

APLICAÇÃO DA METAHEURÍSTICA BUSCA TABU AO PROBLEMA DE ALOCAÇÃO DE AULAS A SALAS EM UMA INSTITUIÇÃO UNIVERSITÁRIA

APPLICATION OF THE TABU SEARCH METAHEURISTIC TO THE CLASSROOM ASSIGNMENT PROBLEM IN A UNIVERSITY INSTITUTE

Resumo

Este artigo trata do Problema de Alocação de Aulas a Salas de uma Instituição Universitária. Na instituição analisada, a resolução deste problema é feita manualmente, tornando o processo árduo e demorado, além de frequentemente não produzir soluções que atendam a todas as restrições do problema. Desta forma, faz-se necessário automatizar o processo de alocação e, além disso, recorrer a estratégias computacionais que proporcionem soluções de qualidade e baixo custo. Devido à natureza combinatória do problema, recorreu-se à metaheurística Busca Tabu, que tem se mostrado adequada para a resolução desta classe de problemas. O algoritmo proposto parte de uma solução inicial gerada por um procedimento construtivo, o qual é capaz de produzir soluções viáveis em menos de um segundo. A seguir, esta solução é refinada pela Busca Tabu usando-se movimentos de realocação e troca de aulas entre salas para explorar o espaço de busca. O algoritmo proposto foi testado usando-se dados relativos à alocação de aulas de um semestre letivo e demonstrou ser bastante eficiente, tendo gerado soluções de alta qualidade quando comparado com a solução manual.

Palavras-chave: Problema de Alocação de Aulas a Salas, Otimização, Metaheurística, Busca Tabu, Instituição Universitária.

Abstract

This paper deals with the Classroom Assignment Problem in a University Institute. In the analyzed case, this problem is being resolved manually, which is an arduous and prolonged procedure, as well as often not capable of producing solutions that satisfy all the constraints. Thus there is a necessity to automatize the assignment process and to resort to computational strategies capable of yielding quality solutions at low costs. Due to the problem's combinatorial nature, use was made of the metaheuristic Tabu Search which has proven satisfactory for resolving these kinds of problems. The proposed method starts from an initial solution generated by a constructive procedure capable of producing feasible solutions in less than a second. Next, this solution is improved by the Tabu Search by applying movements that reallocate and exchange classes between classrooms in order to explore the solution space. The proposed algorithm was tested utilizing data related to the classroom assignment of a given semester and has demonstrated to be very effective, generating high quality solutions when compared to the manual solution.

Keywords: Classroom Assignment Problem, Optimization, Metaheuristic, Tabu Search, University Institution.

1. INTRODUÇÃO

O Problema de Alocação de Aulas a Salas (PAAS) ou *Classroom Assignment Problem* consiste em alocar aulas, com horários de início e término previamente programados, a um número fixo de salas (CARTER & LAPORTE, 1998; SCHAERF, 1999; SOUZA *et al.*, 2002). Esse é um problema típico que surge nas instituições universitárias antes do início dos semestres letivos. Boa parte dessas instituições ainda resolve tal problema manualmente, o que torna o processo árduo e demorado, podendo levar vários dias para ser concluído. Ressalta-se ainda a sua importância quando se trabalha num contexto onde há escassez de salas de aula, que surge naturalmente com o crescimento do número de cursos e alunos da instituição.

O PAAS é um problema clássico de otimização combinatória pertencente à classe NP-hard (CARTER & TOVEY, 1992), em que a determinação da solução ótima do problema, em um período de tempo aceitável, não é uma tarefa simples. Os métodos exatos, empregados na resolução de problemas que fazem parte desta classe, chegam a consumir tempos de ordem exponencial, ainda que sejam de dimensões medianas. Portanto, a utilização exclusiva de algoritmos exatos se torna praticamente inviável. Em vista disso, se faz necessário recorrer a outras técnicas na tentativa de se obter uma solução de qualidade, isto é, próxima à solução ótima e em tempos computacionais baixos. As técnicas heurísticas, de maneira geral, possuem esse perfil, sendo as metaheurísticas as mais indicadas por serem capazes de escapar de ótimos locais (SOUZA *et al.*, 2002).

O presente trabalho propõe um algoritmo heurístico para resolver o PAAS em uma instituição universitária com o objetivo de se obter soluções com alto grau de satisfação e baixo custo computacional. O algoritmo desenvolvido é baseado na metaheurística Busca Tabu tendo em vista seu desempenho satisfatório na resolução de várias classes de problemas de programação de horários (WHITE *et al.*, 2004; SANTOS *et al.*, 2004; ALVAREZ-VALDÉS *et al.*, 2001; HERTZ, 1991).

O restante do artigo está organizado da forma que se segue. A Seção 2 apresenta uma breve revisão bibliográfica sobre a Busca Tabu, destacando alguns trabalhos, relacionados ao PAAS, que fizeram uso desta metaheurística. A Seção 3 caracteriza o problema abordado neste trabalho. A Seção 4 traz os métodos e procedimentos utilizados. A Seção 5 discute os resultados computacionais obtidos. A Seção 6 apresenta as considerações finais deste artigo.

