Um algoritmo de escalonamento de horários para IFCE-Campus Aracati baseado em preferências

Lucas Porfírio da Silva
Estudante - Instituto Federal de Educação, Ciência e Tecnologia do Ceará Campus Aracati (IFCE)
Rodovia CE-040, Km 137,1 s/n Aeroporto, CE – 62.800-000 – Aracati – CE – Brasil
lucasporfirio1997@gmail.com

Diego Rocha Lima
Orientador - Instituto Federal de Educação, Ciência e Tecnologia do Ceará Campus Aracati (IFCE)
Rodovia CE-040, Km 137,1 s/n Aeroporto, CE – 62.800-000 – Aracati – CE – Brazil
diego.rocha@ifce.edu.br

RESUMO

A construção de horários escolares é uma tarefa difícil para a maior parte das instituições de ensino, pois possui vários tipos de restrições e objetivos que variam em cada instituição. No IFCE Campus Aracati umas das restrições é quantidade de dias que o docente está presencialmente no Campus. Neste trabalho criamos uma solução com base em algoritmo guloso e heurísticas para apresentar possíveis soluções a partir de um grupo de restrições pesadas e leves. A principal restrição é a preferência dos professore por dias de trabalho. Ao final, temos um programa que monta algumas grades viáveis e às apresenta de forma organizada em arquivos aos docentes, para apreciação e avaliação dos mesmos.

PALAVRAS CHAVE. Escalonamento de horário, Algoritmo, Preferência docente.

ABSTRACT

The construction of school timetables is a difficult task for most educational institutions, as it has several types of restrictions and objectives that vary in each institution. At IFCE Campus Aracati one of the restrictions is the number of days that the teacher is in person on the Campus. In this work we create a solution based on greedy algorithm and heuristics to present possible solutions from a group of heavy and light constraints. The main restriction is the teachers preference for working days. At the end, we have a program that assembles some viable grids and presents them in an organized way in files to the professors, for their appreciation and evaluation.

KEYWORDS. Timetable schedules. Algorithm. Teacher preference.

1. Introdução

As instituições públicas de ensino superior a medida que se desenvolvem estão sujeitas a enfrentar alguns problemas como a diminuição de orçamento, condições desfavoráveis de trabalho e também equipe docente reduzida ao mínimo necessário para o funcionamento de seus cursos. Dentre estas restrições vamos destacar a última. Tais condições requerem que as organizações melhorem a criação de horários visando reduzir os custos e maximizar o uso dos recursos disponíveis, bem como garantir o bom funcionamento dos cursos, sem deixar de oferecer qualquer disciplina por falta de espaço, horário ou docente.

Neste trabalho abordamos melhoraria nas condição de trabalho da equipe docente otimizando as horas em que estão presente na instituição e focando na opção de escolha de dias para trabalhar. Esse problema na literatura é conhecido como escalonamento e horários universitários. [Abdelhalim e El Khayat, 2016] O problema de escalonamento de horários no cenário escolar/institucional, visa encontrar um método de alocação adequado para eventos em horários e salas predefinido, onde todas as restrições do problema sejam satisfeita.

Eventos incluem estudantes, professores , cursos e disciplinas abrangendo recursos da instituição e equipamentos das salas para conteúdos práticos e teóricos.Além disso horários incluem dois componentes principais, horários diários e horários semanais, que variam entre instituições.

Contudo cada sala de aula possui particularidades como projetor de vídeo, número de assentos e quantidade de quadros disponíveis necessários para cada evento. [Babaei et al., 2015] Os recursos da Instituição são a junção de sala e horário, o qual deve ser alocado para uma turma especifica ao final do escalonamento dos horários.

O problema de escalonamento de horários já é complexo levando em consideração apenas as restrições citadas acima, porém a abordagem demonstrada nesse trabalho conta com restrições específicas como as preferências dos professores para os dias trabalhados e disciplinas ministradas.

