

## UM ALGORITMO GENÉTICO PARA A PROBLEMA DE GERAÇÃO DE HORÁRIOS A PARTIR DE CURRÍCULOS COM DISCIPLINAS COMPARTILHADAS

<sup>1</sup>MARCELO TRINDADE REBONATTO, <sup>1</sup>MATHEUS FELIPE FABIAN, <sup>1</sup>FELIPE GAZZONI FOSCHIERA

<sup>1</sup>Universidade de Passo Fundo (UPF)

<rebonatto@upf.br>, <matheusffabian@gmail.com>, <felipefoschiera@gmail.com>

DOI: 10.21439/conexoes.v19.3440

**Resumo.** A geração de horários semanais de professores e disciplinas (*timetabling*) em Instituições de Ensino é um problema complexo, que demanda um elevado tempo para ser resolvido de forma manual ou computacionalmente. Esse problema é potencializado pelas novas necessidades das instituições, como por exemplo o compartilhamento de turmas entre diferentes cursos. O objetivo deste trabalho é construir e validar um algoritmo genético para geração da grade horária da universidade, considerando restrições como compartilhamento de disciplinas e controle de grade com disciplinas não presenciais. O projeto é uma pesquisa experimental realizada para adequar a geração de grades horárias com as necessidades da universidade. Foi desenvolvida a implementação e validação de um Algoritmo Genético que atende de forma satisfatória aos novos requisitos para a geração da grade horária. Os resultados obtidos se demonstraram excelentes, com indivíduo gerado alcançando 99,92% do *score* máximo. Assim é possível concluir que a resolução do problema, mesmo com novas restrições é possível e viável favorecendo o processo de organização de horários. As contribuições foram uma revisão da literatura acerca de funcionalidades e operadores usados em Algoritmos Genéticos que solucionam um *timetabling*, o algoritmo implementado e validado que gera soluções viáveis e aptas às necessidades impostas pela universidade.

**Palavras-chave:** grades de horários; gerência de cursos; Instituições de Ensino Superior (IES); inicialização heurística; mutação por trocas.

## A GENETIC ALGORITHM FOR THE CURRICULUM-BASED COURSE TIMETABLING WITH SHARED SUBJECTS

**Abstract.** Generating weekly schedules for teachers and subjects (*timetabling*) in educational institutions is a complex problem that requires significant amount of time to be solved manually or computationally. This problem is further amplified by the new necessities of institutions, such as the sharing of subjects among different courses. The objective of this paper is to build and validate a genetic algorithm for generating the university timetabling, considering restrictions such as sharing classes and controlling the timetabling with non-attendance classes. The project is an experimental research carried out to adapt the generation of timetables to the needs of the university. The implementation and validation of a Genetic Algorithm was developed, which satisfactorily meets the new requirements for the timetable. The results were excellent, with the generated individual reaching 99.92% of the maximum score. Therefore, it is possible to conclude that the resolution of the problem, even with new restrictions, is possible and feasible, favoring the process of organizing the Institutions' timetables. The contributions were a review of the literature about features and operators used in Genetic Algorithms that solve a *timetabling*, in addition to the implemented and validated algorithm that generates feasible and suitable solutions to the needs imposed by the university.

**Keywords:** timetables; course management; Higher Education Institutions (HEIs); heuristic initialization; mutation by exchanges.

## 1 INTRODUÇÃO

O problema da geração de grade de horários de cursos, baseada em currículos (*curriculum-based course timetabling problem - CB-CTT*) é um problema de otimização NP-completo, ou seja, não é possível resolvê-lo de forma ótima em tempo polinomial (Cooper; Kingston, 1996; Even, 1975). A resolução deste problema é imprescindível para as Instituições de Ensino Superior (IES), pois ele realiza a distribuição das disciplinas ofertadas pelos cursos aos alunos em períodos semanais pré-definidos, associando-as a professores disponíveis e aptos em ministrá-las (Bettinelli, 2015). Normalmente, essa alocação é realizada de forma manual, pelos gestores dos cursos utilizando planilhas eletrônicas, tarefa que demanda muito esforço e que pode demorar semanas para ser realizada, sem garantias de que as restrições sejam respeitadas. As restrições tradicionalmente encontradas são classificadas como *hard* ou *soft constraints*.

Usar uma solução computacional para resolver o CB-CTT atualmente é uma necessidade, mesmo que a definição final dos horários passe por um processo de refinamento, realizado de forma manual pelos gestores, a fim de satisfazer requisitos específicos. Ao proporcionar uma solução inicial que reduz ao máximo as restrições, em alguns casos chegando a minimizar/eliminar ocorrências de violações, esta excelente solução inicial pode ser aprimorada conforme a necessidade. Com o uso de algoritmos estocásticos de busca, como os algoritmos evolucionários, o problema CB-CTT pode ser modelado com o uso de funções objetivo (Eiben; Smith, 2015) que devem ser otimizadas no processo de satisfazer aos requisitos.

Na literatura, abordagens conhecidas para resolver problemas similares incluem, por exemplo, o uso de meta-heurísticas (Bashab, 2020) como os Algoritmos Genéticos (AGs) (Dunke; Nickel, 2023; Wong, 2022; Alves, 2017; Gozali, 2020; Ngo, 2021), Algoritmos Meméticos (Ahandani; Baghmisheh, 2011; Nguyen *et al.*, 2011) ou Hiper-Heurísticas (Els; Pillay, 2010).

