

Relatório final: modelagem

Convex-Lab*

Apresentação do Relatório

Este relatório detalha o ajuste e a expansão da estratégia de anteriormente apresentada, composta por um modelo matemático de otimização linear e um processamento interativo, para solução do problema da ONG Matemática em Movimento. Os novos resultados obtidos não revelaram inconsistências no modelo ou código, mesmo após testes com diferentes dados de entrada. A Seção 1 descreve brevemente o modelo melhorado implementado para a final. Para esta última etapa, implementamos algumas novas funcionalidades no modelo (Seção 2), além de resolver os problemas encontrados pela UniSoma e pela equipe (Seção 3). Também implementamos, em caráter experimental, uma página web em python para mostrar os principais resultados do modelo e ilustrar as soluções recomendadas (Seção 4).

1 Estratégia para solução do problema

A estratégia apresentada para solução do problema é composta por três fases. Uma nova fase inicial, chamada fase 0, foi implementada para a final. Essa fase é responsável por garantir a factibilidade do problema.

1.1 Fase 0 - melhoria implementada para a final

Na fase 0, primeiro verifica-se a possibilidade de se juntar turmas de alunos de continuidade. Em seguida, fecham-se as turmas existentes desnecessárias para alocar os alunos de continuidade.

Nessa etapa, verifica-se também se será necessária a abertura de novas turmas para alocar esses alunos. Nessa situação, o modelo permite a abertura de novas turmas apenas na fase 1 para alocar os alunos de continuidade, mesmo se essa opção for desabilitada nos parâmetros, a fim de garantir a factibilidade do modelo. Como os objetivos são maximizar a alocação de alunos nas turmas e minimizar o número de turmas, garante-se que apenas turmas necessárias para obrigatoriamente alocar os alunos de continuidade serão abertas. Nesse caso, como o parâmetro original não permitia a abertura de novas turmas, a fase 2 não é executada.

A fase 0, por fim, verifica se o orçamento atual da ONG é suficiente para alocar os alunos de continuidade. Se não for, o orçamento é ajustado. Note que o orçamento só é ajustado se forem esgotadas todas as possibilidades de juntar ou fechar turmas para garantir a factibilidade.

1.2 Fases 1 e 2

Na fase 1, resolve-se um modelo de otimização linear para alocar os alunos de continuidade e preencher as vagas das turmas pré-existentes. Para isso, o modelo minimiza o número de turmas e o número de vagas livres nessas turmas. Nessa primeira fase, o modelo garante que novas turmas são criadas apenas se forem necessárias para alocar os alunos de continuidade, de acordo com a função objetivo. Além disso, o modelo aloca alunos de formulário apenas para preencher essas vagas, de acordo com a ordem de prioridade de data de inscrição. O modelo de otimização linear completo, conforme descrito no relatório inicial, não foi alterado.

* Contato:

Camila Pereira dos Santos Tautenhain - camila.p.san@gmail.com

Dalton Garcia Borges de Souza - dalgarbor@gmail.com

Por fim, na fase 2, um procedimento iterativo é responsável por criar novas turmas para os alunos de formulário, de acordo com o orçamento da ONG. Os alunos de formulário que já foram inscritos na primeira fase são desconsiderados na segunda fase. Nessa etapa, consideram-se os critérios de desempate mais recentes propostos no desafio.

2 Novas funcionalidades e melhorias

As seguintes melhorias foram implementadas na versão final:

1. Obrigatória: Criado o relatório de turmas requisitado para a entrega final – tabela *sol_relatorio_turma*. Além disso, os seguintes novos relatórios foram criados:

- a *sol_formulario_nao_alocados*: lista de alunos de formulário que não puderam ser alocados.
- b *sol_relatorio_turma_lista_alunos*: relatório completo das turmas, com as listas de alunos alocados. Esse relatório é interessante para testar o modelo.
- c *sol_relatorio_geral*: relatório geral com o número total de turmas, alunos alocados e orçamento utilizado.
- d *sol_relatorio_orcamento*: orçamento por turma.
- e *sol_relatorio_serie*: números de turmas e alunos em cada série.
- f *sol_relatorio_escola*: números de turmas, alunos e séries em cada escola.

2. Obrigatória: Para a entrega final a regra de desempate foi revisada. Agora, caso as duas maiores demandas estejam dentro da tolerância estabelecida e sejam na mesma série – critérios de desempate não satisfeitos – há uma conferência para verificar se elas são da mesma escola antes de prosseguir com a escolha aleatória. Se forem na mesma escola, considera-se o desempate estabelecido na pergunta 10 do *checkpoint* 4: tenta-se escolher a escola com menos alunos, depois com menos turmas e só caso o empate continue seleciona-se aleatoriamente.

3. Foram criadas as tabelas *alerta_formulario_duplicado* e *alerta_aluno_duplicado* para indicar alunos de formulário e continuidade, respectivamente, duplicados no arquivo de entrada.

4. O limitante para o número de turmas da fase 1 foi melhor definido, a partir do cálculo do número mínimo de turmas necessárias na fase 0. Essa melhoria resultou em uma melhor eficiência computacional do modelo, devido à menor quantidade de variáveis.