2. REVISÃO BIBLIOGRÁFICA

De acordo com Schaerf (1999), o PAAS é uma variante do problema básico de programação de horários de cursos universitários (*course timetabling*). Recentemente, problemas desta natureza vêm sendo tratados por meio de técnicas heurísticas, em particular, as metaheurísticas. Dentre as mais utilizadas, pode-se citar: *Simulated Annealing*, Algoritmos Genéticos, Colônia de Formigas e Busca Tabu, cujas descrições podem ser encontradas em Glover e Kochenberger (2004). Esta última metaheurística tem sido frequentemente empregada com êxito na resolução de problemas do gênero em questão. Hertz (1991), Carter e Tovey (1992), Costa (1994), Colomi *et al.* (1998), Schaerf (1999), Alvarez-Valdés *et al.* (2001), Souza *et al.* (2002), Willemen (2002), Santos *et al.* (2004), White *et al.* (2004) e Azimi (2005) obtiveram resultados significativos na resolução de problemas relacionados à programação de horários de instituições de ensino, bem como suas variantes, através da implementação deste método.

2.1 BUSCA TABU

A metaheurística Busca Tabu (BT) foi inicialmente desenvolvida por Glover (1986) como uma proposta de solução para problemas de programação inteira. A partir de então, o autor formalizou esta técnica e publicou uma série de trabalhos contendo diversas aplicações da mesma. A experiência tem mostrado a eficiência da Busca Tabu na resolução de vários problemas de diferentes naturezas (GLOVER & LAGUNA, 1993) e atualmente pode-se afirmar que se trata de uma técnica definitivamente consolidada.

Em linhas gerais, a Busca Tabu é um procedimento adaptativo, de busca local, dotado de uma estrutura de memória, que aceita movimentos de piora (quando não há possibilidades de melhora) para escapar de ótimos locais (HIGGINS, 2001; SOUZA, 2000). Sendo um procedimento de busca local, é baseado na noção de vizinhança. A cada iteração, a solução atual s muda para outra que seja sua vizinha no espaço de busca, isto é, para uma solução s' que difere de s por uma modificação. Partindo de uma solução inicial s_0 , um algoritmo Busca Tabu explora, a cada iteração, um subconjunto V da vizinhança $N(s)$ da solução corrente s . O membro s' de V com melhor valor nessa região segundo a função $f(\cdot)$ torna-se a nova solução corrente mesmo que s' seja pior que s isto é, que $f(s') > f(s)$ para um problema de minimização.

O uso sistemático da memória é uma característica essencial da Busca Tabu. Enquanto a maioria dos processos de busca guarda essencialmente o valor de $f(s^*)$ da melhor solução s^* obtida até o momento, a Busca Tabu arquiva as informações em uma espécie de itinerário das

últimas soluções visitadas. Tais informações são utilizadas para conduzir o movimento de uma solução para outra a ser escolhida em $N(s)$. A função da memória consiste em restringir a escolha de algum subconjunto de $N(s)$, proibindo movimentos para algumas soluções vizinhas (HERTZ *et al.*, 1995).

A proibição desses movimentos tem a intenção de impedir o retorno a uma solução já visitada anteriormente. O não veto de determinados movimentos pode fazer com que o algoritmo cicle. Um artifício criado com o intuito de não “autorizar” a ocorrência desses movimentos é a Lista Tabu T , que consiste em uma lista contendo as soluções visitadas durante as últimas $|T|$ iterações sequenciadas na forma *FIFO* (*First In First Out*). Desta forma, o primeiro elemento a entrar na Lista Tabu é o primeiro a sair após $|T|$ iterações.

Os principais parâmetros de controle do método são o tamanho da Lista Tabu, a cardinalidade do subconjunto V das soluções vizinhas testadas a cada iteração e o número máximo de iterações sem melhora na função objetivo (BT_{max}).

3. CARACTERIZAÇÃO DO PROBLEMA

O local selecionado para realização do estudo de caso foi o Centro de Tecnologia (CT) de uma Instituição de Ensino Superior. Este CT possui um total de 28 salas de aula sendo estas divididas em três tipos – com Carteira (C), com Mesa (M) e com Prancheta (P). As salas encontram-se distribuídas em seis diferentes blocos: **A, B, D, E, F e H**. A Tabela 1 ilustra as características de cada sala.

Neste trabalho consideram-se apenas as turmas cujas aulas devem ser alocadas nas salas de aula mencionadas. As aulas ministradas em laboratórios não foram incluídas neste estudo. As aulas são ministradas nos turnos da manhã, tarde e noite, sendo que a maior demanda se dá no período matutino, em que o nível de ocupação das salas chega a 80,3% (vide Tabela 2), caracterizando a complexidade do PAAS neste período. Na Tabela 3, que mostra o percentual de ocupação das salas em cada turno e dia da semana, verifica-se que as terças e quartas pela manhã são os dias em que ocorrem as maiores demandas.

Todas as aulas ministradas no CT são destinadas às turmas dos seguintes cursos: Engenharia Mecânica (**EM**), Engenharia de Alimentos (**EA**), Química Industrial (**QI**), Engenharia Civil (**EC**), Engenharia de Produção Mecânica (**EP**) e Arquitetura e Urbanismo (**AU**). Com exceção deste último, os demais cursos possuem certas disciplinas cujas aulas são lecionadas em outros centros, e a responsabilidade de alocação das mesmas não cabe ao CT. Eventualmente, são alocadas turmas de outros centros nas salas do CT, mas isto só ocorre depois de realizada a alocação de todas as turmas referentes aos cursos supracitados.