Nos últimos anos as IES estão passando por diversas transformações, em virtude das novas metodologias de ensino, pela acirrada concorrência e por crises financeiras. Neste período, a situação sanitária em virtude da COVID-19 agravou a crise financeira de diversos setores da economia, entre eles o da educação. Os efeitos causados pela pandemia ainda vão demorar algum tempo para serem superados. Portanto, uma solução aplicada aos dias atuais deve considerar novas realidades das IES, tais como:

- Disciplinas compartilhadas: ocorrem quando mais de uma oferta da disciplina é prevista, em diferentes cursos. Nesse caso, a disciplina deve ser ministrada por um mesmo professor em igual horário semanal, em todos os cursos envolvidos;
- Créditos não presenciais (EaD): são créditos de uma disciplina que são ministrados de forma remota, sem serem considerados na grade horária do curso, porém a carga horária deve ser alocada ao professor;
- Disciplinas fora da grade horária: são aquelas que possuem um professor definido, porém sem a necessidade de um horário semanal fixo, tais como orientações de trabalhos ou supervisões de estágio.

Dentre as novas funcionalidades encontradas nos problemas de CB-CTT o compartilhamento de disciplinas é o que possui maior impacto para a aplicação da solução em IES. Com a drástica redução dos alunos matriculados, as IES devem otimizar seus processos, incluindo os custos com os professores. Por exemplo, uma disciplina de "Ética Geral", ofertada nos cursos de Agronomia e de Administração, pode ser ministrada compartilhada (em conjunto). Ainda, uma disciplina de Algoritmos, ofertada nos cursos de Ciência da Computação e Engenharia de Computação, pode ser ministrada em apenas uma turma e um professor. Essa estratégia, reduz custos e proporciona maior diversidade de alunos, enriquecendo os debates da disciplina. Porém, para possibilitar o uso de disciplinas compartilhadas é necessário um meticuloso processo de alocação conjunta dos horários, aumentando o esforço para encontrar uma solução das grades dos cursos.

Neste artigo é proposto e validado um AG para a solução do CB-CTT, tendo seus resultados analisados. A solução considera as novas necessidades das IES, como compartilhamento de disciplinas e controle da grade com disciplinas ou créditos não sendo adicionados (porém, contabilizados nos currículos).

## 2 FUNDAMENTAÇÃO

Uma solução ideal para a grade de horários deve satisfazer os *hard constraints* e buscar satisfazer os *soft constraints* da melhor maneira possível, embora eles possam ser violados em determinadas situações (Bashab, 2020). A qualidade de uma solução é melhorada minimizando o número de violações dos *constraints*.

Segundo Bettinelli (2015) os *hard constraints* básicos são: a) um professor não pode ministrar duas ou mais disciplinas ao mesmo tempo (*clash*); b) um professor não pode ministrar uma disciplina em um horário que não possui disponibilidade; c) todos oferecimentos de disciplinas devem ser alocados em períodos semanais distintos.

Ao analisar os *hard constraints*, nota-se a dificuldade em suportar disciplinas compartilhadas. Elas vão de encontro aos *hard constraints*, uma vez que deverão ser implementados *clashes*, porém de forma intencional e controlada.

Diversos *soft constraints* são relacionados: 1) selecionar um professor preferencial para a disciplina; 2) alocar horas o mais próximo possível da quantidade definida para um professor; 3) reduzir o número de dias na semana com aula no curso; 4) minimizar o número de turnos com aulas a, no máximo, dois.

Além dos *constraints* identificados na literatura, alguns requisitos adicionais para a resolução de forma satisfatória pelas IES podem ser relacionadas: a) disciplinas compartilhadas; b) suporte a disciplinas com créditos não presenciais e fora da grade de horário; c) divisão de turmas em vários grupos, a fim de que disciplinas práticas ou com grande número de alunos possam ser implementadas; d) fechamento de grade horária sem a definição do professor; e) controlar o máximo de horas trabalhadas no dia por um professor; e) os professores não devem trabalhar por mais de oito (8) horas diárias, por motivos legais e para garantir a qualidade das aulas ministradas.

Dessa forma, percebe-se que o problema a ser resolvido difere, de forma significativa, do CB-CTT tradicionalmente encontrado. Para tanto, é necessária uma nova abordagem, que contemple os requisitos atualmente demandados.

### 3 TRABALHOS RELACIONADOS

O CB-CTT vem sendo estudado há muito tempo, através de diversas abordagens, como as meta-heurísticas (Bashab, 2020), programação inteira (Feng, 2017; Rappos, 2022) e lógica difusa (Babaei, 2016). Categorias comuns de problemas de geração de horário incluem a geração baseada em currículo (CB-CTT), *post-enrollment timetabling* (PE-CTT) e *examination timetabling* (Bettinelli, 2015).

Entre as meta-heurísticas, colônias artificiais de abelhas (Bolaji, 2013), colônias de formigas (Sakal, 2021), algoritmos meméticos (Nguyen, 2011), otimização por enxame de partículas (Marie-Sainte, 2015), algoritmos híbridos (Jaengchuea; Lohpetch, 2015) e algoritmos genéticos (Gozali, 2020; Ngo, 2021; Alnowaini; Aljomai, 2021), são normalmente encontrados.

