



Universidade Federal de Sergipe  
Centro de Ciências Exatas e Tecnologia  
Departamento de Computação  
Disciplina: Inteligência Artificial  
Professora: Carlos Alberto Estombelo Montesco  
Alunos: Adam Cordeiro Araujo, Felipe Queiroz Barreto e João  
Victor Dantas Batista

## Relatório do Projeto Final - Problemas de satisfação de restrições

### 1. Modelagem do problema

Nos estágios iniciais de modelagem do problema, a equipe considerou tratar os blocos de tempo como variáveis do problema enquanto que as atividades seriam consideradas domínio. Entretanto, lendo o livro *AIMA 3rd Ed.* e em conversa, decidimos optar por inverter a ideia e tratar como domínio os blocos de tempo enquanto que as variáveis seriam as aulas, atividades complementares, blocos de estudo e tempo de descanso.

#### 1.1. Definição do domínio

O domínio do problema compreende os intervalos de **30min** dentro da tabela de horários. Considerando que o horário total disponível diariamente é de **10 horas** e inicialmente são considerados os dias correspondentes de segunda a sexta, temos um total de **100 blocos** de **30min** por semana. A exceção ocorre quando já não há intervalos disponíveis nos horários de segunda a sexta e podemos considerar também o mesmo período de **10h diárias** para o **sábado** (ou seja, existem de fato **mais 20 blocos** ao domínio, que podem ou não ser utilizados). Um certo conjunto de elementos do domínio sempre será atribuído a uma determinada variável.

#### 1.2. Definição das variáveis

As variáveis do problema compreendem todas as possíveis atividades a serem desempenhadas pelo estudante ao longo do seu semestre, semanalmente. São essas atividades que serão inseridas na tabela para montar o horário pessoal do aluno. Elas estão divididas em:

- **Aulas** - por definição possui carga horária fixa de no mínimo blocos de 2 horas dentro da tabela e são definidas de maneira rígida nos horários, podem ser colocadas em qualquer horário e podem se repetir dentro de um mesmo dia ou durante a semana.
- **Atividades complementares** - atividades nas quais o discente pode estar participando. Algumas delas possuem carga horária semanal fixa e outras não podem ser desenvolvidas concomitantemente. O próprio aluno é responsável por determinar a distribuição dessa carga horária ao longo da semana. Os tipos de atividades são:
  - **Iniciação científica (PIBIC)** - carga horária semanal de 20h
  - **Iniciação tecnológica (PIBITI)** - carga horária semanal de 20h
  - **Iniciação à extensão (PIBIX)** - carga horária semanal de 20h
  - **Estágio** - carga horária semanal à depender do local de trabalho
  - **Trabalho** - carga horária semanal à depender do local de trabalho
  - **Atividade voluntária** - carga horária semanal à depender do local de realização da atividade
  - **Palestra/Evento extracurricular** - carga horária semanal a depender do discente
  - **Outros** - carga horária semanal a depender do discente
- **Blocos de estudo** - por definição, cada bloco possui 30min no mínimo dentro da tabela de horários e é obrigatório que para cada disciplina escolhida pelo discente, deve haver um bloco de estudo correspondente que podem estar espalhados de maneira caótica dentro da tabela se for para cumprir os horários propostos.
- **Período de descanso:** por definição, o discente pode definir alguns blocos de descanso e/ou refeições ao longo do horário, podendo ser quantos ele preferir. O período de descanso é pré-definido pelo discente tal qual as aulas também são.

### 1.3. Definição das restrições

As restrições são responsáveis por especificar combinações de valores permitidas. A solução torna-se consistente uma vez que não exista nenhuma restrição violada. Neste problema, elas foram definidas da seguinte forma:

- **R01** - A resolução do código se dá de maneira que as **aulas** têm prioridade sobre as **atividades complementares** que por sua vez tem

prioridade sobre os **blocos de estudo**, quando se refere a ordem de alocação das atividades.

