SOLUÇÃO DO DESAFIO UNISOMA 2019 - Casa da Criança Paralítica

Gabriel Barros Dávila

Guilherme Lima Correa

Vinnicio Resende Bastos

---

1. Introdução

Os problemas de programação de horários são, na verdade, problemas de alocação. Estes constituem em umas das mais importantes atividades administrativas em uma organização e acabam se tornando complicados por mobilizarem uma quantidade significativa de pessoas, tempo, e esforço. Pode-se ver esse tipo de problema em diversos contextos, como por exemplo, na criação de tabelas de horários para jogos, na organização de escalas de horário de trabalho ou até mesmo na construção de grades horárias dos cursos de uma universidade. Uma característica marcante dos problemas de alocação é a sua natureza fortemente associativa, ou seja, soluções para tais problemas são obtidas a partir de associações entre um número de eventos (ex.: aulas, palestras, jogos), e um número limitado de recursos (ex.: tempo, pessoas, salas). Além disso, quaisquer soluções factíveis para o problema devem respeitar a alta quantidades de restrições, e devem atender, da melhor maneira possível, a certas demandas (de professores, institucionais, de redução de custos, entre outras).

O Desafio da Unisoma 2019 consiste em um problema de alocação de crianças em horários de atendimentos disponíveis pela equipe médica da ONG (Casa da Criança Paralítica). Esta Organização não governamental realiza aproximadamente 50.000 atendimentos médicos por ano. Diante dessa quantidade, o planejamento dos atendimentos acaba sendo uma tarefa árdua feita semanalmente pelos funcionários da ONG. Sendo assim, o Desafio 2019 busca solucionar, ou melhor, facilitar essas designações semanais. Portanto, o problema é alocar as crianças maximizando o número de indivíduos atendidos, isto respeitando tanto a disponibilidade de comparecimento da criança a instituição como a disponibilidade do médico, além das demais restrições impostas pela Casa da Criança Paralítica.

2. Revisão literária

O problema estudado neste desafio trata da alocação de crianças em horários de atendimento pela equipe técnica da ONG, e por isso, entende-se que o desafio é uma variação dos Problemas de Programação de Horários (PPH). Portanto, nesta seção serão abordados alguns trabalhos realizados na área de programação de horários em escolas os quais nortearam a modelagem do desafio. Vale ressaltar, que este tipo de modelagem não foi o único analisado pelo grupo para resolver o problema.

Os problemas de alocação de horários, apresentam algumas restrições padrões que devem aparecer em suas modelagens matemáticas. Estas restrições asseguram que não haja conflito de horários, evitando que uma pessoa esteja em mais de um lugar ao mesmo tempo. Ou seja, essas restrições básicas evitam que um professor dê aula para mais de uma turma em um horário e que uma turma receba aula de mais de um professor em um horário. Uma formulação que contém estas restrições pode ser vista em Carvalho (2011), e Martins (2010). Note-se que essas restrições também existem no desafio, visto que, um mesmo médico não pode atender mais de uma criança em um horário e uma criança não pode receber mais de um atendimento em um mesmo horário.

Grobner, Wilke e Buttcher (2003) citado por MONLEVADE (2017) fazem uma generalização das propriedades em comum dos problemas de horários, demonstrando que as entidades são diferenciadas entre recursos e eventos. Os eventos são atribuídos certo número de intervalos de tempos, cada um tendo início e fim, e o conjunto desses eventos podem ser definidos como um cronograma. Já os recursos são normalmente algo único e limitado, como o caso de mão de obra e máquinas. Além disso, existem as restrições que servem para limitar a programação de horários, não deixando que ocorram dois eventos de forma simultânea, por exemplo, e restringindo a relação dos recursos com os eventos. Diante desta generalização dos problemas de horários, nota-se que o Desafio da Unisoma possui diversas semelhanças. Pode-se associar eventos aos atendimentos, enquanto os recursos seriam os próprios especialistas da ONG. Ademais, os atendimentos devem respeitar as disponibilidades tanto das crianças como a dos funcionários, limitando assim a relação de recursos e eventos.

Para finalizar a pesquisa, notamos que o desafio acaba se tornando também uma variação do problema de designação com variáveis binárias. Segundo Arenales et al. (2007) o problema de designação envolve n tarefas e n agentes, sendo cada tarefa executada por um único agente e cada agente executa uma única tarefa, onde a execução de tarefa j pelo agente i tem um custo Cij. O problema consiste em designar tarefas de modo a minimizar o custo total. O Desafio da Unisoma envolve uma quantidade de crianças e uma quantidade de funcionários, que podem ser vistos como tarefas e agentes. A divergência em relação ao problema de designação consiste no índice horário, e ao invés de minimizar custos, o nosso objetivo é maximizar o número de crianças (tarefas) atendidas.

