Um estudo de caso sobre a geração de quadros de horários no departamento de Ciência da Computação da UFRGS

Fábio V. P. Neukirchen
Instituto de Informática - Universidade Federal do Rio Grande do Sul - UFRGS
Porto Alegre - Brasil
fvpneukirchen@inf.ufrgs.br

Árton P. Dorneles
Instituto de Informática - Universidade Federal do Rio Grande do Sul - UFRGS
Porto Alegre - Brasil
arton.dorneles@inf.ufrgs.br

Raul F. Weber
Instituto de Informática - Universidade Federal do Rio Grande do Sul - UFRGS
Porto Alegre - Brasil
weber@inf.ufrgs.br

Luciana S. Buriol
Instituto de Informática - Universidade Federal do Rio Grande do Sul - UFRGS
Porto Alegre - Brasil
buriol@inf.ufrgs.br

RESUMO

A geração de quadro de horários para universidades é um problema clássico de otimização combinatória que leva em consideração um grande número de variáveis e requisitos. Este tipo de problema em geral contém restrições específicas da aplicação, gerando uma grande quantidade de variações do problema. Apesar do problema ser NP-Completo, o mesmo é resolvido manualmente na maior parte das instituições. Isso normalmente ocorre devido à dificuldade de formalizar todas as restrições que cada aplicação requer. Este trabalho apresenta um estudo da geração de quadro de horários no curso de Ciência da Computação da UFRGS. O problema é formalizado através de um modelo matemático e um estudo experimental é realizado em instâncias reais do problema. Os resultados obtidos demonstram que as instâncias consideradas podem ser resolvidas de maneira eficiente com o modelo proposto através de um resolvedor comercial de programação inteira mista.

PALAVRAS CHAVE. Quadros de horários. Programação inteira. Otimização combinatória.

Área Principal: EDU - PO na Educação, OC - Otimização Combinatória, PM - Programação Matemática

ABSTRACT

University timetabling is a classic combinatorial optmization problem that considers a large number of variables and requirements. That type of problem in general has specific application constraints generating a large quantity of variations of the problem. Despite the fact the problem belongs to the NP-Complete class of complexity, the same is usually manually solved in most institutions. That normally occurs because is difficult to formalize all the requirements that the application requires. This work realizes a study about timetabling at the course of Computer Science at UFRGS. The problem is formalized by an mathematical model and one experimental study is realized using real instances of the problem. The results show that the instances can be efficiently solved with the proposed model using a general purpose mixed integer programming solver.

KEYWORDS. University timetabling. Integer programming. Combinatorial optimization.

Main Area: EDU - OR in Education, OC - Combinatorial Optimization, PM - Mathematical Programming

1. Introdução

O problema de geração de quadros de horários é um problema combinatório altamente restrito classificado como NP-completo (Zuters, 2006) para determinar a existência de uma grade de horários válida (problema de decisão), e NP-Díficil para determinar a melhor solução dentre as soluções válidas (problema de otimização). Pode ser encontrado em diversas áreas, tais como no ambiente empresarial (por exemplo, grande empresas com um vasto quadro de funcionários) e educacional (por exemplo, escolas ou universidades, considerando professores, cursos, recursos, entre outros), esportes, hospitais, entre outros. O problema das universidades possui um grande número de variações. Segundo Burke et al. (1997), problemas de geração de quadros de horário variam consideravelmente entre universidades com requisitos específicos para cada uma.

Em muitas universidades, o grande número de eventos associados a professores e recursos, combinados a uma grande variedade de requisitos, tornam a construção de uma grade de horários uma tarefa extremamente complexa. A geração de um quadro de horários ineficiente pode prejudicar o rendimento de alunos e professores. Por exemplo, uma sobrecarga de aulas em um dia da grade de horários dificulta o aprendizado de um estudante, e pode prejudicar o rendimento do professor.

Os primeiros trabalhos científicos sobre geração de quadros de horários remontam da década de 60 e desde então tem demandado uma crescente atenção (Gotlieb, 1963). Em seu estudo, Schaerf (1999) classifica os problemas de geração de quadros de horários educacionais em três grupos:

- Geração de quadros de horários para escolas (*school timetabling*);
- Geração de quadros de horários para universidades (*university timetabling*);
- Geração de quadros de horários para exames (*examination timetabling*).

O desenvolvimento de métodos computacionais para solucionar estes tipos de problemas já atraiu a atenção de pesquisadores de diversas áreas da comunidade científica nos últimos de 50 anos (Petrovic e Burke, 2004).