O problema de escalonamento de horários faz parte do conjunto NP-difícil de otimização combinatória híbrida. [Mahiba e Durai, 2012] [Abbaszadeh e Saeedvand, 2014] [Garey e Johnson, 2002] A questão ocorre no inicio de cada semestre nas escolas e universidades, incluindo a alocação de eventos em uma certa quantidade de horários e salas fixa. Neste problema devemos satisfazer restrições duras e suaves durante a alocação dos eventos nos recursos, que possivelmente obtenham uma total satisfação das restrições para aumentar e fomentar a qualidade das tabelas de horários geradas. [Redl, 2004] [Asmuni, 2008] [Obit, 2010]

Restrições duras devem ser satisfeitas totalmente para gerar uma solução factível sem conflitos. Restrições suaves estão relacionadas a qualidade da solução, que quando maximizada gera a melhor solução factível. Diferente das restrições duras, as restrições suaves podem ser quebradas sem comprometer a factibilidade da solução.

O problema no Instituto Federal do Ceará Campus Aracati se dá quando os professores são dedicação exclusiva, porém residem em outras cidades, o que faz com que seja melhor aproveitar ao máximo os dias em que estão na cidade na qual trabalham. Considerando ainda que alguns docentes dão aula em programas de pós-graduação ou em outros campus que ficam localizados em cidades diferentes.

No geral os professores cumprem toda sua carga horária semanal de trabalho em 3 dias consecutivos no campus. Não que isso seja uma regra, mas de acordo com o escalonamento de horários isso caba sendo possível e viável. O desafio é alinhar todos os eventos e recursos de modo a não infringir as restrições duras do algoritmo e garantir parte das restrições leves. Dessa forma, nosso algoritmo possibilita a criação de cenários(grade de horários) diferentes para cada professor da instituição. Atualmente o campus faz uso de um software para criação da grade, mas que ao

final algumas mudanças pontuais são feitas de forma manual com o intuito de garantir o melhor aproveitamento de dias e horários para cada professor no campus, de acordo com suas disciplinas. O algoritmo aqui apresentado consegue mostrar de forma mais eficiente algumas opções possíveis de grades de horários do semestre atual, distribuindo de maneira eficiente os dias e horários do professor no campus. Fazendo assim que os professores possam analisar mais de uma opção factível, trabalho que antes seria feito manualmente.

Este trabalho está dividido da seguinte forma. Na seção 2 é apresentada a fundamentação teórica e na 3 demonstramos alguns trabalhos relacionados na qual se apresentam outras abordagens para a solução do problema de escalonamento de horários. Já na seção 4 a metodologia é apresentada, onde descrevemos os métodos e heurísticas utilizadas em nosso algoritmo. Na seção 5 estão os resultados obtidos com a implementação do algoritmo. Por fim a seção 6 traz a conclusão de nosso trabalho.

2. Fundamentação Teórica

O problema de escalonamento de horários é um problema de otimização combinatória NP-difícil. [Mahiba e Durai, 2012] No problema abordado a qualidade da solução proposta é mensurada a partir da quantidade de restrições leves respeitadas e a viabilidade da solução somente caso nenhuma restrição pesada seja infringida. [Asmuni, 2008]

O problema de escalonamento de horários se baseia em atribuir uma dupla ou tupla factível de variáveis que otimiza um conjunto de métricas e indicadores com o objetivo de minimizar as diferenças de tempo, maximizar a utilização de espaço minimização de custos relevantes ao uso de recursos é difícil encontrar um algoritmo eficiente para esses problemas, especialmente quando o problema cresce.

Boas soluções podem ser fornecidas quando o problema é melhor compreendido em termos de complexidade ou simplicidade. [Abdelhalim e El Khayat, 2016]. No contexto do IFCE Campus Aracati o problema se mostra de pequena escala no quesito tamanho da instância, se comparado a outras universidades ou institutos. Mas se mostra de complexo quando se trata da quantidade de restrições.

Os parâmetros selecionados para este trabalho incluem as preferências dos professores para dias de trabalho e dias nos quais não podem estar no campus, visando atribuir da melhor forma suas disciplinas e horários. Otimizando o tempo de permanência em sala dos professores permite que a carga horária seja cumprida de forma integral nos dias em que estiver no campus.

3. Trabalhos Relacionados

No trabalho *A Utilization-based Genetic Algorithm for Solving the University Timetabling Problem (UGA)* [Abdelhalim e El Khayat, 2016] é focado a minimização dos espaços vagos de tempo entre as aulas, a maximizar a utilização dos recursos e minimizar os custos de cada recurso para viabilidade é utilizada a restrição de não haver duas aulas usando o mesmo recurso, a capacidade da sala e o número de estudantes matriculados. São usadas preferências do professor para os recursos como restrição leve.

