UMA ABORDAGEM HÍBRIDA DE SAT E BUSCA TABU PARA O PROBLEMA DA PROGRAMAÇÃO DE HORÁRIOS ESCOLARES

George Henrique Godim da Fonseca
Universidade Federal de Ouro Preto
Rua 36, 115 B. Vera Cruz – João Monlevade / MG
george@decea.ufop.br

Rodrigo Geraldo Ribeiro
Universidade Federal de Ouro Preto
Rua 36, 115 B. Vera Cruz – João Monlevade / MG
rodrigogribeiro@decea.ufop.br

Flávio Vinícius Cruzeiro Martins
Universidade Federal de Ouro Preto
Rua 36, 115 B. Vera Cruz – João Monlevade / MG
flavio@decea.ufop.br

RESUMO

O Problema da Programação de Horários Escolares (PPHE) é classificado como NP-Difícil e heurísticas para sua solução são alvo de diversas pesquisas na área de computação, matemática e pesquisa operacional. Normalmente, o problema é resolvido através de abordagens metaheurísticas como Algoritmos Genéticos, *Simulated Annealing* e GRASP. O presente trabalho propõe uma abordagem alternativa. Pretende-se reduzir do problema ao problema da Satisfazibilidade Proposicional (SAT) e resolvê-lo usando um resolvedor SAT para geração de uma solução inicial. Como não é viável tratar todos os requisitos PPHE através de satisfazibilidade, posteriormente, aplicar-se-á uma Busca Tabu para otimização da solução obtida. A eficiência da abordagem é avaliada comparando-a a abordagens conhecidas na literatura.

PALAVARAS CHAVE. Satisfazibilidade Proposicional, Busca Tabu, Programação de Horários Escolares.

Pesquisa Operacional na Educação; Otimização Combinatória; Metaheurísticas.

ABSTRACT

The School Timetabling Problem is classified as NP-Hard and heuristics for its solution are the subject of several researches in computing, mathematics and operational research. Usually the problem is solved by metaheuristics approaches, like Genetic Algorithms, Simulated Annealing and GRASP. This paper proposes an alternative approach. It is intended to reduce the problem to the problem of Propositional Satisfiability (SAT) and solving it in a SAT solver to generate an initial solution. It is not practicable to treat all requirements of school timetabling problem by satisfiability, then will apply a Tabu Aearch to optimize the obtained solution. The efficiency of the approach is evaluated by comparing it to known approaches in literature.

KEYWORDS. Propositional Satisfiability, Tabu Search, School Timetabling Problem.

Operational Research in Education; Combinatorial Optimization; Metaheuristics.

1. Introdução

O problema da programação de horários escolares (PPHE) foi proposto por Gotlieb (1963) e consiste em programar um horário escolar de modo que nenhuma turma ou professor participe de mais de uma aula no mesmo instante de tempo; além disso, o horário deve atender a outras restrições especificadas a priori.

A programação de horários escolares tem sido alvo de diversas pesquisas nas áreas de Teoria da Computação e Pesquisa Operacional. Em Santos e Souza (2007) são apresentadas algumas das razões para este interesse:

- Dificuldade de Resolução: encontrar um quadro de horários que satisfaça todos os interesses envolvidos é uma tarefa difícil, ademais, frequentemente, a simples construção de um quadro de horários válido já é uma tarefa complicada;
- Importância Prática: a confecção de um bom quadro de horários pode melhorar a satisfação do corpo docente e permitir que a instituição de ensino seja mais eficiente na gestão de seus recursos, além do mais, a programação adequada das atividades letivas permite um melhor desempenho dos alunos;
- Importância Teórica: o problema apresentado neste trabalho é classificado como NP-Completo ou NP-Difícil (Garay e Jonhson, 1979) e, progressos na solução de problemas desse tipo são um dos grandes objetivos das pesquisas correntes em computação, matemática e pesquisa operacional.

Normalmente, o problema é resolvido através de abordagens heurísticas, como Busca Tabu (Santos, Ochi e Souza, 2005), *Simullated Annealing*, GRASP (Santos, Ochi, Drummond, e Souza, 2003) e Algoritmos Genéticos (Burke e Petrovic, 2002) (Colorni, Dorigo e Maniezzo, 1998). O presente trabalho pretende realizar a redução do problema ao problema da satisfazibilidade de uma fórmula da lógica proposicional (SAT), submeter a fórmula para resolução usando um resolvedor SAT e otimizar a solução encontrada pelo resolvedor SAT através de uma variante da metaheurística Busca Tabu. Até recentemente, a redução do problema ao SAT não era utilizada devido à ineficiência dos algoritmos para resolução de problemas desse tipo. Porém, na última década, houve um grande avanço no projeto e implementação de resolvedores SAT, hoje capazes de resolver fórmulas com centenas de milhares de variáveis de maneira eficiente (Biere, Heule, Maaren e Walsh, 2009).