Neste trabalho será apresentando uma nova variação do problema de geração de quadros de horários para universidades que ocorre no contexto do curso de Ciência da Computação (CiC) oferecido no Instituto de Informática da Universidade Federal do Rio Grande do Sul (INF/UFRGS). Este curso é organizado em 7 etapas semestrais, onde cada uma tem seu próprio currículo de disciplinas. O currículo de uma etapa pode envolver disciplinas oriundas de outros departamentos e que já possuem horários pré-determinados. Este é o caso, por exemplo, de disciplinas oferecidas pelos Departamento de Matemática e Estatística. Além disso, cada disciplina normalmente é ofertada em várias turmas que compartilham professores e recursos físicos como, por exemplo, salas de aulas, laboratórios e auditórios.

Durante o processo de construção da grade de horários da CiC procura-se reduzir a ocorrência de *períodos ociosos* nos horários dos alunos, bem como utilizar eficientemente os recursos físicos disponíveis. Este último objetivo, em especial, tem ganho maior importância recentemente, devido ao aumento no número de alunos nas universidades brasileiras, inclusive no INF/UFRGS, sem o devido aumento proporcional da infraestrutura física.

O restante deste trabalho está organizado como segue. A Seção 2 define formalmente o problema abordado e apresenta uma formulação de programação linear inteira mista para este. A Seção 3 apresenta resultados computacionais referentes a resolução do modelo proposto utilizando três instâncias reais do INF/UFRGS. A Seção 4 apresenta as conclusões deste estudo e, finalmente, na Seção 5 é apresentada uma breve descrição das atividades desempenhadas individualmente por cada um dos autores neste trabalho.

2. Descrição do Problema e Modelagem Matemática

O objetivo deste problema é construir um quadro de horários semanal para todas as etapas do curso, atendendo restrições obrigatórias e minimizando a ocorrência de situações indesejadas no horário. A semana é divida em um conjunto de D dias, os quais possuem um conjunto de horários H. Um *período* é definido pela tupla (d,h), sendo $d \in D$ e $h \in H$, e assume-se que todos os períodos possuem a duração de uma aula. As aulas que precisam ser lecionadas no curso são definidas através de um conjunto de eventos E que é dividido em dois grupos: *eventos simples* (ES) e *eventos geminados* (EG). As aulas dos eventos simples ocorrem em dias distintos, e no mesmo período (por exemplo, segunda e quarta às 8:30, ou terça e quinta às 13:30). Enquanto as aulas dos eventos geminados ocorrem em um único dia e em períodos consecutivos (por exemplo, das 8:30 às 12:30). A cada etapa $m \in M$ está associada a um determinado conjunto de eventos E_m que compõe o currículo da etapa. O problema ainda considera um conjunto de professores T e um conjunto de tipos de recursos R que podem estar necessários em cada evento. Tanto professores, como recursos podem podem estar indisponíveis em alguns períodos na semana. Alguns eventos não devem ser alocados, preferencialmente, no mesmo período na semana (por exemplo, para separar aulas teóricas e práticas de uma mesma disciplina). Quando essa preferência não é satisfeita, é dito que existe um *conflito* entre eventos.

Um quadro de horários factível deve atender obrigatoriamente todos os *requisitos fortes* apresentados abaixo:

- **H1** Todas as aulas associadas com cada evento devem ser alocadas.
- **H2** Eventos devem ser alocados respeitando a disponibilidade dos professores e recursos envolvidos no evento.
- **H3** Professores devem ministrar no máximo uma aula em cada período.
- **H4** Alguns eventos devem ser alocados em períodos pré-determinados.
- **H5** A utilização de recursos em um dado período não pode exceder a quantidade disponível.
- **H6** A cada 2 dias deve ocorrer no máximo uma aula de um evento simples.
- **H7** Aulas de evento simples com carga horária igual a dois devem ser alocados com exatamente um dia de intervalo.
- **H8** Eventos simples devem ter no máximo uma aula em cada dia.
- **H9** Eventos simples devem ser alocados sempre no mesmo horário (obviamente em dias distintos).
- **H10** Eventos geminados devem ser alocados em sequência em num único dia.

Além da factibilidade, é necessário satisfazer tanto quanto possível os *requisitos fracos* a seguir:

- **S1** Minimizar a ocorrência de períodos ociosos em cada etapa.
- **S2** Minimizar a ocorrência de conflitos entre eventos.
- **S3** Deve-se evitar alocar mais de dois eventos da mesma etapa em um dado período. Os eventos que excedem esse limite são chamados *eventos excedentes*.
- **S4** A utilização máxima de recursos em um período deve ser minimizada.