No trabalho *Solving a Capacitated Exam Timetabling Problem Instance Using a Bi-objective NSGA-II* [Leite et al., 2015] os autores resolvem uma instância de 80 cursos em 5 programas de graus diferentes focando em reduzir o tempo de geração dos horários da universidade substituindo a abordagem manual por uma automatizada, o que aumentou as inscrições de estudantes de outros cursos nas grades compartilhadas.

No trabalho *Multi-objective Evolutionary Algorithms for Resource Allocation Problems* [Datta et al., 2007] utiliza o algoritmo NSGA-II resolvendo duas instâncias diferentes com o mesmo

algoritmo tendo duas instâncias e dois conjuntos de restrições distintos. Na primeira busca-se minimizar a quantidade de espaços livres entre as aulas dos alunos e maximizar a quantidade de espaço entre as aulas de cada professor. Na segunda instância maximizar o retorno econômico, minimizar as emissões de carbono e minimizar a erosão do solo.Neste trabalho são utilizadas várias restrições duras.

Nesse paper *Solving University Timetabling Problems Using Advanced Genetic Algorithms* [Kazarlis et al., 2005]. Foi utilizado um algoritmo genético para resolver o problema com um método de representação indireta, usando dois algoritmos um para classificação da instância baseado no sistema de prioridades e outro para ”escalar”os horários. A solução foi aplicada em uma universidade real e supera em muito a solução feitas a mão.

Solving Class Timetabling Problem of IIT Kanpur using Multi-Objective Evolutionary Algorithm [Datta et al., 2006] neste trabalho expositivo os autores focam em defender a utilização de algoritmos genéticos para geração das escala de horários para o contexto geral do problema e defende que em todas as universidades os problemas são os mesmos e que nas instituições de ensino o trabalho de gerar os horários se mostra uma atividade laboral.

Os trabalhos citados nessa seção ressaltam a importância de se encontrar um método de geração de horários que satisfaça diversos requisitos e restrições tanto dos docentes quanto dos discentes. Isso em um tempo de computação viável. Dependendo do método, e geração de um horário pode consumir bastante tempo e memória.

Devido a complexidade do problema e por ser altamente combinatório, os trabalhos que abordam metaheurísticas ou modelos matemáticos para esse problema costumam apresentar meios de não deixar a solução tão custosa em termos do tempo de execução e uso de memória. Uma dos métodos seria descartar soluções inviáveis, ao invés de testar todas as possibilidades.

Com base nessas informações, nosso algoritmos tenta trazer soluções viáveis em um tempo de computação satisfatório e sem problemas quanto ao uso de memória e tempo de processamento. Por isso em nosso trabalho focamos nas preferências dos professores para os dias de trabalho e disciplina ministradas, além de manter a viabilidade das soluções propostas.

4. Metodologia

Esta seção é reservada para explanar acerca da metodologia usada na geração da solução proposta. Descrevemos como estão organizados os dados, bem como foi implementado o algoritmo de acordo com todas as restrições.

4.1. Organização dos dados

Os dados do trabalho se dividem em 3 partes turmas, professores e disciplinas. Iremos apresentar em detalhes nas subseções especificas.

4.1.1. Turmas

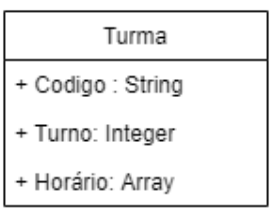


Figura 1: Fluxograma *shift* 1. Fonte: Autoral

Na entrada de dados da turma está presente o turno e o nome da turma, composto pela sigla e o semestre. Na saída dos dados está contido o turno e o nome e um vetor de 20 posições que representa as horas aula da turma. O vetor de saída de horário das turmas contem vinte posições é dividido em 5 grupos de 4 posições cada onde cada grupo representa um dia da semana. No horário do professor está contido o código da disciplina e o índice do professor usado para realizar trocas.