O restante do trabalho é organizado da seguinte maneira: na Seção 2 será descrito o problema da programação de horários escolares bem como sua definição formal, na Seção 3 será descrito o problema da Satisfazibilidade Proposicional e as ferramentas utilizadas para sua solução, além da metaheurística Busca Tabu. Na Seção 4 será explicitada a solução proposta, a codificação da fórmula da lógica proposicional e a variante da Busca Tabu implementada. Na Seção 5 serão apresentados os experimentos computacionais e na Seção 6, as considerações finais.

2. Problema da Programação de Horários Escolares

Segundo Santos e Souza (2007), o Problema de Programação de Horários Escolares (PPHE), também conhecido na literatura como Problema de Programação de Horários Professor × Turma (PPT) é um problema clássico de otimização combinatória sendo provavelmente o problema mais conhecido dos problemas de programação de horários. Nesse problema, cada professor deve lecionar um determinado número de aulas para cada turma em um conjunto de períodos. As alocações devem ser feitas considerando-se a não existência de conflitos, ou seja, cada professor e turma deve se envolver com, no máximo, uma atividade letiva por período. Há ainda várias outras restrições que o horário deve atender.

2.1. Restrições

No PPHE temos duas categorias de restrições: restrições fortes e restrições fracas que serão descritas a seguir.

2.1.1. Restrições Fortes

Restrições desse tipo devem ser satisfeitas a qualquer custo, visto que não é possível a implementação de um quadro de horários que não as satisfaça. As restrições fortes determinam o espaço de busca que será considerado: somente soluções que respeitam todas as restrições desse tipo são consideradas factíveis; se o problema considerado consiste em obter uma solução factível qualquer, então esse denomina-se problema de busca (Santos e Souza, 2007). As restrições fortes do PPHE consideradas são apresentadas a seguir (Souza, Maculan e Ochi, 2004):

a) Não permitir a alocação de mais de um professor em uma mesma classe no mesmo horário;
b) Não permitir a alocação de mais de uma classe a um mesmo professor no mesmo horário;
c) A carga horária de cada classe deve ser atendida;
d) Atender às restrições de horário de professores.

2.1.2. Restrições Fracas

Restrições desse tipo são aquelas cuja satisfação é desejável, mas caso não seja possível respeitá-las, pode-se, ainda assim, implementar o quadro de horários. O atendimento das restrições fracas é a medida de qualidade de um quadro de horários; se o problema considerado consiste em encontrar uma solução que minimize a violação das restrições fracas, tem-se então um problema de otimização (Santos e Souza, 2007). As restrições fracas do PPHE consideradas são apresentadas a seguir (Souza, Maculan e Ochi, 2004):

a) Atender ao número requerido de aulas geminadas de cada classe;
b) Respeitar o limite diário de aulas para cada classe;
c) Evitar "janelas" nos horários dos professores;
d) Minimizar o número de dias que cada professor deverá vir à escola.

2.2. Definição Formal

Tomaremos como base a definição formal do problema de construção de horários escolares apresentada por Neufeld e Tartar (1974) e descrita a seguir. Dados:

a) Um conjunto de professores P = $\{p_i\}$, $1 \leq i \leq \alpha$;
b) Um conjunto de classes C = $\{c_j\}$, $1 \leq j \leq \beta$;
c) Um conjunto de horários semanais H = h_k, $1 \leq k \leq \sigma$, $\sigma = \lambda \cdot \mu$, onde λ é o número de dias semanais letivos e μ é o número de horários por dia;
d) Uma matriz $\alpha \times \beta$ de requisitos R = $[r_{ij}]$, onde $r_{ij} \geq 0$ representa o número de encontros semanais do professor p_i com a classe c_j;
e) Uma matriz $\alpha \times \sigma$ de restrições de indisponibilidade de professores D = $[d_{ik}]$, onde d_{ik} = 1 se o professor p_i está indisponível no horário h_k; caso contrário, $d_{ik} = 0$.

Uma solução do problema de programação de horários escolares é uma matriz $\alpha \times \beta \times \sigma$, F = $[f_{ijk}]$, onde f_{ijk} = 1 se o professor p_i leciona para a classe c_j no horário h_k, que atenda as restrições fortes.