3. Modelagem Matemática

Índices:

$c = 1,..., C$	Representam todas as crianças cadastradas.
$t = 1,..., T$	Representam as especialidades que estão disponíveis na ONG.
f = 1, …, F	Representam os funcionários independente da especialidade.
$h = 1,..., H$	Representam os horários possíveis na semana discretizados como mostra a Figura 1 abaixo, considerando desde as 7:30 até as 17:00 de Segunda a Sexta.

Figura 1: Tempos de atendimento *(h)* discretizados.

	Segunda	Terça	Quarta	Quinta	Sexta
07:30:00	1	20	39	58	77
08:00:00	2	21	40	59	78
08:30:00	3	22	41	60	79
09:00:00	4	23	42	61	80
09:30:00	5	24	43	62	81
10:00:00	6	25	44	63	82
10:30:00	7	26	45	64	83
11:00:00	8	27	46	65	84
11:30:00	9	28	47	66	85
12:00:00	10	29	48	67	86
12:30:00	11	30	49	68	87
13:00:00	12	31	50	69	88
13:30:00	13	32	51	70	89
14:00:00	14	33	52	71	90
14:30:00	15	34	53	72	91
15:00:00	16	35	54	73	92
15:30:00	17	36	55	74	93
16:00:00	18	37	56	75	94
16:30:00	19	38	57	76	95
17:00:00	0	0	0	0	0

Os conjuntos de especialidades *(t)* e horários *(h)* são específicos para cada criança pois dependem da sua disponibilidade de horário individual e da demanda individual de cada especialidade. Além disso, o conjunto de funcionários *(f)* também varia com a especialidade, sendo que cada especialidade tem um conjunto de funcionários.

Dados:

Rc,t Indica a quantidade de atendimentos que a criança *(c)* recebe do tratamento *(t)*.

M Indica um número convenientemente grande, maior ou igual a soma de tratamentos que um período (manhã ou tarde) pode assumir.

Pc,t,f,h Vetor com os coeficientes da variável de decisão principal *(Xc,t,f,h)* .

Variáveis de decisão:

A solução do modelo deve resultar nos valores do seguinte conjunto de variáveis:

X c,t,f,h Variável binária que se igual a 1, indica que a criança (*c)* recebe o tratamento (*t)* pelo funcionário (*f)* no horário (*h)*. Caso contrário se iguala a 0.
Esse conjunto de variáveis abrangem apenas as necessidades das crianças em relação às especialidades e sua disponibilidade de horário, por exemplo, uma criança que não precise de Neurologia não será criado a variável *X c,n,f,h* sendo *n* = Neurologia.

Há, ainda, restrições que demandam a utilização de variáveis auxiliares. No caso das restrições de período, temos os seguintes grupos de variáveis:

Sc Variável binária que para cada criança (c), na Segunda-Feira assume valor 1 se a criança é atendida no período da tarde e 0 caso contrário.

Tc Variável binária que para cada criança (c), na Terça-Feira assume valor 1 se a criança é atendida no período da tarde e 0 caso contrário.

QAc Variável binária que para cada criança (c), na Quarta-Feira assume valor 1 se a criança é atendida no período da tarde e 0 caso contrário.

QIc Variável binária que para cada criança (c), na Quinta-Feira assume valor 1 se a criança é atendida no período da tarde e 0 caso contrário.

SEc Variável binária que para cada criança (c), na Sexta-Feira assume valor 1 se a criança é atendida no período da tarde e 0 caso contrário.

Já para a restrição que garante o período mínimo de 1 dia entre atendimentos da mesma especialidade para mesma criança, usamos as seguintes variáveis auxiliares:

DSTc,t Variável binária que para cada criança (c) e especialidade (t) que para Segunda-Feira e Terça-Feira assume valor 1 se é atendida na Terça-Feira e 0 caso contrário.

DTQAc,t Variável binária que para cada criança (c) e especialidade (t) que para Terça-Feira e Quarta-Feira assume valor 1 se é atendida na Quarta-Feira e 0 caso contrário.

DQAIc,t Variável binária que para cada criança (c) e especialidade (t) que para Quarta-Feira e Quinta-Feira assume valor 1 se é atendida na Quinta-Feira e 0 caso contrário.

DQISc,t Variável binária que para cada criança (c) e especialidade (t) que para Quinta-Feira e Sexta-Feira assume valor 1 se é atendida na Sexta-Feira e 0 caso contrário.

Vale ressaltar que as variáveis *Sc, Tc, QAc, QIc* e *SEc* são específicas para cada criança, por exemplo o conjunto Sc, tem o mesmo comprimento que o conjunto de crianças e o mesmo vale para as outras variáveis. E as variáveis *DSTc,t, DTQAc,t, DQAIc,t* e *DQISc,t* variam para criança e especialidade de tratamento, ou seja, *DTSc,t* é uma matriz dimensões equivalentes ao comprimento do vetor de crianças e do vetor de tratamentos.

Como objetivo de garantir o balanceamento da carga de trabalho de funcionários de uma mesma especialidade, foi preciso criar as seguintes variáveis.

