

# Geração de Timetabling considerando avaliação do ENADE

Matheus Felipe Fabian<sup>1</sup>, Marcelo Trindade Rebonatto<sup>1</sup>

<sup>1</sup>Universidade de Passo Fundo (UPF)  
Programa de Pós-Graduação em Computação Aplicada  
Passo Fundo – RS – Brasil

{194404, rebonatto}@upf.br

**Abstract.** *This paper describes the advances of a project for modification of a genetic algorithm built for the Curriculum-Based Course Timetabling Problem. This very common problem in universities consists of allocating professors to teach subjects at specific times. The objective is that this modification optimizes the concept of the institution's courses that are evaluated in Brazil in the years in which ENADE is taken.*

**Resumo.** *Este artigo descreve os avanços de um projeto para a incrementação de um algoritmo genético construído para resolver o problema do Curriculum-Based Course Timetabling Problem. Este problema muito comum em universidades consiste em alocar professores para ministrar disciplinas em horários específicos. O objetivo é que esta solução se adapte de forma a otimizar o conceito dos cursos de uma instituição de ensino superior que são avaliados no Brasil nos anos em que o ENADE é realizado.*

## 1. Introdução

O Curriculum-Based Timetabling Problem (CB-CTT) é um problema do tipo NP-completo, onde professores são distribuídos para lecionar determinadas disciplinas em determinados períodos de tempo. Para se criar uma solução viável para o problema busca-se satisfazer as *hard* e minimizar as *soft constraints* [Alnowaini and Aljomai 2021]. Por ser um problema de otimização, soluções como algoritmos genéticos [Dunke and Nickel 2023], algoritmos meméticos [Ahandani and Baghmisheh 2011], colônia de formigas [Sakal et al. 2021] entre outras meta-heurísticas são utilizadas, pois resolvem com eficiência problemas de satisfação de restrições.

Esse é um problema muito comum nas universidades e resolvê-lo manualmente leva muito tempo e esforço, sem garantias de que as *constraints* sejam resolvidas com eficiência. A Universidade de Passo Fundo (UPF) já possui uma ferramenta implementada com uma abordagem com algoritmos genéticos.

O algoritmo já construído apresenta bons resultados, conseguindo gerar soluções viáveis que praticamente resolvem todas as *hard constraints*, além de ser totalmente adaptado para o modelo estrutural da universidade, possuindo funcionalidades como disciplinas compartilhadas e disciplinas EAD. Visando sempre uma otimização, com um algoritmo que já é funcional, alguns direcionamentos podem tornar a ferramenta ainda mais útil para a universidade. Visto que no Brasil, as instituições de ensino superior passam

por avaliações nos cursos em determinados períodos de tempo, gerar grades horárias que otimizam em relação aos critérios de avaliação da instituição pode ser um avanço significativo para considerarmos a ferramenta mais útil para a universidade.

O objetivo deste trabalho é melhorar a ferramenta computacional existente para que solução se adapte considerando os critérios utilizados no cálculo da nota do corpo docente na avaliação de cada curso pelo ENADE. Dessa forma, o algoritmo deve gerar soluções que proporcionem maiores conceitos para os cursos da universidade em relação aos insumos relacionados ao corpo docente.

## **2. Referencial Teórico**

Nesta seção serão abordados os dois principais conceitos acerca do trabalho. Primeiramente o tema algoritmo genético, e após, como que realizada a avaliação do sistema de educação superior no Brasil.

### **2.1. Algoritmo genético**

Algoritmos Genéticos são uma categoria dentre os chamados algoritmos evolucionários. Este tipo de algoritmo funciona através de iterações, gerando uma série de indivíduos, e a cada iteração novos indivíduos são gerados e avaliados de acordo com *constraints* definidas para que se adaptem de acordo com a solução desejada [Talbi 2009].