4.1.2. Professores

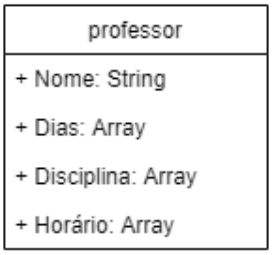


Figura 2: Fluxograma *shift* 1. Fonte: Autoral

Na entrada dos dados temos o Nome do professor, um vetor de disciplinas a ser alocado e um vetor de dias selecionados pelo professor. Na saída dos dados dos professores os dados de entrada estão presentes acrescidos de um vetor de 60 posições das horas aula do professor, pois os professores podem ministrar aulas nos 3 turnos da instituição.

No Campo do professor também existe um campo para indicar os dias do professor sub-divididos em: tipo 1 pode e quer ser alocado nesse dia, tipo 2 pode ser alocado nesse dia mas não prefere e tipo 3 não pode ser alocado nesse dia.

Os horários dos professores se dividem em 3 grupos de 20 posições onde cada seção representa um turno, e cada seção é sub-secionado em 5 grupos de 4 posições onde cada subseção representa um dia da semana

turno*20+dia*4+h, sendo h a célula a, b, c ou d de um dia de aula.

No horário do professor está contido o código da disciplina e o índice da turma usado para realizar trocas. Para fazer a equivalência entre um horário de turma que é representado por um vetor de 20 posições e o horário de professor que é representado por um vetor de 60 posições. É realizada uma operação utilizando o turno representado como turno*20+índice da posição

4.1.3. Disciplinas

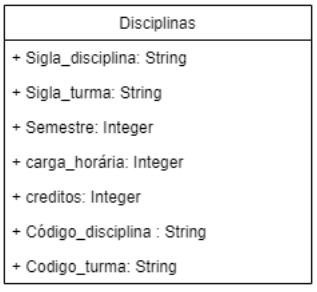


Figura 3: Fluxograma *shift* 1. Fonte: Autoral

A lista de disciplinas contém dados das disciplinas a serem alocadas. Sendo eles: sigla, sigla da turma, semestre, carga horária, créditos da disciplina, código da disciplina e código da turma

4.2. Algoritmo para alocação de horários

Esta seção abrange os métodos e algoritmos utilizadas para gerar soluções viáveis de acordo com as restrições dos professores.

4.2.1. Maior quantidade de horas primeiro

Nesta parte utilizamos um método *First-Fit* para ordenar os professores inicialmente. A ordem de atribuição depende da quantidade de horas do professor. A condição de desempate para o método foca na maior diversidade de disciplinas diferentes.

Por exemplo 2 professores diferentes ambos com 120 horas o professor que alocar mais disciplinas tem prioridade.

4.3. Busca Local

Para a busca local, utilizamos dois métodos que vamos chamar de shift 1 e shift 2 As operações de *shifts* implementados são realizados quando um dos horários estão livres seja a turma ou o professor. Todas as atribuições e trocas são realizadas em dias que o professor selecionou.

As trocas são selecionadas a partir do dia mais ocupado entre todos os professores, pois notamos que ao realizar trocas para dias mais ocupados resulta em melhores soluções.

A intenção dessa operação é reduzir a necessidade de troca a longo prazo,reduzir o tempo ocioso e maximizar a ocupação de horários nos dias selecionados...

4.3.1. Shift 1

Uma troca realizada no horário de um professor especifico entre as turmas do mesmo como visto na figura 4 e no algoritmo 1.

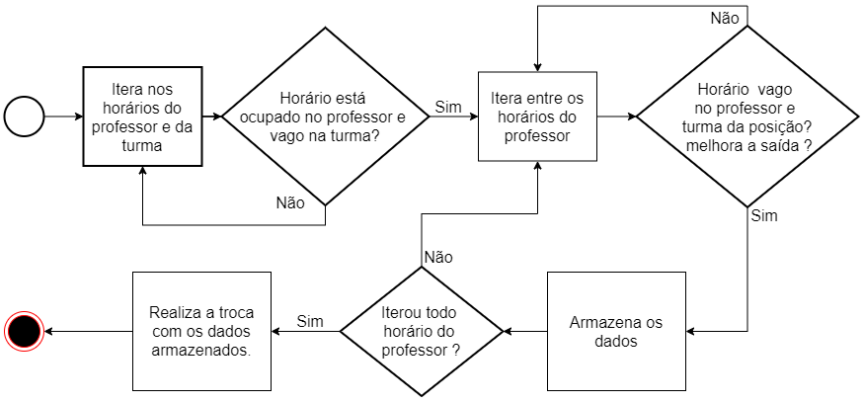


Figura 4: Fluxograma *shift* 1. Fonte: Autoral

4.3.2. Shift 2

Uma troca dentro do horário de dois professores de uma turma especifica como visto na figura 5 , no algoritmo 2 e ilustrado na figura 6

