy & Eberhart, 1995). Modernere Ansätze, wie etwa die Anwendung auf sich ständig ändernde Umgebungen, wurden von Konstantinos Parsopoulos und Michael Vrahatis untersucht (Parsopoulos & Vrahatis, 2002). Diese und einige weitere Forschungsarbeiten, wie etwa die von S. Janson und M. Middendorf aus dem Jahr 2004, haben Tim Blackwell, Jürgen Branke und Xiaodong Li zusammengefasst und verglichen (Blackwell, et al., 2008).

### 3.2 Relevante Studien

Ein Ansatz, der verfolgt wurde, ist die Repulsion von Partikeln untereinander, um Diversität in den Positionen des Schwarms zu garantieren. Dies wurde in Form der Analogie eines Atoms untersucht: (Blackwell, et al., 2008, pp. 197-198) Der Schwarm besteht aus geladenen und neutralen Partikeln, wobei die geladenen Partikel sich untereinander abstoßen und somit eine Wolke um sich nicht abstoßende, neutrale Partikel bilden. Dieses Modell führt zu erhöhter Diversität des konvergierenden Schwarms, was ermöglicht, eine Verschiebung des Optimums besser zu erfassen.

(Blackwell, et al., 2008, p. 198) Die Idee, den Schwarm in mehrere kleinere Schwärme zu unterteilen, um in kürzerer Zeit ein größeres Gebiet innerhalb des Terrains zu untersuchen, kann in Form von „speciation PSO“ (SPSO) oder „multi-swarm PSO“ (MPSO) realisiert werden. Ersteres basiert auf einem Schwarm fester Größe, welcher sich dynamisch in Unterschwärme aufteilt, während MPSO mit mehreren Schwärmen festgelegter Größe beginnt und die Anzahl der Schwärme (dementsprechend auch die Gesamtpopulation des Terrains) dynamisch anpasst.

### **3.3 Vergleich verschiedener Ansätze**

Die beiden in 3.2 vorgestellten Erweiterungen von PSO schneiden in sich langsam verändernden Umgebungen vergleichbar gut ab. In stark dynamischen Terrains kommt MPSO besser mit vielen Änderungen des Optimums zurecht, während SPSO in geringerer Anzahl auftretende, aber dafür stärkere Änderungen besser handhabt (Blackwell, et al., 2008, p. 215).

## **4 Methodik**

### **4.1 Simulationsumgebung und Szenarien**

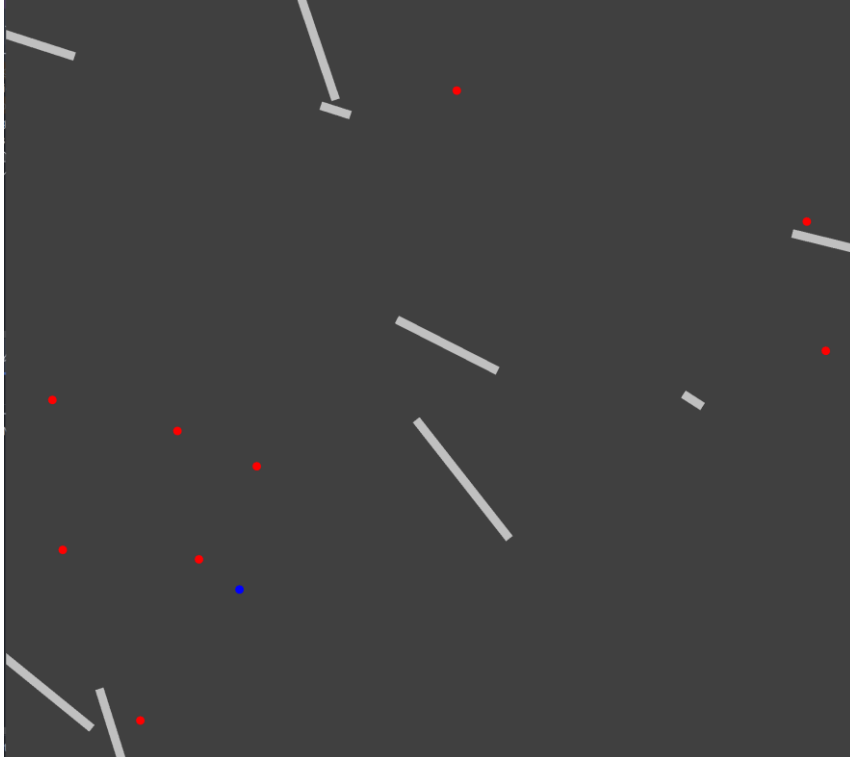
#### **4.1.1 Beschreibung der Testumgebung**