Para cada requisito fraco não atendido está associado um custo na função objetivo, valorado de acordo com a sua importância.

Tabela 1: Notação utilizada pelo modelo

Símbolo	Definição
Conjuntos	
$d \in D$	conjunto de dias da semana.
$h \in H$	conjunto de horários do dia.
$h \in H'$	conjunto dos horários finais de cada turno do dia.
$t \in T$	conjunto de professores.
$e \in E$	conjunto de eventos.
ES	conjunto de eventos simples (aulas distribuídas na semana).
EG	conjunto de eventos geminados (aulas que ocorrem em um único dia da semana).
$r \in R$	conjunto de tipos de recurso.
$m \in M$	conjunto de etapas.
E_t	conjunto de eventos em que o professor t está envolvido.
E_r	conjunto de eventos em que o tipo de recurso r é necessário.
E_m	conjunto de eventos que fazem parte do currículo da etapa m.
Parâmetros	
$W_e \in \mathbb{N}^*$	número de aulas do evento e.
$V_{edh} \in \{0,1\}$	indica se o evento e tem disponibilidade no período (d,h).
$F_{edh} \in \{0,1\}$	indica se o evento e tem uma aula pré-determinada no período (d,h).
$Q_r \in \mathbb{N}^*$	quantidade disponível do tipo de recurso r.
$C_{ij} \in \{0,1\}$	indica se existe um conflito entre os eventos i e j.
$\alpha = 1000$	custo de cada conflito entre eventos.
$\beta = 100$	custo de cada período ocioso.
$\delta = 10$	custo pela utilização de um recurso.
$\gamma = 1$	custo de cada evento excedente.
Variáveis	
$x_{edh} \in \{0,1\}$	define se o evento e está alocado no período (d,h).
$f_{eh} \in \{0,1\}$	indica que todas as aulas do evento e devem ocorrer no horário h.
$g_{edh} \in \{0,1\}$	indica que no período (d,h) começa uma aula geminada.
$b_{mdh} \in \{0,1\}$	indica a ocorrência de um período ocioso em (d,h) na etapa m.
$c_{ijdh} \in \{0,1\}$	indica se os eventos i e j estão alocados no mesmo período (d,h).
$y_{mdh} \in \{0,1\}$	indica se existe no mínimo um evento da etapa m alocado no período (d,h).
$q_{mdh} \in \mathbb{N}$	número de eventos excedentes da etapa m alocados no período (d,h).
$u \in \mathbb{N}$	quantidade máxima de recursos utilizados em um período.

2.1. Formulação matemática do problema

Nesta seção é apresentado uma formulação de programação inteira mista que modela matematicamente todos os requisitos apresentados do problema descrito na seção anterior.

$$\textbf{minimizar} \sum_{m \in m} \sum_{d \in D} \sum_{h \in H} \alpha b_{mdh} + \sum_{i \in E} \sum_{j \in E : i > j} \sum_{d \in D} \sum_{h \in H} \beta c_{ijdh} C_{ij} + \sum_{m \in m} \sum_{d \in D} \sum_{h \in H} \gamma q_{mdh} + \delta u \qquad (1)$$

$$\textbf{s.a.} \sum_{d \in D} \sum_{h \in H} x_{edh} = W_e \qquad \forall e \in E \qquad (2)$$

$$x_{edh} \leq V_{edh} \qquad \forall e \in E, d \in D, h \in H \qquad (3)$$

$$\sum_{e \in E_t} x_{edh} \leq 1 \qquad \forall t \in T, d \in D, h \in H \qquad (4)$$

$$x_{edh} \geq F_{edh} \qquad \forall e \in E, d \in D, h \in H \qquad (5)$$

$$\sum_{e \in E_r} x_{edh} \leq Q_r \qquad \forall r \in R, d \in D, h \in H \qquad (6)$$

$$\sum_{h \in H} (x_{edh} + x_{e,d+1,h}) \leq 1 \qquad \forall e \in ES, d \in D : d < |D| \qquad (7)$$

$$\sum_{h \in H} (x_{eih} + x_{ejh}) \leq 1 \qquad \forall e \in ES, i \in D, j \in D : j \geq i+3, W_e = 2 \qquad (8)$$