Tabela 1 – Características das salas do CT

BLOCO	SALA	CAPACIDADE	TIPO
A	CTA 101	63	C
	CTA 103	49	C
	CTA 105	42	C
	CTA 107	42	C
B	CTB 101	42	C
	CTB 102	15	C
	CTB 103	35	C
	CTB 105	30	C
	CTB 107	30	C
D	CTD 101	42	C
	CTD 103	42	C
	CTD 105	42	C
	CTD 107	42	C
E	CTE 101	30	P
	CTE 103	28	P
	CTE 104	35	C
	CTE105	26	P
F	CTF 101	25	C
H	CTH 101	16	M
	CTH 102	35	C
	CTH 103	16	M
	CTH 105	30	C
	CTH AT1	25	M
	CTH AT2	25	M
	CTH AT3	25	M
	CTH AT4	25	M
	CTH AT5	25	M
	CTH AT6	25	M

Tabela 2 – Ocupação das salas por turno

TURNO	DISPONIBILIDADE (*)	DEMANDA (*)	PORCENTAGEM
Manhã	700	562	80,3%
Tarde	700	391	55,9%
Noite	700	68	9,7%

(*) Em horas-aula por semana

Tabela 3 – Ocupação das salas por dia e turno

DIA	DISPONIBILIDADE POR TURNO*	MANHÃ	%	TARDE	%	NOITE	%
Segunda	140	106	75,7	65	46,4	18	12,9
Terça	140	116	82,9	100	71,4	15	10,7
Quarta	140	119	85,0	73	52,1	17	12,1
Quinta	140	109	77,9	103	73,6	17	12,1
Sexta	140	112	80,0	50	35,7	1	0,71
Total	700	562		391		68	

(*) Em horas-aula

A demanda semanal total corresponde a **1021 horas-aula**. De acordo com o tipo de sala, a distribuição desta se dá conforme a Tabela 4.

Tabela 4 – Demanda de acordo com o tipo de sala

TIPO	DEMANDA (*)	%
Carteira	752	73,7
Mesa	235	23,0
Prancheta	34	3,33

(*) Em horas-aula por semana

A Tabela 5 mostra o número de turmas existentes no CT no primeiro semestre letivo de 2005 e suas respectivas demandas médias de alunos, separadas por curso(s). Algumas disciplinas possuem mais de uma turma, assim como existem turmas mistas, isto é, contendo alunos de cursos diferentes. Por exemplo, no referido semestre há 3 turmas, com demanda média de 20 alunos, contendo simultaneamente estudantes dos cursos EC e EA. A cada semestre letivo o número de turmas sofre uma ligeira alteração, porém, não há variação significativa.

Tabela 5 – Número de turmas e demanda média por Curso(s)

CURSO(S)	NÚMERO DE TURMAS	DEMANDA MÉDIA DE ALUNOS
EM	29	15,82
EA	35	12,28
QI	12	9,69
EC	43	20,55
EP	44	12,66
AU	63	16,92
EC e EM	2	23,5
EC, EA	3	20
EC, EM e EP	2	34,5
EC, EM, QI	1	14
EM e EP	6	37
EM e EA	2	30,5
EM, EA e QI	2	27,5
EA, QI	2	10
EC, EM, EA, QI	3	40
Total de turmas = 248		

3.1 RESTRIÇÕES DO PROBLEMA

As restrições do problema são as seguintes:

- (R1)** Duas ou mais aulas não podem ocorrer simultaneamente na mesma sala;
- (R2)** Aulas de uma determinada turma não podem ser alocadas em mais de uma sala num mesmo horário;

- (R3)** Aulas que necessitam de salas com mesa só podem ser alocadas em salas de seu respectivo tipo;
- (R4)** Aulas que necessitam de salas com prancheta não podem ser alocadas em salas do tipo carteira;
- (R5)** Aulas de uma determinada turma só devem ser alocadas em salas de capacidade maior ou igual à demanda de estudantes desta;
- (R6)** O bloco F deve comportar somente aulas destinadas ao curso EP.

3.2 REQUISITOS DE QUALIDADE DA SOLUÇÃO

Os requisitos de qualidade descritos a seguir devem ser atendidos sempre que possível e o seu não atendimento não implica em inviabilidade.

(Q1) As aulas das turmas de um determinado curso e das turmas mistas devem ser alocadas em um bloco pré-determinado, conforme ilustra a Tabela 6. A razão pela qual a alocação destas aulas deve seguir esta recomendação é justificada pela relação entre as características físicas das salas de tal bloco e o perfil das disciplinas do(s) respectivo(s) curso(s). Por exemplo, o bloco H possui, em sua maioria, salas do tipo mesa, pois visa a atender aos requisitos de grande parte das disciplinas do curso AU, que demandam este tipo de sala. Além disso, o bloco H pertence originalmente ao departamento responsável pelo curso AU. Por outro lado, o bloco B comporta laboratórios designados aos cursos EA e QI. Por isso disciplinas de ambos os cursos devem ser alocadas preferencialmente nesse bloco. Os demais casos seguem, basicamente, a mesma linha de raciocínio, porém, sempre havendo diferentes particularidades em cada curso.

(Q2) As aulas das disciplinas cuja frequência corresponde a dois dias semanais não devem ser alocadas em blocos distintos.

(Q3) As aulas das disciplinas cuja frequência corresponde a três dias semanais não devem ser alocadas em blocos distintos.

(Q4) As aulas das turmas que necessitam de carteiras, exceto as disciplinas do curso AU, não devem ser alocadas nos ateliês, localizados nas salas AT1, AT2, AT3, AT4, AT5 e AT6 do Bloco H.

(Q5) Aulas a serem ministradas, preferencialmente em salas com carteiras, não devem ser alocadas em salas do tipo mesa.

(Q6) Aulas a serem ministradas, preferencialmente em salas com carteiras, não devem ser alocadas em salas do tipo prancheta.

(Q7) Aulas a serem ministradas em salas com pranchetas não devem ser alocadas a salas com mesas.

(Q8) Todas as aulas devem ser alocadas.