4.3.3. Função MAIN do Algoritmo

Inicialmente o algoritmo itera entre os horários do professor em dias de tipo 1, na segunda fase itera nos executa a rotina dos *shifts* e a terceira fase itera nos horários do professor em dias tipo 2, como ilustrado na figura 7

Algoritmo 1: Shift 1

```
dias=[0,1,2,3,4]
horas=[0,1,2,3]
ocupacaoatual = -1 for dia in professor.dias do
    if diasselecionados[dia] == 1 then
        for h in horas do
            pos = dia*4+h
            if professor.horario[turno*20+pos] != "" and
            turma.horario[pos] == "" then
                for dia2 in dias do
                    if diasselecionados[dia2] == 1 then
                        for h in horas do
                            pos2 = dia2*4+h
                            if professor.horario[turno*20+pos2] == "" then
                                if turma2.horario[pos2] == "" then
                                    if ocupacaoatual ¿ ocupacao[pos2] then
                                        armazena dados
            shift2()
```

Substitui os dados nas posições armazenadas.

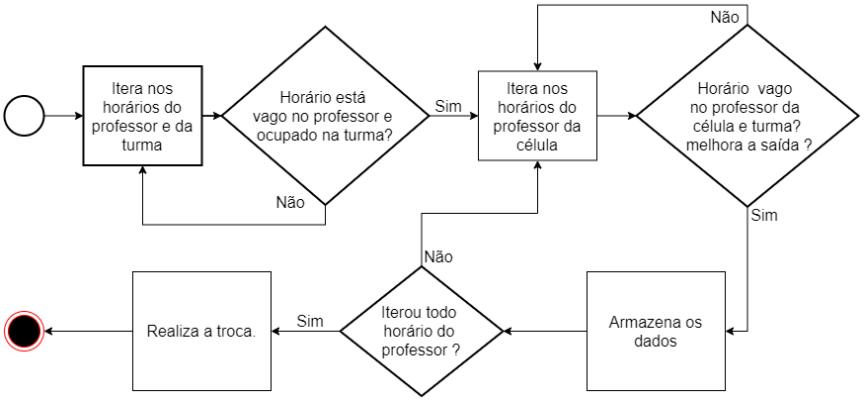


Figura 5: fluxograma *shift* 2. Fonte: Autoral

Algoritmo 2: Shift 2

```
dias=[0,1,2,3,4]
horas=[0,1,2,3]
ocupacaoatual = -1 for dia in dias do
    if professor.diasselecionados[dia] == 1 then
        for h in horas do
            pos = dia*4+h
            shift1();
            if professor.horario[turno*20+pos] == "" and
            turma.horario[pos] != "" then
                for dia2 in dias do
                    if professor2.diasselecionados[dia2] == 1 and
                     professor.diasselecionados[dia2] == 1 then
                        for h in horas do
                            pos2 = dia2*4+h
                            if professor2.horario[turno*20+pos2] == ""and
                             turma.horario[pos2] == "" then
                                if ocupacaoatual ¿ ocupacao[pos2] then
                                    armazena dados
Substitui os dados nas posições armazenadas.
```

