Traduzido do Inglês para o Português - www.onlinedoctranslator.com

Revista Europeia de Pesquisa Operacional 256 (2017) 685–695

Listas de conteúdo disponíveis emCiência Direta

Revista Europeia de Pesquisa Operacional

página inicial do periódico:www.elsevier.com/locate/ejor

Otimização Discreta

Uma abordagem de geração de colunas para horários de ensino médio modelada como um problema de fluxo de multimercadorias

Árton P. Dorneles[um],[*], Olinto CB de Araújo[b], Luciana S. Buriol[um]

[um]Instituto de Informática, Universidade Federal do Rio Grande do Sul, 91501-970 Porto Alegre, RS, Brasil
[b]Colégio Técnico Industrial de Santa Maria, Universidade Federal de Santa Maria, 97105-900 Santa Maria, RS, Brasil

informação do artigo

Histórico do artigo:
Recebido em 19 de agosto de 2015
Aceito em 2 de julho de 2016 Disponível online em 7 de julho de 2016

Palavras-chave:
Horários
MILP
Geração de colunas
Fluxo multicommodity

resumo

O horário escolar é um problema clássico de otimização que tem sido extensivamente estudado devido à sua importância prática e teórica. Consiste em agendar um conjunto de reuniões de classe-professor em um período de tempo predeterminado, satisfazendo requisitos de diferentes tipos. Dada a natureza combinatória deste problema, resolver instâncias médias e grandes de horário para a otimalidade é uma tarefa desafiadora. Quando os recursos são escassos, muitas vezes é difícil encontrar até mesmo uma solução viável. Várias técnicas foram desenvolvidas na literatura para lidar com o problema de horário do ensino médio. Como o uso de métodos exatos, como técnicas de programação matemática, é considerado impraticável para resolver grandes instâncias do mundo real, metaheurísticas e metaheurísticas híbridas são as abordagens de solução mais utilizadas. Neste artigo, propomos um modelo de fluxo multicommodity para o problema de horário do ensino médio. Além disso, aplicamos a decomposição de Dantzig-Wolfe ao modelo proposto, propomos um algoritmo de geração de colunas e apresentamos resultados experimentais em instâncias bem conhecidas do problema. Os resultados mostram que os limites inferiores obtidos por meio de nossa abordagem são estreitos e podem ser gerados mais rapidamente do que abordagens anteriores relatadas na literatura.

© 2016 Elsevier BV Todos os direitos reservados.

1. Introdução

Uma tarefa comum a todas as instituições educacionais é fornecer uma atribuição de aulas que combine professores, alunos, salas e períodos (ou intervalos de tempo) para atingir um cronograma viável, satisfazendo requisitos de natureza diferente.

Normalmente, os requisitos são separados em*duro*e*macio*ones. Por requisitos rígidos queremos dizer aqueles que devem ser satisfeitos, enquanto requisitos suaves podem ser violados, mas devem ser satisfeitos sempre que possível. Requisitos suaves podem ter diferentes níveis de importância e são frequentemente conflitantes entre si, de modo que pode ser impossível satisfazer todos eles ao mesmo tempo. Normalmente, a qualidade de uma solução está associada diretamente à satisfação dos requisitos suaves. Quanto mais requisitos suaves são satisfeitos, o que pode ser contabilizado de forma diferente de acordo com o nível a que cada um pertence, melhor uma solução é considerada.

A qualidade é um atributo crítico da solução porque, uma vez estabelecido o cronograma, ele determinará o uso dos recursos físicos e a rotina diária de várias pessoas, possivelmente milhares, por um longo período que geralmente é de cerca de um ano. Devido à repetição, até mesmo questões menores podem se transformar em grandes problemas no decorrer do tempo, afetando diretamente a qualidade do trabalho dos professores, o aprendizado dos alunos e as despesas da instituição.

Os problemas de horários educacionais têm muitas variantes propostas na literatura, e o conjunto de objetivos e requisitos depende principalmente do contexto da aplicação, da instituição e do local onde ela está localizada (Drexl e Salewski, 1997; Post et al., 2014). Embora existam diversas variantes, os problemas de horários educacionais são comumente compreendidos em três classes: horários escolares, horários de cursos e horários de exames (Schaerf, 1999). Tanto no horário escolar quanto no horário de curso, o objetivo é construir um cronograma semanal. No entanto, no horário escolar, um conjunto de aulas deve ser atribuído a intervalos de tempo, enquanto no horário de curso, um conjunto de cursos universitários deve ser programado, evitando sobreposições de cursos com alunos comuns. Finalmente, no problema do exame, um conjunto de exames deve ser distribuído em um horizonte de tempo, evitando sobreposições para os alunos.

No nosso estudo atual, focamos no problema do horário escolar. Este problema apareceu pela primeira vez na literatura científica na década de 60 (Gotlieb, 1963) e desde então tem ganhado cada vez mais atenção. A variante mais básica do problema é programar um conjunto de eventos (ou reuniões) de classe-professor de tal forma que nenhum professor (nem classe) seja necessário em mais de uma aula por vez. Isso

*Autor correspondente.
*Endereços de e-mail:*arton.dorneles@inf.ufrgs.br,artondorneles@gmail.com (Á.P. Dorneles),olinto@ctism.ufsm.br (OCB de Araújo),buriol@inf.ufrgs.br (LS Buriol).

http://dx.doi.org/10.1016/j.ejor.2016.07.002 0377-2217/©
2016 Elsevier BV Todos os direitos reservados.

O problema básico pode ser resolvido em tempo polinomial por um algoritmo de fluxo de rede de custo mínimo (de Werra, 1971). No entanto, em aplicações do mundo real, os professores podem estar indisponíveis em alguns períodos. Se essa restrição for levada em conta, o problema de horário resultante é NP-completo (Mesmo, Itai e Shamir, 1975).

Na verdade, os problemas de cronograma mais reais vêm à tona como problemas de otimização combinatória que se enquadram na classe NPhard. Por esse motivo, pesquisadores ao redor do mundo investigaram esses problemas e várias técnicas diferentes foram desenvolvidas. Como o uso de métodos exatos, como técnicas de programação matemática, são considerados impraticáveis para resolver instâncias médias e grandes do mundo real, metaheurísticas e metaheurísticas híbridas são as abordagens de solução mais usadas. Para uma visão geral atualizada sobre essas abordagens, encaminhamos o leitor à pesquisa porPillay (2014).

Embora os grupos de pesquisa compartilhem interesses semelhantes, antes de 2011, a maioria das publicações na literatura relatava resultados focados na resolução de variantes de problemas específicos de seus países com casos de teste dependentes de aplicação (frequentemente indisponíveis). Essa metodologia, historicamente, dificultava a comparação de resultados entre diferentes abordagens de solução (Schaerf e Gaspero, 2001). Como uma tentativa de superar esses problemas, ao longo das últimas edições da Conferência Internacional sobre a Prática e Teoria do Horário Automatizado (PATAT), um grupo de pesquisadores de horários do ensino médio desenvolveu um formato baseado em XML, chamado XHSTT (Post e outros, 2012; 2014), para expressar problemas de diferentes países de forma unificada e tem sido amplamente aceito pela comunidade de pesquisa. Recentemente, seu uso foi promovido na Terceira Competição Internacional de Horários (ITC-2011) que executa diferentes métodos sobre instâncias originadas de vários países (Post, Gaspero, Kingston, McCollum e Schaerf, 2016). Problemas que podem ser representados no formato XHSTT são normalmente chamados de Problema Generalizado de Horários do Ensino Médio, doravante denominado GHSTP.

Desde o ITC-2011, várias abordagens heurísticas foram desenvolvidas para resolver o GHSTP. No entanto, nenhum método conhecido é capaz de encontrar soluções exatas para instâncias não triviais do problema em um tempo razoável.Kristiansen, Sørensen e Stidsen (2014)uma formulação de programação inteira foi proposta para o GHSTP, que fornece a melhor abordagem exata para o problema até onde sabemos. Do conjunto de 38 instâncias, para 10 delas uma solução com custo zero já é conhecida, o que define uma solução ótima pela definição de XHSTT. Gurobi foi usado para executar as outras 28 instâncias, e apenas quatro pequenas instâncias foram resolvidas dentro do limite de tempo de 86.400 segundos (1 dia) por execução. Nenhum método exato baseado em limites inferiores foi proposto até agora especificamente adaptado para resolver o GHSTP. Além de estimar a qualidade das soluções heurísticas, um bom limite inferior é um requisito básico para implementar procedimentos eficientes baseados em branch-and-bound.

Neste artigo, em vez de abordar todo o GHSTP, focamos no CTTPCR (Class-Teacher Timetabling Problem with Compactness Requirements), que é um subproblema do GHSTP que modela uma escola brasileira típica e compreende todas as instâncias brasileiras consideradas no ITC-2011. Em contraste com problemas originados da maioria dos outros países, o CTTPCR tem um espaço de busca mais restrito, já que todos os professores são pré-atribuídos a cada evento, e a atribuição de salas não é necessária.

A principal contribuição do nosso trabalho é dupla. Diferentemente dos modelos tradicionais de horários do ensino médio, propomos um modelo de fluxo multicommodity para o CTTPCR. Além disso, também propomos um algoritmo de geração de colunas para resolver o relaxamento linear da decomposição de Dantzig–Wolfe aplicado no modelo de fluxo multicommodity para fornecer limites inferiores fortes para o problema rapidamente. Esses novos limites inferiores são importantes para avaliar a qualidade das soluções obtidas com aplicações heurísticas.

abordagens para o CTTPCR, bem como para métodos projetados para o GHSTP. Além de ser mais rápido e simples do que abordagens anteriores publicadas na literatura, nossa geração de colunas escala melhor para instâncias grandes. Além disso, novos limites melhores foram fornecidos para todas as cinco instâncias do CTTPCR da primeira rodada do ITC-2011. Ao usá-los, fomos capazes de provar a otimalidade da solução de duas instâncias abertas desta competição.

