

UNIVERSIDADE FEDERAL DE JUIZ DE FORA  
INSTITUTO DE CIÊNCIAS EXATAS  
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

# Utilização de algoritmos inteligentes para refinamento da grade de horários do Departamento de Ciência da Computação

Héber Ferreira Marques

JUIZ DE FORA  
JULHO, 2017

# Utilização de algoritmos inteligentes para refinamento da grade de horários do Departamento de Ciência da Computação

HÉBER FERREIRA MARQUES

Universidade Federal de Juiz de Fora

Instituto de Ciências Exatas

Departamento de Ciência da Computação

Bacharelado em Ciência da Computação

Orientador: Luciana Conceição Dias Campos

JUIZ DE FORA

JULHO, 2017

UTILIZAÇÃO DE ALGORITMOS INTELIGENTES PARA  
REFINAMENTO DA GRADE DE HORÁRIOS DO  
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Héber Ferreira Marques

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS  
EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTE-  
GRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE  
BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

Luciana Conceição Dias Campos  
Doutor em Engenharia Elétrica

Lorenza Leão Oliveira Moreno  
Doutor em Informática

Saulo Moraes Villela  
Doutor em Engenharia de Sistemas e Computação

JUIZ DE FORA  
06 DE JULHO, 2017

## Resumo

A alocação de uma grade de horários consiste em agrupar diversos elementos, atendendo a diferentes restrições do problema a ser solucionado. Neste trabalho realizou-se a alocação da grade horária dos cursos presenciais do Departamento de Ciência da Computação da Universidade Federal de Juiz de Fora, respeitando as restrições de cada grade de curso, o horário de trabalho dos professores e o oferecimento de disciplinas para cursos externos. Para realizar a alocação das grades de horários foram utilizados as técnicas Otimização de Colônia de Formigas Artificiais (*Ant Colony Optimization*) e Otimização por Busca Harmônica (*Harmony Search Optimization*). O algoritmo proposto obteve bons resultados, conseguindo alcançar soluções viáveis para auxiliar os responsáveis pelo plano departamental do DCC.

**Palavras-chave:** Agendamento de Horário (*Time Scheduling*), Otimização, Colônia de Formigas Artificiais, Busca Harmônica.

## Abstract

The timetable problem consists in grouping several elements, respecting different constraints of the problem to be solved. In this work, the timetable of courses from the Computer Science Department (DCC) of the Federal University of Juiz de Fora was used. The restrictions of each course grid, the working hours of the teachers and the offering of classes for external courses were respected. In order to realize the allocation of schedules, the techniques of Ant Colony Optimization and Harmony Search Optimization were used. The proposed algorithm obtained good results, achieving viable solutions to assist the responsables for build the timetable of the DCC.

**Keywords:** Timetabling, Optimization, Artificial Ant Colony, Harmony Search.

## Agradecimentos

Agradeço a meus pais, pelo encorajamento e apoio durante toda graduação. À professora Luciana pela orientação e principalmente pela paciência, sem a qual este trabalho não se realizaria.

Aos professores do Departamento de Ciência da Computação pelos seus ensinamentos e aos funcionários do curso, que durante esses anos, contribuíram de algum modo para o meu enriquecimento pessoal e profissional.

Ao Grupo de Educação Tutorial do curso de Ciência da Computação, no qual integrei durante a maior parte de minha graduação, sendo esse o exponencial para o desenvolvimento da pesquisa que embasa o presente trabalho. Todos os getianos sempre me apoiaram e dividiram incertezas e vitórias comigo, por isso, deixo aqui o meu muito obrigado.

# Conteúdo

<b>Lista de Figuras</b>	<b>5</b>
<b>Lista de Tabelas</b>	<b>6</b>
<b>Lista de Abreviações</b>	<b>7</b>
<b>1 Introdução</b>	<b>8</b>
1.1 Motivação . . . . .	8
1.2 Justificativa . . . . .	9
1.3 Objetivos . . . . .	9
<b>2 Fundamentação Teórica</b>	<b>11</b>
2.1 Problema de Alocação de Horários ( <i>Timetabling</i> ) . . . . .	11
2.1.1 Problemas NP-Completo e NP-Difíceis . . . . .	12
2.1.2 Trabalhos Relacionados . . . . .	13
2.2 Métodos de Otimização . . . . .	15
2.2.1 Otimização por Colônia de Formigas . . . . .	15
2.2.2 Otimização por Busca Harmônica . . . . .	17
<b>3 Alocação de Horários para o Departamento</b>	<b>20</b>
3.1 Especificação do Problema . . . . .	20
3.2 Metodologia . . . . .	23
3.2.1 Otimização por Colônia de Formigas (ACO) . . . . .	24
3.2.2 Otimização por Busca Harmônica . . . . .	32
3.3 Busca Local . . . . .	35
3.4 Software do Plano Departamental do DCC UFJF . . . . .	36
<b>4 Estudo de Caso</b>	<b>43</b>
4.1 Resultados Obtidos . . . . .	44
<b>5 Conclusão e Trabalhos Futuros</b>	<b>52</b>
<b>A Anexo I - Algoritmos</b>	<b>53</b>
<b>Bibliografia</b>	<b>57</b>

## Lista de Figuras

2.1	Diagrama de Euler para o Conjunto P, NP, NP-Difícil e NP-Completo. . . . .	12
2.2	Formigas em busca de alimento. Fonte: (BONABEAU; DORIGO; THE- RAULAZ, 1999) . . . . .	16
2.3	Músicos compondo. Adaptado de: (AL-BETAR; KHADER; ZAMAN, 2012). . . . .	19
3.1	Fluxograma representativo da organização dos algoritmos. . . . .	23
3.2	Fluxograma representativo do algoritmo de formigas (ACO). . . . .	25
3.3	Fluxograma representativo do algoritmo de seleção ProximoItemPermutado. . . . .	26
3.4	Geração do vetor pseudo randômico. Adaptado de: (NOTHEGGER et al., 2012). . . . .	27
3.5	Fluxograma representativo do algoritmo de seleção da Roleta. . . . .	28
3.6	Método de geração do vetor a ser selecionado pela Roleta . . . . .	29
3.7	Fluxograma representativo do algoritmo de seleção do Torneio. . . . .	30
3.8	Método de geração do vetor a ser selecionado pelo Torneio. . . . .	30
3.9	Fluxograma da Otimização por Busca Harmônica(HSO). . . . .	33
3.10	Exemplo da geração de uma solução pela Otimização por Busca Harmônica (HSO). . . . .	34
3.11	Exemplo da seleção de horários na Otimização por Busca Harmônica (HSO). . . . .	35
3.12	Fluxograma representativo da Busca Local. . . . .	36
3.13	Modelagem do Banco de Dados utilizado. . . . .	37
3.14	Interface responsável por exibir o Plano departamental. . . . .	38
3.15	Turmas ministradas pelo docente. . . . .	39
3.16	Interface responsável por cadastrar as solicitações de vagas dos coordenadores. . . . .	40
3.17	Exibe as solicitações de vagas dos coordenadores para cada disciplina. . . . .	40
3.18	Turmas cadastradas. . . . .	41
3.19	Docentes cadastrados. . . . .	41
3.20	Disciplinas cadastradas. . . . .	42
3.21	Cursos Cadastrados. . . . .	42
4.1	Gráfico de convergência de penalizações fortes no algoritmo ACO com Ro- leta, Busca Local e HSO. . . . .	47
4.2	Gráfico de convergência de penalizações fracas no algoritmo ACO com Ro- leta, Busca Local e HSO. . . . .	48



## Lista de Tabelas

3.1	Exemplo de um período de uma grade existente em um curso noturno. . .	23
4.1	Valores dos parâmetros . . . . .	44
4.2	Combinações de algoritmos utilizados. . . . .	46
4.3	Resultados obtidos. . . . .	46
4.4	Período com violações de restrições fortes. . . . .	47
4.5	Comparação entre o resultado feito manualmente e o obtido pelo algoritmo para o curso de Ciência da Computação. . . . .	49
4.6	Comparação entre o resultado feito manualmente e o obtido pelo algoritmo para o curso de Sistema de Informação. . . . .	49
4.7	Horários definidos para o Docente P4. . . . .	50

## Lista de Abreviações

DCC	Departamento de Ciência da Computação
UFJF	Universidade Federal de Juiz de Fora
ACO	<i>Ant Colony Optimization</i>
HSO	<i>Harmony Search Optimization</i>
GA	<i>Genetic Algorithm</i>
PSO	<i>Particle Swarm Optimization</i>
BCO	<i>Bee Colony Optimization</i>
HCP	<i>Hard Constraint Penalty</i>
SCP	<i>Soft Constraint Penalty</i>
HM	<i>Harmony Memory</i>

# 1 Introdução

Na alocação de uma grade de horários realiza-se a combinação de diversos elementos, atendendo a diferentes restrições do problema a ser solucionado. Os elementos podem ser os professores, disciplinas, turmas, horários ou salas de aula. Tem-se como restrições a alocação de diferentes turmas para diferentes professores, evitar conflitos de horários entre as turmas, entre outras. A junção de todos esses fatores concebe ao problema um forte caráter combinatório. A complexidade deste tipo de problema é elevada e dificilmente encontra-se uma solução ótima.

## 1.1 Motivação

Obter uma boa solução para o problema de alocação de grade de horários é de extrema importância para as unidades de ensino. Por exemplo, alunos dependem dos horários e turnos das turmas para serem capazes de cursar as disciplinas necessárias em seu período e professores possuem determinados horários de suas preferências para ministrar suas turmas.

O Departamento de Ciência da Computação (DCC) da Universidade Federal de Juiz de Fora (UFJF) necessita solucionar esse problema todo semestre e alocar a grade horária de seus cursos. Como o DCC possui 4 cursos de graduação presenciais, elaborar o plano departamental de todas as turmas para o semestre letivo é uma tarefa complexa.

O plano departamental do DCC deve conter as turmas oferecidas pelo departamento em horários que atendam a grade dos cursos de graduação que o mesmo oferece, sendo eles Ciência da Computação Noturno e Diurno, Engenharia Computacional e Sistemas de Informação. Também são oferecidas turmas em horários específicos para atender a demanda de outros departamentos que possuem em suas grades disciplinas ofertadas pelo DCC, como por exemplo Algoritmos e Laboratório de Programação. Essas são disciplinas básicas para todos os cursos da área de ciências exatas e engenharia.

Além disso, um plano departamental pode possuir diversas outras regras, como

por exemplo atender a restrição de horário dos professores, turmas não terem quatro horas seguidas de duração, entre outras. Isso motivou o desenvolvimento um software que auxilie o departamento a obter este planejamento de forma otimizada, diminuindo o esforço de quem o elabora.

## 1.2 Justificativa

O fator combinatório gerado por todas as regras, citadas na Seção 1.1, transforma a construção do plano departamental, que quando realizado de forma manual, em uma tarefa árdua e complexa.

Analisando esse fator, gera-se a necessidade de um software focado em gerenciar o plano e auxiliar as pessoas responsáveis por construí-lo com diversas opções, com o intuito de ajudá-lo a decidir a melhor solução. Assim, o desenvolvimento do software e o uso de algoritmos inteligentes se fazem necessários e tornam-se importantes como um auxílio ao departamento.

## 1.3 Objetivos

O objetivo deste trabalho é o desenvolvimento de um software capaz de gerenciar todos os dados do plano departamental, utilizando algoritmos inteligentes para realizar a melhor alocação possível, com o intuito de auxiliar as pessoas encarregadas por elaborá-lo.

Seu foco principal ao gerar soluções auxiliares é extinguir colisões de horários e, em alguns casos, grandes janelas entre horários já definidos.

Após uma análise na literatura nota-se que este problema de alocação de grade de horários é bastante abordado, existindo inclusive concursos específicos, envolvendo diferentes abordagens para solucioná-lo. Muitas das abordagens encontradas na literatura utilizam algoritmos inteligentes para a sua resolução.

Algoritmos inteligentes são técnicas inspiradas em processos do âmbito terrestre para a resolução de problemas computacionalmente complexos. Alguns exemplos de abordagens inteligentes são Algoritmos Genéticos, Otimização por Colônia de Formigas, Otimização por Enxame de Partículas, Otimização por Busca Harmônica, entre outros.

---

As abordagens empregadas para a resolução do problema neste trabalho foram Otimização por Colônia de Formigas e Otimização por Busca Harmônica. Ambas abordagens foram selecionadas devido ao sucesso na literatura para a solução de problemas com características semelhantes ao abordado neste trabalho. Os algoritmos foram empregados de forma híbrida onde a memória harmônica trabalha utilizando as soluções encontradas pelo algoritmo de formigas, na tentativa de buscar uma melhor solução.

O trabalho está dividido em duas partes. A primeira é o desenvolvimento de uma interface amigável para a manipulação dos dados da grade de horários do DCC. Já a segunda é a aplicação das técnicas de algoritmos inteligentes para solucionar o problema da grade de horários apresentado. Tendo como objetivo o provimento de diferentes opções viáveis para um especialista tomar a decisão final.

## 2 Fundamentação Teórica

Este capítulo aborda todos os conceitos teóricos necessários para o bom entendimento do trabalho. Inicialmente, descreve-se o problema de alocação de horários apontando diferentes restrições e seus elementos combinatórios. Em seguida, é feita uma breve contextualização sobre os métodos de otimização escolhidos para resolver tal problema.

### 2.1 Problema de Alocação de Horários (*Timetabling*)

A alocação de horários é um problema de satisfação de restrições, comum em diversas áreas em que há a necessidade de se definir determinada pessoa, tarefa ou turmas em algum horário ou local específico. Tem-se que uma boa solução é aquela que ao alocar o objeto suas restrições não são violadas. Normalmente, esse problema ocorre em empresas, indústrias, escolas, esportes, hospitais, entre outros. O foco deste trabalho é a alocação da grade dos cursos do DCC da UFJF.

O problema de alocação de grade de horários em instituições de ensino possui diversas adversidades e fatores que o torna complexo e de difícil solução. Sem ajuda computacional, uma pessoa pode demorar dias para alcançar uma solução e, no final, esse pode não ser o melhor resultado. Alocar todos os recursos solicitados, satisfazendo todos os fatores e restrições envolvidos, revela-se como uma boa solução para esse tipo de problema.

No geral, a alocação de grade de horários de instituições de ensino envolve como recursos os professores, turmas, horários, estudantes e salas de aula. Assim, uma solução viável para esse problema seria a combinação dos recursos de forma a conciliar todos os envolvidos, minimizando a ocorrência de conflitos e respeitando as limitações estabelecidas para o mesmo. As restrições são classificadas em três categorias, apontadas em (SANTOS; SOUZA, 2007), descritas a seguir:

- **Restrições Organizacionais** - são as restrições específicas da universidade, como a alocação de salas de aula e o número de turmas por docente.

- **Restrições Pedagógicas** - normalmente referem-se a questões que melhoram o aprendizado dos alunos envolvidos, como, por exemplo, uma disciplina não possuir quatro horas seguidas de aula.
- **Restrições Pessoais** - levam em consideração as requisições pessoais dos envolvidos, como sua disponibilidade de horário, preferência pelas disciplinas a lecionar, entre outros.

Uma boa solução para este tipo de problema, envolvendo todos os fatores apresentados acima, é complexa e de difícil obtenção. Isso o classifica na família de problemas do tipo NP-Difícil, como demonstrado em (EVEN; ITAI; SHAMIR, 1976).

### 2.1.1 Problemas NP-Completos e NP-Difíceis

No ramo da Computação existem diversos tipos de problemas, nos quais cada um recebe uma determinada classificação conforme a complexidade para sua solução. As principais podem ser vistas na Figura 2.1, onde têm-se as classes P, NP, NP-Completo e NP-Difícil, e seus envolvimento com outras classes.

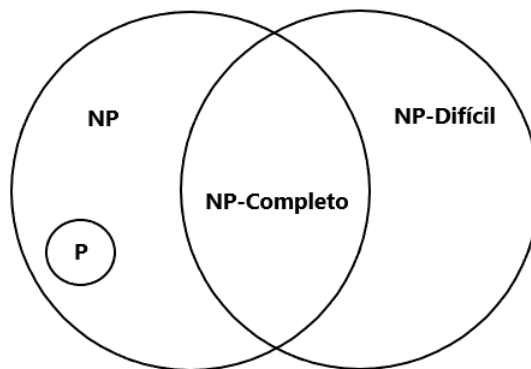


Figura 2.1: Diagrama de Euler para o Conjunto P, NP, NP-Difícil e NP-Completo.

Entre os diferentes problemas, alguns podem ser resolvidos em tempo polinomial, classificados na classe P, que são os mais fáceis de encontrar uma solução ótima. Outros são impossíveis de serem resolvidos em tempo polinomial por um algoritmo determinístico, ou seja, um algoritmo que sempre retorna a mesma solução. Esses pertencem a classe NP.

Além da classe NP, existem duas outras classes, a classe NP-Completo e a NP-Difícil, (GAREY; JOHNSON, 1979). Um problema é NP-Difícil se for tão difícil quanto

os problemas mais difíceis em NP, mas não necessariamente devem estar nessa classe. Um problema A é NP-Difícil se, e somente se, existe um problema NP-Completo B redutível em tempo polinomial para A. Caso um problema pertença a classe NP-Difícil e também a classe NP ele é classificado como NP-Completo.

Problemas NP-Completo podem ser polinomialmente reduzido a outros problemas NP-Completo, esta é uma definição concluída por (COOK, 1971), assim, caso um desses problemas possa ser resolvido em tempo polinomial, todos os outros desta mesma classe também serão. A geração de grade de horários é classificada como um problema NP-Difícil, (EVEN; ITAI; SHAMIR, 1976).

Ao longo dos anos publicam-se, na literatura, novos algoritmos para o problema de alocação de horários, na tentativa de se obter métodos eficientes para sua solução. Alguns são apresentados na próxima seção.

### 2.1.2 Trabalhos Relacionados

Existem, na literatura, diversos artigos sobre o problema de geração de horários (*timetabling*), e neste trabalho focou-se nas soluções obtidas através de algoritmos inteligentes. Tais algoritmos levam em consideração algum fator existente, como, por exemplo, a migração dos pássaros, a busca por alimento das formigas, o comportamento de neurônios, entre outros. Simulando-os com o objetivo de direcionar sua busca a sempre encontrar melhores soluções.

Entre esse algoritmos têm-se: A Otimização por Colônia de Formigas, Otimização por Busca Harmônica, *Simulated Annealing*, Algoritmo Genético, Colônia de Abelhas, Otimização por Enxame de Partículas, entre outras.

A Otimização por Colônia de Formigas, desenvolvida por (BONABEAU; DORIGO; THERAULAZ, 1999), é uma meta-heurística inspirada na busca por alimento das formigas, levando em consideração os feromônios depositados percorrer caminhos em direção ao alimento. A Otimização por Busca Harmônica, (GEEM; KIM; LOGANATHAN, 2001), leva em consideração a forma de compor novas música para gerar as soluções candidatas, nessa é feita a combinação de notas até encontrar uma melodia agradável aos ouvidos.



*Simulated Annealing* é um modelo probabilístico baseado em termodinâmica proposto inicialmente por (KIRKPATRICK; GELATT; VECCHI, 1983). Algoritmo Genético, (MITCHELL, 1996), é uma meta-heurística inspirada em (DARWIN, 1859) com o intuito de encontrar novas soluções seguindo os passos da evolução genética. Colônia de Abelhas, (PHAM et al., 2005), inspira-se na polinização das abelhas. Por fim, Otimização por Enxame de Partículas, (KENNEDY; EBERHART, 1995), é inspirada na forma de migração dos pássaros.

(FONSECA et al., 2012) utilizou a meta-heurística *Simulated Annealing*, uma busca local e sete estruturas de vizinhanças diferentes para a terceira competição internacional de *timetabling* (ITC2011, 2011). Esse método foi um dos finalistas na competição obtendo ótimos resultados para as instâncias disponibilizadas.

(EL-SHERBINY; ZEINELDIN; EL-DHSHAN, 2015) utilizou o Algoritmo Genético, em conjunto com busca local *Hill Climbing Optimization*, (DAVIS, 1991), na tentativa de diminuir a quantidade de evoluções do algoritmo genético e obter uma melhora na solução. O objetivo proposto foi alcançado em várias instâncias.

(NGUYEN; NGUYEN; TRAN, 2012) utiliza a Otimização por Busca Harmônica em conjunto com a Otimização por Colônia de Abelhas. Seu trabalho consistiu em hibridizar essas técnicas em várias instâncias reais e comparar seus resultados com outras heurísticas existentes, comprovando que esta heurística se adapta muito bem ao *timetabling*.