3. Revisão da Literatura

3.1. Satisfazibilidade Proposicional

O problema da satisfazibilidade proposicional (SAT) consiste em determinar uma atribuição de valores às variáveis de uma fórmula da lógica proposicional de maneira a tornar esta fórmula verdadeira. Usualmente instâncias deste problema são representadas por fórmulas na forma normal conjuntiva, que possui a seguinte definição indutiva:

- F e T são fórmulas na forma normal conjuntiva
- Os literais α e $\neg\alpha$ são fórmulas na forma normal conjuntiva, onde α é uma variável da lógica proposicional.
- Uma disjunção $l_1 \vee l_2 \vee ... \vee l_n$, onde cada l_i é um literal, é uma fórmula na forma normal conjuntiva. Dá-se o nome de cláusula a uma disjunção de literais.
- Se α e σ são duas fórmulas na forma normal conjuntiva, então $\alpha \wedge \sigma$ é uma fórmula na forma normal conjuntiva. Uma fórmula na forma normal conjuntiva consiste de uma conjunção de cláusulas.

3.2. Algoritmo DPLL

A maioria dos resolvedores SAT, inclusive o utilizado no trabalho, implementam o algoritmo DPLL para solução do problema da satisfazibilizadade proposicional.

A idéia básica do algoritmo é a de construir uma valoração para uma fórmula fornecida como um conjunto de cláusulas. Inicialmente, todos os literais recebem a valoração "*", representando valor indefinido. A cada iteração do algoritmo, um literal L é escolhido, e faz-se $v(L) = 1$. Com essa valoração, procede-se à simplificação da fórmula. Se essa valoração satisfizer todas as cláusulas, tem-se uma valoração que satisfaz a fórmula inicial. Se alguma cláusula for falsificada pela valoração, altera-se a escolha para $v(L) = 0$. Se nenhuma cláusula for falsificada, nem todas as cláusulas foram satisfeitas, procede-se à próxima escolha de literal. O processo para quando uma valoração for encontrada, em cujo caso a fórmula é satisfazível, ou quando não há literais para serem testados, em cujo caso a fórmula é insatisfazível (Biere, Heule, Maaren e Walsh, 2009).

3.3. Busca Tabu

Conforme definição de Souza (2008), a Busca Tabu é um método de busca local que consiste em explorar o espaço de soluções movendo-se de uma solução para outra que seja seu melhor vizinho. Esta estratégia, juntamente com uma estrutura de memória para armazenar as soluções geradas (ou características destas) permite que a busca não fique presa em um ótimo local. A implementação desenvolvida da Busca Tabu será descrita na Seção 4.5.

4. Solução Proposta

Uma das principais noções da teoria da computabilidade é a redução de problemas. Dizemos que um problema p_1 é redutível a um problema p_2 se existe um algoritmo M que transforma instâncias do problema p_1 em instâncias do problema p_2 de maneira que o resultado de p_2 é igual ou complementar ao resultado de p_1 para a instância em questão (Papadimitriou, 1994). No presente trabalho realizar-se-á a redução do problema de programar um horário escolar ao problema SAT através das codificações descritas nas seções 4.3 e 4.4.

Tratar as restrições fracas *c* e *d* descritas na Seção 2.1.2 através da redução ao problema SAT leva a um número muito grande de cláusulas, inviabilizando o uso da abordagem para tal; então, propõe-se utilizar o resolvedor SAT para gerar uma solução inicial e aplicar uma variante da Busca Tabu para otimizar a solução obtida pelo mesmo.

Nem sempre há uma solução que atenda a todos os requisitos fortes e os requisitos fracos *a* e *b* simultaneamente com custo 0. Situações como essa levariam a uma fórmula insatisfazível. Para tratar essas situações, adicionalmente, foi incluída uma repetição no algoritmo que relaxa as restrições do problema caso esse leve a uma fórmula insatisfazível. O pseudocódigo abaixo resume a abordagem proposta.

```
procedimento gerarHorarios(problema, maxTentativas)
    i ← 0;
    enquanto(i < maxTentativas)
        formula ← codificacao(problema);
        solucao ← resolvedor(formula);
        se existe solucao então  {A fórmula é satisfazível}
            s* ← buscaTabu(solucao);
            retorne s*;
        senão
            remova alguma restrição aleatória de problema;
            i ← i + 1;
        fim-se;
    fim-enquanto;
fim gerarHorarios
```

Figura 01: Abordagem Proposta

4.1. Definição das Variáveis

No presente trabalho, um literal $x_{d,h,e,p,c}$, unidade componente da fórmula da lógica proposicional é composto de:

- Um dia *d*;
- Um horário *h*;
- Um encontro *e* entre a classe *c* e o professor *p*;
- Um professor $p \in$ P;
- Uma classe $c \in$ C.

Um dado literal pode receber o valor 1, indicando a alocação da classe *c* com o professor *p* no encontro *e*, ao horário *h* do dia *d*; ou 0, representando a não alocação. Por exemplo, o literal $x_{2,4,1,p1,c1}$ indica que a classe c_1 terá o encontro (aula) 1 com o professor p_1 no horário 4 do dia 2. Para ilustrar as fórmulas, as variáveis podem ter alguns componentes não especificados, por exemplo, $x_{e,p,c}$ representa uma variável de dia e horário ainda não especificados.