O restante deste artigo está organizado da seguinte forma. Seção 2 apresenta alguns trabalhos relacionados no CTTPCR. Seção 3apresenta formalmente o problema abordado neste estudo, a notação usada para representá-lo e uma formulação de programação inteira mista.Seção 4 apresenta uma decomposição de Dantzig-Wolf para o problema, um algoritmo de geração de colunas para resolvê-lo, bem como duas estratégias de aceleração.Seção 5apresenta resultados experimentais considerando 12 instâncias do mundo real em comparação com abordagens anteriores. Finalmente,Seção 6apresenta nossas principais conclusões e um esboço do trabalho futuro.

2. Trabalhos relacionados

O CTTPCR foi definido pela primeira vez porSouza e Maculan (2000)que propôs a primeira formulação MIP para ele, bem como um conjunto de instâncias que se tornou um testbed básico usado até hoje. Eles mostraram que resolver as instâncias do testbed com um solucionador MIP de propósito geral era impraticável naquela época.

Do lado heurístico,Souza, Ochi e Maculan (2003)propôs uma metaheurística híbrida, chamada GTS-II, para resolver instâncias do CTTPCR. O GTS-II usa uma heurística construtiva randomizada gananciosa para construir uma solução inicial que depois é refinada por uma busca tabu. Como sua busca tabu também inclui soluções inviáveis no espaço de busca, ela foi equipada com um procedimento que é invocado eventualmente em uma tentativa de restaurar a viabilidade da solução atual. No trabalho deSantos, Ochi e Souza (2005) uma busca tabu com estratégias de diversificação foi proposta para resolver o CTTPCR. Seus experimentos mostraram que a busca tabu proposta superou significativamente o GTS-II. Além disso, os autores mostraram empiricamente que a estratégia de diversificação proposta pode melhorar a robustez da busca tabu.

Santos, Uchoa, Ochi e Maculan (2012)propôs novos modelos MIP, bem como um algoritmo de geração de corte e coluna (CCG) usando cortes de Fenchel (Boyd, 1994), fornecendo, pela primeira vez, limites inferiores fortes para instâncias CTTPCR. Esse trabalho é considerado um marco porque estabeleceu uma base confiável para avaliar a qualidade de soluções heurísticas.

Recentemente, propusemos um novo modelo MIP para o CTTPCR (Dorneles, Araújo e Buriol, 2012) e desenvolveu uma abordagem híbrida que combina uma heurística de correção e otimização com um procedimento de descida de vizinhança variável (Dorneles, Araújo e Buriol, 2014). Fomos capazes de provar a otimalidade da solução para a maioria das instâncias testadas e, até onde sabemos, a metodologia apresentada fornece resultados de última geração para o problema.

3. Definição e modelagem do problema

Nesta seção, introduzimos uma nova formulação compacta MIP para o Problema de Horários Classe-Professor com Requisitos de Compactidade (CTTPCR). O problema considera um conjunto de classes$C$e um conjunto de professores$E$. Uma classe$c \in C$é um grupo de alunos que seguem o mesmo curso e têm disponibilidade total. O objetivo do problema é construir um cronograma para uma semana que geralmente é organizado como um conjunto de dias$E$, e cada dia é dividido em um conjunto de períodos$P$Nós chamamos de *intervalo de tempo*um par composto por um dia e um período de aula, (e,p), com$e \in E$e$p \in P$, onde todos os períodos têm a mesma duração. Professores *para*$\in E$pode não estar disponível em alguns horários.

A entrada principal para o problema é um conjunto de eventos que devem ser agendados. Normalmente, um evento é uma reunião entre um professor*para* e uma classe *c* para abordar um assunto específico em uma sala específica. Neste artigo, denotamos um evento por um par (*para*, *c*). O parâmetro O_{tc} determina o *carga de trabalho* de um evento (*para*, *c*), ou seja, o número de aulas que devem ser ministradas pelo professor *para* para a classe *c*. Além disso, cada evento define como as aulas serão distribuídas em uma semana, solicitando uma quantidade de aulas duplas, restringindo o número de aulas por dia e definindo se as aulas ministradas em um mesmo dia devem ser consecutivas.

UM *viável* o horário tem um intervalo de tempo atribuído a cada lição de eventos que satisfazem os requisitos rígidos H1–H6 abaixo:

H1 A carga de trabalho de cada evento deve ser satisfeita.
H2 Um professor não pode ser escalado para mais de uma aula em um determinado período.
As aulas H3 não podem ser ministradas para a mesma turma na mesma turma. riod.
H4 Um professor não pode ser escalado para um período em que ele/ela não está disponível.
H5 O número máximo de aulas diárias de cada evento deve ser atendido.
H6 Duas aulas do mesmo evento devem ser consecutivas quando agendado para o mesmo dia, caso seja necessário pelo evento.

Além da viabilidade em relação às restrições rígidas, o maior número possível dos requisitos flexíveis S1–S3 declarados abaixo deve ser satisfeito:

S1 Evite professores *períodos ociosos*. Um período de um professor é considerado-considerado ocioso se ele/ela tiver aulas atribuídas antes e depois desse período no mesmo dia.
S2 Minimize o número de *dias úteis* para professores. Neste contexto, dia útil significa um dia em que o professor tem pelo menos uma aula atribuída a ele/ela.
S3 Forneça o número de aulas duplas solicitadas por cada evento.

Propomos um modelo de fluxo multicommodity para o CTTPCR no qual os arcos representam transições de períodos de tempo em um gráfico de rede específico. Esta abordagem foi inspirada no trabalho de Steinzen, Gintner, Suhl e Kliewer (2010) onde os autores também usam arcos para representar transições de tempo para o problema de programação integrada de veículos e equipes com vários depósitos. No modelo, cada professor é representado por uma mercadoria. Isso significa que determinar a programação de um professor é o mesmo que encontrar um caminho em um gráfico de rede apropriado. Formalmente, representamos essa rede como um gráfico acíclico direcionado $G=(V, Um)$, onde V é um conjunto de nós e UM é um conjunto de arcos. Embora todas as commodities compartilhem o mesmo conjunto de nós, incluindo o mesmo *fonte* e *afundar* nós, cada mercadoria considera apenas um determinado subconjunto de arcos $UM_{para} \subseteq UM$. Figura 1 apresenta uma ilustração do gráfico G onde todos os tipos de arcos são mostrados para uma determinada mercadoria *para*. A figura é composta de dias (retângulos verticais arredondados) e períodos de um dia (faixas horizontais). Cada bloco de dia tem retângulos verticais sombreados relacionados às atividades de cada classe (duas classes são consideradas no exemplo). A seguir, descrevemos os tipos de arcos de G:

- *Arcos de aula* são usados para indicar quais intervalos de tempo são atribuídos a um determinado professor e classe. Os arcos de lições são geralmente compartilhados entre commodities e têm uma capacidade unitária associada para garantir que sejam usados apenas por uma única commodity (professor) por vez. Além disso, para cada arco de lição $um \in UM_{para}$ está associado um rótulo S_{ta} que representa a duração (em períodos) da lição representada pelo arco. Os arcos de lição são chamados de *arcos de aula única* quando $S_{ta}=1$, e como *arcos de aula dupla* quando $S_{ta}=2$. Na figura, os arcos de aula são todos arcos de curva dentro de um bloco de dia e dentro do retângulo sombreado relacionado a alguma aula.
- C_{para} é o conjunto de *arcos de período ocioso*. Esses arcos são usados para identificar os períodos ociosos de cada professor *para*. Um custo ω está associado para cada arco de período ocioso. Na figura, os arcos de período ocioso são todos arcos retos dentro de um bloco de dia, fora dos retângulos sombreados relacionados às classes.
- Conjuntos P_q *para* e $Pq+$ *para* são conjuntos de arcos auxiliares chamados, respectivamente, de *puxar para dentro* e *retirar* arcos. Enquanto arcos pull-in são todos arcos que entram em um bloco de dia, arcos pull-out são aqueles que saem de um bloco de dia.
- E_{para} é o conjunto de *arcos de dia de trabalho*. Esses arcos são usados para calcular o número de dias úteis de um professor *para*. Um custo *gama* está associado a cada arco de dia útil. Na figura, para cada dia, o nó principal do arco de dia útil corresponde ao nó de cauda de cada arco pull-in para aquele dia.
- B_{para} é o conjunto de *arcos de folga*. Esses arcos são usados quando um professor *para* não ensina nenhuma lição em um determinado dia. Esses são os arcos localizados na base inferior da figura. Seus nós de cauda são os mesmos dos arcos de dias úteis.

Cada caminho na rede é composto por um fluxo binário denotado pela variável x_{ta}, onde $para \in E$ e $um \in UM_{para}$. Cada caminho começa no nó de origem, alterna por diferentes tipos de arcos, terminando no nó de destino, conforme mostrado na Figura 2.