A estrutura básica de um algoritmo genético se baseia em inicialização, operadores genéticos (crossover, mutação) e avaliação. [Jacobson and Kanber 2015]. A inicialização é onde os indivíduos são gerados aleatoriamente ou em alguns casos há o uso de heurísticas para criar indivíduos melhor adaptados [Talbi 2009]. Após a inicialização, muitas vezes se utiliza elitismo, que seleciona alguns indivíduos para compor a nova geração, preservando boas soluções [Jacobson and Kanber 2015].

Para compor o restante da nova geração, os indivíduos passam por transformações através dos operadores genéticos. Esses operadores geralmente são *crossover*, que unifica duas ou mais soluções formando um novo indivíduo, e a mutação que gera novas soluções aplicando pequenas mudanças em indivíduos selecionados. Por fim é realizado a avaliação dos indivíduos, através de uma função objetivo para que se classifique quais indivíduos solucionam o problema de maneira mais eficiente. Este método é repetido diversas vezes até que os critérios de parada sejam satisfeitos [Jacobson and Kanber 2015].

Este tipo de algoritmo é muito utilizado para resolver problemas de otimização da vida real em ciência, engenharia, economia e negócios, pois são problemas complexos e difíceis de resolver. Eles não podem ser resolvidos de maneira exata dentro de um período de tempo razoável, pois existem muitas soluções possíveis [Talbi 2009].

### **2.2. Avaliação da Educação no Brasil**

No Brasil o órgão responsável por avaliar as instituições de ensino superior (IES) é o Sistema Nacional de Avaliação da Educação Superior (SINAES). A avaliação é composta pela Avaliação Institucional da IES, Avaliação do Desempenho dos Estudantes (ENADE) e pela Avaliação dos Cursos de Graduação [Guerra and Cavalcanti 2020].

Dessas três avaliações, a Avaliação dos Cursos é feita através da medição do Conceito Preliminar de Curso (CPC) e o Índice Geral de Cursos (IGC)

[Guerra and Cavalcanti 2020]. O CPC é calculado através de quatro dimensões, sendo elas Desempenho dos Estudantes, Valor agregado pelo processo formativo oferecido pelo curso, Percepção Discente sobre as Condições do Processo Formativo e Corpo Docente [QUINTILIO 2019].

A nota do corpo docente para o cálculo do CPC é obtida a partir da proporção do corpo docente em fatores de titulação e regime de trabalho. Estes dois fatores são compostos de três insumos no qual são calculadas a proporção de doutores, proporção de mestres e proporção de docentes com regime de trabalho parcial ou integral (não horistas) sobre a quantidade total de docentes no curso [QUINTILIO 2019].

### 3. Metodologia

Para que as mudanças realizadas fossem refletidas apenas nos cursos que realizam o ENADE no ano vigente, o início da execução do algoritmo teve de ser alterado para incluir a informação do ano. Além disso, foram adicionadas *constraints* no algoritmo para considerar os insumos utilizados no cálculo da nota do corpo docente do curso, permitindo que as gerações do algoritmo evoluam também com foco na maximização dessa nota. As novas *constraints* foram analisadas para se enquadrarem no que é proposto para o cálculo da nota do CPC. Os pesos e limites de cada uma das novas *constraints* ainda podem ser reavaliados conforme o andamento do trabalho, assim como das *constraints* já existentes podem ser alterados.

Na função de avaliação do algoritmo que calcula a eficiência de cada indivíduo gerado, foi necessário verificar a titulação e o regime de trabalho de cada professor alocado para ministrar uma disciplina em algum curso onde o ENADE será aplicado no ano. Dessa forma, é possível realizar uma contagem dos professores que não atingem aos requisitos impostos para cada um dos insumos utilizados pelo conceito do curso. Essa contagem é contabilizada como número de violações de cada *constraint* criada e é usada na fórmula do cálculo do *score* do indivíduo.