$$W_e f_{eh} = \sum_{d \in D} x_{edh} \qquad \forall e \in ES, h \in H \qquad (9)$$

$$g_{edh} \leq x_{edh} \qquad \forall e \in EG, d \in D, h \in H \setminus H' \qquad (10)$$

$$g_{edh} \leq x_{e,d,h+1} \qquad \forall e \in EG, d \in D, h \in H \setminus H' \qquad (11)$$

$$\sum_{d \in D} \sum_{h \in H \setminus H'} g_{edh} = 1 \qquad \forall e \in EG \qquad (12)$$

$$y_{mdh} >= x_{edh} \qquad \forall m \in M, e \in E_m, d \in D, h \in H \qquad (13)$$

$$y_{mdh} <= \sum_{e \in E_m} x_{edh} \qquad \forall m \in M, d \in D, h \in H \qquad (14)$$

$$b_{mdh} >= -1 + y_{mdi} + y_{mdj} - y_{mdh} \qquad \forall m \in M, d \in D, h \in H, i \in H, j \in H : i < h < j \qquad (15)$$

$$x_{idh} + x_{jdh} \leq 1 + c_{ijdh} \qquad \forall i \in E, j \in E, d \in D, h \in H : i > j, C_{ij} = 1 \qquad (16)$$

$$\sum_{e \in E_m} x_{edp} \leq 2 + q_{mdh} \qquad \forall m \in M, d \in D, h \in H \qquad (17)$$

$$\sum_{r \in R} \sum_{e \in E_r} x_{edh} \leq u \qquad \forall d \in D, h \in H \qquad (18)$$

$$x_{edh} \in \{0,1\} \qquad \forall e \in E, d \in D, h \in H$$

$$f_{eh} \in \{0,1\} \qquad \forall e \in E, h \in H$$

$$g_{edh} \in \{0,1\} \qquad \forall e \in EG, d \in D, h \in H \setminus H'$$

$$c_{ijdh} \in \{0,1\} \qquad \forall i \in E, j \in E, d \in D, h \in H : i > j$$

$$y_{mdh} \in \{0,1\} \qquad \forall m \in M, d \in D, h \in H$$

$$b_{mdh} \in \{0,1\} \qquad \forall m \in M, d \in D, h \in H$$

$$q_{mdh} \in \mathbb{N} \qquad \forall m \in M, d \in D, h \in H$$

$$u \in \mathbb{N}$$

O conjunto de restrições (2) garante que o número de aulas de cada evento seja atendido. O conjunto de restrições (3) garante que o evento seja alocado apenas em períodos disponíveis. O conjunto de restrições (4) proíbe que um professor participe de mais de um evento em cada período. O conjunto de restrições (5) garante que as aulas com períodos pré-determinados sejam alocadas. O conjunto de restrições (6) garante que a quantidade disponível em cada tipo de recurso não seja excedida. O conjunto de restrições (7) garante que aulas de um mesmo evento simples não sejam alocados em dias consecutivos afim de atender o requisito H6. O conjunto de restrições (8) impede que exista mais do que um dia livre entre as aulas de eventos simples com apenas 2 aulas. Esta restrição, combinada com o conjunto de restrições (7) modela o requisito H7. O conjunto de restrições (9) determina que todas as aulas de um evento devem ocorrer exatamente no mesmo horário durante a semana. Os conjuntos de restrições (10), (11) e (12) modelam o requisito H9. Os conjuntos de restrições (10) e (11) identificam aulas que ocorrem em sequência. Enquanto, o conjunto de restrições (12) garante que exista apenas uma aula ocorrendo em sequência. Os conjuntos de restrições (13) e (14) garantem que a variável y_{mdh} seja ativada caso exista no mínimo uma aula do semestre m ocorrendo no período (d,h). Os conjuntos de restrições (13), (14) e (15) modelam o requisito S1 de maneira semelhante a modelagem utilizada por Avella et al. (2007) e Dorneles et al. (2012). O conjunto de restrições (16) ativam a variável c_{ijdh} caso os eventos conflitantes i e j sejam alocados no mesmo período (d,h). O conjunto de restrições (17) identificam na variável q_{mdh} o número de eventos excedentes ocorrendo no semestre m no período (d,h). O conjunto de restrições (18) identifica na variável u a quantidade máxima de recursos utilizados em um dado período.