A seguir, apresentamos uma formulação de programação linear inteira mista para o CTTPCR doravante denotado como F_1. Por conveniência, a notação completa usada na formulação é apresentada em Tabela 1.

$$\textbf{Minimizar} \quad \sum_{para \in E} \sum_{c \in C} \left(\delta g_{tc} + \sum_{um \in C_{para}} \omega x_{ta} + \sum_{um \in E_{para}} gama\, x_{ta} \right) \tag{1}$$

Sujeito a

$$\sum_{um \in UM^{+}_{para\,você}} x_{ta} - \sum_{um \in UM^{-}_{para\,você}} x_{ta} = b_{você} \quad \forall para \in T, você \in V \tag{2}$$

$$\sum_{para \in T} \sum_{um \in UM_{tcdp}} x_{ta} \leq 1 \quad \forall c \in C d \in E,\ p \in P \tag{3}$$

$$\sum_{um \in \bigcup_{e \in D, p \in P} UM_{tcdp}} S_{ta} x_{ta} = O_{tc} \quad \forall para \in V,\ C \in C \tag{4}$$

$$\sum_{um \in \bigcup_{p \in P} UM_{tcdp}} S_{ta} x_{ta} \leq eu_{tc} \quad \forall para \in V,\ C \in C d \in E \tag{5}$$

$$\sum_{um \in \bigcup_{p \in P} UM_{tcdp}} x_{ta} \leq 1 \quad \forall para \in V,\ C \in C d \in E,\ h=1 \tag{6}$$

$$g_{tc} \geq M_{tc} - \sum_{um \in G_{tc}} x_{ta} \quad \forall para \in V,\ C \in C \tag{7}$$

$$\sum_{um \in E_{para}} x_{ta} \geq E_{para}, \quad \forall para \in E \tag{8}$$

$$x_{ta} \in \{0,1\} \quad \forall para \in T,\ um \in UM_{para} \tag{9}$$

$$g_{tc} \geq 0 \ \forall para \in V,\ C \in C \tag{10}$$

A função objetivo minimiza a violação de restrições suaves. O conjunto de restrições de conservação de fluxo (2) garante que o fluxo total de entrada seja igual ao fluxo total de saída de cada nó (exceto fonte e sumidouro), considerando uma determinada mercadoria *para*. Conjunto de restrições (3) garante que a capacidade unitária dos arcos de aula seja respeitada. Pode-se notar que uma estrutura de um problema de fluxo de multicommodities é representada pelos conjuntos de restrições (2) e (3) e pelas duas primeiras partes da função objetivo (1). Esta estrutura só é capaz de abordar os requisitos H2, H3, H4, S1 e S2. Para modelar os requisitos restantes, incluímos restrições adicionais e o último componente da função objetivo. Conjunto de restrições (4) garante que a carga de trabalho de cada evento seja atendida. Conjunto de restrições (5) garante que o número máximo de aulas diárias para cada evento

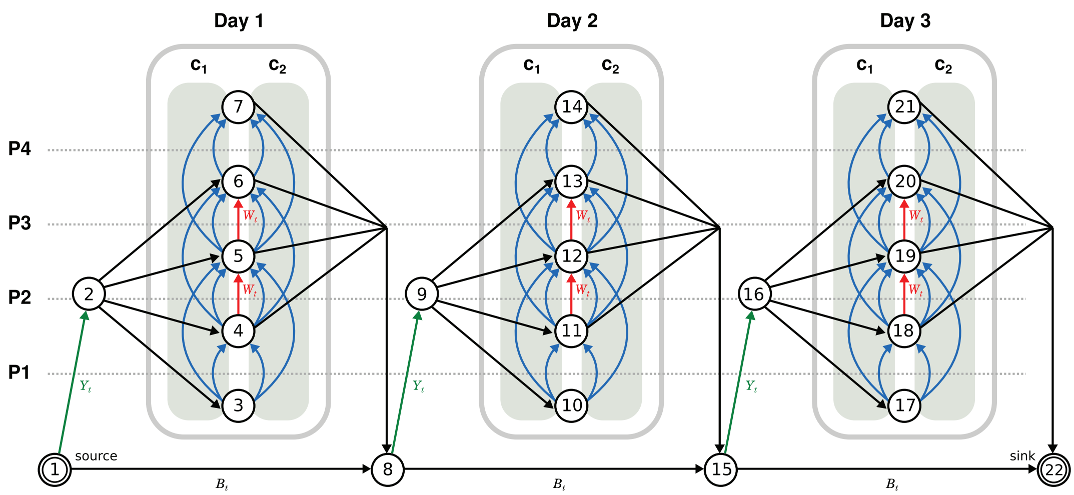

Figura 1. Exemplo de um grafo de rede em uma instância de brinquedo composta por três dias, quatro períodos por dia (P1, P2, P3, P4) e duas classes (c_1, c_2). Cada dia da semana é representado por um retângulo arredondado onde os arcos de aula e os arcos de período ocioso estão localizados. Dentro de cada um, os arcos de aula aparecem em dois grupos representados por um retângulo sombreado, onde cada grupo representa os arcos de aula para as classesc_1ec_2.

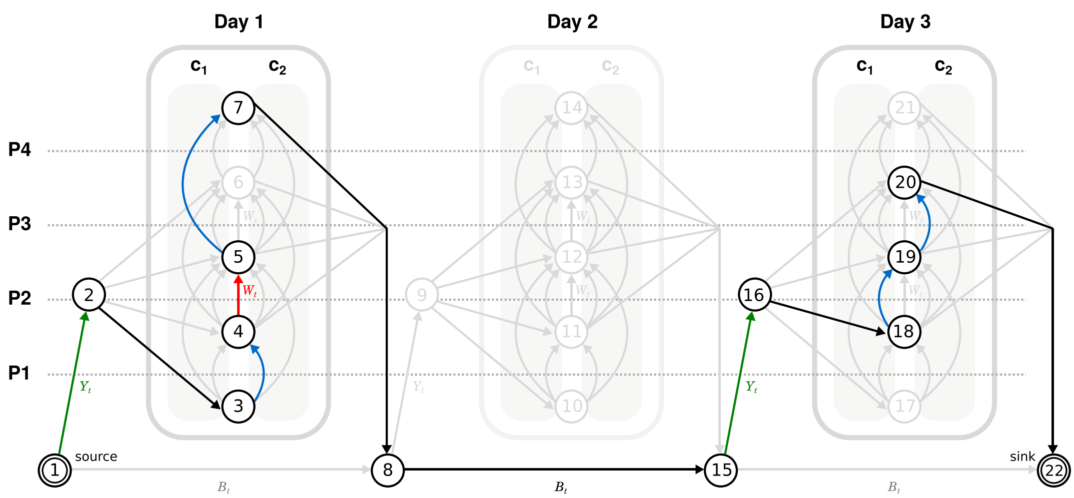

Figura 2. Exemplo de cronograma viável para um professor*para*representado por um caminho na rede. Neste exemplo, um professor trabalha apenas nos dias 1 e 3. No dia 1, ele/ela ensina uma única aula para a classe$c_2$no período P1, fica ocioso no período P2, e depois dá aula dupla a partir do período P3 para a turmac_1. No dia 3, ele/ela ensina uma única aula para a turma$c_1$no período P2 e outro para a turma$c_2$no período P3.

é satisfeito. Conjunto de restrições(6)garante que as aulas do mesmo evento sejam programadas em sequência, permitindo o uso de apenas um arco por dia. Essa restrição só é ativada quando$o$=1. Conjunto de restrições(7)calcula o número de aulas duplas que ocorrem na solução. Conjunto de restrições(8)estabelece um limite inferior para o número mínimo de dias úteis para cada professor. Este limite inferior foi proposto porSouza (2000)e é calculado de acordo comEquação (11).

$$E_{para} = \text{máx.}\left\{\left\lceil \frac{\sum_{para \in E}\sum_{c \in C} O_{tc}}{|P|}\right\rceil, \text{máx.}_{para \in T, c \in C}\left\{\left\lceil \frac{O_{tc}}{eu_{tc}}\right\rceil\right\}\right\} \quad \forall para \in E \qquad (11)$$

Ao usarE'_{para}pode-se obter uma solução simples e computacionalmente barata limite inferior (LB$gama$)para o CTTPCR conforme definido emEquação (12).

$$LIBRA_{gama} = \sum_{para \in E} E_{para}gama \qquad (12)$$

3.1. Cortes adicionais

Embora a formulaçãoF_1é adequado para representar o CTTPCR, devido à estrutura da rede, eventualmente permite a construção de *caminhos sem sentido*que superestimam o custo de soluções subótimas.Figura 3ilustra três casos em que caminhos sem sentido podem ocorrer. No Caso 1, o caminho do fluxo atravessa o componente do dia usando um único nó. Como o caminho não contém nenhum arco de lição, o arco do dia útil é usado desnecessariamente para acessar o componente do dia. No Caso 2, o custo da solução é superestimado porque um arco de período ocioso é mal utilizado. Idealmente, um arco de período ocioso não deve aparecer em um caminho ao suceder um arco de entrada ou preceder um arco de saída. No Caso 3, duas lições únicas são ensinadas em sequência para a mesma classe, enquanto usar um arco de lição dupla seria mais apropriado. Essa situação só pode ocorrer quando o requisito H6 não está sendo considerado, ou seja, o=0. Caso contrário, já é evitado pelo conjunto de restrições(6). Em para evitar esses casos podemos fortalecer nossa formulação com a adição de alguns conjuntos de restrições de corte válidos(13)–(16). Conjunto de restrições(13) proíbe o Caso 1, conjuntos de restrições(14)e(15)proibir Caso 2, e conjunto de restrições(16)proíbe o Caso 3.

$$x_{você} + x_{tj} \le 1 \quad \forall para \in T, você \in V, eu \in Pq_{para}^{-} \cap UM_{paravocê}, eu \in Pq_{para}^{+} \cap UM_{paravocê} \qquad (13)$$

$$x_{você} + x_{tj} \le 1 \quad \forall para \in T, você \in V, eu \in Pq_{para}^{-} \cap UM_{paravocê}, eu \in C_{para} \cap UM_{paravocê}^{+} \qquad (14)$$

$$x_{você} + x_{tj} \le 1 \quad \forall para \in T, você \in V, eu \in Pq_{para}^{+} \cap UM_{paravocê}, eu \in C_{para} \cap UM_{paravocê}^{-} \qquad (15)$$

$$x_{você} + x_{tj} \le 1 \quad \forall para \in V, C \in C d \in E, p \in P, eu \in UM_{tcdp}, eu \in UM_{tcdp+1}, M_{tc} > 0, p < |P|, S_{você} = S_{tj} = 1, o = 0 \qquad (16)$$

Tabela 1
Notação usada na formulação compacta F_1.