Zunächst wird das Terrain mit einer Größe von 1000\*1000 Pixeln initialisiert. Daraufhin werden 10 für die Partikel unpassierbare Hindernisse auf dem Terrain erzeugt. Die Generierung der Hindernisse erfolgt willkürlich: Startposition des Vektors, Winkel und Länge werden allesamt zufällig generiert.

Nun werden das dynamische Ziel in Form einer Beute sowie 10 Wölfe, welche den Partikelschwarm bilden, auf jeweils zufälligen Startpositionen innerhalb der Terrainbegrenzungen

initialisiert. Das Ermitteln des anfänglichen Bewegungsvektors geschieht ebenfalls nach dem Zufallsprinzip.

Abbildung 1: Initialisierte Simulationsumgebung



Ein beispielhaftes Terrain ist in Abbildung 1 zu sehen, wobei die roten Punkte die Wölfe und somit den Partikelschwarm bilden. Der blaue Punkt visualisiert die Beute.

#### **4.1.2 Beschreibung des dynamischen Ziels**

Das dynamische Ziel in Form der Beute wird von den Wölfen gejagt und möchte dementsprechend vor den Wölfen fliehen. Dies wird durch eine repulsive Kraft realisiert, die die Beute

von dem zum jeweiligen Zeitpunkt nächsten Wolf abstößt. Da es nur eine Beute gibt, spielen lokales und globales Maximum hier für die Wegfindung keine Rolle.

## 4.2 Anpassung der PSO an dynamische Ziele

### 4.2.1 Strategien zur Anpassung der Schwarmparameter

Da das lokale Maximum  $pBest$  eines Wolfs nach einer Iteration nicht mehr optimal ist, entfällt dementsprechend die Gewichtung des lokalen Maximums bei der Anpassung des Richtungsvektors. Die eigene momentane Richtung sowie das globale Maximum  $gBest$  spielen allerdings eine Rolle und werden in jeder Bewegung zufällig stark gewichtet, um individuelles Verhalten in jeder Iteration zu gewährleisten.

### 4.2.2 Mechanismen zur Erkennung und Reaktion auf Zieländerungen

Da Wölfe in der realen Welt eine Bewegungsänderung ihrer Beute sofort wahrnehmen, muss sichergestellt werden, dass die Beute stets unter Beobachtung der Wölfe steht. Dies geschieht über eine `Instanzvariable` der Wölfe, in welcher die Beute gespeichert wird, sodass die Position der Beute allzeit bekannt ist. Mithilfe dieser Instanzvariable wird nun der Bewegungsvektor der Wölfe zusätzlich mit der aktuellen Position der Beute sowie einem Zufallswert gewichtet.

