Pesquisa Operacional: Alocação de alunos em tarefas realizadas em grupo

Gabriel L. Gomes[1], Geovane F. S. Santos[1], Luigi D. C. Soares[1]

[1]Instituto de Ciências Exatas e Informática
Pontifícia Universidade Católica de Minas Gerais (PUC-MG)
Belo Horizonte, MG - Brasil

glgomes@sga.pucminas.br, geovane.fonseca@sga.pucminas.br,

luigi.soares@sga.pucminas.br

1. Introdução

Problemas de alocação exploram a busca pela melhor distribuição de recursos dentre as diversas soluções possíveis. Objetiva-se, por exemplo, minimizar o custo total do modelo ou maximizar seu valor final. Tais cenários apresentam como componentes principais conjuntos de: (a) recursos disponíveis; (b) trabalhos que precisam ser executados; e (c) custos ou retornos relativos a cada item [Eilon et al. 2012].

Dentre os contextos relacionados ao assunto abordado encontra-se o de atribuição de pessoas em determinadas tarefas. Nesse caso, entende-se como recurso cada indivíduo presente no cenário. Assim, tem-se por objetivo encontrar a distribuição que melhor atende aos requisitos do problema. Em [Paula et al. 2012], por exemplo, foi apresentado um modelo de atribuição de tutores à cursos de ensino a distância. Já em [Gosselin and Truchon 1986], foi desenvolvido um sistema automatizado para a designação de salas de aulas de acordo com diferentes cursos. O ambiente escolar, portanto, se apresenta como um tema em potencial para o desenvolvimento de pesquisas relacionadas a alocação de recursos.

Atividades acadêmicas são típicamente realizadas de forma coletiva. Nesse contexto, cada indíviduo apresenta facilidades ou dificuldades em determinadas disciplinas. Além disso, existem temas que requerem maior esforço dos participantes. Assim, é necessário também determinar a quantidade de alunos que serão responsáveis por essa tarefa. Para melhor definir a proposta identificada, podemos representá-la por meio de uma pergunta: "Qual a maneira mais adequada de se alocar responsáveis por etapas de determinados trabalhos, de forma a definir os profissionais mais competentes para cada disciplina?".

Dessa forma, o presente trabalho pretende otimizar a atribuição de alunos nas tarefas escolares. Como critério para a alocação, serão consideradas as competências de cada indivíduo em relação as matérias cursadas. Portanto, a justificativa para o desenvolvimento desta pesquisa está na necessidade de determinar a melhor forma de alocar indivíduos, de maneira a maximizar a qualidade dos projetos desenvolvidos. Pensando nisso, tem-se por objetivo a produção de uma aplicação que visa solucionar o problema identificado.

2. Referencial Teórico

2.1. Pesquisa Operacional

Pesquisa Operacional (PO) é um ramo multidisciplinar aplicado a problemas que buscam compreender a condução e a coordenação de operações (i.e. atividades) em uma organização. Ademais, a PO tem como objetivo buscar uma melhor solução (conhecida como solução ótima) para o modelo que representa o problema identificado. Nesse sentido, é dito que busca-se uma melhor solução, porque podem existir diversas soluções ótimas [Hillier and Lieberman 2013].

Para o problema observado, deve-se definir um modelo matemático que busque minimizar ou maximizar um valor objetivo. Além disso, o modelo deve conter as restrições identificadas durante a definição do problema. Problemas matemáticos podem ser de diversos tipos, entre eles os que envolvem programação linear.

2.2. Programação Linear

Programação linear é o mecanismo mais natural a ser utilizado para formular uma vasta gama de problemas com relativamente pouco esforço. Este é caracterizado por funções lineares compostas de n variáveis. O objetivo (minimizar ou maximizar) é linear com relação as incógnitas e as restrições são igualdades ou desigualdades lineares [Luenberger and Ye 2008].

Como exemplo, considere uma refinaria de petróleo que deseja determinar qual o lucro máximo que pode obter considerando a venda de gasolina coum e aditivada. Sabe-se que o lucro da gasolina comum é de R$ 1,00 por litro e o da aditivada é de R$ 2,00 por litro. Além disso, sabe-se que a produção das gasolinas envolve os seguintes itens e seus gastos: queronese (8 un. para gasolina comum e 2 un. para aditivada), álcool (1 un. para gasolina comum e 1 un. para aditivada) e aditivo (2 un. para gasolina comum e 7 un. para aditivada). Por fim, tem-se as seguintes disponibilidades: 16 un. de querosene, 6 un de ácool e 28 un. de aditivo). O modelo matemático para esse problema pode ser definido da seguinte forma:

$$F.O \longrightarrow max\ Z = x + 2y$$

sujeito à:

$$\begin{aligned} 8x + 2y &\leq 16 && (R1) \\ x + y &\leq 6 && (R2) \\ 2x + 7y &\leq 28 && (R3) \\ x, y &\geq 0 && (R4) \end{aligned}$$

2.3. *Simplex*

Simplex é um algoritmo de otimização desenvolvido por George Dantzig, em 1946. Este método é frequentemente adotado para a resolução de problemas lineares. Suas etapas consistem em: (i) determinar a área permissiva (região convexa e que contém todas as soluções possíveis); e (ii) analisar os vértices adjacentes ao atual, de forma

que o valor da função objetivo cresça ou permaneça inalterado. Além disso, o algoritmo é capaz de determinar o tipo da solução (ótima, múltipla, infinita ou impossível). Por fim, pode ser observado que o Simplex segue uma abordagem gulosa. Isto é, ele busca a melhor solução a cada iteração.