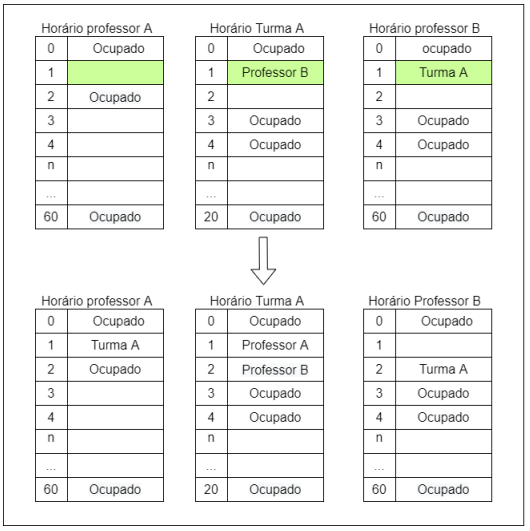


Figura 6: Representação do *shift* 2. Fonte: Autoral

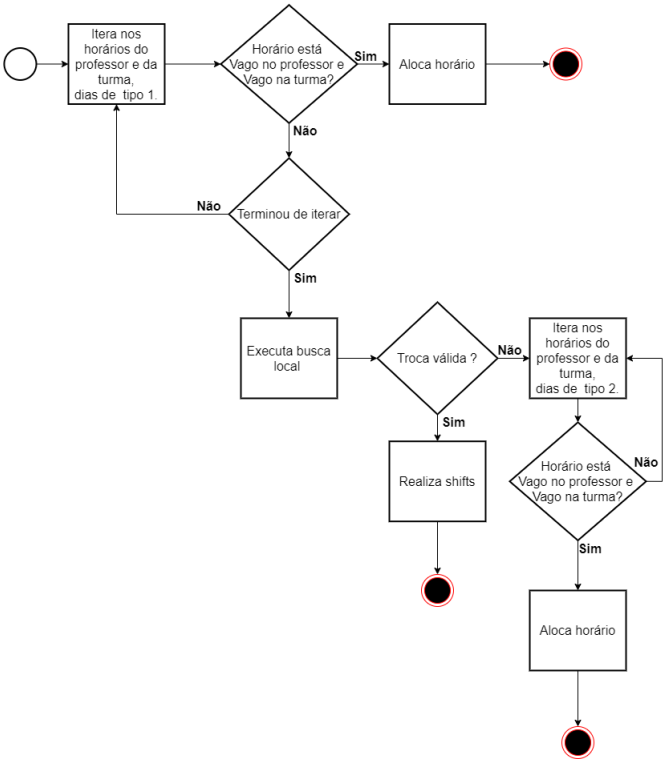


Figura 7: Representação do algoritmo no geral. Fonte: Autoral

5. Testes e Resultados

A linguagem escolhida para a implementação foi o *python* por questão de conhecimento e para usar o *colab* [1], plataforma disseminada no processamento de dados.

Como resultado obtivemos conjuntos de soluções satisfatórios dando uma margem para a discussão entre coordenação e professores na escolha da grade de horários. Também reduzimos a quantidade de tempo necessário para gerar essa solução. Assim as reuniões realizadas no início do semestre para definição dos horários podem ser otimizadas. Isso comparando com o tempo que se leva para construir grades de todas todas as turmas e professores de maneira manual.

Inicialmente os testes foram realizados e apresentados ao eixo de informação e comunicação, comparando-se o a grade utilizada no semestre letivo 2020.1, com as grades geradas pelo nosso sistema. Vale salientar que esse semestre é anterior ao semestre que está em andamento em nosso campus, 2020.2. Essas informações estão contidas na tabela 1.

Após atestado o funcionamento e a qualidade da solução pretende-se expandir a instância para todos os eixos/cursos do Campus.

Professores são aqueles que ministram as disciplinas. Professores de eixo comum ministram disciplinas nesse eixo e nos outros eixos da instituição. Professores exclusivos do eixo ministram disciplinas especificas desse eixo. Disciplinas são parte do curso em semestres específicos. Turmas dividem-se semestre e disciplinas.

Na figura 8 temos uma representação da grade de horários onde cada célula representa um horário, agrupado em pares que são separados pelo tempo de intervalo entre as aulas, dessa

[1] https://colab.research.google.com

Dado	Quantidade
Professores (total)	22
Professores (comum)	9
Professores (do eixo)	13
Disciplinas	43
Turmas	8
duração em horas(por semestre)	400

Tabela 1: dados da instancia de teste

Curso: Ciência da computação
Semestre: 1/tarde

	Segunda-feira	Terça-feira	Quarta-feira	Quinta-feira	Sexta-feira
A					
B					
C					
D					

Figura 8: Representação de horário por curso e turma Fonte:Autoral

forma temos um horário exclusivo para cada turma representadas por uma matriz [5 X 4], dessa forma facilita a identificação de possíveis trocas e alocações horário, evitando as restrições duras da solução.

Com essa quantidade de professores, disciplinas e turmas, chegar em uma solução viável e que agrade a maioria dos professores pode ser trabalhoso e demorado. Mesmo assim estamos apenas na instância de teste que representa apenas uma parte dos cursos do campus.