CT *f* Variável inteira que armazena a carga de trabalho, ou seja, a quantidade total de atendimentos na semana alocados ao funcionário *f.*

CTmax *t* Variável inteira que armazena o maior valor das variáveis CT*f* dos funcionários de um mesmo tratamento *t*.

CTmin *t* Variável inteira que armazena o menor valor das variáveis CT*f* dos funcionários de um mesmo tratamento *t*.

Formulação matemática do problema:

$$Maximizar \sum_{c=1}^{C}\sum_{t=1}^{T}\sum_{f=1}^{F}\sum_{h=1}^{H} Pc,t,f,h \times Xc,t,f,h \quad + \quad \sum_{t=1}^{T}(CTmin\,t \; - \; CTmax\,t\,)$$

(1)

$$\sum_{c=1}^{C} Xc,t,f,h \;\leq\; 1 \qquad h = 1, \ldots, H \qquad t = 1, \ldots, T \qquad f = 1, \ldots, F \tag{2}$$

$$\sum_{t=1}^{T}\sum_{f=1}^{F} Xc,t,f,h \;\leq\; 1 \qquad c = 1, \ldots, C \qquad h = 1, \ldots, H \tag{3}$$

$$\sum_{h=1}^{H}\sum_{f=1}^{F} Xc,t,f,h \;\leq\; Rc,t \qquad c = 1, \ldots, C \qquad t = 1, \ldots, T \tag{4}$$

$$\sum_{t=1}^{T}\sum_{f=1}^{F}\sum_{h=1}^{9} Xc,t,f,h \;\leq\; M \times (1 - Sc) \qquad c = 1, \ldots, C$$

(5.1.1)

$$\sum_{t=1}^{T}\sum_{f=1}^{F}\sum_{h=10}^{19} Xc,t,f,h \;\leq\; M \times Sc \qquad c = 1, \ldots, C$$

(5.1.2)

$$\sum_{t=1}^{T}\sum_{f=1}^{F}\sum_{h=20}^{28} Xc,t,f,h \;\leq\; M \times (1 - Tc) \qquad c = 1, \ldots, C$$

(5.2.1)

$$\sum_{t=1}^{T}\sum_{f=1}^{F}\sum_{h=29}^{38} Xc,t,f,h \;\leq\; M \times Tc \qquad c = 1, \ldots, C \tag{5.2.2}$$

$$\sum_{t=1}^{T}\sum_{f=1}^{F}\sum_{h=39}^{47} Xc,t,f,h \;\leq\; M \times (1 - QAc) \qquad c = 1, \ldots, C \tag{5.3.1}$$

$$\sum_{t=1}^{T}\sum_{f=1}^{F}\sum_{h=48}^{57} Xc,t,f,h \;\leq\; M \times QAc \qquad c = 1, \ldots, C \tag{5.3.2}$$

$$\sum_{t=1}^{T}\sum_{f=1}^{F}\sum_{h=58}^{66} Xc,t,f,h \;\leq\; M \times (1 - QIc) \qquad c = 1, \ldots, C \tag{5.4.1}$$

$$\sum_{t=1}^{T}\sum_{f=1}^{F}\sum_{h=67}^{76} Xc,t,f,h \;\leq\; M \times QIc \qquad c = 1, \ldots, C \tag{5.4.2}$$

$$\sum_{t=1}^{T}\sum_{f=1}^{F}\sum_{h=77}^{85} Xc,t,f,h \;\leq\; M \times (1 - SEc) \qquad c = 1, \ldots, C \tag{5.6.1}$$

$$\sum_{t=1}^{T}\sum_{f=1}^{F}\sum_{h=86}^{95} Xc,t,f,h \ \leq \ M \times SEc \qquad c = 1, \ldots, C \qquad (5.6.2)$$

$$\sum_{f=1}^{F}\sum_{h=1}^{19} X,c,t,f,h \ \leq \ (1 - DSTc,t) \qquad c = 1, \ldots, C \qquad t = 1, \ldots, T \qquad (6.1.1)$$

$$\sum_{f=1}^{F}\sum_{h=20}^{38} X,c,t,f,h \ \leq \ DSTc,t \qquad c = 1, \ldots, C \qquad t = 1, \ldots, T$$

(6.1.2)

$$\sum_{f=1}^{F}\sum_{h=20}^{38} X,c,t,f,h \ \leq \ (1 - DTQAc,t) \qquad c = 1, \ldots, C \qquad t = 1, \ldots, T$$

(6.2.1)

$$\sum_{f=1}^{F}\sum_{h=39}^{57} X,c,t,f,h \ \leq \ DTQAc,t \qquad c = 1, \ldots, C \qquad t = 1, \ldots, T \qquad (6.2.2)$$

$$\sum_{f=1}^{F}\sum_{h=39}^{57} X,c,t,f,h \ \leq \ (1 - DQAIc,t) \qquad c = 1, \ldots, C \qquad t = 1, \ldots, T$$

