



Hochschule Rhein-Waal
Marie-Curie-Strasse 1, 47533 Kleve
Faculty Technology and Bionics
Bionics M. Sc.

EVOLUTIONARY ALGORITHMS

As a heuristic approach to the fair distribution of children in
classes in primary schools

Sophia Felicia Salome Döring
31519

Module:
Teacher:
Submission Date:

TABLE OF CONTENTS

| | |
|---|----|
| 1. INTRODUCTION | 4 |
| 1.1 EVOLUTIONARY ALGORITHMS | 5 |
| 1.2 EVOLUTIONARY ALGORITHMS FOR MULTI-OBJECTIVE OPTIMIZATION PROBLEMS . | 6 |
| 1.3 AIM OF THIS WORK | 6 |
| 2. MATERIAL AND METHODS | 7 |
| 2.1 SYSTEM REQUIREMENTS | 7 |
| 2.2 IMPLEMENTATION | 7 |
| 2.2.1 CREATION OF CANDIDATES(INDIVIDUALS) | 7 |
| 2.2.2 IMPLEMENTATION OF FITNESS | 8 |
| 2.2.3 REPRODUCTION | 10 |
| 2.2.4 EVOLUTION | 10 |
| 3. RESULTS | 10 |
| 4. DISCUSSION | 10 |
| 5. CONCLUSION | 10 |
| 5.1 OTHER SOURCES | 12 |

LIST OF FIGURES

LIST OF TABLES

| | | |
|---|---|---|
| 1 | STUDENTS CHARACTERISTICS AND THEIR POSSIBLE VALUES. | 8 |
| 2 | EIGHT STUDENTS WITH RANDOMLY CHOSEN VALUES | 8 |

ABSTRACT

1. INTRODUCTION

Equality, inclusion, and fairness of an educational system are of importance for the right to education. Even though this right and equal opportunity are stated in the German constitution (as well as in other constitutions), their implementation is complex and a controversial issue. (Zhang et al., 2014)

In order to realize the multifaceted challenge of equality, inclusion, and fairness (or to come as close as possible), a lot of work is put in in schools, especially at the beginning of a child's school career, to find a fair and equitable class distribution, to achieve a pleasant class climate, to support children who need more support in different areas, and to create a familiar environment, e.g. through friends from preschool/kindergarten.

But the controversy already starts with the question of what conditions, characteristics, and abilities of children are crucial for class formation. Depending on the school's philosophy, country, and other regulations, not only teachers, but also the parents can take part in the decision, especially if there are classes for specialization. Therefore, teachers spent a significant amount of time finding a fitting and fair class distribution before the beginning of the school year.

In the German public school system, a school entrance examination is conducted to assess whether a child is ready to start school. This examination evaluates not only child's school readiness but also a child's cognitive abilities, language skills, hyperactivity levels, and other developmental factors. Schools use these assessments to form balanced classes, ensuring that children with varying skill levels and degrees of hyperactivity are evenly distributed. The goal is for each class to have average skill and hyperactivity levels close to the overall average, promoting an equitable learning environment across classes. Even though this principle seemed to have little effect on the cognitive and non-cognitive abilities of children (Schneider, 2013), it has a positive effect on the stress level of the teacher and therefore on the learning environment. Additionally, the schools aim to put befriended children in the same class as placing friends in the same class can have several beneficial impacts on both social and academic outcomes. (Hallinan and Tuma, 1978), (Gail, 2020)

To create a class formation that, satisfies these goals and creates the desired environment for children and teachers can be challenging. Therefore, teachers spent a significant amount of time finding a fitting and fair class distribution before the beginning of the school year.

This complex challenge calls for innovative solutions that use modern technology, data-driven decision-making, and heuristic approaches. Within this context, Evolutionary Algorithms emerge as a promising tool to navigate the intricacies of class allocation. These computational methods are abstracted from the principles of evolution, natural selection, and genetic variation. Evolutionary Algorithms are particularly useful when dealing with complex and non-linear optimization, constraint satisfaction problems or multi-objective optimization problems such as class allocation.