- **R02** - Um determinado intervalo de horários só pode ser atribuído a uma atividade, ou seja, não pode haver sobreposição de atividades. Sejam  $v$  e  $w$  variáveis quaisquer do conjunto  $V$  das variáveis do problema:

$$v \neq w$$

- **R03** - A quantidade total de horas de aulas, atividades complementares e blocos de estudo não pode ultrapassar a quantidade total de horas disponíveis na semana. Seja  $h$  a quantidade total de horas disponíveis em um dia  $d$ , temos:

$$h_d \leq 20$$

- **R04** - As atividades **PIBITI** e **PIBIC** são mutuamente excludentes, só podendo escolher para realizar uma das atividades por vez. Logo,

$$Pibiti \vee Pibic$$

- **R05** - **PIBIC** ( $P_c$ ) e **PIBITI** ( $P_{ti}$ ) precisam ter a carga horária de 20h semanais. Seja  $t$  a carga horária semanal da atividade:

$$t_{P_c} = 20 \text{ ou } t_{P_{ti}} = 20$$

- **R06** - As atividades **Trabalho** e **Estágio** são mutuamente excludentes, só podendo escolher para realizar uma das atividades por vez. Logo,

$$Trabalho \vee Estágio$$

- **R07** - O sábado só é utilizado caso o horário máximo dos 5 dias úteis (segunda a sexta) não seja suficiente para lidar com as atividades não alocadas. Ou caso haja alguma atividade que necessariamente necessite do sábado.
- **R08** - Blocos de atividades complementares devem, na medida do possível, estar sempre juntos.
- **R09** - Para cada aula (seja uma aula qualquer  $a$ , do conjunto de aulas  $A$ ), deve haver ao menos um bloco de estudos (representado por  $b$ ) de 30min. A duração semanal dos blocos podem ser definidas pelo

discente, bem como a quebra desse bloco em subdivisões. O que o discente não define é onde serão alocados cada um destes blocos.

$$\forall a \in A, b_a \geq 30$$

## 2. Escolha da heurística

A heurística escolhida para resolvermos este problema foi a *Forward Checking Strategy*. Isso se deve ao fato de que sempre que uma variável  $X$  é atribuída, este processo estabelece uma consistência para ele: para cada variável  $Y$  não atribuída que está conectada a  $X$  pela restrição, delete do domínio de  $Y$  qualquer valor que seja inconsistente com o valor escolhido para  $X$ .

A despeito de outras heurísticas como o simples backtracking, o *Forward Backtracking* é melhor pois não entra em algumas das possibilidades que já estão alocadas, uma vez que previamente já elimina valores que viriam a falhar, assim reduzindo o número de nós abertos diferente das versões *Backtracking*, que utilizam busca exaustiva, o que aumenta consideravelmente a quantidade de nós visitados, mesmo quando aplicado junto a heurísticas como MRV, DEG etc.

## 3. Implementação

A seguir, apresentamos as classes presentes na implementação, dividindo-as em classes de representação e as classes responsáveis por definir as restrições.

### 3.1 Classes de representação

- A classe **Variable** define as variáveis utilizadas possui um nome e um numeral que foi colocado dentro da classe para permitir a presença de variáveis de mesmo nome;
- A classe **horaDia** é responsável por definir as características de cada bloco de hora que é utilizado no domínio. Define um **dia** e uma **hora**;
- A classe **BlocoDeEstudo** é responsável por definir as características de um bloco de estudo na tabela. Define um nome, um ref que é a aula a qual é possuído e o tempo que é o tamanho do bloco;
- A classe **Descanso** é responsável por definir as características de um qualquer bloco que tenha um dia e uma hora específica para acontecer;

- A classe **Resposta** foi criada unicamente para permitir o retorno das informações necessárias para o print na resolução