AGs estão entre as meta-heurísticas populares para a resolução de *timetabling*, com mais publicações que outros métodos, apenas atrás dos algoritmos híbridos (Bashab, 2020). AGs pertencem a classe dos algoritmos evolucionários, inspirados no processo de seleção natural a partir da teoria da evolução de Darwin (Eiben; Smith, 2015).

AGs começam com uma população inicial de cromossomos (indivíduos), em alguns casos gerados aleatoriamente, que representam soluções para um problema. Os cromossomos (soluções) são melhorados por gerações sucessivas e avaliados por meio de sua adaptação ao meio (problema) através da função de *fitness*. A avaliação busca representar o quão boa é uma solução para um problema (Eiben; Smith, 2015).

Uma revisão de literatura foi realizada, buscando compreender quais operadores genéticos (OGs) e métodos de inicialização são utilizados (Tabela 1). Em alguns AGs são encontrados recursos de elitismo, reinicialização parcial e *niching* (Shir, 2012).

Raghavjee e Pillay (2010) propuseram o *Informed Genetic Algorithm* para resolver o problema de *timetabling* em IES. A execução é realizada em duas fases, sendo que na primeira possui como objetivo resolver as *hard constraints* e na segunda o AG é usado para minimizar as *soft constraints* das soluções da primeira fase.

Abordagens híbridas envolvendo AGs podem ser encontradas, como em Ahandani e Baghmisheh (2011) onde é proposto o uso de *memetic algorithms* combinando AG e métodos de busca local no processo. Na mesma linha, Abdelhalim e Khayat (2016) introduziram a um AG heurísticas para geração de boas soluções iniciais. Métodos de busca local são usados na inicialização das populações em (Nguyen, 2011; Badoni, 2014).

Diferentes alternativas foram exploradas em operadores genéticos, como a *truncation selection* uma estratégia de seleção que consiste em ordenar indivíduos pelo *fitness* e selecionar alguma proporção dos melhores (Abdelhalim; Khayat, 2016; Nguyen, 2011). Em Alves (2017), uma versão modificada do OX *crossover* é utilizada, sendo um operador de permutação que recombina indivíduos, trocando um intervalo de alelos consecutivos e preenchendo os valores restantes na ordem em que apareceram anteriormente no pai. A exploração do operador de

UM ALGORITMO GENÉTICO PARA A PROBLEMA DE GERAÇÃO DE HORÁRIOS A PARTIR DE CURRÍCULOS COM DISCIPLINAS COMPARTILHADAS

**Tabela 1:** Métodos de inicialização, operadores genéticos e funcionalidades.

| Referência                      | Inicialização | Seleção                    | Recombinação       | Mutação                  | Elitismo | Disciplinas    |     | Fechamento de Grade |
|---------------------------------|---------------|----------------------------|--------------------|--------------------------|----------|----------------|-----|---------------------|
|                                 |               |                            |                    |                          |          | compartilhadas | EaD |                     |
| (Borges, 2003)                  | Aleatória     | Roleta                     | <i>PMX</i>         | <i>Swap</i>              | Sim      | Não            | Não | Não                 |
| (Fucilini <i>et al.</i> , 2008) | Aleatória     | Roleta                     | Ponto único        | <i>Swap</i>              | Não      | Não            | Não | Não                 |
| (Raghavjee; Pillay, 2010)       | Heurística    | Torneio                    | -                  | <i>Swap</i>              | Sim      | Não            | Não | Não                 |
| (Suyanto, 2010)                 | Heurística    | <i>Truncamento</i>         | -                  | Dirigida                 | Sim      | Não            | Não | Não                 |
| (Els; Pillay, 2010)             | Heurística    | Torneio                    | Ponto único        | Troca de heurística      | Não      | Não            | Não | Não                 |
| (Ahandani; Baghmisheh, 2011)    | Aleatória     | Roleta                     | <i>Uniforme</i>    | -                        | Sim      | Não            | Não | Não                 |
| (Phuc <i>et al.</i> , 2011)     | Aleatória     | Torneio, <i>Rank</i>       | Múltiplos          | Troca aleatória          | Sim      | Não            | Não | Não                 |
| (Nguyen <i>et al.</i> , 2011)   | <i>Grasp</i>  | <i>Truncamento</i>         | Múltiplos          | Troca aleatória          | Sim      | Não            | Não | Não                 |
| (Badoni <i>et al.</i> , 2014)   | Busca local   | Torneio                    | <i>Uniforme</i>    | <i>Neighborhood move</i> | Sim      | Não            | Não | Não                 |
| (Jaengchuea; Lohpetch, 2015)    | Busca local   | Torneio                    | <i>Uniforme</i>    | -                        | Sim      | Não            | Não | Não                 |
| (Abdelhalim; Khayat, 2016)      | Heurística    | <i>Truncamento</i>         | <i>Utilization</i> | <i>Targeted</i>          | Sim      | Sim            | Não | Não                 |
| (Alves <i>et al.</i> , 2017)    | Aleatória     | Aleatória                  | <i>OX</i>          | <i>Swap</i>              | Não      | Não            | Não | Não                 |
| (Gozali <i>et al.</i> , 2020)   | Heurística    | Roleta                     | Ponto múltiplo     | Múltiplos                | Sim      | Não            | Não | Não                 |
| (Alnowaini; Aljomai, 2021)      | Aleatória     | Roleta                     | Ponto único        | Troca aleatória          | Não      | Não            | Não | Não                 |
| (Wong <i>et al.</i> , 2022)     | Busca local   | Estado <i>estacionário</i> | Ponto único        | Múltiplos                | Não      | Não            | Não | Não                 |
| (Dunke; Nickel, 2023)           | Aleatória     | Torneio                    | Ponto único        | Troca aleatória          | Não      | Não            | Não | Não                 |

Fonte: Autores.

recombinação (um ponto, dois pontos ou uniforme) pode ser encontrada em (Nguyen, 2011; Phuc, 2011), com diferentes estratégias associadas às frequências de melhoria da nova geração.

