IX Encontro de Modelagem Computacional
Belo Horizonte, MG
15 a 17 de novembro de 2006
CEFET-MG - IPRJ/UERJ
ISBN: 978-85-99836-02-6

UM ESTUDO SOBRE METAHEURÍSTICAS APLICADAS AO PROBLEMA DE HORÁRIO ESCOLAR

Alessandra Martins Coelho - ale_contatos@ig.com.br
Sérgio Ricardo de Souza - sergio@dppg.cefetmg.br
Marcelo Caramuru Pimentel Fraga- caramuru@hotmail.com
Centro Federal de Educação Tecnológica de Minas Gerais - CEFETMG - DPPG
30510-000 Av. Amazonas, 7675 - Nova Gameleira - Belo Horizonte - MG - Brasil

Resumo. *Este artigo apresenta um estudo sobre aplicação das metaheurísticas* Simulated Annealing, Iterated Local Search *e Busca Tabu ao Problema de Horário Escolar. O objetivo é a geração de quadros de horários para professores, em dois turnos simultaneamente. As metodologias propostas foram testadas utilizando-se dados reais do Centro Federal de Educação Tecnológica de Rio Pomba-MG. Os resultados computacionais são comparados dentre as metaheurísticas propostas e com os quadros de horários produzidos manualmente pela própria instituição de ensino.*

Palavras-Chave: *Problema de Horário Escolar; Metaheurísticas.*

1 INTRODUÇÃO

O Problema de Horário Escolar, também referenciado como problema classe-professor, pode admitir uma enorme variedade de formulações, em função da instituição de ensino e do sistema educacional no qual ela está inserida. A maior parte destas formulações parte da premissa de que, para a ocorrência de uma determinada aula, é necessário que todas as entidades envolvidas (professores, alunos e salas) estejam disponíveis no intervalo de tempo definido para o acontecimento da mesma; que nenhuma entidade individual (professores, alunos ou salas) esteja alocada a dois eventos simultaneamente; e que a carga-horária de cada disciplina, para cada classe, seja cumprida. Para maiores detalhes quanto a essas formulações, veja Schaerf (1999).

Devido à dificuldade de se obter uma solução manual, à característica combinatória e à difícil generalização (em virtude da quantidade de variantes que o problema pode assumir), diversos algoritmos e heurísticas vêm sendo propostas, com o objetivo de tentar resolver diferentes aspectos do problema. Dentre esses métodos, cita-se as metaheurísticas *simulated annealing*, em Abramson et al. (1999), Busca Tabu, em Hertz (1992) e Algoritmos Genéticos, em Burke et al. (1995). Esse artigo é organizado como segue: a seção seguinte apresenta o problema abordado. A seção 3 revisa as metaheurísticas adotadas na solução do problema, enquanto a seção 4 mostra a aplicação das mesmas ao problema em tela. A seção 5 apresenta os resultados alcançados do desenvolvimento realizado e seção 6 finalizada o artigo, apresentando conclusões a respeito do trabalho.

2 O PROBLEMA DE HORÁRIO EDUCACIONAL ABORDADO

O Centro Federal de Educação Tecnológica de Rio Pomba - CEFET-RP, situado à Av. Dr. José Sebastião da Paixão, s/n, bairro Lindo Vale, Rio Pomba, Minas Gerais, é a instituição de ensino considerada para análise. Essa instituição de ensino oferece, atualmente, os cursos de ensino médio, ensino técnico e ensino superior (nível tecnológico) e funciona em três turnos: manhã, tarde e noite. Os cursos de ensino médio e técnicos possuem uma carga-horária semanal de 20 horas-aula, distribuídas uniformemente em um único turno, de segunda a sexta-feira. Entre a segunda e a terceira aula, há um intervalo de 20 minutos, para o turno diurno, e de 10 minutos para o turno noturno. As aulas dos cursos de nível superior geralmente são programadas para o período diurno.

