

Aplicação da metaheurística Busca Tabu na resolução do Problema de Alocação de Salas do Centro de Tecnologia da UFPB

Anand Subramanian (DEP/UFPB) anandsubraman@hotmail.com
José Maurício Fernandes Medeiros (DI/UFPB) mauricio_ganz@yahoo.com
Lucídio dos Anjos Formiga Cabral (DE/UFPB) lucidio@de.ufpb.br
Marccone Jamilson Freitas Souza (DECOM/UFOP) marcone@iceb.ufop.br

Resumo

Este artigo trata do Problema de Alocação de Salas do Centro de Tecnologia da Universidade Federal da Paraíba. Atualmente, sua resolução é efetuada manualmente, tornando o processo árduo e demorado. Desta forma, faz-se necessário recorrer a estratégias computacionais que proporcionem soluções de qualidade e baixo custo. Para tanto, utilizou-se a metaheurística Busca Tabu, que tem se mostrada adequada para a resolução de problemas desta natureza. O presente trabalho procurou identificar as particularidades do problema, para então determinar as restrições de viabilidade e os requisitos de qualidade. Desenvolveu-se ainda um procedimento heurístico para gerar uma solução inicial de boa qualidade. Compararam-se as soluções inicial, final e a que foi gerada manualmente e executada no semestre avaliado. O procedimento Busca Tabu demonstrou ser bastante eficiente e robusto, tendo gerado soluções de alta qualidade.

Palavras-chave: Problema de Alocação de Salas; Otimização; Busca Tabu.

1. Introdução

O Problema de Alocação de Salas (PAS) ou *Classroom Assignment* consiste em alocar aulas, com horários de início e término previamente programados, a um número fixo de salas (CARTER; LAPORTE, 1998, SCHAERF, 1999). Diversas instituições universitárias se deparam com o PAS durante o início de cada semestre letivo. Boa parte destas 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 sala de aulas, ou seja, demandas elevadas, que surgem naturalmente com o crescimento do número de cursos e alunos da instituição.

O PAS é um problema clássico de otimização combinatória pertencente à classe NP-hard (CARTER; TOVEY, 1992), onde 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 de algoritmos determinísticos se torna praticamente inviável. Em vista disso, se faz necessário recorrer a métodos ou técnicas não-determinísticas na tentativa de se obter uma solução de qualidade, isto é, próxima à solução ótima. As técnicas heurísticas, de maneira geral, possuem este perfil, sendo as metaheurísticas as mais indicadas por serem capazes de escapar de pontos ótimos locais (SOUZA *et al*, 2002). Em virtude de seu desempenho satisfatório na resolução de problemas de alocação, optou-se por fazer uso da metaheurística Busca Tabu (BT), com o intuito de se obter uma solução com alto grau de satisfação e baixo custo computacional.

2. Caracterização do Problema

O local selecionado para realização do estudo foi o Centro de Tecnologia (CT) da

Universidade Federal da Paraíba (UFPB). O CT possui um total de 28 salas de aula sendo estas divididas em três tipos – Carteira (17), Mesa (8) e Prancheta (3). As salas encontram-se distribuídas em seis diferentes blocos: **A, B, D, E, F e H**. A capacidade destas varia entre 15 e 63. Neste trabalho consideram-se apenas as turmas cujas aulas devem ser alocadas nas salas 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 predominância destas (demanda) se dá no período matutino. As terças e quartas são os dias de maior demanda e o nível de ocupação das salas chega a 80,3% no turno matutino nesses dias, caracterizando a complexidade do PAS. A demanda semanal total corresponde a **1021 horas-aula**. 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.

2.1 Restrições do problema

As restrições do problema são as seguintes: **(i)** Duas ou mais aulas não podem ocorrer simultaneamente na mesma sala; **(ii)** aulas de uma determinada turma não podem ser alocadas em mais de uma sala num mesmo horário; **(iii)** aulas que necessitam de mesa só podem ser alocadas em salas de seu respectivo tipo; **(iv)** aulas que necessitam de prancheta não podem ser alocadas em salas do tipo carteira; **(v)** aulas de uma determinada turma só devem ser alocadas em salas de capacidade maior ou igual à demanda de estudantes desta; **(vi)** o bloco F deve comportar somente aulas destinadas ao curso de EP.