Dentre os trabalhos consultados, além dos operadores genéticos foram buscadas funcionalidades que contemplem a nova realidade de requisitos das IES em relação ao problema CB-CTT. Foram buscados suporte a: a) disciplinas compartilhadas; b) disciplinas com créditos não presenciais e fora da grade de horário; c) divisão de turmas em vários grupos; d) fechamento de grade horária sem a definição do professor; e) controle do máximo de horas trabalhadas no dia por um professor.

Nenhum dos trabalhos consultados buscou resolver o problema considerando disciplinas no formato EaD ou que não devam ser considerados na grade regular do curso. De mesma forma, os trabalhos não contemplavam múltiplas turmas, no mesmo horário ou outro, o limite de horas diárias por professor e o fechamento da grade de um curso sem a indicação do professor. Em relação a disciplinas compartilhadas, apenas o trabalho descrito em (Abdelhalim; Khayat, 2016), considera a ocorrência de disciplinas compartilhadas, porém de forma superficial. Assim, um novo AG foi proposto, atendendo aos requisitos tradicionais do problema CB-CTT e considerando todas as novas restrições, não contempladas pelos trabalhos descritos na Tabela 1.

#### 4 METODOLOGIA

No AG proposto, apenas o elitismo, o cruzamento e a mutação são utilizados. As disciplinas tem um número de créditos equivalentes à períodos semanais. Cada dia de segunda à sexta contém quatro períodos pela manhã, à tarde e à noite. Além disso, há quatro períodos no sábado de manhã, totalizando 64 períodos semanais.

Os cursos são divididos em níveis, onde ocorrem os oferecimentos das disciplinas. Cada nível de um curso tem um turno especificado, referente à um grupo de períodos semanais, e as alocações de disciplinas devem ocorrer nesses turnos. São possíveis 32 tipos de turnos diferentes. Por exemplo, um curso noturno possui um turno diferente de outro que funciona manhã e tarde ou do que funciona à noite e no sábado.

Cada indivíduo possui uma representação computacional de sua constituição, de forma similar ao genótipo em genética (Eiben; Smith, 2015). Um cromossomo na solução desenvolvida é representado utilizando três camadas. Os indivíduos são indexados inicialmente pelo curso (similar a alelos em genes) sendo em seguida organizados pelos níveis dos cursos. A última camada de cada indivíduo representa os períodos semanais de cada nível de um curso. Em cada um desses períodos são armazenadas tuplas ⟨professor, disciplina⟩, representando a alocação de uma disciplina no horário. O número de cursos assim como o de níveis é variável, porém o número de horários semanais é fixo: 64.

Com o objetivo de atender as novas funcionalidades propostas como disciplinas não presenciais, que não devem ser alocadas na grade do curso e várias turmas da mesma disciplina, estruturas adicionais nos períodos de cada nível dos cursos foram adicionadas. Nessas estruturas, pares de professores e disciplinas são alocados para manter o controle dessas funcionalidades, agendando horários para os professores relacionados.

As disciplinas compartilhadas são controladas com marcações específicas em cada período. Elas fazem com que os operadores genéticos considerem o conjunto de disciplinas que fazem parte de um compartilhamento como uma única. Além disso, na função de avaliação, a marcação de uma disciplina compartilhada faz com que os

*constraints* relacionados possam ser considerados, fazendo com que na evolução natural do processo as disciplinas que porventura saiam do horário do compartilhamento sejam penalizadas.

#### 4.1 Inicialização da população

A criação de uma solução (cromossomo) consiste na alocação de professores em horários válidos para as disciplinas do currículo de cada curso. Esse processo se repete na inicialização de todos os indivíduos da população inicial, alocando apenas professores aptos a ministrar as disciplinas. Diferentes estratégias podem ser utilizadas para uma alocação inicial de professores e disciplinas aos horários dos cursos, como a geração aleatória (Alves, 2017) ou o uso de heurísticas (Alsuwaylimi; Fieldsend, 2019). Para o AG proposto, a geração aleatória não seria efetiva, pois deve-se considerar as disciplinas compartilhadas, garantindo com que no mesmo horário, professor e disciplinas ofertadas em cursos diferentes sejam alocadas. Então, o uso de heurísticas foi adotado.

Heurísticas que levam em consideração os compartilhamentos foram aplicadas a geração da população inicial. As compartilhadas são geradas antes das que não são. Além disso, disciplinas compartilhadas com maior número de restrições em horários (por exemplo, apenas no sábado) ou as que contém os maiores números de cursos relacionados são geradas antes das demais. A primeira vez que uma disciplina compartilhada é gerada, seu horário e professor são aleatoriamente definidos, sendo os mesmos utilizados para a atribuição das demais disciplinas que compõem o compartilhamento.