No processo de programação da agenda dos professores dessa instituição de ensino, deve-se observar que alguns cursos possuem características específicas. Em todos os casos, além dos requisitos mínimos essenciais para que um horário de qualquer instituição de ensino possa ser utilizado, quais sejam, não ocorrer sobreposição de professores ou classes; cumprimento da programação da carga-horária semanal de cada disciplina; e respeito aos horários de indisponibilidade do professor, são observados os seguintes requisitos:

Requisitos não-essenciais comuns à programação de horários dos professores da instituição de ensino:

- as aulas dos professores devem ser programadas de tal modo que exista pelo menos um dia da semana livre das atividades de sala de aula;
- o número de horários ociosos na programação diária dos professores deve ser reduzido.
- deve-se evitar que um professor lecione mais de uma disciplina por dia, para uma mesma classe;

Requisitos observados para a programação de horários dos professores do Ensino Médio:

- Essenciais
  - as aulas de educação-física são geminadas e realizadas nos dois últimos horários;
  - uma classe não pode ter mais que duas aulas diárias de uma mesma disciplina;
- Não-Essenciais
  - deve-se evitar a quebra de aulas. Considera-se como quebra de aulas a ocorrência de aulas para uma mesma classe, separadas por um intervalo;

Requisitos observados para a programação de horários dos professores dos cursos Técnico em Informática e Técnico em Meio-Ambiente:

- Essenciais
  - uma classe não pode ter mais que duas aulas diárias de uma mesma disciplina;
  - as aulas são sempre geminadas;
  - deve-se evitar que um professor dê mais de uma disciplina por dia para uma mesma classe.

Requisitos observados para a programação de horários dos professores dos cursos técnicos da área de Agropecuária, Técnico em Secretariado e Técnico em Gestão do Agronegócio:

- Essencial
  - não é permitida a ocorrência de quebras de aula;

No entanto, esses cursos admitem, dependendo da disciplina, qualquer configuração de aula, ou seja, admitem a configuração de $1, 2, 3$ ou 4 aulas diárias.

Além disso, os cursos técnicos da área de agropecuária (agropecuária, agroindústria e zootecnia) possuem um dia da semana reservado para treinamento operacional. Neste caso específico, deve-se evitar que seja programado para os horários de segunda-feira de manhã, para as turmas do turno da manhã e os horários da sexta-feira à tarde para as turmas do turno da tarde.

Em se tratando do professor, este pode lecionar várias disciplinas e estar envolvido em mais de um curso. Devido à modalidade de ensino, alguns professores podem acumular mais aulas durante um certo período do ano.

Em geral, antes da elaboração dos quadros de horários, os professores se reúnem para escolher as turmas as quais eles preferem lecionar, no caso do ensino médio, e as disciplinas que eles preferem lecionar, no caso dos demais cursos. Os professores efetivos têm preferência na escolha das turmas e disciplinas. Já no caso dos professores substitutos, procura-se atender às suas solicitações quanto aos dias de indisponibilidade e solicitação do número de dias de distribuição de suas aulas.

Pelo fato de haver vários cursos com características específicas, o planejador de horários do CEFET-RP decompõe o problema de alocação dos professores às turmas em vários subproblemas. Cada subproblema diz respeito a uma modalidade de curso. Os professores do curso técnico em Informática, por ter maior necessidade do uso dos laboratórios de informática, confeccionam o quadro de horários e o encaminha para o setor pedagógico.

De posse dos horários disponíveis dos laboratórios de informática em cada turno e dos horários disponíveis dos professores, o setor pedagógico informa aos responsáveis pelos cursos técnico em Meio-Ambiente, técnico em Gestão do Agronegócio e técnico em Secretariado, que confeccionam seus quadros de horários e os encaminham ao setor pedagógico.

Para o curso de ensino médio, é utilizado um programa para a confecção do primeiro quadro de horários dos professores. Eventuais modificações são feitas manualmente.

Os responsáveis pelos cursos de nível superior também encaminham os quadros de horários para o referido setor.

O planejador de horários, de posse desses quadros de horários, faz a alocação dos professores às turmas dos demais cursos. Em todos os casos, é observado o horário de indisponibilidade e/ou solicitação de folga dos professores.

No decorrer do ano letivo, os horários das aulas podem sofrer alterações, por motivo de afastamento de algum professor.

3 METAHEURÍSTICAS BASEADAS EM TRAJETÓRIA

3.1 *Simulated annealing*

Simulated annealing (SA) é uma metaheurística que realiza uma busca local de propósito geral, inspirada no processo de recozimento de materiais. Sua introdução é devida aos trabalhos de Kirkpatrick et al. (1983).