**Kommentiert [AS1]:** Solltest du erläutern, was die genau macht, oder kann das als bekannt vorausgesetzt werden?

# 5 Experimente und Ergebnisse

## 5.1 Durchführung der Experimente

### 5.1.1 Beschreibung der durchgeführten Tests

Der Bewegungsvektor jedes Wolfs wird mit drei Faktoren gewichtet: Der eigenen momentanen Richtung  $wWolf$ , dem `globalen` Maximum  $pG$  und der Position der Beute  $pPrey$ . Die Beute hat

**Kommentiert [AS2]:** globalen

## Experimente und Ergebnisse

---

nur zwei Gewichtungen: ebenfalls die eigene momentane Richtung  $w_{Prey}$  und die Position des nächsten Wolfs  $p_{Wolf}$ . In den durchgeführten Tests wurden diese Parameter kleinschrittig verändert und die Simulation mit den geänderten Parametern jeweils 20-mal durchgeführt. Hierbei wurde die Zeit, die der Schwarm benötigt, um die Beute zu erreichen, in Sekunden gemessen und mit zwei Nachkommastellen ausgegeben. Abschließend wurde aus den 20 Simulationen der Mittelwert für die Erreichung des Ziels berechnet.

### 5.1.2 Parameter und Einstellungen

Testfall 1

wWolf	pG	pPrey	wPrey	pWolf
1.5	0.3	1.25	3	-1.5

Testfall 2

wWolf	pG	pPrey	wPrey	pWolf
1.5	0.3	1.25	2.5	-1.75

Testfall 3

wWolf	pG	pPrey	wPrey	pWolf
1.2	0.5	1.1	3	-1.5

Testfall 4

wWolf	pG	pPrey	wPrey	pWolf
1.0	0.5	1.5	3	-1.5

Testfall 5

wWolf	pG	pPrey	wPrey	pWolf
1.0	0.5	1.5	3	-2.25

## 5.2 Ergebnisse

### 5.2.1 Darstellung der Resultate

Auswertung der Testergebnisse

Testfall 1	Testfall 2	Testfall 3	Testfall 4	Testfall 5
5,46s	5,00s	6,00s	5,47s	5,52s

### 5.2.2 Analyse der Ergebnisse

Testfall 2 liefert das beste Ergebnis, da der Algorithmus hier im Schnitt die kürzeste Zeit benötigt, um ans Ziel zu kommen. Das schlechteste Ergebnis liefert Testfall 3, welcher eine Sekunde länger braucht. Die anderen drei Fälle liegen von der Laufzeit her inmitten des besten und schlechtesten Falls. Hierbei ähneln sich die Laufzeiten so sehr, dass Testfall 1, 4 und 5 als gleichwertig angesehen werden können.

## 6 Diskussion

### 6.1 Interpretation der Ergebnisse

In Testfall 2, welcher das beste Ergebnis erzielt hat, wurde  $wPrey$  schwächer,  $pWolf$  dafür aber stärker gewichtet. Das führt dazu, dass die Beute ihre momentane Bewegungsrichtung weniger wahrscheinlich beibehält und im Gegenzug den Fokus auf den nächstgelegenen Wolf legt. Dadurch steigt die Wahrscheinlichkeit, in Richtung eines anderen Wolfs zu laufen und somit schneller gefangen zu werden, was zu einer kürzeren Laufzeit führt.

Der 3. Testfall ist darauf ausgelegt, dass die Wölfe ihre eigene Bewegungsrichtung sowie die Position der Beute etwas mehr vernachlässigen und sich stattdessen stärker an dem globalen Maximum  $pG$  orientieren. Infolgedessen nimmt das Wolfsrudel oft eine schlangenartige Form an, wodurch ein kleinerer Bereich des Terrains abgedeckt wird, was der Beute eine größere Fläche zur Flucht bietet. Dementsprechend dauert es hier länger, bis das Ziel erreicht wird.

Konträr zu Testfall 1, wo  $wWolf$  am meisten Einfluss nimmt, wird in Testfall 4 stattdessen  $pPrey$  stärker gewichtet. Beide Fälle kommen zu einem nahezu identischen Ergebnis.

Am interessantesten ist der Unterschied (bzw. Nicht-Unterschied) zwischen Testfall 4 und 5. Vier der fünf Parameter sind identisch, lediglich  $pWolf$  wird im 5. Testfall deutlich stärker gewichtet. Dennoch wird annähernd das gleiche Ergebnis erzielt.