(AL-BETAR; KHADER; ZAMAN, 2012) hibridizou a Otimização por Busca Harmônica com o *Hill Climbing Optimization*, (DAVIS, 1991), e Otimização por Enxame de Partículas. Com a obtenção de várias soluções ótimas ou próximas às ótimas conhecidas na literatura, demonstrou que essa combinação de técnicas funcionou muito bem para o *timetabling*.

Em (NOTHEGGER et al., 2012) utilizou-se a Otimização por Colônia de Formigas, na tentativa de resolver o problema de grade de horários. Essa proposta ganhou o segundo concurso internacional de *timetabling*, (ITC2007, 2007), com ótimos resultados para as instâncias da competição.

Esses são alguns algoritmos disponíveis na literatura que alcançaram boas soluções

para o problema abordado. Esse trabalho utiliza-se desses como inspiração e duas dessas heurísticas na tentativa de obter resultados satisfatórios, sendo essas detalhadas na Seção 2.2.

## 2.2 Métodos de Otimização

Como visto na seção anterior os métodos utilizados para resolver o problema de *timetabling* são, no geral, algoritmos inteligentes. Duas dessas técnicas, a Otimização por Colônia de Formigas e Otimização por Busca Harmônica, foram as escolhidas para resolver o problema deste trabalho.

A motivação dessa escolha foi devido à proximidade das versões do problemas de alocação de horários apresentados em (NOTHEGGER et al., 2012) e (GEEM; KIM; LOGANATHAN, 2001), com o abordado neste trabalho e os resultados obtidos pelos mesmos. A hibridização dessas duas técnicas não é frequentemente encontrada na literatura, isso incentivou o uso de ambas para analisar seus comportamentos em conjunto.

### 2.2.1 Otimização por Colônia de Formigas

A otimização por colônia de formigas (ACO), (BONABEAU; DORIGO; THERAULAZ, 1999), é uma meta-heurística baseada na busca por alimento das formigas. Sempre que uma formiga carrega um alimento de sua fonte para seu ninho ela deixa um rastro de uma substância denominada feromônio. Outras formigas podem sentir essa substância e a quantidade depositada no local, funcionando como uma forma de comunicação.

Há dois meios das formigas buscarem alimento, como citado em (BONABEAU; DORIGO; THERAULAZ, 1999), a primeira é as formigas seguirem outras formigas e uma trilha de feromônios em direção a uma fonte de comida, essa é denominada recrutamento. Já a segunda é quando as formigas vão em direção à fonte de comida seguindo um rastro de feromônios depositado posteriormente, esse método é denominado recrutamento em massa, o nome vem do fato que quanto maior a quantidade de feromônios depositado no local, maior a probabilidade da formiga seguir aquele caminho.

A Figura 2.2 apresenta um experimento presente em (BONABEAU; DORIGO;

THERAULAZ, 1999) que comprova a busca de alimento das formigas. Inicialmente, as formigas possuem a mesma probabilidade de seguir qualquer um dos caminhos. Aos poucos esses indivíduos começam a transitar de forma mais rápida no menor trajeto e isso faz com que a quantidade de feromônios nesse aumente mais que no outro. Com o decorrer do tempo as formigas percebem esta diferença e optam pelo caminho menor, deixando assim sua trilha cada vez mais forte e fazendo com que mais formigas escolham esta trajetória.

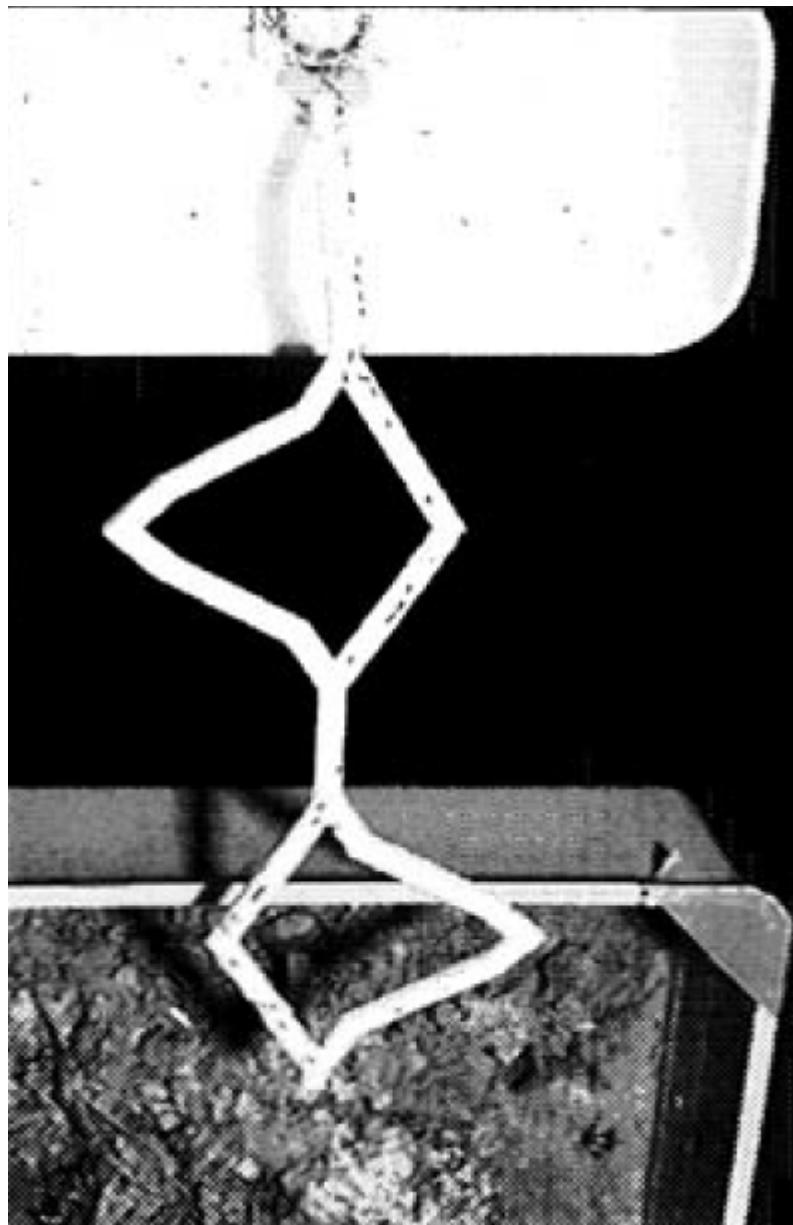


Figura 2.2: Formigas em busca de alimento. Fonte: (BONABEAU; DORIGO; THERAULAZ, 1999)

O ACO leva em consideração a busca por alimento das formigas. As “formigas

artificiais” encontram diversas soluções para o problema, essas são classificadas de acordo com a sua qualidade. Na natureza, as formigas depositam uma maior quantidade de feromônios nos caminhos mais curtos, no ACO isso ocorre de forma semelhante, mas o indicativo dos melhores caminhos é a qualidade da solução. Com o passar do tempo são priorizadas as melhores soluções encontradas devido à concentração de feromônios nesses locais, mas, assim como na vida real, há sempre a possibilidade de ser escolhido, em algum momento, uma nova solução e então encontrar um resultado melhor que os encontrados até o momento.

### 2.2.2 Otimização por Busca Harmônica

A otimização por busca harmônica (HSO) é uma heurística inspirada na forma como músicos compõem suas melodias, como, por exemplo, em uma improvisação de jazz.

Um músico inicia uma melodia de forma totalmente aleatória, feita no próprio palco da apresentação, quando está realizando uma improvisação de jazz. A partir desta melodia criada, outros músicos começam a compor melodias para seus instrumentos. Para isso, leva-se sempre em consideração as combinações de notas estarem no mesmo tom e serem agradáveis aos ouvidos das pessoas que estão na plateia. Com o tempo as novas melodias tendem a ficar cada vez melhores até alcançar um estado “ótimo”.

A técnica HSO, como demonstrado em (GEEM; KIM; LOGANATHAN, 2001), inspira-se nesse procedimento para gerar as soluções candidatas. Na improvisação de jazz é realizado um arranjo de notas para formar uma melodia. Analogamente, no HSO, os fatores do problema são combinados, representando as notas. A melodia é a solução gerada após essas associações serem finalizadas.

Para realizar essa combinação é necessário um conjunto de diferentes soluções para o algoritmo utilizar como base, denominando-o como memória harmônica (HM - *Harmony Memory*).

O HSO consiste em três etapas:

- Inicializar a memória harmônica;
- Improvisar uma nova harmonia (Solução);

- Atualizar a memória harmônica.

A memória harmônica pode ser inicializada de forma totalmente randômica ou com alguma heurística que induz as primeiras soluções serem melhores.

Os dois últimos passos são repetidos várias vezes até atingir um resultado considerado bom. Esses passos se inspiram na parte em que os músicos combinam suas ideias tentando gerar uma boa melodia e, caso isso realmente ocorra, a solução é armazenada na memória harmônica. Após ser armazenado na memória harmônica, a mesma é utilizada para gerar outras combinações.

A Figura 2.3 demonstra como seria a transformação de notas em fatores do problema. Por exemplo, o conjunto de notas (Dó, Mi, Sol) seria representado como (100, 203, 504), (Lá, Si, Sol) por (220, 400, 700), (Ré, Sol, Si) por (104, 50, 600) e assim por diante, nos quais cada número é um determinado dado representado no problema e cada conjunto é uma solução. O HSO combina esses conjuntos na tentativa de gerar melhores resultados. O conjunto (100, 203, 504) pode ser combinado com os conjuntos (220, 400, 700) e (104, 50, 600) formando uma nova solução, representada pelo conjunto (220, 50, 504).

Ambas técnicas citadas nesta Seção são utilizadas em conjunto para a resolução do problema de grade de horários dos cursos do DCC, foco deste trabalho. O algoritmo de formigas gera e otimiza as soluções iniciais e a busca harmônica é responsável por refiná-las.

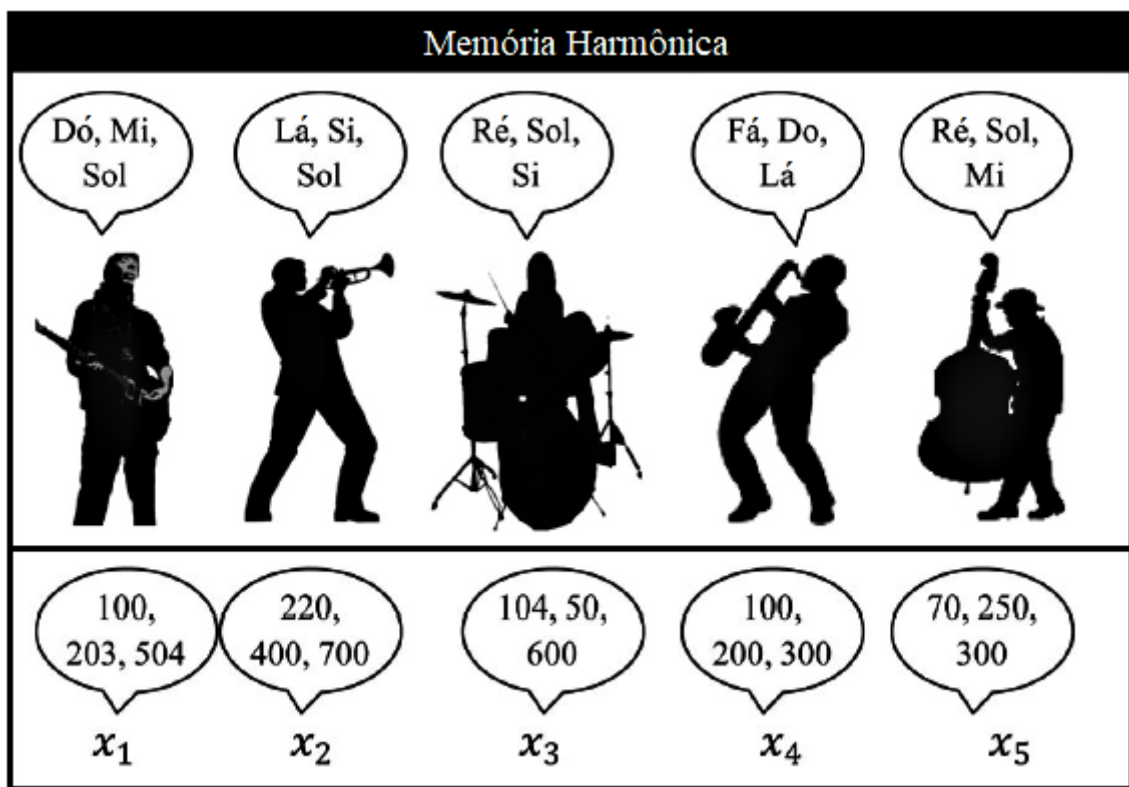


Figura 2.3: Músicos compondo. Adaptado de: (AL-BETAR; KHADER; ZAMAN, 2012).

## 3 Alocação de Horários para o Departamento

Neste capítulo é abordada a alocação de horários especificamente para o Departamento de Ciência da Computação da Universidade Federal de Juiz de Fora focando principalmente em suas diferenças em relação ao problema aplicado a outros locais. Logo após, é realizada uma abordagem detalhada dos algoritmos utilizados. Por fim, explica-se o software desenvolvido e suas diversas funcionalidades implementadas.

### 3.1 Especificação do Problema

O Departamento de Ciência da Computação da Universidade Federal de Juiz de Fora é um dos muitos existentes na universidade. Esse responsabiliza-se por gerenciar todos os cursos e disciplinas referentes a área de computação dispostos pela UFJF.

Como mencionado no Capítulo 2, existem diferentes versões do problema de alocação de horários, cada uma com suas próprias particularidades. O DCC é responsável por gerenciar quatro cursos de graduação presencial respeitando um conjunto de restrições que caracteriza sua versão. Entre as restrições deve-se respeitar a grade de todos os cursos ao selecionar horários para as turmas, atender restrições de horários dos professores, entre outros. Nessa seção são abordadas todas as peculiaridades específicas para esse problema, explicando-as e apontando a forma de organização escolhida para tal conjunto.

As restrições foram divididas em dois grupos com o intuito de auxiliar a resolução do problema. Tem-se assim o grupo das regras que não podem ser violadas (restrições fortes) e o das que preferencialmente não devem ser violadas (restrições fracas). Ao violar cada uma dessas, a solução recebe uma pontuação de penalidade, que possui valor diferente dependendo do grupo em que a restrição está associada. Uma solução ótima é aquela que não possui penalizações.

As restrições fortes devem ser atendidas sempre, pois estas visam o bom funcio-

namento do departamento. Caso não sejam atendidas, a solução não poderá ser aplicada no semestre letivo sem modificações. A seguir a listagem com as mesmas:

- Respeitar as turmas que possuem horários fixos pré-definidos;
- Levar em consideração o turno da turma ao determinar seus horários;
- Garantir que turmas que estão no mesmo período de uma determinada grade não coincidam seus horários;
- Uma turma não pode conter conflitos em seus próprios horários.

Já as leves, exibidas a seguir, têm o intuito de melhorar a solução como um todo e atender as diferentes demandas necessárias no departamento:

- Atender as restrições de horário dos professores;
- Turmas não terem quatro horas seguidas de duração;
- Não possuir dois horários às 21 horas para a mesma turma;
- Laboratórios de programação não serem às 21 horas;
- Coordenador e vice-coordenador do curso não darem aula no mesmo horário.

Dentre as restrições citadas algumas necessitam ser melhor explicadas para um melhor entendimento dos aspectos do departamento e do problema no geral.

O DCC gerencia a grade de quatro cursos de graduação presencial: Ciência da Computação diurno e noturno, Sistemas de Informação e Engenharia Computacional. Cada um possui grades específicas de disciplinas obrigatórias, ordenadas por semestre, onde, ao segui-las, o aluno finaliza o curso no tempo esperado.

Levando isso em consideração, o algoritmo não pode alocar turmas em horários que impossibilitem o aluno de cursar todas as disciplinas do semestre letivo que estiver periodizado.

Analisando as grades de todos os cursos em conjunto, existem poucos horários disponíveis para as turmas que atendem mais de um curso. Quando as turmas estão



adequadas a este requisito, devem possuir horários que respeitem todas as grades que fazem parte. Encontrar um horário que seja apto a todos os cursos é uma tarefa árdua.

Algumas turmas possuem horários fixos, essas normalmente atendem a outros cursos de outros departamentos. Mudar seus horários acarretaria em alterações na grade dos cursos externos. Esta alteração não é viável, pois isso causaria uma mudança geral em todos os departamentos afetados, o que não é desejável.

Todas as turmas referentes a disciplinas do DCC possuem turnos pré-definidos, os quais indicam para o algoritmo qual período do dia devem ser alocadas, facilitando assim, de maneira geral, a restrição citada.

Coordenadores de curso devem comparecer a determinadas reuniões para tratar assuntos normalmente relacionados à coordenação, e caso não possam comparecer, é necessário enviar seus vice para representá-los. É desejável que não haja coincidência entre os horários das turmas do coordenador e do vices, caso seja solicitado para alguma atividade da coordenação.

Alguns professores possuem restrições de horários para trabalho, pois nem todos trabalham com dedicação exclusiva. O sistema deve levar em conta este fato e tentar alocar, sempre que possível, o horário desejado desses docentes.

Após a alocação dos horários para as disciplinas obrigatórias, são alocadas na grade de cada curso as possíveis disciplinas eletivas do semestre. É desejável alocar o maior número de eletivas em horários não coincidentes com obrigatórias do mesmo período, para que os alunos possam selecioná-las.

A Tabela 3.1 mostra um exemplo de um período da grade de um curso noturno. Pode-se reparar que as turmas A e B das disciplinas DCC003 e DCC004, em itálico, estão no mesmo espaço de tempo, desta forma a solução representada possui penalidade de horário devido a violação da terceira restrição forte já citada nesta seção.

Levando em consideração todas essas restrições, foi desenvolvido um algoritmo híbrido utilizando duas técnicas inteligentes na tentativa de solucionar o problema e obter a melhor solução possível. Com isso, serve de auxílio para as pessoas responsáveis pela elaboração do plano departamental, de modo a realizar poucas ou nenhuma alteração para colocá-la em prática. Esse algoritmo é detalhado na Seção 3.2.

Tabela 3.1: Exemplo de um período de uma grade existente em um curso noturno.

	<b>SEG</b>	<b>TER</b>	<b>QUA</b>	<b>QUI</b>	<b>SEX</b>
<b>19:00</b>	DCC001 - A	DCC003 - A	DCC002 - B		DCC004 - B
<b>20:00</b>	DCC001 - A	DCC003 - A	DCC002 - B		DCC004 - B
<b>21:00</b>			DCC001 - A	<i>DCC003 - A</i> <i>DCC004 - B</i>	DCC002 - B
<b>22:00</b>			DCC001 - A	<i>DCC003 - A</i> <i>DCC004 - B</i>	DCC002 - B

## 3.2 Metodologia

Nesta seção é apresentado um algoritmo híbrido utilizando em conjunto as técnicas de Otimização por Colônia de Formigas e Otimização por Busca Harmônica.

No fluxograma (Figura 3.2) é apresentado como foram utilizados os algoritmos de forma conjunta. O intuito é o ACO ser o primeiro a executar e retornar suas melhores soluções encontradas. Após o termino, a Memória Harmônica é inicializada com esses resultados. Por fim, o HSO é executado, e a Memória Harmônica o direciona, diminuindo assim o tempo para encontrar melhores soluções.

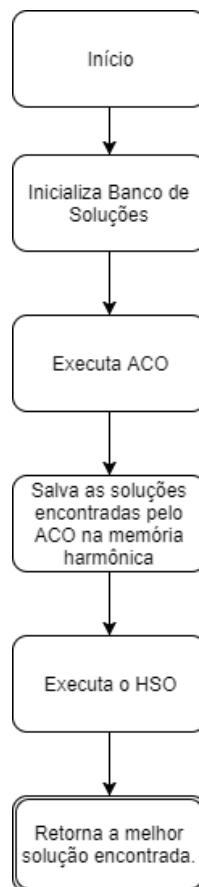


Figura 3.1: Fluxograma representativo da organização dos algoritmos.

Utilizou-se um sistema híbrido, pelo fato que somente a técnica de Otimização por Colônia de Formigas não alcançou uma solução sem violar as penalizações fortes. Após pesquisas na literatura, foram encontrados ótimos resultados com combinações de algoritmos, principalmente utilizando o HSO. Isso motivou o uso deste algoritmo.

### 3.2.1 Otimização por Colônia de Formigas (ACO)

Na Seção 2.2.1 é explicado toda a ideia em que o algoritmo de Otimização por Colônia de Formigas se baseia. Nesta seção, utiliza-se todos os conceitos explicados e detalha-se o seu funcionamento dentro do problema proposto. A Figura 3.2 exibe um fluxograma que representa o funcionamento do ACO utilizado.