O processo inicia-se com uma solução s qualquer. Uma busca aleatória é realizada no espaço da vizinhaça de s. Para cada vizinho s' encontrado, verifica-se o valor de Δ, dado pela diferença entre o valor da função objetivo de s' e o valor da função objetvo de s. Se Δ for menor que 0, a solução é aceita e torna-se a solução corrente; caso contrário, a solução poderá ser aceita com uma certa probabilidade, dada por $Prob = e^{-\Delta/T}$, sendo T o parâmetro de controle da temperatura.

Tipicamente, a metaheurística é iniciada com uma temperatura alta. A temperatura é reduzida lentamente, através de uma expressão de resfriamento, até encontrar uma região específica. A probabilidade de se aceitar movimentos de piora em temperaturas elevadas é bem alta. A probabilidade de aceitação de movimentos de piora diminui à medida que a temperatura decresce. Com o valor da temperatura baixo, o processo praticamente não aceita movimentos de piora e o procedimento passa a se comportar como o método da descida, até a temperatura estar próxima de zero, situação na qual o sistema é congelado. O sistema também pode congelar-se quando a taxa de aceitação de movimentos cair abaixo de um valor predeterminado ou quando o valor da função objetivo ficar inalterado por um determinado número de temperaturas.

3.2 *Iterated local search*

A metaheurística *Iterated Local Search* (ILS), idealizada por Baxter (1981), conforme descrito em Lourenço et al. (2002), é considerada como o esquema mais geral entre as estratégias explorativas. Baseia-se na idéia de aplicar uma busca local em uma solução inicial qualquer, até que se encontre um ótimo local, e, então, perturbar a solução encontrada e reiniciar a busca local. Essa perturbação deve ser tal que possibilite a manutenção de características da região do ótimo local e, além disso, deve evitar um reinício aleatório. Por outro lado, a perturbação deve ser de tal monta que seja suficiente para escapar de um ótimo local e permitir a exploração de outras regiões do espaço de buscas. ILS é, portanto, um método de busca local, que procura focar a busca não no espaço completo de soluções, mas em um subespaço definido por soluções que são ótimos locais de determinado procedimento de otimização.

3.3 Busca tabu

A metaheurística Busca Tabu (BT) foi proposta independentemente por Glover (1986) e Hansen (1986). Trata-se de um procedimento adaptativo, dotado de uma estrutura de memória, que aceita movimentos de piora para escapar de ótimos locais.

Partindo de uma solução inicial qualquer, a cada iteração do método, o espaço de soluções é explorado em busca de seu melhor vizinho. O melhor vizinho encontrado torna-se, então, a solução corrente, mesmo que este piore o valor da melhor solução encontrada até o momento. O processo é repetido até que o critério de parada adotado seja satisfeito.

Aceitando-se soluções de piora como melhor vizinho, o procedimento procura escapar de ótimos locais encontrados durante a sua execução, podendo, entretanto, gerar ciclos, fazendo com que se retorne a uma solução já gerada anteriormente. Isto acontece pois, ao se realizar uma busca na vizinhança de uma solução gerada a partir de um movimento de piora, o melhor vizinho encontrado será a solução anterior, que deu origem à atual, fazendo com que o procedimento entre em ciclo.

Para reduzir o risco da ciclagem, o método trabalha com uma lista, denominada lista tabu, em que se armazenam os movimentos que, quando realizados, levam a uma solução já analisada anteriormente, os quais são denominados de movimentos tabus.

A lista tabu clássica contém os movimentos reversos aos últimos $|T|$ movimentos realizados (sendo $|T|$ um parâmetro do método) e funciona como uma fila de tamanho fixo, isto é, quando um novo movimento é inserido à lista, o movimento mais antigo armazenado sai. Dessa maneira, na exploração do subconjunto da vizinhança da solução corrente s, ficam excluídos da busca os vizinhos s' que são obtidos de s por movimentos m que constam na lista tabu.

O tamanho $|T|$ da lista tabu define por quantas iterações um movimento será considerado tabu. Uma lista tabu de tamanho muito grande pode ser restritiva, uma vez que torna tabu um grande número de movimentos. Isso faz com que o procedimento termine de forma prematura, pois não encontra opções de movimentos a realizar, já que a maioria deles está definido como tabu. Já uma lista de tamanho pequeno aumenta a possibilidade de ciclagem, uma vez que reduz o intervalo de tempo no qual um movimento é considerado tabu.

