Universidade Federal Rural de Pernambuco
Departamento de Estatística e Informática

Uma aplicação da Programação Inteira no *School Timetabling Problem*

Rodrigo Hime

Recife

Janeiro de 2015

Rodrigo Hime

Uma aplicação da Programação Inteira no *School Timetabling Problem*

Orientador: Jones Albuquerque
Co-orientador: Glauco Gonçalves

Monografia apresentada ao Curso Bacharelado em Sistemas de Informação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Recife

Janeiro de 2015

Agradeço aos meus amigos e colegas de curso que contribuíram nessa jornada.
Especialmente a minha querida família que sempre acreditou em mim.

Agradecimentos

Agradeço, primeiramente, à minha família pela educação, apoio e compreensão constantes durante toda a minha vida, sem vocês eu não teria chegado aqui.

Aos meus amigos que participaram de grande parte da minha vida nos últimos seis anos na faculdade. Em especial a Bruno Moura, Matheus Benício, Victor Luna e Rafael Ribeiro pelas saídas, projetos mirabolantes e pela amizade. E a Ana Maria Medeiros pela ajuda com o Abstract.

A todos os amigos da equipe UKM's.

Agradeço aos professores Glauco e Jones pela confiança e apoio na realização deste trabalho.

Agradeço aos amigo da Ustore pelo apoio na realização deste trabalho seja na revisão como nos conselhos nessa reta final. Em especial ao time do MidiaCenter, Jean Louis, Lucas Inojosa, Bruno Moura e Victor Casé.

Agradeço aos todos que fazem parte da UFRPE, professores, coordenadores e servidores por me proporcionar o conhecimento necessário para minha formação e pelo excelente trabalho realizado.

Resumo

Este trabalho trata de um tema estudado há muitos anos pela Pesquisa Operacional: a resolução de um problema de Timetabling, ao qual se encaixa o Problema da Programação de Horários em Escolas (*School Timetabling*). É um problema combinatorial NP-completo que pode ser modelado matematicamente como um problema de Programação Inteira em que devem ser considerados os requisitos pedagógicos e organizacionais de cada escola e restrições pessoais. Destaca-se que em cada escola estes requisitos variam dificultando que seja desenvolvido um modelo genérico para este problema. Neste trabalho serão apresentadas a modelagem e implementação de uma solução, para o problema do calendário escolar, baseada em programação inteira. Foi tomando como caso de estudo a criação da grade de horário dos cursos de Pós-Graduação do Departamento de Estatística e Informática da UFRPE. São 2 cursos envolvendo 58 disciplinas e 30 professores. Um modelo de programação inteira foi desenvolvido para o problema e implementado em Python utilizando a biblioteca Pulp e o *solver* Gurobi Optimizer. Uma aplicação web foi desenvolvida para cadastro das entradas (professores, disciplinas, cursos, salas e preferências de alocação dos professores) necessárias para processamento da grade de horário. Os resultados foram bastante satisfatórios pois a grade de horários pode ser gerada em apenas 14 minutos e 20 segundos, contra 1 a 2 semanas de forma manual, dependendo do número de choques de horários entre os professores.

Palavras-chave: Programação Inteira , School Timetabling Problem, Otimização Combinatória

Abstract

This paper attends to a many years studied topic by Operational Reaserch: solving a Timetabling problem, in which fits the School Timetabling. It is a NP-complete combinatorial problem that can be mathematically model with a Integer Programming in which should be deliberated the pedagogical and organizational requirements of each school and personal restrictions. In every school, these requirements are diversified, making difficult a generical model to be developed for that problem. This paper will introduce the modeling and implementation of a solution to the school calendar problem, completely based on Integer Programming. It was taken as case study the designing of a schedule grid of the Post-Graduation courses of the Department Of Statistics and Informatics from UFRPE. These two courses involve 58 disciplines and 30 professors. A Integer Programming model was developed to the problem and implemented in Python using the Pulp library and the Gurobi Optimizer solver. A web application was developed to input registries (professors,disciplines, courses, rooms and professors allocation preferences) needed to the school grid processing. The results were quite satisfactory because the school grid can be generated in only 14 minutes and 20 seconds, against 1-2 weeks manually, depending on the number of schedule shocks between professors.

Keywords: Integer Programming , School Timetabling Problem, Combinatorial Optimization

Sumário

Lista de Tabelas

Lista de Figuras

Capítulo 1

Introdução

1.1 Apresentação

O Problema da criação de horários escolares (*School Timetabling*) consiste em fixar uma sequência de agendamentos de aulas envolvendo professores e grupos de estudantes (que possuem um mesmo currículo de disciplinas) em um período pré-determinado (tipicamente uma semana). Esta sequência de agendamentos deve satisfazer um conjunto de requisitos didáticos, físicos e organizacionais. Aparece muitas vezes como um problema de Programação Inteira (mais detalhes no capítulo 2), mas as suas variáveis de decisão são binárias. Neste trabalho será apresentado um modelo de programação inteira para o problema em questão e a sua resolução com o auxílio do *solver* Gurobi[10]. Para tornar a solução aplicável e de fácil utilização foi desenvolvida uma aplicação web.

1.2 Justificativa (motivação)

A criação dos horários de aulas é um problema recorrente em todo o início de período letivo nas instituições de ensino. Resolvê-lo manualmente, dependendo do tamanho, pode ser uma tarefa árdua, e durar dias ou até semanas para sua conclusão, podendo gerar resultados insatisfatórios com relação a diversos requisitos. Muitas vezes, professores reclamam por um mínimo de aulas geminadas (aulas em um único dia na semana) para algumas disciplinas e, principalmente, por uma grade de horário com menos períodos vagos entre uma aula e outra.

Além disso, dependendo do número de disciplinas e professores envolvidos no problema, ele se torna manualmente intratável. A aplicação de métodos automatizados pode trazer grandes vantagens, pois a satisfação dos interesses dos docentes e da instituição, pode ser atingida de forma mais rápida e exigindo menor esforço humano.

1.3 Objetivos

O objetivo deste trabalho é mostrar uma abordagem para a automatização do processo de construção da grade de horário de cursos e escolas, visando reduzir o tempo de construção e o esforço humano empregado no processo, com foco na melhoria da satisfação do professores com a grade de horário gerada. Para isso foi tomado como exemplo os cursos de Pós-Graduação do Departamento de Estatística e Informática da UFRPE. São 2 cursos envolvendo 58 disciplinas e 30 professores. Dessa forma, pretende-se atender a objetivos específicos tais como:

- Desenvolver um modelo de Programação Inteira para os cursos em questão, maximizando as preferências dos professores em relação a escolha de suas turmas e horários alocados, e minimizando o número de períodos vagos entre suas aulas.
- Automatizar o processo de construção de grades de horário dos cursos em questão, utilizando o *solver* Gurobi para resolver o modelo proposto e desenvolvendo uma aplicação web para tornar a solução de fácil utilização pelos responsáveis pela geração da grade de horário.

Não faz parte do escopo deste trabalho cobrir todas as variantes da classe de problemas de alocação de horários (*timetabling*) nem tão pouco explanar todas as minúcias da Programação Inteira. Também não faz parte do escopo deste trabalho cobrir todas as peculiaridades e restrições do processo de criação da grade de horário dos cursos tomados como exemplo.

1.4 Organização do trabalho

Este trabalho está organizando nos seguintes capítulos.

- O capítulo 1 dá uma introdução ao problema abordado, apresentando o trabalho, sua justificativa, seus objetivos e as contribuições obtidas.
- O capítulo 2 faz uma explanação a cerca do problema abordado, contextualiza a programação linear e apresenta as ferramentas utilizadas na solução.
- O capítulo 3 apresenta um modelo de programação inteira para o problema.
- O capítulo 4 apresenta a implementação da solução modelada no capítulo 3 bem como os seu resultado.
- No capítulo 5 restam as considerações finais sobre o trabalho e possíveis trabalhos futuros.

Capítulo 2

Referencial Teórico/Revisão da Literatura

2.1 O Problema da Criação de Horários Escolares

O problema da criação de quadro de horário escolar (*School Timetalbing*) pertence à grande família de problemas de agendamento (*Scheduling Problems*).

"*Scheduling Problems* dizem respeito à alocação de recursos limitados para tarefas ao longo do tempo. É um processo de tomada de decisão que tem como objetivo a otimização de um ou mais objetivos [...] Os diferentes componentes de um problema de agendamento são as tarefas, as possíveis restrições, os recursos e a função objetivo" [27, p. 5]

"O *School Timetable Problem* diz respeito a todos os estabelecimentos de ensino ou universidades, uma vez que envolvem calendarização dos cursos, assegurando a disponibilidade de professores, alunos e salas de aula [...] " [27, p. 7]

Segundo [4] quando se trata de problemas de programação de horários em uma instituição de ensino, o objetivo é programar uma série de encontros, geralmente no período de uma semana, entre professores e alunos satisfazendo um conjunto de restrições que envolvem recursos (professores, salas de aula), aspectos pedagógicos e satisfabilidade/indisponibilidade dos envolvidos.

2.1.1 Características de um Problema de Horário

Devido a sua natureza combinatória problemas de alocação de horários são considerados, em quase todas as suas variantes, como membro do conjunto de problemas NP-Completo, o que significa ser um problema de alta complexidade computacional onde o esforço computacional para sua resolução cresce exponencialmente com o número de variáveis. Esta classe de problemas pode ser caracterizada por não possuir um algoritmo eficiente que resolva estes problemas em tempo computacional razoável (tempo polinomial).

Estes problemas podem admitir uma grande variedade de formulações, objetivos e restrições que podem variar de instituição a instituição, mas de modo geral é possível dividir os problemas de quadro de horário escolar em três classes [5]:

- Problema de programação de horários de Escolas (School Timetabling)
- Problema de programação de horários de Cursos/Universidades (Course Timetablig ou University Timetabling)
- Problema de programação de horários de Exames (Exam Timetabling).

Esta divisão é baseada no tipo de entidade envolvida (Universidade ou Escola) e principalmente nos tipos de restrições associadas ao problema. Estas restrições podem ser classificadas em diversos tipos onde as principais são: restrições rígidas (*hard constraints*) e restrições leves (*soft constraints*).

Segundo [26] as restrições rígidas são aquelas que não podem ser violadas em nenhuma hipótese, pois tonariam inviável o cumprimento da grade de horário, como por exemplo um professor não pode dar aula à turmas diferentes em salas distintas no mesmo dia e horário. Já as restrições leves, são restrições desejáveis, que devem ser preferencialmente seguidas mas sua violação não inviabiliza a solução, como por exemplo a grade de horário deve ter o menor número possível de janelas entre as aulas dos professores.

2.1.2 Problema de Horários de Escolas

O Problema de Horários de Escolas refere-se à construção da grade de horários de instituições de ensino que têm uma programação semanal que se repete em todas as semanas do período (semestral ou anual).

Segundo [5] O problema básico consiste em, dado um conjunto de classes, um conjunto de professores, um conjunto de períodos e o número de aulas que cada professor deve ministrar a cada classe, evitar que os professores estejam alocados em mais de uma classe no mesmo horário; evitar que as classes tenham aula com mais de um professor ao mesmo tempo; e fazer com que a carga-horária semanal de cada disciplina seja cumprida.

2.1.3 Problema de Horários de Cursos/Universidades

Este tipo de problema refere-se a alocação de aulas de uma instituição de ensino, na qual uma disciplina pode fazer parte do currículo de vários cursos.

Diferente do que acontece no Problema de Horários Escolares no qual cada classe tem um conjunto fixo de estudantes, o Problema de Horário de Cursos pode ter estudantes de cursos diferentes fazendo a mesma disciplina. Se duas disciplinas possuem estudantes em comum e seus horários conflitam, então elas não poderão ser programadas para o mesmo período.

Segundo [5] Esse problema consiste, então, em elaborar um quadro semanal com o horário de todas as aulas de um conjunto de disciplinas de cursos universitários, minimizando a sobreposição de encontros de disciplinas que possuem estudantes em comum.