## 3.2 Restrições

- A restrição **isSeted** é responsável por setar uma atividade/aula em uma posição específica da tabela;
- A restrição **AtividadesComp** é responsável por definir as características de uma Atividade Complementar na tabela, fato que faz com que possa ser dado qualquer nome. Define nome, dia e quantidade de horas;
- A restrição **definidorDeDia** é responsável por indicar a qual dia uma determinada atividade complementar pertencerá;
- A restrição **TrabalhoOuEstagio** verifica se o discente está fazendo estágio ou trabalho ao mesmo tempo, o que não deve ocorrer;
- Semelhante a restrição acima, **pibicOUpibiti** verifica se o discente está fazendo Pibic ou Pibiti ao mesmo tempo;
- A restrição **SetDescanso** é responsável por alocar os blocos de descanso do discente;
- A restrição **sabadoTaOK** verifica se os os horários entre segunda e sexta são suficientes para encaixar as atividades e caso não esteja, libera a utilização do sábado para as atividades que não conseguiram ser alocadas nos dias úteis;
- A restrição **DiffNotEqualConstraint** é aplicado a cada variável e serve para impedir a existência de mais de uma variável ocupando um mesmo domínio

## 3.3 Classes da Implementação

- **MapSemana** é a classe responsável por montar o problema em si ela recebe as várias variáveis da entrada e monta o **cspdia** que depois é resolvido. É a classe que possui a maior quantidade de métodos e funções, pois a mesma tem que montar as variáveis e manter o controle sobre o tempo gasto na semana.
- **PrintTabela** é a classe responsável por fazer a amostragem da tabela de maneira toda montada com as cores presentes.

- **TesteTudo** recebe o *cspdia* e usa todas as funções de *solver* para resolver o problema testando os resultados
- As classes representadas pelo nome teste simplesmente possuem casos base que foram utilizados nas medições e podem ser visualizados na **parte 4**

### 3.4 Entradas do Código

As entradas para o correto funcionamento do código é a seguinte:

- **caso:** inteiro que serve para escolher qual caso predefinido pelo professor deve ser utilizado, varia entre 1, 2 e 3. Caso seja escolhido algum número fora desses, uma tabela sem aulas será instanciada.
- **horasPIBITI** e **horasPIBIC:** são arrays de inteiros que vão de 0 a 5 em tamanho onde cada posição representa um dia e a quantidade de horas a ser utilizado.
- **horariosEstagio** e **horariosTrabalho:** são arrays de inteiros que vão de 0 a 6 em tamanho onde cada posição representa um dia e a quantidade de horas a ser utilizado. O motivo que permite estes a ser de 0 a 6 é que os mesmos podem ocupar o sábado se necessário.
- **atividades:** ArrayList formado por instâncias da classe *AtividadesComp*
- **blocos:** ArrayList formado por instâncias da classe *BlocoDeEstudo*
- **descanso:** ArrayList formado por instâncias da classe *Descanso*

Exemplo:

```

1  System.out.println("Tabela da Semana");
2  System.out.println("=====Testes para o Exemplo1=====");
3  solver = new FlexibleBacktrackingSolver<Variable, String>().set(CspHeuristics.mrvDeg()).set(new ForwardCheckingStrategy<>());
4  int caso = 2;
5  int[] horariosPIBITI = {7,7,7,9,10};
6  int[] horariosPIBIC = {};
7  int[] horariosEstagio = {};
8  int[] horariosTrabalho = {0,8,0,6,1,4};
9  ArrayList<AtividadesComp> atividades = new ArrayList<>();
10
11  ArrayList<BlocoDeEstudo> blocos = new ArrayList<>();
12  blocos.add(new BlocoDeEstudo("COMP0409",4));
13  blocos.add(new BlocoDeEstudo("COMP0412",4));
14  blocos.add(new BlocoDeEstudo("COMP0408",4));
15  blocos.add(new BlocoDeEstudo("COMP0438",4));
16  blocos.add(new BlocoDeEstudo("COMP0461",4));
17
18  ArrayList<Descanso> descanso = new ArrayList<>();
19  descanso.add(new Descanso("Descanso",0,5));
20  descanso.add(new Descanso("Descanso",1,9));
21  descanso.add(new Descanso("Descanso",2,6));
22  descanso.add(new Descanso("Descanso",3,14));
23  descanso.add(new Descanso("Descanso",4,9));
24  CSP<Variable, String> cspdia = new MapSemana(caso, horariosPIBITI, horariosPIBIC, atividades, horariosEstagio, horariosTrabalho, blocos, descanso);
25
26  new TesteTudo(cspdia);