4 APLICAÇÕES DE HEURÍSTICAS AO PROBLEMA DE HORÁRIO ESCOLAR

As subseções a seguir apresentam, em detalhes, como as heurísticas apresentadas anteriormente foram aplicadas ao problema proposto.

4.1 Representação de uma solução

Uma solução s é representada por uma matriz $Q_{m \times p}$ de valores inteiros, na qual cada linha i representa a alocação semanal de aulas do professor i. Cada elemento q_{ik} indica a atividade do professor i no horário k. Os valores atribuídos a cada elemento devem pertencer ao conjunto $\{-2, -1, 0, 1, 2, 3, \ldots, \infty\}$, de modo que, caso $q_{ik} = -2$, o professor i, no horário k, está indisponível para a atividade de aula, mas poderá assumir carga horária, se necessário; caso $q_{ik} = -1$, o professor i, no horário k, está indisponível para a atividade de aulas e não poderá, em hipótese alguma, assumir encargos de aula. Caso $0 \leq q_{ik} \geq n$, o professor i está alocado para ministrar aulas para alguma classe no horário k. Caso, por fim, $q_{ik} = \infty$, o professor está disponível para a atividade de aulas, mas não possui aulas alocadas no horário k. Uma representação da matriz Q está apresentada na tabela 1.

Tabela 1: Representação do Quadro de Horários

	Semana Letiva																			
Prof.	Segunda				Terça				Quarta				Quinta				Sexta			
	h0	h1	h2	h3	h4	h5	h6	h7	h8	h9	h10	h11	h12	h13	h14	h15	h16	h17	h18	h19
1	-1	-1	1	1	-1	-1	5	5	-1	-1	-1	-1	-1	-1	0	0	-1	-1	4	4
2	-1	-1	11	11	-1	-1	∞	∞	-1	-1	9	9	-1	-1	∞	∞	-1	-1	-1	-1
3	-1	-1	-1	-1	∞	∞	∞	∞	∞	∞	∞	∞	4	4	∞	∞	∞	4	4	4
4	9	0	∞	∞	∞	∞	∞	11	∞	9	4	1	5	11	∞	5	4	∞	1	0
.																				
.																				
.																				
m	4	∞	∞	11	∞	∞	∞	∞	∞	∞	11	∞	9	9	∞	5	-1	-1	-1	-1

4.2 Estrutura de vizinhança

Uma solução s' é dita vizinha de s se for obtida a partir de um movimento de realocação ou troca de aulas em s.

Dois movimentos de realocação foram definidos para a obtenção da vizinhança: o primeiro realiza, em cada linha i da matriz Q, a troca de dois valores distintos e não negativos. Esse movimento é identificado pela tripla $< i, k_1, k_2 >$, na qual k_1 e k_2 representam os horários nos quais as atividades q_{ik_1} e q_{ik_2} do professor i serão permutadas.

O segundo movimento realiza, em cada linha i da matriz Q com aulas obrigatoriamente duplas, a troca de dois valores duplos distintos e não negativos. Este movimento é identificado pela tripla $< i, k_{1a}k_{1b}, k_{2a}k_{2b} >$, na qual $k_{1a}k_{1b}$ e $k_{2a}k_{2b}$ representam os horários nos quais as atividades $q_{ik_{1a}}q_{ik_{1b}}$ e $q_{ik_{2a}}q_{ik_{2b}}$ do professor i serão permutadas.

4.3 Geração da solução inicial

Optou-se pela geração de soluções iniciais aleatórias para diversificar as soluções. Um dia da semana é selecionado aleatoriamente e a aula alocada no primeiro horário disponível daquele dia.