Símbolo	Definição
Conjuntos	
$e \in E$	Dias da semana.
$p \in P$	Períodos de um dia.
$para \in E$	Conjunto de professores (commodities).
$c \in C$	Conjunto de classes.
$você \in V$	Conjunto de todos os nós.
$um \in UM_{para}$	Conjunto de todos os arcos da mercadoria *para* ($UM_{para} \subseteq UM$).
$um \in UM_{tcdp}$	Conjunto de arcos de lições da mercadoria *para* na aula *c*, dia *e*, e período *p*. Conjunto de todos os arcos do nó de
$um \in UM^{-}_{paravocê}$	entrada *você* para mercadoria *para*. Conjunto de todos os arcos do nó de saída *você* para mercadoria *para*.
$um \in UM^{+}_{paravocê}$	Conjunto de todos os arcos de pull-in para commodities *para*. Conjunto de todos os arcos de extração para
$um \in Pq_{para}$	mercadoria *para*. Conjunto de todos os arcos de jornada de trabalho do professor *para*. Conjunto de todos os
$um \in Pq_{para}$	arcos de períodos ociosos do professor *para*.
$um \in E_{para}$	
$um \in C_{para}$	
$um \in G_{tc}$	Conjunto de todos os arcos de aula dupla do professor *para* e classe *c*.
Parâmetros	
$b_{você}$	Tem valor 1 quando *você* é a fonte, −1 quando *você* é a pia, caso contrário 0.
$O_{tc} \in$ Não	Número de aulas que o professor *para* deve ensinar para a classe *c*. Número
$eu_{tc} \in \{1, 2\}$	máximo diário de aulas que o professor *para* pode ensinar para a classe *c*. Duração
$S_{ta} \in \{1, 2\}$	do arco *um* para a mercadoria *para*.
$M_{tc} \in$ Não	Quantidade mínima de aulas duplas exigida pelo professor *para* na aula *c*.
$E_{para} \in$ Não	Quantidade mínima de dias úteis para o professor *para*. Indica se o
$o \in \{0, 1\}$	requisito H6 é levado em conta. Custo de cada aula dupla necessária não
$\delta=1$	satisfeito.
$\omega=3$	Custo para cada período ocioso.
$gama=9$	Custo para cada dia útil.
Variáveis	
$x_{ta} \in \{0, 1\}$	Indica se a mercadoria *para* usa arco *um*.
$g_{tc} \geq 0$	Número de aulas duplas não satisfeitas *c* ensinado pelo professor *para*.

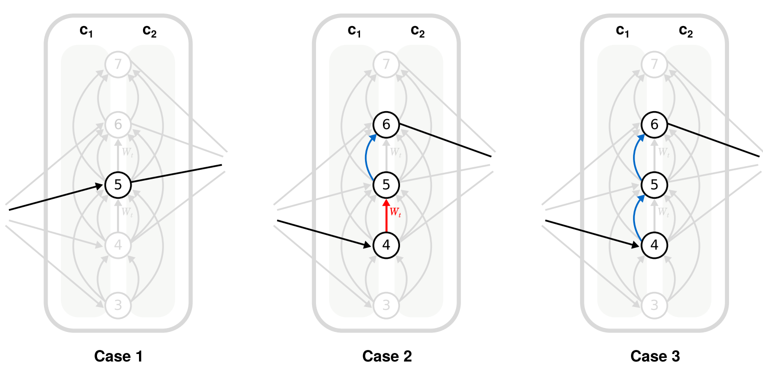

Figura 3. Exemplo de três casos diferentes em que caminhos sem sentido podem ser formados.

4. Geração de colunas aplicada ao problema

Aplicando os princípios de decomposição de Dantzig-Wolfe (Dantzig e Wolfe, 1960) na formulação compacta F_1, podemos obter uma formulação alternativa para o CTTPCR, denotada como *problema principal* (MP). Nesta formulação, afirmada por (17)–(20), deixar Eu_{para} ser o conjunto de todos os caminhos possíveis para um professor *para* que satisfazem todos os requisitos rígidos, exceto H3. Para cada caminho $eu \in Eu_{para}$ está associado um custo não negativo E_{tj} quanto à satisfação de requisitos suaves. Além disso, definimos uma variável binária λ_{tj} que indica se o caminho *eu* é selecionado pelo professor *para*.

$$\textbf{Minimizar} \quad \sum_{para \in E} \sum_{eu \in Eu_{para}} E_{tj} \lambda_{tj} \tag{17}$$

$$\textbf{sujeito a} \quad \sum_{eu \in Eu_{para}} \lambda_{tj} = 1 \qquad \forall\, para \in E \tag{18}$$

$$\sum_{para \in T} \sum_{j \in Eu_{para}} \sum_{um \in UM_{tcdp}} \bar{x}_{ta\,eu} \lambda_{tj} \leq 1 \qquad \forall\, c \in C, d \in E, p \in P \tag{19}$$

$$\lambda_{tj} \in \{0,1\} \qquad \forall\, para \in T,\ j \in Eu_{para} \tag{20}$$

O objetivo do MP, representado por Equação (17), é minimizar o custo dos caminhos selecionados. Conjunto de restrições (18) garante que exatamente um caminho seja selecionado para cada professor. Conjunto de restrições (19) garante que a capacidade unitária dos arcos de aula seja respeitada, onde $\bar{x}_{ta\,eu}$ indica se o arco *um* é usado no caminho *eu* do professor *para*. Ao resolver o MP, pode-se obter uma solução ótima inteira para o CTTPCR. No entanto, isso pode ser impraticável dada a enorme cardinalidade de Eu_{para} em problemas enfrentados em aplicações reais. Em vez disso, propomos resolver um relaxamento linear do MP por meio de uma abordagem de geração de colunas, com o propósito de atingir limites inferiores estreitos para o problema.

A rigidez do limite depende principalmente de como os requisitos são distribuídos entre os problemas mestre e de precificação. No entanto, como regra geral, quanto mais requisitos são tratados no problema de precificação, mais rígido é o limite produzido pelo problema de precificação. geração de umn. Com esta regra em mente, observamos que no modelo estendido proposto por Santos et al. (2012), os requisitos H1, H3 e S3 são tratados no problema mestre. Portanto, nesta pesquisa, propomos um modelo no qual todos os requisitos, exceto H3, são tratados no problema de precificação. Fomos capazes de realizar isso usando uma representação multicommodity.

Em uma implementação simples, um procedimento de geração de colunas começa com um problema mestre preenchido com um conjunto restrito de colunas, doravante denominado *problema mestre restrito* (RMP). A cada iteração, o RMP é resolvido e suas variáveis duais são usadas para precificar novas colunas resolvendo subproblemas. Durante a resolução de cada subproblema (problema de precificação), colunas com um custo reduzido negativo são adicionadas ao RMP. Este procedimento é repetido até que nenhuma coluna com custo reduzido negativo seja encontrada. No nosso caso, consideramos o RMP declarado por (21)–(25).

$$\textbf{Minimizar} \quad \sum_{para\in E} \left(\sum_{eu\in Eu_{para}} E_{tj}\lambda_{tj} + e_{para} por_{para} \right) \tag{21}$$

sujeito a

$$\sum_{eu\in Eu_{para}} \lambda_{tj} + por_{para} = 1 \quad \forall para\in E \tag{22}$$

$$\sum_{para\in T} \sum_{j\in Eu_{para}} \sum_{um\in UM_{tcdp}} x_{ta\,eu}\lambda_{tj} \leq 1 \quad \forall c\in C d\in E,\ p\in P \tag{23}$$

$$\lambda_{tj} \geq 0 \quad \forall para\in T,\ j\in Eu_{para} \tag{24}$$

$$por_{para} \geq 0 \quad \forall para\in E \tag{25}$$

Observe que as variáveis são contínuas e seus limites superiores são implícitos pelo conjunto de restrições (22). Além disso, optamos por iniciar o conjunto inicial de colunas introduzindo uma variável artificial por_{para} para cada professor, que é penalizado com um alto custo na função objetivo. Conforme apontado por Lübbecke e Desrosiers (2005), atribuir arbitrariamente um custo muito alto para variáveis artificiais pode retardar a convergência da geração de colunas. Assim, para manter a penalização o mais baixa possível, definimos o custo de e_{para} de acordo com Equação (26)

$$e_{para} = \delta \sum_{c\in C} M_{tc} + \omega\, \text{máx.}(0, |P|-2)|E| + gama|E|. \tag{26}$$

O custo e_{para} é igual à soma de três partes que correspondem, respectivamente, aos limites superiores dos custos das restrições suaves S1, S2 e S3. Em outras palavras, e_{para} é um limite superior trivial para o custo de um caminho de professor (E_{tj}).