(6.3.1)

$$\sum_{f=1}^{F}\sum_{h=58}^{76} X,c,t,f,h \ \leq \ DQAIc,t \qquad c = 1, \ldots, C \qquad t = 1, \ldots, T \qquad (6.3.2)$$

$$\sum_{f=1}^{F}\sum_{h=58}^{76} X,c,t,f,h \ \leq \ (1 - DQISc,t) \qquad c = 1, \ldots, C \qquad t = 1, \ldots, T$$

(6.4.1)

$$\sum_{f=1}^{F}\sum_{h=77}^{95} X,c,t,f,h \ \leq \ DQISc,t \qquad c = 1, \ldots, C \qquad t = 1, \ldots, T \qquad (6.4.2)$$

$$\sum_{c=1}^{C}\sum_{h=1}^{H}\sum_{t=1}^{T} Xc,t,f,h \ = \ CTf \qquad f = 1, \ldots, F \qquad (7.1)$$

$$CTf \ \leq \ CTmax\,t \qquad t = 1, \ldots, T \qquad f = 1, \ldots, F \qquad (7.2)$$

$$CTmin\,t \ \leq \ CTf \qquad t = 1, \ldots, T \qquad f = 1, \ldots, F \qquad (7.2)$$

A primeira parte da função objetivo (1) consiste na soma de todas as variáveis $X_{c,t,f,h}$ multiplicado pelo seu coeficiente do vetor $P_{c,t,f,h}$, o objetivo desta parcela é maximizar os atendimentos na semana. Já a segunda parte da função objetivo, consiste em minimizar a diferença entre o funcionário com maior carga horário e o funcionário mais ocioso, visando não sobrecarregar nenhum funcionário através do balanceamento de atendimentos. A primeira restrição (2) obriga que 1 funcionário de uma determinada especialidade só possa atender uma criança em um horário específico. A segunda restrição (3) força 1 criança a ser atendida por um único funcionário de uma especialidade num único horário.

A terceira restrição (4) limita a quantidade de atendimentos que uma criança específica pode receber de uma determinada especialidade no valor $R_{c,t}$ que é número de atendimentos que cada criança demanda da especialidade *(t)*.

A quarta restrição (5) é, na verdade, uma família de restrições que garantem que uma criança só será atendida no período da manhã ou no período da tarde de um mesmo dia, para isso usamos as variáveis binárias S_c, T_c, QA_c, QI_c e SE_c como eventos excludentes dentro de um mesmo dia, por exemplo: Nas restrições 5.1.1 e 5.1.2 que se referem a Segunda-Feira, caso a criança seja atendida no período da manhã S_c se tornará 0 obrigando que na restrição 5.1.2 não exista atendimentos na parte da tarde. Os somatórios de 1 a 9 representam o período da manhã da Segunda (seguindo a Tabela 1 de discretização) e de 10 a 19 o período da tarde.

A quinta restrição (6), que também é uma família que foi destrinchada, mas dessa vez não pelos períodos manhã e tarde, mas sim os dias da semana garantem que consultas de uma mesma especialidade para uma única criança tenham um intervalo de pelo menos um dia. Para isso, usamos o mesmo artifício que o conjunto de restrições anterior, utilizando as variáveis $DST_{c,t}$, $DTQA_{c,t}$ $DQAI_{c,t}$ e $DQIS_{c,t}$ causando eventos excludentes numa dupla de dias.

O sexto grupo de restrições (7) assegura que as variáveis $CTmax_t$ e $CTmin_t$ assumam o maior e o menor valor, respectivamente, dentre as cargas de trabalho de todos os funcionários de uma mesma especialidade. Este par de variáveis é utilizado na função objetivo como forma de diminuir a diferença entre o funcionário que atende mais crianças e o funcionário com menor carga de trabalho.

Além das restrições da modelagem acima, algumas observações devem ser ressaltadas:

O1 Dar prioridade para crianças que já foram atendidas na semana anterior,

O2 Regime quinzenal dos atendimentos da especialidade de nutrição, A Observação O1 foi feita com lógica de programação no código, que serão esclarecidas no descritivo da implementação do mesmo. Além disso o vetor *P* também contribui para resolução desse entrave uma vez que atribui um peso maior para as variáveis que assumiram valor igual a 1 no último agendamento, favorecendo que crianças regulares sejam agendadas a não ser que essa ação infactibilize o problema. A O2 também foi resolvida com um artifício de programação e utiliza também o vetor P, mas ao invés de atribuir pesos maiores para as variáveis associadas a crianças regulares ele atribui para as variáveis que não foram atendidas em nutrição na semana anterior e possuem uma frequência de atendimento quinzenal, para isso, como premissa consideramos que a primeira solução deve alocar todas as nutrições possíveis, como se nenhum atendimento dessa especialidade houvesse sido efetuado na semana antecedente.

4. Implementação

Linguagens de programação

Para o desenvolvimento da aplicação foram utilizadas 3 linguagens de programação distintas, a fim de maximizar as potencialidades de cada uma. São elas VBA, Python e R, elas foram usadas, respectivamente, para limitação de entradas de dados no excel, construção do *Back-End*, construção do *Front-End*.