Foram usadas ainda outras heurísticas, buscando obter uma população inicial com um maior *fitness*. No caso de disciplinas que podem ser ministradas por vários professores, a escolha prioritária é pelo professor que possui a maior diferença entre as horas já alocadas e as que deve alocar para o professor. O horário é escolhido de forma que não cause um *clash* ou fique no mesmo horário de outra disciplina já alocada no (curso, nível).

As heurísticas são aplicadas apenas em parte da população na inicialização, pois embora a qualidade média das soluções melhore com seu uso, é a diversidade de soluções aleatórias que leva a população as melhores soluções (Surry; Radcliffe, 1996).

#### 4.2 Avaliação dos indivíduos

A avaliação dos indivíduos é essencial em um AG (Eiben; Smith, 2015). A função de avaliação define o quão adaptado ao ambiente um indivíduo está, representando o quanto das restrições do problema são atendidas na solução.

Optou-se por usar uma função de avaliação baseada na atribuição de pesos e limites as *constraints* (Bettinelli, 2015). O uso de funções de avaliação com pesos é uma estratégia utilizada em Nguyen (2011), Jaengchuea e Lohpetch (2015), sendo flexível ao permitir que as *hard constraints* possuam pesos maiores do que as *soft* (Bettinelli, 2015). Com o objetivo de evitar a inversão de prioridades, um limite é definido para cada uma das *constraints* (Abdelhalim; Khayat, 2016; Alves, 2017). A Equação 1 descreve como os pesos e os limites são empregados na função de avaliação.

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

Na Equação 1,  $w_i$  é o peso da *constraint*  $i$ ,  $v_i$  é o número de violações da restrição na solução e  $l_i$  é um limite que representa o valor máximo de uma determinada *constraint*. Cada restrição contribui individualmente para o *fitness*, sendo que cada *constraint* pode variar de zero (0) ( $l_i$  violações) a  $w_i$  (não ocorrem violações). Quanto maior o valor de  $f_i$ , mais adaptado ao meio é o indivíduo (melhor é a solução).

#### 4.3 Seleção dos indivíduos

Em AGs, o processo de seleção escolhe os melhores indivíduos da população, com o objetivo de realizar o cruzamento e substituir os cromossomos com menor *fitness* (menos adaptados) (Eiben; Smith, 2015).

Neste trabalho, foram analisados os mecanismos de seleção proporcional (roleta), utilizada em Bolaji (2013), Fucilini (2008), Gozali (2020), Borges (2003), Alnowaini e Aljomai (2021) e a seleção com base em torneio, empregada em (Jaengchuea; Lohpetch, 2015; Badoni, 2014).

Segundo Shukla (2015), a melhor técnica em termos de taxa de convergência e complexidade no tempo é a seleção com torneio. O número de participantes deve ser cuidadosamente selecionado, sob pena da perda de diversidade. Foram realizados experimentos para determinar a quantidade adequada de participantes, a fim de produzir os melhores resultados. Os experimentos foram conduzidos a partir dos dados encontrados em (Lavinias, 2018) e foi definido pelo uso de torneios entre 10 indivíduos.

#### 4.4 Cruzamento e elitismo

No cruzamento ou operador de recombinação dois pais são selecionados para a geração de dois filhos na nova geração, combinando o material genético dos pais (Eiben; Smith, 2015).

Neste trabalho foram analisados o cruzamento em um ou dois pontos e o cruzamento uniforme (Sastry, 2005; Eiben; Smith, 2015). A aplicação do cruzamento em níveis com disciplinas compartilhadas gerou indivíduos com menor aptidão (*fitness*) média do que seus pais, devido a alta possibilidade de desfazer um compartilhamento. Dessa forma, foi aplicado o cruzamento em um ponto, apenas em níveis de cursos que não possuem disciplinas compartilhadas.

Na solução proposta foi utilizado o elitismo que consiste da cópia de alguns dos cromossomos com os melhores *fitness* da população atual para a próxima geração, prevenindo que boas soluções sejam perdidas, de forma similar a (Eiben; Smith, 2015; Feng, 2017; Surry; Radcliffe, 1996).

#### 4.5 Mutação

O operador de mutação é utilizado para manter a diversidade da população, tornando possível testar novas possibilidades no espaço de busca da solução dos problemas (Eiben; Smith, 2015).

Neste trabalho foram exploradas as mutações como mecanismos de trocas (Alves, 2017; Bolaji, 2013; Phuc, 2011). Ela se justifica pela composição dos genes que representam associações de professores e disciplinas em períodos de tempo predefinidos. Dessa forma, é possível que seja trocado um professor que ministra a disciplina ou que ocorra a troca de horários alocados num curso e nível.

As estratégias de mutação empregadas sofrem adicional complexidade devido a ocorrência das disciplinas compartilhadas. Nesses casos, a mutação de uma disciplina compartilhada faz com que a mudança ocorra em todos os cursos e níveis que compõem o compartilhamento. Além disso, a mudança de turno respeita os turnos de funcionamento dos cursos que fazem parte do compartilhamento.

### 5 RESULTADOS

A Universidade de Passo Fundo (UPF) contém sete *campi* em seis cidades, seis unidades acadêmicas onde funcionam aproximadamente 70 cursos de graduação. Possui ainda 15 programas de pós-graduação *stricto sensu*, seis programas de residências e diversos cursos de especialização *lato sensu*, nas modalidades presencial, híbrido ou *online*. O estudo de caso foi realizado na UPF com os cursos de graduação. O AG descrito neste artigo está sendo responsável pela geração inicial da grade de horários desde o ano de 2021. Resolver o problema neste cenário é uma excelente validação para o AG proposto.