```

### 3.5 Plugins presentes

- **ANSI Escape in Console:** utilizado para fazer a coloração do print no console

### 3.6 Classe Pertencentes ao *aima* que tiveram que ser alterados

- **Assignment:** foi necessário fazer o *import* do *horaDia* a nossa classe de domínio e adicionar o método **RetornaOMapa** para conseguir pegar o resultado e por na tabela.
- **CSP:** foi necessário importar o *horaDia* e fazer um cast nas variáveis referentes ao Domínio.
- **Variables:** foi necessário fazer uma alteração no *equals* para permitir a presença de variáveis com o mesmo nome.
- **Domain:** foi necessário importar o *horaDia* e fazer um cast nas variáveis referentes ao Domínio.

## 4. Resultados obtidos

Para este problema, a equipe decidiu executar cinco diferentes testes para cada um dos três casos definidos no enunciado do problema. Cada um dos testes foi executado para cada uma das onze seguintes heurísticas:

- *Flexible Backtracking*
- *Flexible Backtracking MRV*
- *Flexible Backtracking DEG*
- *Flexible Backtracking LCV*
- *Forward Checking Strategy*
- *Flexible Backtracking FC + MRV & DEG*
- *Flexible Backtracking AC3*
- *Flexible Backtracking AC3 + MRV & DEG + LCV*
- *Back jumping Backtracking Solver*
- *Back jumping Backtracking Solver + MRV & DEG*

Por simplicidade, exibiremos o resultado das tabelas de horários para cada um dos três casos apenas para a heurística escolhida, que foi o **Forward Checking Strategy**. Nos anexos deste relatório, consta uma tabela com a média de nós analisados e tempo de execução de todos os testes para todas as heurísticas. Para os casos abaixo, a legenda em cores é representada da seguinte forma:

- [Azul escuro] - Aulas
- [Verde] - Pibic ou Pibiti
- [Vermelho] - Trabalho ou Estágio
- [Magenta] - Outras atividades complementares
- [Amarelo] - Blocos de estudo
- [Azul claro] - Descanso

- **Caso 1**

Se refere a implementação de teste presente no arquivo **testeCaso1Terceiro.java**, utilizando as seguintes entradas:

**Caso1(20 blocos de 30min)**

- **Pibic (40 blocos de 30 min):**

- Segunda: 8 ou 4h
- Terça: 8 ou 4h
- Quarta: 8 ou 4h
- Quinta: 8 ou 4h
- Sexta: 8 ou 4h

- **Estagio (10 blocos de 30 min):**

- Segunda: 2 ou 1h
- Terça: 2 ou 1h
- Quarta: 2 ou 1h
- Quinta: 2 ou 1h
- Sexta: 2 ou 1h

- **Palestra(6 blocos de 30 min):**

- Segunda: 3 ou 1h30min
- Quarta: 3 ou 1h30min

**Voluntaria(13 blocos de 6h30 min):**

- Terça: 3 ou 1h30min
- Sábado: 10 ou 5h

- **Descanso (8 blocos de 30 min):**

- Segunda: 1 ou 30min
- Terça: 2 ou 1h
- Quinta: 2 ou 1h
- Sexta: 1 ou 30min
- Sabado: 1 ou 30min