Como dito, a linguagem VBA foi utilizada para controlar a entrada manual de dados como cadastramentos de crianças e funcionários e seus horários. Usamos tanto as ferramentas disponíveis no Excel quanto novas macros para que a entrada equivocada de dados não atrapalhasse nossa solução. Além disso, buscamos fazer o mínimo de alteração possível no modelo de planilha disponibilizado para não atrapalhar os processos que já existem.

Como especificado pela organização do desafio, a biblioteca *Shiny* do R será utilizada para construção da interface com o usuário final. Este pacote possibilita a criação de aplicações altamente interativas e robustas, sem exigir habilidades avançadas em desenvolvimento web. Outra biblioteca do R fundamental à solução é a *reticulate*, utilizada na conexão da interface frontal ao motor de otimização implementado em Python, a medida que permite a integração de scripyts das duas linguagens.

Algumas características importantes do Python influenciaram a escolha desta linguagem para compor o Back-End da aplicação. A primeira delas é a facilidade de manipulação ágil de extensas quantidades de dados. Através da biblioteca *pandas* é possível importar planilhas de arquivos Excel (.xlsx, .xls) para estruturas chamadas *Data Frames*, um tipo de objeto desta biblioteca. Os *Data Frames* são estruturas de dados tabulares, capazes de receber operações básicas de busca, seleção, adição, renomeação e exclusão, o que facilita o tratamento dos dados.

Uma segunda vantagem do Python é a extensa quantidade de bibliotecas destinadas à modelagem matemática, solvers e outros importantes etapas na criação de modelos de otimização. No caso desta aplicação, será utilizado um software open source desenvolvido pela Google denominado *OR – Tools,* destinado à otimização combinatória. O OR – Tools dispõem da capacidade de resolução de recorrentes problemas da pesquisa operacional, como o roteamento de veículos, fluxo em rede, scheduling, assim como modelos de programação linear, inteira ou mista. O grande diferencial desta plataforma é a variedade de solvers que podem ser utilizados na resolução dos problemas. A biblioteca permite tanto a execução de pacotes comerciais como Gurobi e CPLEX, quanto a de open sources como GLPK, GLOP e CBC.

O modelo utilizado na solução do desafio pode ser caracterizado como um problema de programação linear inteira (ILP – "*integer linear programming*"), uma vez que todas suas variáveis assumem apenas valores discretos (e binários), e a função objetivo e as restrições são expressões lineares. Logo, será utilizado o solver open source "*Coin- or Branch and Cut*" (CBC), capaz de resolver este tipo de problema. Este solver, utilizado em aplicações como AMPL e GAMS, também pode ser chamado a partir da biblioteca OR – Tools para o Python.

Lógica de funcionamento da aplicação

Para um melhor entendimento do funcionamento da aplicação, foram selecionadas etapas estratégicas à execução da solução, são elas: baixar as linguagens e ambientes de trabalho, baixar as bibliotecas e instalá-las, abrir o aplicativo, importação dos dados, verificação de inconsistências, compilação de dados da semana anterior, compilação de informação das crianças, implementação e resolução do modelo matemático, exportação dos resultados, construção de *dashboards*, geração de arquivos finais (Excel, PDF). Vale ressaltar que para evitar problemas no funcionamento da aplicação será disponibilizada uma pasta os arquivos necessários para aplicação. O Excel usado tanto para o preenchimento por parte do CCP quanto para rodar a otimização deve ser o **"Excel_Final.xlsm"**, pois ele possui as macros que evitam erros de entrada manual nos dados.

Pensando em um computador que não tenha todos os insumos para rodar a aplicação o início do processo seria baixar as linguagens e os ambientes de trabalho. O passo a passo para para essa e outras etapas pode ser verificado no **Manual do protótipo.** Em relação às bibliotecas, a instalação será feita por um script em R e as bibliotecas podem ser visualizadas na **Tabela 1** abaixo.

Tabela 1: Bibliotecas utilizadas na aplicação

Linguagem	**Biblioteca**
Python	numpy ortools pandas openpyxl reportlab
R	shiny shinydashboard DT Reticulate ggplot2 Graphics

Feito isso pode-se abrir o aplicativo e fazer a importação de dados, o próprio usuário irá fazer o upload do arquivo de Excel da semana anterior através do Shiny. O objetivo é evitar problemas referentes à localização do arquivo Excel da semana anterior em diferentes diretórios.

Após transferir este arquivo Excel, pode-se pressionar o botão "Otimizar" e iniciará a etapa de tratamento e manipulação dos dados. Nessa parte usamos a biblioteca **Reticulate** para utilizar a biblioteca **Pandas** no R, dessa forma o arquivo em Excel é lido como um *Data Frame* e conseguimos rodar o script em Python chamado "Otimização". O primeiro passo desse script é a identificação, notificação ao usuário e correção das inconsistências contidas no arquivo. As inconsistências tratadas são na **Tabela 2** abaixo.