2.4. Programação Inteira

Programação Inteira diz respeito a problemas de otimização em que algumas ou todas as variáveis são restritas a números inteiros. Em muitos cenários, o termo se refere à Programação Linear Inteira. Nesta, a função objetivo e as restrições do modelo (além das restrições de números inteiros) são lineares. Existe, ainda, um caso particular conhecido como Programação Linear Inteira 0-1, em que tais variáveis se restringem a valores binários (i.e. assumem valor zero ou um). Neste contexto, encontram-se os problemas de alocação de recursos.

2.5. *Branch-and-Bound*

O método *Branch-and-Bound* se baseia na ideia de desenvolver uma enumeração das soluções pertencentes a região permissiva. Isto é, possíveis candidatas à solução ótima inteira do problema. O termo *branch* se refere ao fato de que o algoritmo particiona o espaço de soluções. Nesse caso, cada solução é resolvida pelo método *Simplex*. Dessa forma, apenas uma fração das soluções factíveis é realmente examinada [Alves 2019].

3. Revisão da Literatura

A presente revisão tem por objetivo apresentar de forma mais detalhada assuntos diratemente ligados à área de PO. Em especial, busca-se aprofundas nas temáticas que dizem respeito a alocação de recursos, posto que o problema identificado situa-se nesta área. A seguir encontram-se detalhadas algumas aplicações relacionadas a PO.

3.1. Pesquisa Operacional na Educação

Em [Johnes 2015], são descritos diversos problemas relacionados ao contexto da educação, além de algumas abordagens possíveis de serem utilizadas em cada situação. Educadores enfrentam uma infinidade de problemas no dia a dia de suas instituições. Eles se relacionam, por exemplo, com a alocação ótima de seu orçamento, ou simplesmente o lugar onde cada turma deve ocorrer e quem deve ensiná-la. Dentre as mais diversas aplicações, PO também apresenta ferramentas endereçadas a essas questões. Esses problemas são categorizados como problemas de *management, timetabling* e *scheduling*.

É possível visualizar o setor de educação como uma série de componentes, tendo em vista que este é composto por diferentes níveis, como primário, secundário, profissional e terciário. Esses níveis estão interligados de tal maneira que cada indivíduo pode seguir um caminho que atenda às suas próprias aspirações educacionais e de treinamento. Portanto, pode-se definir a educação como um sistema. A adoção dessa visão permite que os pesquisadores modelem o sistema usando uma variedade de abordagem e forneçam previsões úteis para gerentes, planejadores e formuladores de determinadas políticas educacionais.

3.1.1. Alocação de recursos

Planejadores estão interessados em projeções futuras dos alunos e de necessidades (em termos de professores e equipamentos) em todos os níveis de ensino. Governantes precisam saber não somente quantos estudantes se concentram em cada nível educacional, mas também quanto dinheiro é necessário para financiar os números previstos. Um educador ou uma escola também tem como necessidade alocar seus recursos para fornecer educação de acordo com a demanda prevista. Mas para o sistema educacional como um todo, conflitos potenciais entre objetivos conflitantes devem ser considerados. Isso nos leva à tomada de decisões multi-objetivas, nas quais *goal programming* - um ramo da otimização multiobjetivo, que pode ser pensado como uma extensão ou generalização da programação linear para lidar com múltiplas medidas objetivas normalmente conflitantes - é uma abordagem metodológica popular.

3.1.2. *Scheduling*

Aplicações de PO para roteamento e agendamento de problemas no contexto educacional são inúmeras. Os exemplos incluem a roteirização e o agendamento de autocarros, o agendamento de cursos e exames, a atribuição de alunos a turmas e a de estudantes a grupos de trabalho. Transporte de alunos e entrega de alimentos às escolas são exemplos de problemas desse tipo. Programação inteira - um modelo de progração linear no qual algumas ou todas as variáveis do problema pertencem ao conjunto dos números inteiros - tem sido utilizada em problemas similares: alocação de contratos dos responsáveis pelo transporte escolar e pela entrega alimentícia.

3.1.3. Atribuição

Existem diversas situações em que os elementos de uma população devem ser atribuídos a grupos, salas, instituições ou horários. Esta é uma área na qual ferramentas de PO podem ser aplicadas com um sucesso considerável. A necessidade de designar alunos para grupos (e.g. distribuir alunos em classes) é uma questão frequentemente encontrada no dia a dia das escolas e universidades.

Esse tipo de problema pode chegar a ser tão complexo que um modelo de programação inteira se torna algo inviável. Isto, porque esse método não pode entregar uma resposta em um limite de tempo aceitável. Por conseguinte, é prática comum recorrer a uma abordagem heurística que identificará uma solução num prazo viável. Esta, por sua vez, não será ótima, mas pode ser garantida como tendo um certo nível mínimo de qualidade.