O algoritmo é executado  $N$  vezes na tentativa de encontrar uma nova solução. Para cada iteração,  $M$  formigas criam novas soluções e buscam horários para as turmas utilizando os métodos ProximoItemPermutado, Roleta ou Torneio que são explicados mais a diante. Ao fim da seleção, os horários são avaliados, garantindo que para cada turma seus horários não possuam conflitos entre si, para verificar a validade da solução. Caso for verdadeiro é avaliado a quantidade de restrições infringidas pela solução encontrada e a próxima formiga gera uma nova solução. Caso contrário uma nova solução é selecionada para a formiga atual. Após todas as formigas possuírem uma nova solução, os feromônios são atualizados levando em consideração as melhores soluções encontradas. Os resultados são ranqueadas de acordo com a avaliação realizada após sua geração. No final as melhores soluções são retornadas para inicializar a Memória Harmônica do HSO.

Tem-se a seguir três métodos utilizados pelo ACO para alocação dos horários das turmas, onde cada um possui características distintas e comuns. O ProximoItemPermutado foca em uma busca gulosa randomizada para encontrar suas soluções. O algoritmo da Roleta, leva em consideração a porcentagem dos feromônios depositados em cada horário para então selecionar a solução. Por fim, o algoritmo do Torneio sorteia três horários onde o ganhador é o que possuir maior ou menor quantidade de feromônios, variando de acordo com um sorteio realizado. Todos estes algoritmos possuem em comum os feromônios como fator principal de escolha.

A ideia do método ProximoItemPermutado, Figura 3.3, descrito em (NOTHEG-

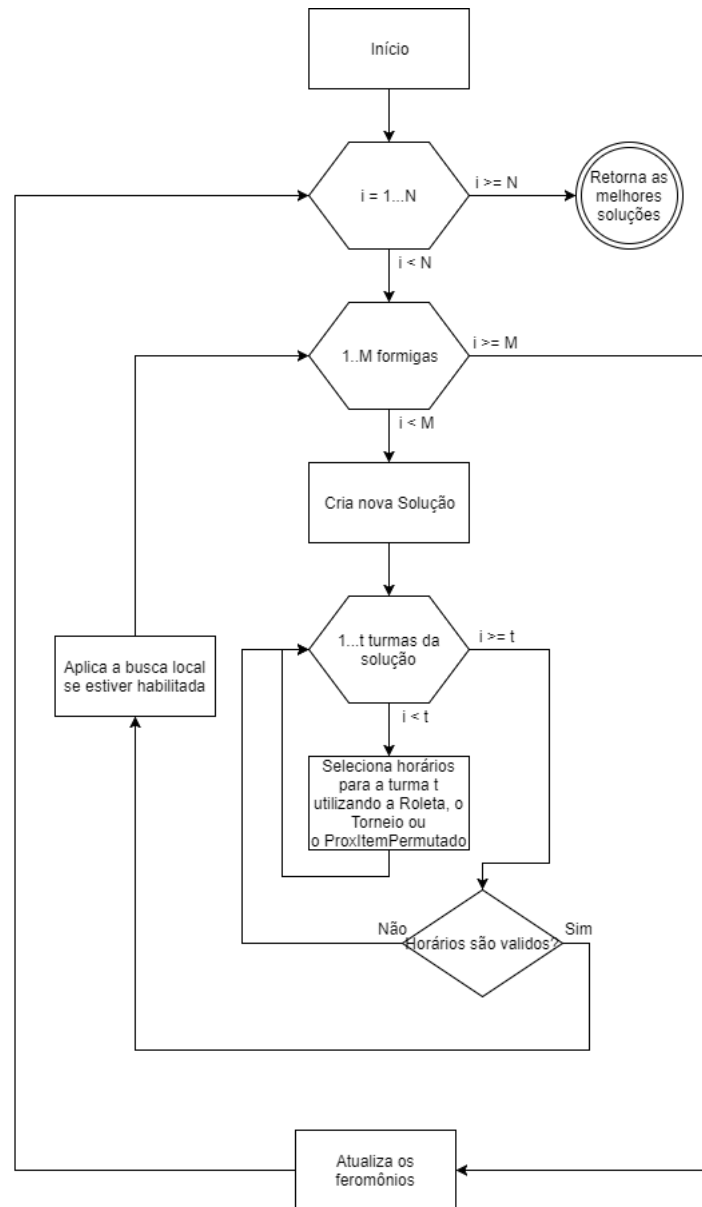


Figura 3.2: Fluxograma representativo do algoritmo de formigas (ACO).

GER et al., 2012), é utilizar um algoritmo guloso randomizado para alocar horários para as turmas. Uma ordenação gulosa randomizada é aquela que, apesar do vetor ser ordenado seguindo uma determinada regra, seu conteúdo não necessariamente precisa respeitá-la, podendo variar algumas posições. Por exemplo, na Figura 3.4 o vetor  $\omega$  está ordenado de forma gulosa randomizada.

O método ProximoItemPermutado possui algumas características que melhoram a performance e os resultados do algoritmo, por exemplo, o vetor guloso randomizado que contém os horários a serem alocados. Esse vetor é ordenado de acordo com a necessidade do algoritmo de buscar novos horários candidatos a solução, diminuindo, assim, o

processamento.

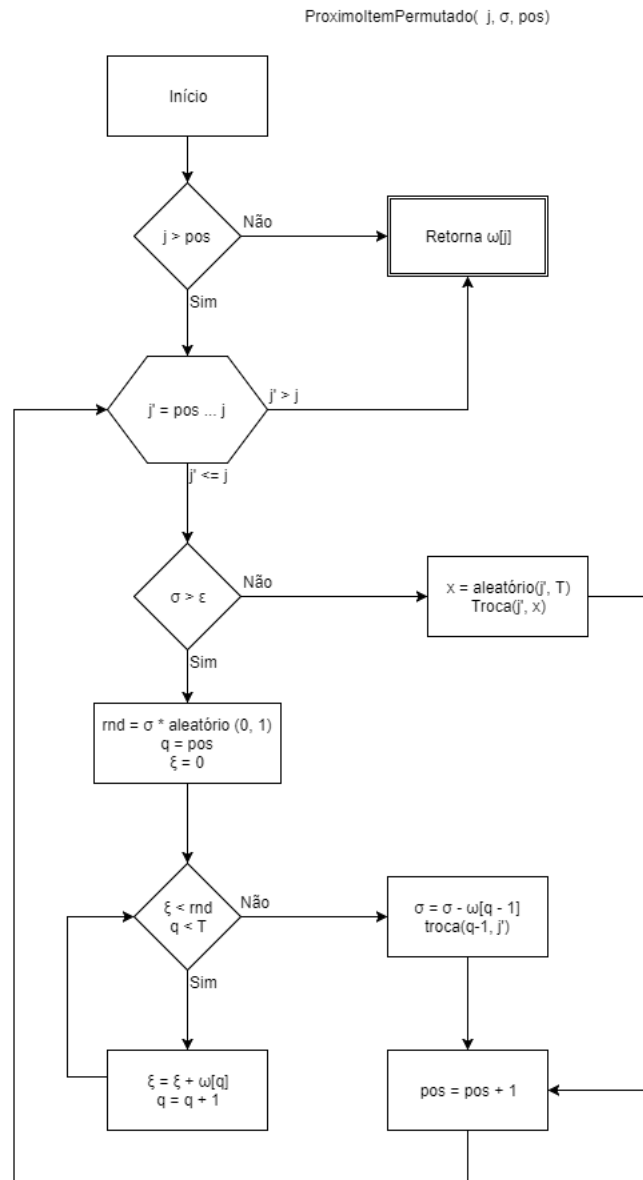


Figura 3.3: Fluxograma representativo do algoritmo de seleção ProximoItemPermutado.

O ProximoItemPermutado, recebe três parâmetros.  $J$  indica a posição do vetor de feromônios que o algoritmo requisita para a solução,  $\sigma$  representa o somatório dos feromônios existentes ainda não ordenados e  $pos$  representa até qual posição do vetor  $\omega$  está ordenada.  $\omega$  é responsável por armazenar os horários que foram ordenados de forma gulosa randomizada por esse algoritmo. Inicialmente é verificado se a posição desejada já foi ordenada, caso verdadeiro basta retorná-la. Caso seja falso, o algoritmos ordena o vetor  $\omega$  até alcançar a posição  $J$  desejada. Em cada iteração, realiza-se uma verificação para analisar se o somatório de feromônios  $\sigma$  é maior que o parâmetro  $\epsilon$ . Esse responsabiliza-se por determinar um somatório mínimo de feromônios. Caso for verdadeiro, a quantidade

de feromônios é tão baixa que uma ordenação não faz uma diferença considerável, então é feita uma troca totalmente randômica. Se for falso, é sorteado um valor aleatório entre zero e um, multiplicado pelo total de feromônios e salvo na variável  $\text{rnd}$ . Esse é responsável por, em conjunto com o total de horas  $T$ , definir qual posição  $q$ , calculada a partir do somatório de  $\omega[q]$  e  $\text{rnd}$ , deve ser trocada com  $j'$ , posição atual a ser modificada, para a ordenação do vetor  $\omega$ .

A Figura 3.4 mostra o resultado final da matriz de feromônios após a aplicação do `ProximoItemPermutado`. Essa possui o vetor  $\tau_i^S$ , que representa os horários  $S$  para a turma  $i$ , e seus respectivos feromônios, que são os valores dentro de  $\tau$ . Inicialmente, tem-se o vetor  $\tau_i^S$ , de feromônios sem sofrer alterações. A cada processamento do algoritmo um novo índice desse vetor é ordenado de forma gulosa randomizada de acordo com a necessidade de novos horários solicitados para a solução e salvos em  $\omega$ . Após o término de todo processamento, o vetor  $\omega$  não está totalmente ordenado, devido ao mesmo ter sofrido uma ordenação gulosa randomizada, mas encontra-se de acordo com o proposto pelo algoritmo. O fato da organização do vetor  $\omega$  não estar totalmente ordenada não afeta a solução, auxiliando a diversificar os resultados.

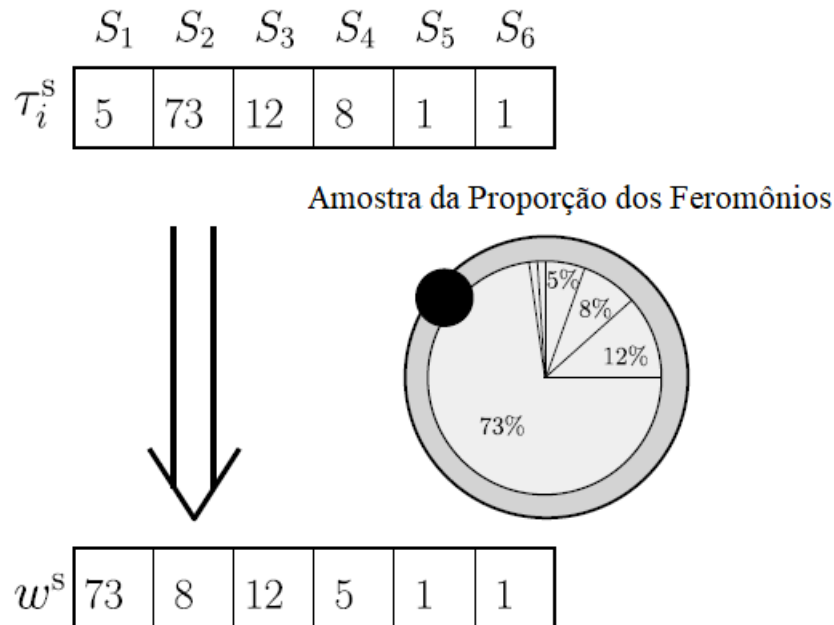


Figura 3.4: Geração do vetor pseudo randômico. Adaptado de: (NOTHEGGER et al., 2012).

Outra opção de otimização local utilizada no lugar do algoritmo `ProximoItem-`

Permutado foi o algoritmo da Roleta. Esse é um método cujo todos os horários possuem uma porcentagem de seleção definida, considerando sua quantidade de feromônios. O mínimo é de 1%, mesmo se sua quantidade de feromônios for inferior a este valor. Essa medida foi tomada para que todos tenham possibilidade de ser selecionados.

O fluxograma representado na Figura 3.5 indica o funcionamento do algoritmo da Roleta. Cria-se, inicialmente, um vetor vazio para armazenar a porcentagem de todos os horários a serem selecionados. Em seguida, realiza-se o somatório dos feromônios existentes em cada horário candidato à solução. Após isso, para cada horário  $i$ , verifica-se a quantidade de feromônios  $\tau_i$  é menor que 1, se for verdadeiro, adiciona-se ao vetor roleta uma posição para esse o horário. Caso seja falso, adiciona-se o número de casas condizentes a porcentagem de feromônios em  $i$  relacionados ao total existente. Cada posição no vetor roleta representa 1% da quantidade total de feromônios. Por fim, sorteia-se uma posição deste vetor, que será o horário selecionado.

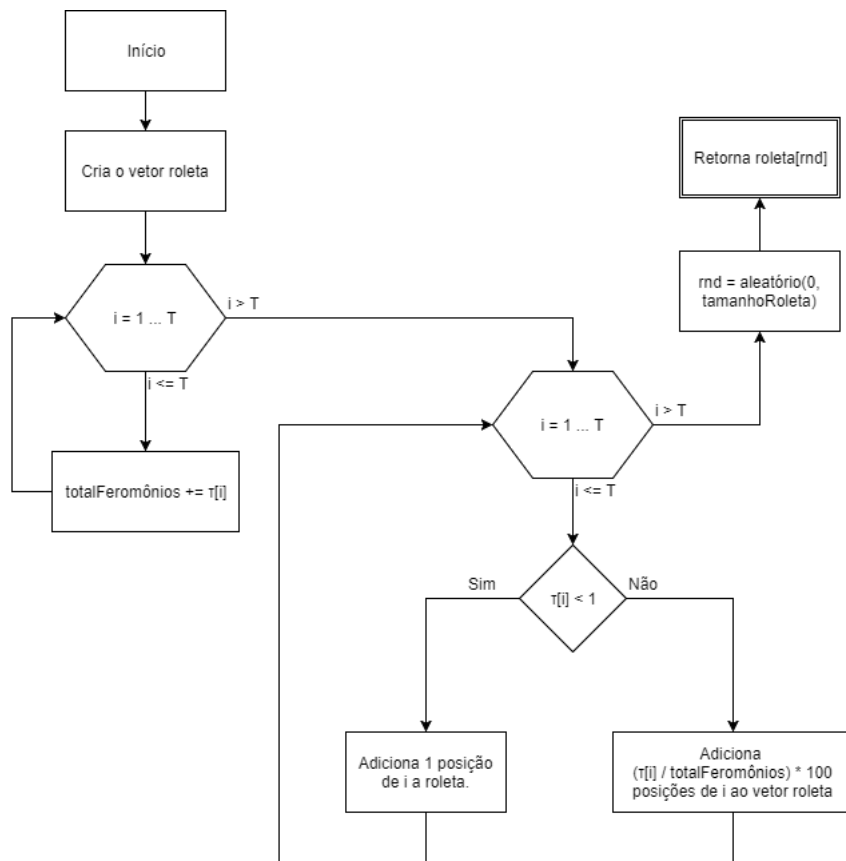


Figura 3.5: Fluxograma representativo do algoritmo de seleção da Roleta.

Na Figura 3.6 tem-se dois vetores, o  $T$  e o Roleta. No primeiro, cada índice  $i$  representa um espaço de tempo (hora) e seu conteúdo é a quantidade de feromônios. No

segundo, o conteúdo de cada posição é um índice de  $T$  e sua quantidade de repetições é definida pelo número de feromônios existentes no conteúdo de  $T_i$ . Por exemplo, o espaço de tempo um possui quatro posições no vetor Roleta. O algoritmo da Roleta recebe o vetor  $T$  e gera o Roleta a fim de sortear o horário candidato para a nova solução. Um fator importante de se notar é que a posição 4 não possui feromônios, mas o mesmo obteve um índice no vetor Roleta. Isso ocorre para que todos possam probabilidade de serem selecionados.

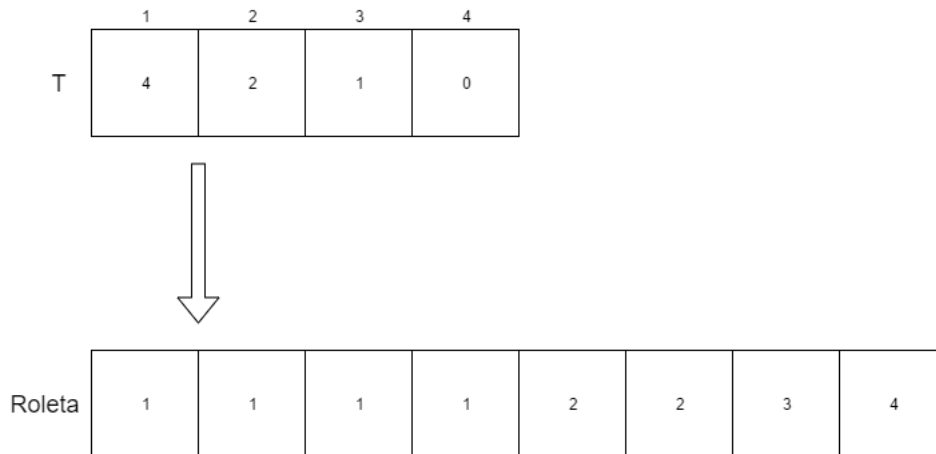


Figura 3.6: Método de geração do vetor a ser selecionado pela Roleta

Um terceiro algoritmo de seleção utilizado foi o Torneio, Figura 3.7, muito utilizado em Algoritmos Genéticos. Esse recebe três parâmetros:  $\tau$ , vetor de feromônios para cada horário,  $C$ , parâmetro responsável por indicar a probabilidade de ser utilizado o horário com maior ou menor número de feromônios, e  $K$  que representa o tamanho do torneio. Inicialmente  $K$  horários são selecionados de forma randômica e adicionados ao vetor  $\omega$ . Após isso, um número entre zero e cem é sorteado e salvo em  $rnd$ . Para os  $K$  horários sorteados verifica-se se  $rnd$  é maior que  $C$ . Se for verdade, é salvo o horário com maior quantidade de feromônios como vencedor, caso contrário, o menor é selecionado.

A Figura 3.8 apresenta em  $T$  três espaço de tempo, 7, 2 e 5. Esses foram sorteados aleatoriamente pelo algoritmo do Torneio. Após isso é realizado outro sorteio e comparado seu valor com  $C$ . O que possui maior número de feromônios foi o vencedor, portanto a posição representada pelo maior feromônio, que é 50, foi selecionada.

Ao final de cada iteração é realizado o aumento da quantidade de feromônios. Para isso, utilizou-se como base as equações descritas em (NOTHEGGER et al., 2012).



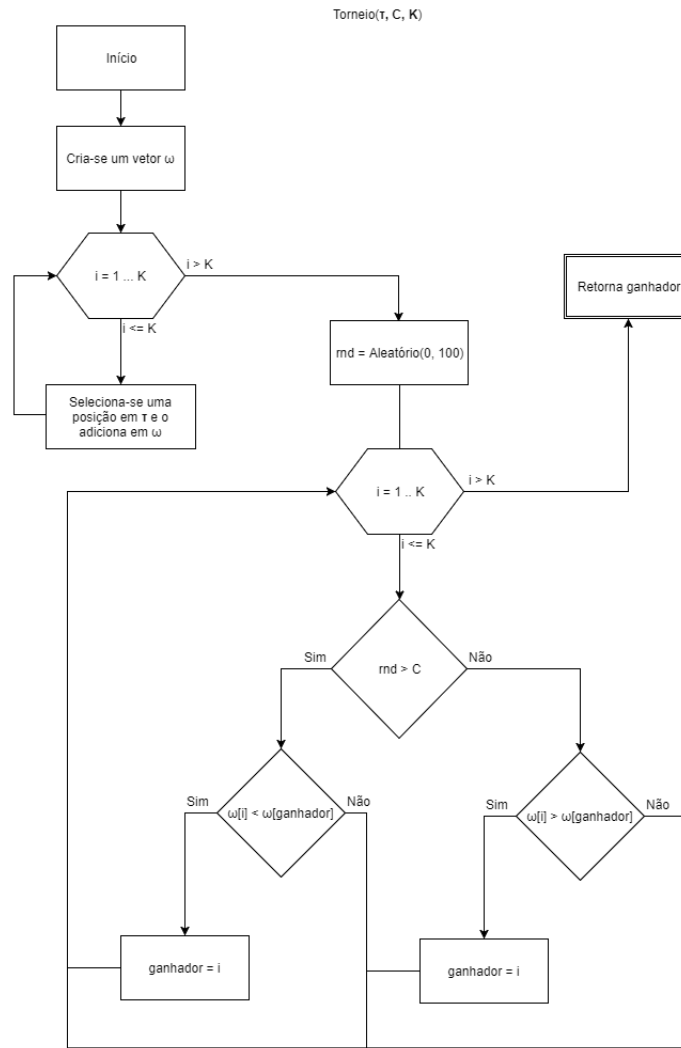


Figura 3.7: Fluxograma representativo do algoritmo de seleção do Torneio.