Após contadas as violações, as novas *constraints* finalmente podem ser inseridas no cálculo do *score* do indivíduo. A fórmula do cálculo é apresentada na Equação 1. Nela,  $w_i$  é o peso da *constraint*,  $v_i$  é o número de violações da *constraint* e  $l_i$  é o limite da *constraint*, sendo que o *score* de cada *constraint* deve variar de zero ( $l_i$  violações) a  $w_i$  (não ocorrem violações). O somatório do *score* de cada *constraint* formará o *fitness* ( $f_i$ ) do indivíduo. Quanto maior o valor de  $f_i$ , mais adaptado ao meio é o indivíduo.

$$f_i = \sum_{i=1}^n w_i \times \frac{l_i - v_i}{l_i} \quad (1)$$

Para cada alteração realizada, o algoritmo genético será validado de forma a calcular a nota resultante na grade gerada para verificar se ela está aumentando de acordo com cada mudança realizada. Para cada validação será necessário que se execute o algoritmo um número determinado de vezes para obter o resultado médio das gerações, buscando obter indivíduos com aptidão superior as gerações sem a alteração realizada.

Os testes devem ser realizados de forma a comparar execuções com as alterações e execuções sem as alterações. Devem ser analisados se com os resultados das novas execuções as proporções dos professores que atingem aos insumos avaliados pelo ENADE

umentem, mas também cuidando com possíveis efeitos colaterais em outras pontas da solução, por exemplo, o número de *clashes* (conflitos onde um professor é alocado para lecionar duas disciplinas ao mesmo tempo) não deve aumentar consideravelmente, se não a solução gerada não pode ser considerada uma solução viável para o problema.

No atual momento as principais modificações já foram realizadas e as primeiras execuções foram feitas. Os primeiros resultados já produzem soluções com uma porcentagem maior de professores alocados com titulação e regime de trabalho desejados. Também se observa um pequeno aumento na quantidade de conflitos do tipo *clash*. Os próximos passos serão analisar se essas mudanças são estatisticamente significativas.

#### 4. Conclusão

Este trabalho traz uma proposta de modificação do algoritmo genético implementado para o CB-CTT, direcionando as soluções a gerarem grades horárias que levem um conceito maior para avaliação do ENADE perante os cursos da universidade. O trabalho ainda se encontra em desenvolvimento, e os primeiros testes mostraram otimismo em relação aos resultados.

Porém também é preciso frisar que apenas foram realizadas pequenas execuções para ser ter uma ideia de como o algoritmo está se comportando com as novas mudanças. Mais execuções devem ser realizadas, com mais indivíduos e gerações. Os pesos das *constraints* ainda podem ser alterados e uma análise estatística será realizada em torno desses testes para que se possa afirmar a relevância dos resultados adquiridos.

#### References

- Ahandani, M. A. and Baghmisheh, M. T. V. (2011). Memetic algorithms for solving university course timetabling problem. In *2011 1st International eConference on Computer and Knowledge Engineering (ICCKE)*, pages 40–44. IEEE.
- Alnowaini, G. and Aljomai, A. A. (2021). Genetic Algorithm For Solving University Course Timetabling Problem Using Dynamic Chromosomes. In *2021 International Conference of Technology, Science and Administration (ICTSA)*, pages 1–6. IEEE.
- Dunke, F. and Nickel, S. (2023). A matheuristic for customized multi-level multi-criteria university timetabling. *Annals of Operations Research*.
- Guerra, M. and Cavalcanti, L. (2020). *AVALIAÇÃO DA EDUCAÇÃO SUPERIOR NO BRASIL: DAS PRIMEIRAS REGULACOES ATÉ O SISTEMA NACIONAL DE AVALIAÇÃO DA EDUCAÇÃO SUPERIOR (SINAES)*. UFPB.
- Jacobson, L. and Kanber, B. (2015). *Genetic Algorithms in Java Basics*. Apress.
- QUINTILIO, R. (2019). Cálculo da nota final do Exame Nacional de Desempenho dos Estudantes (Enade). Technical report, Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira.
- Sakal, J., Fieldsend, J. E., and Keedwell, E. (2021). Learning assignment order in an ant colony optimiser for the university course timetabling problem. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO '21*, page 77–78, New York, NY, USA. Association for Computing Machinery.
- Talbi, E.-G. (2009). *Metaheuristics: From Design to Implementation*.