- **Almoço(2 blocos de 30 min):**

- Segunda: 1 ou 30min
- Terça: 1 ou 30min



HORARIO	[	SEGUNDA	]	[	TERÇA	]	[	QUARTA	]	[	QUINTA	]	[	SEXTA	]	[	SÁBADO	]
Horário 01	[	PIBIC-----	]	[	PIBIC-----	]	[	PIBIC-----	]	[	PIBIC-----	]	[	PIBIC-----	]	[	Voluntaria-	]
Horário 02	[	Descanso---	]	[	PIBIC-----	]	[	PIBIC-----	]	[	PIBIC-----	]	[	PIBIC-----	]	[	Voluntaria-	]
Horário 03	[	PIBIC-----	]	[	PIBIC-----	]	[	PIBIC-----	]	[	PIBIC-----	]	[	PIBIC-----	]	[	Voluntaria-	]
Horário 04	[	PIBIC-----	]	[	PIBIC-----	]	[	PIBIC-----	]	[	PIBIC-----	]	[	PIBIC-----	]	[	Voluntaria-	]
Horário 05	[	PIBIC-----	]	[	COMP0455---	]	[	PIBIC-----	]	[	COMP0455---	]	[	PIBIC-----	]	[	Voluntaria-	]
Horário 06	[	PIBIC-----	]	[	COMP0455---	]	[	PIBIC-----	]	[	COMP0455---	]	[	PIBIC-----	]	[	Voluntaria-	]
Horário 07	[	PIBIC-----	]	[	COMP0455---	]	[	PIBIC-----	]	[	COMP0455---	]	[	PIBIC-----	]	[	Voluntaria-	]
Horário 08	[	PIBIC-----	]	[	COMP0455---	]	[	PIBIC-----	]	[	COMP0455---	]	[	PIBIC-----	]	[	Voluntaria-	]
Horário 09	[	Almoco-----	]	[	PIBIC-----	]	[	Estagio----	]	[	PIBIC-----	]	[	Estagio----	]	[	Voluntaria-	]
Horário 10	[	Descanso---	]	[	Descanso---	]	[	Estagio----	]	[	Descanso---	]	[	Descanso---	]	[	Descanso---	]
Horário 11	[	PIBIC-----	]	[	Descanso---	]	[	Palestra---	]	[	Descanso---	]	[	Estagio----	]	[	Voluntaria-	]
Horário 12	[	Estagio----	]	[	Almoco-----	]	[	Palestra---	]	[	PIBIC-----	]	[	EstCOMP0455	]	[	-----	]
Horário 13	[	Estagio----	]	[	PIBIC-----	]	[	Palestra---	]	[	COMP0481---	]	[	EstCOMP0481	]	[	-----	]
Horário 14	[	Palestra---	]	[	PIBIC-----	]	[	EstCOMP0408	]	[	COMP0481---	]	[	-----	]	[	-----	]
Horário 15	[	Palestra---	]	[	PIBIC-----	]	[	EstCOMP0408	]	[	COMP0481---	]	[	-----	]	[	-----	]
Horário 16	[	Palestra---	]	[	Estagio----	]	[	EstCOMP0455	]	[	COMP0481---	]	[	-----	]	[	-----	]
Horário 17	[	COMP0408---	]	[	Estagio----	]	[	COMP0408---	]	[	PIBIC-----	]	[	-----	]	[	-----	]
Horário 18	[	COMP0408---	]	[	Voluntaria-	]	[	COMP0408---	]	[	PIBIC-----	]	[	-----	]	[	-----	]
Horário 19	[	COMP0408---	]	[	Voluntaria-	]	[	COMP0408---	]	[	Estagio----	]	[	-----	]	[	-----	]
Horário 20	[	COMP0408---	]	[	Voluntaria-	]	[	COMP0408---	]	[	Estagio----	]	[	-----	]	[	-----	]

## • Caso 2

Se refere a implementação de teste presente no arquivo **testeCaso24.java**, utilizando as seguintes entradas:

- **Pibic (40 blocos de 30 min):**
  - Segunda: 3 blocos (1h30)
  - Terça: 8 blocos (4h)
  - Quarta: 8 blocos (4h)
  - Quinta: 9 blocos (4h30)
  - Sexta: 12 blocos (6h)
- **Trabalho (11 blocos de 30 min):**
  - Quinta: 3 blocos (1h30)
  - Sábado: 8 blocos (4h)
- **Aulas:**
  - Quantidade informada inicialmente no Caso 2;
- **Blocos de estudo (20 blocos de 30min na semana):**
  - 4 blocos para cada disciplina existente no horário
- **Descanso:**
  - Segunda: 1 bloco no 5º horário, 1 bloco no 6º horário, 1 bloco no 7º horário, 1 bloco no 12º horário e 1 bloco no 13º horário

HORARIO	[	SEGUNDA	]	[	TERÇA	]	[	QUARTA	]	[	QUINTA	]	[	SEXTA	]	[	SÁBADO	]
Horário 01	[	COMP0409---	]	[	PIBIC-----	]	[	COMP0409---	]	[	PIBIC-----	]	[	COMP0438---	]	[	Trabalho---	]
Horário 02	[	COMP0409---	]	[	PIBIC-----	]	[	COMP0409---	]	[	PIBIC-----	]	[	COMP0438---	]	[	Trabalho---	]
Horário 03	[	COMP0409---	]	[	PIBIC-----	]	[	COMP0409---	]	[	PIBIC-----	]	[	COMP0438---	]	[	Trabalho---	]
Horário 04	[	COMP0409---	]	[	PIBIC-----	]	[	COMP0409---	]	[	PIBIC-----	]	[	COMP0438---	]	[	Trabalho---	]
Horário 05	[	PIBIC-----	]	[	COMP0412---	]	[	PIBIC-----	]	[	COMP0412---	]	[	COMP0438---	]	[	Trabalho---	]
Horário 06	[	Descanso---	]	[	COMP0412---	]	[	PIBIC-----	]	[	COMP0412---	]	[	COMP0438---	]	[	Trabalho---	]
Horário 07	[	Descanso---	]	[	COMP0412---	]	[	PIBIC-----	]	[	COMP0412---	]	[	COMP0438---	]	[	Trabalho---	]
Horário 08	[	Descanso---	]	[	COMP0412---	]	[	PIBIC-----	]	[	COMP0412---	]	[	COMP0438---	]	[	Trabalho---	]
Horário 09	[	COMP0408---	]	[	PIBIC-----	]	[	COMP0408---	]	[	PIBIC-----	]	[	PIBIC-----	]	[	EstCOMP0438	]
Horário 10	[	COMP0408---	]	[	PIBIC-----	]	[	COMP0408---	]	[	PIBIC-----	]	[	PIBIC-----	]	[	EstCOMP0438	]
Horário 11	[	COMP0408---	]	[	PIBIC-----	]	[	COMP0408---	]	[	PIBIC-----	]	[	PIBIC-----	]	[	EstCOMP0438	]
Horário 12	[	COMP0408---	]	[	PIBIC-----	]	[	COMP0408---	]	[	PIBIC-----	]	[	PIBIC-----	]	[	EstCOMP0438	]
Horário 13	[	Descanso---	]	[	EstCOMP0409	]	[	PIBIC-----	]	[	PIBIC-----	]	[	PIBIC-----	]	[	EstCOMP0461	]
Horário 14	[	Descanso---	]	[	EstCOMP0409	]	[	PIBIC-----	]	[	Trabalho---	]	[	PIBIC-----	]	[	EstCOMP0461	]
Horário 15	[	COMP0461---	]	[	EstCOMP0409	]	[	COMP0461---	]	[	Trabalho---	]	[	PIBIC-----	]	[	EstCOMP0461	]
Horário 16	[	COMP0461---	]	[	EstCOMP0409	]	[	COMP0461---	]	[	Trabalho---	]	[	PIBIC-----	]	[	EstCOMP0461	]
Horário 17	[	COMP0461---	]	[	EstCOMP0412	]	[	COMP0461---	]	[	EstCOMP0408	]	[	PIBIC-----	]	[	-----	]
Horário 18	[	COMP0461---	]	[	EstCOMP0412	]	[	COMP0461---	]	[	EstCOMP0408	]	[	PIBIC-----	]	[	-----	]
Horário 19	[	PIBIC-----	]	[	EstCOMP0412	]	[	PIBIC-----	]	[	EstCOMP0408	]	[	PIBIC-----	]	[	-----	]
Horário 20	[	PIBIC-----	]	[	EstCOMP0412	]	[	PIBIC-----	]	[	EstCOMP0408	]	[	PIBIC-----	]	[	-----	]