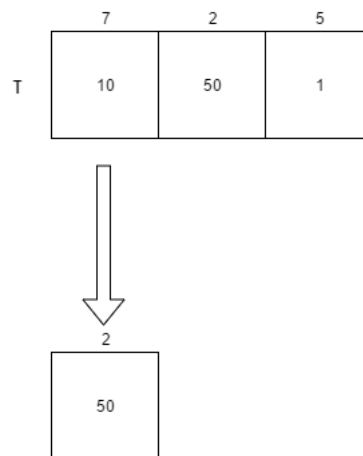


Figura 3.8: Método de geração do vetor a ser selecionado pelo Torneio.

Essas levam-se em consideração a quantidade de violações de restrições fortes (HCP) e a quantidade de violações de restrições fracas (SCP). Isso em conjunto com uma leve evaporação de feromônio para evitar convergências para ótimos locais.

A equação 3.1 realiza o cálculo da variável  $f$ , para isso é dividido  $V_0$  pelas penalizações fortes (HCP) da solução, ou retornado  $V_1$  caso o HCP for zero. Já a equação 3.2 realiza o cálculo da variável  $g$ , essa funciona de forma análoga a 3.1, com exceção de seus valores, que são os fatores leves violados (SCP),  $U_0$  e  $U_1$ . As constantes  $V_0$ ,  $V_1$ ,  $U_0$  e  $U_1$  são parâmetros do problema. Na 3.4 é calculado o  $\Delta n\tau_{ij}$ , responsável por punir o acréscimo de feromônios de um horário caso o mesmo viole algum fator de penalizações leves.  $\gamma$  é uma constante da equação.

$$f = \begin{cases} U_1 & \text{se o HCP for zero} \\ U_0/HCP & \text{caso contrário.} \end{cases} \quad (3.1)$$

$$g = \begin{cases} V_1 & \text{se o SCP for zero} \\ V_0/SCP & \text{caso contrário.} \end{cases} \quad (3.2)$$

$$\Delta\tau_{ij} = \begin{cases} f \cdot g & \text{se a turma } i \text{ e o horário } j \text{ faz parte da solução.} \\ 0 & \text{caso contrário.} \end{cases} \quad (3.3)$$

$$\Delta n\tau_{ij} \leftarrow (1 - (1 - \gamma)^{SCP(i)}) \cdot f \cdot g \quad (3.4)$$

Obtendo-se os valores de  $f$  e  $g$  e  $\Delta n\tau$  a equação 3.5 os utiliza para calcular o  $\tau_{ij}^s$ , responsável por determinar, em conjunto com a equação 3.4, a quantidade que os feromônios dos horários selecionados para esta solução sofrem acréscimo. Por fim, a equação 3.5 utiliza todos esses fatores para atualizar os feromônios dos horários definidos na solução, onde  $\tau_{ij}$  é a quantidade de feromônios que o horário atual possui.

$$\tau_{ij}^s \leftarrow \max(0, \tau_{ij}^s + \Delta\tau_{ij}^s - \Delta n\tau_{ij}^s) \quad (3.5)$$

As equações 3.6 e 3.7, também baseadas em (NOTHEGGER et al., 2012), são responsáveis pela evaporação dos feromônios. Isso ocorre de forma sutil a cada iteração, 3.6, ou de forma mais agressiva, realizando uma normalização caso a solução esteja estagnada durante 500 iterações, 3.7.

$$\tau_{ij} \leftarrow (1 - \rho)\tau_{ij} \quad (3.6)$$

$$\tau_{ij} = (\tau_{ij} - n\tau) \cdot \frac{n\tau\omega}{\max(\tau_{ij})} + n\tau \quad (3.7)$$

Ao final de cada iteração é aplicada a matriz de feromônios  $\tau_{ij}$  a equação 3.6, sendo  $\rho$  é uma constante. Após ocorrer várias iterações e não obter uma melhora na melhor solução encontrada pelo ACO, utiliza-se a equação 3.7 para normalizar o valor contido na matriz de feromônios. Isso ocorre na tentativa de eliminar ótimos locais alcançados pelo algoritmo. Na equação 3.7 tem-se que  $\tau_{ij}$  é a quantidade atual de feromônios,  $n\tau$  é a média de feromônios e  $\omega$  é uma constante.

Este algoritmo como mencionado gera um número  $M$  de soluções, nas quais essas são utilizadas como ponto de partida para a técnica de busca harmônica.

### 3.2.2 Otimização por Busca Harmônica

A Busca Harmônica, explicada na Seção 2.2.2, utiliza as soluções encontradas pelo ACO em sua Memória Harmônica, as combinam gerando novas soluções e no final retorna as melhores. O fluxograma da Figura 3.9, explica o funcionamento do algoritmo implementado de busca harmônica.

A memória harmônica (HM) do HSO é inicializada pelo algoritmo do ACO. Essa etapa normalmente é gerada de forma randômica, mas isso pode ocasionar um aumento de tempo necessário para o algoritmo convergir às melhores soluções. Esse fato foi predominante para escolher a inicialização através das soluções encontradas pelo algoritmo ACO.

A Figura 3.9 representa um fluxograma do funcionamento da otimização por busca harmônica, baseado em (AL-BETAR; KHADER; ZAMAN, 2012). Assim como o ACO, tem-se inicialmente um laço que executa um número  $N$  de vezes, sendo  $N$  um parâmetro do algoritmo. Em cada execução será gerada uma nova solução, essa possui todas as turmas contidas no Plano Departamental. Em seguida, é selecionado um horário para cada turma ser ministrada, para isso, tem-se um parâmetro HMCR que determina

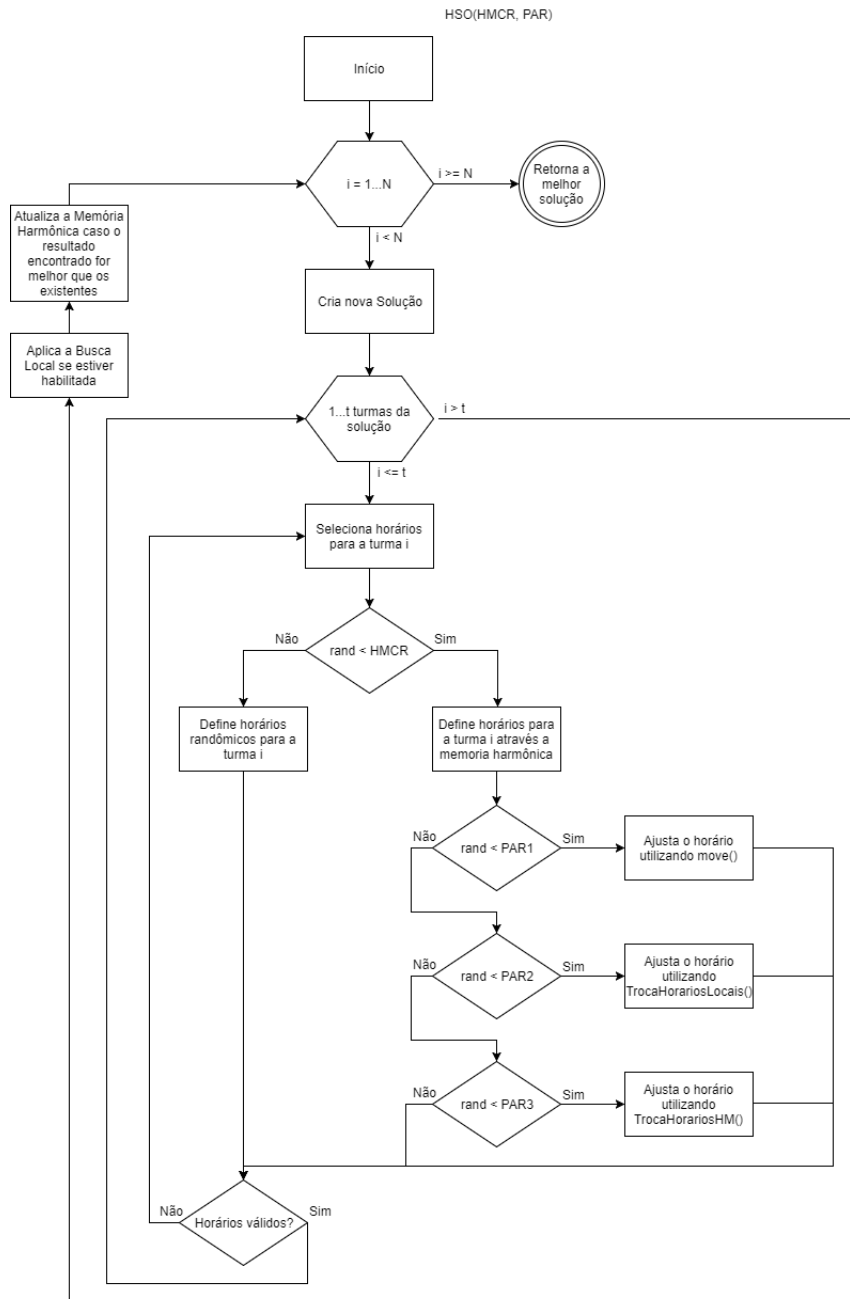


Figura 3.9: Fluxograma da Otimização por Busca Harmônica(HSO).

através de uma escolha randômica se é utilizado como base a Memória Harmônica, ou se são totalmente aleatórios. Caso a HM for utilizada, seleciona-se aleatoriamente uma solução contida na mesma e utiliza o horário da turma correspondente a  $i$ .

Após selecionar os horários a partir da memória harmônica, os mesmos podem sofrer modificações dependendo de um novo valor randômico que será comparado com os parâmetros PAR1, PAR2 e PAR3. Caso a comparação com PAR1 for bem sucedida, aplica-se uma mudança no horário atual para qualquer horário que não está sendo utilizado na nova solução. Já PAR2 realiza a troca do horário selecionado com outro existente na

solução. Por fim, caso nenhum dos dois sejam selecionados existe a probabilidade da comparação obter sucesso com PAR3, sendo este responsável por buscar outro horário para a solução baseando-se na Memória Harmônica.

Caso a solução seja válida e sua pontuação menor que as existentes na memória harmônica, a pior é retirada e a nova é inserida mantendo sempre o vetor ordenado. Para encontrar o valor de PAR1, PAR2 e PAR3 basta dividir PAR igualmente entre PAR1, PAR2 e PAR3.

O melhor resultado encontrado é retornado para o software, esse os exhibe para a pessoa responsável pelo plano avaliá-lo, realizar alguma alteração, caso necessário, e aplicá-la ao semestre. A interface é explicada mais detalhadamente na Seção 3.4.

A Figura 3.10 exhibe um exemplo do funcionamento do HSO. Representa-se a Memória Harmônica com quatro soluções, a nova solução gerada pelo HSO e essa mesma solução após aplicar-se as funções utilizadas em PAR1, PAR2 e PAR3, Figura 3.9. Cada solução possui dez turmas e o conteúdo de cada uma representa seu horário.

Memória Harmônica							Nova Solução		Nova Solução com ajustes	
	S[1]	S[2]	S[3]	S[4]		S[nova]		S[nova]		
t[1]	8	9	13	21	⇒	t[1]	8	t[1]	8	
t[2]	13	18	17	9		t[2]	17	t[2]	18	
t[3]	11	13	18	11		t[3]	11	t[3]	11	
t[4]	18	10	13	19		t[4]	13	t[4]	13	
t[5]	8	11	15	17		t[5]	11	t[5]	15	
t[6]	12	19	16	17		t[6]	17	t[6]	17	
t[7]	8	17	17	20		t[7]	8	t[7]	8	
t[8]	20	15	10	10		t[8]	15	t[8]	11	
t[9]	17	19	13	18		t[9]	18	t[9]	18	
t[10]	10	18	21	17		t[10]	18	t[10]	17	

Figura 3.10: Exemplo da geração de uma solução pela Otimização por Busca Harmônica (HSO).

A solução S[nova], Figura 3.10, baseou-se na Memória Harmônica para ser definida. Por exemplo, a turma t[8] de S[nova] utilizou o horário de t[8], contido na Solução S[2]. Após essa seleção realizou o comparativo com PAR para definir se deve aplicar ajustes nos horários definidos.

As turmas t[2], t[5], t[8] e t[10] sofreram alterações em seus horários. Em t[2] aplicou-se a função Move() e alterou seu horário de 17 horas para às 18 horas, t[5] e t[8] tiveram seus horários trocados, pois aplicou-se a função TrocaHoráriosLocais(). Por fim, em t[10], houve a troca do seu horário com o de S[4][10], solução 4, turma 10, contido na Memória Harmônica.

Na Figura 3.11 tem-se outro exemplo do funcionamento do HSO, onde  $S[1]$ ,  $S[2]$  e  $S[3]$  são três soluções existentes na Memória Harmônica e seus valores representam o horário de suas turmas  $t[1]$ ,  $t[2]$  e  $t[3]$ , respectivamente. Inicialmente selecionou-se na Memória Harmônica três horários para as turmas sendo eles (220, 50, 504). Após primeira definição dos horários aplicou-se a função  $\text{Move}()$  no horário de  $t[1]$  e o mesmo foi alterado de 220 para 221. Em  $t[2]$  empregou-se a função  $\text{TrocaHorariosLocais}()$  alternando o horário de  $t[2]$  e  $t[3]$  entre si. Por fim,  $t[3]$  sofreu o efeito da função  $\text{TrocaHorariosHM}()$  e selecionou-se outro horário retirado da memória harmônica para a solução.

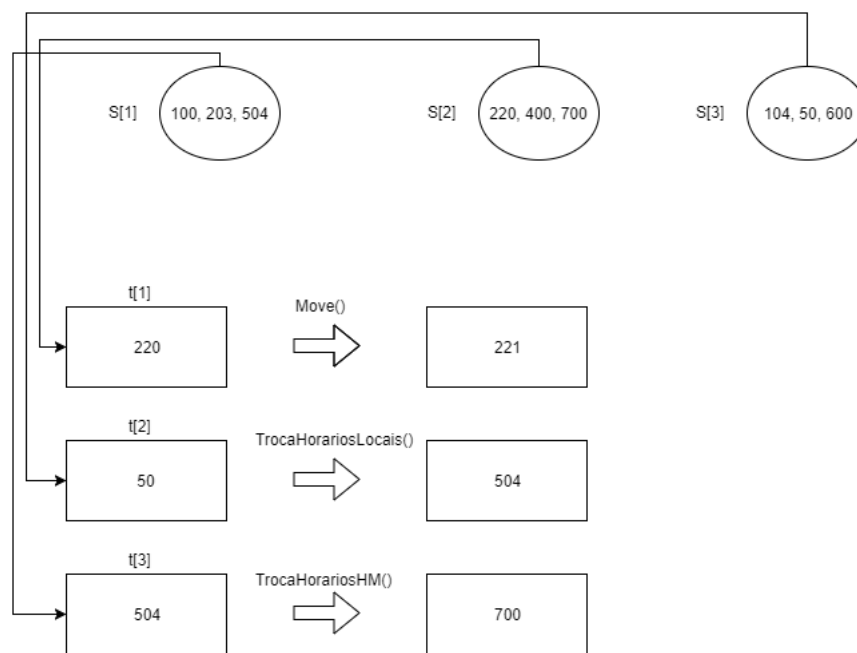


Figura 3.11: Exemplo da seleção de horários na Otimização por Busca Harmônica (HSO).

### 3.3 Busca Local

Para melhorar a solução foi implementada uma busca local de trocas. Essa funciona de forma a trocar o horário de duas turmas que estão violando penalizações. Utiliza-se essa busca nos métodos ACO e HSO, Figuras 3.2 e 3.9, e caso a solução melhore as mudanças são mantidas e o resultado retornado para os algoritmos.

Conforme apontado na Busca Local, Figura 3.12, a mesma é realizada um número  $N$  de vezes, onde  $N$  é um parâmetro. Inicialmente, é realizada uma busca na solução, retorna-se então duas turmas que violam restrições e seus horários são trocados entre si.

Sempre que a alteração não melhora a solução, é mantido o estado original da mesma.

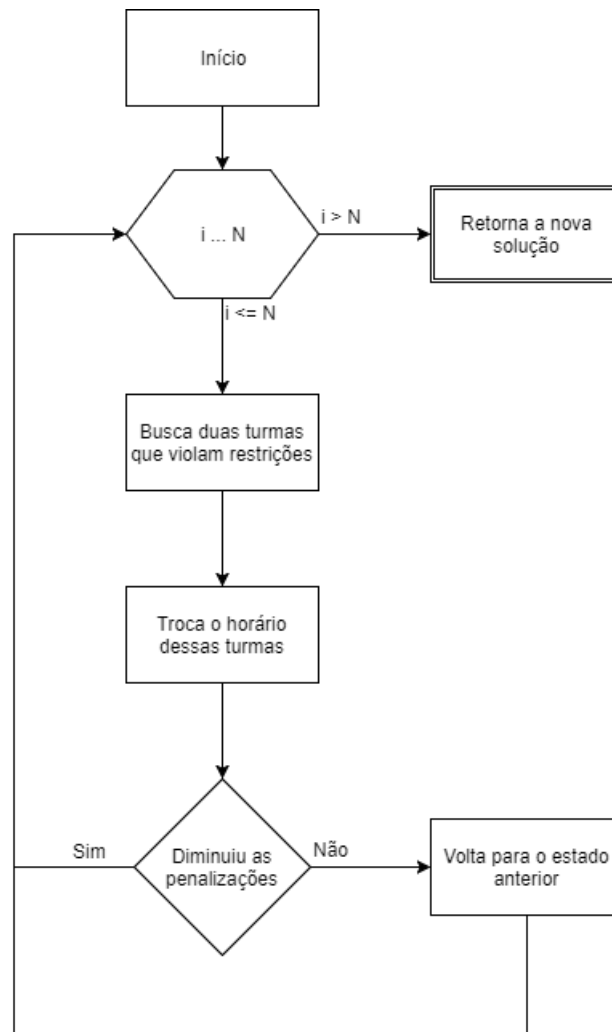


Figura 3.12: Fluxograma representativo da Busca Local.

No geral, buscas locais são executadas até não haver mais possibilidade de melhora, mas notou-se que a partir de um número  $N$  iterações a mesma não gerava evoluções significativas na solução. Assim, optou-se por definir um número fixo de iterações.

### 3.4 Software do Plano Departamental do DCC UFJF

Desenvolveu-se um sistema responsável por gerenciar todos os dados referentes ao plano departamental do DCC. Para isso, modelou-se um banco de dados, Figura 3.13, capaz de armazenar todas as informações existentes com o intuito de evitar o retrabalho em reinserção de dados.

O modelo do banco de dados conta com as seguintes tabelas para armazenamento

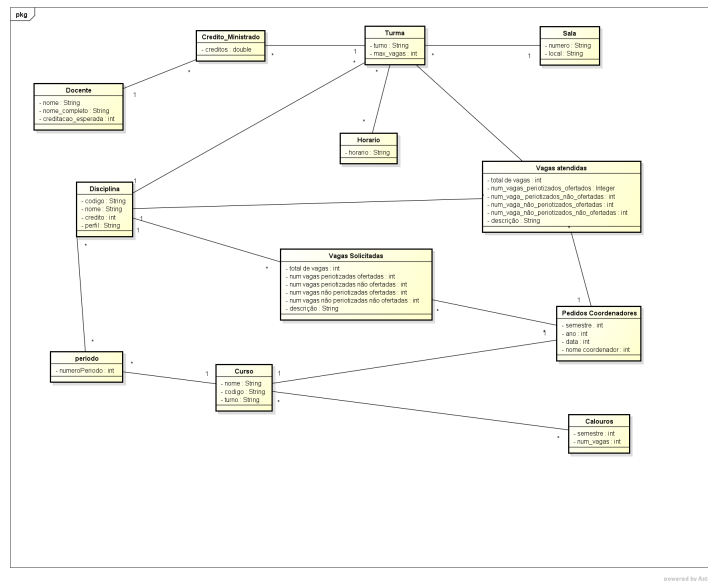


Figura 3.13: Modelagem do Banco de Dados utilizado.

das informações utilizadas pelo algoritmo híbrido:

- Disciplina;
- Turma;
- Docente;
- Horários;
- Curso.