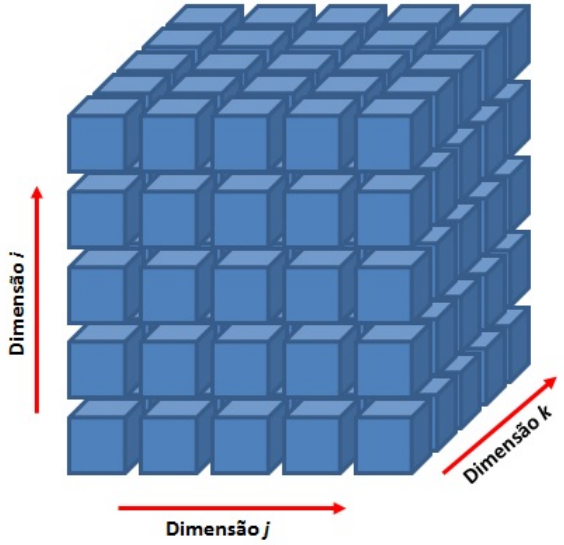


Figura 9: Representação dos horários em cada turno. Fonte: Autoral

A figura 9 representa uma matriz de três dimensões [I X J X K]:

- J(eixo X) representa a quantidade de dias de aula;
- I(eixo Y) representa a quantidade máxima de eventos por turno;
- K(eixo Z) representa a quantidade de turmas alocadas.

Na instância de teste temos uma matriz [5 X 4 X 11] pois a instância de teste considera apenas o eixo da informação e comunicação que possui 8 (oito) turmas no de Bacharelado em Ciência da Computação(BCC) e 3 (três) turmas Técnico em Informática(TI)

Relembrando que uma das restrições duras, mais óbvias, requer que nenhum professor esteja alocado para dois eventos no mesmo horário. A figura 9 mostra que todos os pontos [I,J,K] variando K e mantendo os mesmos J e I, sejam professores diferentes.

Na instância geral do problema temos uma matriz [5 X 4 X 24] sendo que existem salas do campus que cumprem funções especificas como laboratórios de informática,química ,aquicultura e hotelaria. Com usos específicos exclusivos de algumas disciplinas e cursos.

Ao final da execução, o programa converte os resultados e cria dois arquivos em Portable Document Format(PDF). Um com o horário individual de cada professor como mostrado na figura 10 e outro com os horários da cada turma, como mostrado na figura 11. Nesta última, cada disciplina vem acompanhada do nome do respectivo docente, suprimido neste exemplo com uma tarja preta. Para que outros horários sejam gerados, utilizamos uma estratégia *multi-start* e alteramos a ordem dos professores na base de dados, mantendo-os em seus respectivos dias e disciplinas.

X	Segunda-feira	Terça-feira	Quarta-feira	Quinta-feira	Sexta-feira
07:15-08:15					
08:15-09:15					
09:25-10:25					
10:25-11:25					
13:15-14:15			Comp - (80H) - BCC	LP1 - (80H) - BCC	
14:15-15:15			Comp - (80H) - BCC	LP1 - (80H) - BCC	
15:25-16:25			Comp - (80H) - BCC	LP1 - (80H) - BCC	
16:25-17:25			Comp - (80H) - BCC	LP1 - (80H) - BCC	
18:30-19:20					ED - (80H) - BCC
19:20-20:10					ED - (80H) - BCC
20:20-21:10			TC - (80H) - BCC	ED - (80H) - BCC	TC - (80H) - BCC
21:10-22:00			TC - (80H) - BCC	ED - (80H) - BCC	TC - (80H) - BCC

Figura 10: Representação dos horários de um professor. Fonte: Autoral

X	Segunda-feira	Terça-feira	Quarta-feira	Quinta-feira	Sexta-feira
18:30-19:20	BD2 - (80H) - BCC - [redacted]	EP - (40H) - BCC - [redacted]	CN - (80H) - BCC - [redacted]	RC2 - (80H) - BCC - [redacted]	POO - (80H) - BCC [redacted]
19:20-20:10	BD2 - (80H) - BCC - [redacted]	EP - (40H) - BCC - [redacted]	CN - (80H) - BCC - [redacted]	RC2 - (80H) - BCC - [redacted]	POO - (80H) - BCC [redacted]
20:20-21:10	AA - (40H) - BCC - [redacted]	BD2 - (80H) - BCC - [redacted]	CN - (80H) - BCC - [redacted]	RC2 - (80H) - BCC - [redacted]	POO - (80H) - BCC [redacted]
21:10-22:00	AA - (40H) - BCC - [redacted]	BD2 - (80H) - BCC - [redacted]	CN - (80H) - BCC - [redacted]	RC2 - (80H) - BCC - [redacted]	POO - (80H) - BCC [redacted]