Outra dificuldade e que o número de salas e seus tamanhos exercem um papel muito importante nesse problema, fato que pode ser desconsiderado no problema de Horário de Escolas

2.1.4 Problema de Horários de Exames

O Problema de Horários de Exames requer a programação de um determinado número de exames, dentro de um determinado período de tempo, em um conjunto de salas.

Segundo [5] o Problema de Horário de Exames é semelhante ao Problema de Horário de Cursos diferenciando-se pelo tipo de restrições impostas pois há tipos diferentes de requisitos, como, por exemplo, permitir somente um exame por dia para cada estudante ou evitar que os exames para cada estudante sejam consecutivos.

2.2 Técnicas Utilizadas para Resolução de Problemas de Horário

Ao longo dos anos diversas estratégias foram desenvolvidas para a resolução de problemas de horários. De maneira geral essas estratégias podem ser divididas em: estratégias exatas e inexatas.

As estratégias exatas utilizam métodos que buscam a solução ótima para o problema em todos os casos, mas que não garante que será rápido em todas as situações, apesar de resolver muito bem, em um tempo bom, pequenas instâncias do problema.

Dentre os métodos exatos destacam-se:

- Algoritmo simplex
- Branch e bound
- Programação dinâmica
- Relaxação lagrangeana

Dentre as estratégias inexatas destacam-se os métodos heurísticos ou metaheurísticos. A palavra heurística vem da palavra grega Heuriskein, que significa descobrir (e que deu origem também ao termo Eureca). As heurísticas são procedimentos de busca intuitivos desenvolvidos para a obtenção de soluções de problemas que não possuam solução matemática exata satisfatória.

Dentre os métodos heurísticos destacam-se:

- Simulated annealing
- Algoritmos genéticos
- Busca tabu

- Algoritmo da colônia de formigas
- Redes neurais
- Greedy Randomized Adaptive Search Procedures (GRASP)
- Algoritmos gulosos

Utilizar uma estratégia exata é somente possível para pequenas instâncias porque o tempo para encontrar a solução em grandes instâncias pode se tornar inviável. Para instâncias grandes, este tipo de problema deve ser abordado por métodos heurísticos.

Carvalho, Rodrigo[4] aborda o problema da alocação de horários escolares utilizando um método composto por três meta-heurísticas conhecidas na literatura *Iterated Local Search* (ILS), *Variable Neighborhood Descent* (VND) e *Greedy Randomized Adaptive Search Procedure* (GRASP). O autor usa o método GRASP para gerar uma solução inicial e os métodos ILS e VND para refinar o resultado.

Para a alocação otimizada de horários de turmas em escolas de ensino médio Kotsko, EG da Silva[11] desenvolveu um modelo de Programação Inteira e implementado com o pacote computacional LINGO (*Language for Iteractive General Optimize*).

No trabalho apresentado por Abensur, EO e Oliveira, RC[1] foi apresentada uma heurística construtiva, desenvolvida pelo autor do trabalho, que opta por gerar uma solução inicial dando preferência as turmas com maior carga horária e aos professores de grau maior, na segunda fase da solução uma heurística baseada no procedimento *all pairs* foi utilizada para refinar a solução inicial até atingir a condição de parada definida.

A próxima seção apresenta a formulação matemática do problema usando um modelo de Programação Inteira.

2.2.1 Programação Linear

Os primeiros conceitos da programação linear foram desenvolvidos entre 1947 e 1949, durante a segunda guerra mundial, por George Dantzig para serem aplicados a programas militares, desde a área logística, até à estratégica. Foi após a guerra que ele foi impulsionado a encontrar formas eficientes de desenvolver esta metodologia. Foi Dantzig o primeiro a reconhecer que um programa de planejamento poderia ser expresso por um sistema de inequações lineares, assim como foi o primeiro a apresentar, na forma de uma expressão matemática explicita, um critério para seleção do melhor plano, ao que hoje chamamos função objetivo.

A Programação Linear visa fundamentalmente encontrar a melhor solução para problemas que tenham seus modelos representados por expressões lineares. A grande aplicabilidade e simplicidade que a caracterizam devem-se à linearidade do modelo. A tarefa da Programação Linear consiste na maximização ou minimização de uma função linear, denominada função objetivo, respeitando-se um sistema linear de igualdades ou desigualdades que recebem o nome de restrições do modelo [24]. Suas variáveis pertencem ao conjunto dos números reais.

As restrições representam normalmente limitações de recursos disponíveis (capitais, mão-de-obra, recursos minerais ou fatores de produção) ou, então exigências e condições que devem ser cumpridas no problema. Essas restrições do modelo determinam uma região à qual damos o nome de "Conjunto de Soluções Viáveis". A melhor das soluções viáveis, isto é, aquela que melhor maximiza ou minimiza a função objetivo denomina-se 1 "Solução Ótima". O objetivo da Programação Linear consiste na determinação da solução ótima.

Por exemplo: Uma empresa fabrica cadeiras e mesas. Cada cadeira necessita de 5 tábuas de madeira e cada mesa 20. Ao todo temos 400 tábuas. Cada cadeira precisa de 10 horas de trabalho e cada mesa 15 horas. Temos 450 horas de trabalho disponíveis. Queremos maximizar o lucro. O lucro por cadeira é 45 e por mesa é 80.

Podemos definir duas variáveis:

- x_1: número de cadeiras,
- x_2: número de mesas

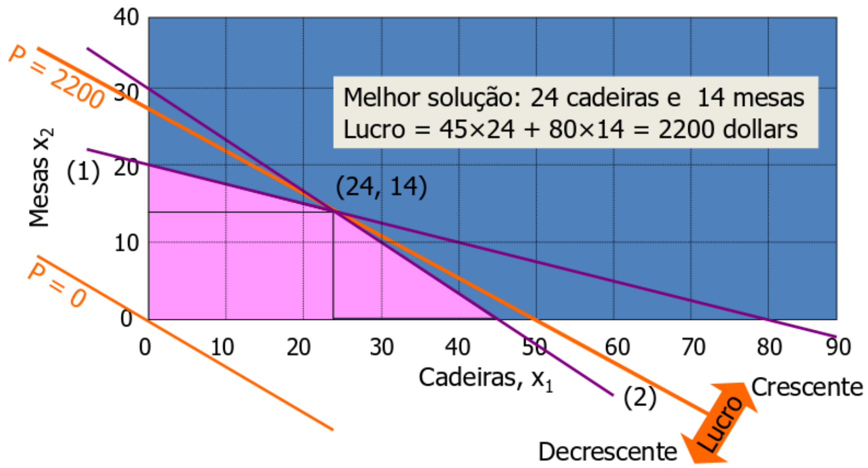
O objetivo é maximizar Z:

- $Z = 45x_1 + 80x_2$

Sujeito a:

- $5x_1 + 20x_2 \leq 400$
- $10x_1 + 15x_2 \leq 450$
- $x_1 \geq 0; \quad x_1 \in \mathbb{R}$
- $x_2 \geq 0; \quad x_2 \in \mathbb{R}$

Para maximizar o lucro devemos descobrir os valores de x_1 e x_2 (número de cadeiras e de mesas respectivamente) que der o maior resultado. A resposta pode ser visualizada na figura abaixo:



Segundo [24] as etapas básicas para formular um modelo linear são:

- Identificação das variáveis de decisão
- Identificação da função objetivo
- Identificação das restrições
- Formulação matemática

São necessários dois passos para a resolução de programação linear, o primeiro é a modelagem do problema e o segundo é a obtenção da solução do modelo.

2.2.1.1 Programacao Inteira

A Programação Inteira pode ser entendida como uma caso específico da Programação Linear, onde as variáveis devem ser inteiras. Um problema de programação inteira é um problema de otimização matemática no qual algumas ou todas as variáveis do problema pertencem ao conjunto dos números inteiros. Uma vez que problemas de otimização combinatória podem ser codificados como problemas inteiros, produzir soluções para eles é um problema NP-completo.

Exemplos Aplicações:

- Problemas de investimentos.
- Problemas de custo fixo.
- Problemas de alocação de armazéns.
- Problemas de sequenciamento de tarefas.
- Roteamento de veículos, linearização de função objetivo com produto de variáveis, problema do caixeiro viajante, problemas de "matching", de "covering", de "partitioning", e de "packing").

2.2.1.2 Decisão Binária

Talvez o tipo mais usado de variável inteira é a variável binária: uma variável inteira restrita a assumir os valores 0 ou 1. Muitos problemas combinatórios práticos podem ser vistos como um conjunto de perguntas que podem ser respondidas com "sim"ou "não", para muitas dessas decisões, um modelo de programação linear binária é adequado.

$$\text{Solução. Usar variável binária} \quad Y_i = \begin{cases} 1 & \text{, se a decisão for sim} \\ 0 & \text{, se a decisão for não} \end{cases} , \quad \forall : i \in \mathbb{Z}$$

Quando somente uma das variáveis de decisão num conjunto pode ser sim podemos definir restrições . Exemplo:

$$\sum_i Y_i = 1, \text{Se exatamente uma decisão no conjunto tiver que ser sim;} \quad \forall : i \in \mathbb{Z}$$

$$\sum_i Y_i \leq 1, \text{Se no máximo uma variável de decisão no conjunto tiver que ser sim;} \quad \forall : i \in \mathbb{Z}$$

Quando a decisão k depende de uma decisão j anterior, então temos que um decisão k é contingente na decisão j, se a decisão k puder ser "sim" somente se a decisão j for "sim". Exemplo:

$Y_k \leq Y_j$, ou seja, quando $Y_j = 1$ dá escolha livre para Y_k, mas se $Y_j = 0$ força $Y_k = 0$.

Capítulo 3

Modelagem Matemática

O modelo matemático, assim como o sistema de informação, que será apresentado no capítulo 4, foi construído baseando-se nos trabalhos [19], [23], [8], [4], [2], [1], [25], [28] e análise do funcionamento do processo de alocação de professores dos cursos de Pós Graduação do Departamento de Estatística e Informática da Universidade Federal Rual de Pernambuco.

3.1 Formulação do Problema

3.1.1 Variáveis do Problema

Os conjuntos de variáveis de entrada para a alocação de cada professor são definidas como:

- Conjunto de Professores = {1,...,P}
- Conjunto de Turmas = {1,...,T}
- Conjunto de Dias = {1, 2, 3, 4, 5}, respectivamente, {segunda, terça, quarta, quinta, sexta}
- Conjunto de Salas = {1, ..., S}
- Conjunto de Horários = {1, 2, 3, 4}, respectivamente, {08:00 as 10:00, 10:00 as 12:00, 14:00 as 16:00, 16:00 as 18:00}

- Conjunto com o número de dias que um professor gostaria de lecionar suas aulas para determinada turma (1 ou 2 dias). Para cada (professor *vs* turma) será atribuído um opção em = {1 dia, 2 dias}

- Conjunto com o peso da preferência do professor para as turmas. (Caso o professor não lecione a matéria da turma será atribuído 0 para o peso) = {0, 3, 8, 13}

- Conjunto de horários preferenciais das aulas de cada professor. Para cada (professor *vs* dia *vs* horário) = {1, 3, 8}

O número total de variáveis de decisão para os 30 professores é definido pelo produto cartesiano dos conjuntos de entrada do problema X(professor, turma, dia, horário, sala). São 30 professores, 58 turmas, 5 dias semanais, 4 horários diários, 6 salas de aula, totalizando 208.800 variáveis de decisão. O número de salas foi definido como sendo o número mínimo para alocar todas as 58 turmas. Cada turma deve ser alocada em dois horários semanais totalizando 116 horários para as 58 turmas portanto são necessárias 6 salas de aula já que cada sala gera 20 horários semanais (5 dias *vs* 4 horários) dando um total de 120 horários.

3.1.2 Função Objetivo do Problema

A função objetivo deve dar preferência às alocações considerando as premissas abaixo.