Tabela 2: Inconsistência e como elas são tratadas

	Inconsistência	**Ação tomada**
I1	A criança tem disponibilidade, mas não consta na planilha de atendimentos regular nem esporádico.	Criança não é considerada na otimização.
I2	Criança tem frequência de atendimento semanal maior que 3.	O número de atendimento torna-se 3.
I3	Criança estava alocada para mais de um médico num mesmo horário.	Criança é realocada em uma das especialidades de forma a otimizar o problema.
I4	Criança não está cadastrada.	Criança não é considerada na otimização.

Além das Inconsistências acima, algumas mensagens são mostradas: as crianças que têm mais demanda de atendimentos que horários disponíveis, crianças não foram atendidas e se a criança não teve todos atendimentos alocados.

Apesar do modelo de arquivo de Excel já conter restrições que minimizam as inconsistências como: limitar entradas de dados de status, horário de atendimento, período do dia, tipo de funcionário (especialidade) com os dados existentes na aba “Auxiliar”, achamos interessante executar uma dupla validação dos dados para assegurar a qualidade da aplicação

O próximo passo, ainda dentro do script em Python, é compilar as informações referentes às alocações da semana anterior em um *Data Frame.* Esta tabela informa onde as crianças foram alocadas na semana passada, facilitando a garantia de permanência destes horários para as semanas seguintes, assim sanando a O1 citada na **Formulação matemática do problema**. Também é informado a disponibilidade e indisponibilidade de cada funcionário em sua especialidade. Um exemplo do *Data Frame* está disposto na **Figura 2**.

Figura 2 - *Data Frame* contendo dados da semana anterior

	Dia	Horário	Tratamento	Funcionário	Horário Discretizado
Criança Atendida					
Criança 01	Segunda	07:30:00	Fonoaudiologia	Funcionário 1	1
Indisponível	Segunda	08:00:00	Fonoaudiologia	Funcionário 1	2
Criança 02	Segunda	13:00:00	Fonoaudiologia	Funcionário 1	12
Criança 01	Segunda	07:30:00	Fonoaudiologia	Funcionário 6	1
Indisponibilidade	Segunda	08:00:00	Fonoaudiologia	Funcionário 6	2
Criança 02	Segunda	13:00:00	Fonoaudiologia	Funcionário 6	12
Criança 04	Segunda	07:30:00	Terapia Ocupacional	Funcionário 2	1
Criança 04	Sexta	17:00:00	Terapia Ocupacional	Funcionário 2	0
Criança 04	Segunda	09:00:00	Nutrição	Dayana	4
Criança 1	Terça	09:00:00	Nutrição	Dayana	23
1234	Terça	13:30:00	Nutrição	Dayana	32
Isabela	Segunda	07:30:00	Pedagogia	Ana Cecília	1
Ana Luiza	Segunda	08:00:00	Pedagogia	Ana Cecília	2
Enzo	Segunda	08:30:00	Pedagogia	Ana Cecília	3
Lorena	Segunda	09:00:00	Pedagogia	Ana Cecília	4
Caio	Segunda	09:30:00	Pedagogia	Ana Cecília	5
Bruno	Segunda	10:00:00	Pedagogia	Ana Cecília	6

Uma vez que informações essenciais das crianças estão dispersas em diferentes abas do arquivo Excel, optou-se por compilar estes dados em um único *Data Frame*. Nesta tabela, as crianças correspondem aos índices, enquanto as colunas informam a disponibilidade, demanda de atendimentos e informação se seus atendimentos podem ser reagendados. Em cada linha da coluna de disponibilidade, foi criado o set de horários discretizados (lista do Python) com a disponibilidade de determinada criança de receber tratamentos. Para cada linha da coluna de tratamentos, foi criado um dicionário Python ondes as chaves são os tratamentos específicos demandados por cada criança e os valores para estas chaves são a quantidade de tratamentos por semana daquela especialidade. Na coluna que checa os reagendamentos, fora criado o mesmo dicionário da coluna de tratamentos, a diferença é que os valores de cada chave são "Sim" ou "Não", indicando se aquela especialidade pode ser reagendada. Um exemplo desta tabela está disposto na **Figura 3**.

Figura 3 - Compilado de dados da crianças.