3.2. Pesquias Operacional em Redes de Distribuição de Energia Elétrica

Uma outra situação é a definição de dimensionamento de equipes para atendimentos em determinada rede de distrubuição. Por meio da Pesquisa Operacional (PO), o modelo matemático de programação inteira e o algoritmo de Floyd é possível determinar, de forma otimizada, o dimensionamento e atendimento das equipes a

fim de satisfazer os clientes de determinada agência. Em [Maria Steiner 2006] é explorada uma solução genérica dessa aplicação, que pode ser aplica em quaisuqer outras redes de distribuição elétricas, devido o seu resultado satisfatório.

Para uma rede de distribuição de energia, se faz necessária a presença de centrais de atendimento para descentralizar o controle dos usuários e apresentar melhor eficiência no suporte, caso algum problema aconteça. Junto a isso, é notório que para cada tipo de disfunção, medidas específicas serão necessárias para a solução, o que requer pessoas qualificadas para cada um destes tipos. Em [Maria Steiner 2006], é explorada, por meio da PO, a alocação de pessoas e/ou equipes, para uma determinada agência, a fim de solucionar o defeito emergenciais de forma imediata e que o tempo de deslocamento seja o mínimo possível.

Cada tipo de chamado aberto pelo cliente é registrado via software e, por meio dele é possível saber o tipo de serviço e onde se encontra e, assim, encaminhada à unidade de atendimento mais próxima. Para o despacho das equipes para atendimento desta solicitação, é utilizado o algoritmo de Floyd, a fim de se encontrar o caminho minimo até o local da ocorrência. No artigo é abordado um exemplo de atendimento emergencial para uma determinada agência, onde se encontra determinada quantidade de funcionários próprios e tercerizados, com o objetivo de distribuir os eletricitas ou equipes de forma que, caso aconteça mais de uma solicitação de emergência, será possível o atendimento sem que prejudique a promessa de auxílio imediato.

4. Metodologia

A metodologia é descrita em três etapas, sendo elas: (a) a identificação do problema; (b) a modelagem; e (c) a análise dos resultados. Na identificação, hipóteses foram levantadas e a de maior destaque foi selecionada. Já na modelagem, verificou-se a classificação do problema quanto a sua linearidade e o modelo matemático foi construído. Por fim, uma aplicação foi desenvolvida para observar e examinar os resultados do exemplo apresentado.

O tempo previsto para a realização do trabalho prático foi de quatro meses, ou seja, um semestre acadêmico. Desse tempo, cerca de três semanas foram destinadas a definição do problema a ser abordado. Na semana posterior, realizou-se a sua validação. Após isso, um mês foi necessário para o desenvolvimento do modelo e do esboço da aplicação. Esta, por sua vez, necessitou de mais um mês para Ser finalizada. O último mês da previsão foi usado para as correções necessárias e a elaboração da análise do que foi feito.

4.1. Modelagem

Para melhor visualização do problema, optou-se por modelá-lo na forma de um grafo bipartido (figura 2). Um grafo $G\ (V, A)$ é definido pelo par de conjuntos V e A, onde o primeiro refere-se aos vértices e o segundo as arestas da estrutura. Dessa forma, tem-se que o conjunto de vértices será composto por estudantes e tarefas, enquanto as arestas determinarão a relação entre um aluno a_i e as respectivas tarefas t. Cada aresta possuirá um peso com um certo valor v representando a competência do aluno naquela área. Os vértices à esquerda do grafo representam os alunos, enquanto os à

direita indicam as tarefas. Ao lado de cada tarefa se encontra a restrição referente ao nível de dificuldade. A tabela 1 apresenta as competências de cada aluno, enquanto 2 a tabela contém as dificuldades de cada tarefa. O modelo 1 é uma representação gráfica para este exemplo.

	Tarefa 1	Tarefa 2	Tarefa 3	Tarefa 4	...	Tarefa k
Aluno 1	a_1	a_2	a_3	a_4	...	a_k
Aluno 2	b_1	b_2	b_3	b_1	...	b_k
Aluno 3	c_1	c_2	c_3	c_4	...	c_k
$\vdots$					...	
Aluno n	n_1	n_2	n_3	n_4	...	n_k

Tabela 1. Modelo gerénico para competências de cada aluno.

	Dificuldade
Tarefa 1	R_{n+1}
Tarefa 2	R_{n+2}
Tarefa 3	R_{n+3}
$\vdots$	
Tarefa k	R_{n+k}

Tabela 2. Modelo genérico para dificuldade de cada tarefa.