- Maximizar a chance de alocação das turmas preferenciais dos professores.

- Maximizar a chance de alocação dos professores em seus dias e horários preferências.

- Maximizar a chance de alocação dos professores em determinada turma em 1 ou 2 dias dependendo da preferência deles.

- Minimizar o número de períodos vagos entre as aulas dos professores.

3.1.3 Restrições do Problema

As restrições do modelo estão descritas a seguir:

Restrições rígidas:

- Cada turma deve ter apenas um professor.
- Cada professor deve estar em apenas uma aula (turma/sala) em um (dia/horário).
- Cada professor deverá dar 4 aulas semanais em uma turma que esteja alocado o que corresponde a 2 segmentos de horários.
- As aulas de uma turma devem estar alocadas em apenas uma sala.
- Cada sala de aula deve estar alocada para no máximo uma aula(professor/turma) em um dia e horário.
- Cada turma deve estar alocada em apenas uma sala.
- Caso as aulas de uma turma ocorram em um único dia na semana as aulas devem ser alocadas ou nos dois primeiros ou nos dois últimos horários.
- Cada professor deverá estar alocado em 1 ou 2 dias na semana para uma determinada turma.

Restrições leves:

- Deve-se minimizar o número de períodos vagos entre as aulas dos professores.

3.1.4 Modelagem Matemática do Problema

Para atender algumas premissas definidas para a função objetivo foram estabelecidos pesos visando as preferências de alocação dos professores. Para atender a preferência de horários dos professores, definiu-se $\beta_{p,d,h} \in \{3,\ 5,\ 8\}$ para o peso da alocação do p-ésimo professor no d-ésimo dia e h-ésimo horário. Para a preferência de turmas do professores definiu-se $\alpha_{p,t} \in \{0,\ 3,\ 8\ ,13\}$ para o peso da alocação do p-ésimo professor para a t-ésima turma, o valor 0 para α, corresponde as turmas em que o professor não está apto para ministra aulas.

Para atender as preferências dos professores em lecionar suas aulas para uma determinada turma em 1 ou 2 dias, algumas variáveis de decisão auxiliares tiveram que ser criadas. Foram criadas as variáveis auxiliares $Q1_{p,t} = \{0,\ 1\}$ e $Q2_{p,t} = \{0,\ 1\}$ que juntamente com os pesos $pq1_{p,t} = pq2_{p,t} = \{1,\ 10\}$ maximizam a preferência dos professores em ter suas aulas em 1 ou 2 dias na semana. A variável $Q1_{p,t}$, corresponde a alocação do p-ésimo professor na t-ésima turma em 1 único dia na semana e $Q2_{p,t}$ em dois dias na semana. Os pesos $pq1_{p,t}$ e $pq2_{p,t}$, correspondem ao peso da preferência do p-ésimo professor em lecionar para a t-ésima turma, o valor de $pq1_{p,t}$ será 10 se o p-ésimo professor preferir suas aulas em 1 único dia na semana ou 1 caso contrário, o valor de $pq2_{p,t}$ será 10 se o p-ésimo professor preferir suas aulas em 2 dias na semana ou 1 caso contrário.

Para minimizar o número de períodos vagos entre as aulas dos professores em um mesmo dia, foi elabora uma estratégia para penalizar as alocações de professores que ocorrerem em determinadas combinações no mesmo dia. Foram identificadas 3 combinações em que acontecem períodos vagos na alocação do professor em um dia. A primeira é quando um professor é alocado no primeiro e no terceiro horário e não é alocado no segundo. A segunda e quando o professor é alocado no segundo e no quarto horários e não é alocado no terceiro. A terceira combinação é quando o professor é alocado no primeiro e no quarto horários e não é no segundo e terceiro horários. Para cada combinação foi criada uma variável de decisão auxiliar $J1_{p,d} = \{0,\ 1\}$, $J2_{p,d} = \{0,\ 1\}$, $J3_{p,d} = \{0,\ 1\}$. A variável $J1_{p,d}$, corresponde a alocação do p-ésimo professor no d-ésimo dia nos horários 1 e 3 e não alocado no horário 2. A variável $J2_{p,d}$, corresponde a alocação do p-ésimo professor no d-ésimo dia nos horários 2 e 4 e não alocado no horário 3. A variável $J3_{p,d}$, corresponde a alocação do p-ésimo professor no d-ésimo dia nos horários 1 e 4 e não alocado nos horários 2 e 3. Se $J1_{p,d}$ ou $J2_{p,d}$ for igual a 1 o resultado da função objetivo será penalizado em 5. Se $J3_{p,d}$ for igual a 1 o resultado da função objetivo será penalizado em 10 portanto o dobro das outras já que o neste caso existem dois períodos vagos.

Maximizar:

$$
\begin{aligned}
\mathbb{Z} = &\sum_{p=1}^{P}\sum_{d=1}^{5}\sum_{h=1}^{4}\sum_{s=1}^{S}\sum_{t=1}^{T} \alpha_{p,t}\beta_{p,d,h}X_{p,d,h,s,t} \quad + \\
&\sum_{p=1}^{P}\sum_{t=1}^{T}(pq1_{p,t}Q1_{p,t} + pq2_{p,t}Q2_{p,t}) \quad + \\
&\sum_{p=1}^{P}\sum_{d=1}^{D}(-5J1_{p,d} - 5J2_{p,d} - 10J3_{p,d})
\end{aligned}
\tag{3.1}
$$

Onde:

p , é o índice do conjunto de professores.

d , é o índice do conjunto de dias da semana.

h , é o índice do conjunto de horários que é formado pelo conjunto de segmento de horários {1, 2, 3, 4}, totalizando 4 segmentos. Onde cada índice representa um conjunto de horários disponíveis. Cada índice do conjunto representa um horário {1:{8:00 as 10:00}, 2:{10:00 as 12:00}, 3:{14:00 as 16:00}, 4:{16:00 as 18:00}}.

s é o índice do conjunto de salas.

t é o índice do conjunto de turmas (disciplinas).

$\boldsymbol{X_{p,d,h,s,t} \in \{0,\ 1\}}$ é 1, se o p-ésimo professor for designado à t-ésima turma no d-ésimo dia no h-eximo horário e na s-ésima sala. 0, caso contrario.

$\boldsymbol{Z_{p,t} \in \{0,\ 1\}}$ é 1, se o p-ésimo professor for designado à t-ésima turma. 0, caso contrario.

$\boldsymbol{Y_{s,t} \in \{0,\ 1\}}$ é 1, se a t-ésima turma for alocada na s-ésima sala. 0, caso contrario.

$\boldsymbol{Q1_{p,t} \in \{0,\ 1\}}$ é 1, se o p-ésimo professor for designado para a t-ésima turma com as aulas em um único dia na semana. 0 caso contrario.

$\boldsymbol{Q2_{p,t} \in \{0,\ 1\}}$ é 1, se o p-ésimo professor for designado para a t-ésima turma com as aulas divididas em dois dias na semana. 0 caso contrario.

$\boldsymbol{J1_{p,d} \in \{0,\ 1\}}$ é 1, se o o p-ésimo professor for alocado no d-ésimo dia nos horários {1 e 3} e não for alocado no horário {2} o que corresponde a uma janela entre os horários {1 e 3}.

$\boldsymbol{J2_{p,d} \in \{0,\ 1\}}$ é 1, se o o p-ésimo professor for alocado no d-ésimo dia nos horários {2 e 4} e não for alocado no horário {3} o que corresponde a uma janela entre os horários {2 e 4}.

$\boldsymbol{J3_{p,d} \in \{0,\ 1\}}$ é 1, se o p-ésimo professor for alocado no d-ésimo dia nos horários {1 e 4} e não for alocado no horário {2 e 3} o que corresponde a duas janela entre os horários {1 e 4}.

$\boldsymbol{\alpha_{p,t} \in \{0,\ 3,\ 8,\ 13\}}$ Peso correspondente a preferência do p-ésimo professor em ser alocado na t-ésima turma.

$\boldsymbol{\beta_{p,d,h}} \in \mathbf{\{3,\ 5,\ 8\}}$ Peso correspondente a preferência do p-ésimo professor em ser alocado no d-ésimo dia, no h-ésimo horário.

$\boldsymbol{pq1_{p,t}} \in \mathbf{\{1,\ 10\}}$ Peso correspondente a preferência do professor em lecionar suas aulas para uma determinada turma em um único dia na semana. Será 10 se o p-ésimo professor preferir lecionar suas aulas para a t-ésima turma em um único dia na semana. 1 caso contrario.

$\boldsymbol{pq2_{p,t}} \in \mathbf{\{1,\ 10\}}$ Peso correspondente a preferência do professor em lecionar suas aulas para uma determinada turma em dois dias na semana. Será 10, se o p-ésimo professor preferir lecionar suas aulas para a t-ésima turma em dois dias diferentes na semana. 1, caso contrario.

Sujeito as restrições:

- Cada Turma deve ter apenas um professor:

$$\sum_{p=1}^{P} Z_{p,t} \leq 1, \quad \forall : t \in T \tag{3.2}$$

- Cada Professor só pode esta alocado no máximo em uma única aula (professor/turma/sala) em um (dia/horário):

$$\sum_{t=1}^{T} \sum_{s=1}^{S} X_{p,d,h,s,t} \leq 1, \quad \forall : d \in D; h \in H; p \in P \tag{3.3}$$

- Cada professor deve dar 4 aulas semanais em uma turma. O que corresponde a 2 segmentos de horários semanais:

$$\sum_{h=1}^{4} \sum_{d=1}^{5} \sum_{s=1}^{S} X_{p,d,h,s,t} = 2Z_{p,t}, \quad \forall : p \in P; t \in T \tag{3.4}$$

- Cada professor poderá estar alocado em 1 ou 2 dias para uma determinada turma:

$$\sum_{s=1}^{S} \sum_{h=1}^{4} X_{p,d,h,s,t} = Q1_{p,t} + 2Q2_{p,t}, \quad \forall : p \in P; t \in T; d \in D \tag{3.5}$$

- Cada sala de aula deve está alocada para no máximo uma aula(professor/turma) em um dia e horário:

$$\sum_{p=1}^{P}\sum_{t=1}^{T} X_{p,d,h,s,t} \leq 1, \quad \forall : s \in S; d \in D; h \in H \tag{3.6}$$

- Cada turma deve estar alocada em apenas uma sala:

$$\sum_{s=1}^{S} Y_{s,t} \leq 1, \quad \forall : t \in T \tag{3.7}$$

- Relação entre Y e X:

$$X_{p,d,h,s,t} \leq Y_{s,t} \quad \forall : p \in P; d :\in D; h \in H; s \in S; t \in T \tag{3.8}$$

- Este conjunto de restrições define que, caso uma turma tenha suas aulas agrupadas em um único dia na semana, as aulas ocorram ou nos dois primeiros horários ou nos dois últimos.

$$\begin{aligned}
X_{p,d,h=1,s,t} + X_{p,d,h=3,s,t} \leq 1 \quad \forall : p \in P; d :\in D; s \in S; t \in T\\
X_{p,d,h=1,s,t} + X_{p,d,h=4,s,t} \leq 1 \quad \forall : p \in P; d :\in D; s \in S; t \in T\\
X_{p,d,h=2,s,t} + X_{p,d,h=3,s,t} \leq 1 \quad \forall : p \in P; d :\in D; s \in S; t \in T\\
X_{p,d,h=2,s,t} + X_{p,d,h=4,s,t} \leq 1 \quad \forall : p \in P; d :\in D; s \in S; t \in T
\end{aligned} \tag{3.9}$$

- Este conjunto de restrições tem o objetivo de penalizar alocações de professores com janelas entre as aulas visando minimizar o número de janelas entre as aulas dos professores:

$$\begin{aligned}
\sum_{s=1}^{S}\sum_{t=1}^{T} X_{p,d,h=1,s,t} - X_{p,d,h=2,s,t} + X_{p,d,h=3,s,t} \leq 1 + J1_{p,d}, \quad \forall : p \in P; d :\in D\\
\sum_{s=1}^{S}\sum_{t=1}^{T} X_{p,d,h=2,s,t} - X_{p,d,h=3,s,t} + X_{p,d,h=4,s,t} \leq 1 + J2_{p,d}, \quad \forall : p \in P; d :\in D\\
\sum_{s=1}^{S}\sum_{t=1}^{T} X_{p,d,h=1,s,t} - X_{p,d,h=2,s,t} - X_{p,d,h=3,s,t} + X_{p,d,h=4,s,t} \leq 1 + J3_{p,d}, \quad \forall : p \in P; d :\in D
\end{aligned} \tag{3.10}$$

Capítulo 4

Implementação

Esse capítulo descreve o sistema de informação denominado ClassMaker, desenvolvido para automatizar o processo de geração da grade de horários dos cursos do Departamento de Estatística e Informática da Universidade Federal Rural de Pernambuco de acordo com o modelo apresentada no capítulo 3.