4.4 Função de avaliação

Uma solução s é avaliada com base na função:

$$f(Q) = \omega f_1(Q) + \delta f_2(Q) + \rho f_3(Q) \tag{1}$$

A função $f_1(Q)$ é representada pelo somatório do número de vezes que um professor ministra aulas para mais de uma classe no horário k, somada ao número de vezes que uma turma está sem atividade nesse mesmo horário k. A função $f_2(Q)$ representa o somatório do número de vezes em que é violado o número máximo de aulas diário de uma mesma disciplina. Por fim, a função $f_3(Q)$ representa o somatório de vezes que as solicitações dos professores não são atendidas, ou seja, representa a ocorrência de horários ociosos na programação diária do professor; se existem lacunas entre duas aulas programadas para uma mesma classe, no mesmo dia; se os dias solicitados pelo professor para a distribuição de suas aulas é desrespeitado; e se o número de aulas duplas (geminadas) solicitadas pelo professor ou curso não é atendido. Deve-se observar que uma solução s somente será viável se $f_1(Q)$ e $f_2(Q)$ forem iguais a zero.

A tabela 2 apresenta os parâmetros efetivamente utilizados na função de avaliação.

Tabela 2: Pesos para função de avaliação

Peso	**Valor**
ω	100
δ	50
ρ	1
α	2
β	5
γ	10
σ	1

4.5 Aplicação de *simulated annealing*

A temperatura inicial utilizada no SA foi gerada por simulação. O procedimento inicia partindo de um valor baixo para a temperatura. A cada temperatura, um número de vizinhos são gerados e verifica-se a quantidade de soluções que são aceitas. Se o número de vizinhos aceitos for alto, como por exemplo 90%, a temperatura corrente é retornada como temperatura inicial; caso contrário, deve-se aumentar a temperatura e repetir o processo.

As figuras 1 e 2 apresentam, respectivamente, o pseudocódigo do SA e o pseudocódigo do procedimento para encontrar a temperatura inicial.

Os parâmetros utilizados no SA são descritos na tabela 3.

4.6 Aplicação de *iterated local search*

Considerou-se quatro níveis de perturbação. Em seu nível mais baixo, uma perturbação consiste em uma troca aleatória de um valor distinto e não negativo, simples ou duplo, por outro valor, distinto e não negativo, simples ou duplo. A segunda perturbação realiza o procedimento por duas vezes e assim sucessivamente. Logo após a perturbação, a solução passa por uma busca

procedimento $SimulatedAnnealing(f(\cdot), N(\cdot), \alpha, iterMax, Tfinal, s)$
$s^* \leftarrow$ s; // melhor solução obtida até então
iterT $\leftarrow$ 0; //número de iterações na temperatura T
T $\leftarrow$ TemperaturaInicial;
enquanto (T $>$ Tfinal) **faca**
  enquanto (iterT $<$ iterMax) **faca**
    iterT $\leftarrow$ iterT +1;
    Gere Um Vizinho Qualquer $s' \in N(s)$;
    $\Delta = f(s') - f(s)$;
    se ($\Delta < 0$) **entao**
      $s \leftarrow s'$;
      se($f(s') < f(s^*)$) **entao**
      $s^* \leftarrow s'$;
    senao
      tome x $\in [0, 1]$
      se (x $< e^{-\Delta/T}$) **entao**
      $s^* \leftarrow s'$;
    fim-se;
  fim-enquanto;
  $T_k = \alpha T_{k-1}$;
  iterT $\leftarrow$ 0;
fim-enquanto;
$s \leftarrow s^*$;
retorne s;
fim $SimulatedAnnealing$;

Figura 1: Algoritmo *Simulated Annealing*

Tabela 3: Parâmetros utilizados no *Simulated Annealing*

Parâmetros	**Valor**	**Descrição**
α	0.99	taxa de resfriamento
γ	0.8	porcentagem de soluções aceitas no algoritmo de simulação da temperatura
δ	1.1	acréscimo da temperatura no algoritmo de simulação da temperatura
Tfinal	0.01	temperatura final do SA
Tinicial	4	temperatura inicial utilizada no algoritmo de simulação da temperatura
SAmax	$(p \times Hs \times 2) \times nTurmas \times Dens$	número de iterações em cada temperatura

local. Caso a solução encontrada após a busca local seja pior que a solução anterior, o movimento é desfeito, o nível de perturbação cresce e uma nova perturbação é aplicada à solução. Caso contrário, a solução encontrada passa a ser a solução corrente e o nível de perturbação atual é aplicado. Considera-se, como solução corrente, a melhor solução encontrada até então. O procedimento pára após um certo número de iterações.

