*„Optimaler Sitzplan im Klassenraum auf Basis   
evolutionärer Algorithmen“*

**Studienarbeit II**

für die Prüfung zum

Bachelor of Science

der Angewandten Informatik

an der Dualen Hochschule Baden-Württemberg Stadt

von

Michael Sprauer

September 2013

Bearbeitungszeitraum 6. Semester

Matrikelnummer, Kurs 5147809, TAI10B2

Ausbildungsfirma Thales Defence & Security Systems GmbH, Pforzheim

Betreuer Prof. Dr. H. Braun

1. Erklärung

gemäß § 5 (2) der „Studien- und Prüfungsordnung DHBW Technik“ vom 18. Mai 2009.

Ich habe die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet.

Karlsruhe, Mai 2013

MICHAEL SPRAUER

1. Abstrakt

In der Schule müssen Lehrer regelmäßig einen Sitzplan erstellen, um in der Klasse eine optimale Lernatmosphäre zu gewährleisten. Dabei sind viele - Teils widersprüchliche – Anforderungen zu berücksichtigen. Beispielsweise müssen einige Schüler in der ersten Reihe sitzen, da sie eine Sehschwäche haben oder besonderer Betreuung durch den Lehrkörper bedürfen. Außerdem sollen nur die Schüler nebeneinander sitzen, die sich nicht vom Lernen abhalten. Besser währen Sitznachbarn, die das Lernen fördern. Damit insgesamt das Lernen in der Klasse optimal läuft muss auch der Geräuschpegel niedrig sein. Wer sich also zu gut versteht darf auch nicht nebeneinander sitzen. Diese Herausforderungen implizieren bereits, dass es keinen perfekten Sitzplan gibt. Diese Studienarbeit verwendet evolutionäre Algorithmen, um den optimalen Sitzplan mit den gegebenen Einschränkungen und Parametern zu finden.

1. Abstract

Creating a seating plan is a conventional task for every teacher. A well-defined seating plan is mandatory for an effective learning atmosphere in every class room. The requirements are partly conflicting. For example some of the students have to sit in the front row, because they have debility of sight or need special mentoring from the teacher. Other students can’t be placed next to each other, because they distract each other. In the best case the neighbours support each other. To facilitate the best possible learning environment the noise level should be very low. Chatting students shouldn’t sit side by side. All these challenges already imply that there is no such thing like a perfect seating plan. This thesis uses genetic algorithms to find the best possible solution considering all the given constraints and parameters.

Inhaltsverzeichnis

[1. Einleitung und Motivation 1](#_Toc350453766)

[1.1 Zielsetzung 1](#_Toc350453767)

[1.2 Wirtschaftliche Aspekte 1](#_Toc350453768)

[1.3 Gliederung der Arbeit 1](#_Toc350453769)

[2. Grundlagen: Evolutionäre Algorithmen 3](#_Toc350453770)

[3. Implementierung 4](#_Toc350453771)

[3.1 Software 4](#_Toc350453772)

[3.2 Entwicklungsumgebung 4](#_Toc350453773)

[3.3 Architektur 4](#_Toc350453774)

[3.3.1 Schnittstelle zwischen Netzwerkstack und Anwendungen 4](#_Toc350453775)

[4. Zusammenfassung und Ausblick 5](#_Toc350453776)

[4.1 Offene Punkte 5](#_Toc350453777)

[4.2 Performance 5](#_Toc350453778)

# Einleitung und Motivation

In der Schule müssen Lehrer regelmäßig einen Sitzplan erstellen, um in der Klasse eine optimale Lernatmosphäre zu gewährleisten. Dabei sind viele – zum Teil widersprüchliche – Anforderungen zu berücksichtigen. Beispielsweise müssen einige Schüler in der Ersten Reihe sitzen, da sie eine Sehschwäche haben oder besonderer Betreuung durch den Lehrkörper bedürfen. Außerdem sollen nur die Schüler nebeneinander sitzen, die sich nicht vom Lernen abhalten. Besser währen Sitznachbarn, die das Lernen fördern. Damit insgesamt das Lernen in der Klasse optimal läuft muss auch der Geräuschpegel niedrig sein. Wer sich also zu gut versteht darf auch nicht nebeneinander sitzen. Diese Herausforderungen implizieren bereits, dass es keinen perfekten Sitzplan gibt. Einen brauchbaren Plan zu erstellen, in dem die möglichst viele Aspekte berücksichtigt werden erfordert einen erheblichen Zeitaufwand.

## Zielsetzung

Ziel der Arbeit ist es einen optimalen Sitzplan zu finden, der die gegebenen Rahmenbedingungen und Parameter berücksichtigt. Dabei werden die folgenden Kriterien Berücksichtigt:

* Beziehung zum direkten Nachbar
* Beziehung zum übernächsten Nachbar
* Entfernung zur Tafel

## Wirtschaftliche Aspekte

Die Zeit zur Erstellung eines Sitzplans bedeutet für den Lehrer in der Regel einen Aufwand von etwa einer Stunde. Je nach Verhalten der Klasse oder Eignung des Sitzplans ist dieser Vorgang mehrmals pro Schuljahr notwendig. Diese Zeit könnte effektiver zur besseren Vorbereitung des Unterrichts verwendet werden.