In the context of this Applied Research Project for the Master's Degree Bionics (M. Sc.) a first implementation of an EA is created, which solves the class partitioning problem or presents a sufficiently adequate solution.

1.1 EVOLUTIONARY ALGORITHMS

Evolutionary algorithms (EAs) are population-based stochastic search techniques inspired by the principles of biological evolution. They are designed to find approximate solutions to complex problems (Bartz-Beielstein et al., 2014). Due to their flexibility and robustness, EAs excel at handling non-linear, high-dimensional, and poorly understood problems. These characteristics make them suitable for a wide range of applications. EAs offer a heuristic approach to optimization, making it possible to find satisfactory solutions in cases where optimal solutions either do not exist or are infeasible to compute using conventional methods. (Bartz-Beielstein et al., 2014; Michalewicz and Fogel, 2013)

Michalewicz and Fogel (2013) describes the important steps and principles of EA's as follows:

1. **Creating an initial Population:** A population is a collection of individuals (or candidate solutions). Individuals represent potential solutions to the problem being addressed. Each individual contains a set of parameters or variables, often referred to as its genotype, which encodes a solution. Individuals have the same genes, but different alleles (variation of a gene). The initial populations exists of an amount of individuals of different alleles.
2. **Implementing a Fitness:** fitness refers to the measure of how well an individual performs with respect to the optimization problem. The fitness function evaluates the quality of each individual in the population and is a mathematical or computational function.
3. **Create an Evolutionary Loop:** The evolutionary loop is the iterative process in evolutionary algorithms (EAs) that simulates the principles of natural evolution, such as selection, variation, and reproduction. The loop continuously evolves a population of candidate solutions through multiple generations until a stopping condition (e.g., a maximum number of generations, a fitness threshold, or convergence) is met. The loop can be divided into different important aspects:
 - a) **Choose Parents:** Choose Parents is the process of choosing individuals from the population to produce offspring for the next generation. The aim of parent selection is to favor individuals with higher fitness, allowing better solutions to pass their traits to the next generation while maintaining diversity in the population to explore different areas of the solution space. There are different options for the selecting method such as Roulette Wheel Selection or Tournament Selection.
 - b) **Create Offspring:** Create Offspring involves applying genetic operators to selected parent individuals to generate new solutions for the next generation. The two primary genetic operators used for creating offspring are crossover and mutation.
 - c) **Selection of the next Generation:** Selection of the next Generation is the step that determines which individuals from the current population will be chosen to contribute their genetic material to the next generation. The goal is to ensure that the more fit individuals are preserved and allowed to reproduce, while also maintaining diversity within the population to prevent converging to a local max and to explore different areas of the solution space. There are different options for the selecting method such as Roulette Wheel Selection or Tournament Selection.

1.2 EVOLUTIONARY ALGORITHMS FOR MULTI-OBJECTIVE OPTIMIZATION PROBLEMS

Multi-objective problems (MO) involve optimizing two or more conflicting objectives simultaneously. Unlike single-objective optimization, where a single goal is pursued, multi-objective optimization requires balancing trade-offs among multiple criteria, sometimes conflicting each other. There are several possible approaches to deal with multi-objective problems, these are the most common ones (Bartz-Beielstein et al., 2014):

- **Transfer to a single-objective problem:** combining multiple objectives into one comprehensive objective function. This transformation allows optimization techniques focusing on single objectives to be applied.
- **Pareto-Based Methods:** solving by finding a set of solutions that represent trade-offs between conflicting objectives. Pareto-based methods aim to find the Pareto optimal set, which consists of solutions where no objective can be improved without worsening at least one other objective.
- **Lexicographical Approach :** assigns varying levels of priority to different objectives, focusing on optimizing them in a specific order based on their importance. Higher-priority objectives are addressed first, and once they are optimized, attention shifts to the next objectives in the sequence.