2.2 Requisitos de Qualidade da Solução

Os requisitos de qualidade descritos a seguir deverão ser atendidos sempre que possíveis. O não cumprimento de determinado requisito não implica em inviabilidade do resultado: **(a)** as aulas das turmas de um determinado curso e das turmas mistas devem ser preferencialmente alocadas em um bloco pré-determinado. A razão pela qual a alocação destas aulas deve seguir esta recomendação justifica-se pela relação entre as características físicas das salas de tal bloco e o perfil das disciplinas do(s) respectivo(s) curso(s); **(b)** as aulas das disciplinas cuja frequência corresponde a 2 dias semanais não devem ser alocadas em blocos distintos; **(c)** as aulas das disciplinas cuja frequência corresponde a 3 dias semanais não devem ser alocadas em blocos distintos; **(d)** as aulas das turmas que necessitam de carteiras, exceto as disciplinas do curso de AU, não devem ser alocadas nos ateliês; **(e)** aulas a serem ministradas preferencialmente em carteiras não devem ser alocadas em salas do tipo mesa; **(f)** aulas a serem ministradas preferencialmente em carteiras não devem ser alocadas em salas do tipo prancheta; **(g)** aulas a serem ministradas em prancheta não devem ser alocadas em mesa; **(h)** Todas as aulas devem ser alocadas.

Os requisitos de qualidade (d), (e), (f), e (g) estão intrinsecamente relacionados à questão ergonômica, especialmente no que tange a biomecânica ocupacional, haja vista 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 situação, em que sua postura corporal se encontre inadequada, acarretando em um possível estado de fadiga muscular, e conseqüentemente, tendo seu rendimento consideravelmente comprometido.

2.3 Função Objetivo

A função objetivo ou de avaliação deve ser construída de acordo com as restrições e os

requisitos de qualidade existentes. Sendo assim, a função objetivo possuirá 14 termos no total, sendo 6 referentes às restrições e os demais 8 correspondentes aos requisitos de qualidade, onde uma penalidade é definida pelo não atendimento destes requisitos. Desta forma, a função objetivo deve ser minimizada e pode ser expressa da seguinte maneira:

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

onde

$$R(S) = \tau[R1(S) + R2(S) + R3(S) + R4(S) + R5(S) + R6(S)] \quad (2)$$

e

$$Q(S) = Q1(S) + Q2(S) + Q3(S) + Q4(S) + Q5(S) + Q6(S) + Q7(S) + Q8(S) \quad (3)$$

Para garantir que a solução do problema seja viável, a equação (2) precisa ser zerada e os termos $R1(S)$, $R2(S)$, $R3(S)$, $R4(S)$, $R5(S)$ e $R6(S)$ devem ser obrigatoriamente nulos, e para tanto, o parâmetro τ deve ter um valor elevado, ou seja, a eventual ocorrência de um evento restrito acarretará na inviabilidade da solução.

Os termos $Q1(S)$, $Q2(S)$, $Q3(S)$, $Q4(S)$, $Q5(S)$, $Q6(S)$, $Q7(S)$ e $Q8(S)$ da equação (3) correspondem aos requisitos de qualidade (a), (b), (c), (d), (e), (f), (g) e (h) respectivamente. Desmembrando cada um destes, tem-se:

$$(a) \ Q1(S) = \alpha A_n; \quad (b) \ Q2(S) = \beta B_n; \quad (c) \ Q3(S) = \gamma C_n; \quad (d) \ Q4(S) = \delta D_n;$$

$$(e) \ Q5(S) = \varepsilon E_n; \quad (f) \ Q6(S) = \sigma F_n; \quad (g) \ Q7(S) = \lambda G_n; \quad (h) \ Q8(S) = \theta H_n.$$

em que n é o número de iterações; A_n , B_n , C_n , D_n , E_n , F_n , G_n e H_n correspondem ao número de vezes (neste caso a quantidade de horas-aula) em que seu requisito (a), (b), (c), (d), (e), (f), (g) e (h), respectivamente, não é atendido na n -ésima iteração; α , β , γ , δ , ε , σ , λ e θ são os pesos relacionados aos requisitos de qualidade (a), (b), (c), (d), (e), (f), (g) e (h), respectivamente.