Tabela 6 – Alocação preferencial de turmas a um bloco pré-determinado

CURSO(S)	BLOCO
EM	A
EA	B
QI	B
EC	D
EP	F
AU	H
EC e EM	D
EC, EA	D
EC, EM e EP	D
EC, EM, QI	D
EM e EP	A
EM e EA	A
EM, EA e QI	A
EA, QI	B
EC, EM, EA, QI	D

Os requisitos de qualidade (Q4) a (Q7) estão intrinsecamente relacionados à questão ergonômica, especialmente no que tange à biomecânica ocupacional. Isto se deve ao fato de que se uma determinada disciplina tiver sua aula alocada em uma sala cuja mobília não seja adequada para o cumprimento de suas devidas atividades, o aluno poderá estar sujeito a uma postura corporal inadequada, acarretando um possível estado de fadiga muscular e, conseqüentemente, comprometendo seu rendimento acadêmico.

A não alocação de todas as aulas não caracteriza, neste caso, uma inviabilidade da solução, pois, nestas situações, as mesmas podem ser alocadas em outros centros ou até mesmo em outras salas de aula do próprio CT, como, por exemplo, naquelas destinadas aos cursos de pós-graduação. Assim sendo, apesar de o não atendimento ao requisito (Q8) representar, a princípio, uma inviabilidade, ele foi tratado como sendo de qualidade.

3.3 FUNÇÃO DE AVALIAÇÃO

Uma solução s é avaliada por uma função f , dada pela expressão (1), a qual deve ser minimizada. A função de avaliação possui duas parcelas, sendo a primeira referente às penalizações pelo não atendimento às restrições (*vide* seção 3.1) e a segunda correspondente às penalizações pelo não atendimento aos requisitos de qualidade (*vide* seção 3.2):

$$f(s) = R(s) + Q(s) \quad (1)$$

sendo:

$$R(s) = M \times \sum_{i=R1}^{R6} r_i(s) \quad (2)$$

e

$$Q(s) = \sum_{j=Q1}^{Q8} \alpha_j q_j(s) \quad (3)$$

em que $r_i(s)$ e $q_j(s)$ correspondem, respectivamente, ao número de vezes (neste caso, a quantidade de horas-aula) que a restrição $i \in \{R1, ..., R6\}$ e o requisito de qualidade $j \in \{Q1, ..., Q8\}$ não é atendido na solução s . Nas expressões (2) e (3), M é um parâmetro de penalidade associado ao não atendimento das restrições, enquanto $\alpha_j \geq 0$ é a penalidade associada ao requisito de qualidade q_j .

Para garantir que a solução do problema seja viável, $R(s)$ deve assumir valor zero, o que ocorre quando cada termo $r_i(s)$ se anula. Para alcançar este objetivo, o parâmetro de penalidade M deve ter um valor suficientemente elevado na função de avaliação.

Os valores atribuídos às penalizações referentes aos requisitos de qualidade são diretamente proporcionais às suas respectivas relevâncias e devem ser parametrizados de acordo com o interesse do usuário. A ideia é que o mesmo possa gerar diferentes soluções, por meio da variação de tais penalidades e, então, efetuar sua escolha da solução.

4. MÉTODOS E PROCEDIMENTOS

Os métodos e procedimentos utilizados para a resolução do PAAS estão subdivididos em dois momentos. O primeiro corresponde à geração de uma solução inicial de relativa qualidade e o segundo está relacionado à aplicação da metaheurística Busca Tabu, com o objetivo de aprimorar a solução inicialmente estabelecida.

4.1 GERAÇÃO DA SOLUÇÃO INICIAL

A solução inicial é gerada por um procedimento construtivo determinístico. Seu pseudocódigo encontra-se exposto no Quadro 1.

O procedimento funciona da seguinte forma. Primeiramente carregam-se os dados referentes às salas e as turmas. Em seguida, as salas são designadas aos seus respectivos blocos e as turmas aos seus respectivos cursos, onde estas são dispostas em ordem decrescente de demanda.

A partir de então, as turmas são alocadas da seguinte maneira. Inicialmente, as turmas são alocadas às salas contidas nos blocos previamente determinados, conforme ilustra a Tabela 6, e com aulas a serem ministradas em uma mesma sala. Aquelas que não conseguirem ser designadas, são alocadas em diferentes blocos e não necessariamente na mesma sala. Se, porventura, a designação de uma determinada aula a uma sala, não tenha sido possível, esta é, então, alocada a uma sala virtual. Na verdade, trata-se de um artifício lógico criado para a situação em que uma turma não tenha conseguido ser alocada.

Procedimento GeracaoSolucaoInicial	
1.	CarregarDados();
2.	<u>para</u> (bloco = 0 até bloco < total de blocos cadastrados) <u>faça</u>
3.	<u>para</u> (curso = 0 até curso < total de cursos cadastrados) <u>faça</u>
4.	Turmas[curso] = PrepararTurmas(curso); {Retorna uma lista das turmas de um dado curso em ordem decrescente de demanda}
5.	AlocarTurmas(Salas, Turmas) {Tenta alocar as aulas das turmas de cada curso em seu respectivo bloco e em salas iguais}
6.	<u>para</u> todas as turmas de um determinado curso <u>faça</u>
7.	<u>para</u> todas as salas de um determinado bloco <u>faça</u>
8.	<u>se</u> a sala tiver horários disponíveis <u>então</u>
9.	Alocar();
10.	AlocarBlocosDistintos(Salas, Turmas) {Tenta alocar as aulas das turmas ainda não alocadas em blocos distintos}
11.	<u>para</u> todas as turmas de um determinado curso <u>faça</u>
12.	<u>se</u> (turma não foi alocada) <u>então</u>
13.	<u>para</u> todos os blocos <u>faça</u>
14.	<u>para</u> todas as salas deste bloco <u>faça</u>
15.	<u>se</u> a sala tiver horários disponíveis <u>então</u>
16.	Alocar();
17.	<u>senão</u>
18.	<u>se</u> o bloco atual é o bloco virtual <u>então</u>
19.	Cadastrar Sala Virtual {Atualizar ‘Turmas’}
20.	Alocar();
21.	Retorna SolucaoInicial;
Fim GeracaoSolucaoInicial	