Para a escolha de uma busca local para ser utilizada com o ILS, algumas heurísticas de descida foram testadas e seus comportamentos observados. Dentre essas, o método de descida aleatório não-ascendente mostrou-se o mais apropriado para ser utilizado com o problema de horário em questão, por aceitar movimentos laterais. O pseudocódigo de ILS é apresentado na figura 3.

```
procedimento TemperaturaInicial(δ, γ, SAmax, T_0, s)
    T ← T_0; // temperatura corrente
    continua ← TRUE;
    enquanto (continua) faca
        aceitos ← 0; //quantidade de vizinhos aceitos na temperatura T
        para iterT = 1 ate SAmax faca;
            Gere Um Vizinho Qualquer s' ∈ N(s);
            Δ = f(s') − f(s);
            se (Δ < 0) entao
                aceitos = aceitos + 1;
            senao
                s* ← s';
            senao
                tome x ∈ [0, 1]
                se (x < e^(−Δ/T)) entao
                aceitos = aceitos + 1;
            fim-se;
        fim-para;
        se (aceitos ≥ γ × SAmax) entao
            continua ← FALSE;
        senao
            T ← δ × T; //aumenta a temperatura
        fim-se;
    fim-enquanto;
    Retorne s;
fim TemperaturaInicial;
```

Figura 2: Procedimento Temperatura Inicial

4.7 Aplicação de busca tabu

A implementação realizada de BT introduz o uso de listas dinâmicas, com o fim de minimizar a possibilidade de ciclagem associada à dificuldade de definição do tamanho da lista tabu. Uma lista labu dinâmica tem seu tamanho alterado periodicamente durante a execução do método. Além disso, utilizou-se um critério de aspiração que retira, sob certas circunstâncias, o *status* tabu de um determinado movimento, de forma que uma solução s' vizinha de s pode ser gerada se $f(s') < (f(s))$, mesmo que o movimento m esteja na lista tabu. Para tal, aceita-se um movimento tabu se este levar a uma solução s' cujo valor seja melhor que o valor da melhor solução s^* encontrada até o momento.

Os parâmetros efetivamente utilizados no algoritmo Busca Tabu são descritos na tabela 4, enquanto o pseudocódigo associado é apresentado na figura 4.

5 RESULTADOS

Os algoritmos foram implementados na linguagem C++, usando o compilador C++ Builder 6.0, e testado em microcomputadores Pentium IV, 2.4 GHz, com 256MB de memória RAM, com sistema operacional Windows NT.

```
procedimento ILS
s ← solucaoInicialAleatoria;
s ← buscaLocal(s);
s* ← s;
enquanto (iter < iterMax) faca
    iter++;
    faca
        escolha um professor aleatoriamente;
        se (obrigatorioDuplas) faca
            s’ ← vizinhoDuplas;
        senao
            s′ ← vizinhoSimples;
        fim-se;
        perturbacao++;
    enquanto (perturbacao < perturbacaoMax);
    s″ ← buscaLocal(s′);
    se (s″ < s*) entao
        s* ← s″;
    senao
        s″ ← s;
        se (perturbacaoMax < 4)
            perturbacaoMax++;
        senao
            perturbacaoMax ← 1;
        fim-se;
    fim-se;
    perturbacao ← 0;
fim-enquanto
retorne s*;
fim ReiniciaPopulacao
```

Figura 3: Pseudocódigo ILS

Tabela 4: Parâmetros utilizados no *Busca Tabu*

Parâmetros	**Valor**	**Descrição**
Tmin	10	tamanho mínimo da lista tabu
Tmax	15	tamanho máximo da lista tabu
$BTmax$	40	número máximo de iterações sem melhora no valor da melhor solução

5.1 Problemas-teste

Considerou-se um conjunto de problemas-teste relativos à programação de horários dos cursos de Ensino Médio e Técnico do Centro Federal de Educação Tecnológica de Rio Pomba, nos anos de 2005 e 2006. Apresenta-se na tabela 5 algumas de suas características.