Após a resolução do RMP, o próximo passo consiste em um esquema de precificação múltipla, onde o subproblema P_{para} é resolvido para cada $para\in E$. Equações (27)–(35) apresentar a formulação de P_{para}:

$$P_{para} = \begin{cases} \textbf{Minimizar} \quad R_{para} = \sum \delta g_{tc} + \sum gama\, x_{ta} + \sum_{um\in C_{para}} \omega x_{ta} \\ \quad - \sum_{c\in C} \sum_{d\in D} \sum_{p\in P} \sum_{um\in UM_{tcdp}} \sigma_{CDP} x_{ta} - \pi_{para} & (27) \\ \textbf{Sujeito a} \\ \sum_{um\in UM_{v}} x_{ta} - \sum_{um\in UM_{para\,você}} x_{ta} = bvocê \quad \forall você\in V & (28) \\ \sum_{um\in \bigcup_{d\in D, p\in P} UM_{tcdp}} S_{ta} x_{ta} = O_{tc} \quad \forall c\in C & (29) \\ \sum_{um\in \bigcup_{p\in P} UM_{tcdp}} S_{ta} x_{ta} \leq eu_{tc} \quad \forall c\in C d\in E & (30) \\ \sum_{um\in \bigcup_{p\in P} UM_{tcdp}} x_{ta} \leq 1 \quad \forall c\in C d\in E,\ h=1 & (31) \\ g_{tc} \geq M_{tc} - \sum_{um\in G_{tc}} x_{ta} \quad \forall c\in C & (32) \\ \sum_{um\in E_{para}} x_{ta} \geq E'_{para} & (33) \\ x_{ta}\in\{0,1\} \quad \forall um\in UM_{para} & (34) \\ g_{tc} \geq 0 \quad \forall c\in C & (35) \end{cases}$$

Assumindo π_{para} e σ_{CDP} como variáveis duais associadas, respectivamente, aos conjuntos de restrições (22) e (23), o custo reduzido R_{para} é definido por Equação (27). Por fim, observe que os conjuntos de restrições restantes, a saber (28)-(35) são análogos aos apresentados em F_1.

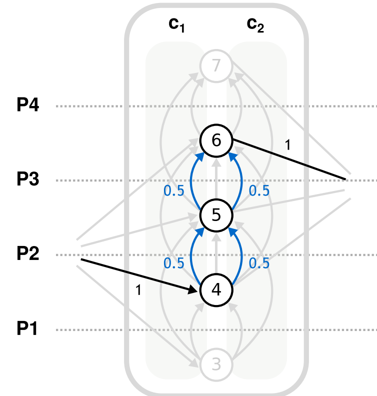

Figura 4. Exemplo de uma solução de subproblema relaxada em um dia útil. O número ao lado dos arcos representa o respectivo fluxo de valor. Podemos observar que os fluxos integrais que passam pelos arcos pull-in e pull-out determinam se o respectivo professor ensinará do período P2 ao P3. No entanto, o fluxo é dividido em arcos de aula e não podemos determinar para cada período se uma aula deve ser dada ou para a classe c_1 ou c_2.

4.1. Estratégias de aceleração

Ao comparar o esforço computacional esperado necessário para resolver os problemas mestre e de precificação, é fácil prever que o gargalo de todo o processo de geração de colunas está na resolução do problema de precificação. P_{para}. Além de ser um problema inteiro, P_{para} também tem que abordar a maioria dos requisitos do CTTPCR. Recorremos a um solucionador MIP para resolvê-lo, no entanto, pode-se notar que mesmo usando um solucionador MIP de última geração, a solução P_{para} para a otimalidade ainda pode ser demorado. Isso é particularmente notável no que diz respeito à resolução de instâncias médias e grandes do problema. Assim, para acelerar a resolução de P_{para} com um solucionador MIP, propomos as duas estratégias descritas a seguir.

A primeira estratégia é baseada no princípio de que qualquer coluna com custo reduzido negativo contribui para melhorar o valor objetivo do problema mestre restrito. Portanto, P_{para} não precisa ser resolvido exatamente em cada iteração, pois isso é obrigatório apenas na última. Com isso em mente, projetamos nosso algoritmo de geração de colunas para operar em duas fases sequenciais (I e II). Na fase I, em vez de resolver um subproblema até a otimalidade, paramos o solucionador assim que ele encontrar uma solução viável comprovadamente dentro de uma porcentagem dada um longe do ideal. Quando não há mais soluções com R_{para}<0 pode ser gerado usando o valor atual de um, o algoritmo muda para a fase II onde um é definido como zero e, consequentemente, os subproblemas são resolvidos de forma otimizada.

A segunda estratégia de aceleração consiste em resolver uma versão relaxada são de P_{para}, doravante denotado por P'_{para}, onde a integralidade con-as restrições são aplicadas apenas para as variáveis associadas ao pull-in e arcos de extração. Em outras palavras, significa que P'_{para} é capaz de pré-determinar precisamente os primeiros e últimos períodos de aula para cada dia útil de um professor. No entanto, como consequência do relaxamento, pode não ser possível identificar exatamente em qual classe uma aula deve ser dada, conforme ilustrado na Figura 4.

Apesar de perder alguma informação, o relaxamento diminui consideravelmente o número de variáveis inteiras do subproblema. Enquanto P_{para} tem um grande número de variáveis inteiras que dependem de o número de eventos, P'_{para} usa somente $|P_{para}^{q} \cup P_{para}^{q}| \times |E| = 2|P| \times |E|$ bi-variáveis genéricas. Em casos práticos, $|E|$ é normalmente limitado por 6 (segunda a sábado) e $|P|$ dificilmente ultrapassa 20, mesmo em escolas que mantêm programas de período integral. Nesse sentido, é seguro afirmar que para

Tabela 2
Principais características das instâncias testadas.

Eu ia	Nome	$\|E\|$	$\|P\|$	$\|E\|$	$\|C\|$	$\|E\|$	$(para,c)\in E M_{tc}$	$(para,c)\in E O_{tc}$	LIBRA$_{gama}$	BKV	BKV$_{isso\ é}$
1	Inst1	5	5	8	3	21	21	75	189	202	–
2	Inst2	5	5	14	6	63	29	150	333	333	–
3	Inst3	5	5	16	8	69	4	200	414	423	–
4	Inst4	5	5	23	12	127	66	300	639	652	–
5	Inst5	5	5	31	13	119	71	325	756	762	–
6	Inst6	5	5	30	14	140	63	350	738	756	–
7	Inst7	5	5	33	20	205	84	500	999	1017	–
UM	BrazilInstance1	5	5	8	3	21	21	75	189	200	11
E	BrazilInstance4	5	5	23	12	127	66	300	621	648	27
E	BrazilInstance5	5	5	31	13	119	71	325	756	775	19
F	BrazilInstance6	5	5	30	14	140	63	350	738	779	41
G	BrazilInstance7	5	5	33	20	205	84	500	999	1052	53

para fins práticos, o número de variáveis inteiras de P'_{para} é contenso entre 12|P| e 240 para qualquer instância do mundo real.

Por fim, é importante ressaltar que mesmo com P'_{para} ser menos restritivo que P_{para}, como mostramos mais adiante no computacional resultados, ambos os problemas muitas vezes encontram o mesmo valor objetivo, ou seja, R'_{para}-R_{para}. Como resultado, o limite inferior obtido pela geração de colunas geração usando P'_{para} é fortemente próximo daquele obtido usando P_{para}. Referimo-nos ao método de geração de colunas que utiliza P_{para} na etapa de precificação como Geração de Coluna de Precificação Inteira (IPCG), enquanto o versão que usa P'_{para} é identificado como Coluna de Preços Relaxados Geração (RPGG).

5. Experimentos computacionais

Nesta seção apresentamos uma avaliação experimental para os modelos e métodos propostos. As instâncias são resolvidas pelo CPLEX 12.6.0 (IBM, 2013) com configurações padrão, mas usando um único núcleo. Os algoritmos foram implementados em C++ usando o compilador g++4.6.1. Os resultados experimentais foram computados em um Desktop-PC equipado com um processador Intel Core i5-2300 com clock de 2,8 gigahertz, 4 gigabytes de RAM, sobre um sistema operacional Linux de 64 bits. Ao longo desta seção, relatamos os resultados de uma execução para cada algoritmo testado, uma vez que são determinísticos. Os parâmetros do modelo matemático *gama*, ω, e δ foram definidos como 9, 3 e 1, respectivamente.