### • Caso 3

Se refere a implementação de teste presente no arquivo **testeCaso33.java** utilizando as seguintes entradas:

- **Sem blocos de PIBIC ou PIBITI**
- **Estágio (36 blocos de 30 min):**
  - Segunda: 6 blocos (3h)
  - Terça: 7 blocos (3h30)
  - Quarta: 7 blocos (3h30)
  - Quinta: 6 blocos (3h)
  - Sexta: 3 blocos (1h30)
  - Sábado: 7 blocos (3h30)
- **Blocos de estudo (12 blocos de 30 min na semana):**
  - Terça: 8 blocos (4h)
  - Quinta: 4 blocos (2h)
  - Sábado: 8 blocos (4h)
- **Aulas:**
  - Quantidade informada inicialmente no caso 3.
- **Descanso (5 blocos de 30 min na semana):**
  - 1 bloco de 30 minutos em cada dia de segunda à sexta.

HORARIO	[	SEGUNDA	][	TERÇA	][	QUARTA	][	QUINTA	][	SEXTA	][	SÁBADO	]
Horário 01	[	ELET0043----	][	MAT0096-----	][	MAT0154-----	][	MAT0096-----	][	MAT0154-----	][	Estagio-----	]
Horário 02	[	ELET0043----	][	MAT0096-----	][	MAT0154-----	][	MAT0096-----	][	MAT0154-----	][	Estagio-----	]
Horário 03	[	ELET0043----	][	MAT0096-----	][	MAT0154-----	][	MAT0096-----	][	MAT0154-----	][	Estagio-----	]
Horário 04	[	ELET0043----	][	MAT0096-----	][	MAT0154-----	][	MAT0096-----	][	MAT0154-----	][	Estagio-----	]
Horário 05	[	ESTAT0011---	][	COMP0409----	][	ESTAT0011---	][	COMP0409----	][	Estagio-----	][	Estagio-----	]
Horário 06	[	ESTAT0011---	][	COMP0409----	][	ESTAT0011---	][	COMP0409----	][	Estagio-----	][	Estagio-----	]
Horário 07	[	ESTAT0011---	][	COMP0409----	][	ESTAT0011---	][	COMP0409----	][	Estagio-----	][	Estagio-----	]
Horário 08	[	ESTAT0011---	][	COMP0409----	][	ESTAT0011---	][	COMP0409----	][	EstCOMP0415-	][	EstMAT0096--	]
Horário 09	[	COMP0415----	][	COMP0412----	][	COMP0415----	][	COMP0412----	][	COMP0417----	][	EstMAT0096--	]
Horário 10	[	COMP0415----	][	COMP0412----	][	COMP0415----	][	COMP0412----	][	COMP0417----	][	EstMAT0096--	]
Horário 11	[	COMP0415----	][	COMP0412----	][	COMP0415----	][	COMP0412----	][	COMP0417----	][	-----	]
Horário 12	[	COMP0415----	][	COMP0412----	][	COMP0415----	][	COMP0412----	][	COMP0417----	][	-----	]
Horário 13	[	Estagio-----	][	Estagio-----	][	Estagio-----	][	Estagio-----	][	EstCOMP0415-	][	-----	]
Horário 14	[	Estagio-----	][	Estagio-----	][	Estagio-----	][	Estagio-----	][	EstCOMP0417-	][	-----	]
Horário 15	[	Descanso----	][	Estagio-----	][	Estagio-----	][	Estagio-----	][	EstELET0043-	][	-----	]
Horário 16	[	Estagio-----	][	Estagio-----	][	Estagio-----	][	Estagio-----	][	Descanso----	][	-----	]
Horário 17	[	Estagio-----	][	Estagio-----	][	Descanso----	][	Estagio-----	][	EstELET0043-	][	-----	]
Horário 18	[	Estagio-----	][	Estagio-----	][	Estagio-----	][	Descanso----	][	EstESTAT0011	][	-----	]
Horário 19	[	Estagio-----	][	Descanso----	][	Estagio-----	][	Estagio-----	][	EstMAT0154--	][	-----	]
Horário 20	[	EstCOMP0412-	][	Estagio-----	][	Estagio-----	][	EstCOMP0409-	][	-----	][	-----	]