Quadro 1 – Pseudocódigo do procedimento de geração da solução inicial

4.2 CARACTERIZAÇÃO DA VIZINHANÇA E ALGORITMO BUSCA TABU

Para explorar o espaço de soluções são usados dois tipos de movimento, realocação e troca, para definir as vizinhanças N^R e N^T , respectivamente. O primeiro movimento consiste em alterar a sala em que uma aula deve ser ministrada; enquanto o segundo consiste em trocar a alocação de duas aulas ministradas em salas distintas. No problema considerado uma solução s' é dita vizinha de s se $s' \in N(s) = N^R(s) \cup N^T(s)$.

Na Figura 1 ilustra-se a realocação das aulas da turma X, da sala 101 para a sala 103. Em (a) mostra-se o quadro de horários antes do movimento e em (b), depois do movimento.

Na Figura 2 mostra-se o movimento de troca envolvendo as aulas das turmas A e D, que após o movimento são ministradas nas salas 103 e 101, respectivamente.

Segunda-feira	Sala	
Horário	101	103
07:00-08:00	X	
08:00-09:00	X	
09:00-10:00		
10:00-11:00		B
11:00-12:00		B

(a)

Segunda-feira	Sala	
Horário	101	103
07:00-08:00		X
08:00-09:00		X
09:00-10:00		
10:00-11:00		B
11:00-12:00		B

(b)

Figura 1 – Movimento de realocação de uma aula

Terça-feira	Sala	
Horário	101	103
07:00-08:00		C
08:00-09:00		C
09:00-10:00	A	D
10:00-11:00	A	D
11:00-12:00	A	

(a)

Terça-feira	Sala	
Horário	101	103
07:00-08:00		C
08:00-09:00		C
09:00-10:00	D	A
10:00-11:00	D	A
11:00-12:00		A

(b)

Figura 2 – Movimento de troca entre aulas que ocorrem em salas distintas

O Quadro 2 mostra como a metaheurística Busca Tabu foi adaptada para resolver o PAAS. Inicialmente é gerada uma solução inicial por meio da heurística exposta no Quadro 1. Em seguida, a cada iteração do método escolhe-se aleatoriamente uma aula. Para esta aula executa-se o melhor movimento com relação às vizinhanças de realocação e troca. O movimento não tabu dessa vizinhança restrita V que produzir a solução com o valor mais favorável para a função de avaliação é realizado, independentemente de a solução gerada ser pior ou melhor que a solução corrente. Um movimento tabu também pode ser realizado, desde que produza uma solução melhor que a gerada até então (critério de aspiração por objetivo). Realizado um movimento, armazena-se na lista tabu o atributo $\langle \text{código da disciplina, bloco destino, sala de destino, bloco origem, sala origem} \rangle$ referente à aula que foi sorteada. A inserção desse atributo na lista impede que uma aula selecionada retorne à sua sala e bloco de origem por $|T|$ iterações. O procedimento termina quando o número máximo de iterações ($IterMax$) for atingido ou quando o número máximo de iterações sem que haja melhora na função de avaliação ($BTmax$) for alcançado.

Procedimento BT	
1.	$s_0 \leftarrow \text{GeracaoSolucaoInicial}()$,
2.	$s^* \leftarrow s_0$; {Melhor solução obtida até então}
3.	$s \leftarrow s_0$;
4.	$\text{IterT} \leftarrow 0$; {Contador do número de iterações}
5.	$\text{MelhorIter} \leftarrow 0$; {Iteração mais recente que forneceu s^* }
6.	Seja BTmax o número de iterações sem melhora em s^* ;
7.	$T \leftarrow \emptyset$; {Lista Tabu}
8.	<u>enquanto</u> $(\text{IterT} - \text{MelhorIter} \leq \text{BTmax}$ e $\text{IterT} < \text{IterMax})$ <u>faça</u>
9.	$\text{IterT} \leftarrow \text{IterT} + 1$;
10.	Seja $s' \leftarrow s \oplus m$ o melhor elemento de $V \subseteq N(s)$ tal que o movimento m não seja tabu ($m \notin T$) ou se tabu, atenda ao critério de aspiração, isto é, $f(s') < f(s^*)$
11.	Atualize a Lista Tabu;
12.	$s \leftarrow s'$;
13.	<u>se</u> $f(s') < f(s^*)$ <u>então</u>
14.	$s^* \leftarrow s'$;
15.	$\text{MelhorIter} \leftarrow \text{IterT}$;
16.	<u>fim se</u> ;
17.	<u>fim-enquanto</u> ;
18.	Retorne s^* ;
	fim BT ;

Quadro 2 – Algoritmo Busca Tabu

5. RESULTADOS COMPUTACIONAIS

O algoritmo proposto na seção 4 foi implementado na linguagem de programação C++, utilizando o compilador Borland C++ Builder 6.0 e executado em uma máquina com processador Intel Pentium Centrino 1,86 GHz com 1024 MB de memória RAM e sistema operacional Windows XP – Home Edition.

5.1 AJUSTE DOS PARÂMETROS

A calibração dos parâmetros do algoritmo proposto foi feita por meio de uma bateria de testes preliminares, sendo os valores adotados apresentados na Tabela 7.