Além desses dados, o software também gerencia outras informações pertinentes a plano departamental. Essas informações são relevantes em outros processos da montagem do plano departamental, mas por não serem o foco da otimização, o algoritmo inteligente não leva-os em consideração. Sendo essas:

- Créditos Ministrado;
- Salas de Aula;
- Calouros;
- Pedidos de Coordenadores;
- Vagas Atendidas;



- Vagas Solicitadas.

Cada docente é responsável por ministrar um determinado número de disciplinas gerando créditos (horas-aula) para os mesmos, esses são armazenados na tabela “Crédito Ministrado”. Todo início de semestre uma nova leva de alunos ingressa nos cursos, a quantidade total é armazenados em “Calouros”. Coordenadores realizam pedidos de vagas para as respectivas disciplinas cursadas por seus alunos. Para isso tem-se as tabelas pedido de coordenadores e vagas solicitadas. Quando essas vagas são atendidas elas são movidas para “Vagas Atendidas”.

Ao executar o programa exibe-se, inicialmente, a interface, Figura 3.14, onde a pessoa responsável pela gerência do plano tem a opção de escolher qual o ano e o semestre que deseja-se gerenciar. Inserindo-os, o programa faz uma busca no banco de dados e caso já existam todas as informações referentes são exibidas, caso contrário, cria-se automaticamente todo o ambiente necessário para facilitar a inserção de novos dados.

Semestre	Nome	Código	Horário 1	Horário 2	Créditos	Docente	Numero de Vagas
1	ALGORITMOS	DCC119-A	Segunda-feira 08:00	Quinta-feira 08:00	4	PISSINI	50
1	ALGORITMOS	DCC119-B	Segunda-feira 08:00	Quinta-feira 08:00	4	GIULIANO	50
1	ALGORITMOS	DCC119-C	Segunda-feira 08:00	Quinta-feira 08:00	4	GANATO	50
1	ALGORITMOS	DCC119-D	Segunda-feira 14:00	Quinta-feira 14:00	4	GRASIELE	50
1	ALGORITMOS	DCC119-E	Segunda-feira 14:00	Quinta-feira 14:00	4	LUCIANA BRUG	50
1	ALGORITMOS	DCC119-F	Segunda-feira 14:00	Quinta-feira 14:00	4	BERNARDO	50
1	ALGORITMOS	DCC119-G	Segunda-feira 14:00	Quinta-feira 14:00	4	GIULIANO	50
1	ALGORITMOS	DCC119-H	Segunda-feira 19:00	Quinta-feira 19:00	4	GRASIELE	50
1	ALGORITMOS	DCC119-X	Segunda-feira 14:00	Quinta-feira 14:00	4	LORENZA	50
1	LABORATORIO DE CIENCIAS	ICE902-A	Segunda-feira 11:00		2	GIULIANO	50
1	LABORATORIO DE CIENCIAS	ICE902-B	Terça-feira 16:00		2	GRASIELE	50
1	LABORATORIO DE CIENCIAS	ICE902-C	Sexta-feira 12:00		2	IGOR KNOP	50
1	LABORATORIO DE CIENCIAS	ICE902-D	Terça-feira 14:00		2	IGOR KNOP	50
1	LABORATORIO DE CIENCIAS	ICE902-E	Segunda-feira 14:00		2	PROF REDES	50
1	LABORATORIO DE CIENCIAS	ICE902-F	Quarta-feira 17:00		2	PROF REDES	50
1	LABORATORIO DE CIENCIAS	ICE902-G	Quinta-feira 08:00		2	PROF REDES	50
1	LABORATORIO DE CIENCIAS	ICE902-H	Segunda-feira 10:00		2	IGOR KNOP	50
1	LABORATORIO DE CIENCIAS	ICE902-I	Quinta-feira 08:00		2	PROF TE01	50
1	LABORATORIO DE CIENCIAS	ICE902-J	Segunda-feira 16:00		2	PROF TE01	50
1	LABORATORIO DE CIENCIAS	ICE902-K	Sexta-feira 12:00		2	PROF REDES	50
1	LABORATORIO DE CIENCIAS	ICE902-L	Terça-feira 14:00		2	IGOR KNOP	50
1	LABORATORIO DE CIENCIAS	ICE902-M	Sexta-feira 19:00		2	IGOR KNOP	50
1	LABORATORIO DE CIENCIAS	ICE902-N	Sexta-feira 09:00		2	GIULIANO	50
1	LABORATORIO DE CIENCIAS	ICE902-O	Quarta-feira 19:00		2	IGOR KNOP	50
1	LABORATORIO DE CIENCIAS	ICE902-P	Terça-feira 19:00		2	GRASIELE	50
1	LABORATORIO DE CIENCIAS	ICE902-Q	Quinta-feira 19:00		2	GRASIELE	50
1	LABORATORIO DE PROGRAMAÇÃO	DCC120-A	Terça-feira 08:00		2	GRASIELE	50
1	LABORATORIO DE PROGRAMAÇÃO	DCC120-AA	Terça-feira 08:00		2	GIULIANO	50
1	LABORATORIO DE PROGRAMAÇÃO	DCC120-B	Terça-feira 10:00		2	GIULIANO	50
1	LABORATORIO DE PROGRAMAÇÃO	DCC120-AB	Terça-feira 10:00		2	PROF TE01	50
1	LABORATORIO DE PROGRAMAÇÃO	DCC120-C	Terça-feira 14:00		2	LORENZA	50
1	LABORATORIO DE PROGRAMAÇÃO	DCC120-CC	Terça-feira 14:00		2	JURY	50
1	LABORATORIO DE PROGRAMAÇÃO	DCC120-D	Terça-feira 19:00		2	LORENZA	50
1	LABORATORIO DE PROGRAMAÇÃO	DCC120-E	Terça-feira 19:00		2	PROF REDES	50
1	LABORATORIO DE PROGRAMAÇÃO	DCC120-F	Terça-feira 19:00		2	PROF REDES	50
1	LABORATORIO DE PROGRAMAÇÃO	DCC120-G	Quarta-feira 08:00		2	PROF TE01	50
1	LABORATORIO DE PROGRAMAÇÃO	DCC120-GS	Quarta-feira 08:00		2	GANATO	50
1	LABORATORIO DE PROGRAMAÇÃO	DCC120-H	Quarta-feira 10:00		2	PROF TE01	50
1	LABORATORIO DE PROGRAMAÇÃO	DCC120-I	Quarta-feira 10:00		2	FABRICIO	50
1	LABORATORIO DE PROGRAMAÇÃO	DCC120-II	Quarta-feira 14:00		2	IGOR KNOP	50
1	LABORATORIO DE PROGRAMAÇÃO	DCC120-III	Quarta-feira 14:00		2	BERNARDES	50
1	LABORATORIO DE PROGRAMAÇÃO	DCC120-IV	Quarta-feira 17:00		2	FABRICIO	50
1	LABORATORIO DE PROGRAMAÇÃO	DCC120-V	Quarta-feira 17:00		2	PROF REDES	50
1	LABORATORIO DE PROGRAMAÇÃO	DCC120-VI	Quarta-feira 19:00		2	RODRIGO LUIS	50
1	METODOLOGIA IDENTIFICACAO EM COMP	DCC123-A	Sexta-feira 19:00	Quarta-feira 12:00	4	ALESSANDRE	50
1	ESTRUTURA DE DADOS	DCC131-A	Quinta-feira 21:00	Segunda-feira 19:00	4	ITAMAR	50
1	LABORATORIO DE PROGRAMAÇÃO II	DCC107-A	Segunda-feira 14:00	Sexta-feira 12:00	4	VANA	50
1	LABORATORIO DE PROGRAMAÇÃO II	DCC107-AA	Segunda-feira 14:00	Sexta-feira 12:00	4	PAULO	50
1	LABORATORIO DE PROGRAMAÇÃO II	DCC107-AB	Quinta-feira 19:00		2	HEIDER	50
1	LABORATORIO DE PROGRAMAÇÃO III	DCC156-A	Segunda-feira 19:00		2	GRO	50
1	LABORATORIO DE PROGRAMAÇÃO WEB	DCC121-A	Sexta-feira 21:00		2	IGOR KNOP	50
1	LINGUAGEM DE PROGRAMAÇÃO II	EADCC031-A	Quarta-feira 12:00	Sexta-feira 10:00	4	ROMUALDO	50
1	PROGRAMAÇÃO PARA WEB II	EADCC035-A	Quarta-feira 14:00	Quinta-feira 19:00	4	VICTOR	50
1	SEMINARIO EM COMPLUCAO VIII	DCC104-A	Sexta-feira 19:00		2	IGOR KNOP	50

Figura 3.14: Interface responsável por exibir o Plano departamental.

Na Figura 3.14 a interface em questão exibe todas as informações para o bom entendimento do resultado do Plano Departamental, essas são: o semestre atual, o nome da disciplina e seu código, o horário que foi definido pelo programa ou pelo professor responsável pelo plano, os créditos dessa disciplina, o professor responsável para ministrá-la e sua quantidade de vagas. Também possui um botão responsável para a geração de

uma nova grade com os algoritmos descritos no Capítulo 3.

A Figura 3.15 exibe os dados dos Docentes, sendo as quantidades de disciplinas ministradas por ele e o somatório de sua carga horária (créditos). Essas informações podem ser ocultadas, caso haja a necessidade, clicando-se no botão presente na primeira coluna e na linha desejada. Também possui um botão responsável por gerar um relatório com todos os dados presentes no plano departamental.

Código do Professor	Nome do Professor	Disciplinas	Código da Disciplina	Creditos da Disciplina	Créditos: Abais / Esperados
P1	ALESSANDREIA	METODOLOGIA CIENTIFICA EM COMPUTAÇÃO	DCC123 - A	4	0 / 8
P2	ALEX	ENGENHARIA DE SOFTWARE	DCC081 - A	4	0 / 8
P3	ANDRE	ANÁLISE DE DESEMPENHO DE SISTEMAS	DOC074 - A	4	0 / 8
P4	ARBEX	LABORATÓRIO DE REDES DE COMPUTADO	DOC072 - A	2	0 / 8
		INTERF. HÍBRIDO-COMPUTADOR	DOC174 - A	4	0 / 8
		MODELAGEM DE SISTEMAS	DCC117 - A	4	0 / 8
		ASPECTOS AVANÇADOS EM BANCO DE DAD.	DCC077 - A	4	0 / 8
		SISTEMAS DE APOIO À DECISÃO	DCC166 - A	4	0 / 8
P5	BARRIERE	CIRCUITOS DIGITAIS	DCC122 - A	4	0 / 8
		FUNDAMENTOS DA ORGANIZAÇÃO DE COMP.	EADCC0207 - A	2	0 / 8
		INTRODUÇÃO AOS SISTEMAS OPERACIONAIS	EADCC0206 - A	2	0 / 8
P6	BERNARDES	LABORATÓRIO DE PROGRAMAÇÃO	DCC126 - B	2	0 / 8
		PROCESSAMENTO DE IMAGENS	DCC265 - A	4	0 / 8
P7	BERNHARDO	ALGORITMOS	DCC119 - F	4	4 / 8
P8	CANNATO	LABORATÓRIO DE PROGRAMAÇÃO	DCC126 - OS	2	0 / 8
		ALGORITMOS	DCC119 - C	4	0 / 8
P9	CIRO	LABORATÓRIO DE PROGRAMAÇÃO IV	DCC156 - A	2	2 / 8
P20	CRISTIANO	TOPICOS EM COMPUTAÇÃO CIENTIFICA II	DCC089 - A	4	4 / 8
E1	ESTATISTICA 1	CALCULO DE PROBABILIDADES	EST001 - A	4	0 / 8
		LABORATÓRIO DE PROGRAMAÇÃO	DCC126 - F	2	0 / 8
P10	FABRÍCIO	LABORATÓRIO DE PROGRAMAÇÃO	DCC126 - HH	2	0 / 8
		LABORATÓRIO DE PROGRAMAÇÃO	DCC126 - J	2	0 / 8
P11	FERNANDA	PROJETO DE SOFTWARE EDUCACIONAL	EADCC0203 - A	4	4 / 8
F1	FISICA 1	FISICA 1	FS001 - B	4	10 / 8
		LABORATÓRIO DE FISICA	FS004 - A	2	0 / 8
F2	FISICA 2	FISICA 2	FS002 - A	4	0 / 8
		FISICA 2	FS002 - B	4	0 / 8
F3	FISICA 3	FISICA 3	FS003 - A	4	0 / 8
		FISICA 3	FS003 - B	4	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - A	2	0 / 8
P48	GUILIANO	LABORATÓRIO DE CIENCIAS	ICE002 - H	2	16 / 8
		LABORATÓRIO DE PROGRAMAÇÃO	DCC126 - B	2	0 / 8
		ALGORITMOS	DCC119 - B	4	0 / 8
		ALGORITMOS	DCC119 - G	4	0 / 8
		LABORATÓRIO DE PROGRAMAÇÃO	DCC126 - AA	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - B	2	0 / 8
		ALGORITMOS	DCC119 - H	4	0 / 8
P47	GRASIELE	LABORATÓRIO DE CIENCIAS	ICE002 - P	2	16 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - Q	2	0 / 8
		LABORATÓRIO DE PROGRAMAÇÃO	DCC126 - A	2	0 / 8
		ALGORITMOS	DCC119 - D	4	0 / 8
P12	HEDER	LABORATÓRIO DE PROGRAMAÇÃO II	DCC107 - AA	2	2 / 8
P13	HELIO	COMPUTAÇÃO EVOLUCIONISTA	DCC087 - A	4	0 / 8
		CALCULO NUMERICO	DCC086 - E	4	0 / 8
		LABORATÓRIO DE PROGRAMAÇÃO	DCC126 - I	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - O	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - M	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - L	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - H	2	18 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - C	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - D	2	0 / 8
P14	IGOR KNOP	LABORATÓRIO DE CIENCIAS	ICE002 - A	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - B	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - C	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - D	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - E	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - F	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - G	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - H	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - I	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - J	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - K	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - L	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - M	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - N	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - O	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - P	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - Q	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - R	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - S	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - T	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - U	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - V	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - W	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - X	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - Y	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - Z	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - AA	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - AB	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - AC	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - AD	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - AE	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - AF	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - AG	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - AH	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - AI	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - AJ	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - AK	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - AL	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - AM	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - AN	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - AO	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - AP	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - AQ	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - AR	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - AS	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - AT	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - AU	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - AV	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - AW	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - AX	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - AY	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - AZ	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - BA	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - BB	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - BC	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - BD	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - BE	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - BF	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - BG	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - BH	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - BI	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - BJ	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - BK	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - BL	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - BM	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - BN	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - BO	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - BP	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - BQ	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - BR	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - BS	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - BT	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - BU	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - BV	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - BW	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - BX	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - BY	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - BZ	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - CA	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - CB	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - CC	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - CD	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - CE	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - CF	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - CG	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - CH	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - CI	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - CJ	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - CK	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - CL	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - CM	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - CN	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - CO	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - CP	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - CQ	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - CR	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - CS	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - CT	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - CU	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - CV	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - CW	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - CX	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - CY	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - CZ	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - DA	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - DB	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - DC	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - DD	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - DE	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - DF	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - DG	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - DH	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - DI	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - DJ	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - DK	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - DL	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - DM	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - DN	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - DO	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - DP	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - DQ	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - DR	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - DS	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - DT	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - DU	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - DV	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - DW	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - DX	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - DY	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - DZ	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - EA	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - EB	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - EC	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - ED	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - EE	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - EF	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - EG	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - EH	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - EI	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - EJ	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - EK	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - EL	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - EM	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - EN	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - EO	2	0 / 8
		LABORATÓRIO DE CIENCIAS	ICE002 - EP	2	0 / 8

Código do Curso	Nome do Curso	Código da Disciplina	Nome da Disciplina	Total de Vagas	Períodizados	Não Períodizados	Observações
0001		0001	ALGORITMOS	0	0	0	
0001		0002	LABORATÓRIO DE PROGRAMAÇÃO	0	0	0	
0001		0003	METODOLOGIA CIENTÍFICA EM C.	0	0	0	
0001		0004	LABORATÓRIO DE CIÊNCIAS	0	0	0	
0001		0005	ESTRUTURA DE DADOS I	0	0	0	
0001		0006	ESTRUTURA DE DADOS II	0	0	0	
0001		0007	SEMINÁRIO EM COMPUTAÇÃO VIII	0	0	0	
0001		0008	LABORATÓRIO DE PROGRAMAÇÃO II	0	0	0	
0001		0009	LABORATÓRIO DE PROGRAMAÇÃO III	0	0	0	
0001		0010	LABORATÓRIO DE PROGRAMAÇÃO IV	0	0	0	
0001		0011	LINGUAGEM DE PROGRAMAÇÃO II	0	0	0	
0001		0012	PROGRAMAÇÃO PARA WEB I	0	0	0	
0001		0013	SISTEMAS OPERACIONAIS	0	0	0	
0001		0014	ORGANIZAÇÃO DE COMPUTADORES	0	0	0	
0001		0015	LABORATÓRIO DE SISTEMAS OPE.	0	0	0	
0001		0016	ORCULOS DIGITAIS	0	0	0	
0001		0017	PROGRAMAÇÃO PARA EULA	0	0	0	
0001		0018	INTRODUÇÃO AOS SISTEMAS OPE.	0	0	0	
0001		0019	FUNDAMENTOS DA ORGANIZAO.	0	0	0	
0001		0020	BANCO DE DADOS	0	0	0	
0001		0021	ASPECTOS AVANÇADOS EM BANI.	0	0	0	
0001		0022	RECUPERAÇÃO DA INFORMAÇÃO	0	0	0	
0001		0023	SISTEMAS DE APOIO À DECISÃO	0	0	0	
0001		0024	COMPUTAÇÃO GRÁFICA	0	0	0	
0001		0025	PROCESSAMENTO DE IMAGENS	0	0	0	
0001		0026	REALIDADE VIRTUAL	0	0	0	
0001		0027	ORIENTAÇÃO A OBJETOS	0	0	0	
0001		0028	ENGENHARIA DE SOFTWARE	0	0	0	
0001		0029	INTELIGÊNCIA DE SISTEMAS	0	0	0	
0001		0030	INTERAÇÃO HUMANO-COMPUTADOR	0	0	0	
0001		0031	MODELAGEM DE SISTEMAS	0	0	0	
0001		0032	INTERFACE USUÁRIO E MÁQUINA	0	0	0	
0001		0033	PROJETO DE SOFTWARE EDUCAC.	0	0	0	
0001		0034	ÉTICA NA ENGENHARIA DE SO.	0	0	0	
0001		0035	INTELIGÊNCIA ARTIFICIAL	0	0	0	
0001		0036	TEORIA DOS GRÁFOS	0	0	0	
0001		0037	COMPUTAÇÃO EVOLUCIONISTA	0	0	0	
0001		0038	REDES NEURONAIS ARTIFICIAIS	0	0	0	
0001		0039	INTELIGÊNCIA COMPUTACIONAL	0	0	0	
0001		0040	LÓGICA E FUNDAMENTOS PARA A.	0	0	0	
0001		0041	LÓGICA PARA COMPUTAÇÃO	0	0	0	
0001		0042	INFORMÁTICA NA EDUCAÇÃO	0	0	0	
0001		0043	ASPECTOS LEGAIS DA INFORMÁT.	0	0	0	
0001		0044	CÁLCULO NUMÉRICO	0	0	0	
0001		0045	TOPICOS EM COMPUTAÇÃO CEN.	0	0	0	
0001		0046	PESQUISA OPERACIONAL	0	0	0	
0001		0047	SOLUÇÃO NUMÉRICA DE EQUILÍB.	0	0	0	
0001		0048	REDES DE COMPUTADORES	0	0	0	
0001		0049	LABORATÓRIO DE REDES DE CO.	0	0	0	
0001		0050	AVALIAÇÃO DE DESEMPENHO DE S.	0	0	0	
0001		0051	SEMINÁRIO EM COMPUTAÇÃO VI	0	0	0	
0001		0052	ADMINISTRAÇÃO DE REDES	0	0	0	
0001		0053	INTRODUÇÃO A SISTEMAS DE INFO.	0	0	0	
0001		0054	ASPECTOS ORGANIZACIONAIS II	0	0	0	
0001		0055	ANÁLISE E PROJETO DE ALGORIT.	0	0	0	
0001		0056	LINGUAGEM DE PROGRAMAÇÃO	0	0	0	
0001		0057	TEORIA DA COMPUTAÇÃO	0	0	0	

Figura 3.16: Interface responsável por cadastrar as solicitações de vagas dos coordenadores.