3. Experimentos Computacionais

Nesta seção, é apresentada uma avaliação experimental do modelo matemático apresentado na seção anterior. O objetivo desta seção é responder as seguintes questões:

i) O modelo proposto é apropriado para resolver o problema abordado?

ii) Como as soluções obtidas pelo modelo se comparam com as soluções geradas manualmente?

Para avaliar a modelagem proposta foi utilizado um conjunto formado por três instâncias reais fornecidas pelo departamento do curso de Ciência da Computação do Instituto de Informática da Universidade Federal do Rio Grande do Sul. As instâncias referem-se aos anos de 2011 (2º semestre), 2013 (2º semestre) e 2014 (1º semestre). As suas principais características são apresentadas na Tabela 2. A primeira coluna apresenta a identificação das instâncias. As colunas 2-7 apresentam, respectivamente, o número de dias, períodos, professores, eventos, tipos de recursos e etapas. As últimas colunas apresentam, respectivamente, o total de aulas da instância, o número de eventos conflitantes e a quantidade total de recursos disponíveis. Todas as instâncias estão disponibilizadas em `http://www.inf.ufrgs.br/~fvpneukirchen/tt-ufrgs`.

Tabela 2: Principais características das instâncias.

Instância	$\|D\|$	$\|H\|$	$\|T\|$	$\|E\|$	$\|R\|$	$\|M\|$	$\sum_{e\in E} W_e$	$\sum_{i,j\in E:i>j} C_{ij}$	$\sum_{r\in R} Q_r$
INF2011-2	5	5	49	96	3	7	180	106	16
INF2013-2	5	5	55	103	3	7	191	69	18
INF2014-1	5	5	57	114	3	7	221	85	18

Os resultados dos experimentos foram obtidos em um computador equipado com um processador Intel® Pentium(R) Dual CPU T3400 de 2.16GHz, com 4GB de memória RAM, executando um sistema operacional Linux de 64 bits. Para resolver as instâncias foi utilizado o resolvedor comercial CPLEX na versão 12.5 com suas configurações padrão.

Na Tabela 3 são apresentadas as soluções geradas manualmente pelo departamento de Ciência da Computação da UFRGS em comparação com os resultados obtidos pelo CPLEX utilizando a formulação proposta na seção anterior. As colunas obj_m e obj_c apresentam, respectivamente, o valor objetivo da solução manual e da solução obtida pelo CPLEX. Os requisitos fracos não atendidos na solução são apresentados nas colunas $\sum b$, $\sum c$, $\sum q$ e u que informam, respectivamente, a quantidade de períodos ociosos, o total de conflitos entre eventos, o total de eventos excedentes e o total de recursos utilizados. A coluna *tempo* apresenta o tempo de execução utilizado pelo CPLEX em segundos. A coluna gap_c representa o desvio percentual entre a solução obtida pelo CPLEX e o limitante inferior da solução fornecido pelo CPLEX. A coluna gap_m representa o desvio percentual entre a solução obtida pelo CPLEX e a solução encontrada manualmente. Os valores tabelados nesta coluna são obtidos por $100 * (obj_c - obj_m)/obj_c$. Finalmente, as colunas *#rest* e *#var* apresentam, respectivamente, o número de restrições e de variáveis do modelo informados pelo CPLEX.

Tabela 3: Resultados comparativos entre as soluções manuais e as soluções obtidas pelo CPLEX.

	Soluções Manuais					CPLEX									
Instância	obj_m	$\sum b$	$\sum c$	$\sum q$	u	tempo(s)	obj_c	$\text{gap}_c(\%)$	$\text{gap}_m(\%)$	#rest	#var	$\sum b$	$\sum c$	$\sum q$	u
INF2011-2	1239	3	0	9	12	3,51	904	0	−37,1	15790	8406	0	0	4	9
INF2013-2	5372	3	4	22	13	4,50	905	0	−493,6	14987	7076	0	0	5	9
INF2014-1	11367	4	10	27	13	5,98	1028	0	−1005,7	16229	7566	2	0	8	10

Analisando os resultados da Tabela 3, é possível perceber que o CPLEX obteve soluções factíveis ótimas para as três instâncias consideradas utilizando menos de 10 segundos de processamento. Além disso, o custo das soluções obtidas pelo CPLEX são significativamente menores em comparação ao custo das soluções geradas manualmente. Em média, as soluções obtidas pelo método proposto são 512% melhores que a solução manual. Este ganho expressivo, é devido principalmente a redução do número de eventos em posição de conflitos e da redução de períodos ociosos. Pode-se perceber também que houve uma significativa redução no número total de recursos utilizados e eventos excedentes.