5.1. Conjunto de dados

Para avaliar o algoritmo utilizamos as instâncias apresentadas em Tabela 2. Nesta tabela, as duas primeiras colunas apresentam o identificador da instância e o nome. Como seus nomes são longos, usamos os identificadores para encurtar a referência ao longo do texto. As instâncias são divididas em dois conjuntos. As instâncias 1–7 compreendem o conjunto 1 e estão disponíveis no repositório (LABIC, 2008). O requisito (H6) não é considerado neste grupo de instâncias. As instâncias A, D, E, F, G, do conjunto 2, são versões diferentes das instâncias 1, 4, 5, 6, 7, respectivamente. Elas diferem principalmente em dois aspectos: no conjunto 2, os professores estão disponíveis em todos os períodos, e o requisito (H6) é considerado. Essas modificações tornaram as instâncias mais desafiadoras. O conjunto 2 compreende todas as instâncias usadas na primeira rodada da Terceira Competição Internacional de Horários de 2011 (CIT, 2011) que nosso modelo pode manipular. Eles fazem parte do arquivo XHSTT-2012 que está disponível em https://www.utwente.nl/ctit/hstt/archives/XHSTT-2012/XHSTT-2012-separate_instances.zip. Outras instâncias disponíveis neste arquivo levam em conta requisitos que o CTTPCR não lida, como atribuição de recursos, eventos vinculados e eventos com vários professores. Usamos as versões originais das instâncias (primeira rodada do ITC2011), pois algumas delas foram modificadas após a competição. Colunas |E| e |P| mostram o número de dias e períodos, respectivamente, enquanto as colunas |E|, |C| e |E| apresentar o número de professores, turmas e eventos, respectivamente, onde $E=\{(t, c): para\in V, C\in C, H_{tc}>0\}$. Colunas e $(para,c)\in E M_{tc}$ $(para,c)\in E O_{tc}$ apresentam o total de aulas duplas necessárias e a carga horária total, respectivamente. Coluna LB_y apresenta um limite inferior simples calculado de acordo com Equação (12). Coluna BKV apresenta os melhores valores conhecidos para essas instâncias, todos eles obtidos por Dorneles e outros (2014) exceto as instâncias E e G. Os melhores valores conhecidos para as instâncias E e G foram obtidos neste trabalho usando a mesma abordagem de correção e otimização proposta em Dorneles e outros (2014) embora usando um limite de tempo de 10 horas. Os valores da solução relatados na coluna BKV foram calculados da mesma forma que todos os trabalhos anteriores no CTTPCR, ou seja, usando a função objetivo definida na formulação F_1. Entretanto, no ITC-2011, os valores da solução foram relatados com uma função objetivo diferente usando o validador HSEVal.[1] Coluna $BKV_{isso\ é}$ apresenta os valores fornecidos por HSEval para as melhores soluções conhecidas do conjunto-2. As células do conjunto-1 são preenchidas com "-", pois as instâncias 1–7 não foram testadas no ITC-2011. A única diferença entre os valores de solução computados por F_1 e HSEval é que este último não inclui o valor de LB_y como parte do custo da solução, ou seja, $BKV_{isso\ é}=BKV-LB_y$.

As próximas seções têm como objetivo apresentar resultados dos algoritmos e modelos apresentados neste trabalho, bem como compará-los com os resultados anteriores do estado da arte para o problema.

5.2. Soluções inteiras obtidas por modelos MIP

O primeiro experimento visa comparar resultados de um modelo MIP tradicional (Dorneles e outros, 2014) e o modelo de fluxo F_1 proposto neste trabalho. Cada instância foi executada por no máximo 7200 segundos (2 horas) para cada modelo. Tabela 3 apresenta para cada instância e modelo MIP, o valor objetivo encontrado (*Obj*), os tempos de execução (*Tempo*), o número de colunas (*#col*) e linhas (*#linha*) gerado pelo respectivo modelo, a lacuna fornecida pelo CPLEX (*Brecha$_C$*) e a Lacuna (*Brecha$_B$*) do valor mais conhecido calculado como 100∗(*Obj*−*BKV*)/*min(BKV,Objeto)*. Os valores mais baixos para cada coluna são mostrados em negrito.

Das 12 instâncias, o modelo de fluxo foi capaz de encontrar melhores resultados em seis instâncias, e resultados semelhantes em duas instâncias. Apenas em quatro instâncias o modelo proposto anteriormente encontrou melhores resultados do que F_1. No entanto, em média, o modelo tradicional encontrou melhores resultados de gap. Isso se deve aos resultados da instância 7 que F_1 encontrou uma solução final com um custo objetivo consideravelmente maior que o valor objetivo encontrado pelo modelo tradicional. O número de colunas de F_1 foi maior em apenas três casos, enquanto o número de linhas do modelo tradicional foi maior em todos os casos.

[1] O HSEVal verifica a viabilidade da solução e também calcula o valor da solução. Ele está disponível em Português Português-jeff/cgi-bin/hseval.cgi.

Tabela 3
Resultados da comparação entre modelos MIP usando um limite de tempo de 2 horas.

Eu ia	Dorneles e outros (2014)						F_1					
	Obj	*Tempo*(horas)	# coluna	# linha	Brecha$_c$	Brecha$_B$	Obj	*Tempo*	# coluna	# linha	Brecha$_c$	Brecha$_B$
1	**202**	**2**	1306	2914	6.44	**0,00**	**202**	**2 horas**	**1163**	**1187**	**5,82**	**0,00**
2	340	2	3100	6855	2.06	2.10	**333**	**196 segundos**	**3021**	**2480**	**0,00**	**0,00**
3	**426**	**2**	2898	6532	**2,82**	**0,71**	429	**2 horas**	**2424**	**2305**	3,50	1,42
4	653	2	5221	11.642	2.14	0,15	**652**	**988 segundos**	**4174**	**4081**	**0,00**	**0,00**
5	782	**2**	**6349**	14.057	3.32	2,62	**777**	**2 horas**	6602	**6613**	**2,70**	**1,97**
6	**780**	**2**	**6850**	15.153	**5.38**	**3.17**	804	**2 horas**	7000	**6653**	8.02	6.35
7	**1043**	**2**	**9155**	20.242	**4.22**	**2,56**	1645	**2 horas**	9364	**8805**	38,81 61,75	
UM	**200**	**2**	2011	3910	5,50	**0,00**	**200**	**2 horas**	**1566**	**1513**	**4,00 0,00**	
E	735	**2**	10.132	19.008	15,51	13.43	**726**	**2 horas**	**7168**	**7571**	**12.12**	**12.04**
E	868	**2**	10.244	19.513	12,90 12,00		**812**	**2 horas**	**7568**	**9594**	**5.36**	**4,77**
F	1174	**2**	11.530	21.772	37,14 50,71		**952**	**2 horas**	**8312**	**10.012**	**20,69 22,21**	
G	**1248**	**2**	16.200	30.328	19,95	**18,63**	1285	**2 horas**	**11.380**	**14.121**	**19,88**	22h15
Média	**704**	2	7083	14.327	**9,78 8,84**		735	**6099 segundos**	**5812**	**6245**	10.08 11.05	

Tabela 4
Resultados da comparação entre relaxamentos lineares.

Eu ia	P_1			Dorneles e outros (2014)			IPCG(um=0 por cento)		
	LIBRA	Tempo (segundos)	Brecha$_{eu}$(por cento)	LIBRA	Tempo (segundos)	Brecha$_{eu}$(por cento)	LIBRA	Tempo (segundos)	Brecha$_{eu}$(por cento)
1	189	**0,1**	6,88	189	**0,1**	6,88	**202**	11.3	**0,00**
2	**333**	**0,4**	**0,00**	**333**	1.1	**0,00**	**333**	8.7	**0,00**
3	414	**0,2**	2.17	414	0,9	2.17	**423**	12.0	**0,00**
4	643	**0,8**	1,40	639	2.2	2.03	**652**	10.0	**0,00**
5	756	**2.3**	0,79	756	6.8	0,79	**762**	13,5	**0,00**
6	738	**2.8**	2,44	738	9.0	2,44	**756**	40,5	**0,00**
7	999	**7.8**	1,80	999	24.1	1,80	**1017**	25,5	**0,00**
UM	190	**0,1**	5.26	189	0,2	5,82	**200**	6.7	**0,00**
E	635,5	**3.8**	1,97	621	14.1	4,35	**646**	26.4	**0,31**
E	767,5	**3.9**	0,98	756	13.8	2,51	**775**	20.2	**0,00**
F	754	**5.5**	3.32	738	23.0	5,56	**773**	45.2	**0,78**
G	1023	**12.0**	2,83	999	77,2	5.31	**1039**	85,1	**1,25**
Média	620	**3.3**	2,49	614	14,5	3.31	**632**	25,4	**0,20**

5.3. Limites inferiores para o problema

Esta seção tem como objetivo apresentar e comparar resultados de limites inferiores fornecidos pelo relaxamento linear de F_1(denotado como F_1'), o limite inferior encontrado pelo relaxamento linear do modelo tradicional de Dorneles e outros (2014), e o método IPCG proposto neste trabalho com um= 0 por cento. Tabela 4 apresenta para cada instância e método, o limite inferior encontrado (LB), os tempos de execução (Tempo) em segundos e o desvio percentual do melhor valor conhecido para o limite inferior *Brecha$_{eu}$*, que é calculado como 100∗(*BKV*−*LIBRA*)/*LIBRA*. Os melhores resultados para cada coluna são mostrados em negrito.

Em resumo, pode-se observar pelos resultados que F_1' é o método mais rápido enquanto IPCG fornece os melhores limites inferiores considerando todas as instâncias testadas. Ambas as abordagens, F_1' e IPCG, apresentam melhorias significativas em comparação com Dorneles e outros (2014). F_1' fornece limites inferiores melhores ou iguais do que Dorneles e outros (2014), usando cerca de quatro vezes menos tempo, em média. Embora IPCG passou cerca de duas vezes o tempo de Dorneles e outros (2014), o *Brecha$_{eu}$* alcançado pelo primeiro é aproximadamente dezesseis vezes menor. Ao comparar IPCG e F_1' o *Brecha$_{eu}$* encontrado por IPCG é consideravelmente melhor, mas demora mais para ser executado.

5.4. Teste de parâmetros para o algoritmo de geração de colunas proposto

Nesta seção, avaliamos o desempenho de diferentes configurações e estratégias de aceleração para a geração de colunas proposta. Tabela 5 apresenta resultados médios em relação a IPCG e RPGG com vários valores para um(apresentado em Seção 4.1). Para ambos os métodos são relatados os tempos totais de execução (*Tempo*) em segundos, o total número de colunas geradas (*#col*), o número total de iterações (*#iter*), e a porcentagem do tempo total gasto na etapa de precificação (*RP*). O menor tempo de execução é mostrado em negrito. Os valores são mostrados entre parênteses ao lado das colunas *#coluna* e *#iter* apresenta o número de colunas/iterações que foram inseridas/realizadas na fase II do algoritmo. Lembramos que quando um for definido como zero, a fase II não precisa ser executada.

