MO824A/MC859A
Tópicos em Otimização Combinatória
Relatório da Atividade 9
Equipe 7

Bruno Mendes Richau (RA 157743) [1]

Eduardo Barros Innarelli (RA 170161) [2]

Rogério Guimarães Meirelles (RA 160245) [3]

Resumo. Este documento é o relatório da equipe 7 referente à atividade 9 da disciplina MO824A/MC859A - Tópicos em Otimização Combinatória, oferecida no segundo semestre de 2020 pela Universidade Estadual de Campinas. A atividade consiste em um estudo de caso para um problema de atribuição de carga didática para professores, denominado problema de alocação de professores (PAP). Para solucionar o PAP, apresentaremos uma formulação de um modelo de programação linear para o solucionador Gurobi e uma implementação da metaheurística de Busca Tabu.

1. Descrição do problema

Considere uma instituição de ensino superior com P professores e D disciplinas a serem oferecidas em um semestre. O coordenador da instituição decidirá quais disciplinas e em quais períodos da semana cada professor deverá lecionar.

O coordenador decidiu adotar uma nova abordagem para o critério de alocação de disciplinas, baseada nos históricos de avaliação dos professores realizados pelos alunos nos semestres anteriores. Cada professor $p \in \{1, \dots, P\}$ possui uma avaliação $a_{pd} \in [0, 100]$ referente a uma disciplina $d \in \{1, \dots, D\}$. Se um docente p nunca ofereceu uma disciplina d, sua avaliação correspondente será $a_{pd} = 0$. Ao decidir sobre as alocações, o coordenador deve respeitar as seguintes restrições:

1. Uma disciplina pode estar alocada a no máximo um professor.

[1] B157743@dac.unicamp.br
[2] E170161@dac.unicamp.br
[3] R160245@dac.unicamp.br

2. Uma semana é constituída de um conjunto de $T = 20$ períodos ($t \in \{1, \ldots, T\}$). Cada dia é dividido em quatro períodos onde cada período corresponde a duas horas-aulas (8:00-10:00, 10:00-12:00, 14:00-16:00 e 16:00-18:00).

3. Cada disciplina d exige h_d períodos em uma semana ($h_d \in [1, 3]$).

4. O instituto possui S salas, portanto no máximo S disciplinas podem ser ministradas no mesmo período.

5. Cada professor p possui restrições de quais períodos não pode lecionar, se $r_{pt} = 1$ então o professor p pode lecionar no período t. Caso contrário, $r_{pt} = 0$.

6. Um professor pode estar alocado a mais de uma disciplina, no entanto a carga de disciplinas do professor deve somar no máximo $H = 3$ períodos (ou 6 horas-aula).

7. Um professor não pode oferecer duas disciplinas no mesmo período.

Além disso, o coordenador propõe que caso não seja possível alocar alguma disciplina, deve-se aplicar um penalidade de -100 na função objetivo. Sendo assim, os **dados de entrada do problema** são: $P, D, T, S, H, h_d, a_{pd}$ e r_{pt}

2. Modelo matemático

Com base no problema enunciado, temos que nosso objetivo é alocar os professores com a maior avaliação por disciplina possível, penalizando disciplinas não alocadas. Para isso precisaremos definir as seguintes variáveis:

1. $x_{p,d}$ indica se professor p é alocado a disciplina d.

2. y_d indica se a disciplina d é alocada a algum professor.

3. $z_{p,t}$ indica se professor p é alocado no período t.