Beyond these approaches, there are many other variations and hybrid methods. These are often specialized for particular use cases or aim to enhance specific aspects, such as maintaining diversity in a population, setting dominance relationships, weighting objectives, or solving sub-problems more effectively. (Ojha et al., 2019)

1.3 AIM OF THIS WORK

In this project, an evolutionary algorithm is applied to address a class distribution problem. Although there are no clear and direct trade-offs in the problem, subtle, non-visible trade-offs may arise due to specific individuals or a small subset of individuals, especially in a small school setting. The key question is whether a Pareto-optimal method, which requires more computational power, is necessary for this problem, or if transforming it into a single-objective problem would be sufficient.

For the single-objective problem the Weighted-Sum-Method (WSM) is used, where the fitness value of each characteristic of one candidate are given a weight and then summed up.

For the pareto-optimal method the Strength Pareto Evolutionary Algorithm 2 (SPEA2) created by Zitzler (2001) is implemented. SPEA2 assigns fitness based on dominance (how many solutions a solution outperforms) and density (how crowded the space around a solution is), using the "k-th nearest neighbor" method to ensure diverse solutions. It selects parents from a combination of the main population and an archive of the best non-dominated solutions.

It is hypothesized that a Pareto-optimal method, Strength Pareto Evolutionary Algorithm 2 (SPEA2), will either yield better results or match the performance of the single-objective approach, Weighted-Sum-Method (WSM). The quality of the algorithm will be evaluated based on computational time, the number of generations required, and the average fitness of the individuals in the final population.

2. MATERIAL AND METHODS

In the following chapter the system requirements for a basic approach for this problem are specified and the implementation of the evolutionary algorithm is described.

2.1 SYSTEM REQUIREMENTS

The following functional requirements and restraints are defined for the first approach of the algorithm:

1. **Restraints for class distribution**, class size, and number of students are predefinitions and data from Ministry of Education for public primary schools of NRW as well as from some public primary schools of NRW (für Schule und Bildung des Landes Nordrhein-Westfalen, 2021; der Ständigen Konferenz der Kultusminister der Länder in der Bundesrepublik Deutschland, 2020) (Neuß-Grevenbroicher Zeitung, 2016):
 - The class size must be over 14 and must be under 30
 - The minimal number of students is 31 the maximal number 600
 - The minimal class number is 2 and the maximum 7
2. **Important characteristics for class distribution** of the children are taken from:
 - The mean of males and females per class should be near the means of the overall students enrolling
 - The mean and standard deviation of hyperactivity per class should be near the mean and standard deviation of the overall students enrolling
 - The mean and standard deviation of language skill per class should be near the mean and standard deviation of the overall students enrolling
 - The mean and standard deviation of logical skill per class should be near the mean and standard deviation of the overall students enrolling
 - It should be possible to enter a max. of three friends per student.

The following non-functional requirements for the first basic version are defined:

1. **Programming language** is C
2. **The processing time** of the program does not exceed 60 s.

2.2 IMPLEMENTATION

The Evolutionary Algorithm to solve the problem was implemented in several separated steps.

2.2.1 CREATION OF CANDIDATES(INDIVIDUALS)

For creating a candidate (a class distribution) the first step is to implement the students themselves and their characteristics. The values for characteristics are shown in Table 1.

Table 1: **Students characteristics and their possible values.**Very of the six characteristics has a number (index) in the code and a possible value.

| no. | characteristics | possible values | value in |
|-----|----------------------|---|---|
| 0 | Student number | 0 to (no. of students - 1) | 0 to (no. of st |
| 1 | Sex | male, female | 0,1 |
| 2 | Hyperactivity level | very low, low, average, high, very high | 0 to |
| 3 | Logical skill level | very low, low, average, high, very high | 0 to |
| 4 | Language skill level | very low, low, average, high, very high | 0 to |
| 5-7 | Friendships | respective student no. | 0 to (no. of students - 1), 0 to (no. of st |
| 8 | Class | 0 to (no. of classes-1) | 0 to (no. of |