O sistema foi desenvolvido em dois módulos. O primeiro é uma aplicação Web que é responsável pelo gerenciamento de todos os dados de entrada do problema: cursos, professores, disciplinas, salas de aula e preferências dos professores. O segundo é a aplicação denominada Resolvedor responsável por implementar e resolver computacionalmente o modelo baseado na nos dados enviados pela aplicação Web. A figura abaixo ilustra a arquitetura do sistema.

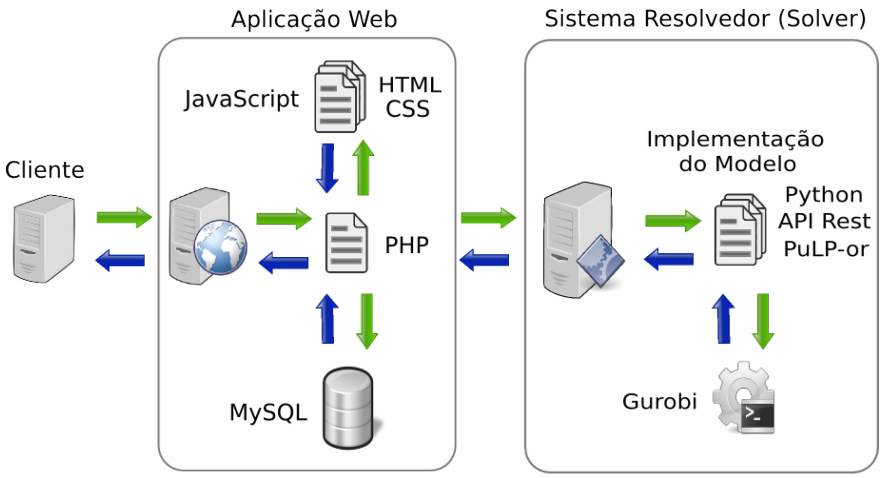


Figura 4.1: Arquitetura do Sistema

Essa arquitetura favorece o desempenho do sistema pois do Módulo Resolvedor poderá ser implantado em um servidor separado Módulo Web não tendo que dividir seus recursos com o resto da aplicação.

4.1 Módulo Resolvedor

O Módulo resolvedor é o coração da aplicação onde é feita a implementação do modelo e resolução do problema. A estratégia definida para solução foi criar um serviço (Web Service) para receber os dados cadastrados através do Módulo Web, implementar o modelo com os dados recebidos, resolver esse modelo em um *solver* e enviar de volta o resultado ao Módulo Web.

Foi implementado um serviço, por meio de uma API Rest em Python, que recebe um arquivo JSON do Módulo Web com os dados necessários para gerar a grade de horários (professores, turmas, salas, preferências de turmas dos professores, preferência de dias dos professores e preferência de alocação das aulas em 1 ou 2 dias dos professores nas turmas), com o auxílio da biblioteca PuLP transforma esses dados em um arquivo LP (formato para escrever modelos de programação linear) baseado no modelo e em seguida expõe esse arquivo ao *solver*.

Foram testados os *solvers* GLPK [9], CPLEX [7] e GUROBI [10], a escolha entre os três foi definida pelo melhor desempenho. O Gurobi foi o que teve o melhor desempenho dos três, resolvendo o problema em 14 minutos e 20 segundos, o CPLEX e o GLPK foram descartados após 1 horas executando sem obter o resultado.

4.1.1 Gurobi Optimizer

Gurobi Optimizer é um solucionador (*solver*) para problemas de Programação Linear (LP), Programação Inteira Mista (MIP) e Programação Quadrática (QP), sua licença e comercial porém é possível obter uma licença acadêmica grátis.

Gurobi inclui implementações avançadas dos algoritmos mais recentes incluindo: *parallel branch-and-cut, non-traditional tree-of-trees search, multiple default heuristics, cutting planes, symmetry detection, primal simplex, dual simplex, parallel barrier with crossover,* *concurrent optimization.* Gurobi permite paralelismo de memória compartilhada e, portanto, é capaz de explorar simultaneamente qualquer número de processadores e núcleos por processador. Sua implementação é determinista e, portanto, duas execuções separadas no mesmo modelo irão produzir resultados idênticos.

Informações de como obter uma licença, instalar e usar o Gurobi podem ser encontradas em [20].

4.1.2 Pulp

Muitos "resolvedores" de Programação Linear Inteira Mista (PLIM) estão disponíveis tanto comerciais como não comerciais. Pulp é uma biblioteca escrita em Python para implementação de modelos de programação linear (LP) e Programação Inteira (IP). Ele converte expressões Python em números brutos (ou seja, matriz e vetor que representam o modelo) e em seguida expõem esses dados para o *solver* escolhido. Pulp pode gerar arquivos LP ou MPS e chamar outros softwares como GLPK [9], CPLEX [7], COIN-OR [6] e GUROBI [10] para resolver o modelo. Ele está sob uma licença BSD (de código aberto). Sua documentação pode ser encontrada em [21].

A adoção do pulp, como ferramenta para implementar o modelo do problema, foi determinada devido sua implementação ser feita em linguagem de programação (Python) o que da flexibilidade na automatização da solução, além de sua simplicidade e facilidade de utilização que derivam do Python. A seguir uma breve demonstração de sua utilização.

4.1.2.1 Criando uma variável

Para criar uma variável usa-se o método LpVariable():

Para criar uma variável $0 <= x <= 3$:

```
>>> x = LpVariable("x", 0, 3)
```

Para criar uma variável binária:

```
>>> y = LpVariable("y", 0, 1, cat=LpBinary)
```

4.1.2.2 Criando um problema

Para criar um problema usa-se método LpProblem():

```
>>> probMin = LpProblem("meuProblemaMin",LpMinimize)
>>> probMax = LpProblem("meuProblemaMax",LpMaximize)
```

4.1.2.3 Criando Expressões e Restrições

Combine variáveis para criar expressões e restrições e adicionar ao problema:

```
>>> probMin += x + y <= 2
```

Se for adicionada uma expressão (não uma restrição), ela se tornará a função objetivo:

```
>>> probMin += -4*x + y
```

4.1.2.4 Escolhendo o Solver e executando

Para executar utiliza-se o método solve() passando o “resolvedor” como parâmetro:

```
>>> status = probMin.solve(GLPK(msg = 0))
```

4.1.2.5 Exibindo o resultado

Para exibir o resultado da solução:

```
>>> LpStatus[status]
'Optimal'
```

Pode-se pegar o valor das variáveis usando value():

```
>>> value(x)
2.0
```

Para exibir o resultado da solução:

```
>>> value(probMin.objective)
-8.0
```

Exemplos mais completos da utilização do PuLP podem ser encontrados em [15] e [16]

4.2 Módulo Web

A função da aplicação web é fornecer uma interface amigável para o usuário utilizar a implementação do Resolvedor assim como gerenciar os dados de entrada necessários para o Resolvedor gerar a grade de horário. A figura 4.2 exibe a tela inicial da aplicação.

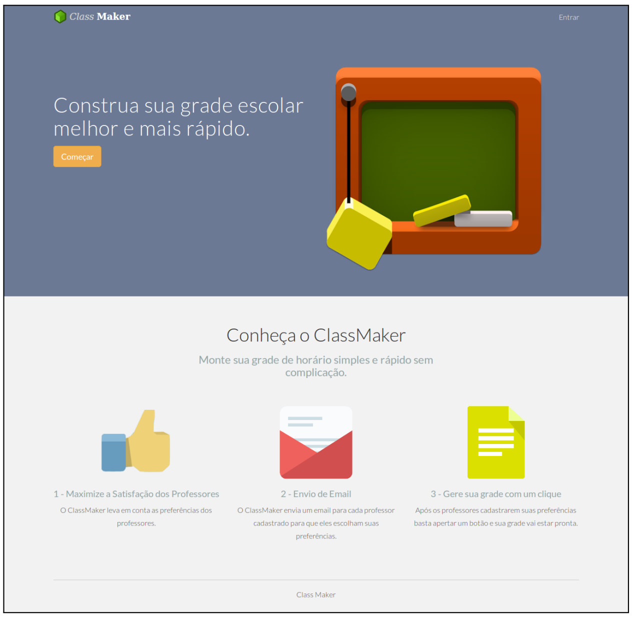


Figura 4.2: ClassMake - Tela Inicial

Esses dados são inseridos no sistema a partir de uma interface desenvolvido em PHP + HTML + CSS + JavaScript + MySQL. Abaixo são descritas as suas entradas.

- Cursos (neste trabalho: PPGIA e PGBIOM)
- Disciplinas dos cursos
- Professores dos cursos
- Salas disponíveis para os cursos

- Disciplinas preferenciais para alocação dos professores
- Dias da semana preferências para alocação dos professores
- Preferência dos professores na distribuição das aulas na semana para determinada disciplina: 1 ou 2 dias

A figura 4.3 exibe as telas de gerenciamento de cursos.

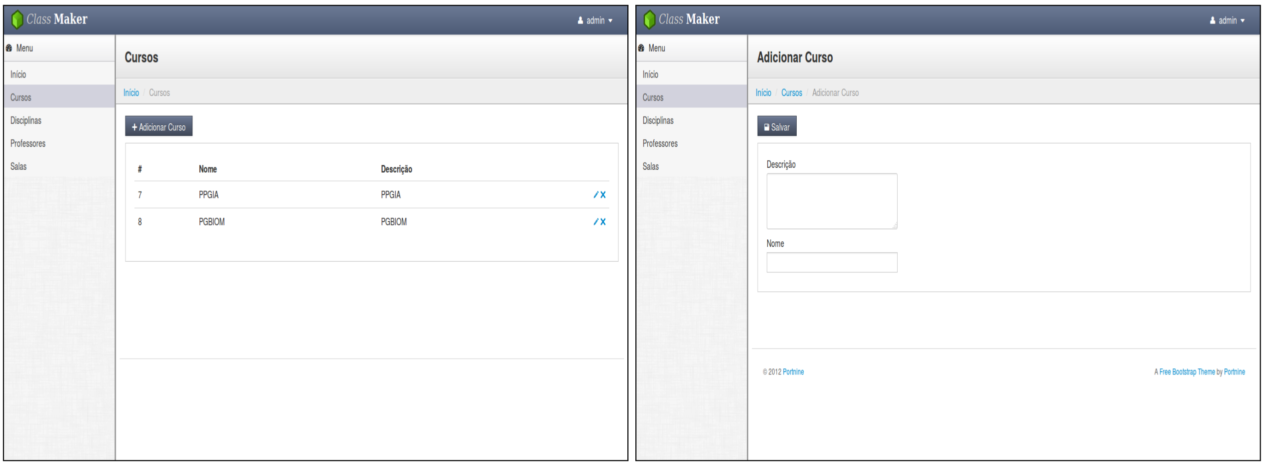


Figura 4.3: ClassMaker - Gerenciamento de Cursos

A 4.4 exibe as telas de gerenciamento de disciplinas.