3. Revisão Bibliográfica

De acordo com Schaerf (1999), o PAS 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. Esta última tem sido frequentemente empregada com êxito na resolução de problemas do gênero em questão. Hertz (1991), Carter e Tovey (1992) Costa (1994), Colorni *et al.* (1998), Schaerf (1999), Alvarez-Valdés *et al.* (2001), Souza *et al.* (2002), White *et al.* (2004) obtiveram resultados significativos na resolução de problemas relacionados à programação de horários de instituições de ensino, bem como suas variantes, por meio da implementação desta técnica.

3.1 Metaheurísticas

Nos últimos anos, tem-se constatado avanços consideráveis, tanto no campo teórico como nas aplicações das metaheurísticas, na tentativa de se determinar soluções aproximadas de problemas complexos de otimização combinatória. As metaheurísticas consistem em um processo iterativo que conduzem e modificam as operações de uma heurística subordinada de tal forma a produzir, eficientemente, soluções de alta qualidade (OSMAN, 2002). São caracterizadas também por apresentarem estratégias de alto nível que exploram o espaço de busca através da utilização de diferentes métodos.

3.2 Busca Tabu

A metaheurística 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 metaheurística BT utiliza uma lista contendo o histórico da evolução do processo de busca, de modo a evitar ciclagem; incorpora uma estratégia de balanceamento entre os movimentos aceitos, rejeitados e aspirados; e adota procedimentos de diversificação e intensificação para o processo de busca.

A cada iteração, a solução atual (S) muda para outra que seja sua vizinha no espaço de busca (S'). Partindo de uma solução inicial S_0 , um algoritmo BT 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 (SOUZA, 2000).

A proibição de determinados movimentos tem a intenção de impedir que a solução retorne ao ponto de mínimo local nas T iterações seguintes. O não veto de determinados movimentos pode fazer com que o algoritmo entre em *loop*. Um artifício criado com o intuito de não “autorizar” a ocorrência destes movimentos é a Lista Tabu. Esta possui uma lista de tamanho t contendo as soluções visitadas durante as últimas T iterações seqüenciadas na forma *Fifo* (*first in first out*).

Souza (2000) e White *et al.* (2004) ressaltam a existência de um mecanismo, relacionado com a Lista Tabu, que anula o *status* tabu de um movimento, denominado função de aspiração. Se um movimento pode proporcionar uma melhora considerável da função objetivo, então o *status* tabu é abandonado e a solução resultante é aceita como potencial vizinho.

4. Métodos e Procedimentos

Os métodos e procedimentos utilizados para a resolução do PAS estão subdivididos em dois momentos. O primeiro corresponde à geração de uma solução inicial de relativa qualidade, que usa a lógica similar à empregada pelas heurísticas construtivas, e o segundo está relacionado à aplicação da metaheurística BT, com o objetivo de aprimorar a solução inicialmente estabelecida. O procedimento BT já fora descrito anteriormente; entretanto, a estrutura da vizinhança será tratada em detalhes nesta seção.

4.1 Geração da Solução Inicial

O procedimento é caracterizado 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 e com aulas a serem ministradas em uma mesma sala. Aquelas que não conseguirem serem designadas serão alocadas em diferentes blocos e não necessariamente na mesma sala. Se, por ventura, a designação de uma determinada aula a uma sala, não tenha sido possível, esta é, então, alocada em uma sala virtual. Na verdade, trata-se de um artifício lógico criado para a situação em que uma turma não tenha conseguido ser alocada. O pseudocódigo da geração da solução inicial encontra-se exposto no Quadro 1.