Além da representação do literal as seguintes funções e predicados serão definidos para representar algumas relações que ocorrem frequentemente:

- **numEncontros(p, c)**

Seja *p* um professor e *c* uma classe, *numEncontros*(*p*, *c*) representa o número de encontros requeridos pela classe *c* com o professor *p*;

- **mesmoProfessor(p, p')**
  Sejam *p* e *p'* dois professores.

$$mesmoProfessor(p, p') = \begin{cases} \text{T, se os professores } p \text{ e } p' \text{ são o mesmo professor;} \\ \text{F, caso contrário.} \end{cases}$$

- **mesmoPeríodo(dh, dh')**
  Sejam *dh* e *dh'* dois pares dia-horário.

$$mesmoPer\acute{\imath}odo\ (dh, dh') = \begin{cases} \text{T, se os pares dia-horário } dh \text{ e } dh' \text{ são iguais;} \\ \text{F, caso contrário.} \end{cases}$$

- **mesmoCurso(c, c')**
  Sejam *c* e *c'* duas classes.

$$mesmoCurso(c,\ c') = \begin{cases} \text{T, se as classes } c \text{ e } c' \text{ pertencem ao mesmo curso} \\ \quad \text{(grade horária);} \\ \text{F, caso contrário.} \end{cases}$$

- **professorIndisponível(p, d, h)**
  Seja *p* um professor, *d* um dia e *h* um horário.

$$professorIndispon\acute{\imath}vel(p, d, h) = \begin{cases} \text{T, se o professor } p \text{ está indisponível no horário} \\ \quad h \text{ do dia } d; \\ \text{F, caso contrário.} \end{cases}$$

- **horariosDistintos(d, n)**
  Retorna *n* horários distintos $h_1, h_2, ..., h_n$ no dia *d*.

- **encontrosDistintos(p, c, n)**
  Retorna *n* encontros distintos $e_1, e_2, ..., e_n$, entre o professor *p* e a classe *c*.

- **minAulasGem(c)**
  Retorna o número mínimo de aulas geminadas requerido pela classe *c*.

- **limDiárioAulas(c)**
  Retorna o limite diário de aulas da classe *c*.

4.2. Notação Utilizada

A notação utilizada para a representação das fórmulas da lógica proposicional é a mesma da utilizada em Fat (2004). Para exemplificar a notação, consideremos a seguinte fórmula na forma normal conjuntiva:

$$(x_{1,1} \vee x_{1,3}) \wedge (x_{2,1} \vee x_{2,3}) \wedge (x_{3,1} \vee x_{3,3})$$

Onde o primeiro componente de cada variável representa uma classe *c* do conjunto C = {1, 2, 3} e o segundo, um horário *h* do conjunto H = {1, 2, 3}. A mesma fórmula, na notação utilizada, considerando o predicado *ímpar*(*x*) verdadeiro se *x* é ímpar, é representada por:

$$\bigwedge_{c \in C} \ \bigvee_{h \in H \mid \acute{\imath}mpar(h)} x_{c,h}$$

4.3. Codificação das Restrições Fortes

A codificação de cada restrição descrita na Seção 2.1.1 é dada a seguir:

4.3.1. Não permitir a alocação de um professor a mais de uma classe no mesmo horário

Para modelar essa restrição, adiciona-se predicados do tipo se o professor *p* é alocado ao horário *h* do dia *d*, então nenhuma das outras aulas ministradas pelo professor *p* pode ser alocada no horário *h* do dia *d*.

Pela notação definida na Seção 4.2, temos:

$$\bigwedge_{x_{dhepc}} \quad \bigwedge_{x'_{dhepc} \mid mesmoProfessor(p,\, p') \,\wedge\, mesmoPeríodo(dh,\, dh')} \neg x_{d,h,e,p,c} \vee \neg x'_{d,h,e,p,c}$$

4.3.2. Não permitir a alocação de classes pertencentes ao mesmo curso no mesmo horário

Para atender a esse requisito, deve-se adicionar predicados do tipo se a classe *c* for alocada ao horário *h* do dia *d*, então nenhuma das outras classes pertencentes ao mesmo curso pode ser alocada no horário *h* do dia *d*. Formalmente, temos:

$$\bigwedge_{x_{dhepc}} \quad \bigwedge_{x'_{dhepc} \mid mesmoCurso\,(c,\, c') \,\wedge\, mesmoPeríodo(dh,\, dh')} \neg x_{d,h,e,p,c} \vee \neg x'_{d,h,e,p,c}$$