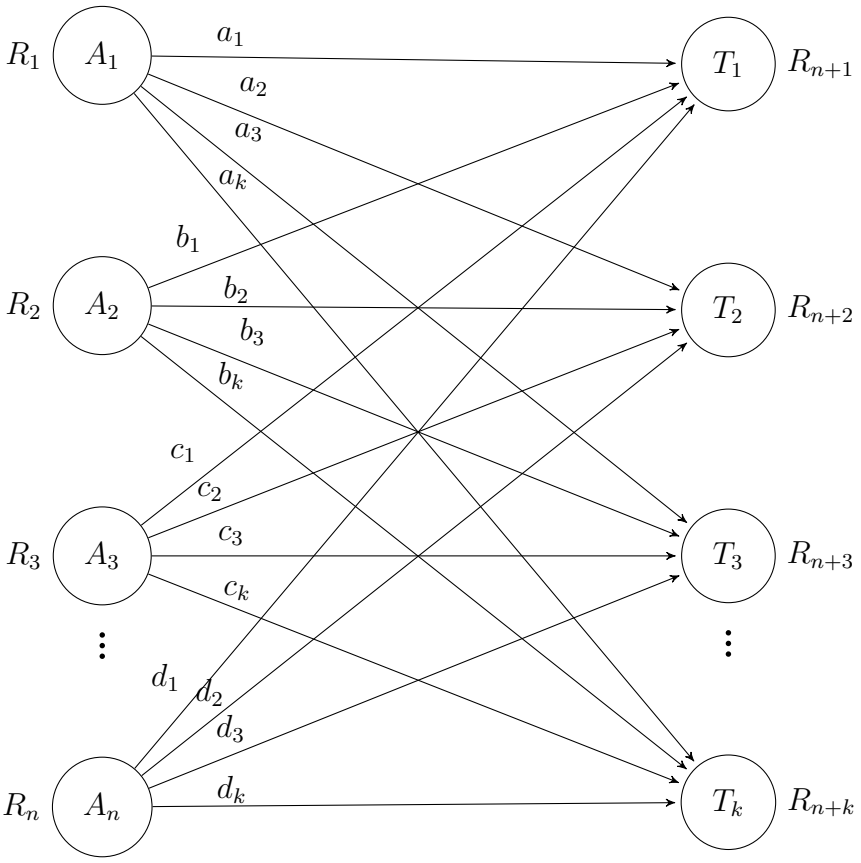

Figura 1. Representação de um problema genérico através de um grafo bipartido.

4.2. Modelo Matemático

De maneira geral, considerando n alunos e k tarefas, a função objetivo pode ser definida da seguinte forma (x representa a alocação ou não de determinado aluno em cada tarefa com dificuldade d, e c sua competência):

Sendo A o conjunto de alunos e T o conjunto de tarefas, tem-se:

$$F.O \longrightarrow max\ Z = \sum_{a=1}^{n} \sum_{t=1}^{k} c_a^t * x_a^t$$

sujeito à:

$$\sum_{t=1}^{k} x_{a_i}^t \geq 1\ \forall a_i \in A\ (1)$$

$$1\ \leq \sum_{a=1}^{n} c_a^{t_i} * x_a^{t_i} \geq d_{t_i}\ \forall t_i\ \in T\ (2)$$

$$x_{a_i}^{t_j} \geq 0\ \forall a_i \in A, \forall t_j \in T (3)$$

(1) - Cada integrante deve ter pelo menos 1 alocação
(2) - Cada tarefa deverá ter entre 1 e d_t competência para ser suprida
(3) - Alocação não negativa

4.3. Instrumentos Utilizados

O projeto se divide em duas seções, *Back-end* e *Front-end*. Para o primeiro, optou-se pela utilização da linguagem de programação Python. Nele se encontra toda a lógica referente ao servidor. Este foi desenvolvido com o auxílio do framework Flask RESTplus, que é utilizado para a criação de uma API REST. Nesse servidor, encontra-se a implementação do modelo do problema, realizada por meio da biblioteca PuLP. Os modelos construídos e solucionados no *back-end* são armazenados em um banco de dados. Para isso, optou-se pela utilização do MongoDB, um banco não-relacional.

Além da modelagem do cenário, o PuLP oferece métodos de solução do problema. Apesar disso, ela não se restringe a própria implementação, uma vez que permite ao desenvolvedor adotar *solvers* externos. Nesse sentido, fez-se a escolha pelo GLPK. Este é um pacote de software destinado a resolver programação linear em grande escala, programação inteira mista e outros problemas relacionados. Para a solução de problemas lineares que incluem restrições de variáveis inteiras, o framework utiliza o método *Branch-and-Bound*.

O *Front-end*, por sua vez, consiste em uma aplicação móvel nativa (Android), desenvolvida na linguagem Kotlin. A implementação do aplicativo foi realizada de acordo com o padrão MVVM (*Movel, View, ViewModel*), utilizando a coleção de componentes Jetpack[1]. A comunicação entre a aplicação e o servidor é realizada por

[1]https://developer.android.com/jetpack?hl=pt-br

meio de requisições HTTP (GET, POST), utilizando, para isso, uma combinação entre *Coroutines*[2,3] e a biblioteca *Retrofit*[4].

5. Desenvolvimento

O problema a ser solucionado é de programação linear e envolve a alocação de estudantes em determinadas tarefas. Para tanto, serão consideradas informações de três aluno e cinco tarefas distintas. Cada indivíduo apresenta um certo nível de competência em cada uma das cinco tarefas (tabela 3). Por sua vez, cada tarefa possui um determinado nível de dificuldade (tabela 4). O grafo 2 foi modelado a partir das seguintes restrições:

1. Cada aluno deve estar alocado a pelo menos uma tarefa;
2. Cada tarefa deve ter um somatório total de competências associadas maior que 1, tendo em vista que todas devem ser realizadas;
3. Cada tarefa deve ter um somatório total de competências associadas menor que sua dificuldade.