With these values a list of students was implemented with uniformly randomly chosen values for sex, hyperactivity, logical skill, language skill, and friendships. (see Table 2)

Table 2: **Eight students with randomly chosen values.** The eight students are resrepresentative for the 59 students in the program, the values are uniformly random chosen.

| student no. | sex | hyperactivity | log. skill | lang. skill | friendships |
|-------------|--------|---------------|------------|-------------|-------------|
| 0 | female | high | average | very high | 3 |
| 1 | male | very high | high | low | 4 |
| 2 | male | low | very high | high | |
| 3 | female | very low | high | very high | 0 |
| 4 | male | very high | average | very low | 1 |
| 5 | male | average | high | high | 6,7,2 |
| 6 | female | very high | very low | average | 5,7 |
| 7 | male | very low | very high | low | 5,6 |

After implementing the list of students, a population size of 1000 candidates is created. To generate a matrix of students in a uniformly random order, a random permutation is generated for the numbers from 0 to 58, and each student is assigned to their respective number.

To create classes, the number of students in each class is determined by generating a standard normal random variable $N(19,6711,56)$. The mean is the theoretical average of students per class. The standard deviation is calculated in a way that ensures the number of students per class does not exceed the allowed limits (15 and 29). To convert normally distributed values from uniformly random distributed values, the Box-Muller-Transformation is used. If the number of students per class exceeds the allowed limits, the values are recalculated to stay within the boundaries.

2.2.2 IMPLEMENTATION OF FITNESS

To compare the fitness values of candidates and to heuristically solve this multi-objective optimization problem, the difference between the mean of each class in one candidate and the mean of all students is calculated and summed up for the four characteristics: sex, hyperactivity, logical skill, and language skill. For every characteristic, the fitness values for one characteristic (sex, hyperactivity, logical skill level, and language skill level) are calculated as follows (Equation 1):

$$fit_{char} = \sum_1^i \sqrt{(\bar{x}_{students} - \bar{x}_i)^2} + \sum_1^i \sqrt{(\sigma_{students} - \sigma_i)^2} \quad (1)$$

fit_{char} is the fitness of a characteristic, $\bar{x}_{students}$ is the mean of the values of the respective characteristics of all students in one candidate, \bar{x}_i is the mean of the values of the respective characteristics of class i of the candidate, $\sigma_{students}$ is the standard deviation of the values of the respective characteristics of all students in one candidate, and σ_i is the standard deviation of the values of the respective characteristics of class i of the candidate.

For calculating the fitness value of friendships for a candidate a maximal value was defined (3 times the number of students), which is the value of not fulfilling any friendships with the respective combination of students and classes. For every fulfilled friendship 1 is subtracted.

For every fitness value, the optimal is 0.

As two different methods, transfer to a Weighted-Sum-Method (WSM) and Strength Pareto Evolutionary Algorithm 2 (SPEA2).

For the **Weighted-Sum-Method (WSM)** the fitness was calculated as follows:

$$fit_{WSM} = \sum_0^i (w_i \cdot f_{char_i}) \quad (2)$$

Where f_{char_i} is the fitness of characteristic i and w_i is the weight put on the fitness. As the different characteristics are of different scale and types the weights where assigned as follows:

- w_i of $i = 1, 2, 3$ are 1
- w_i of $i = 0$ is 4
- w_i of $i = 4$ is $\frac{\text{number-of-students}}{4}$

This method enables an equal influence of the characteristics on the fitness value.