4.3.3. A carga horária de cada classe deve ser atendida

Essa restrição é codificada adicionando-se para cada encontro, de cada classe, uma disjunção de todos os horários possíveis, de modo a garantir que ao menos um deles deva ser assinado com T. A codificação abaixo ilustra a situação:

$$\bigwedge_{x_{epc}} \ \bigvee_{d \,\in\, \mathrm{D}} \ \bigvee_{h \,\in\, \mathrm{H}} x_{d,h,e,p,c}$$

4.3.4. Atender às restrições de horário de professores

Eventualmente, professores têm outros compromissos, como cursos de aperfeiçoamento ou exercício da docência em outra instituição para complementar a renda, sendo impossível que compareçam à escola em determinados horários. Essas restrições são codificadas adicionando-se a negação dos literais relacionados ao professor com os horários de indisponibilidade.

$$\bigwedge_{x_{dhepc} \mid professorIndisponível(p,d,h)} \neg x_{d,h,e,p,c}$$

4.4. Codificação das Restrições Fracas

A codificação das restrições *a* e *b* descritas na Seção 2.1.2. são dadas a seguir.

4.4.1. Atender ao número requerido de aulas geminadas de cada classe

Muitas classes requerem um determinado número de aulas geminadas para possibilitar um aprendizado melhor do conteúdo lecionado. Para atender ao número requerido de aulas geminadas de cada classe, adiciona-se à fórmula cláusulas do tipo se um encontro da classe foi alocado no horário *h*, então, outro encontro da classe será alocado no horário *h* + 1. O inverso também é válido, conforme a codificação abaixo:

$$\bigwedge_{x_{pc}} \ \bigwedge_{d \,\in\, \mathrm{D}} \ \bigwedge_{\mathrm{g} = 1 \text{ até } minAulasGem(\mathrm{c})} \ \bigwedge_{h = 1 \text{ até } |\mathrm{H}| - 1} \begin{array}{l} \neg x_{d,h,g*2+1,p,c} \vee x_{d,h+1,\, g*2+2,p,c} \ \wedge \\ x_{d,h,g*2+1,p,c} \vee \neg x_{d,h+1,\, g*2+2,p,c} \end{array}$$

4.4.2. Respeitar o limite diário de aulas

Para algumas classes é importante não concentrar muitas aulas num dado dia uma vez que seu conteúdo é cansativo ou exige muito esforço dos estudantes, comprometendo o aprendizado. Para respeitar o limite diário de aulas de cada classe, dado que o limite diário de aulas da classe *c* é *l*, adiciona-se cláusulas do tipo se *l* encontros da classe *c* foram alocados no dia *d*, então o encontro *l* + 1 da classe c não pode ser alocado no dia *d*. Com o intuito de simplificar a fórmula, foram empregadas as seguintes abreviaturas: *hD*(*d*, *n*) ≡ *horáriosDistintos*(*d*, *n*), *eD*(*p*, *c*, *n*) ≡ *encontrosDistintos*(*p*, *c*, *n*) e *limDA*(*c*) ≡ *limDiarioAulas*(*c*).

$$\bigwedge_{x_{pc}} \bigwedge_{d \in \mathrm{D}} \bigwedge_{h_1,h_2,\ldots,h_n = hD(\mathrm{d},\, limDA(\mathrm{c}))} \bigwedge_{e_1,e_2,\ldots,e_n = \mathrm{eD}(\mathrm{p},\, \mathrm{c},\, limDA(\mathrm{c}))} x_{d,h1,e1,p,c} \vee x_{d,h2,e2,p,c} \ldots \vee \neg x_{d,hn,en,p,c}$$

4.5. Implementação da Busca Tabu

Ao utilizar lógica proposicional para gerar a solução inicial, garante-se que caso haja uma solução que atenda a todos os requisitos tratados com custo 0, ela será encontrada. A partir dessa solução inicial gerada pelo resolvedor SAT, a metaheurística Busca Tabu pretende explorar a vizinhança da solução de modo a encontrar vizinhos de custo menor no que tange as restrições fracas *a*, *c* e *d*. A restrição fraca *b*, assim como em Santos (2006), é tratada como uma restrição forte. É importante também que os outros requisitos já atendidos continuem sendo atendidos. A variação da busca tabu desenvolvida é apresentada no pseudocódigo da figura a seguir:

```
procedimento BT(s, f(.), BTmax, |T|, A(.))
    s* ← s;             {Melhor solução obtida até então}
    Iter ← 0;           {Contador do número de iterações}
    MelhorIter ← 0;     {Iteração mais recente que forneceu s*}
    T ← ∅;              {Lista Tabu}
    enquanto (Iter - MelhorIter ≤ BTmax) faça
        Iter ← Iter + 1;
        Seja s' ← s ⊕ m o melhor elemento de V ⊆ N(s) tal que
            o movimento m não infrinja as restrições fortes a, b, c e d e
            o movimento m não infrinja a restrição fraca b e
            o movimento m não seja tabu (m ∉ T) ou
            s' atenda a condição de aspiração (f(s') < A(f(s)));
        Atualize a lista tabu T;
        s ← s';
        se (f(s) < f(s*)) então
            s* ← s;
            MelhorIter ← Iter;
        fim-se;
        Atualize a função de aspiração A;
    fim-enquanto;
    s ← s*;
    Retorne s;
fim BT;
```