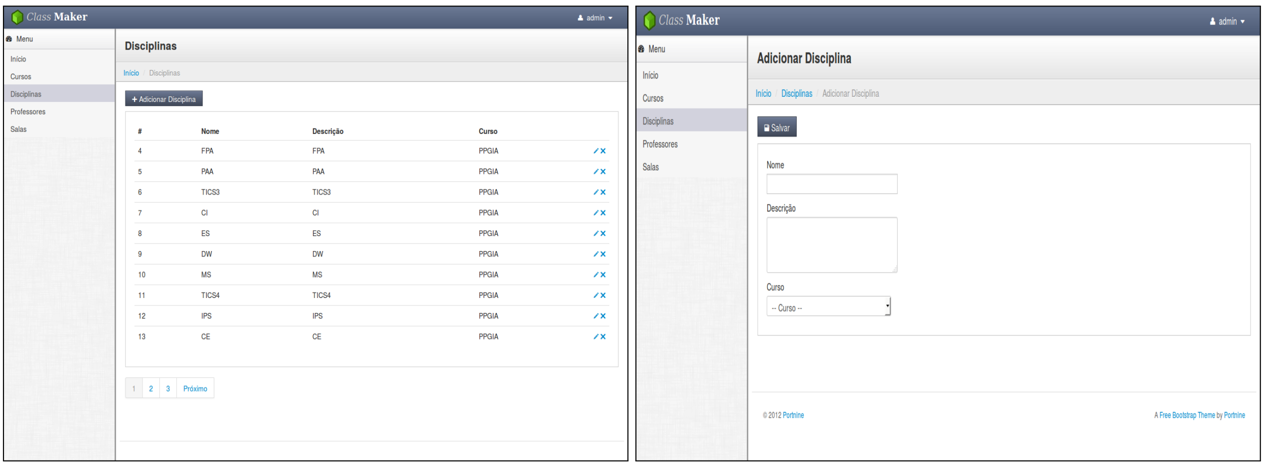


Figura 4.4: ClassMaker - Gerenciamento de Disciplinas

As figura 4.5 exibe as telas de gerenciamento de salas.

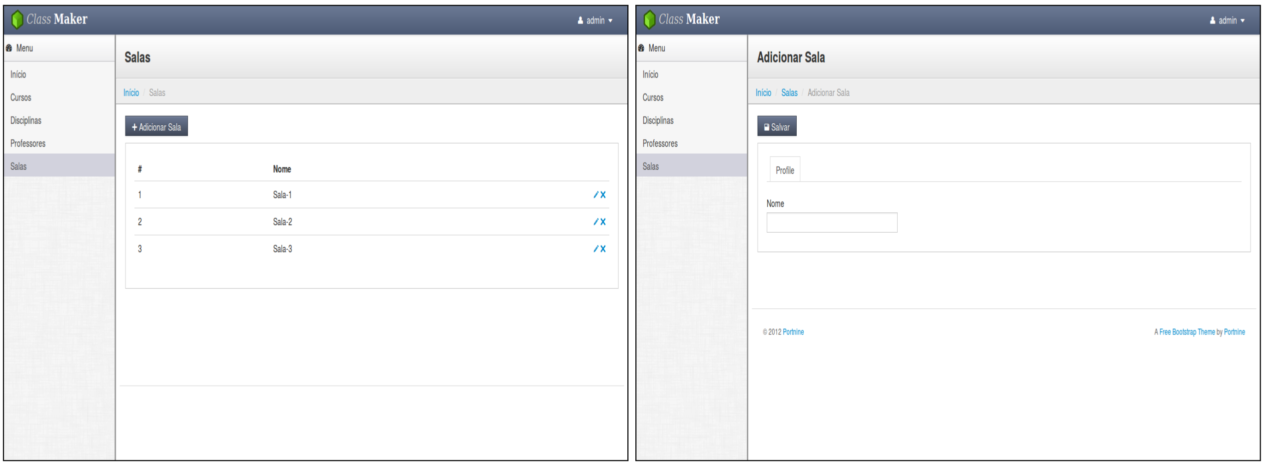


Figura 4.5: ClassMaker - Gerenciamento de Salas

A 4.6 exibe as telas de gerenciamento de professores.

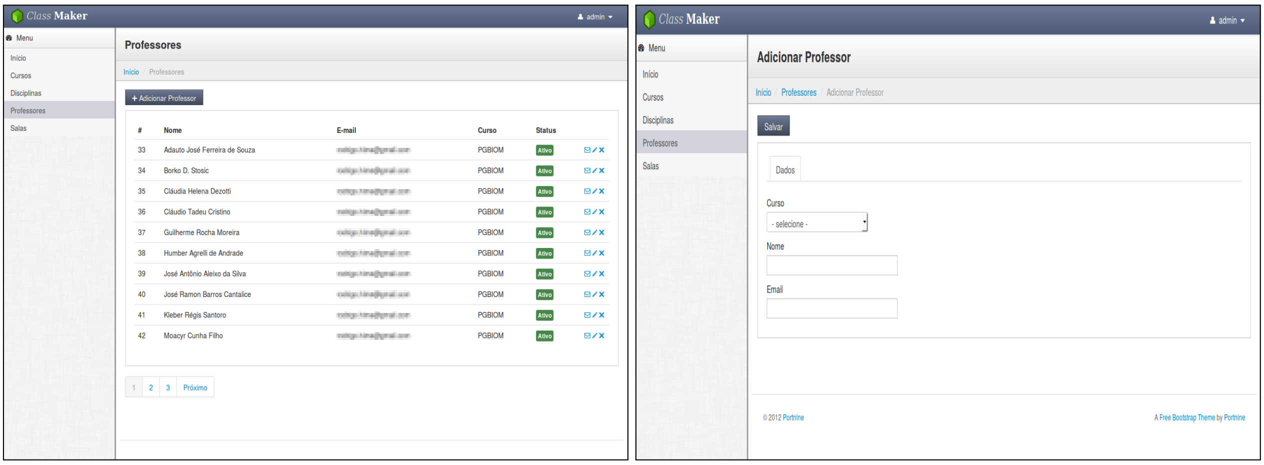


Figura 4.6: ClassMaker - Gerenciamento de Professores

Depois do cadastro de um professor o sistema envia um e-mail para o professor cadastrado com um link para o formulário de cadastro de suas preferências. A figura abaixo demonstra o e-mail enviado aos professores. A figura 4.7 ilustra o e-mail enviado.

Figura 4.7: ClassMaker - E-mail enviado pelo sistema

As figuras 4.8, 4.9, 4.10, 4.11 e 4.12 ilustram o cadastro das preferências dos professores no sistema.

Figura 4.8: ClassMaker - ClassMaker - Escolha das disciplinas que o professor pode lecionar

Figura 4.9: ClassMaker - Escolha das disciplinas em que o professor mais gostaria de lecionar

Figura 4.10: ClassMaker - Escolha do agrupamento da aulas - 1 ou 2 dias

Figura 4.11: ClassMaker - Cadastro das preferências de horários

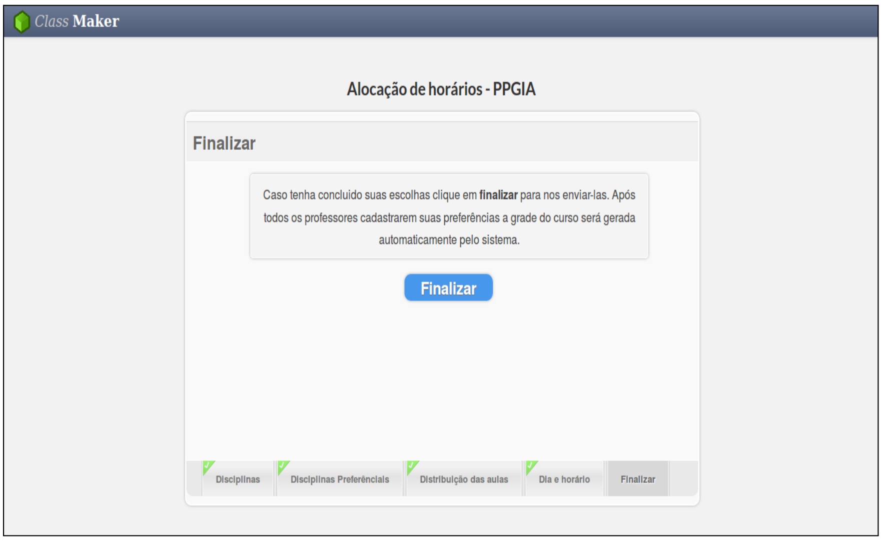


Figura 4.12: ClassMaker - Finalização do cadastro das preferências dos professores

Após todos os professores cadastrarem suas preferências o administrador do sistema pode gerar a grade de horário clicando no botão "Gerar Horário". O resultado pode ser visto na figura 4.13.

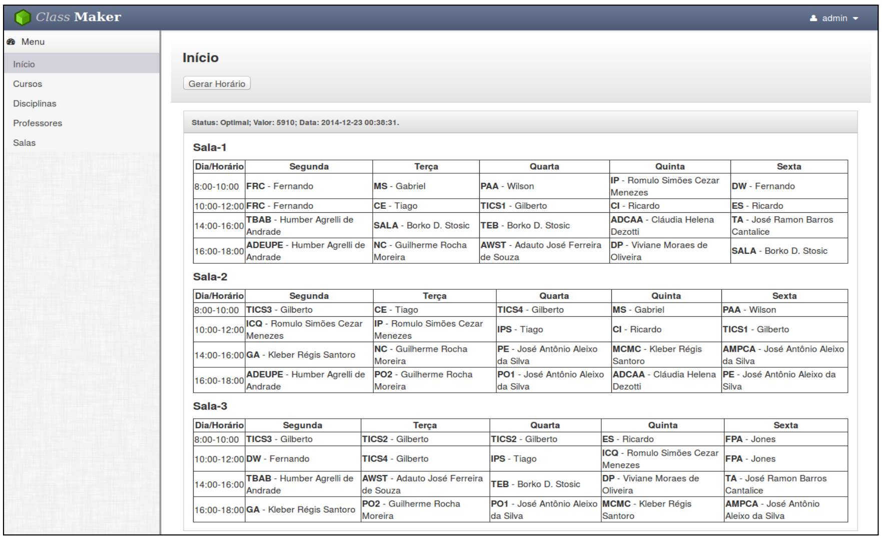


Figura 4.13: ClassMaker - Resultado - Grade de horário gerada

4.3 Resultados

Para realização dos experimentos foram cadastrados 30 professores, 58 turmas, 6 salas baseando-se nos cursos do Departamento de Estatística e Informática da UFRPE. Os professores cadastrados bem como as disciplinas foram obtidos com os funcionários responsáveis pelo gerenciamento dos horários no departamento. As preferências de turmas, preferências de horários e preferências de agrupamento das aulas em 1 ou 2 dos professores foram geradas de maneira arbitrária e não representam a real preferência dos professores. Os dados utilizados nos testes podem ser observados nas tabelas a 4.1, 4.2, 4.3, 4.4, 4.5 e 4.6:

Salas	
Índice	Sala
1	Sala-01
2	Sala-02
3	Sala-03
4	Sala-04
5	Sala-06
6	Sala-07

Tabela 4.1: Salas cadastradas

Turmas					
Índice	Turma	Índice	Turma	Índice	Turma
1	PPGIA-FPA	21	PGBIOM-DP	41	PGBIOM-MCMC
2	PPGIA-PAA	22	PGBIOM-ES	42	PGBIOM-NC
3	PPGIA-TICS3	23	PGBIOM-ENPACA	43	PGBIOM-PO1
4	PPGIA-CI	24	PGBIOM-F	44	PGBIOM-PO2
5	PPGIA-ES	25	PGBIOM-GB	45	PGBIOM-PE
6	PPGIA-DW	26	PGBIOM-GA	46	PGBIOM-P
7	PPGIA-MS	27	PGBIOM-IA	47	PGBIOM-PA
8	PPGIA-TICS4	28	PGBIOM-IB	48	PGBIOM-PQ
9	PPGIA-IPS	29	PGBIOM-IE	49	PGBIOM-ST
10	PPGIA-CE	30	PGBIOM-ICQ	50	PGBIOM-SGI
11	PPGIA-FRC	31	PGBIOM-IP	51	PGBIOM-SALA
12	PPGIA-TICS1	32	PGBIOM-LTG	52	PGBIOM-SDD
13	PPGIA-TICS2	33	PGBIOM-MAPS	53	PGBIOM-TA
14	PGBIOM-ADCAA	34	PGBIOM-MEAGM	54	PGBIOM-TAAC
15	PGBIOM-ADEUPE	35	PGBIOM-MEAPC	55	PGBIOM-TBAB
16	PGBIOM-ASACA	36	PGBIOM-MEAPA	56	PGBIOM-TMACA
17	PGBIOM-AMPCA	37	PGBIOM-MQACA	57	PGBIOM-TEB
18	PGBIOM-AWST	38	PGBIOM-MRAA	58	PGBIOM-USADB
19	PGBIOM-CGTA	39	PGBIOM-ML		
20	PGBIOM-DSAAAB	40	PGBIOM-MLG		

Tabela 4.2: Turmas Cadastradas

Professores			
Índice	Professor	Índice	Professor
1	Adauto José Ferreira de Souza	16	Viviane Moraes de Oliveira
2	Borko D. Stosic	17	Wilson Rosa de Oliveira Junior
3	Cláudia Helena Dezotti	18	Gilberto
4	Cláudio Tadeu Cristino	19	Tiago
5	Guilherme Rocha Moreira	20	Glauco
6	Humber Agrelli de Andrade	21	Giordano
7	José Antônio Aleixo da Silva	22	Jones
8	José Ramon Barros Cantalice	23	Ceça
9	Kleber Régis Santoro	24	Gabriel
10	Moacyr Cunha Filho	25	Fernando
11	Paulo José Duarte Neto	26	Ricardo
12	Rosângela Paula Teixeira Lessa	27	Cícero
13	Romulo Simões Cezar Menezes	28	Catão
14	Tatijana Stosic	29	Adriano
15	Tiago Alessandro Espínola Ferreira	30	Wilson

Tabela 4.3: Professores Cadastrados

Pesos das preferências de horário dos professores																															
Dia	Horário	Professor																													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	1	8	5	5	5	5	8	3	5	8	3	5	5	3	8	3	3	8	8	3	3	8	3	3	3	5	8	3	3	5	8
	2	8	5	5	5	5	8	3	5	8	3	8	5	3	8	3	3	8	8	3	3	8	3	3	3	5	8	3	3	5	8
	3	8	5	5	5	5	3	8	5	8	3	8	5	3	8	3	3	8	3	8	5	8	3	3	3	5	8	5	3	5	8
	4	8	5	5	5	5	3	8	5	8	3	5	5	3	8	3	3	8	3	8	5	8	3	3	3	5	8	5	3	5	8
2	1	3	8	3	8	3	8	5	5	8	5	3	5	3	5	5	3	8	8	3	3	8	8	8	8	5	8	8	3	5	3
	2	3	8	3	8	3	8	5	5	8	5	3	5	3	5	5	3	8	8	3	3	8	8	8	8	5	8	8	3	5	3
	3	5	8	3	8	3	3	5	5	8	5	5	5	3	5	5	3	8	3	8	8	8	8	8	8	5	3	3	8	5	3
	4	5	8	3	8	3	3	5	5	8	5	5	5	3	5	5	3	8	3	8	8	8	8	8	8	5	3	3	8	5	3
3	1	5	3	5	5	8	8	5	8	8	5	8	5	5	8	5	3	3	5	3	3	8	8	3	3	3	8	8	3	5	8
	2	5	3	5	5	8	8	5	8	8	5	8	5	5	8	5	3	3	5	3	3	8	8	3	3	3	8	8	3	5	8
	3	5	3	5	5	8	3	5	3	8	5	8	5	5	8	5	3	3	5	8	5	8	8	3	3	3	3	3	8	5	8
	4	5	3	5	5	8	3	5	3	8	5	8	5	5	8	5	3	3	5	8	5	8	8	3	3	3	3	3	8	5	8
4	1	5	8	5	8	5	8	3	5	8	5	3	5	5	5	5	8	3	8	3	3	3	8	8	5	8	5	8	3	5	8
	2	5	8	5	8	5	8	3	5	8	5	3	5	5	5	5	8	3	8	3	3	3	8	8	5	8	5	8	3	5	8
	3	5	8	5	8	5	5	8	5	8	5	3	5	5	5	5	8	3	3	8	8	3	8	8	5	8	3	3	8	5	8
	4	5	8	5	8	5	5	8	5	8	5	3	5	5	5	5	8	3	3	8	8	3	8	8	5	8	3	3	8	5	8
5	1	3	3	5	3	8	8	5	5	3	8	8	5	8	5	3	8	3	5	3	3	3	8	5	5	8	5	8	3	5	5
	2	3	3	5	3	8	8	5	5	3	8	8	5	8	5	3	8	3	5	3	3	3	8	5	5	8	5	8	3	5	5
	3	3	3	5	3	8	3	5	5	3	8	8	5	8	5	3	8	3	3	5	8	3	8	5	5	8	3	3	8	5	5
	4	3	3	5	3	8	3	5	5	3	8	8	5	8	5	3	8	3	3	5	8	3	8	5	5	8	3	3	8	5	5

Tabela 4.4: Pesos dos horários de preferência dos Professores

Pesos das preferências dos professores pelas turmas																														
Turmas	***Professores***																													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	3	3	3	0	13	0	3	0	8	8	3
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	8	0	8	3	0	8	8	8	0	3	0	3
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	3	0	3	8	3	0	0	0	3
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	3	0	0	3	0	13	8	13	3	3
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	3	3	0	0	8	8	3	0	8	3
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	3	0	3	0	0	0	3	3	8	3	3
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	13	3	8	0	0	8	0	8	3	13	8
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	8	8	3	0	0	8	0	0	3	8	0	3
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	3	8	0	0	13	0	8	0	13	0	3	3
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	13	0	8	8	8	3	0	0	0	8	0	8
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	3	0	8	0	8	3	0	3	0	3
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	3	3	8	3	0	0	0	0	0	0	3
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	8	0	3	3	3	3	3	13	3	8	3
14	3	3	8	3	0	0	13	3	3	8	3	8	3	13	8	3	8	0	0	0	0	0	0	0	0	0	0	0	0	0
15	8	3	3	3	3	3	0	8	13	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	13	13	8	8	8	8	0	8	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	1	0	3	3	3	3	13	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	8	8	0	13	13	0	0	0	8	0	3	3	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	8	0	3	3	3	0	0	0	3	0	8	8	8	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	3	8	3	3	0	8	0	0	3	8	0	13	13	0	0	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0
21	3	3	3	3	3	3	0	0	0	13	0	3	3	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0
22	8	3	8	8	8	0	8	8	13	0	3	8	0	0	0	3	8	0	0	0	0	0	0	0	0	0	0	0	0	0
23	8	0	3	0	8	0	8	13	8	8	3	3	3	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0
24	3	0	0	3	3	3	8	13	8	3	0	8	0	0	0	3	8	0	0	0	0	0	0	0	0	0	0	0	0	0
25	3	0	8	0	3	3	0	3	0	3	0	0	8	8	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0
26	3	13	3	3	3	8	0	8	0	3	0	3	3	0	8	8	13	0	0	0	0	0	0	0	0	0	0	0	0	0
27	13	3	3	3	3	3	8	0	0	8	0	8	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	3	3	8	0	0	0	0	0	0	0	0	3	13	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	3	8	0	0	3	0	8	8	3	3	0	8	0	3	3	13	8	0	0	0	0	0	0	0	0	0	0	0	0	0
30	3	3	3	3	13	0	0	3	8	8	0	0	0	0	3	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0
31	3	0	8	13	8	3	8	8	8	0	8	0	8	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0
32	3	0	3	8	3	0	0	0	3	0	0	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	0	3	3	3	3	3	8	0	0	3	0	8	3	3	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	8	0	0	3	8	0	0	3	3	8	3	13	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
35	0	8	3	3	3	3	3	8	0	8	0	8	3	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	0	13	13	13	13	13	13	3	0	0	8	3	3	8	3	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0
37	3	3	3	3	8	3	3	8	8	13	0	0	8	0	0	3	8	0	0	0	0	0	0	0	0	0	0	0	0	0
38	8	0	8	3	3	0	3	3	3	8	3	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
39	0	3	3	3	8	3	0	13	0	3	3	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
40	3	8	3	3	3	8	3	0	0	3	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
41	3	3	3	3	3	3	0	0	8	3	13	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0
42	0	0	13	0	8	0	8	0	3	8	8	8	3	0	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
43	0	0	3	8	3	0	0	8	8	8	8	8	8	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0
44	3	8	0	8	3	3	8	0	0	0	0	3	13	8	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0
45	0	3	8	3	13	8	8	3	0	0	0	13	3	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0
46	3	3	3	13	3	3	3	3	3	3	13	0	0	0	3	8	8	0	0	0	0	0	0	0	0	0	0	0	0	0
47	8	8	8	3	3	3	3	0	8	0	0	0	0	0	0	8	3	0	0	0	0	0	0	0	0	0	0	0	0	0
48	0	13	8	3	8	0	0	8	3	8	3	3	0	8	0	3	8	0	0	0	0	0	0	0	0	0	0	0	0	0
49	8	3	3	0	0	0	0	8	0	0	3	3	0	3	0	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0
50	3	8	13	8	0	0	0	3	0	3	3	8	8	3	3	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0
51	3	3	0	0	3	8	3	3	8	3	0	8	3	0	3	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0
52	8	0	8	13	0	0	0	8	0	8	3	3	3	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0
53	3	0	8	3	0	0	0	8	3	3	3	8	8	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0
54	0	3	3	3	3	3	3	0	8	3	0	3	3	3	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0
55	13	8	3	3	3	8	3	0	3	8	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
56	3	13	0	8	0	3	3	3	8	3	0	0	3	0	0	8	8	0	0	0	0	0	0	0	0	0	0	0	0	0
57	3	3	13	3	3	3	3	3	3	8	0	0	3	3	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0
58	0	0	8	3	0	0	0	8	0	0	3	8	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Tabela 4.5: Preferência de turmas dos Professores