	Tarefa 1	Tarefa 2	Tarefa 3	Tarefa 4	Tarefa 5
Aluno 1	3	5	5	3	8
Aluno 2	8	5	5	2	3
Aluno 3	8	3	3	2	2

Tabela 3. Exemplo competências de cada aluno.

	Dificuldade
Tarefa 1	13
Tarefa 2	8
Tarefa 3	8
Tarefa 4	13
Tarefa 5	5

Tabela 4. Exemplo dificuldade de cada tarefa.

[2]https://developer.android.com/jetpack?hl=pt-br
[3]https://github.com/JakeWharton/retrofit2-kotlin-coroutines-adapter
[4]https://square.github.io/retrofit/

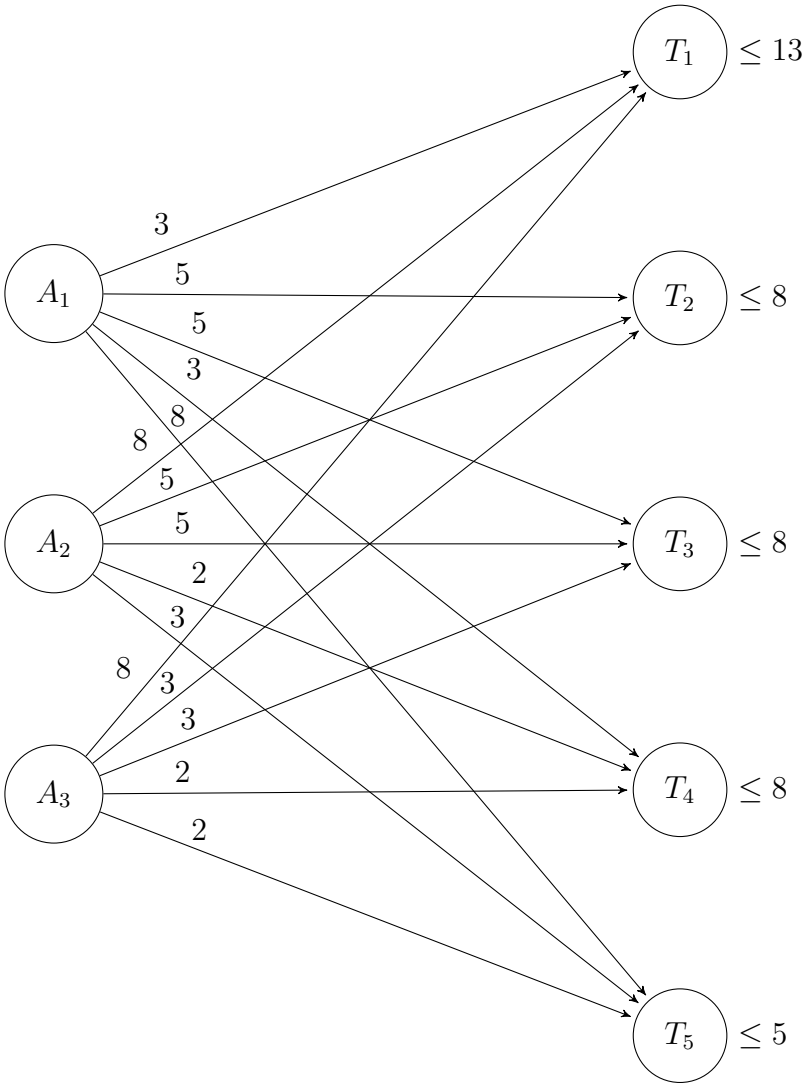

Figura 2. Exemplo da representação do problema através de um grafo bipartido.

5.1. Modelo Matemático

O objetivo a ser alcançado é de maximizar a quantidade total de competências alocadas em todos os projetos. Para tanto, deve-se levar em consideração as restrições definidas anteriormente. A partir do grafo, é possível identificar que as variáveis da função serão as arestas entre alunos e tarefas. Além disso, as restrições do modelo serão os vértices da estrutura. Dessa forma, tem-se um total de 15 variáveis e 8 restrições. Assim, o modelo inicial pode ser definido da seguinte forma (considere $x_{a_i}^{t_j}$ como a variável referente ao aluno i, tarefa j:

$$\begin{aligned} F.O \longrightarrow max\ Z = 3x_{a_1}^{t_1} + 5x_{a_1}^{t_2} + 5x_{a_1}^{t_3} + 3x_{a_1}^{t_4} + 8x_{a_1}^{t_5} + \\ 8x_{a_2}^{t_1} + 5x_{a_2}^{t_2} + 5x_{a_2}^{t_3} + 2x_{a_2}^{t_4} + 3x_{a_2}^{t_5} + \\ 8x_{a_3}^{t_1} + 3x_{a_3}^{t_2} + 3x_{a_3}^{t_3} + 2x_{a_3}^{t_4} + 2x_{a_3}^{t_5} \end{aligned}$$

sujeito à:

$$x_{a_1}^{t_1} + x_{a_1}^{t_2} + x_{a_1}^{t_3} + x_{a_1}^{t_4} + x_{a_1}^{t_5} \geq 1 \qquad (R1)$$
$$x_{a_2}^{t_1} + x_{a_2}^{t_2} + x_{a_2}^{t_3} + x_{a_2}^{t_4} + x_{a_2}^{t_5} \geq 1 \qquad (R2)$$
$$x_{a_3}^{t_1} + x_{a_3}^{t_2} + x_{a_3}^{t_3} + x_{a_3}^{t_4} + x_{a_3}^{t_5} \geq 1 \qquad (R3)$$

$$1 \leq 3x_{a_1}^{t_1} + 8x_{a_2}^{t_1} + 8x_{a_3}^{t_1} \leq 13 \qquad (R4)$$
$$1 \leq 5x_{a_1}^{t_2} + 5x_{a_2}^{t_2} + 3x_{a_3}^{t_2} \leq 8 \qquad (R5)$$
$$1 \leq 5x_{a_1}^{t_3} + 5x_{a_2}^{t_3} + 3x_{a_3}^{t_3} \leq 8 \qquad (R6)$$
$$1 \leq 3x_{a_1}^{t_4} + 2x_{a_2}^{t_4} + 2x_{a_3}^{t_4} \leq 8 \qquad (R7)$$
$$1 \leq 8x_{a_1}^{t_5} + 3x_{a_2}^{t_5} + 2x_{a_3}^{t_5} \leq 5 \qquad (R8)$$

restrições de não negatividade:

$$x_{a_1}^{t_1}, x_{a_1}^{t_2}, x_{a_1}^{t_3}, x_{a_1}^{t_4}, x_{a_1}^{t_5} \geq 0$$
$$x_{a_2}^{t_1}, x_{a_2}^{t_2}, x_{a_2}^{t_3}, x_{a_2}^{t_4}, x_{a_2}^{t_5} \geq 0$$
$$x_{a_3}^{t_1}, x_{a_3}^{t_2}, x_{a_3}^{t_3}, x_{a_3}^{t_4}, x_{a_3}^{t_5} \geq 0$$

5.2. Modelo Dual

As variáveis da função objetiva do modelo dual são obtidas por meio das restrições do primal. Estão, portanto, relacionadas à quantidade de tarefas mínimas para cada aluno e a dificuldade de cada tarefa. Com isso, é possível perceber que tais variáveis possuem propósitos distintos (i.e. relacionadas às pessoas ou às tarefas). Isto provoca incertezas com relação a interpretabilidade do modelo. Consequentemente, tem-se que os modelos dual e primal não convergem.

A seguir se encontra um exemplo com uma versão simplificada do modelo proposto:

Modelo Primal:

$$F.O \longrightarrow max\ Z = 3x_{a_1}^{t_1} + 5x_{a_1}^{t_2} + 8x_{a_2}^{t_1} + 5x_{a_2}^{t_2}$$

sujeito à:

$$x_{a_1}^{t_1} + x_{a_1}^{t_2} \geq 1 \qquad (R1)$$
$$x_{a_2}^{t_1} + x_{a_2}^{t_2} \geq 1 \qquad (R2)$$
$$1 \leq 3x_{a_1}^{t_1} + 8x_{a_2}^{t_1} \leq 13 \qquad (R5)$$
$$1 \leq 5x_{a_1}^{t_2} + 5x_{a_2}^{t_2} \leq 8 \qquad (R6)$$

restrições de não negatividade:

$$x_{a_1}^{t_1}, x_{a_1}^{t_2}, \qquad x_{a_2}^{t_1}, x_{a_2}^{t_2} \geq 0$$

Modelo Dual:

$$F.O \longrightarrow min\ Zy = -y1 - y2 - y3 - y4 + 13y5 + 8y6$$

sujeito à:

$$-y1 - 3y3 + 3y5 \geq 3 \quad (R1)$$
$$-y1 - 5y4 + 5y6 \geq 5 \quad (R2)$$
$$-y2 - 8y3 + 8y3 \geq 8 \quad (R3)$$
$$-y2 - 5y4 + 5y6 \geq 5 \quad (R4)$$

restrições de não negatividade:

$$y1, y2, y3, y4, y5, y6 \geq 0$$

5.3. Exemplo de codificação

A seguir encontra-se um exemplo de modelagem de problemas por meio da biblioteca PuLP. Além da definição do modelo, tem-se a chamada do método *solve* para a resolução do problema. Nesta, é possível determinar *solvers* externos de acordo com a necessidade do desenvolvedor. No exemplo, é possível perceber a utilização do pacote GLPK.

```
# Criacao da variavel "prob" que sera responsavel pelos dados do problema
prob = LpProblem("Group tasks", LpMaximize)

# Criacao das variaveis do aluno x tarefa, definindo seu tipo como binario
x1t1 = LpVariable("Student 1 - Task 1", cat = LpBinary)
x1t2 = LpVariable("Student 1 - Task 2", cat = LpBinary)
⋮
x3t4 = LpVariable("Student 3 - Task 4", cat = LpBinary)
x3t5 = LpVariable("Student 3 - Task 5", cat = LpBinary)


# Insercao da funcao objetivo
prob += 3*x1t1 + 5*x1t2 + 5*x1t3 + 3*x1t4 + 8*x1t5 \
     + 8*x2t1 + 5*x2t2 + 5*x2t3 + 2*x2t4 + 3*x2t5 \
     + 8*x3t1 + 3*x3t2 + 3*x3t3 + 2*x3t4 + 2*x3t5 \
     , "Allocated members"

# Inclusao das restricoes do problema
prob += 1 <= x1t1 + x1t2 + x1t3 + x1t4 + x1t5 <= 4,
       "Lower and Upper tasks bounds to member A"
⋮
prob += 1 <= 8*x1t5 + 3*x2t5 + 2*x3t5 <= 5,
       "Lower and Upper complexity bounds to task 5"

# Resolucao do problema pelo metodo Simplex, utilizando a biblioteca GLPK
prob.solve(pulp.GLPK())

# Solucao encontrada: alocacao de tarefa para cada membro do grupo
for v in prob.variables():
    print(v.name, "=", v.varValue)
```