4.2 Caracterização da Vizinhança

Conforme exposto anteriormente, a solução S do problema sofre uma alteração, a cada

iteração, passando a assumir um valor S' , onde esta nova solução é pertencente à vizinhança de S . Esta modificação, ocorrente no esquema de alocação de S , pode ser efetuada de duas formas: (1) Alterando a sala em que uma aula deve ser ministrada; (2) trocando a alocação de duas aulas, ministradas em salas distintas, entre si.

Procedimento Gerando Solução Inicial

```

1. CarregarDados( );
2. para (bloco = 0 até bloco < total de blocos cadastrados) faça
3.   para (curso = 0 até curso < total de cursos cadastrados) faça
4.     Turmas[curso] = PrepararTurmas(curso);      {Esta função retorna uma lista das turmas
de um determinado curso em ordem decrescente de demanda}
5.   AlocaTurmas(Salas, Turmas)                    {Tenta alocar as aulas das turmas de cada curso em
seu respectivo bloco e em salas iguais}
6.   para todas as turmas de um determinado curso faça
7.     para todas as salas de um determinado bloco faça
8.       se a sala tiver horários disponíveis então
9.         Aloca( );
10.  AlocaBlocosSeparados(Salas, Turmas)           {Tenta alocar as aulas das turmas ainda não
alocadas em blocos separados}
11.  para todas as turmas de um determinado curso faça
12.    se (turma não foi alocada) então
13.      para todos os blocos faça
14.        para todas as salas deste bloco faça
15.          se a sala tiver horários disponíveis então
16.            Aloca( );
17.          senão
18.            se o bloco atual é o bloco virtual então
19.              Cadastra Sala Virtual {Atualiza 'Turmas'}
20.              Aloca( );
21.  Retorna SolucaoInicial;
Fim Gerando Solução Inicial

```

Quadro 1 – Algoritmo Gerando Solução Inicial

Na Figura 1 ilustra-se a realocação de uma determinada aula de uma turma para outra que apresenta horário disponível para recebê-la. Não sendo possível alterar o local de ocorrência de uma determinada aula com esse tipo de movimento, tenta-se trocar esta de sala com uma aula correspondente a outra turma, isto é, uma aula de uma turma passa a ocupar o lugar da outra e vice-versa.

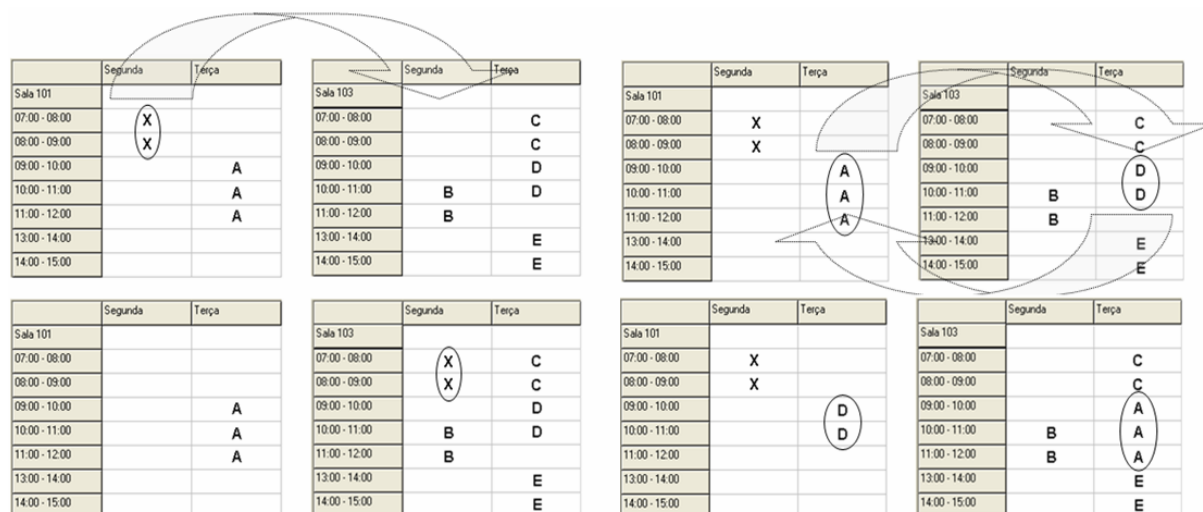


Figura 1 – Alterando a sala em que uma aula deve ocorrer e trocando a alocação de duas aulas entre si

5. Resultados Computacionais

Os algoritmos propostos nas seções 3 e 4 foram implementados na linguagem de programação C++ utilizando o compilador Borland C++ Builder 6.0 e foram executados em um Notebook Intel Pentium Centrino 1.86GHz com 1024MB de memória RAM.