3 Problemas resolvidos

Nessa entrega final, foram corrigidos os problemas apontados pela equipe do Desafio UniSoma:

1. Quando o modelo era otimizado entre anos, os alunos reprovados não eram alocados corretamente: esse problema foi corrigido na versão final. Agora, definem-se corretamente as séries dos alunos reprovados.

2. Em um dos cenários, o modelo era infactível e não obtinha nenhum resultado: na entrega inicial, o modelo era infactível quando o orçamento era insuficiente, quando a opção de possibilitar novas turmas era desativada e era necessário abrir mais turmas para alocar os alunos de continuidade ou quando o orçamento era insuficiente para abrir todas as turmas pre-existentes. No modelo final, essas situações foram corrigidas com a implementação de uma fase 0, na qual as turmas são ajustadas, antes da execução do modelo na fase 1. A fase 0 foi descrita na Seção 1.1.

Foi incluída uma tabela *sol_relatorio_parametro_ajuste* para indicar se os parâmetros *limite_custo* e *possibilita_abertura_novas_turmas* foram ajustados para permitir a factibilidade do problema. Observe que o orçamento só é ajustado se forem esgotadas todas as opções de junção e fechamento de turmas existentes.

No *cenario_5*, juntam-se as turmas 2 e 3 e as turmas 8 e 9. Além disso, as turmas 3 e 9 são fechadas. Apesar disso, ainda foi necessário aumentar o orçamento em R\$700,00. Nesse cenário, nenhum aluno de formulário foi alocado porque o orçamento foi ultrapassado.

O cenário *Modelo de Dados - MatMov - v1* agora é factível, pois turmas existentes desnecessárias são fechadas na fase 0.

O cenário *Modelo de Dados - MatMov - v3* agora é factível forçando a abertura de novas turmas para alocar os alunos de continuidade.

3. Eram abertas turmas desnecessariamente: na entrega inicial, um coeficiente na solução objetivo para priorizar os alunos de formulário de acordo com a ordem de inscrição foi mal estimado e, por conta disso, dominava as parcelas da função objetivo associadas a minimização de vagas livres e número de turmas abertas. Na versão final, estimamos esse valor corretamente. Além dessa correção, a fase 0 implementada para final, além de resolver a infactibilidade, permite juntar turmas e fechar turmas não existentes, de modo que turmas existentes não são desnecessariamente abertas.

Agora, apenas 10 turmas são abertas no cenário *cenário_2*. Essa correção também reduziu o número de turmas abertas nos outros cenários considerados.

4 Relatório Web

Paralelamente ao arquivo python do modelo (*convex-lab_final_modelo*), entrega obrigatória do desafio, um segundo arquivo python é apresentado (*convex-lab_final_modelo_interface*), que realiza a otimização do modelo e gera uma interface com o relatório web, a fim de facilitar na visualização das entradas e saídas do modelo. Ambos arquivos se encontram no mesmo diretório e compartilham do mesmo arquivo .db de entrada. O relatório web consta de 9 seções, sendo três descritivas dos dados de entrada e 6 descritivas dos dados de saída do modelo. Grande parte dos elementos do relatório são interativos, como as tabelas, que permitem ordenação e filtragem de valores, e os gráficos do tipo sunburst, que permite interação por click, e de barras.

As entradas são informações previamente fornecidas pela ONG e que podem alterar as saídas do modelo, de acordo com os valores atribuídos. As entradas são: (1) Lista de alunos de continuidade; (2) Lista de alunos de formulário inscritos/ (3) Parâmetros, Escolas, Regiões, Turmas e Séries. Todos descritos em tabelas separadas, porém em uma mesma janela do relatório web.

Já as saídas, boa parte na forma de relatórios, contêm informações provenientes da execução do modelo, construídas de acordo com recomendações do desafio e relatórios propostos pelo grupo. As saídas são:

- Relatório Geral: compila um conjunto de indicadores da solução recomendada, divididos em três grupos: Custos, Turmas e Alunos.
- Relatório de Turmas: mostra lista de turmas recomendadas e seus detalhamento quanto aos alunos. Um gráfico de barras é apresentado, o qual responde aos filtros na tabela ao qual é vinculado.
- Lista de alunos de continuidade alocados.
- Listas de alunos de formulário alocados e não alocados.
- Relatório descritivo de turmas: apresenta informações detalhadas das turmas e se as mesmas foram aprovadas ou não.
- Relatório descritivo dos custos: reúne informações de custo por turma.

Após execução do arquivo 2 (*convex-lab_final_modelo_interface*), o relatório web é gerado no navegador em **127.0.0.1:4050**. Uma prévia do relatório web para os dados de entrada disponibilizados pode ser encontrada [aqui](#), e pode ser acessado de qualquer dispositivo.

Para executar o relatório web, as seguintes bibliotecas são necessárias, na versão apresentada ou superior: (1) *setuptools*==50.3.2, (2) *dash*==1.16.0, (3) *dash-table*==4.10.1, (4) *dash-daq*==0.5.0, (5) *dash-bootstrap-components*==0.10.7, e (6) *plotly*==4.10.0

Informações mais detalhadas na leitura do relatório web são fornecidas ao se executar o relatório. Um segundo README (README-2) também é apresentado com o segundo arquivo, a fim de facilitar no processo de configuração das bibliotecas e execução do arquivo.