Foram utilizados 116 cursos/currículos e 681 professores, o que gera um elevado número de cursos e disciplinas ofertadas, tornando-se um problema de grandes dimensões, necessitando uma solução computacional para resolver este *timetabling*. O *dataset* utilizado na validação contém dados utilizados no primeiro semestre de 2023.

O número de cursos usados é maior que os da IES. Tal fato ocorre devido às reformulações curriculares, onde novos currículos são introduzidos se somando aos currículos em execução. É como se novos cursos fossem criados, ampliando a quantidade de compartilhamentos entre currículos sendo descontinuados e em implantação.

O *dataset* utilizado contém 1.368 disciplinas, porém houve 1.990 ofertas destas disciplinas, onde a mesma disciplina é ofertada mais de uma vez (em mais de um curso). Destaca-se a quantidade de compartilhamentos (212), congregando 535 oferecimentos de disciplinas. Um compartilhamento tem em média 2,52 cursos, sendo que o maior agrupamento possui 11 cursos. Os compartilhamentos representam em torno de 25% dos oferecimentos de disciplinas, demonstrando a importância desta funcionalidade.

### 5.1 Constraints, pesos e limites

Foram utilizadas 10 *constraints*, sendo quatro consideradas *hard* e seis *soft*. A função de avaliação (Equação 1) utiliza pesos e limites, detalhados na Tabela 2.

**Tabela 2:** *Constraints* atuais com seus pesos e limites.

| <i>Hard Constraint</i> | Peso    | Limite | <i>Soft Constraint</i> | Peso  | Limite |
|------------------------|---------|--------|------------------------|-------|--------|
| Oferecimento           | 500.000 | 2.000  | Horas Dia              | 2.000 | 2.000  |
| Não Compartilhado      | 300.000 | 2.000  | Máximo Horas           | 2.000 | 2.000  |
| <i>Clash</i>           | 200.000 | 2.000  | Turnos Excesso         | 200   | 200    |
| Indisponibilidade      | 100.000 | 40.000 | Prioridade             | 200   | 200    |
|                        |         |        | Horas Professor        | 20    | 4.000  |
|                        |         |        | Excesso Dias           | 20    | 200    |

Fonte: Autores.

Os pesos de cada *constraint* foram atribuídos de acordo com a relevância do que está sendo analisado. O oferecimento de todas as disciplinas ocorre por padrão no AG, porém, deixar uma disciplina de fora da grade é um grave problema. Em seguida o compartilhamento de disciplinas, a quantidade de *clashes* e a indisponibilidade do professor em um dia/turno entre as *constraint* do tipo *hard*. As do tipo *soft*, recebem pesos menores pois são responsáveis por ajustes pontuais. Os limites foram definidos buscando valores que não gerem uma contribuição negativa no *score* do indivíduo.

### 5.2 Análise e discussão dos resultados

Foram realizados diferentes testes com taxas de inicialização com heurísticas, métodos para a seleção de indivíduos, formas de mutação e *crossover*. Estes testes foram realizados com pequena população (500) e baixo número de gerações (1.500). A partir deles foi possível determinar as melhores escolhas, analisando em cada caso qual foi a melhor média de *score* dos indivíduos. As opções de operadores genéticos e taxas utilizadas para a geração dos resultados apresentados foi:

- Inicialização: aleatória e heurística (20%); Taxa de elitismo: 5%;
- Seleção: Torneio com 10 competidores; *Crossover*: Um ponto (20%);
- Mutação: Troca de professor e de períodos (60%);
- População: 1.000 indivíduos; Gerações: 5.000.

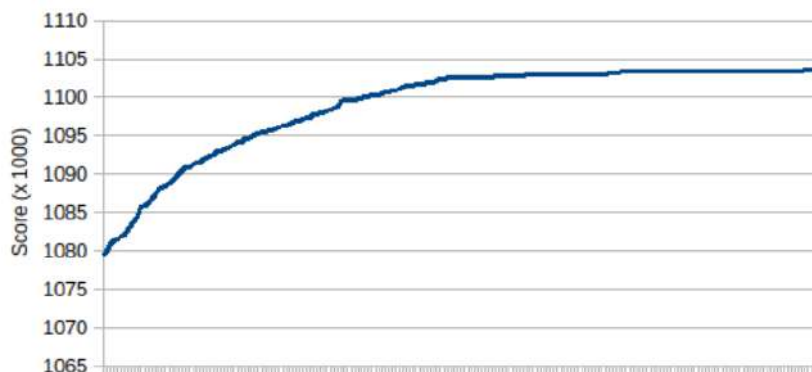
A Figura 1 contém a evolução do *score* do melhor indivíduo. Embora o AG tenha executado por 5.000 gerações, em apenas 206 houve aumento do *score* do melhor. O melhor indivíduo é gerado com *fitness* aproximado de 97,74% do máximo do *score*.

Analisando os resultados da Figura 1, pode-se perceber que quando houve alteração do *score* ocorreu o seu aumento. Uma informação implícita na Figura 1 é que dentre as 206 modificações do *score* do melhor indivíduo, em torno de 94% das modificações ocorrem até a geração 1.000, onde o número de violações das *hard constraints* alcançou quase o valor final. Pode-se perceber também que a estratégia de inicialização com heurísticas inicia a população com indivíduo(s) com alto *score*.