Saída obtida após execução do programa:

Integrante	Tarefa I	Tarefa II	Terefa III	Tarefa IV	Tarefa V
1	1	0	1	1	0
2	1	1	0	1	1
3	0	1	1	1	1

Tabela 5. Alocação dos membros

Como os resultados são binários, a tabela apresenta a alocação ou não de uma tarefa para um determinado integrante. Ou seja, para o membro 1, foram alocadas as tarefas I, II e III e assim por diante.

5.4. Solução

A aplicação móvel será responsável por recolher os dados do modelo a ser analisado. Após a confirmação pelo usuário, estes serão enviados ao servidor. Este, por sua vez, receberá os dados e gerará o modelo matemático de acordo com as restrições definidas. Após isto, o problema será resolvido pelo GLPK e o resultado será armazenado no banco de dados, onde o aplicativo poderá consultar e disponibilizar ao usuário as informações obtidas.

Figura 3. Tela de Login

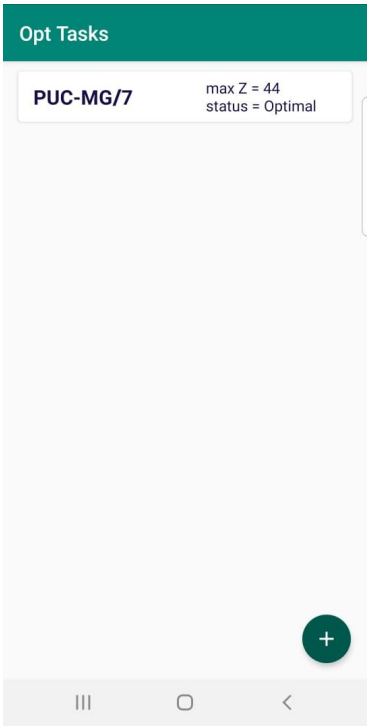

Figura 4. Lista de Simulações 1

Figura 5. Simulação - Alocações

6. Conclusão

Neste trabalho abordamos a otimização de alocação de tarefas em um determinado grupo de pessoas. Nele, cada pessoa possui uma certa competência c para cada