Figura 11: Representação dos horários de uma turma,dados pessoais ocultos. Fonte: Autoral

6. Conclusões e Trabalhos Futuros

Com base em tudo que foi demonstrado neste artigo o trabalho foi considerado promissor pelos docentes. Foi possível atingir nossos objetivos, gerando grades de horários viáveis de acordo com as restrições do professores quanto a sua disponibilidade de dias e preferência por disciplinas. Foi possível apresentar também nossos resultados aos professores do Eixo de Informática e receber uma avaliação dos mesmos onde em sua maioria, consideraram que os horários montados são satisfatórios.

Assim, podemos concluir que o trabalho pode trazer alguns benefícios ao eixo de informática e ao campus como um todo:

- Redução do tempo e quantidade de reuniões para geração dos horários(feito antes do início de cada semestre);
- Possibilidade dos professores avaliarem mais de um cenário para seus horários;
- Melhor aproveitamentos dos horários do professor no campus, nos dias de trabalho;
- Como as grades são geradas de forma instantânea, os professores também tem a possibilidade de verificar como ficariam seus horários se ministrassem outras disciplinas;

Como trabalhos futuros pretendemos ampliar o uso do programa para toda a instituição. Tornar o sistema mais robusto, permitindo o uso em outros campus do IFCE e demais instituições de ensino. Pretendemos ainda adicionar mais restrições, solicitadas pelos docentes como: não ter uma mesma disciplina alocadas em 4 horários consecutivos, horário livre em algum dos turnos e adicionar as disciplinas optativas que podem ser ministradas no contra-turno. Outro objetivo futuro é deixar a plataforma mais usual, de forma que os coordenadores possam usar e simular.

Referências

Abbaszadeh, M. e Saeedvand, S. (2014). A fast genetic algorithm for solving university scheduling problem. *IAES International Journal of Artificial Intelligence*, 3(1):7–15.

Abdelhalim, E. A. e El Khayat, G. A. (2016). A utilization-based genetic algorithm for solving the university timetabling problem (uga). *Alexandria Engineering Journal*, 55(2):1395–1409.

Asmuni, H. (2008). *Fuzzy methodologies for automated university timetabling solution construction and evaluation*. PhD thesis, University of Nottingham.

Babaei, H., Karimpour, J., e Hadidi, A. (2015). A survey of approaches for university course timetabling problem. *Computers & Industrial Engineering*, 86:43–59.

Datta, D., Deb, K., e Fonseca, C. M. (2006). Solving class timetabling problem of iit kanpur using multi-objective evolutionary algorithm. *KanGAL, Report*, 2006006:1–10.

Datta, D., Deb, K., e Fonseca, C. M. (2007). Multi-objective evolutionary algorithms for resource allocation problems. In *International Conference on Evolutionary Multi-Criterion Optimization*, p. 401–416. Springer.

Garey, M. R. e Johnson, D. S. (2002). *Computers and intractability*, volume 29. wh freeman New York.

Kazarlis, S., Petridis, V., e Fragkou, P. (2005). Solving university timetabling problems using advanced genetic algorithms. *GAs*, 2(7):8–12.

Leite, N., Neves, R., Horta, N., Melício, F., e Rosa, A. C. (2015). Solving a capacitated exam timetabling problem instance using a bi-objective nsga-ii. In *Computational Intelligence*, p. 115–129. Springer.

Mahiba, A. A. e Durai, C. A. D. (2012). Genetic algorithm with search bank strategies for university course timetabling problem. *Procedia Engineering*, 38(0):253–263.

Obit, J. H. (2010). *Developing novel meta-heuristic, hyper-heuristic and cooperative search for course timetabling problems*. PhD thesis, University of Nottingham.

Redl, T. A. (2004). A study of university timetabling that blends graph coloring with the satisfaction of various essential and preferential conditions. Technical report.