For the **Strength Pareto Evolutionary Algorithm 2 (SPEA2)** the implementation of the fitness is taken from Zitzler (2001). The fitness consist of two values, the strength value and the density value. The strength is calculated as follows:

$$fit_{SPEA2_t} = \sum_0^j S_j \quad (3)$$

Where the fitness fit_{SPEA2_t} for one individual t is the sum of strength values $S_j \in [0; 1)$. strength values S_j is assigned by comparing each fit_{char_i} of the respective individual t with the fit_{char_i} of every other individual j .

$$S_j = \begin{cases} 0, & \text{if individual } t \text{ is dominated by or has equal values as individual } j \text{ in all characteristic} \\ 1, & \text{if individual } t \text{ dominates individual } j \text{ in one characteristics} \end{cases}$$

The density of each individual t is calculated using the k-nearest neighbor method. Therefore, the Euclidean distances to every other individual j is calculated:

$$d_{t,j} = \sqrt{\sum_0^i (fit_{char_i,t} - fit_{char_i,j})^2} \quad (4)$$

Where $fit_{char_i,t}$ is the the fitness of individual t for the characteristics i and $fit_{char_i,j}$ is the fitness of individual j for the characteristics i.

The distance of individual j to every other individual is calculated and then sorted in list from smallest to largest distance. To determine the k-th neighbor k is calculated as follows

$$k = \sqrt{N + N_a} \quad (5)$$

N is the population size and N_a is the size of the population in the archive. The distance $d_{t,j}$ at the position k of the list is then used to calculate the density D_t :

$$D_t = \frac{1}{d_{t,j} + 2} \quad (6)$$

The overall fitness of individual t is the sum of the fitness value fit_{SPEA2_t} and the density value D_t is then calculated as follows:

$$F_t = fit_{SPEA2_t} + D_t \quad (7)$$

2.2.3 REPRODUCTION

For selecting parents a tournament selection after horn nafpliotis was implemented. Therefore, two parents as well as a comparison set of 10 individuals were uniformly randomly selected from the population. The parents dominating (Pareto-based) more of the individuals in the comparison set are selected with a chance of 95% to procreate.

After the selection of a parent, point mutations are used to alter the permutation of students in the parent and to get a child with slightly different fitness values. If the child is dominating the parent it has a 95% of being part of the next generation (replacing the parent) and vice versa.

The selection and reproduction process is repeated 10 times.

2.2.4 EVOLUTION

The process of reproduction is repeated for 100 generations

3. RESULTS

4. DISCUSSION

5. CONCLUSION

REFERENCES

LITERATURE

- Bartz-Beielstein, T., Branke, J., Mehnen, J., and Mersmann, O. (2014). Evolutionary algorithms. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4(3):178–195.
- der Ständigen Konferenz der Kultusminister der Länder in der Bundesrepublik Deutschland, S. (2020). Vorgaben für die Klassenbildung 2020/2021.
- für Schule und Bildung des Landes Nordrhein-Westfalen, M. (2021). Das Schulwesen in Nordrhein-Westfalen aus quantitativer Sicht 2020/21 - Statistische Übersicht Nr. 413.
- Hallinan, M. T. and Tuma, N. B. (1978). Classroom effects on change in children’s friendships. *Sociology of Education*, pages 270–282.
- Michalewicz, Z. and Fogel, D. B. (2013). *How to Solve It: Modern Heuristics*. Springer Science & Business Media.
- Ojha, M., Singh, K. P., Chakraborty, P., and Verma, S. (2019). A review of multi-objective optimisation and decision making using evolutionary algorithms. *International Journal of Bio-Inspired Computation*, 14(2):69–84.
- Schneider, T. (2013). School class composition and student development in cognitive and non-cognitive domains: longitudinal analyses of primary school students in germany. *Integration and inequality in educational institutions*, pages 167–190.
- Zhang, H., Chan, P. W. K., and Boyle, C. (2014). *Equality in education: Fairness and Inclusion*. Springer.
- Zitzler, E. (2001). Spea2: Improving the strength pareto evolutionary algorithm. *TIK Report*.

5.1 OTHER SOURCES

Neuß-Grevenbroicher Zeitung (2016). Eine der größten grundschulen in nrw. https://rp-online.de/nrw/staedte/neuss/eine-der-groessten-grundschulen-in-nrw_aid-21453971, Accessed on the [2024-10-25].

Gail, C. (2020). What the research says about the academic power of friendship. <https://www.kqed.org/mindshift/56979/what-the-research-says-about-the-academic-power-of-friendship>, Accessed on the [2024-10-25].