Código da Disciplina	Nome da Disciplina	Código do Curso	Nome do Curso	Total de Vagas Solicitadas	Total de Vagas Atendidas	Períodizados Solicitados	Períodizados Atendidos	Não Períodizados Solicitados	Não Períodizados Atendidos	Observações
DCC119	ALGORITMOS	65A	Sistema de Informação	0	0	0	0	0	0	
45A	Eng. Computacional	0	0	0	0	0	0	0	0	
35A	Ciência da Computação	0	0	0	0	0	0	0	0	
65B	Computação (Estatas)	0	0	0	0	0	0	0	0	
DCC120	LABORATÓRIO DE PROG.	45A	Eng. Computacional	0	0	0	0	0	0	
35A	Ciência da Computação	0	0	0	0	0	0	0	0	
65B	Computação (Estatas)	0	0	0	0	0	0	0	0	
DCC123	METODOLOGIA CIENTÍFICA	65A	Sistema de Informação	0	0	0	0	0	0	
45A	Eng. Computacional	0	0	0	0	0	0	0	0	
35A	Ciência da Computação	0	0	0	0	0	0	0	0	
65B	Computação (Estatas)	0	0	0	0	0	0	0	0	
ICE002	LABORATÓRIO DE CIEN.	45A	Eng. Computacional	0	0	0	0	0	0	
35A	Ciência da Computação	0	0	0	0	0	0	0	0	
65B	Computação (Estatas)	0	0	0	0	0	0	0	0	
DCC012	ESTRUTURA DE DADOS	45A	Eng. Computacional	0	0	0	0	0	0	
35A	Ciência da Computação	0	0	0	0	0	0	0	0	
65B	Computação (Estatas)	0	0	0	0	0	0	0	0	
DCC013	ESTRUTURA DE DADOS	45A	Eng. Computacional	0	0	0	0	0	0	
35A	Ciência da Computação	0	0	0	0	0	0	0	0	
65B	Computação (Estatas)	0	0	0	0	0	0	0	0	
DCC104	SEMINÁRIO EM COMPUT.	45A	Eng. Computacional	0	0	0	0	0	0	
35A	Ciência da Computação	0	0	0	0	0	0	0	0	
65B	Computação (Estatas)	0	0	0	0	0	0	0	0	
DCC107	LABORATÓRIO DE PROG.	45A	Eng. Computacional	0	0	0	0	0	0	
35A	Ciência da Computação	0	0	0	0	0	0	0	0	
65B	Computação (Estatas)	0	0	0	0	0	0	0	0	
DCC121	LABORATÓRIO DE PROG.	45A	Eng. Computacional	0	0	0	0	0	0	
35A	Ciência da Computação	0	0	0	0	0	0	0	0	
65B	Computação (Estatas)	0	0	0	0	0	0	0	0	
DCC156	LABORATÓRIO DE PROG.	45A	Eng. Computacional	0	0	0	0	0	0	
35A	Ciência da Computação	0	0	0	0	0	0	0	0	
65B	Computação (Estatas)	0	0	0	0	0	0	0	0	
EADCC031	LINGUAGEM DE PROGR.	45A	Eng. Computacional	0	0	0	0	0	0	
35A	Ciência da Computação	0	0	0	0	0	0	0	0	
65B	Computação (Estatas)	0	0	0	0	0	0	0	0	
EADCC035	PROGRAMAÇÃO PARA WE	45A	Eng. Computacional	0	0	0	0	0	0	
35A	Ciência da Computação	0	0	0	0	0	0	0	0	
65B	Computação (Estatas)	0	0	0	0	0	0	0	0	
DCC062	SISTEMAS OPERACIONAIS	45A	Eng. Computacional	0	0	0	0	0	0	
35A	Ciência da Computação	0	0	0	0	0	0	0	0	
65B	Computação (Estatas)	0	0	0	0	0	0	0	0	
DCC070	ORGANIZAÇÃO DE COMPU	45A	Eng. Computacional	0	0	0	0	0	0	
35A	Ciência da Computação	0	0	0	0	0	0	0	0	
65B	Computação (Estatas)	0	0	0	0	0	0	0	0	
65A	Sistema de Informação	0	0	0	0	0	0	0	0	

Figura 3.17: Exibe as solicitações de vagas dos coordenadores para cada disciplina.

peço pelo programa. É fornecido uma opção através de um botão para importar turmas já criadas em semestres anteriores.

As Figuras 3.19, 3.20 e 3.21 possuem uma tabela onde todos os dados dos professores, disciplinas e cursos, respectivamente, são inseridos no programa.

O software foi desenvolvido em conjunto com os professores responsáveis pela criação do plano departamental no DCC, o que reforça a necessidade do desenvolvimento do mesmo e aumenta a probabilidade de melhores resultados. No próximo capítulo, é

Nome	Plano Departamental	Resultado	Cadastro de Pedidos	Pedidos Coordenadores	Turmas	Docente	Disciplinas	Cursos	Calendario
Código	Turno	Maximo de Vagas	Nome da Disciplina	Código da Disciplina	Sala	Dia	Dia	Horario Fixo	
B	Diurno	100	TEORIA DOS NUMEROS	MAT007		Quarta-feira 14:00	Sexta-feira 14:00	<input checked="" type="checkbox"/>	
A	Nocturno	100	TEORIA DOS NUMEROS	MAT007		Quarta-feira 19:00	Sexta-feira 19:00	<input checked="" type="checkbox"/>	
B	Diurno	100	GEOMETRIA ANALITICA	MAT004		Quarta-feira 14:00	Sexta-feira 14:00	<input checked="" type="checkbox"/>	
A	Nocturno	100	GEOMETRIA ANALITICA	MAT004		Quarta-feira 19:00	Sexta-feira 19:00	<input checked="" type="checkbox"/>	
B	Diurno	100	EQUAÇÕES DIFERENCIAIS	MAT006		Quarta-feira 14:00	Sexta-feira 14:00	<input checked="" type="checkbox"/>	
A	Nocturno	100	EQUAÇÕES DIFERENCIAIS	MAT006		Quarta-feira 19:00	Sexta-feira 19:00	<input checked="" type="checkbox"/>	
B	Diurno	100	CALCULO 3	MAT003		Quarta-feira 14:00	Sexta-feira 14:00	<input checked="" type="checkbox"/>	
A	Nocturno	100	CALCULO 3	MAT003		Quarta-feira 19:00	Sexta-feira 19:00	<input checked="" type="checkbox"/>	
B	Diurno	100	CALCULO 2	MAT002		Quarta-feira 14:00	Sexta-feira 14:00	<input checked="" type="checkbox"/>	
A	Nocturno	100	CALCULO 2	MAT002		Quarta-feira 19:00	Sexta-feira 19:00	<input checked="" type="checkbox"/>	
B	Diurno	100	CALCULO 1	MAT001		Quarta-feira 14:00	Sexta-feira 14:00	<input checked="" type="checkbox"/>	
A	Nocturno	100	CALCULO 1	MAT001		Quarta-feira 19:00	Sexta-feira 19:00	<input checked="" type="checkbox"/>	
B	Diurno	100	ALGEBRA LINEAR	MAT005		Terça-feira 14:00	Quinta-feira 14:00	<input checked="" type="checkbox"/>	
A	Nocturno	100	ALGEBRA LINEAR	MAT005		Terça-feira 19:00	Quinta-feira 19:00	<input checked="" type="checkbox"/>	
B	Diurno	100	LABORATORIO DE FISICA	FB004		Segunda-feira 21:00	Quarta-feira 21:00	<input checked="" type="checkbox"/>	
A	Nocturno	100	LABORATORIO DE FISICA	FB004		Segunda-feira 21:00	Quarta-feira 21:00	<input checked="" type="checkbox"/>	
B	Diurno	100	FISICA 3	FB003		Terça-feira 14:00	Quinta-feira 14:00	<input checked="" type="checkbox"/>	
A	Nocturno	100	FISICA 3	FB003		Terça-feira 19:00	Quinta-feira 19:00	<input checked="" type="checkbox"/>	
B	Diurno	100	FISICA 2	FB002		Terça-feira 14:00	Quinta-feira 14:00	<input checked="" type="checkbox"/>	
A	Nocturno	100	FISICA 2	FB002		Terça-feira 19:00	Quinta-feira 19:00	<input checked="" type="checkbox"/>	
B	Diurno	100	FISICA 1	FB001		Terça-feira 14:00	Quinta-feira 14:00	<input checked="" type="checkbox"/>	
A	Nocturno	100	FISICA 1	FB001		Terça-feira 19:00	Quinta-feira 19:00	<input checked="" type="checkbox"/>	
B	Diurno	100	CALCULO DE PROBABILIDADES	EST001		Terça-feira 14:00	Quinta-feira 14:00	<input checked="" type="checkbox"/>	
A	Nocturno	100	CALCULO DE PROBABILIDADES	EST001		Terça-feira 19:00	Quinta-feira 19:00	<input checked="" type="checkbox"/>	
B	Diurno	50	LINGUAGENS FORMAIS E AUTOMA	DC003		Terça-feira 14:00	Quinta-feira 14:00	<input checked="" type="checkbox"/>	
A	Nocturno	50	LINGUAGENS FORMAIS E AUTOMA	DC003		Terça-feira 19:00	Quinta-feira 19:00	<input checked="" type="checkbox"/>	
B	Diurno	50	INFORMÁTICA NA EDUCAÇÃO	EADCC038		Terça-feira 21:00	Quarta-feira 21:00	<input checked="" type="checkbox"/>	
A	Nocturno	50	INFORMÁTICA NA EDUCAÇÃO	EADCC038		Terça-feira 21:00	Quarta-feira 21:00	<input checked="" type="checkbox"/>	
B	Diurno	50	LABORATORIO DE PROGRAMACAO II	DC019		Quarta-feira 19:00	Quinta-feira 19:00	<input checked="" type="checkbox"/>	
A	Nocturno	50	LABORATORIO DE PROGRAMACAO II	DC019		Quarta-feira 19:00	Quinta-feira 19:00	<input checked="" type="checkbox"/>	
B	Diurno	50	ALGORITMOS	DC019		Terça-feira 08:00	Quinta-feira 14:00	<input checked="" type="checkbox"/>	
A	Nocturno	50	ALGORITMOS	DC019		Terça-feira 14:00	Quinta-feira 14:00	<input checked="" type="checkbox"/>	
B	Diurno	50	ALGORITMOS	DC019		Segunda-feira 14:00	Quarta-feira 14:00	<input checked="" type="checkbox"/>	
A	Nocturno	50	ALGORITMOS	DC019		Segunda-feira 14:00	Quarta-feira 14:00	<input checked="" type="checkbox"/>	
B	Diurno	50	ALGORITMOS	DC019		Segunda-feira 08:00	Quarta-feira 08:00	<input checked="" type="checkbox"/>	
A	Nocturno	50	ALGORITMOS	DC019		Segunda-feira 08:00	Quarta-feira 08:00	<input checked="" type="checkbox"/>	
B	Diurno	50	TEORIA DA COMPUTACAO	DC055		Terça-feira 10:00	Quarta-feira 16:00	<input checked="" type="checkbox"/>	
A	Nocturno	50	TEORIA DA COMPUTACAO	DC055		Terça-feira 10:00	Quarta-feira 16:00	<input checked="" type="checkbox"/>	
B	Diurno	50	LINGUAGEM DE PROGRAMACAO	DC019		Terça-feira 08:00	Quinta-feira 14:00	<input checked="" type="checkbox"/>	
A	Nocturno	50	LINGUAGEM DE PROGRAMACAO	DC019		Terça-feira 08:00	Quinta-feira 14:00	<input checked="" type="checkbox"/>	
B	Diurno	50	ANALISE E PROJETO DE ALGORIT	DC001		Terça-feira 08:00	Quinta-feira 10:00	<input checked="" type="checkbox"/>	
A	Nocturno	50	ANALISE E PROJETO DE ALGORIT	DC001		Terça-feira 08:00	Quinta-feira 10:00	<input checked="" type="checkbox"/>	
B	Diurno	50	INTRODUÇÃO A SISTEMAS DE INFO	DC013		Terça-feira 19:00	Quarta-feira 19:00	<input checked="" type="checkbox"/>	
A	Nocturno	50	INTRODUÇÃO A SISTEMAS DE INFO	DC013		Terça-feira 19:00	Quarta-feira 19:00	<input checked="" type="checkbox"/>	
B	Diurno	50	ASPECTOS ORGANIZACIONAIS DE	DC045		Terça-feira 19:00	Quarta-feira 21:00	<input checked="" type="checkbox"/>	
A	Nocturno	50	ASPECTOS ORGANIZACIONAIS DE	DC045		Terça-feira 19:00	Quarta-feira 21:00	<input checked="" type="checkbox"/>	
B	Diurno	50	SEMINARIO EM COMPUTACAO VI	DC002		Quarta-feira 08:00	Quinta-feira 08:00	<input checked="" type="checkbox"/>	
A	Nocturno	50	SEMINARIO EM COMPUTACAO VI	DC002		Quarta-feira 08:00	Quinta-feira 08:00	<input checked="" type="checkbox"/>	
B	Diurno	50	REDES DE COMPUTADORES	DC042		Quarta-feira 14:00	Segunda-feira 08:00	<input checked="" type="checkbox"/>	
A	Nocturno	50	REDES DE COMPUTADORES	DC042		Quarta-feira 14:00	Segunda-feira 08:00	<input checked="" type="checkbox"/>	
B	Diurno	50	ADMINISTRACAO DE REDES	DC014		Segunda-feira 08:00	Quarta-feira 19:00	<input checked="" type="checkbox"/>	
A	Nocturno	50	ADMINISTRACAO DE REDES	DC014		Segunda-feira 08:00	Quarta-feira 19:00	<input checked="" type="checkbox"/>	
B	Diurno	50	TOPICOS EM COMPUTACAO CEN	DC090		Terça-feira 10:00	Segunda-feira 08:00	<input checked="" type="checkbox"/>	
A	Nocturno	50	TOPICOS EM COMPUTACAO CEN	DC090		Terça-feira 10:00	Segunda-feira 08:00	<input checked="" type="checkbox"/>	
B	Diurno	50	SOLUCOES NUMERICAS DE EQUAC	DC190		Terça-feira 19:00	Segunda-feira 19:00	<input checked="" type="checkbox"/>	
A	Nocturno	50	SOLUCOES NUMERICAS DE EQUAC	DC190		Terça-feira 19:00	Segunda-feira 19:00	<input checked="" type="checkbox"/>	
B	Diurno	50	RESOLUCAO OPERACIONAL	DC008		Terça-feira 19:00	Segunda-feira 19:00	<input checked="" type="checkbox"/>	
A	Nocturno	50	RESOLUCAO OPERACIONAL	DC008		Terça-feira 19:00	Segunda-feira 19:00	<input checked="" type="checkbox"/>	
B	Diurno	50	CALCULO NUMERICO	DC008		Terça-feira 12:00	Sexta-feira 14:00	<input checked="" type="checkbox"/>	
A	Nocturno	50	CALCULO NUMERICO	DC008		Terça-feira 12:00	Sexta-feira 14:00	<input checked="" type="checkbox"/>	
B	Diurno	50	CALCULO NUMERICO	DC008		Quarta-feira 08:00	Segunda-feira 08:00	<input checked="" type="checkbox"/>	
A	Nocturno	50	CALCULO NUMERICO	DC008		Quarta-feira 08:00	Segunda-feira 08:00	<input checked="" type="checkbox"/>	
B	Diurno	50	CALCULO NUMERICO	DC008		Terça-feira 14:00	Quarta-feira 17:00	<input checked="" type="checkbox"/>	
A	Nocturno	50	CALCULO NUMERICO	DC008		Terça-feira 14:00	Quarta-feira 17:00	<input checked="" type="checkbox"/>	
B	Diurno	50	ASPECTOS LEGAIS DA INFORMATI	EADCC049		Segunda-feira 10:00	Terça-feira 14:00	<input checked="" type="checkbox"/>	
A	Nocturno	50	ASPECTOS LEGAIS DA INFORMATI	EADCC049		Segunda-feira 10:00	Terça-feira 14:00	<input checked="" type="checkbox"/>	
B	Diurno	50	TEORIA DOS GRAFOS	DC059		Terça-feira 12:00	Quarta-feira 08:00	<input checked="" type="checkbox"/>	
A	Nocturno	50	TEORIA DOS GRAFOS	DC059		Terça-feira 12:00	Quarta-feira 08:00	<input checked="" type="checkbox"/>	
B	Diurno	50	REDES NEURAS ARTIFICIAIS	DC066		Segunda-feira 21:00	Sexta-feira 21:00	<input checked="" type="checkbox"/>	
A	Nocturno	50	REDES NEURAS ARTIFICIAIS	DC066		Segunda-feira 21:00	Sexta-feira 21:00	<input checked="" type="checkbox"/>	

Figura 3.18: Turmas cadastradas.