Obtendo assim o seguinte modelo matemático de programação linear inteira do PAP:

$$\text{maximizar} \quad \sum_{d=1}^{D}\sum_{p=1}^{P} a_{p,d}x_{p,d} - 100\sum_{d=1}^{D}(1-y_d) \tag{2.1}$$

$$\text{sujeito a:} \sum_{p=1}^{P} x_{p,d} = y_d, \qquad \forall d \in \{1,\dots,D\} \tag{2.2}$$

$$\sum_{t=1}^{T} z_{p,t} = \sum_{d=1}^{D} h_d x_{p,d}, \qquad \forall p \in \{1,\dots,P\} \tag{2.3}$$

$$\sum_{p=1}^{P} z_{p,t} \leq S, \qquad \forall t \in \{1,\dots,T\} \tag{2.4}$$

$$z_{p,t} \leq r_{p,t}, \qquad \forall p \in \{1,\dots,P\},\ t \in \{1,\dots,T\} \tag{2.5}$$

$$\sum_{t=1}^{T} z_{p,t} \leq H, \qquad \forall p \in \{1,\dots,P\} \tag{2.6}$$

$$x_{p,d}, y_d, z_{p,t} \in \{0,1\}, \qquad \forall p \in \{1,\dots,P\},\ d \in \{1,\dots,D\},\ t \in \{1,\dots,T\} \tag{2.7}$$

Nossa função objetivo (2.1) maximiza os professores com a maior avaliação possíveis alocados para cada disciplina. Contudo, para garantirmos um resultado correto foram necessárias as seguintes restrições:

(2.2) Garante que uma disciplina pode estar alocada a no máximo um professor.

(2.3) O número de períodos que um professor trabalha deve corresponder ao número de períodos das disciplinas que ele oferece.

(2.4) O instituto possui S salas, portanto no máximo S professores podem trabalhar no mesmo período.

(2.5) Cada professor p possui restrições de quais períodos não pode lecionar, se $r_{pt} = 1$ então o professor p pode lecionar no período t. Caso contrário, $r_{pt} = 0$.

(2.7) Um professor pode estar alocado a mais de uma disciplina, no entanto a carga de disciplinas do professor deve somar no máximo H períodos.

3. Metaheurística Busca Tabu

A Busca Tabu é uma metaheurística baseada em busca local proposta por Glover em 1986 que surgiu da ideia de utilizar uma memória para superar ótimos locais. A técnica permite a realização de movimentos que não melhoram a solução atual, evitando ciclos por meio de memórias chamadas de listas tabu. Uma lista tabu é uma estrutura de dados responsável por armazenar os últimos movimentos aplicados por um determinado número de iterações, a fim de evitar que eles sejam desfeitos por novos movimentos. Assim, uma descrição básica do método consiste em construir uma solução inicial e realizar iterativamente um movimento de vizinhança que não seja proibido, guardando-o na lista tabu.

Em termos práticos, para adaptar a Busca Tabu a um problema de otimização combinatória, quatro elementos principais devem ser definidos: a representação da solução, a construção da solução inicial, a estrutura de vizinhança e o tamanho da lista tabu (indicando a quantidade de iterações durante as quais a realização de um movimento será proibida). Nesta seção será descrito como os elementos da busca tabu foram definidos para o PAP.

Os critérios de parada utilizados foram o número de iterações e o tempo de execução máximo.

3.1. Representação da Solução

Uma solução do PAP consiste de professores alocados a disciplinas em certos períodos. Um professor p alocado a uma disciplina d no período t pode ser representado pela tripla (p, d, t). É possível, assim, verificar se a inserção de uma tripla em uma solução corrente mantém a solução viável, o que é importante para a manipulação da lista de candidatos.

A dificuldade se encontra, principalmente, na restrição de igualdade 2.3, pois uma solução só é viável se a disciplina d for alocada h_d vezes. Para facilitar, relaxamos essa restrição, permitindo um menor número de alocações e aplicando uma penalização se for o caso. Note que uma solução vazia é viável nessa modelagem. Com a penalização, o método tende a completar uma solução parcial, até com que essa seja viável para o PLI.

3.2. Heurística Construtiva

A construção da solução inicial segue uma estratégia gulosa no menor custo de inserção do elemento na solução atual. O elemento de melhor performance que gera uma solução factível é escolhido a cada iteração da heurística, até que a inserção de mais um elemento piora a solução corrente. O custo de um elemento é calculado a partir da contribuição dele para a solução. Se mais de um elemento possuir o mesmo custo de inserção, a escolha é feita aleatoriamente.