Ao final da execução do AG, a solução alcançou excelentes resultados (Tabela 3). Praticamente todas as *hard constraint* foram respeitadas. Restaram apenas seis casos de *clashes*. Isso representa, por exemplo uma manhã e metade de outra (poderia ser tarde ou noite). Os casos foram analisados e representavam um problema insolúvel, uma vez que havia apenas um professor apto a ministrar a disciplina, e não havia possibilidades de trocas, devido ao turno de funcionamento dos cursos envolvidos.

Os resultados apresentados na Tabela 3 confirmam a otimização realizada pelo AG. Todas as *constraint* foram reduzidas, sendo que três delas foram zeradas. Não houve a eliminação dos *clashes*, porém ocorreu uma redução

Figura 1: Evolução do Score do melhor indivíduo.



Fonte: Autores.

de aproximadamente 98,5%, reduzindo de 233 para seis casos. O oferecimento não apresentou alteração e manteve o mesmo número durante toda a evolução. Isso se explica em virtude de que os oferecimentos são gerados na inicialização e após não há mais como serem modificados. O número de professores que extrapola as 8 horas diárias também manteve-se o mesmo. Tal fato se deve a quantidade de ofertas de disciplinas com apenas um professor apto a ministrá-las (581). O *score* do melhor indivíduo alcançou 99,92% do maior possível, impulsionado pela quantidade de violações do tipo *hard* que foram minimizadas/eliminadas.

Tabela 3: Quantidade de violações de *constraints* na execução/evolução

| <i>Hard Constraint</i> | Inicial | Final | <i>Soft Constraint</i> | Inicial   | Final     |
|------------------------|---------|-------|------------------------|-----------|-----------|
| Oferecimento           | 0       | 0     | Horas Dia              | 64        | 36        |
| Não Compartilhado      | 8       | 0     | Máximo Horas           | 77        | 77        |
| <i>Clash</i>           | 233     | 6     | Turnos Excesso         | 19        | 8         |
| Indisponibilidade      | 79      | 0     | Prioridade             | 1         | 1         |
|                        |         |       | Horas Professor        | 10.906,88 | 16.779,62 |
|                        |         |       | Excesso Dias           | 332       | 285       |

Fonte: Autores

Foram alocados 184 *slots* sem a definição de um professor em 51 disciplinas diferentes. A origem desse número leva em conta como os professores são considerados aptos a ministrar uma disciplina: quando já a ministraram. Nestes 51 casos de disciplinas, o *dataset* não conta com professores aptos a ministrar essas disciplinas no momento da geração. São em geral, disciplinas novas nos currículos onde os docentes irão ter de ministrá-las pela primeira vez ou que ainda serão contratados. Além disso, ocorre uma alocação sem a definição de um professor quando mais de uma turma da mesma disciplina ocorre no mesmo horário e não há mais de um professor apto a assumi-la. O fechamento da grade dos cursos é imprescindível, pois pode influenciar na contratação dos professores.

## 6 CONCLUSÃO

Neste trabalho foi descrito o projeto, a implementação, a apresentação e a discussão de resultados de um AG para resolução do CB-CTT com novas necessidades das IES. A principal funcionalidade, não encontrada nos demais trabalhos da literatura é a exploração de um *clash* controlado, usado para implementar o compartilhamento de disciplinas. Outros recursos não encontrados em trabalhos correlatos como as disciplinas EaD, sem horário definido e fechamento de grade sem indicação de professor também estão presentes na solução.

Com os resultados obtidos, pode-se concluir que o AG proposto auxilia de forma significativa no processo de alocação das grades horárias, facilitando o trabalho dos coordenadores de curso e possibilitando otimizações em relação a carga horária docente, um recurso de elevado impacto financeiro nas IES.



Como trabalhos futuros, está em desenvolvimento uma versão para alocação conjunta de cursos de graduação e de *stricto sensu*, uma vez que grande parte dos professores são os mesmos. Outra iniciativa é uma otimização do perfil dos professores (titulação e regime de trabalho) nos cursos envolvidos em processo de avaliação externa, como é o caso do Exame Nacional de Desempenho dos Estudantes (Enade).