## Gliederung der Arbeit

Diese Arbeit ist grob in zwei Teile unterteilt. Das Kapitel 2 Beschäftigt sich mit den theoretischen Grundlagen der evolutionären Algorithmen. Dann folgt in Kapitel 3 die Beschreibung der konkreten Realisierung der gestellten Aufgabe. Abschließend fasst Kapitel 4 die erzielten Ergebnisse zusammen und gibt einen Ausblick auf Untersuchungsmöglichkeiten um die Ergebnisse weiter zu verbessern.

# Grundlagen: Evolutionäre Algorithmen

Dieses Kapitel soll die Grundlagen der Studienarbeit erläutern. Als Verfahren zur Lösung des gestellten Problems eignen sich die evolutionären Algorithmen besonders. Dieses Verfahren wurde zwischen 1970 und 1980 von verschiedenen Wissenschaftlern unabhängig voneinander als effiziente Weise der Optimierung von numerischen Problemen erkannt. In Deutschland wurde das Thema durch Schwefel [1] und Rechenberg [2] unter dem Namen „*Evolutionsstrategie*“ bekannt. Auf der anderen Seite des Atlantiks veröffentlichten Goldberg und Holland [3] ähnliche Gedanken unter dem Title „*genetische Algorithmen*“. Insbesondere bei nicht linearen Optimierungsproblemen muss man Abwägen zwischen Rechenaufwand und Nutzen.

Problemklassifikation: NP? Daher evo Algo.

Die evolutionären Algorithmen bilden die Evolutionstheorie Darwins nach. Dort unterscheidet man zwischen Genotyp und Phänotyp. Der Phänotyp entspricht dabei dem konkreten Exemplar einer Spezies. Der Genotyp ist die genetisch kodierte Information über alle Merkmale und Eigenschaften dieser Art. Der Genotyp mutiert durch äußere Einflüsse bzw. wird durch die Fortpflanzung rekombiniert. Der neu Entstandene Genotyp bzw. der dazugehörige Phänotyp wird nun einem Fitnesstest unterzogen. Überlebt dieser Genotyp bzw. kann mehr Nachkommen erzeugen als andere Genotypen, spricht dies für seine Fitness. Damit werden nur die Vererbungslinien weiter verfolgt, die eine bessere Fitness haben als andere.

Diese Überlegung wird nun auf das gestellte Problem übertragen. Der Genotyp wird im Folgenden auch aus Repräsentation bezeichnet und in Kapitel 2.1 genauer beschrieben. Darauf folg die Erläuterung der Initialen Erstellung des ersten Genotyps durch eine sogenannte Greedy-Funktion in Kapitel 2.2. Kapitel 2.3 geht auf die Angewendete Fitnessfunktion ein. Die Kapitel 2.4, 2.5 und 2.6 beschreiben die Mutation, Reproduktion und Selektion im Kontext der gestellten Aufgabe. Abschließend beschreibt der letzte Abschnitt 2.7 das Ende dieses Iterativen Prozesses.

## Representation

Die Repräsentation bezeichnet die Kodierung eines konkreten Sitzplans.

## Greedy

## Fitness-Funktion

## Mutationsoperatoren

## Reproduktion

## Selektion

## Haltebedingung

# Implementierung

## Software

Eclipse, Git, JUnit, Swing

## Entwicklungsumgebung

Eclipse, Git, JUnit, Swing

## Software-Architektur

## User Interface

# Zusammenfassung und Ausblick

## Offene Punkte

Dieses Kapitel soll aufzeigen welche Teilaspekte des Projekts im Sinne der Anforderungen nicht optimal gelöst werden konnten.

## Verbesserungsmöglichkeiten

Parallelisierung der genetischen Algorithmen [4]

Mehrkernoptimierung. Evtl. CUDA (Siehe Andy)

1. Glossar

**Es wurden keine Verzeichnis-Einträge gefunden.**

1. Abbildungsverzeichnis
2. Tabellenverzeichnis
3. Listings
4. Literaturverzeichnis

[1] H.-P. Schwefel, „Evolutionsstrategie und numerische Optimierung“, Technische Universität Berlin, 1975.

[2] I. Rechenberg, „Evolutionsstrategie–Optimierung technischer Systeme nach Prinzipien der biologischen Evolution“, 1973.

[3] D. E. Goldberg und J. H. Holland, „Genetic algorithms and machine learning“, *Machine Learning*, Bd. 3, Nr. 2, S. 95–99, 1988.

[4] E. Cantú-Paz, „A survey of parallel genetic algorithms“, *Calculateurs paralleles, reseaux et systems repartis*, Bd. 10, Nr. 2, S. 141–171, 1998.