Tabela 7 – Parâmetros do procedimento Busca Tabu	
PARÂMETRO	VALOR
IterMax (Número total de iterações)	120000
BTmax (Número de iterações sem melhora)	20000
$ T $ (Tamanho da lista tabu)	100

O algoritmo foi testado em dois cenários distintos com relação aos pesos conferidos aos requisitos de qualidade. A parametrização dos mesmos encontra-se na Tabela 8. O objetivo é avaliar a sensibilidade do algoritmo no que se diz respeito ao atendimento de tais requisitos em função da importância atribuída.

Tabela 8 – Pesos relativos aos requisitos de qualidade

REQUISITO DE QUALIDADE	PESO	VALOR DO PESO	
		Cenário 1	Cenário 2
(Q1) As aulas devem ser alocadas em blocos pré-determinados de acordo com a turma	α_1	1000	1
(Q2) As aulas das disciplinas cuja frequência corresponde a 2 dias semanais não devem ser alocadas em blocos distintos	α_2	1	100
(Q3) As aulas das disciplinas, cuja frequência corresponde a 3 dias semanais, não devem ser alocadas em blocos distintos	α_3	1000	100
(Q4) Aulas a serem ministradas em salas do tipo carteiras não devem ser alocadas nos ateliês.	α_4	1000000	100
(Q5) Aulas a serem ministradas em salas do tipo carteiras não devem ser alocadas em salas com mesas	α_5	1000	10
(Q6) Aulas a serem ministradas em salas do tipo carteiras não devem ser alocadas em salas com pranchetas	α_6	100000	1
(Q7) Aulas a serem ministradas em salas do tipo prancheta não devem ser alocadas em salas com mesas	α_7	1000000	10
(Q8) Todas as aulas devem ser alocadas	α_8	1000000000	1000000000

5.2 COMPARAÇÃO ENTRE AS SOLUÇÕES INICIAL, FINAL E MANUAL

O algoritmo proposto foi executado 20 vezes em cada cenário, partindo de semente de números aleatórios diferentes.

A Tabela 9 detalha a quantidade, em horas-aula, de restrições violadas apenas na solução manual, uma vez que não houve inviabilidades nas soluções geradas pelo algoritmo proposto.

Tabela 9 – Inviabilidades da solução manual

RESTRIÇÃO	QUANTIDADE DE OCORRÊNCIAS (EM HORAS-AULA)	PERCENTUAL
R2) Aulas de uma determinada turma não podem ser alocadas em mais de uma sala num mesmo horário	4	0,4%
R3) aulas que necessitam de mesa só podem ser alocadas em salas de seu respectivo tipo	11	4,7%
R5) Não se pode alocar aulas cujas turmas apresentam uma demanda de alunos maior que a capacidade da sala	60	5,9%
R6) Aulas de turmas não pertencentes ao curso de EP não podem ser alocadas no Bloco F	17	1,9%

A Tabela 10 mostra o relatório da execução dos procedimentos computacionais que deram origem às soluções inicial e final, bem como da solução obtida manualmente. Os

resultados computacionais levaram em consideração as informações relacionadas ao não atendimento dos requisitos de qualidade, em horas-aula.

Tabela 10 – Comparação entre os resultados das soluções manual, inicial e final

REQUISITO DE QUALIDADE	QUANTIDADE DE HORAS-AULA EM QUE O REQUISITO NÃO FOI ATENDIDO			
	Solução manual	Solução inicial	Cenário 1	Cenário 2
			Solução final	Solução final
Q1	340	258	215	266
Q2	23	97	130	33
Q3	0	18	0	12
Q4	35	23	2	2
Q5	64	31	37	33
Q6	42	50	27	45
Q7	22	0	0	0
Q8	48	0	0	0

Com relação à solução manual, verifica-se, pela Tabela 9, que as restrições não foram atendidas em 92 horas-aula, o que representa 12,9% do total a ser distribuído. Dentre estas, destaca-se o fato de que em 5,9% do total de horas-aula, as turmas são alocadas a salas com capacidade inferior à demanda de alunos. Adicionalmente, pela Tabela 10, nota-se que 48 horas-aula sequer foram alocadas.

Por outro lado, apenas a fase construtiva do algoritmo proposto já foi suficiente para produzir uma solução inicial viável, além de ter sido capaz de alocar todas as aulas, conforme mostra a Tabela 10. Ressalta-se, ainda, que a solução inicial é obtida quase que instantaneamente, em menos de um segundo de processamento.

Tendo em vista a inviabilidade da solução manual, é feita a seguir apenas a comparação entre a solução final de cada cenário e a solução inicial.

No cenário 1, em foram dados maior importância aos requisitos de qualidade (Q8), (Q7), (Q4) e (Q6), observa-se que nos dois primeiros houve atendimento na íntegra, tal como na solução inicial, ao passo que os dois últimos foram substancialmente melhorados em 91,3% e 46,0%, respectivamente. Nos requisitos de importância intermediária (Q1) e (Q3), houve melhoria de 16,7% no primeiro e pleno atendimento no segundo, ao contrário da solução inicial na qual havia 18 horas-aula não atendidas em relação a (Q3). Já o requisito (Q5) apresentou piora de 19,4%. Entretanto, isto pode ser justificado pelo fato de (Q5) ter o mesmo peso de (Q1) e (Q3) e tratando-os em conjunto houve uma melhora de 17,9%. O requisito de qualidade (Q2) apresentou uma substancial piora, de 34,0%, porém isto pode ser facilmente justificado pelo fato deste apresentar relevância bem inferior às demais, ou seja, o atendimento a este requisito foi naturalmente “sacrificado” para que fossem privilegiados aqueles de maior expressividade.