Figura 6. Simulação - Tarefas

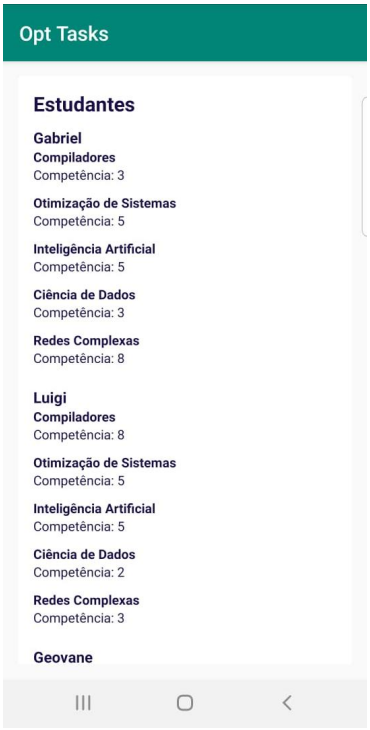

Figura 7. Simulação - Alunos

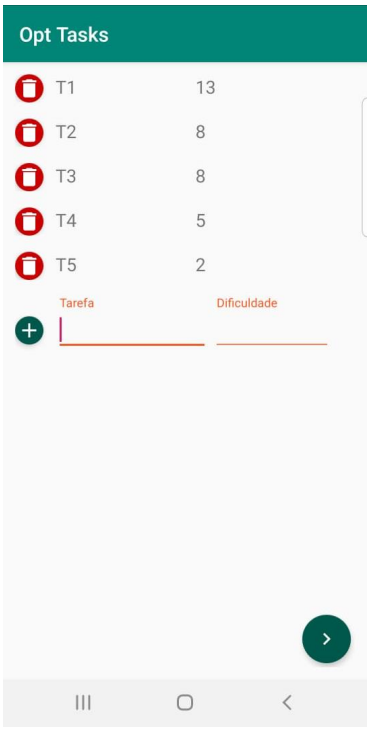

Figura 8. Tarefas - Adicionar

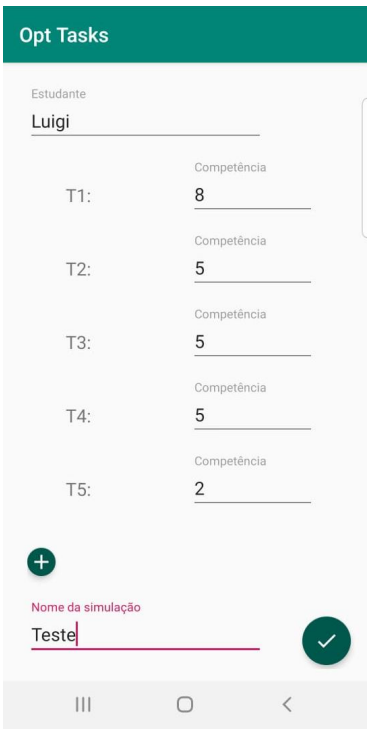

Figura 9. Alunos - Adicionar 1

Figura 10. Alunos - Adicionar 2

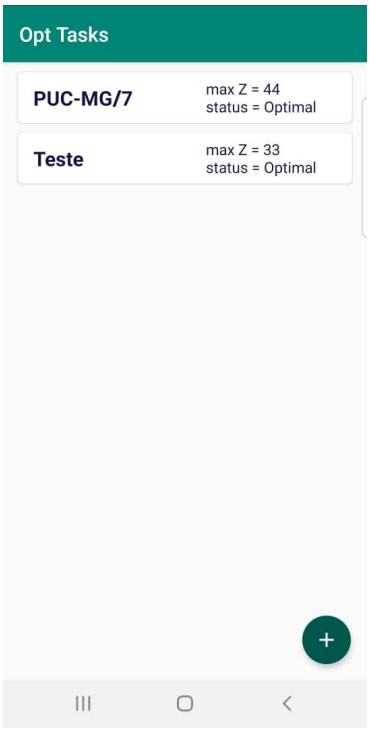

Figura 11. Lista de Simulações 2

atividade a que, por sua vez apresenta um nível de dificuldade d. Para solucionar este cenário, construímos um modelo matemático baseado nestes parâmetros. Ademais,

foi utilizado a abordagem de pesquisa operacional, uma vez que esse tipo de problema é caracterizado por ser de otimização.

Este trabalho foi importante para nosso aprofundamento neste tema, uma vez que explora o acompanhamento e análise de otimização de um processo existente. Além disso, nos permitiu aperfeiçoar competências de investigação, organização e aplicação da disciplina estudada.

7. Considerações Finais

O presente trabalho apresenta certas limitações. A primeira restrição do trabalho encontrada é em relação a aplicação Android. Nesta, não foi desenvolvida duas das etapas de operação CRUD. Nesse caso, foram feitas as etapas de criação e de leitura, faltando as de exclusão e de atualização da simulação.

O modelo dual também representa uma limitação, pois este não apresenta função objetivo e restrições claras. Além disso, os modelos dual e primal não convergem. Tendo isso em vista, é preciso realizar uma investigação mais aprofundada para extrair as possíveis informações que ainda não foram identificadas.

Outra limitação encontrada se refere à análise de sensibilidade do problema. Esta não é resolvida pelo GLPK usando o modelo proposto. Dessa forma, é preciso realizar o estudo do modelo e adaptá-lo de forma que ele possa gerar a análise de sensibilidade.

Por fim, há algumas oportunidades para trabalhos futuros. A análise de sensibilidade do modelo apresentado, bem como a portabilidade para outras plataformas, são exemplos para esta lacuna.

8. Links Úteis

- O projeto se encotra se encontra no seguinte repositório do *Github*, ou acesso pelo link: https://github.com/luigidcsoares/opt-tasks
- O servidor está rodando em núvem no *Heroku*, ou acesso pelo link: http://opttasks.herokuapp.com/
- O apk para a aplicação pode ser encontrada nas *releases* do repositório, ou acesso pelo link: https://github.com/luigidcsoares/opt-tasks/releases

Referências

[Alves 2019] Alves, D. R. (2019). Notas de aula em otimização de sistema.

[Eilon et al. 2012] Eilon, S., K. Holstein, W., L. Ackoff, R., and Tanenbaum, M. (2012). Operations research.

[Gosselin and Truchon 1986] Gosselin, K. and Truchon, M. (1986). Allocation of classrooms by linear programming. *Journal of the Operational Research Society*, 37(6):561–569.

[Hillier and Lieberman 2013] Hillier, F. and Lieberman, G. (2013). *Introdução à Pesquisa Operacional*. AMGH, Porto Alegre.

[Johnes 2015] Johnes, J. (2015). Operational research in education. *European Journal of Operational Research*, 243(3):683 – 696.

[Luenberger and Ye 2008] Luenberger, D. G. and Ye, Y. (2008). *Linear and Nonlinear Programming*. Springer, US.

[Maria Steiner 2006] Maria Steiner, Clarice Einhardt, D. B. E. A. V. C. Z. (2006). Técnicas da pesquisa operacional aplicadas à logística de atendimento aos usuários de uma rede de distribuição de energia elétrica. *Revista Eletrônica Sistemas & Gestão 1*, 1(3):229–243.

[Paula et al. 2012] Paula, P., Ferreira Sarmento, W. W., Sérgio, H., Pequeno, M., and Paillard, G. (2012). Support to decision making in the allocation of tutors in distance learning courses using integer programming. pages 1–4.