Preferências dos professores pelas aulas em 1 ou 2 dias para as turmas																														
Turmas	***Professores***																													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	2	1	1	2	2	1	2	1	2	2	2	2	2	1	1	1	2	1	2	1	2	2	2	2	1	2	1	2	2	2
2	1	2	1	2	2	2	1	1	2	2	2	2	2	2	1	1	2	2	2	2	2	2	2	1	2	2	1	2	2	2
3	2	1	2	2	2	2	2	1	2	2	2	2	2	2	1	2	2	2	2	2	2	2	1	2	1	2	1	2	2	2
4	2	2	2	2	1	2	2	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1	2	1	2	2	1	1	2	2
5	2	2	2	2	2	1	2	1	2	2	2	2	2	2	1	2	2	2	2	2	2	2	2	2	1	2	1	2	2	2
6	1	1	1	2	1	2	2	1	2	2	2	2	2	2	1	1	2	2	2	2	2	2	2	2	2	2	1	2	2	1
7	2	1	2	2	2	2	2	1	2	2	2	2	2	2	1	2	2	2	2	2	2	2	2	1	2	2	1	1	2	2
8	2	2	1	2	2	2	2	1	2	2	2	2	2	2	1	1	2	2	1	2	2	2	2	2	1	2	1	2	2	2
9	2	2	1	2	2	2	2	1	2	1	2	2	2	2	1	1	2	1	2	2	1	2	2	1	2	2	1	2	2	2
10	2	2	1	2	2	2	2	2	2	2	2	2	2	2	1	2	1	2	2	2	1	2	2	1	2	2	1	2	2	2
11	1	1	1	2	2	2	2	2	2	2	2	2	2	2	1	1	2	2	1	2	1	1	2	1	2	2	1	1	2	2
12	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	2	2	2	2	2	2	2	2	2	2	2	1	2	2	2
13	1	2	1	2	2	2	2	2	2	2	2	2	2	2	1	1	2	2	2	2	2	2	2	1	2	2	1	2	2	1
14	2	2	2	2	2	2	2	1	1	2	2	1	2	2	1	2	2	2	2	2	2	2	1	2	2	2	1	2	2	1
15	2	2	2	2	2	2	2	1	2	1	2	2	2	1	1	2	2	2	2	2	2	2	2	1	2	2	2	2	2	2
16	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1	1	2	2	2	2	2	2	2	1	1	1	1	2	1	1
17	2	2	1	2	2	2	2	1	2	2	2	2	2	2	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
18	1	1	1	2	2	2	2	1	2	2	2	2	2	1	1	1	2	2	2	2	2	2	2	1	1	1	1	1	1	1
19	2	1	1	2	1	2	2	1	2	2	2	1	2	2	1	1	2	2	2	2	2	2	2	1	1	1	1	1	1	1
20	2	2	1	2	2	2	2	1	2	1	2	2	2	2	1	1	2	2	2	2	2	2	2	1	1	1	1	1	1	1
21	2	1	2	2	2	2	2	1	1	1	2	2	2	2	1	1	2	2	2	2	2	2	2	1	1	1	1	1	1	1
22	2	2	1	2	2	2	2	1	2	2	2	2	2	1	1	1	2	2	2	2	2	2	2	1	1	1	1	1	1	1
23	2	2	1	2	2	2	2	1	2	2	2	2	2	2	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
24	2	2	2	2	2	2	2	1	2	2	2	2	2	2	1	2	2	1	2	2	2	2	2	2	2	2	2	2	2	2
25	2	2	2	2	1	2	2	1	2	2	2	2	2	2	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
26	1	1	2	1	2	1	2	1	2	2	2	2	2	2	1	1	2	2	2	2	2	2	2	1	1	1	1	1	1	1
27	2	2	2	2	2	2	2	1	2	2	1	2	2	2	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
28	1	2	1	2	2	2	2	1	2	1	2	2	2	1	1	1	2	2	2	2	2	2	2	1	1	1	1	1	1	1
29	2	1	1	1	2	2	2	1	2	2	2	2	2	2	1	1	2	2	2	2	2	2	2	1	1	1	1	1	1	1
30	2	2	1	2	2	2	2	1	2	2	2	2	2	2	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
31	1	1	2	1	2	2	2	1	1	2	2	2	2	1	1	1	2	2	2	2	2	2	2	1	1	1	1	1	1	1
32	2	2	2	2	2	2	2	1	2	2	2	2	2	2	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
33	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1	1	2	2	2	2	2	2	2	1	1	1	1	1	1	1
34	2	2	2	2	2	2	2	1	2	2	2	2	2	2	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
35	2	2	2	2	2	2	2	1	1	2	2	2	2	2	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
36	1	2	2	2	2	1	2	2	2	2	1	2	2	2	1	1	2	2	2	2	2	2	2	1	1	1	1	1	1	1
37	2	2	2	2	1	2	2	1	2	2	2	2	2	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
38	1	2	2	1	1	2	1	1	2	2	2	2	2	2	1	1	2	2	2	2	2	2	2	1	1	1	1	1	1	1
39	2	2	2	2	2	2	2	1	2	2	2	2	2	2	1	1	2	2	2	2	2	2	2	1	1	1	1	1	1	1
40	2	1	2	2	2	2	2	1	2	2	2	2	2	2	1	1	2	2	2	2	2	2	2	1	1	1	1	1	1	1
41	2	1	2	2	2	2	2	1	1	2	2	2	2	2	1	1	2	2	2	2	2	2	2	1	1	1	1	1	1	1
42	1	1	1	2	2	2	2	1	2	2	1	2	2	2	1	1	2	2	2	2	2	2	2	1	1	1	1	1	1	1
43	2	2	2	1	1	2	2	1	2	2	2	2	2	2	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
44	2	2	2	2	1	2	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
45	2	2	1	2	2	2	2	1	2	2	2	2	2	2	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
46	1	2	2	2	2	2	2	1	2	1	2	2	2	2	1	1	2	2	2	2	2	2	2	1	1	1	1	1	1	1
47	2	2	1	2	2	2	2	1	2	2	2	2	2	2	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
48	1	2	2	2	2	2	2	1	2	1	2	2	2	2	1	1	2	2	2	2	2	2	2	1	1	1	1	1	1	1
49	2	2	2	2	1	2	2	1	2	1	2	2	2	2	1	1	2	2	2	2	2	2	2	1	1	1	1	1	1	1
50	2	2	2	2	2	2	1	1	2	2	2	2	2	2	1	1	2	2	2	2	2	2	2	1	1	1	1	1	1	1
51	2	2	2	2	2	1	2	1	2	2	2	2	2	2	1	1	2	2	2	2	2	2	2	1	1	1	1	1	1	1
52	2	2	8	2	2	2	2	1	2	2	2	2	2	2	1	1	2	2	2	2	2	2	2	1	1	1	1	1	1	1
53	2	2	2	2	2	2	2	1	2	2	2	2	2	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
54	2	2	1	2	2	2	2	1	2	2	2	2	2	2	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
55	2	2	2	2	2	2	2	1	2	2	2	2	2	2	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
56	1	2	2	1	2	2	2	1	2	2	2	2	2	2	1	1	2	2	2	2	2	2	2	1	1	1	1	1	1	1
57	2	2	2	2	2	2	2	1	2	1	2	2	2	2	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
58	2	2	2	2	2	0	1	1	2	2	2	2	2	2	1	1	2	2	2	2	2	2	2	1	1	1	1	1	1	1

Tabela 4.6: Preferência de alocação em 1 ou 2 dias nas turmas

Os resultados foram bastante satisfatórios pois o tempo para o sistema gerar a grade de horários resultante foi de apenas 14 minutos e 20 segundos. Todas as restrições rígidas foram respeitadas. Em relação às premissas definidas na função objetivo, todas as restrições rígidas foram respeitadas e apenas um período vago foi encontrado entre as aulas de todos os professores.

O resultado da alocação pode ser verificado nas tabelas abaixo:

Sala 1				
Segunda-feira	Terça-feira	Quarta-feira	Quinta-Feira	Sexta-Feira
8:00 - 10:00 PPGIA-TICS3: Gilberto	8:00 - 10:00 PPGIA-TICS3: Gilberto	8:00 - 10:00 PGBIOM-ENPACA: José Ramon Barros Cantalice	8:00 - 10:00 PPGIA-DW: Jones	8:00 - 10:00 PGBIOM-PO2: Romulo Simões Cezar Menezes
10:00 - 12:00 PGBIOM-P: Paulo José Duarte Neto	10:00 - 12:00 PGBIOM-IP: Cláudio Tadeu Cristino	10:00 - 12:00 PGBIOM-ENPACA: José Ramon Barros Cantalice	10:00 - 12:00 PGBIOM-IP: Cláudio Tadeu Cristino	10:00 - 12:00 PGBIOM-SGI: Cláudia Helena Dezotti
14:00 - 16:00 PGBIOM-SGI: Cláudia Helena Dezotti	14:00 - 16:00 PGBIOM-TA: Cláudio Tadeu Cristino	14:00 - 16:00 PGBIOM-P: Paulo José Duarte Neto	14:00 - 16:00 PGBIOM-PO2: Romulo Simões Cezar Menezes	14:00 - 16:00 PGBIOM-NC: Guilherme Rocha Moreira
16:00 - 18:00 PGBIOM-AMPCA: José Antônio Aleixo da Silva	16:00 - 18:00 PGBIOM-TA: Cláudio Tadeu Cristino	16:00 - 18:00 PPGIA-DW: Jones	16:00 - 18:00 PGBIOM-AMPCA: José Antônio Aleixo da Silva	16:00 - 18:00 PGBIOM-NC: Guilherme Rocha Moreira

Tabela 4.7: Resultado - Sala 1

Sala 2				
Segunda-feira	Terça-feira	Quarta-feira	Quinta-Feira	Sexta-Feira
8:00 - 10:00 PGBIOM-IA: Adauto José Ferreira de Souza	8:00 - 10:00 PGBIOM-LTG: Cláudio Tadeu Cristino	8:00 - 10:00 PPGIA-ES: Ricardo	8:00 - 10:00 PGBIOM-LTG: Cláudio Tadeu Cristino	8:00 - 10:00 PPGIA-TICS2: Cícero
10:00 - 12:00 PGBIOM-IA: Adauto José Ferreira de Souza	10:00 - 12:00 PGBIOM-PQ: Borko D. Stosic	10:00 - 12:00 PPGIA-TICS2: Cícero	10:00 - 12:00 PGBIOM-PQ: Borko D. Stosic	10:00 - 12:00 PGBIOM-IB: Romulo Simões Cezar Menezes
14:00 - 16:00 PGBIOM-F: Kleber Régis Santoro	14:00 - 16:00 PGBIOM-MEAPC: Borko D. Stosic	14:00 - 16:00 PGBIOM-IB: Romulo Simões Cezar Menezes	14:00 - 16:00 PPGIA-TICS1: Tiago	14:00 - 16:00 PGBIOM-DSAAAB: Romulo Simões Cezar Menezes
16:00 - 18:00 PPGIA-ES: Ricardo	16:00 - 18:00 PGBIOM-MEAPC: Borko D. Stosic	16:00 - 18:00 PGBIOM-F: Kleber Régis Santoro	16:00 - 18:00 PPGIA-TICS1: Tiago	16:00 - 18:00 PGBIOM-DSAAAB: Romulo Simões Cezar Menezes

Tabela 4.8: Resultado - Sala 2

Sala 3				
Segunda-feira	Terça-feira	Quarta-feira	Quinta-Feira	Sexta-Feira
8:00 - 10:00 PGBIOM-MEAPA: Wilson Rosa de Oliveira Junior	8:00 - 10:00 PGBIOM-TMACA: Borko D. Stosic	8:00 - 10:00 PGBIOM-ICQ: Guilherme Rocha Moreira	8:00 - 10:00 PGBIOM-TMACA: Borko D. Stosic	8:00 - 10:00 PGBIOM-IE: Viviane Moraes de Oliveira
10:00 - 12:00 PGBIOM-GB: Wilson Rosa de Oliveira Junior	10:00 - 12:00 PGBIOM-GA: Wilson Rosa de Oliveira Junior	10:00 - 12:00 PPGIA-PAA: Giordano	10:00 - 12:00 PGBIOM-IE: Viviane Moraes de Oliveira	10:00 - 12:00 PGBIOM-ICQ: Guilherme Rocha Moreira
14:00 - 16:00 PGBIOM-GA: Wilson Rosa de Oliveira Junior	14:00 - 16:00 PGBIOM-GB: Wilson Rosa de Oliveira Junior	14:00 - 16:00 PGBIOM-ML: Guilherme Rocha Moreira	14:00 - 16:00 PGBIOM-USADB: Rosângela Paula Teixeira Lessa	14:00 - 16:00 PGBIOM-DP: Moacyr Cunha Filho
16:00 - 18:00 PPGIA-PAA: Giordano	16:00 - 18:00 PGBIOM-MEAPA: Wilson Rosa de Oliveira Junior	16:00 - 18:00 PGBIOM-ML: Guilherme Rocha Moreira	16:00 - 18:00 PGBIOM-USADB: Rosângela Paula Teixeira Lessa	16:00 - 18:00 PGBIOM-DP: Moacyr Cunha Filho