No cenário 2 optou-se por utilizar uma configuração de pesos mais equilibrada em relação ao cenário 1, à exceção do requisito (Q8), cujo atendimento integral foi mantido. Nos outros requisitos de maior importância, ou seja, (Q2), (Q3) e (Q4), houve melhora de 66,0%, 33,3% e 91,3%, respectivamente, em relação à solução inicial. Nos requisitos de importância intermediária (Q5) e (Q7), houve piora de 6,5% no primeiro e manutenção do atendimento integral no segundo. No requisito (Q1) e (Q6) houve piora de 3,1% no primeiro e melhora de 10,0% no segundo. Contrariamente ao cenário 1, não houve piora substancial em nenhum dos requisitos, provavelmente devido ao fato de que os pesos adotados foram mais homogêneos.

5.3 VARIABILIDADE DAS SOLUÇÕES FINAIS GERADAS PELO ALGORITMO

Nesta seção, avalia-se o algoritmo com relação à variabilidade das soluções finais no que tange ao atendimento dos requisitos de qualidade.

As Tabelas 11 e 12 mostram os resultados das 20 execuções, bem como o tempo médio de execução do algoritmo proposto, nos cenários 1 e 2, respectivamente. Os números destacados em negrito correspondem à quantidade, em horas-aula, de violações aos requisitos de qualidade da melhor solução encontrada.

Tabela 11 – Resultados do cenário 1

Execução	Tempo (s)	Quantidade de horas-aula em que o requisito não foi atendido							
		(Q1)	(Q2)	(Q3)	(Q4)	(Q5)	(Q6)	(Q7)	(Q8)
1	1658	216	118	0	4	40	28	0	0
2	1620	218	139	0	3	36	32	0	0
3	1531	215	116	0	2	36	33	0	0
4	1576	215	123	0	5	38	29	0	0
5	1603	220	140	0	5	40	26	0	0
6	1357	220	137	0	6	44	26	0	0
7	1598	215	130	0	2	37	27	0	0
8	1625	216	127	0	3	41	27	0	0
9	1618	220	138	0	2	41	28	0	0
10	1869	214	135	0	3	40	30	0	0
11	1908	216	116	0	2	40	27	0	0
12	1561	220	140	0	6	40	27	0	0
13	1627	218	112	0	5	36	35	0	0
14	1547	217	119	0	5	41	27	0	0
15	2350	217	126	0	4	42	27	0	0
16	1481	219	140	0	4	42	29	0	0
17	1716	214	124	0	2	40	29	0	0
18	1600	218	140	0	4	40	27	0	0
19	1558	217	139	0	6	42	27	0	0
20	1558	218	140	0	4	36	30	0	0
Média	1648	217,2	130,0	0,0	3,9	39,6	28,6	0	0
Desv. Pad.	203,8	2,0	10,0	0,0	1,4	2,3	2,4	0	0

Pelas tabelas 11 e 12, observa-se que a carga horária média de violações aos requisitos de qualidade é bem próxima àquela referente à melhor solução. No cenário 1, as violações médias diferem em no máximo 2,6 horas-aula da solução de valor mais favorável da função de avaliação, enquanto que no cenário 2 esta diferença é de no máximo 5,9 horas-aula. À exceção do requisito (Q2), o desvio-padrão das violações foi baixo, sendo no máximo de 2,4 e 4,0 horas-aula nos cenários 1 e 2, respectivamente. Quanto ao desvio-padrão relativo ao requisito (Q2), é interessante notar que este tinha a menor relevância no cenário 1 e quando lhe foi dada uma maior importância relativa (cenário 2), seu desvio-padrão diminuiu, mas à custa do aumento de todos os demais nos quais havia violação.

Tabela 12 – Resultados do cenário 2

Execução	Tempo (s)	Quantidade de horas-aula em que o requisito não foi atendido							
		(Q1)	(Q2)	(Q3)	(Q4)	(Q5)	(Q6)	(Q7)	(Q8)
1	1550	264	32	18	4	33	46	0	0
2	1209	264	28	24	6	32	51	0	0
3	1648	258	43	18	4	34	47	0	0
4	1599	266	33	12	2	33	45	0	0
5	1582	262	49	6	2	31	46	0	0
6	1293	262	38	18	2	33	43	0	0
7	1243	259	36	18	10	33	44	0	0
8	1295	257	40	12	6	33	44	0	0
9	1414	264	45	18	2	31	44	0	0
10	1436	266	41	18	3	31	43	0	0
11	1411	268	58	18	2	26	51	0	0
12	1530	268	30	18	5	33	50	0	0
13	1524	264	38	18	2	27	51	0	0
14	1017	264	35	18	6	31	48	0	0
15	1483	269	38	18	3	36	45	0	0
16	1500	265	34	12	4	28	50	0	0
17	1521	261	27	18	3	34	45	0	0
18	1503	258	46	12	4	28	48	0	0
19	1183	263	45	12	4	31	53	0	0
20	1457	268	41	12	3	32	44	0	0
Média	1420	263,5	38,9	15,9	3,9	31,5	46,9	0	0
Desv. Pad.	163,1	3,6	7,6	4,0	2,0	2,5	3,1	0	0

Os resultados completos das alocações feitas nos dois cenários estão disponíveis no endereço <http://www.decom.ufop.br/prof/marcone/projects/paas.html>.