Figura 02: Implementação da Busca Tabu

No algoritmo da Figura 02, *s* representa a solução encontrada pelo resolvedor SAT, *f*(.), a função objetivo que será descrita detalhadamente na Seção a seguir, BT_{Max} = 100, o limite de iterações sem melhora aceitado pela Busca Tabu, |*T*| = 15, o comprimento da lista tabu e *A*(.), a função de aspiração. *A*(.) aceita um movimento tabu apenas se ele conduzir a um vizinho melhor que *s** ($A(f(s)) = f(s^*)$).

5. Resultados Experimentais

Para avaliação dos resultados obtidos foram consideradas duas variáveis: tempo levado para alcance da solução e qualidade da solução encontrada. A qualidade da solução é definida pela função objetivo a seguir.

5.1. Função Objetivo

A programação de horários escolares é um problema de decisão multicritério porque, para determinar a qualidade de uma programação, faz-se necessário considerar diferentes objetivos (Souza, Guimarães e Costa, 2002). Para avaliar uma programação de horários escolares, os requisitos do problema são classicamente organizados em requisitos essenciais e requisitos não-essenciais, equivalentes, respectivamente a restrições fortes e fracas já definidas anteriormente.

Desse modo, uma programação (ou solução) s pode ser medida com base em duas componentes, uma de inviabilidade (*g*(*s*)), a qual mede o não atendimento aos requisitos essenciais, e outra de qualidade (*h*(*s*)), a qual mede o não atendimento aos requisitos considerados não essenciais. A função de custo de uma solução *s*, a qual deve ser minimizada, pode ser calculada, portanto, pela seguinte expressão:

$$\boldsymbol{f(s) = g(s) + h(s)}$$

A parcela *g*(*s*), que mensura o nível de inviabilidade de uma solução *s*, é avaliada com base na expressão:

$$g(s) = \sum_{k=1}^{K} \alpha_k I_k$$

onde *K* é número de medidas de inviabilidade, I_k o valor da k-ésima medida de inviabilidade e α_k o peso associado à essa k-ésima medida.

A parcela *h*(*s*), que mensura a qualidade de uma solução *s*, é avaliada com base na seguinte função:

$$h(s) = \sum_{l=1}^{L} \beta_l Q_l$$

onde *L* representa o número de medidas de qualidade, Q_l o valor da l-ésima medida de qualidade e β_l o peso associado a essa l-ésima medida. Deve ser observado que uma solução *s* é viável se e somente se *g*(*s*) = 0. Nas componentes da função *f*(*s*) os pesos dados às diversas medidas refletem a importância relativa de cada uma delas (Souza, Guimarães e Costa, 2002).

Assumimos aqui, o peso associado a todas as medidas de inviabilidade é 50 e o peso associado às restrições fracas, *a*, *b*, *c* e *d*, respectivamente 10, 5, 3 e 1. As restrições *a* e *b* receberam uma importância maior, uma vez que no Instituto para o qual a solução foi direcionada, os aspectos pedagógicos do quadro de horários recebem uma importância maior que os aspectos de atendimento ao corpo docente. Seguir-se-á os critérios para a contagem de violação de cada restrição definidos em Santos (2006).

5.2. Testes Realizados

Para efeito de teste do protótipo desenvolvido, foram considerados os dados referentes ao Instituto de Ciência Exatas e Aplicadas da Universidade Federal de Ouro Preto (ICEA/UFOP) para o 1º semestre de 2011. O mesmo possui aproximadamente cinquenta docentes e cento e dezessete classes, agrupadas em períodos. O Instituto funciona de segunda a sexta e as aulas são distribuídas em dez horários de cinquenta minutos cada.

Além dos dados para construção de horários do ICEA, foram utilizadas instâncias de problemas de construção de horários escolares disponíveis no sítio do laboratório de inteligência computacional da Universidade Federal Fluminense (http://labic.ic.uff.br/).

Todos os testes foram realizados em um computador AMD Athlon(tm) Dual Core 2.2 Ghz com 1GB de Memória RAM sobre sistema operacional Windows XP Professional. A linguagem de programação utilizada para desenvolvimento do software foi Java, que possibilita a incorporação do SAT4J (Berre e Parrain, 2010), resolvedor SAT escolhido para compor o projeto.