As colunas p, nTurmas e Aulas representam, respectivamente, o número de professores, o número de classes e as aulas a serem alocadas. A coluna Duplas indica o número de aulas

procedimento $BuscaTabu(f(.), \mathcal{N}(.), a(.), |\mathcal{V}|, f_{min}, |T|, BTmax, s)$
$s^\star \leftarrow s$; // Melhor solução obtida até então
$Iter \leftarrow 0$; //número de iterações
$MelhorIter \leftarrow 0$; //Iteração mais recente que forneceu $s^\star$
$T \leftarrow \emptyset$; // lista tabu
Inicialize a função de aspiração a;
enquanto ($f(s) > f_{min}$ e $Iter - MelhorIter < BTmax$) **faca**
    $Iter \leftarrow Iter + 1$;
    se $s' \leftarrow s \oplus m$ for o melhor elemento de $\mathcal{V} \subset \mathcal{N}(s)$ tal que
        o movimento m não seja tabu $(m \notin T)$ **ou**
        s' atenda a condição de aspiração $(f(s') < a(f(s)))$ **entao**
        atualize lista tabu T;
    se$(Iter - MelhorIter > Titer)$ **entao**
        se $(f(s) < f(s^\star))$ **entao**
            $s^\star \leftarrow s$;
            $MelhorIter \leftarrow Iter$;
    $|T| = aleatorio(Tmin, Tmax)$;
    $s \leftarrow s'$;
fim-enquanto;
$s \leftarrow s^\star$;
retorne s;
fim $BuscaTabu$;

Figura 4: Pseudo-código do Método Busca Tabu

Tabela 5: Características dos problemas testados

Problema-Teste	p	nClasses	Aulas	Duplas	Dens
1	17	7	140	70	0,38
2	17	12	240	67	0,69
3	18	12	240	53	0,61

duplas requeridas em cada problema. A coluna Dens indica a densidade do quadro de horários dos professores, ou seja, a razão entre a carga horária dos mesmos, somada ao número de horários indisponíveis, e o número de horários reservados para a programação das aulas de todas as classes. Esses dados referem-se à programação dos quadros de horários de dois turnos em simultâneo. No problema 1, todas as aulas são geminadas. Nos demais problemas considera-se que todos os professores com mais de duas horas-aula semanais para uma mesma classe gostariam de ter, pelo menos, uma aula geminada. Cada classe possui 20 aulas semanais, que são distribuídas uniformemente de segunda a sexta-feira.

Para efeito de comparação dos resultados obtidos pelos algoritmos analisados, o tempo médio gasto pelo algoritmo *Simulated Annealing* em 50 execuções de cada problema-teste, serviu como parâmetro para os demais algoritmos.

A tabela 6 apresenta a porcentagem de soluções viáveis que cada algoritmo conseguiu encontrar, sendo a busca limitada aos seguintes tempos de execução, em segundos, para cada problema: (1)19 s, (2)174 s e (3)304 s. Apesar do problema-teste 1 ser aparentemente mais simples, pelo fato de possuir um número reduzido de turmas e professores e apresentar uma densidade menor, os algoritmos SA e ILS não conseguiram 100% das soluções viáveis, nas 50 execuções. No entanto, o algoritmo BT conseguiu atender satisfatoriamente aos requisitos de todos os problemas-teste.

Tabela 6: Capacidade de encontrar soluções viáveis (%)

	Problemas-teste		
Algoritmo	**1**	**2**	**3**
Simulated Annealing	86	100	100
Iterated Local Search	96	100	100
Busca Tabu	100	100	100

Tabela 7: Tempo médio gasto para encontrar a primeira solução viável (s)

	Problemas-teste					
	1		**2**		**3**	
Algoritmo	fo	tempo	fo	tempo	fo	tempo
Simulated Annealing	6,40	7,17	179,14	46,28	217,46	70,48
Iterated Local Search	4,60	3,75	107,72	12,56	80,64	25,58
Busca Tabu	4,20	1,62	40,20	15,51	86,40	11,69

A média dos valores da função objetivo, para os primeiros quadros de horários viáveis encontrados e o tempo médio gasto pelos algoritmos, são apresentados na tabela 7. Ao comparar o valor da solução e o tempo despendido pelo algoritmo, pode-se dizer que o algoritmo BT apresenta o melhor desempenho, na busca pela primeira solução viável.