Nome	Plano Departamental	Resultado	Cadastro de Pedidos	Pedidos Coordenadores	Turmas	Docente	Disciplinas	Cursos	Calendario
Código	Turno	Maximo de Vagas	Nome da Disciplina	Código da Disciplina	Sala	Dia	Dia	Horario Fixo	
B	Diurno	100	TEORIA DOS NUMEROS	MAT007		Quarta-feira 14:00	Sexta-feira 14:00	<input checked="" type="checkbox"/>	
A	Nocturno	100	TEORIA DOS NUMEROS	MAT007		Quarta-feira 19:00	Sexta-feira 19:00	<input checked="" type="checkbox"/>	
B	Diurno	100	GEOMETRIA ANALITICA	MAT004		Quarta-feira 14:00	Sexta-feira 14:00	<input checked="" type="checkbox"/>	
A	Nocturno	100	GEOMETRIA ANALITICA	MAT004		Quarta-feira 19:00	Sexta-feira 19:00	<input checked="" type="checkbox"/>	
B	Diurno	100	EQUAÇÕES DIFERENCIAIS	MAT006		Quarta-feira 14:00	Sexta-feira 14:00	<input checked="" type="checkbox"/>	
A	Nocturno	100	EQUAÇÕES DIFERENCIAIS	MAT006		Quarta-feira 19:00	Sexta-feira 19:00	<input checked="" type="checkbox"/>	
B	Diurno	100	CALCULO 3	MAT003		Quarta-feira 14:00	Sexta-feira 14:00	<input checked="" type="checkbox"/>	
A	Nocturno	100	CALCULO 3	MAT003		Quarta-feira 19:00	Sexta-feira 19:00	<input checked="" type="checkbox"/>	
B	Diurno	100	CALCULO 2	MAT002		Quarta-feira 14:00	Sexta-feira 14:00	<input checked="" type="checkbox"/>	
A	Nocturno	100	CALCULO 2	MAT002		Quarta-feira 19:00	Sexta-feira 19:00	<input checked="" type="checkbox"/>	
B	Diurno	100	CALCULO 1	MAT001		Quarta-feira 14:00	Sexta-feira 14:00	<input checked="" type="checkbox"/>	
A	Nocturno	100	CALCULO 1	MAT001		Quarta-feira 19:00	Sexta-feira 19:00	<input checked="" type="checkbox"/>	
B	Diurno	100	ALGEBRA LINEAR	MAT005		Terça-feira 14:00	Quinta-feira 14:00	<input checked="" type="checkbox"/>	
A	Nocturno	100	ALGEBRA LINEAR	MAT005		Terça-feira 19:00	Quinta-feira 19:00	<input checked="" type="checkbox"/>	
B	Diurno	100	LABORATORIO DE FISICA	FB004		Segunda-feira 21:00	Quarta-feira 21:00	<input checked="" type="checkbox"/>	
A	Nocturno	100	LABORATORIO DE FISICA	FB004		Segunda-feira 21:00	Quarta-feira 21:00	<input checked="" type="checkbox"/>	
B	Diurno	100	FISICA 3	FB003		Terça-feira 14:00	Quinta-feira 14:00	<input checked="" type="checkbox"/>	
A	Nocturno	100	FISICA 3	FB003		Terça-feira 19:00	Quinta-feira 19:00	<input checked="" type="checkbox"/>	
B	Diurno	100	FISICA 2	FB002		Terça-feira 14:00	Quinta-feira 14:00	<input checked="" type="checkbox"/>	
A	Nocturno	100	FISICA 2	FB002		Terça-feira 19:00	Quinta-feira 19:00	<input checked="" type="checkbox"/>	
B	Diurno	100	FISICA 1	FB001		Terça-feira 14:00	Quinta-feira 14:00	<input checked="" type="checkbox"/>	
A	Nocturno	100	FISICA 1	FB001		Terça-feira 19:00	Quinta-feira 19:00	<input checked="" type="checkbox"/>	
B	Diurno	100	CALCULO DE PROBABILIDADES	EST001		Terça-feira 14:00	Quinta-feira 14:00	<input checked="" type="checkbox"/>	
A	Nocturno	100	CALCULO DE PROBABILIDADES	EST001		Terça-feira 19:00	Quinta-feira 19:00	<input checked="" type="checkbox"/>	
B	Diurno	50	LINGUAGENS FORMAIS E AUTOMA	DC003		Terça-feira 14:00	Quinta-feira 14:00	<input checked="" type="checkbox"/>	
A	Nocturno	50	LINGUAGENS FORMAIS E AUTOMA	DC003		Terça-feira 19:00	Quinta-feira 19:00	<input checked="" type="checkbox"/>	
B	Diurno	50	INFORMÁTICA NA EDUCAÇÃO	EADCC038		Terça-feira 21:00	Quarta-feira 21:00	<input checked="" type="checkbox"/>	
A	Nocturno	50	INFORMÁTICA NA EDUCAÇÃO	EADCC038		Terça-feira 21:00	Quarta-feira 21:00	<input checked="" type="checkbox"/>	
B	Diurno	50	LABORATORIO DE PROGRAMACAO II	DC019		Quarta-feira 19:00	Quinta-feira 19:00	<input checked="" type="checkbox"/>	
A	Nocturno	50	LABORATORIO DE PROGRAMACAO II	DC019		Quarta-feira 19:00	Quinta-feira 19:00	<input checked="" type="checkbox"/>	
B	Diurno	50	ALGORITMOS	DC019		Terça-feira 08:00	Quinta-feira 14:00	<input checked="" type="checkbox"/>	
A	Nocturno	50	ALGORITMOS	DC019		Terça-feira 14:00	Quinta-feira 14:00	<input checked="" type="checkbox"/>	
B	Diurno	50	ALGORITMOS	DC019		Segunda-feira 14:00	Quarta-feira 14:00	<input checked="" type="checkbox"/>	
A	Nocturno	50	ALGORITMOS	DC019		Segunda-feira 14:00	Quarta-feira 14:00	<input checked="" type="checkbox"/>	
B	Diurno	50	ALGORITMOS	DC019		Segunda-feira 08:00	Quarta-feira 08:00	<input checked="" type="checkbox"/>	
A	Nocturno	50	ALGORITMOS	DC019		Segunda-feira 08:00	Quarta-feira 08:00	<input checked="" type="checkbox"/>	
B	Diurno	50	TEORIA DA COMPUTACAO	DC055		Terça-feira 10:00	Quarta-feira 16:00	<input checked="" type="checkbox"/>	
A	Nocturno	50	TEORIA DA COMPUTACAO	DC055		Terça-feira 10:00	Quarta-feira 16:00	<input checked="" type="checkbox"/>	
B	Diurno	50	LINGUAGEM DE PROGRAMACAO	DC019		Terça-feira 08:00	Quinta-feira 14:00	<input checked="" type="checkbox"/>	
A	Nocturno	50	LINGUAGEM DE PROGRAMACAO	DC019		Terça-feira 08:00	Quinta-feira 14:00	<input checked="" type="checkbox"/>	
B	Diurno	50	ANALISE E PROJETO DE ALGORIT	DC001		Terça-feira 08:00	Quinta-feira 10:00	<input checked="" type="checkbox"/>	
A	Nocturno	50	ANALISE E PROJETO DE ALGORIT	DC001		Terça-feira 08:00	Quinta-feira 10:00	<input checked="" type="checkbox"/>	
B	Diurno	50	INTRODUÇÃO A SISTEMAS DE INFO	DC013		Terça-feira 19:00	Quarta-feira 19:00	<input checked="" type="checkbox"/>	
A	Nocturno	50	INTRODUÇÃO A SISTEMAS DE INFO	DC013		Terça-feira 19:00	Quarta-feira 19:00	<input checked="" type="checkbox"/>	
B	Diurno	50	ASPECTOS ORGANIZACIONAIS DE	DC045		Terça-feira 19:00	Quarta-feira 21:00	<input checked="" type="checkbox"/>	
A	Nocturno	50	ASPECTOS ORGANIZACIONAIS DE	DC045		Terça-feira 19:00	Quarta-feira 21:00	<input checked="" type="checkbox"/>	
B	Diurno	50	SEMINARIO EM COMPUTACAO VI	DC002		Quarta-feira 08:00	Quinta-feira 08:00		

Clc/Idigo	Nome	Perfi
DCC001	ANALISE E PROJETO DE ALGORITMOS	12.TEO
DCC008	CALCULO NUMERICO	09.NUMOC
DCC012	ESTRUTURA DE DADOS II	02.AVAN
DCC013	ESTRUTURA DE DADOS	02.AVAN
DCC014	INTELIGENCIA ARTIFICIAL	07.IAC
DCC019	LINGUAGEM DE PROGRAMACHO	12.TEO
DCC025	ORIENTADO A OBJETOS	06.ES
DCC042	REDES DE COMPUTADORES	10.REDES
DCC055	TEORIA DA COMPUTAÇÃO	13.TEO
DCC059	TEORIA DOS GRAFOS	07.IAC
DCC060	BANCO DE DADOS	04.BD
DCC061	ENGENHARIA DE SOFTWARE	06.ES
DCC062	SISTEMAS OPERACIONAIS	03.ARSO
DCC063	LINGUAGEM FORTRAN E ALGAMATOS	13.TEO
DCC065	COMPUTAÇÃO GRAFICA	06.CO
DCC066	PROCESSAMENTO DE IMAGENS	05.CO
DCC067	COMPUTAÇÃO EVOLUCIONISTA	07.IAC
DCC068	REDES NEURAS ARTIFICIAS	07.IAC
DCC070	ORGANIZAO DE COMPUTADORES	03.ARSO
DCC071	LABORATORIO DE SISTEMAS OPERACIONAIS	03.ARSO
DCC072	LABORATORIO DE REDES DE COMPUTADORES	10.REDES
DCC074	AVALIAO DE DESEMPENHO DE SISTEMAS	10.REDES
DCC077	ASPECTOS AVANÇADOS EM BANCO DE DADOS	04.BD
DCC090	TOPICOS EM COMPUTAÇÃO CIENTIFICA II	09.NUMOC
DCC102	SEMINARIO EM COMPUTAÇÃO VI	10.REDES
DCC104	SEMINARIO EM COMPUTAÇÃO VIII	02.AVAN
DCC107	LABORATORIO DE PROGRAMAÇÃO II	02.AVAN
DCC117	MODELAGEM DE SISTEMAS	06.ES
DCC118	ALGORITMOS	01.BASICO
DCC120	LABORATORIO DE PROGRAMAÇÃO	01.BASICO
DCC121	LABORATORIO DE PROGRAMAÇÃO WEB	02.AVAN
DCC122	CIRCUITOS DIGITAIS	03.ARSO
DCC123	METODOLOGIA CIENTIFICA EM COMPUTAÇÃO	01.BASICO
DCC125	PROGRAMAÇÃO PARALELA	03.ARSO
DCC133	INTRODUÇÃO A SISTEMAS DE INFORMAÇÃO	11.SI
DCC136	INTELIGENCIA COMPUTACIONAL	07.IAC
DCC144	ADMINISTRAO DE REDES	10.REDES
DCC145	ASPECTOS ORGANIZACIONAIS DE SISTEMAS DE INFORMAO	11.SI
DCC156	LABORATORIO DE PROGRAMAÇÃO IV	02.AVAN
DCC160	LOGICA E FUNDAMENTOS PARA A COMPUTAÇÃO	07.IAC
DCC163	PERGUNTA OPERACIONAL	09.NUMOC
DCC164	RECUPERAO DA INFORMAO	04.BD
DCC166	SISTEMAS DE APOIO E DECISAO	04.BD
DCC174	INTERAO HUMANO-COMPUTADOR	09.ES
DCC180	SOLUÇO NUMÉRICA DE EQUAÇÕES DIFERENCIAIS	09.NUMOC
DCC181	VERIFICAÇÃO CIENTIFICA	05.CO
EAD000003	LOGICA PARA COMPUTAÇÃO	07.IAC
EAD000006	INTRODUÇÃO AOS SISTEMAS OPERACIONAIS	03.ARSO
EAD000007	FUNDAMENTOS DA ORGANIZAO DE COMPUTADORES	03.ARSO
EAD000025	MODELAGEM DE SISTEMAS	06.ES
EAD000031	LINGUAGEM DE PROGRAMAÇÃO II	02.AVAN
EAD000035	PROGRAMAÇÃO PARA WEB II	02.AVAN
EAD000039	INTERFACE USUÁRIO E MQUINA	06.ES
EAD000037	PROJETO DE SOFTWARE EDUCACIONAL	06.ES

Figura 3.20: Disciplinas cadastradas.

Nome	Clc/Idigo	Turno	Calouros no Primeiro Semestre	Calouros no Segundo Semestre
Sistema de Informação	45A	Nocturno	50	45
Eng. Computacional	45A	Diurno	50	45
Ciência da Computação	35A	Nocturno	50	50
Computação (Exatas)	65B	Diurno	50	45

Figura 3.21: Cursos Cadastrados.

## 4 Estudo de Caso

Este capítulo apresenta os resultados obtidos executando cada um dos algoritmos mencionados no Capítulo 3, a fim de validar as técnicas de Otimização por Colônia de Formigas e Otimização por Busca Harmônica no problema de geração de Grade de Horários. O foco do trabalho é a grade de horários dos cursos do DCC.

A grade de horários do DCC contempla um total de quatro cursos de graduação presencial. Dessa forma, tem-se um total de quatro grades a se respeitar ao definir os horários das turmas. Conta-se também com um total de 48 professores, 127 turmas e 58 disciplinas. Além dessas deve-se também respeitar os horários fixos de 21 turmas fornecidas por outros departamentos em que alguns cursos necessitam em sua grade. Esses foram os dados utilizados para o desenvolvimento da solução.

O software foi desenvolvido em Java e os algoritmos ACO, HSO, Roleta, Torneio, ProximoItemPermutado e Busca Local implementados em C++. Os experimentos foram executados em um computador com processador Intel Core I5-4570 CPU 3,2GHz, 8,00 GB RAM DDR3, AMD Radeon R9 280x e sistema operacional Windows 10 Pro 64 bits.

Para realizar os experimentos foi necessário uma base que informasse a proximidade dos resultados obtidos pelos algoritmos descritos no Capítulo 3 com os utilizados na prática. Para isso, utilizou-se o Plano Departamental do Departamento de Ciência da Computação do primeiro semestre letivo de 2017. O fato de ser uma base real fornece um comparativo com os dados utilizados atualmente.

Antes, porém, realizou-se uma calibragem dos parâmetros apresentados no Capítulo 3 e presentes na Tabela 4.1. Os algoritmos foram executados de maneira intermitente durante sete dias verificando-se as possíveis combinações dos valores de cada parâmetro. A combinação que obteve melhores resultados é descrita nessa tabela.

Os oito primeiros parâmetros contidos na Tabela 4.1 são utilizados nas equações responsáveis pela atualização de feromônios dispostas na Seção 3.2.1. Esses são:  $\rho$  com valor 0,2,  $\gamma$  0,2,  $\omega$  0,8,  $V_0$  100,  $V_1$  1000,  $U_0$  200 e  $U_1$  com valor 2000.  $\epsilon$  indica, no algoritmo ProximoItemPermutado, se a ordenação realizada, com o intuito de retornar o

Tabela 4.1: Valores dos parâmetros

Parâmetro	Valor
$\omega$	0,8
$\rho$	0,2
$\gamma$	0,2
$\epsilon$	0,1
$V_0$	100
$V_1$	1000
$U_0$	200
$U_1$	2000
$C$	95%
Número de estagnações para a normalização de feromônios	500
Quantidade de repetições da busca local	50
HMCR	0,7
PAR	0,3
Número de formigas	50
Tamanho da Memória Harmônica	20
Tamanho do torneio	3

horário solicitado, ocorre de forma randômica e possui valor 0,1. A porcentagem  $C$  de se escolher a solução com melhor nível de feromônios no torneio é de 95%. Ao realizar 500 iterações sem melhora na melhor solução encontrada pelo ACO, realiza-se a normalização dos feromônios, a fim de evitar ótimos locais.

O HSO possui  $HMCR$ , responsável por indicar se utiliza-se a Memória Harmônica para gerar a solução, possui valor 0,7.  $PAR$  indica a probabilidade de utilizar as funções  $Move()$ ,  $TrocaHoráriosLocais()$  e  $TrocaHoráriosHM()$  no horário definido pela HM e possui valor 0,3.

A Busca Local, Figura 3.12, possui um total de 50 repetições. O número de formigas utilizadas no ACO foi 50, o tamanho na Memória Harmônica 20 e a quantidade de horários competidores no torneio são 3.

## 4.1 Resultados Obtidos

Com todos os parâmetros calibrados iniciou-se os experimentos dos Algoritmos descritos no Capítulo 3. Para isso, utilizou-se como base a Tabela 4.2 com todas as combinações testadas. As combinações 1, 2, 3 e 7 são os algoritmos sem hibridização e Busca Local, variando somente a forma de construção da solução no ACO, por exemplo, entre a Roleta

e o Torneio. Já 4, 5, 6 e 14 são os algoritmos ACO e HSO em conjunto com a Busca Local. A partir da combinação 8 tem-se as hibridizações dos algoritmos “Otimização por Colônia de Formigas com a Otimização por Busca Harmônica”, sendo essas as combinações 8, 9 e 10. Por fim, em 11, 12 e 13 têm-se as hibridizações de ACO e HSO com a utilização da Busca Local.

Os resultados referentes às combinações encontram-se na Tabela 4.3. Nesta tem-se a quantidade de iterações realizadas para obter a solução, a quantidade de penalizações fortes e fracas violadas, e o tempo gasto para executar esse número de iterações.

É possível observar na Tabela 4.3 que a combinação responsável pelos melhores resultados foi a de número 12, ACO com Roleta, Busca Local e HSO, alcançando uma solução viável. Essa foi a única capaz de zerar as restrições fortes, mas não conseguiu zerar as leves. Vale enfatizar que pelo fato da roleta utilizar uma estratégia com números randômicos, nem sempre se alcançou valor zero no HCP.

Observa-se também que soluções que envolveram o método da roleta obtiveram resultados mais satisfatórios, como, por exemplo, a combinação 5, obtendo o segundo melhor resultado.

Nenhuma outra combinação, além da 12, conseguiu alcançar resultados que são aptos a serem aplicados ao DCC, ou seja, resultados com penalizações fortes zeradas.

Ainda assim é possível utilizá-los para avaliar alguns pontos importantes. É possível perceber que a Busca Local possui uma grande influência na melhora da solução, devido a suas trocas de horários serem extremamente eficientes quando várias penalizações são violadas. Mas no momento em que o resultado possui poucas violações, a mesma torna-se pouco eficiente. O HSO auxilia a zerar as penalidades fortes quando as mesmas encontram-se próximas de zero. Seu fator aleatório aumenta a probabilidade de se explorar uma vizinhança, antes não visitada, e encontrar novas soluções.

Ao analisar os resultados utilizando o quesito tempo pode-se perceber que a combinação 12, apesar de gastar um dos maiores períodos de tempo para convergir, obteve a melhor solução. Esse tempo gasto é totalmente viável para o planejamento das grades horárias do DCC.

A Tabela 4.4 é uma solução obtida pela Combinação 13. Essa exhibe o quarto



Tabela 4.2: Combinações de algoritmos utilizados.

<b>Combinações</b>	<b>Algoritmos Utilizados</b>
Combinação 1	ACO com ProximoItemPermutado
Combinação 2	ACO com Roleta
Combinação 3	ACO com Torneio
Combinação 4	ACO com ProximoItemPermutado e Busca Local
Combinação 5	ACO com Roleta e Busca Local
Combinação 6	ACO com Torneio e Busca Local
Combinação 7	HSO
Combinação 8	ACO com ProximoItemPermutado e HSO
Combinação 9	ACO com Roleta e HSO
Combinação 10	ACO com Torneio e HSO
Combinação 11	ACO com ProximoItemPermutado, Busca Local e HSO
Combinação 12	ACO com Roleta, Busca Local e HSO
Combinação 13	ACO com Torneio, Busca Local e HSO
Combinação 14	HSO e Busca Local

Tabela 4.3: Resultados obtidos.

<b>Método</b>	<b>Número de iterações</b>	<b>HCP</b>	<b>SCP</b>	<b>Tempo(s)</b>
Combinação 1	20000	52	4	404
Combinação 2	20000	23	5	981
Combinação 3	20000	111	3	414
Combinação 4	20000	33	6	1825
Combinação 5	20000	3	8	1668
Combinação 6	20000	105	5	1895
Combinação 7	20000	24	3	31
Combinação 8	20000	24	1	416
Combinação 9	20000	23	5	1017
Combinação 10	20000	23	2	397
Combinação 11	2000	20	9	1825
Combinação 12	2000	0	3	1768
Combinação 13	2000	23	3	1859
Combinação 14	20000	20	6	986

período do Curso de Sistema de Informação do DCC. Utilizou-se o exemplo em questão devido a violação de penalidades fortes existentes na mesma. Nota-se que ambas as disciplinas DCC042 e DCC012, em itálico, violaram a restrição forte referente ao turno da disciplina, alocando-as durante horários diurnos.

A Figura 4.1 analisa as penalizações dos resultados encontrados com o passar do tempo executando-se a Combinação 12. O HCP é representado na vertical do gráfico e o tempo na horizontal. Observa-se que a convergência segue um determinado padrão. Inicialmente, a solução converge de forma rápida, diminuindo bruscamente as restrições infringidas. Quando a quantidade de restrições alcança o valor quatro ou menor o gráfico

Tabela 4.4: Período com violações de restrições fortes.

Sistema de Informação	Solução Proposta				
Período:	4				
	Seg	Ter	Qua	Qui	Sex
08:00					
09:00					
10:00					
11:00					
12:00					DCC012
13:00					DCC012
14:00			DCC042	DCC012	
15:00			DCC042	DCC012	
16:00					
17:00					
18:00					
19:00	DCC117		DCC070	DCC060	DCC060
20:00	DCC117		DCC070	DCC060	DCC060
21:00				DCC117	DCC070
22:00				DCC117	DCC070

continua a decrementar, mas de forma sutil, até alcançar duas violações. Após isso, o algoritmo consegue zerar o HCP, mas para isso necessita uma demanda maior de tempo.

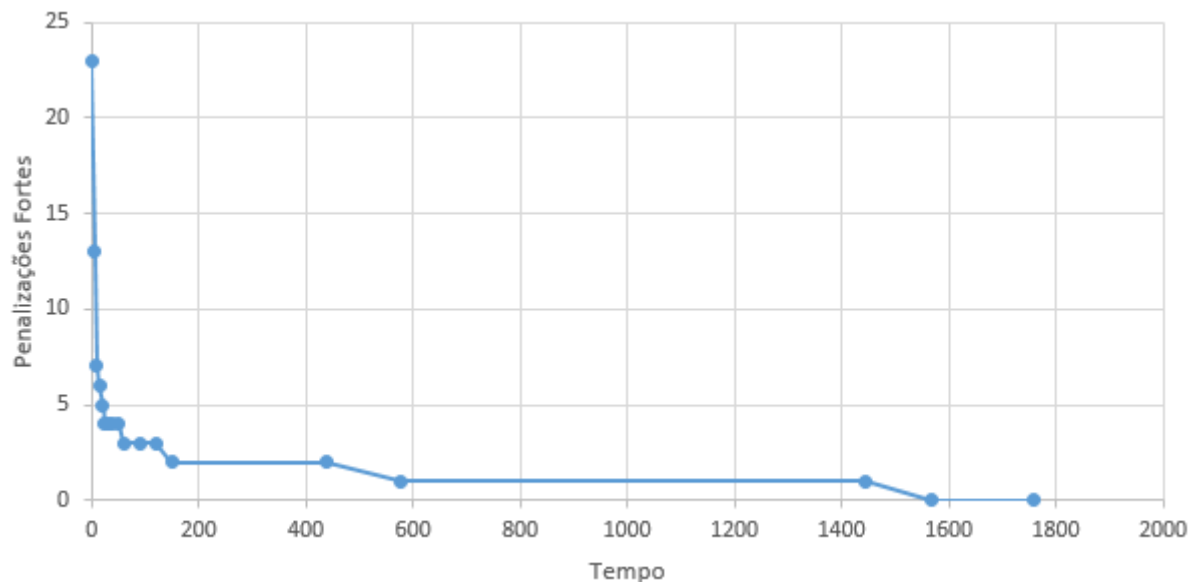


Figura 4.1: Gráfico de convergência de penalizações fortes no algoritmo ACO com Roleta, Busca Local e HSO.