3.3. Estratégia Tabu

Para implementar a lista tabu foi utilizada uma Double Ended Queue (DEQUE) de inteiros, i.e. uma fila com inserção e remoção de elementos nas duas extremidades. Ao realizar um novo movimento, o elemento que foi inserido ou removido da solução é adicionado na lista tabu e, com isso, todo movimento envolvendo esse elemento se torna proibido durante as próximas tenure iterações.

3.4. Movimentos de Vizinhança

A etapa de busca local consiste de um movimento de vizinhança na solução S construída anteriormente. Para o problema PAP, uma vizinhança imediata consiste nas soluções obtidas ao inserir ou remover um elemento em S, ou trocar um elemento de S por outro que não está na solução. A lista de candidatos é atualizada a cada iteração, para conter apenas elementos factíveis.

A exploração dessa vizinhança é feita através da estratégia best-improving, que trata de avaliar todos os movimentos possíveis a partir da solução atual, e escolher aquele que fornece a maior melhoria. Em outras palavras, todas as inserções, remoções e trocas são avaliadas, e o melhor movimento é escolhido.

4. Experimentos computacionais

Os experimentos foram realizados em uma máquina Linux Ubuntu 20.10, com uma CPU Intel(R) Core(TM) i5-8300H (4C/8T) com 2.30GHz de frequência do *clock* e 32GB de RAM operando com 2667 MHz. A modelagem foi implementada no Python 3.8.3 e resolvida com o Gurobi 9.0.3, enquanto a Busca Tabu foi implementada em Java 11 com o framework de Busca Tabu fornecido pelos docentes da disciplina. Sendo que o tempo máximo de execução adotado foi de 30 minutos e foram utilizadas as instâncias fornecidas na disciplina para os experimentos, que variam entre:

- Quantidade de professores P = 50, 70, 100
- Quantidade de disciplinas D = 50, 70, 100, 150
- Quantidade de salas S = 1, 3, 5, 6, 8, 10, 15, 20

5. Resultados

Os resultados obtidos pela resolução exata do PLI com o Gurobi encontram-se na tabela 1 abaixo, onde:

- *Valor* corresponde ao custo da solução encontrada;
- *Tempo* corresponde ao tempo em segundos para a solução ser encontrada;
- *Iterações* é a quantidade de iterações realizadas até encontrar a solução;
- *Variáveis* é a quantidade de variáveis no modelo utilizado;
- *Restrições* é a quantidade de restrições no modelo utilizado.

Analisando os resultados, podemos observar que:

Tabela 1: Experimentos utilizando o PAP PLI

Instância	**Valor**	**Tempo (s)**	**Iterações**	**Variáveis**	**Restrições**
P50D50S1	-1292	0.02	353	3550	1170
P50D50S3	2271	0.86	14670	3550	1170
P50D50S5	4770	0.30	2189	3550	1170
P70D70S1	-3058	0.04	470	6370	1630
P70D70S3	924	0.86	12044	6370	1630
P70D70S5	4265	0.22	4378	6370	1630
P70D100S6	4610	0.29	3531	8500	1660
P70D100S8	7177	4.18	46793	8500	1660
P70D100S10	9616	1.43	7611	8500	1660
P100D150S10	7626	39.07	264286	17150	2370
P100D150S15	13259	15.30	90702	17150	2370
P100D150S20	13259	8.42	32220	17150	2370

- As instâncias *P50D50S1* e *P70D70S1* não produziram solução viável, por isso sofreram a penalização inserida na função objetivo e o seu valor foi negativo. Isso ocorre devido ao número muito grande de professores e disciplinas para se alocar em uma única sala.