Os resultados obtidos pelo software desenvolvido, considerando a melhor solução encontrada dentre cinco execuções, são apresentados na tabela a seguir, onde NRH representa o número de restrições de horários de professores; TSAT, o tempo de processamento levado pelo resolvedor SAT; e TBT, o tempo de processamento levado pela implementação da Busca Tabu:

Instância	**Número de Classes**	**NRH**	**TSAT (seg.)**	**TBT (seg.)**	***g*(*s*)**	***h*(*s*)**
1	21	40	0,102	0,146	0	68
2	63	25	0,086	1,150	0	114
3	69	80	0,417	2,103	0	151
4	127	170	1,185	11,284	0	220
5	119	0	0,315	12,961	0	274
6	140	10	1,390	14,410	0	271
7	205	0	0,503	69,804	0	375
ICEA	117	95	1,987	46,069	0	356

Tabela 01: Resultados Obtidos pelo Software

Pode-se observar que, em todas as instâncias, as soluções obtidas têm custo 0 de inviabilidade, ou seja, satisfazem todos os requisitos essenciais. As soluções apresentaram também um custo baixo de qualidade, proporcional à complexidade de cada instância. Outro aspecto relevante é o tempo de processamento, que foi muito pequeno em todas as instâncias.

5.3. Trabalhos Relacionados

Em Fat (2004) utilizou-se resolvedores SAT para a solução de um modelo do problema de programação de horários escolares semelhante ao aqui apresentado, entretanto alguns requisitos importantes não foram tratados, como minimização do número de dias que cada professor tem que vir à escola e minimização de janelas no horário dos professores. Do ponto de vista computacional os resultados obtidos não foram muito animadores: não raro, o tempo de processamento passava de 2 horas para instâncias não muito grandes.

Outro relato relevante da aplicação de Satisfizibilidade Proposicional na programação de horários escolares pode ser encontrado em Achá e Nieuwenhuis (2010), onde, reduzindo o PPHE ao problema *Weighted Partial* Max SAT (Anstotegui, Bonet, e Levy, 2010) obteve-se um resultado expressivo: a melhor solução conhecida para 10 das 32 instâncias abordadas.

Em Santos (2006) é apresentada uma abordagem de Programação Linear Inteira Mista para resolver o problema apresentado. Na referida abordagem obteve-se resultados significativos, principalmente para as instâncias com um número reduzido de classes. Para três das instâncias abordadas foram encontradas as soluções ótimas. A tabela abaixo compara os resultados obtidos aos de Santos (2006) considerando os mesmos pesos para as restrições:

Instância	Tempo Total (seg.)			Resultado Obtido ($f(s)$)		
	Santos (2006)	**Presente Trabalho**	**Diferença (%)**	**Santos (2006)**	**Presente Trabalho**	**Diferença (%)**
1	0,5	0,2	150,0	202	204	0,1
2	11,4	1,2	850,0	333	351	5,4
3	4,8	2,5	560,0	423	472	11,5
4	91,2	12,4	635,5	653	669	2,4
5	179,4	13,2	1259,1	766	834	8,8
6	231,1	15,8	1362,7	760	825	8,5
7	2678,3	70,3	3709,8	1029	1152	11,9

Tabela 02: Comparação de Resultados com Santos (2006)

O trabalho de Santos (2006) foi implementado em C++ utilizando o pacote CPLEX 10 sobre um computador Dell Optiplex G620 com processador Pentium D 3.0Ghz e 2GB de memória RAM, rodando o sistema operacional Linux.

Apesar das soluções obtidas por Santos (2006) possuírem um custo menor que as observadas nesse trabalho, o tempo de processamento das instâncias, mesmo numa plataforma mais poderosa, foi consideravelmente maior que o tempo demandado na presente abordagem. Outro aspecto relevante é o fato de o trabalho de Santos (2006) utilizar-se do CPLEX, uma ferramenta proprietária de otimização em detrimento do presente trabalho que se utiliza apenas de componentes gratuitos de software.

6. Conclusões

Neste trabalho descreveu-se como resolver o problema da programação de horários escolares utilizando uma abordagem híbrida de SAT e Busca Tabu ainda não explorada na literatura.

Os resultados preliminares são promissores já que todas as soluções encontradas foram viáveis e mantiveram-se com um custo próximo dos custos apresentados por Santos (2006). O tempo demandado para obtenção das soluções foi um diferencial, em torno de 10 vezes mais rápido que em outras abordagens, o que indica que a utilização de resolvedoes SAT para geração de soluções iniciais em conjunto com heurísticas de otimização para refinamento das soluções é viável para resolver o problema da programação de horários escolares.