	Disponibilidade	Tratamentos	Pode reagendar
Criança 09	[1, 2, 3, 4, 5, 6, 7, 8, 9, 20, 21, 22, 23, 24...	{'Fonoaudiologia': 1, 'Terapia Ocupacional': 1...	{'Fonoaudiologia': 'Não', 'Terapia Ocupacional...
Criança 12	[77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 8...	{'Fonoaudiologia': 1, 'Terapia Ocupacional': 1...	{'Fonoaudiologia': 'Não', 'Terapia Ocupacional...
Criança 20	[10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 29, 3...	{'Fonoaudiologia': 1, 'Terapia Ocupacional': 1...	{'Fonoaudiologia': 'Não', 'Terapia Ocupacional...
Criança E17	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14...	{'Terapia Ocupacional': 1, 'Fisioterapia': 1}	{}
Criança E31	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14...	{'Fisioterapia': 1}	{}
Criança E11	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14...	{'Terapia Ocupacional': 1, 'Fisioterapia': 1}	{}
Criança 16	[58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 6...	{'Fonoaudiologia': 0, 'Terapia Ocupacional': 3...	{'Fonoaudiologia': 'Não', 'Terapia Ocupacional...
[illegible]	[illegible]	[illegible]	[illegible]

Após compiladas as informações da semana anterior na tabela da **Figura 2** e os dados de cada criança na tabela da **Figura 3**, a próxima etapa foi a implementação computacional do modelo matemático. Para criar as variáveis de decisão Xc,t,h,f foi utilizado um set de crianças que contém todas as crianças cadastradas na planilha de “Disponibilidade da Criança” do arquivo modelo Excel. No entanto, crianças que estavam com a disponibilidade cadastrada, mas não apresentavam nenhuma demanda de atendimentos nas abas “Atendimento Regular” e “Atendimento Esporádico” foram excluídas deste set. Para cada criança do conjunto citado acima foi utilizado um set distinto de horários disponíveis segundo as informações da tabela representada na **Figura 3**. O mesmo foi realizado para as especialidades das crianças, para cada uma delas, o set de especialidades representava apenas os tratamentos que a criança demanda. O set de funcionários também fora criado diferentemente para cada especialidade, respeitando os funcionários disponíveis em cada área.

O intuito de criar diferentes sets para os índices das variáveis de decisão Xc,t,h,f é assegurar que variáveis inúteis não fossem declaradas. Por exemplo, caso a “Criança 04” não tenha disponibilidade no horário “5”, nenhuma variável $X4,5,*,*$ será declarada. O mesmo ocorre para todos os horários indisponíveis, tratamentos não demandados e funcionário não requisitados por cada criança. Desta forma, o modelo diminui consideravelmente o número de variáveis, tempo de processamento e eliminando a necessidade de restrições extras que excluiriam tais variáveis inúteis.

A **Figura 4** apresenta o exemplo dos índices (em forma de tuplas do Python) de algumas das variáveis criadas referentes à “Criança 09”. É possível observar que esta criança demanda os atendimentos de fonoaudiologia, nutrição, neurologia e terapia ocupacional.

Figura 4 - Índices de algumas das variáveis da “Criança 09”

```
[('Criança 09', 1, 'Fonoaudiologia', 'Funcionário 1'),
 ('Criança 09', 1, 'Fonoaudiologia', 'Funcionário 6'),
 ('Criança 09', 1, 'Terapia Ocupacional', 'Funcionário 2'),
 ('Criança 09', 1, 'Terapia Ocupacional', 'Funcionário 3'),
 ('Criança 09', 1, 'Neurologia', 'Funcionário 4'),
 ('Criança 09', 1, 'Nutrição', 'Dayana'),
 ('Criança 09', 2, 'Fonoaudiologia', 'Funcionário 1'),
 ('Criança 09', 2, 'Fonoaudiologia', 'Funcionário 6'),
 ('Criança 09', 2, 'Terapia Ocupacional', 'Funcionário 2'),
 ('Criança 09', 2, 'Terapia Ocupacional', 'Funcionário 3'),
 ('Criança 09', 2, 'Neurologia', 'Funcionário 4'),
 ('Criança 09', 2, 'Nutrição', 'Dayana'),
 ('Criança 09', 3, 'Fonoaudiologia', 'Funcionário 1'),
 ('Criança 09', 3, 'Fonoaudiologia', 'Funcionário 6'),
 ('Criança 09', 3, 'Terapia Ocupacional', 'Funcionário 2'),
 ('Criança 09', 3, 'Terapia Ocupacional', 'Funcionário 3'),
 ('Criança 09', 3, 'Neurologia', 'Funcionário 4'),
 ('Criança 09', 3, 'Nutrição', 'Dayana'),
 ('Criança 09', 4, 'Fonoaudiologia', 'Funcionário 1'),
```

As variáveis de decisão auxiliares para cada família de restrição foram declaradas respeitando seus índices e o carácter binário de cada uma. Após implementar as restrições elencadas na seção de **Modelagem Matemática** deste relatório e definir a função objetivo como a somatória de todas as variáveis $X_{c,t,h,f}$, foi necessário integrar as informações referentes aos atendimentos da semana anterior ao modelo matemático para garantir que estes horários fossem preservados na medida do possível.

Através de uma lógica de programação, para cada linha do *Data Frame* ilustrado na **Figura 2**, caso a criança ainda demande aquele atendimento e não possa ter seu horário remanejado naquela especialidade, o coeficiente na função objetivo é igualado a um valor convenientemente alto (1000). O objetivo de aumentar o coeficiente da função objetivo destas variáveis é assegurar que o modelo dê preferência para que estas variáveis assumam o valor 1, permitindo ainda que a variável também possa assumir o valor zero (0) caso o modelo seja infactibilizado pela ativação da variável.