- Quado se aumenta o número de salas disponíveis, enquanto o número de professores e disciplinas segue inalterado, é possível viabilizar a solução. Podemos verificar esse fato comparando as instâncias *P50D50S1* e *P50D50S3*, e *P70D70S1* e *P70D70S3*. O valor da função objetivo aumenta conforme cresce o número de salas disponíveis.

- Para a maioria das instâncias, quando se aumenta o número de salas e se mantém o número de professores e disciplinas, o tempo de execução diminuiu. De todas as 12 instâncias verificadas, *P100D150S10* é a que teve execução mais demorada por uma grande margem. Porém, *P100D150S15* demorou menos da metade do tempo da instância anterior *P100D150S10*. Analogamente, *P100D150S20* também demorou aproximadamente metade do tempo da instância anterior *P100D150S15*.

- Comparando as instâncias *P100D150S15* e *P100D150S20*, percebemos que ambas obtiveram o mesmo valor da função objetivo, mesmo possuindo números de salas diferentes. Neste caso, entendemos que as 15 salas existentes em *P100D150S15* são suficientes para alocar os 100 professores às 150 disciplinas. Por isso, as 5 salas adicionais em *P100D150S20* não aprimoraram a solução.

Já os resultados obtidos pela resolução da busca tabu encontram-se na tabela 2 abaixo:

Ao contrário do método exato, a Busca Tabu não alcançou resultados satisfatórios. Conforme descrito na seção 3.1, um elemento da solução consiste de uma tripla

Tabela 2: Experimentos Utilizando a Busca Tabu

Instância	**Valor**	**Tempo (s)**	**Iterações**
P50D50S1	-1477	3.73	29
P50D50S3	1100	118.65	105
P50D50S5	1892	252.42	154
P70D70S1	-3064	6.59	1
P70D70S3	522	302.16	113
P70D70S5	3322	907.08	379
P70D100S6	98	1271.72	97
P70D100S8	-300	1361.22	84
P70D100S10	-300	1387.06	84
P100D150S10	-5065	1790.72	25
P100D150S15	-5265	1657.22	22
P100D150S20	-5065	1796.42	25

(p, d, t) com o professor p, a disciplina d em que está alocado e o período t em que oferece a disciplina. Logo, podem haver até $P \cdot D \cdot T$ elementos candidatos a entrar na solução corrente. Considerando que a cada iteração da busca a lista de candidatos é atualizada, torna-se inviável executar muitas iterações nas instâncias maiores, o que justifica os custos negativos que o algoritmo alcançou nessas instâncias.

Até a instância *P70D70S5* observa-se que o método consegue encontrar uma solução com um custo razoável, ainda que pior que o ótimo. Isso indica que há espaço para otimização, seja por meio do uso de estratégias alternativas ou principalmente pela adoção de uma abordagem diferente na representação da solução, que diminuísse a quantidade de candidatos. Otimizar a metaheurística poderia ser o próximo passo deste trabalho, para fins de aprendizado. Contudo, levando em conta os resultados obtidos e que o PAP é um problema que dificilmente exigiria instâncias maiores do que as disponibilizadas, é muito pouco provável que a Busca Tabu ou mesmo outro método apresente algum ganho em relação a resolução exata do PAP.

Referências

[1] M. Gendreau and J.-Y. Potvin, "Tabu search," in Handbook of Metaheuristics (M. Gendreau and J.-Y. Potvin, eds.), vol. 146 of International Series in Operations Research & Management Science, ch. 2, pp. 41–59, Springer Science+Business Media, 2010.

[2] F. Glover, M. Laguna, P. Pardalos, D.-Z. Du, and R. Graham, "Tabu search: effective strategies for hard problems in analytics and computational science," Handbook of Combinatorial Optimization, 2nd ed, vol. XXI, pp. 3261–3362, 01 2013.

[3] M. Gendreau, A. Hertz, and G. Laporte, “A tabu search heuristic for the vehicle routing problem,” Mgmt Sci, vol. 40, pp. 1276–1290, 10 1994.