Como possíveis trabalhos futuros, pretende-se submeter a mesma abordagem a outros problemas variantes do PPHE, além de outros problemas de agendamento em geral. Pretende-se adicionalmente, propor codificações mais eficientes para o problema e testar o uso de outras metaheurísticas como Algoritmos Genéticos e *Simulated Annealing* para refinar a solução encontrada pelo resolvedor SAT.

Referências

Achá, R. A. e Nieuwenhuis, R., Curriculum-based Course Timetabling with SAT and MaxSAT. In: *Practice and Theory of Automated Timetabling (PATAT)*, 2010, Belfast. Proceedings of the 8th International Conference on the Practice and Theory of Automated Timetabling. Belfast, UK, 2010. v. 1. p. 42-57.

Anstotegui, C., Bonet, M. L. e Levy, J., A New Algorithm forWeighted Partial MaxSAT. In: *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010, Atlanta, USA.

Berre, D. L., Parrain, A., The Sat4j library, release 2.2. In: *Journal on Satisfiability, Boolean Modeling and Computation*, 2010, p. 59-64.

Biere, A., Heule, M., Maaren, H. V. e Walsh, T., *A history of Satisfiability*. In: Handbook of Satisfiability, 2009. IOS Press, Amsterdan, Nethelands. p. 3-55.

Biere, A., Heule, M., Maaren, H. V. e Walsh, T., *Satisfiability by Search: The DPLL Algorithm*. In: Handbook of Satisfiability, 2009. IOS Press, Amsterdan, Nethelands. p. 110-114.

Burke, E. K. e Petrovik, S., *Recent directions in automated timetabling*. European Journal of Operational Research, Volume 140, Issue 2. 2002.
Colorni, A., Dorigo, M., Maniezzo, V., *Metaheuristics for high school timetabling. Computational Optimization and Applications*, Volume 3, Issue 3, 1998.
Fat K. C. A., *School Timetabling using Satisfiability Solvers*. 2004. Dissertação de Mestrado. Delft University of Technology, Delft, Nethelands.
Garey, M. R. e Jonhson, D. S., *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, CA, USA, 1979.
Gotlieb, C. C., The construction of class-teacher time-tables. *Proc. IFIP Congress 62*, Munich, North Holland Pub. Co., Amsterdam, 1963, pp. 73-77.
Neufeld, G. A. e Tartar, J., Graph coloring conditions for the existence of solutions to the timetabling problem. *Communications of the ACM*, Volume 17, Issue 8, 1974.
Papadimitriou, C. H., *Computational Complexity*, 1994. Addison-Wesley Publishing Company, San Diego, USA.
Santos, H. G., *Formulações e Algoritmos para o Problema da Programação de Horários em Escolas*. 2006. Tese (Doutorado em Computação). Universidade Federal Fluminense, Niterói, Rio de Janeiro.
Santos, H. G., Ochi, L. S., Drummond, L. M. A. e Souza, M. J. F., GRASP with Data Mining for The School Timetabling Problem. In: *Two Days on Combinatorial Optimization: A Bridge Between Rio and Niterói*, 2003, Rio de Janeiro. Two Days on Combinatorial Optimization: A Bridge Between Rio and Niterói, 2003.
Santos, H. G., Ochi, L. S. e Souza, M. J. F., *A Tabu search heuristic with efficient diversification strategies for the class/teacher timetabling problem*. ACM Journal of Experimental Algorithmics, v. 10, p. 2.9, 2005.
Santos, H. G. e Souza M. J. F., Programação de horários em instituições educacionais: formulações e algoritmos. *XXXIX Simpósio Brasileiro de Pesquisa Operacional, 2007*, Fortaleza. Anais do XXXIX SBPO. Rio de Janeiro : SOBRAPO, 2007. v. 1. p. 2827-2882.
Souza, M. J. F., *Inteligência Computacional para Otimização*. 2008. (Desenvolvimento de material didático ou instrucional - Material didático).
Souza, M. J. F., Guimarães, I. F. e Costa, F. P., Um algoritmo evolutivo híbrido para o problema de programação de horários em escolas. In: *XXII Encontro Nacional de Engenharia de Produção*, 2002, Curitiba. Anais do XXII ENEGEP. Santa Bárbara D´Oeste : ABEPRO, 2002. v. 1. p. 1-8.
Souza, M. J. F., Maculan, N. e Ochi, L. S., A GRASP-Tabu Search Algorithm for Solving School Timetabling Problems. In: Maurício G. C. Resende; Jorge Pinho de Sousa. (Org.). *METAHEURISTICS: Computer Decision-Making.* Dordrech: Kluwer Academic Publishers, 2004, v. 86, p. 659-672.