Como exemplo podemos utilizar o caso da "Criança 01", que foi alocada na semana anterior (no arquivo modelo de Excel) em duas especialidades distintas no horário discretizado 1. Caso o modelo fixasse o valor destas duas variáveis em 1, esta condição não respeitaria a restrição de que cada criança não pode receber dois atendimentos no mesmo horário e infactibilizaria o modelo. No entanto, ao aumentar o valor do coeficiente destas variáveis na FO, asseguramos que uma destas variáveis assumirá o valor 1, enquanto a outra assumirá o valor 0, preservando a factibilidade do modelo.

Para lidar com a periodicidade quinzenal da especialidade de nutrição também implementamos uma lógica de programação que altera algumas componentes do modelo matemático já implementado. Caso uma criança tenha sido atendida na semana anterior na especialidade de nutrição, tenha frequência quinzenal de atendimentos e ainda conste esta demanda de tratamento, o programa fixará todas as variáveis referentes ao tratamento de nutrição desta criança no valor 0. Desta forma, a criança não será atendida na semana presente, uma vez que foi atendida na semana anterior. Caso a criança demande um atendimento em nutrição na semana presente e não tenha sido atendida na semana anterior, o modelo irá alocá-la normalmente.

Ao final do script em Python, os dados do solver serão compilados em um *Data Frame*. Nesta tabela é possível realizar buscas filtradas por crianças, especialidades ou funcionários. Uma ilustração de parte desta tabela está disposta na **Figura 5**. Além disso geramos algumas estatísticas sobre a solução e a formatação de um PDF com o horário final de todas as crianças. Em relação às estatísticas, são gerados o número de crianças atendidas em todos os atendimentos, número das que foram parcialmente atendidas (foram atendidas em uma ou mais especialidades mas não em todas que demandava) e crianças que não foram atendidas em nenhuma especialidade. Já em relação ao PDF, usamos o *Data Frame* da **Figura 5** para gerar uma saída de um PDF individual para cada criança.

Figuras 5 - *Data Frame* com resultados do Solve.

Crianças Atendidas	Horário	Tratamento	Funcionário
Criança 09	1	Fonoaudiologia	Funcionário 6
Criança 09	3	Neurologia	Funcionário 4
Criança 09	4	Nutrição	Dayana
Criança 09	6	Terapia Ocupacional	Funcionário 3
Criança 12	78	Neurologia	Funcionário 4
Criança 12	79	Fonoaudiologia	Funcionário 6
Criança 12	82	Nutrição	Dayana
Criança 12	84	Terapia Ocupacional	Funcionário 2
Criança 20	10	Neurologia	Funcionário 4
Criança 20	15	Nutrição	Dayana
Criança 20	16	Terapia Ocupacional	Funcionário 3
Criança 20	18	Fonoaudiologia	Funcionário 6
Criança E17	13	Terapia Ocupacional	Funcionário 2
Criança E11	60	Terapia Ocupacional	Funcionário 3
Criança 16	62	Terapia Ocupacional	Funcionário 3
Criança 16	65	Nutrição	Dayana
Criança 16	66	Neurologia	Funcionário 4
Criança E13	8	Terapia Ocupacional	Funcionário 2

Por último, os dados são transferidos novamente ao R, Além disso são gerados gráficos no próprio R e botões que acionam o script em Python para gerar tabelas e dados no ShinyApp.

6. Referências Bibliográficas

MONLEVADE, João. Modelagem para a definição de turnos – extensão e novas restrições. Trabalho de conclusão de curso - Instituto de Ciências Exatas e Aplicadas Colegiado do Curso de Engenharia de Produção. Universidade Federal de Ouro Preto. Minas Gerais, Ouro Preto, 2017.

ARENALES, M.; ARMENTANO, V.; MORABITO, R.; YANASSE, H. Pesquisa Operacional. Rio de Janeiro: Elsevier, 2007.

GRÖBNER, Matthias; WILKE, Peter; BÜTTCHER, Stefan. A standard framework for timetabling problems. In: International Conference on the Practice and Theory of Automated Timetabling. Springer, Berlin, Heidelberg, 2002. p. 24-38

CARVALHO, R.. Abordagem heurística para o problema de programação de horários de cursos. 2011. Dissertação – Engenharia Elétrica, Escola de Engenharia da Universidade Federal de Minas Gerais, Belo Horizonte.

MARTINS, J. P.. O Problema do Agendamento Semanal de Aulas. Dissertação - Instituto de Informática, Universidade Federal de Goiás, Goiânia, 2010.

Sites das documentações das principais bibliotecas e solvers utilizados:

- https://developers.google.com/optimization/
- https://pandas.pydata.org/pandas-docs/stable/
- https://rstudio.github.io/reticulate/
- https://shiny.rstudio.com/
- https://projects.coin-or.org/Cbc