Tabela 4.9: Resultado - Sala 3

Sala 4				
Segunda-feira	Terça-feira	Quarta-feira	Quinta-Feira	Sexta-Feira
8:00 - 10:00	8:00 - 10:00 PPGIA-FPA: Gabriel	8:00 - 10:00	8:00 - 10:00 PGBIOM-MEAGM: Rosângela Paula Teixeira Lessa	8:00 - 10:00 PGBIOM-MRAA: Cláudia Helena Dezotti
10:00 - 12:00 PGBIOM-SALA: Kleber Régis Santoro	10:00 - 12:00 PPGIA-FPA: Gabriel	10:00 - 12:00 PGBIOM-SALA: Kleber Régis Santoro	10:00 - 12:00 PGBIOM-MEAGM: Rosângela Paula Teixeira Lessa	10:00 - 12:00 PGBIOM-MCMC: Paulo José Duarte Neto
14:00 - 16:00 PGBIOM-MCMC: Paulo José Duarte Neto	14:00 - 16:00 PPGIA-TICS4: Gabriel	14:00 - 16:00 PGBIOM-ADCAA: Tatijana Stosic	14:00 - 16:00 PGBIOM-MRAA: Cláudia Helena Dezotti	14:00 - 16:00 PPGIA-FRC: Fernando
16:00 - 18:00 PGBIOM-SDD: Cláudia Helena Dezotti	16:00 - 18:00 PPGIA-TICS4: Gabriel	16:00 - 18:00 PGBIOM-ADCAA: Tatijana Stosic	16:00 - 18:00 PGBIOM-SDD: Cláudia Helena Dezotti	16:00 - 18:00 PPGIA-FRC: Fernando

Tabela 4.10: Resultado - Sala 4

Sala 5				
Segunda-feira	Terça-feira	Quarta-feira	Quinta-Feira	Sexta-Feira
8:00 - 10:00 PGBIOM-MLG: Humber Agrelli de Andrade	8:00 - 10:00 PGBIOM-MLG: Humber Agrelli de Andrade	8:00 - 10:00 PGBIOM-PA: Kleber Régis Santoro	8:00 - 10:00 PPGIA-IPS: Ceça	8:00 - 10:00 PGBIOM-MQACA: Moacyr Cunha Filho
10:00 - 12:00	10:00 - 12:00 PPGIA-CI: Ricardo	10:00 - 12:00 PPGIA-CI: Ricardo	10:00 - 12:00 PPGIA-IPS: Ceça	10:00 - 12:00 PGBIOM-MQACA: Moacyr Cunha Filho
14:00 - 16:00 PGBIOM-MAPS: José Antônio Aleixo da Silva	14:00 - 16:00 PGBIOM-PA: Kleber Régis Santoro	14:00 - 16:00 PPGIA-CE: Tiago	14:00 - 16:00 PGBIOM-MAPS: José Antônio Aleixo da Silva	14:00 - 16:00 PGBIOM-PO1: Paulo José Duarte Neto
16:00 - 18:00	16:00 - 18:00 PPGIA-MS: Glauco	16:00 - 18:00 PPGIA-CE: Tiago	16:00 - 18:00 PPGIA-MS: Glauco	16:00 - 18:00 PGBIOM-PO1: Paulo José Duarte Neto

Tabela 4.11: Resultado - Sala 5

Sala 6				
Segunda-feira	Terça-feira	Quarta-feira	Quinta-Feira	Sexta-Feira
8:00 - 10:00 PGBIOM-ADEUPE: Kleber Régis Santoro	8:00 - 10:00 PGBIOM-TAACA: Kleber Régis Santoro	8:00 - 10:00 PGBIOM-ST: Adauto José Ferreira de Souza	8:00 - 10:00 PGBIOM-ES: Kleber Régis Santoro	8:00 - 10:00 PGBIOM-PE: Guilherme Rocha Moreira
10:00 - 12:00 PGBIOM-AWST: Humber Agrelli de Andrade	10:00 - 12:00 PGBIOM-TAACA: Kleber Régis Santoro	10:00 - 12:00 PGBIOM-PE: Guilherme Rocha Moreira	10:00 - 12:00 PGBIOM-CGTA: Romulo Simões Cezar Menezes	10:00 - 12:00 PGBIOM-AWST: Humber Agrelli de Andrade
14:00 - 16:00 PGBIOM-TBAB: Adauto José Ferreira de Souza	14:00 - 16:00 PGBIOM-ST: Adauto José Ferreira de Souza	14:00 - 16:00 PGBIOM-ES: Kleber Régis Santoro	14:00 - 16:00 PGBIOM-ASACA: Borko D. Stosic	14:00 - 16:00 PGBIOM-TEB: Cláudia Helena Dezotti
16:00 - 18:00 PGBIOM-TBAB: Adauto José Ferreira de Souza	16:00 - 18:00 PGBIOM-ST: Adauto José Ferreira de Souza	16:00 - 18:00 PGBIOM-CGTA: Romulo Simões Cezar Menezes	16:00 - 18:00 PGBIOM-ASACA: Borko D. Stosic	16:00 - 18:00 PGBIOM-TEB: Cláudia Helena Dezotti

Tabela 4.12: Resultado - Sala 6

Capítulo 5

Considerações Finais

Este trabalho apresentou uma aplicação da programação inteira no *School Timetabling Problem*. Neste capítulo consta a conclusão do trabalho, bem como possíveis trabalhos futuros.

5.1 Conclusão do Trabalho

O *School Timetablin Problem* tem a sua definição bastante diversificada de acordo com as características de cada instituição e suas restrições, sejam físicas, organizaçionais ou pessoais. Este trabalho apresentou um modelo genérico baseado nos cursos de Pós-Graduação do Departamento de Estatística e Informática da Universidade Federal Rural de Pernambuco mostrando que é possível automatizar o processo de criação da grade horária de cursos e escolas, utilizando um modelo de Programação Inteira. Apesar de sua utilização para este tipo de problema não seja recomendada, devido a sua elevada complexidade computacional, a utilização da Programação Inteira se mostrou uma excelente alternativa para a resulção de instâncias pequenas do problema.

5.2 Trabalhos Futuros

Um possível trabalho futuro seria a incrementação do modelo desenvolvido visando cobrir todos os seus requisitos para a geração da grade de horários cursos do Departamento de Estatística e Informática da UFRPE, para que o sistema possa ser utilizando pelos funcionários do departamento. Outro possível trabalho seria executar testes de carga para avaliar o limite e o desempenho do sistema em instancias maiores do problema.

Este trabalho também pode servir de base para a resolução desse problema utilizando métodos heurísticos ao invés da programação inteira a fim de resolver instâncias grandes do problema.

Referências Bibliográficas

[1] EO Abensur and RC de Oliveira. Um método heurístico contrutivo para o problema da grade horária escolar. *Revista Eletrônica Pequisa Operacional para o Desenvolvimento*, 4:230–248, 2012.

[2] Guilherme Brandelli Bucco. *Contrução de um modelo de programação linear para o University Timetabling Probblem*. Dissertação, Universidade Federal do Rio Grande do Sul, 2014.

[3] Luiz Carlos and Gomes Filho. *Alocação deprofessores em horários do plantão de dúvidas para o curso e colégio Objetivo: Uma abordagem heurística*. Dissertação, Escola Politécnica da USP, 2006.

[4] Rodrigo Carvalho. *Abordagem heurística para o problema de programação de horários de cursos*. Dissertação, Universidade Federal de Minas Gerais, 2011.

[5] AM Coelho. *Uma abordagem via algoritmos meméticos para a soluçao do problema de horário escolar*. Dissertação, CEFET-MG, 2006.

[6] COIN-OR. http://www.coin-or.org/, 2014.

[7] CPLEX. http://www.cplex.com/, 2014.

[8] Abdelaziz Dammak, Abdelkarim Elloumi, Hichem Kamoun, and Jacques a. Ferland. Course timetabling at a Tunisian University: A case study. *Journal of Systems Science and Systems Engineering*, 17(3):334–352, July 2008.

[9] GLPK. http://www.gnu.org/software/glpk/glpk.html, 2014.

[10] GUROBI. http://www.gurobi.com/, 2014.

[11] EG da Silva Kotsko. Otimização na alocação de professores na construção de uma grade horária escolar. *Revista do Centro de Ciências Agrárias e Ambientais*, 1:31–45, June 2005.

[12] ELM Lobo. *Uma solução do problema de horário escolar via algoritmo genético paralelo*. Dissertação, CEFET-MG, 2005.

[13] Jean Paulo Martins. *O Problema do Agendamento Semanal de Aulas*. Dissertação, Universidade Federal de Goiás, 2010.

[14] Iago Carrara Roque Mateus Tartaglia, Lana Mara Rodrigues dos Santos. Um modelo matemático de programação linear inteira para a alocação de horários na Escola Estadual Effie Rolfs. In *Simpósio Acadêmico de Engenharua de Produção*, pages 1–10, 2013.

[15] S Mitchell. An introduction to pulp for Python programmers. Technical report, Light Metals Research Center, University of Auckland, Auckland, New Zeland, 2009.

[16] Stuart Mitchell, M O'Sullivan, and Iain Dunning. PuLP: a linear programming toolkit for python. Technical report, Department of Engineering Science, The University of Auckland, Auckland, New Zeland, 2011.

[17] Arnaldo Moura, Rafael Scaraficci, Rafael Silveira, and Volnei dos Santos. Técnicas Metaheurísticas Aplicadas À Construção De Grades Horárias Escolares. *XXXVI SBPO - O Impacto da Pesquisa Operacional nas Novas Tendências Multidisciplinares*, pages 1–12, 2004.

[18] Keith Murray, T Müller, and H Rudová. Modeling and solution of a complex university course timetabling problem. Technical report, Purdue University, West Lafayette, USA, 2007.

[19] Fernanda Navarro and Frederico Coelho. Estudo de Coloração Aplicado ao Problema de Alocação de Horário de Professores. Technical report, Departamento de Ciência da Computação - UNIPAC, Barbacena, MG - Brasil, 2009.

[20] Gurobi Optimization. Gurobi optimizer quick start guide. Technical report, 2014.

[21] Optimization with PuLP. https://www.coin-or.org/PuLP/, 2014.

[22] Marta Pina and Cardoso Unitau. Estudo para automação de horários escolares em uma instituição de ensino. In *3 Simpósio Hepertexto e Tecnologias na Educação*, volume 1, pages 1–20, Recife, PE - Brasil, 2010.

[23] Camilo José Bornia Poulsen. *Desenvolvimento de um Modelo para o School Timetabling Problem Baseado na Meta-Heurística Simulated Annealing*. Dissertação, Universidade Federal do Rio Grande do Sul, 2012.

[24] O. Sousa. *Aspectos Práticos da Programação Linear*. Dissertação, Universidade Federal de Santa Catarina, 2009.

[25] Morgana Spindler. *Uma proposta de solução para problemas horário educacional utilizando busca dispersa e reconexão de caminhos*. Dissertação, UNIVERSIDADE DO VALE DO RIO DOS SINOS, 2010.

[26] F Vieira and H Macedo. Sistema de alocação de horários de cursos universitários: um estudo de caso no departamento de computação da Universidade Federal de Sergipe. *Scientia Plena*, 7:1–12, 2011.

[27] Jean-Charles Billaut Vincent T'kindt. *Multicriteria Scheduling Second Edition*, volume 40. Springer, March 2001.

[28] Bruno Missi Xavier, Dalessandro Soares Vianna, Helder Gomes Costa, Universidade Federal Fluminense, Willen Borges Coelho, and Timetabling Problems. Proposta de Alocação de Horários de Professores e Turmas em Instituições de Ensino Superior Utilizando uma Heurística VNS/VND. In *XLVSBPO - Simpósio Brasileiro de Pesquisa Operacional*, pages 832–844, Natal, RN - Brasil, 2013.

Apêndice A

Exemplo de arquivo LP

```
Minimize
obj: - 2 x3

Subject To
c1: x2 - x1 <= 10
c2: x1 + x2 + x3 <= 20

Bounds
x1 <= 30
2 <= x3 <= 3

s.i.
x3
x1 >= 2.1

End
```