Tabela 8: Comparação entre Média das soluções finais - Solução Manual

	Problemas-teste		
Algoritmo	**1**	**2**	**3**
Simulated Annealing	16,18	48,52	70,84
Iterated Local Search	4,04	26,22	32,28
Busca Tabu	0	34,6	25,6
Solução Manual	0	112	132

A tabela 8 apresenta a média dos valores da função objetivo das soluções finais obtidas pelos algoritmos e o valor da função objetivo para os quadros de horários produzidos manualmente pelo CEFET-RP. Nessa tabela, pode-se observar que, para o problema-teste 1, para o qual a solução manual consegue atender a todos os objetivos, somente o algoritmo BT consegue atendê-los. Com relação aos demais problemas-teste, os três algoritmos conseguiram resultados superiores aos quadros de horários produzidos manualmente, representando uma melhoria acima de 56% para o problema 2 e acima de 46% para o problema (3).

De acordo com os resultados apresentados nas tabelas 7 e 8, no que se refere ao valor das soluções, pode-se observar que, de acordo com o tempo de execução adotado, o SA conseguiu melhorar a qualidade das soluções finais em $72,92\%$, no problema 1 e $67,42\%$, no problema 2. O ILS melhorou a primeira solução viável em $75,66\%$, no problema 1 e $59,97\%$, no problema 2. Por fim, o BT conseguiu uma melhora com relação à primeira solução viável de $13,93\%$, no problema 1 e $70,37\%$, no problema 2.

6 CONCLUSÃO

Este trabalho realizou uma comparação entre as metaheurísticas *Simulated Annealing*, *Iterated Local Search* e Busca Tabu aplicados ao Problema de Horário Escolar. Foram realizados testes com dados reais do CEFET-RP para a geração dos quadros de horários dos professores, em dois turnos em simultâneo, utilizando dados de 2005 e 2006. Dos algoritmos propostos, *Iterated Local Search* e Busca Tabu conseguiram melhores resultados que o algoritmo *Simulated Annealing*. A média dos resultados encontrados pelos algoritmos foram superiores quando comparados com o valor dos quadros de horários produzidos manualmente, uma vez que conseguiram satisfazer a um maior número de professores quanto à minimização do número de dias de distribuição de suas aulas; minimização do número de horários ociosos na agenda diária do professor; redução do número de quebras de aulas e maximização do número de aulas duplas, representando uma melhoria acima de 56% para o problema (2) e acima de 46% para o problema (3).

REFERÊNCIAS

Abramson, D., Dang, H., & Krisnamoorthy, M., 1999. Simulated annealing cooling schedules for the school timetabling problem. *Asia-Pacific Journal of Operational Research*, vol. 16, pp. 1–22.

Baxter, J., 1981. Local optima avoidance in depot location. *Journal of the Operational Research Society*, vol. 32, pp. 815–819.

Burke, E. K., Elliman, D., & Weare, R. F., 1995. A hybrid genetic algorithm for highly constrained timetabling problems. In *Proceedings of the 6th International Conference on Genetic Algorithms*, pp. 605–610, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Glover, F., 1986. Future paths for integer programming and links to artificial intelligence. vol. 5, pp. 553–549.

Hansen, P., 1986. The steepest ascent mildest descent heuristic for combinatorial programming. In *Proceedings of Congress on Numerical Methods in Combinatorial Optimization*, Capri, Itália.

Hertz, A., 1992. Finding a fesible course schedule using tabu search. *Discrete applied mathematics*, vol. 35, pp. 255–270.

Kirkpatrick, S., Gellat, D., & Vecchi, M., 1983. Optimization by Simulated Annealing. *Science*, vol. 220, pp. 671–680.

Lourenço, H. R., Martin, O., & Stuetzle, T., 2002. Iterated local search. In *In: F. Glover and G. Kochenberger, editors, Handbook of Metaheuristics*, pp. 321–353, Norwell, MA. Kluwer Academic Publishers.

Schaerf, A., 1999. A survey of automated timetabling. *Artificial Intelligence*, vol. 13, n. 2, pp. 87–127.

Abstract. *This article shows a study about the application of the metaheuristics Simulated Annealing, Iterated Local Search and Tabu Search for solving the school timetabling problem, in two turns, in simultaneous way, satisfying a set of constraints. The methodology was tested with real data, provided by Centro Federal de Educação Tecnológica de Rio Pomba-MG. Computational results are compared between the used metaheuristics, mainly, and the timetabling produced manually by the teaching institution.*

Keywords: School Timetabling; Metaheuristics.