Gostaríamos de destacar que, independentemente da utilização de uma tarifação flexível, todos os limites inferiores gerados por RPGG corresponderam exatamente aos obtidos por IPCG em Tabela 4(discutiremos mais este tópico na próxima subseção). Assim, os resultados mostrados em Tabela 5 estão focados em apresentar suas diferenças em termos de tempos de execução e número de iterações.

Da tabela, destaca-se que RPGG é mais rápido que IPCG quando o mesmo valor de um é considerado. Ambos os algoritmos são afetados de forma semelhante de acordo com as mudanças no parâmetro um, demorando mais quando um= 0 por cento e um>10 por cento. Nestes casos, a desaceleração é causada devido à qualidade das colunas geradas na etapa de precificação. Por um lado, quando um= 0 por cento, tempo computacional extra é gasto para gerar colunas de alta qualidade, garantindo a otimalidade para cada problema de precificação. Por outro lado, quando um>10 por cento, embora a etapa de precificação seja mais rápida, ela adiciona um número maior de colunas de baixa qualidade ao mestre, o que aumenta o número de iterações necessárias para atingir a otimização, especialmente na fase II.

Um cenário adequado para um, que fornece um bom equilíbrio entre qualidade e esforço computacional para gerar uma coluna, é composto por 1 por cento≤ um≤10 por cento. Nesta faixa, encontramos os melhores resultados gerais alcançados por RPGG com um= 4 por cento, que combina as duas estratégias de aceleração propostas em

Tabela 5
Resultados médios para todas as instâncias comparando diferentes configurações para a geração de colunas proposta.

	IPCG				RPGG			
um(por cento)	Tempo (segundos)	# coluna	# iter	pr (porcentagem)	Tempo (segundos)	# coluna	# iter	pr (porcentagem)
0	25,4	33	784	94,7	15.7	33	782	91,7
1	24,8	34 (1)	778	95,1	15.0	34 (1)	781	91,3
2	21.4	35 (1)	780	94,2	13.9	35 (1)	805	90,3
3	19,7	35 (1)	781	93,8	14.0	37 (1)	830	89,6
4	19.4	36 (1)	789	93,7	**13.8**	38 (1)	847	89,2
5	18.9	36 (1)	795	93,4	14.2	40 (1)	891	88,7
6	19.0	37 (1)	799	93,3	14.6	42 (1)	927	88,2
7	18,5	37 (1)	806	92,9	15.1	45 (1)	942	88,1
8	18.2	38 (1)	798	92,8	15.9	47 (1)	987 (3)	87,3
9	18.0	39 (1)	803	92,8	15,5	47 (1)	996	86,8
10	18.6	40 (1)	822 (1)	92,8	16.9	51 (1)	1039 (1)	86,3
20	24.2	53 (3)	893 (16)	92,6	17.0	58 (1)	1074 (6)	85,8
30	41,5	101 (5)	1027 (69)	93,2	32,7	105 (2)	1441 (27)	85,7
40	44,0	112 (8)	990 (157)	94,2	39,8	129 (5)	1333 (76)	87,4
50	35,0	79 (13)	873 (266)	94,5	31.6	124 (9)	1104 (166)	89,8
60	32.1	65 (17)	830 (350)	94,8	31.4	128 (11)	1082 (225)	90,2
70	35,3	77 (18)	823 (371)	95,2	34.1	144 (11)	1078 (217)	90,6
80	35,9	79 (20)	809 (413)	95,8	32,7	134 (12)	1061 (236)	91,2
90	35,9	77 (23)	827 (464)	95,6	29,6	124 (13)	1043 (259)	91,0

Tabela 6
Resultados apresentando a diferença ( ) entre os custos reduzidos proporcionados por P_{para} e P'_{para}.

Eu ia	# preços	> 0(por cento)		≤1(por cento)		> 1 (por cento)		Máx( )	Média( )
1	232	51	(21,98)	231	(99,57)	1	(0,43)	1.08	0,06
2	420	37	(8.81)	420	(100,00)	0	(0,00)	0,89	0,02
3	416	5	(1.20)	416	(100,00)	0	(0,00)	0,56	0,00
4	782	66	(8.44)	782	(100,00)	0	(0,00)	1,00	0,02
5	806	74	(9.18)	806	(100,00)	0	(0,00)	0,50	0,01
6	1020	101	(9,90)	1020	(100,00)	0	(0,00)	0,38	0,01
7	1023	114	(11.14)	1023	(100,00)	0	(0,00)	0,61	0,02
UM	248	5	(2.02)	248	(100,00)	0	(0,00)	0,83	0,00
E	759	161	(21.21)	758	(99,87)	1	(0,13)	1.18	0,04
E	1023	146	(14.27)	1023	(100,00)	0	(0,00)	0,85	0,03
F	1320	270	(20h45)	1320	(100,00)	0	(0,00)	1,00	0,05
G	1650	619	(37,52)	1649	(99,94)	1	(0,06)	1.12	0,07
Média	808	137	(13,84)	808	(99,95)	0	(0,05)	0,83	0,03

Seção 4.1. No entanto, considerando que as acelerações para IPCG(um= 9 por cento), RPGG(um=0 por cento) e RPGG(um=4 por cento) calculado sobre IPCG(um=0 por cento) são, respectivamente, 1,41, 1,61 e 1,84, pode-se observar que as estratégias de aceleração propostas são capazes de melhorar significativamente a convergência da geração de colunas, mesmo se usadas exclusivamente. Na verdade, a estratégia de preços relaxados pode fornecer uma aceleração maior do que a introdução de uma um>0 por cento. Quando ambas as estratégias são usadas juntas, os resultados são ligeiramente melhores.

5.5. Valores objetivos fornecidos por P_{para} e P'_{para}

Nesta seção, avaliamos empiricamente o nível de aproximação ção fornecida por P'_{para} em comparação com P_{para}. Desde P'_{para} é um relaxação de P_{para}, podemos denotar a diferença entre a resposta ótima reduziu o custo desses problemas por $= R_{para} - R'_{para}$. Para medir a magnitude dessa diferença, corremos IPCG usando um= 0 por cento. Durante a etapa de precificação, além de resolver P_{para} nós também resolvemos P'_{para} para fins de cálculo do valor de . Relatamos em Tabela 6, para cada instância, o número total de problemas de preços resolvidos (*#preço*). Além disso, para os casos de > 0, ≤1, e > 1, nós informe o número de ocorrências de cada um desses casos, e a porcentagem dessas ocorrências considerando o número total de precificações resolvidas. Finalmente, os valores máximo e médio de são fornecidos nas duas últimas colunas.

Analisando a tabela, pode-se notar que, em média, apenas 13,84 por cento dos problemas de preços resolvidos revelaram uma diferença entre interpolação P_{para} e P'_{para}. No entanto, quase 100 por cento das diferenças são, em média, menores ou iguais a um custo unitário (ver coluna ≤1). Além disso, entre todas as execuções, apenas três problemas de preços resultaram em uma diferença maior que um custo unitário (ver coluna > 1). Nesses casos raros, o valor ainda mal ultrapassava o custo de uma unidade.

Conforme mostrado na última coluna da tabela, a média é somente 0,03, significando assim que a diferença entre os valores calculados por P'_{para} e P_{para} é minúsculo. Na verdade, 0,03 corresponde a um valor que é cerca de 33 vezes menor do que a penalidade de menor custo associada a uma aula dupla não satisfeita (δ=1). Atribuímos a esta pequena diferença o fato de que os limites inferiores obtidos por IPCG e RPGG são os mesmos, conforme mencionado na seção anterior.

5.6. Comparando o método proposto com uma Geração de Corte e Coluna

Nesta seção, comparamos nosso método de geração de colunas com a abordagem proposta por Santos et al. (2012), doravante denominado Geração de Corte e Coluna (CCG). Usamos o original deles CCG implementação, que foi gentilmente fornecida pelos autores. Para comparar os resultados, incluímos o requisito H6 em sua implementação. Tabela 7 apresenta resultados para cada instância comparando o desempenho de CGG e RPGG usando um= 4 por cento. Para ambos os métodos são relatados os tempos totais de execução (*Tempo*) em segundos, o número total de colunas geradas (*#col*), e o número total de iterações (*#iter*) realizados. Ambos os métodos, CCG e

Tabela 7
Resultados da comparação entre os limites inferiores fornecidos pela Geração de Corte e Coluna (CCG) proposto por Santos et al. (2012) e a proposta de Geração de Coluna de Preços Relaxados (RPGG).