**OBS:** O *minConflictSolver* não é apresentado nos resultados porque não estava se mostrando resultados de maneira suficiente para podermos mostrar aqui. Já o *TreeCsp* não foi utilizado pois a entrada deve estar no formato de árvore e como iniciamos utilizando uma versão modificada do *MapColoring*, deixamos de lado.

## 5.Comentando o código

Essa parte do relatório se destina única e exclusivamente a comentar sobre erros ou decisões de projeto um pouco duvidosas durante o desenvolvimento. É importante primeiro comentar sobre o que foi o maior tempo perdido no desenvolvimento do trabalho e que foi o não entendimento de que as variáveis presentes no *aima* não precisavam seguir os padrões de String e poderia instanciar classes próprias.

Outro ponto e esse acabou sendo um erro que não foi realmente corrigido é o fato de que tivemos que utilizar um pré-processamento no controle de horas de cada dia. Foi formulada uma solução para isso, mas devido ao prazo de entrega que, mesmo estendido, ainda foi apertado para entregar o trabalho, não havendo tempo suficiente para tentar utilizá-lo.

## 6. Conclusão

Este trabalho tem por objetivo demonstrar e discutir a aplicação de satisfação de restrição em problemas de alocação de aulas, atividades complementares e

blocos de estudo de um discente de universidade. Este tipo de problema leva em conta um conjunto de objetos cujo estado dos mesmos deve satisfazer uma série de restrições. Inicialmente, a solução proposta neste trabalho tinha o propósito de resolver três casos iniciais de combinações de variáveis em determinados blocos de horários. Para cada um dos três casos, projetamos cinco diferentes entradas para testar através da execução do problema para dez tipos de heurísticas diferentes. Entre estas, foi escolhida a Forward Checking como aquela responsável por resolver o problema de maneira mais eficiente, levando em consideração número de nós visitados e tempo de execução, além disso, é projetada para que, para cada variável Y não atribuída que está conectada a X pela restrição, delete do domínio de Y qualquer valor que seja inconsistente com o valor escolhido para X. Isso diminui consideravelmente a quantidade de domínios possíveis para cada variável, em relação a outras heurísticas.

## **Referência Bibliográfica**

Russell, Stuart J., Peter Norvig, and Ernest Davis. *Artificial Intelligence: A Modern Approach*. 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2010.

**Tabela 01 - Médias de nós analisados e tempo de execução para cada heurística**

<b>Heurística</b>	<b>Médias</b>	
	<b>Nós analisados</b>	<b>Tempo de execução (nanossegundos)</b>
<b>Flexible Backtracking</b>	4702,73	18261073,47
<b>Flexible Backtracking MRV</b>	4702,73	15583293,2
<b>Flexible Backtracking DEG</b>	4702,73	2082354453
<b>Flexible Backtracking LCV</b>	4702,73	33410772573
<b>Forward Checking Strategy</b>	2461,2	181995873,3
<b>Flexible Backtracking FC + MRV &amp; DEG</b>	2461,2	179500540,1
<b>Flexible Backtracking AC3</b>	2461,2	15843310633
<b>Flexible Backtracking AC3 + MRV &amp; DEG + LCV</b>	2461,2	28061437900
<b>Back jumping Backtracking Solver</b>	5461,2	18482053,33
<b>Back jumping Backtracking Solver + MRV &amp; DEG</b>	5461,2	17416513,2