Conforme apresentado na Tabela 2, percebe-se que as instâncias apresentam diferentes dimensões, principalmente, em relação ao número de professores e eventos. As instâncias INF2011-2, INF2013-2 e INF2014-1 estão organizadas em ordem crescente de tamanho. Neste contexto, observa-se que, enquanto a qualidade das soluções obtidas pelo resolvedor não foi influenciada significativamente com o aumento do tamanho das instâncias, a qualidade das soluções obtidas manualmente piorou visivelmente. Este comportamento, no entanto, é esperado, visto que o aumento no tamanho da instância reflete no aumento do espaço de busca, o que torna cada vez mais complexa a criação do horário, em especial, quando realizada por humanos.

Em contato com a Coordenação do Curso, verificou-se que, em termos práticos, os resultados obtidos pelo modelo proposto são muito semelhantes aos obtidos manualmente e poderiam perfeitamente ser utilizados em uma divulgação oficial de horários do curso. Todas as soluções geradas pelo método proposto foram avaliadas qualitativamente pela Coordenação do curso e não foi encontrada nenhuma grade de horários que não satisfizesse os critérios atualmente utilizados para a geração manual dos horários, o que permite afirmar que os conjuntos de restrições utilizados foram corretamente implementados e que são suficientes para atender os critérios da Coordenação.

4. Considerações Finais

Neste trabalho, foi proposto um modelo de programação inteira mista que resolve uma variação do problema de geração de quadros de horários em universidades adaptado ao contexto do curso de Ciência da Computação do Instituto de Informática da Universidade Federal do Rio Grande do Sul. O modelo proposto foi avaliado através de três instâncias reais utilizando um resolvedor comercial de programação inteira mista. Os resultados experimentais demonstraram que é possível resolver o problema de forma bastante eficiente, sendo possível obter soluções ótimas para todas as instâncias. Em comparação com soluções geradas manualmente, a abordagem proposta apresenta soluções significativamente melhores. Portanto, esta abordagem poderá ser usada para resolver novas instâncias desse problema, possibilitando a geração automática de quadros de horários para o Instituto de Informática.

5. Participação dos Autores

Este trabalho é uma extensão de uma pesquisa sobre *school timetabling* conduzida por um aluno de doutorado (segundo autor) publicada em (Dorneles et al., 2012). O primeiro autor deste trabalho, aluno de iniciação científica (IC), adaptou o modelo matemático publicado em (Dorneles et al., 2012), incluindo novas restrições, variáveis e parâmetros necessários para representar adequadamente as particularidades do problema encontrado na CiC/UFRGS. Além das atividades de modelagem, o aluno de IC efetuou todo processo de coleta dos dados, implementação do método e execução dos experimentos. Todas as atividades foram realizadas sob a orientação do segundo e quarto autor. O terceiro autor, coordenador da CiC/UFRGS, auxiliou o aluno de IC tanto no processo de definição do problema como na avaliação dos resultados.

Agradecimentos

Este trabalho foi apoiado pelo departamento de Ciência da Computação do Instituto de Informática da Universidade Federal do Rio Grande do Sul e pelo Programa Petrobras de Formação de Recursos Humanos.

Referências

Avella, P., Salerno, S., e Vasil, I. (2007), A computational study of local search algorithms for Italian high-school timetabling, *Journal of Heuristics*, 543–556.

Burke, E., Jackson, K., Kingston, J. H., e Weare, R. (1997), Automated university timetabling: The state of the art, *The computer journal*, 9(40), 565–571.

Dorneles, Á. P., Araújo, O. C. B., e Buriol, L. S. (2012), The impact of compactness requirements on the resolution of high school timetabling problem, In *Proceedings of XLIV Simpósio Brasileiro de Pesquisa Operacional (SBPO 2012)*, 3336–3347.

Gotlieb, C. (1963), The construction of class-teacher timetables, In *IFIP*, 73–77, Amsterdam. North-Holland.

Petrovic, S. e Burke, E. K. (2004), University timetabling, In *Handbook of scheduling: algorithms, models, and performance analysis*, 1–23, Nottingham. UK.

Schaerf, A. (1999), A survey of automated timetabling, *Artificial Intelligence Review*, 13(2), 87–127.

Zuters, J. (2006), An ensemble of neural networks as part of a ga-based model to solve the school timetabling problem, In *Local proceedings of the 7th International Baltic Conference on Databases and Information Systems*, 175–182, University of Latvia. Latvia.