6. CONCLUSÕES

Neste trabalho, realizou-se um levantamento da demanda correspondente à alocação das aulas no âmbito das salas disponíveis nos blocos do Centro de Tecnologia de uma Instituição Universitária. A partir disto, foi possível apresentar um diagnóstico preciso de ocupação das salas por turno, dia da semana e tipo de sala. Tal estudo permitiu aumentar a

eficiência da gestão administrativa do Centro no que diz respeito à mudança de horários de determinada turma, ou até mesmo na definição de horários para novas turmas, exercendo-se assim um maior aproveitamento do conjunto de salas de aula. Foi constatada a existência de uma maior demanda no turno da manhã, onde cerca de 80% do total de horas-aula semanais encontram-se designadas neste período, tornando o processo de alocação de aulas a salas bastante complexo neste turno.

No tocante à otimização do processo, desenvolveu-se uma estratégia construtiva para gerar uma solução inicial, baseada no conhecimento prévio das características do problema. Esta abordagem construtiva foi capaz de gerar, quase que instantaneamente, soluções que contemplaram todas as restrições de viabilidade com um adicional qualitativo, se comparado com a solução obtida manualmente. Esta, por sua vez, não atendia plenamente aos critérios de viabilidade, bem como deixou de alocar 48 horas-aula (aproximadamente 4,7% do total).

A fase de refinamento do algoritmo proposto foi feita pela metaheurística Busca Tabu. Para testá-la, foram utilizados dois cenários que se diferiam com relação aos pesos atribuídos aos requisitos de qualidade. Em ambos, a Busca Tabu produziu soluções finais de qualidade compatível com os valores dos pesos considerados, mostrando-se adequada à resolução do Problema de Alocação de Aulas a Salas.

AGRADECIMENTOS

O último autor agradece à FAPEMIG, processo CEX PPM 00357/09, e ao CNPq, processo 482765/2010-0, pelo apoio recebido.

REFERÊNCIAS

ALVAREZ-VALDÉS, R.; CRESPO, E; TAMARIT, J. M. Tabu Search: An efficient metaheuristic for university organization problems. **Revista Investigacion Operacional**, v. 22, n. 2, p. 104-113, 2001.

AZIMI, Z. N. Hybrid heuristics for Examination Timetabling problem. **Applied Mathematics and Computation**, v. 163, n. 2, p. 705-733, 2005.

BLUM, C.; ROLI, A. Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. **ACM Computer Surveys**, v. 35, n. 3, p. 268-308, 2003.

CARTER, M. W.; LAPORTE, G. Recent Developments in Practical Course Timetabling. **Lecture Notes in Computer Science**, v. 1408, p. 3-19, 1998.

CARTER, M. W.; TOVEY, C.A. When Is the Classroom Assignment Problem Hard?. **Operations Research Supplement 1**, v. 40, p. 28-39, 1992.

COLORNI, A.; DORIGO, M.; MANIEZZO, V.; Metaheuristics for High School Timetabling. **Computational Optimization and Applications**, v. 9, n. 3, p. 275-298, 1998.

COSTA, D. A tabu search algorithm for computing an operational timetable. **European Journal of Operational Research**, v. 76, p. 98-110, 1994.

GLOVER, F. Future paths in Integer Programming and links to Artificial Intelligence. **Computers and Operations Research**, v. 13, n.5, p. 533-549, 1986.

GLOVER, F.; KOCHENBERGER, G. A. **Handbook of Metaheuristics**, Kluwer Academic Publishers, Boston, 2003.

GLOVER, F.; LAGUNA, M. Tabu Search. **Modern Heuristics Techniques for Combinatorial Problems**, Blackwell Scientific Publications, Oxford, p. 70-150, 1993.

HERTZ, A. Tabu search for large scale timetabling problems. **European Journal of Operational Research Society**, v. 54, p. 39-47, 1991.

HERTZ, A.; TAILLARD, E.; de WERRA, D. A Tutorial on Tabu Search. **Proc. of Giornate di Lavoro AIRO'95**, (Entreprise Systems: Management of Technological and Organizational Changes), p. 13-24, 1995.

HIGGINS, A. J. A dynamic tabu search for large-scale generalized assignment problems. **Computers and Operations Research**, v. 28, p. 1039-1048, 2001.

OSMAN, I. H. Focused issue on applied meta-heuristics. **Computers and Industrial Engineering**, v. 44, p. 205-207, 2002.

SANTOS, H. G.; OCHI, L. S.; SOUZA, M. J. F. An Efficient Tabu Search Heuristic for the School Timetabling Problem. **Lecture Notes in Computer Science**, v. 3059, p. 468-481, 2004.

SCHAERF, A. A Survey of Automated Timetabling. **Artificial Intelligence Review**, v. 13, p.87-127, 1999.

SOUZA, M. J. F. **Programação de horários em escolas**: uma aproximação por metaheurísticas. Tese (Doutorado), Programa de Engenharia de Sistemas e Computação, COPPE/UFRJ, Rio de Janeiro, Brasil, 2000.

SOUZA, M. J. F., XAVIER, A. X.; ARAÚJO, C. R. Experiências com a utilização de *Simulated Annealing* e Busca Tabu na resolução do Problema de Alocação de Salas. In: Simpósio Brasileiro de Pesquisa Operacional, 34, 2002, São Paulo. **Anais ...** São Paulo: SBPO, p. 1100-1110, 2002.

WHITE, G. M.; Xie, B. S.; Zonjic, S. Using tabu search with longer-term memory and relaxation to create examination timetables. **European Journal of Operational Research**, v. 153, p. 90-91, 2004.

WILLEMEN, R. **School timetable construction**: algorithms and complexity. Tese (doutorado), Technische Universiteit Eindhoven, Holanda, 2002.