A evolução do SCP é indicada na Figura 4.2. Essa é análoga a Imagem 4.1, com a exceção que na vertical do gráfico é indicado as violações de restrições leves. Nota-se que o tempo inicial não possui valor zero, isso porque as restrições fracas só são contabilizadas

no último fator forte encontrado. Por exemplo, no momento em que é alterado o valor das penalidades fortes, as penalidades fracas também são alteradas. Isso ocorre, pois, as penalidades fortes sempre prevalecerem sobre as fracas. Assim, o SCP só é contabilizado na última restrição forte definida pela solução.

A partir do momento que o algoritmo alcança o valor zero no HCP, começa-se a analisar as penalidades fracas. Suas violações diminuem suavemente, ocorrendo poucas convergências de forma bruta. Nota-se que o valor zero nunca é alcançado. Devido ao fato do SCP ser uma restrição que mesmo com a violação, a solução mantém-se viável.

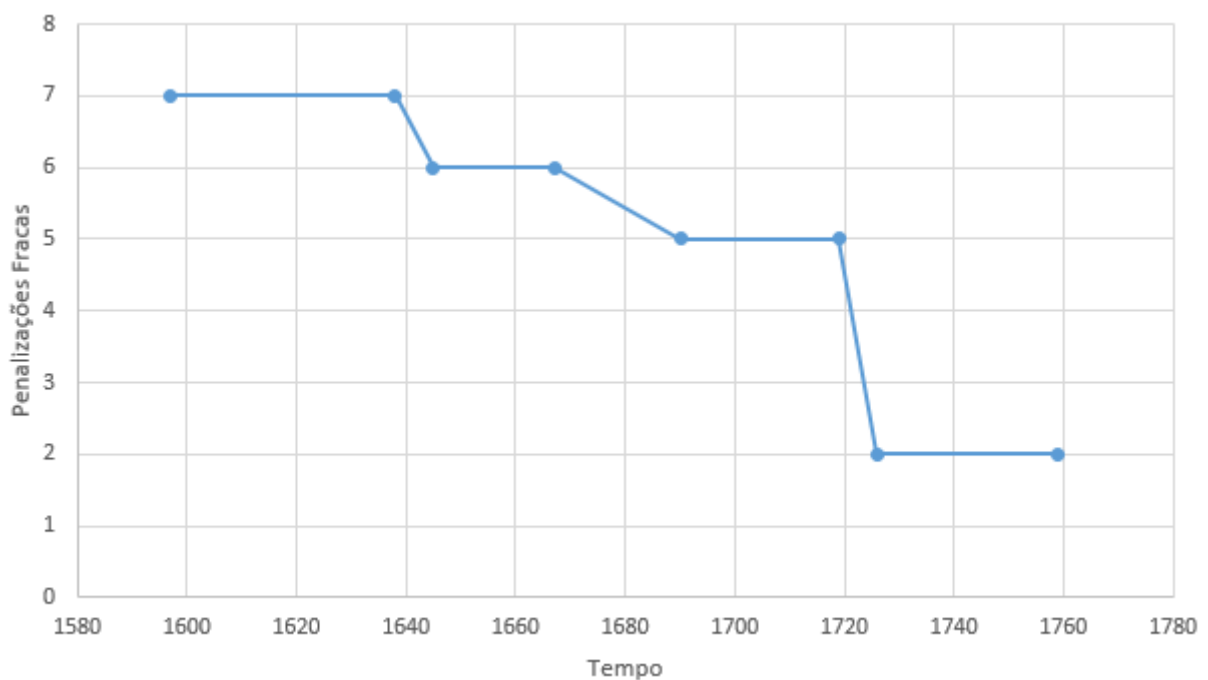


Figura 4.2: Gráfico de convergência de penalizações fracas no algoritmo ACO com Roleta, Busca Local e HSO.

Como forma de comparação e validação das soluções encontradas utilizou-se as grades dos cursos de Ciência da Computação e Sistema de Informação oferecidos pelo DCC. Ambos os cursos são de períodos noturnos, e suas disciplinas encontram-se entre 19 e 23 horas.

As Tabelas 4.5 e 4.6 exibem os períodos pares e suas disciplinas dos cursos de Ciência da Computação Noturno e Sistema de Informação respectivamente. São exibidos somente os períodos pares, devido ao fator de ingresso destes cursos, que ocorre sempre no segundo semestre, dessa forma, os períodos pares são cursados no primeiro semestre e os ímpares no segundo.

Ciência da Computação Noturno										
	Solução Proposta					Grade Atual				
Período:	2									
Horário	Seg	Ter	Qua	Qui	Sex	Seg	Ter	Qua	Qui	Sex
18:00										
19:00	DCC013	FIS073	MAT156	FIS073	DCC107	DCC107	FIS073	MAT156	FIS073	DCC013
20:00	DCC013	FIS073	MAT156	FIS073	DCC107	DCC107	FIS073	MAT156	FIS073	DCC013
21:00	FIS077	MAT158	MAT158	DCC013	MAT156	FIS077	MAT158	MAT158	DCC013	MAT156
22:00	FIS077	MAT158	MAT158	DCC013	MAT156	FIS077	MAT158	MAT158	DCC013	MAT156
Período:	4									
Horário	Seg	Ter	Qua	Qui	Sex	Seg	Ter	Qua	Qui	Sex
18:00										
19:00	DCC117	FIS075	DCC070	FIS075	MAT029	DCC012	FIS075	DCC070	FIS075	MAT029
20:00	DCC117	FIS075	DCC070	FIS075	MAT029	DCC012	FIS075	DCC070	FIS075	MAT029
21:00	DCC012	DCC012	MAT029	DCC117	DCC070	DCC070	DCC117	MAT029	DCC012	DCC117
22:00	DCC012	DCC012	MAT029	DCC117	DCC070	DCC070	DCC117	MAT029	DCC012	DCC117
Período:	6									
Horário	Seg	Ter	Qua	Qui	Sex	Seg	Ter	Qua	Qui	Sex
18:00										
19:00	DCC174	DCC063	DCC163	DCC042	DCC063	DCC063	DCC042		DCC174	DCC042
20:00	DCC174	DCC063	DCC163	DCC042	DCC063	DCC063	DCC042		DCC174	DCC042
21:00		DCC174	DCC042		DCC163	DCC163	DCC063	DCC174	DCC163	
22:00		DCC174	DCC042		DCC163	DCC163	DCC063	DCC174	DCC163	

Tabela 4.5: Comparação entre o resultado feito manualmente e o obtido pelo algoritmo para o curso de Ciência da Computação.

Sistema de Informação										
	Solução Proposta					Grade Atual				
Período:	2									
Horário	Seg	Ter	Qua	Qui	Sex	Seg	Ter	Qua	Qui	Sex
18:00										
19:00	DCC013		MAT156	CAD076	DCC107	DCC107		MAT156	CAD076	DCC013
20:00	DCC013		MAT156	CAD076	DCC107	DCC107		MAT156	CAD076	DCC013
21:00	CAD076		DCC121	DCC013	MAT156	CAD076		DCC121	DCC013	MAT156
22:00	CAD076		DCC121	DCC013	MAT156	CAD076		DCC121	DCC013	MAT156
Período:	4									
Horário	Seg	Ter	Qua	Qui	Sex	Seg	Ter	Qua	Qui	Sex
18:00										
19:00	DCC117	DCC042	DCC070	DCC060	DCC060	DCC012	DCC042	DCC070	DCC060	DCC042
20:00	DCC117	DCC042	DCC070	DCC060	DCC060	DCC012	DCC042	DCC070	DCC060	DCC042
21:00	DCC012	DCC012		DCC117	DCC070	DCC070	DCC117	DCC060	DCC012	DCC117
22:00	DCC012	DCC012		DCC117	DCC070	DCC070	DCC117	DCC060	DCC012	DCC117
Período:	6									
Horário	Seg	Ter	Qua	Qui	Sex	Seg	Ter	Qua	Qui	Sex
18:00										
19:00	DCC174	DCC166	DCC077	DCC156	DCC156	DCC077	DCC166	DCC156	DCC174	DCC156
20:00	DCC174	DCC166	DCC077	DCC156	DCC156	DCC077	DCC166	DCC156	DCC174	DCC156
21:00		DCC174		DCC166	DCC077		DCC077	DCC174	DCC166	
22:00		DCC174		DCC166	DCC077		DCC077	DCC174	DCC166	

Tabela 4.6: Comparação entre o resultado feito manualmente e o obtido pelo algoritmo para o curso de Sistema de Informação.

Apresenta-se na Tabela 4.5 a grade proposta pelo algoritmo (esquerda) e a definida pelo departamento (direita). Nessa tem-se os dias úteis da semana e os horários do

período noturno (19 às 23 horas). Comparando ambas é possível visualizar uma grande semelhança em horários, devido às disciplinas com horários fixos. Também é observado diferenças como, por exemplo, no quinto período, em que houve uma otimização por parte do algoritmo nos horários e dias das disciplinas, não alocando turmas às 21h, quando essa é a única cursada no dia, o que normalmente é uma preferência dos alunos.

A Tabela 4.6 é análoga a 4.5, mudando somente o curso representado. Ao comparar as duas tabelas pode-se observar semelhanças entre horários de disciplinas cursadas por ambos os cursos, mesmo quando essas encontram-se em períodos opostos, como por exemplo a DCC013. Isso indica que o algoritmo respeitou a terceira restrição forte, mencionada na Seção 3.1, “Garantir que turmas que estão no mesmo período de uma determinada grade não coincidam seus horários”. Na solução as turmas que influenciam períodos distintos dos cursos são encaixadas perfeitamente no horário, como por exemplo a DCC174.

Ao observar a Tabela 4.7 percebe-se que os horários do Docente P4 foram respeitados. Essa contém todos os horários aptos a conter turmas e os espaços de tempo de cor cinza, que são os que o professor prefere não lecionar.

Docente: P4	Grade Proposta				
	Seg	Ter	Qua	Qui	Sex
08:00					
09:00					
10:00					
11:00					
12:00					
13:00					
14:00					
15:00					
16:00					
17:00					
18:00					
19:00		DCC166		DCC060	
20:00		DCC166		DCC060	
21:00			DCC060	DCC166	
22:00			DCC060	DCC166	

Tabela 4.7: Horários definidos para o Docente P4.

A solução utilizada para comparação não violou nenhuma restrição forte, mas violou algumas preferências de horários de professores, que são restrições leves.

Ao analisar a etapa de construção de soluções pelo algoritmo, percebeu-se várias dificuldades encontradas pelo mesmo durante o processo. Uma das principais é o fato de

alocar horários para turmas que satisfaçam várias grades, sem que uma comprometa a outra. Isso ocorre, pois a alteração de um horário pode gerar conflitos com outras grades, aumentando de forma significativa a quantidade de restrições fortes infringidas.

Com o passar do tempo, notou-se que o ACO e o HSO em conjuntos não eram o suficiente para gerar uma solução sem violar alguma restrição forte. Optou-se então pela adição de uma busca local. Após isso, o algoritmo obteve uma melhora significativa, alcançando-se soluções com penalizações fortes zeradas. As penalizações leves foram minimizadas e as pessoas responsáveis por ajustar o plano podem decidir sobre manter o plano com essas violações de restrições fracas se ela não for um problema para o DCC. Caso seja, o software pode sugerir outra solução para ser analisada novamente.

## 5 Conclusão e Trabalhos Futuros

Este trabalho propôs um método híbrido para a solução do problema de Alocação de Horário para o Departamento de Ciência da Computação da Universidade Federal de Juiz de Fora. O software desenvolvido e a otimização do plano departamental possuem o intuito de auxiliar a pessoa responsável pelo plano a encontrar a melhor solução. O software possui também a opção de exportar as diversas soluções encontradas para caso tenha-se a necessidade de compará-las ou editá-las de acordo com o desejado.

Ao verificar as soluções chega-se a conclusão que o resultado obtido pelo software alcançou o desejado, que é apresentar uma solução capaz de auxiliar o responsável a elaborar o plano final do departamento. Levando isto em consideração a utilização deste programa irá reduzir o esforço necessário durante a elaboração do mesmo.

Com a solução gerada pelo algoritmo, levando em consideração as restrições não violadas, estima-se o aumento da satisfação dos professores nos horários de suas aulas. Isso levando em consideração que o algoritmo preza por alocar horários desejados pelos docentes. Alunos também são beneficiados podendo cursar mais disciplinas e ter menos horários vagos.

Para trabalhos futuros é proposto a melhora no software e no algoritmo inteligente para tentar zerar todas as restrições leves, além das fortes, colocadas no sistema. Para isso propõe-se a alteração da busca local para outras mais robustas e melhores otimizadas. Além disso, propõe-se a inserção de novas restrições, com o intuito de dificultar o problema, e a utilização de um software responsável por calibrar os parâmetros.

## A Anexo I - Algoritmos

Nesta seção são apresentados os anexos referentes aos algoritmos explicados no capítulo 3.

---

### Algoritmo 1: Algoritmo Híbrido

---

```

1 s ← inicializa banco de soluções;
2 s ← executa ACO;
3 memória harmônica ← s;
4 executa HSO;
5 retorna as melhores soluções encontradas pelo HSO;
```

---



---

### Algoritmo 2: ACO

---

```

1 for  $i = 1 \dots N$  do
2   for cada formiga  $k = 1$  até  $m$  do
3     for cada evento do
4       | assinale a ele uma posição baseado nos feromônios;
5     end for
6   end for
7   for cada solução do
8     | aumenta o nível de feromônios;
9   end for
10  evaporação de feromônios;
11 end for
```

---



---

**Algoritmo 3:** HSO

---

```
1 for  $i = 1 \dots N$  do
2    $x =$  nova solução;
3   for  $i = 1 \dots E$  do
4     if  $U(0,1) \leq HMCR)$  then
5        $x_i =$  realize a consideração da memória;
6       if  $rnd \leq PAR1)$  then
7          $x_i =$  ajusta o tom move();
8       end if
9       if  $rnd \leq PAR2)$  then
10         $x_i =$  ajusta o tom trocaHorariosLocais();
11      end if
12      if  $rnd \leq PAR3)$  then
13         $x_i =$  ajusta o tom trocaHorariosMemóriaHarmonica();
14      end if
15    end if
16    else
17       $x_i =$  consideração randômica;
18    end if
19  end for
20  if  $x$  melhor que pior solução de  $HM$  then
21    atualiza  $HM$ ;
22  end if
23 end for
```

---

**Algoritmo 4: ProxItemPermutado(j)**


---

```

1  if  $j > pos$  then
2    for  $j' = pos \dots j$  do
3      if  $\sigma > \epsilon$  then
4         $rnd = \sigma * rand(0,1);$ 
5         $q = pos;$ 
6         $\xi = 0;$ 
7        while  $\xi < rnd$  and  $q < T$  do
8           $\xi = \xi + \omega_q^s;$ 
9           $q = q + 1;$ 
10       end while
11        $\sigma = \sigma - \omega_{q-1}^s;$ 
12        $swap(q - 1, j');$ 
13     end if
14     else
15        $swap(rnd, j');$ 
16     end if
17      $pos = pos + 1;$ 
18   end for
19 end if
20 return  $\omega_j^s;$ 

```

---

**Algoritmo 5: Roleta()**


---

```

1   $roleta \leftarrow vetor[];$ 
2  for  $i = 0 \dots numeroHorarios$  do
3     $totalFeromonios += \omega_i ;$ 
4  end for
5  for  $i = 0 \dots numeroHorarios$  do
6    if  $\omega_i < 1$  then
7       $roleta.adicionaUltimaPosicao() = i;$ 
8    end if
9    else
10      $qtdCasasRoleta = ceil(\omega_i * 100) / totalFeromonios;$ 
11     for  $i = 0 \dots numeroHorarios$  do
12        $roleta.adicionaUltimaPosicao() = \omega_i;$ 
13     end for
14   end if
15 end for
16  $rnd \leftarrow rand(0, roleta.tamanho);$ 
17 return  $roleta[rnd];$ 

```

---

---

**Algoritmo 6:** Torneio()

---

```
1 selecionado ← vetor[D];
2 for  $i = 0 \dots D$  do
3   | selecionado[i] = rand(0, numeroHorarios);
4 end for
5 ganhador = selecionado[0];
6 if  $\text{rand}(0,100) > C$  then
7   | for  $i = 1 \dots D$  do
8     | | if  $\omega_i > \omega_{\text{ganhador}}$  then
9       | | | ganhador = selecionado[i];
10    | | end if
11   | end for
12 end if
13 else
14   | for  $i = 1 \dots D$  do
15     | | if  $\omega_{\text{selecionado}[i]} < \omega_{\text{ganhador}}$  then
16       | | | ganhador = selecionado[i];
17     | | end if
18   | end for
19 end if
20 return ganhador;
```

---

---

**Algoritmo 7:** Busca Local(Soluções)

---

```
1 for  $i = 0 \dots N$  do
2   | busca em s duas turmas que violam restrições;
3   | troca o horário das turmas;
4   | if Não diminuiu as penalizações then
5     | | volta para o estado anterior as trocas;
6   | end if
7 end for
```

---

## Bibliografia

- AL-BETAR, M. A.; KHADER, A. T.; ZAMAN, M. University course timetabling using a hybrid harmony search metaheuristic algorithm. In: . [S.l.]: IEEE Transactions on systems, man, and cybernerics-part C: Applications and reviews, vol. 42, NO. 5, september, 2012.
- BONABEAU, E.; DORIGO, M.; THERAULAZ, G. Swarm intelligence from natural to artificial systems. In: . [S.l.]: Oxford, 1999.
- COOK, S. A. The complexity of theorem-proving procedures. In: . [S.l.]: University of Toronto, 1971.
- DARWIN, C. *The origin of species by means of natural selection: or, the preservation of favored races in the struggle for life*. [S.l.: s.n.], 1859.
- DAVIS, L. D. Bit-climbing, representational bias, and test suite design. In: . [S.l.]: In R. K. Belew and L. B. Booker (eds.), Proceedings of the Fourth International Conference on Genetic Algorithms, 18-23. San Mateo, CA: Morgan Kaufmann., 1991.
- EL-SHERBINY, M. M.; ZEINELDIN, R. A.; EL-DHSHAN, A. M. Genetic algorithm for solving course timetable problems. In: . [S.l.]: International Journal of Computer Applications, 2015.
- EVEN, S.; ITAI, A. E.; SHAMIR, A. On the complexity of timetabling and ulticommodity flow problems. In: . [S.l.]: SIAM Journal of Computation, 1976.
- FONSECA, G. H. G. et al. A sa-ils approach for the high school timetabling problem. In: . [S.l.]: Practice and Theory of Automated Timetabling (PATAT 2012), 2012.
- GAREY, M.; JOHNSON, D. Computers and intractability: A guide to the theory of np-completeness. In: . [S.l.]: Freeman, San Francisco, CA, USA, 1979.
- GEEM, Z. W.; KIM, J. H.; LOGANATHAN, G. A new heuristic optimization algorithm: Harmony search. In: . [S.l.]: Simulation, 2001.
- ITC2007. *ITC2007 Second international timetabling competition*. 2007. Disponível em: <http://www.cs.qub.ac.uk/itc2007/>.
- ITC2011. *ITC2011 Third international timetabling competition*. 2011. Disponível em: <https://www.utwente.nl/ctit/hstt/itc2011/welcome/>.
- KENNEDY, J.; EBERHART, R. Particle swarm optimization. In: . [S.l.]: Proceedings of IEEE International Conference on Neural Networks, 1995.
- KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. Optimization by simulated annealing. In: . [S.l.]: Science 220, 671–680, 1983.
- MITCHELL, M. An introduction to genetic algorithms. In: . [S.l.]: MIT Press, 1996.

NGUYEN, K.; NGUYEN, P.; TRAN, N. A hybrid algorithm of harmony search and bees algorithm for a university course timetabling problem. In: . [S.l.]: IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 1, No 1, January, 2012.

NOTHEGGER, C. et al. Solving the post enrolment course timetabling problem by ant colony optimization. In: . [S.l.]: Springer, 2012.

PHAM, D. et al. The bees algorithm. In: . [S.l.]: Technical Note, Manufacturing Engineering Centre, Cardiff University, UK, 2005.

SANTOS, H. G.; SOUZA, M. J. F. Programação de horários em instituições educacionais: Formulações e algoritmos. In: . [S.l.]: SBPO, 2007.