				CCG			RPGG			
Eu ia	LIBRA	LIBRA$_{isso\ é}$	Lacuna (porcentagem)	Tempo (segundos)	# coluna	# iter	Tempo (segundos)	# coluna	# iter	Aceleração
1	202	–	0,00	**0,17**	351	15	3.26	248	32	0,05
2	333	–	0,00	**5.12**	960	30	6.34	476	35	0,81
3	423	–	0,00	**2.22**	916	28	4.24	414	27	0,52
4	652	–	0,00	40,25	1474	44	**6,64**	735	37	6.06
5	762	–	0,00	34,43	1888	35	**11.29**	813	28	3.05
6	756	–	0,00	72,25	2102	56	**13.37**	835	29	5.41
7	1017	–	0,00	395,63	2284	67	**16h30**	1020	32	24.27
UM	200*	11*	0,00	**0,33**	443	19	3,25	232	30	0,10
E	646*	25*	0,31	50,74	1524	44	**14h35**	847	40	3,54
E	775*	19*	0,00	97,30	1918	73	**20,36**	1184	53	4,78
F	773*	35*	0,78	34,49	1865	37	**23,53**	1338	49	1,47
G	1039*	40*	1,25	451,38	2740	78	**42,22**	2026	63	10,69
Média	631	–	0,20	98,69	1539	44	**13,76**	847	38	5.06

RPGG, obteve exatamente os mesmos valores de limite inferior, que são relatados na coluna *LIBRA*. Coluna *LIBRA$_{isso\ é}$* apresenta os limites inferiores para instâncias do conjunto-2 de acordo com a função objetivo de HSEval (*LIBRA$_{isso\ é}$* = *LIBRA* − *LBy*), ou seja, *LIBRA$_{isso\ é}$* é um limite inferior para *BKV$_{isso\ é}$*. Coluna *Brecha* relata a lacuna de otimalidade para cada instância. É calculado por 100∗(*BKV*-*LIBRA*)/*LIBRA*. Finalmente, coluna *Aceleração* apresenta a aceleração de RPGG sobre CCG. Coluna *Aceleração* é calculado por CCG/RPG. Os resultados com menor tempo de execução são mostrados em negrito. Os valores marcados com (*) são novos melhores limites inferiores.

De acordo com os resultados, além CCG e RPGG são capazes de fornecer os mesmos limites inferiores que são mais rígidos do que aqueles fornecidos pelo relaxamento do modelo F_1, eles apresentam um desempenho distinto de acordo com o tamanho da instância. Enquanto CCG atinge tempos de execução curtos em quatro pequenas instâncias (1,2,3 e A), RPGG escala melhor, com desempenho mais rápido nas oito instâncias médias e grandes restantes. Além disso, nosso método é aproximadamente 5 vezes mais rápido do que CCG, em média, e particularmente nas instâncias maiores, a melhoria de desempenho se torna mais proeminente, sendo cerca de 24 vezes mais rápida na instância 7 e cerca de 10 vezes mais rápida na instância G.

Além de encontrar os limites ótimos para todas as instâncias do conjunto-1, fomos capazes de encontrar novos limites inferiores mais estreitos para todas as instâncias do conjunto-2. Esses novos resultados permitem reduzir a lacuna média de otimalidade para essas instâncias de 2,09 por cento, conforme apresentado em Dorneles e outros (2014), para 0,20 por cento. Além disso, a otimalidade foi provada para as instâncias A e E, uma vez que os limites inferiores encontrados coincidiram com os melhores valores conhecidos.

6. Conclusões

Neste artigo, abordamos o Problema de Horários Professor-Classe com Requisitos de Compactidade (CTTPCR), que é uma variante bem conhecida do Problema de Horários do Ensino Médio originado em escolas brasileiras. Instâncias do CTTPCR compreendem um subconjunto das instâncias consideradas na Terceira Competição Internacional de Horários realizada em 2011, que também avaliou instâncias de vários outros países. Além de uma nova formulação de programação matemática baseada em uma rede de fluxo de multicommodities para o CTTPCR, propusemos uma abordagem de geração de colunas, usando duas estratégias de aceleração, para provar limites inferiores fortes para este problema.

Em comparação com a geração de colunas de última geração para CTTPCR, os resultados experimentais mostram que nossa abordagem é capaz de produzir os mesmos limites inferiores, embora com duas vantagens significativas: i) o método é mais simples; ii) e é cinco vezes mais rápido em média. Além disso, melhoramos os limites inferiores mais conhecidos de 5 de 12 instâncias da literatura. Entre esses novos resultados, dois são provados como ótimos (a saber, instância A e E). Esses resultados mostram que a técnica proposta é eficiente para produzir limites inferiores para o CTTPCR, motivando seu uso para variantes desse problema.

Como trabalho futuro, há várias direções nas quais esta pesquisa pode ser estendida. Primeiramente, pode-se propor um método personalizado para resolver P_{para} ou P'_{para} de uma forma mais eficiente do que usar um genérico Solucionador de MIP. Em segundo lugar, como cerca de 90 por cento do tempo computacional é gasto na etapa de precificação, os ganhos com a paralelização podem ser promissores. De fato, dado que uma precificação é resolvida para cada professor, eles poderiam ser resolvidos trivialmente em paralelo. Finalmente, para aproveitar todo o potencial da geração de colunas, pode-se propor estratégias de ramificação dentro de uma estrutura de ramificação e preço para fornecer, em última análise, soluções inteiras ótimas para o problema.

Agradecimentos

Este trabalho é apoiado pela CAPES e Petrobras, Brasil. Gostaríamos de agradecer ao Professor Haroldo Gambini Santos por compartilhar seus códigos de geração de colunas, o que nos tornou possível comparar resultados na mesma máquina.

Referências

Boyd, EA (1994). Planos de corte de Fenchel para programas inteiros. *Pesquisa Operacional, 42*(1), 53–64.
Dantzig, GB, & Wolfe, P. (1960). Princípio de decomposição para programas lineares. *Op-pesquisa de geracoes, 8*(1), 101–111.
Dorneles, Á. P., Araújo, OCB, & Buriol, LS (2012). O impacto da compacidade re-requisitos sobre a resolução do problema de horários do ensino médio. Em *Anais do XLIV Simpósio Brasileiro de Pesquisa Operacional (SBPO 2012)* (págs. 3336–3347).
Dorneles, Á. P., Araújo, OCB, & Buriol, LS (2014). Uma heurística de consertar e otimizar para o problema de horários do ensino médio. *Computadores e Pesquisa Operacional, 52*, 29–38.
Drexl, A., & Salewski, F. (1997). Requisitos de distribuição e compacidade con-tensões nos horários escolares. *Revista Europeia de Pesquisa Operacional, 102*(1), 193–214.
Even, S., Itai, A., & Shamir, A. (1975). Sobre a complexidade do cronograma e multi-problemas de fluxo de mercadorias. Em *Anais do 16º simpósio anual sobre fundamentos da ciência da computação, SFCS '75* (pp. 184–193). Washington, DC, EUA: IEEE Computer Society.
Gotlieb, C. (1963). A construção de horários de classe-professor. Em *Anais do IFIP, Amsterdã* (págs. 73–77).
Português IBM (2013). *Manual do usuário do ILOG CPLEX 12.6. Mountain View, Califórnia.*
ITC (2011). Terceira competição internacional de horários. Português http://www.utwente.nl/ctit/hstt/itc2011/. (acessado em 23.07.15).
Kristiansen, S., Sørensen, M. e Stidsen, TR (2014). Programação inteira para o problema generalizado de horários do ensino médio. *Revista de Agendamento, 18*(4), 377–392.
LABIC (2008). Instâncias de horários escolares. http://labic.ic.uff.br/Instance/index.php?dir=Horário Escolar/. (acessado em 23.07.15).
Lübbecke, ME, & Desrosiers, J. (2005). Tópicos selecionados em geração de colunas. *Opera-Pesquisa de ações, 53*(6), 1007–1023.
Pillay, N. (2014). Uma pesquisa sobre horários escolares. *Anais de Operações Re-pesquisa, 218*(1), 261–293.

Post, G., Ahmadi, S., Daskalaki, S., Kingston, JH, Kyngas, J., Nurmi, C., & Ranfilho, D. (2012). Um formato XML para benchmarks em horários de ensino médio. *Anais de Pesquisa Operacional, 194*, 385–397.
Post, G., Gaspero, L., Kingston, JH, McCollum, B., & Schaerf, A. (2016). O terceiro competição internacional de horários.*Anais de Pesquisa Operacional, 239*, 69–75.
Post, G., Kingston, J., Ahmadi, S., Daskalaki, S., Gogos, C., Kyngas, J., et al. (2014). XHSTT: um arquivo XML para problemas de horários do ensino médio em diferentes países. *Anais de Pesquisa Operacional, 218*, 295–301.
Santos, HG, Ochi, LS, & Souza, MJ (2005). Uma heurística de busca tabu com eficiência estratégias de diversificação eficientes para o problema de horários de aula/professor. *Revista de Algoritmos Experimentais, 10*, 1–16.
Santos, HG, Uchoa, E., Ochi, LS, & Maculan, N. (2012). Limites fortes com corte e geração de colunas para horários de professores de turma.*Anais de Pesquisa Operacional, 194*, 399–412.
Schaerf, A. (1999). Uma pesquisa sobre horários automatizados.*Revisão de Inteligência Artificial, 13*(2), 87–127.
Schaerf, A., & Gaspero, LD (2001). Técnicas de busca local para educação problemas de horários. Em*Anais do 6º simpósio internacional sobre pesquisa operacional na Eslovênia (sor-01)*(págs. 13–23).
Souza, M. (2000).*Programação de horários em escolas: UMA próxima por meta-heurísticas.*(Ph.D. tese). Universidade Federal do Rio de Janeiro.
Souza, M. e Maculan, N. (2000). Melhorando quadros de horário de escolas através de caminhos mínimos.*Tendências em Matemática Aplicada e Computacional, 1*(2), 515–524.
Souza, M., Ochi, L., & Maculan, N. (2003). Metaheurísticas: Decisão computacional–fazendo.*Um algoritmo de busca GRASP-tabu para resolver problemas de horários escolares* (pp. 659–672). Editora acadêmica Kluwer.
Steinzen, I., Gintner, V., Suhl, L., & Kliewer, N. (2010). Um aplicativo de rede tempo-espaço abordagem para o problema de programação integrada de veículos e equipes com múltiplos depósitos.*Ciência dos Transportes, 44*(3), 367–382.
de Werra, D. (1971).*Construção de horários escolares por métodos de fluxo.*(págs. 12–22). INFORMAÇÕES. (9)