## REFERÊNCIAS

- ABDELHALIM, E. A.; KHAYAT, G. A. E. A utilization-based genetic algorithm for solving the university timetabling problem (uga). **Alexandria Engineering Journal**, Faculty of Engineering, Alexandria University, v. 55, n. 2, p. 1395–1409, Jun 2016.
- ALNOWAINI, G.; ALJOMAI, A. A. Genetic algorithm for solving university course timetabling problem using dynamic chromosomes. *In: 2021 International Conference of Technology, Science and Administration*. Taiz University: IEEE, 2021. p. 1–6.
- ALSUWAYLIMI, A. A.; FIELDSEND, J. E. A new initialisation method for examination timetabling heuristics. *In: 2019 IEEE Symposium Series on Computational Intelligence, SSCI 2019*. Xiamen, China: IEEE, 2019. p. 1636–1643.
- ALVES. A recursive genetic algorithm-based approach for educational timetabling problems. *In: Designing with Computational Intelligence*. Cham: Springer International Publishing, 2017. p. 161–175.
- BABAEI. Using fuzzy c-means clustering algorithm for common lecturers timetabling among departments. *In: 2016 6th International Conference on Computer and Knowledge Engineering*. University of Mashhad: IEEE, 2016. p. 243–250.
- BADONI. A new hybrid algorithm for university course timetabling problem using events based on groupings of students. **Computers Industrial Engineering**, Elsevier Ltd, v. 78, p. 12–25, Dec 2014.
- BASHAB. A systematic mapping study on solving university timetabling problems using meta-heuristic algorithms. **Neural Computing and Applications**, v. 32, n. 23, p. 17397–17432, Dec 2020.
- BETTINELLI. An overview of curriculum-based course timetabling. **TOP**, v. 23, n. 2, p. 313–349, Jul 2015.
- BOLAJI. A modified artificial bee colony algorithm for post-enrolment course timetabling. *In: TAN, Y.; SHI, Y.; MO, H. (Ed.). Advances in Swarm Intelligence*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. p. 377–386.
- BORGES, S. K. **Resolução de Timetabling Utilizando Algoritmos Genéticos e Evolução Cooperativa**. Dissertação (Mestrado) — Universidade Federal do Paraná, Curitiba, 2003. 103 p. Dissertação (Mestrado em Informática).
- COOPER, T. B.; KINGSTON, J. H. The complexity of timetable construction problems. *In: BURKE, E.; ROSS, P. (Ed.). Practice and Theory of Automated Timetabling*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996. p. 281–295.
- DUNKE, F.; NICKEL, S. A matheuristic for customized multi-level multi-criteria university timetabling. **Annals of Operations Research**, Apr 2023.
- EIBEN, A.; SMITH, J. **Introduction to Evolutionary Computing**. 2. ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015. v. 33. 1–258 p. ISSN 0368-492X.
- ELS, R.; PILLAY, N. An evolutionary algorithm hyper-heuristic for producing feasible timetables for the curriculum based university course timetabling problem. *In: Second World Congress on Nature and Biologically Inspired Computing*. Kitakyushu, Japan: IEEE, 2010. p. 460–466.
- EVEN, S. e. a. On the complexity of time table and multi-commodity flow problems. *In: 16th Annual Symposium on Foundations of Computer Science*. University of California: IEEE, 1975. v. 1975-Octob, p. 184–193.
- FENG. An integer program and a hybrid genetic algorithm for the university timetabling problem. **Optimization Methods and Software**, Taylor Francis, v. 32, n. 3, p. 625–649, May 2017.
- FUCILINI. Timetabling com algoritmos genéticos: resultados, restrições e exploração do paralelismo. **Hifen**, v. 32, n. 62, p. 156–163, Dec 2008.
- GOZALI. Solving university course timetabling problem using localized island model genetic algorithm with dual dynamic migration policy. **IEEJ Transactions on Electrical and Electronic Engineering**, v. 15, n. 3, p. 389–400, 2020.
- JAENGCHUEA, S.; LOHPETCH, D. A hybrid genetic algorithm with local search and tabu search approaches for solving the post enrolment based course timetabling problem: outperforming guided search genetic algorithm. *In: 7th International Conference on Information Technology and Electrical Engineering*. Huangshan, Anhui, China: IEEE, 2015. v. 463, p. 29–34.

- LAVINAS. Experimental analysis of the tournament size on genetic algorithms. *In: 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. Miyazaki, Japan: IEEE, 2018. p. 3647–3653.
- MARIE-SAINTE, S. L. A survey of particle swarm optimization techniques for solving university examination timetabling problem. *Artificial Intelligence Review*, v. 44, n. 4, p. 537–546, Dec 2015.
- NGO, T. A genetic algorithm for multi-objective optimization in complex course timetabling. *In: Proceedings of the 2021 10th International Conference on Software and Computer Applications*. New York: ACM, 2021. p. 229–237.
- NGUYEN. Memetic algorithm for a university course timetabling problem. *In: TAN, H. (Ed.). Informatics in Control, Automation and Robotics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. p. 67–71.
- PHUC. A new hybrid ga-bees algorithm for a real-world university timetabling problem. *In: 2011 International Conference on Intelligent Computation and Bio-Medical Instrumentation*. Wuhan, China: IEEE, 2011. p. 321–326.
- RAGHAVJEE, R.; PILLAY, N. An informed genetic algorithm for the high school timetabling problem. *In: Proceedings of the 2010 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists*. New York: ACM, 2010. p. 408–412.
- RAPPOS. A mixed-integer programming approach for solving university course timetabling problems. *Journal of Scheduling*, v. 25, n. 4, p. 391–404, Aug 2022.
- SAKAL. Learning assignment order in an ant colony optimiser for the university course timetabling problem. *In: Proceedings of the Genetic and Evolutionary Computation Conference Companion*. New York: ACM, 2021. p. 77–78.
- SASTRY. Genetic algorithms. *In: Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*. Boston, MA: Springer US, 2005. p. 97–125.
- SHIR, O. M. Niching in evolutionary algorithms. *In: Handbook of Natural Computing*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 1035–1069.
- SHUKLA. Comparative review of selection techniques in genetic algorithm. *In: 2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management*. Greater Noida, India: IEEE, 2015. p. 515–519.
- SURRY, P. D.; RADCLIFFE, N. J. Inoculation to initialise evolutionary search. *In: FOGARTY, T. C. (Ed.). Evolutionary Computing*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996. p. 269–285.
- WONG. A genetic algorithm for the real-world university course timetabling problem. *In: IEEE 18th International Colloquium on Signal Processing Applications*. Selangor, Malaysia: IEEE, 2022. p. 46–50.