### 6.2 Stärken und Schwächen der angepassten PSO

Die angepasste PSO bietet viele Vorteile, wenn es um das Lösen dynamischer Probleme geht. Änderungen am Optimum werden sofort erkannt und nehmen unverzüglich Einfluss auf das Schwarmverhalten. Ebenso kann mit den Eingabeparametern experimentiert werden, um möglichst optimale Werte zu finden. Des Weiteren ist der Speicherbedarf geringer als bei herkömmlicher PSO, da das lokale Maximum der Partikel vernachlässigt wird und somit nicht gespeichert werden muss.



Andererseits beinhaltet der Algorithmus auch einige Zufallselemente, die eine statistische Auswertung der Ergebnisse eventuell verzerren. Zusätzlich kann so nie bewusst das exakt gleiche Verhalten simuliert werden.

## 7 Schlussfolgerungen und Ausblick

### 7.1 Zusammenfassung der wichtigsten Erkenntnisse

Zusammenfassend lässt sich sagen, dass eine auf das gegebene Problem spezifizierte PSO eine angemessene Methodik ist, um ein dynamisches Optimum zu finden. PSO bietet die Möglichkeit einer effizienten und performanten Implementierung, um Speicherbedarf und Laufzeit möglichst gering zu halten.

Zur selben Zeit ist die in dieser Arbeit verwendete Methodik bei weitem nicht optimal. PSO-Varianten, wie die in Kapitel 3.2 beschriebenen MPSO und SPSO, sind zwar deutlich komplexer, gleichzeitig aber auch zuverlässiger.

Das Lösen von dynamischen Optimierungsproblemen durch PSO ist also definitiv ein interessantes Forschungsfeld, welches in der Zukunft aufregende und spannende wissenschaftliche Arbeiten verspricht.

### 7.2 Mögliche zukünftige Forschungsrichtungen

Eine mögliche Forschungsrichtung, in der dynamische PSO eine Rolle spielen könnte, ist die Wegplanung autonomer Fahrzeuge. Ein selbstfahrendes Auto muss stets auf Änderungen im Straßenverkehr reagieren und diese auch frühzeitig antizipieren können, um nicht nur Unfällen vorzubeugen, sondern ebenfalls eine möglichst effiziente Routenplanung vorzunehmen, die Reisedauer und -kosten minimiert. Hierfür könnte PSO in der Zukunft eine ausschlaggebende Rolle spielen.

### *Schlussfolgerungen und Ausblick*

---

Weitergehend könnte PSO für die Finanzmarktanalyse spezifiziert werden. Marktbedingungen schwanken stets und können sich innerhalb kurzer Zeitspannen drastisch verändern. Mit einer hierfür angepassten PSO könnte eventuell eine zuverlässige Vorhersage über zukünftige Marktbedingungen getroffen werden und somit eine möglichst zeitnahe Anpassung an die nachfolgenden Bedingungen erfolgen.

## Literaturverzeichnis

Blackwell, Branke & Li, 2008. *Swarm Intelligence (pp 193-217)*. s.l.:s.n.

Clerc, 2006. *Particle Swarm Optimization*. s.l.:s.n.

Kennedy & Eberhart, 1995. *A new optimizer using particle swarm theory*. s.l.:s.n.

Parsopoulos & Vrahatis, 2002. *Recent approaches to global optimization problems through particle swarm optimization. Natural Computing pp. 235-306*. s.l.:s.n.

# A Anhang 1

## A.1 GitHub-Repository

[https://github.com/BenediktLoehn/KI\\_Hausarbeit](https://github.com/BenediktLoehn/KI_Hausarbeit)

### Erklärung zur selbstständigen Bearbeitung einer Hausarbeit

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

_____	_____	_____
Ort	Datum	Unterschrift im Original