5.1 Ajuste dos Parâmetros

A parametrização dos dados de entrada do procedimento BT e dos pesos referentes aos requisitos de qualidade foi efetuada após uma série de execuções do programa, onde se variou o número total de iterações; o número máximo de iterações sem melhora (BT_{max}); o tamanho da Lista Tabu e os valores dos pesos de cada requisito de qualidade. Os ajustes paramétricos estabelecidos podem ser visualizados nas Tabelas 1 e 2.

Número Total de Iterações	BT_{max}	Tamanho da Lista Tabu
120000	20000	100

Tabela 1 – Parâmetros do procedimento Busca Tabu

Requisito de Qualidade	Peso	Valor do Peso
(a)	α	1000
(b)	β	1
(c)	γ	1000
(d)	δ	1000000
(e)	ε	1000
(f)	σ	100000
(g)	λ	1000000
(h)	θ	1000000000

Tabela 2 – Pesos relativos aos requisitos de qualidade

O algoritmo foi executado 20 vezes, onde se adotou como solução final aquela que apresentou o melhor valor para a função objetivo, ou seja, aquela que conseguiu atender, relativamente, a mais critérios de qualidade.

5.2 Comparação entre os resultados das soluções inicial, final e manual

A Tabela 3 mostra o relatório da execução dos procedimentos computacionais que deram origem as 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: (a), (b), (c), (d), (e), (f), (g) e (h), em horas-aula.

É importante destacar que os requisitos de qualidade de maior importância, no caso, (d), (f) e (h), cujos valores dos pesos são superiores aos demais, apresentaram considerável melhoria se comparada com a solução manual, isto é, o requisito (d), segundo mais relevante, quase foi atendido por completo, apresentando melhora de 94%; o requisito (f), terceiro mais importante, apresentou uma melhora de 55%; e o requisito (h), mais relevante de todos, foi atendido na íntegra. O requisito de qualidade (b) apresentou uma piora substancial, porém isto pode ser facilmente justificado pelo fato deste apresentar relevância inferior às demais, ou seja, o atendimento a este requisito foi naturalmente “sacrificado” para que fossem privilegiados aqueles de maior expressividade. Por outro lado, os requisitos de importância intermediária, (a) e (e), apresentaram melhoras de 36,4% e 37,5% respectivamente. Já o requisito (c), cuja importância é a mesma de (a) e (e), apresentou certa piora, porém isto não compromete a qualidade global da solução.

Requisito de Qualidade	Resultados		
	Quantidade de horas-aula em que o requisito não foi atendido		
	Solução Manual	Solução Inicial	Solução Final
(a) As aulas devem ser alocadas em blocos pré-determinados de acordo com a turma	340	258	216
(b) As aulas das disciplinas, cuja frequência corresponde a 2 dias semanais, não devem ser alocadas em blocos distintos	23	97	120
(c) As aulas das disciplinas, cuja frequência corresponde a 3 dias semanais, não devem ser alocadas em blocos distintos	0	18	24
(d) Aulas a serem ministradas em salas do tipo carteiras não devem ser alocadas nos ateliês.	35	23	2
(e) Aulas a serem ministradas em salas do tipo carteiras não devem ser alocadas em mesas	64	31	40
(f) Aulas a serem ministradas em salas do tipo carteiras não devem ser alocadas em pranchetas	42	50	27
(g) Aulas a serem ministradas em salas do tipo prancheta não devem ser alocadas em mesas	22	0	0
(h) Todas as aulas devem ser alocadas	48	0	0

Tabela 3 – Comparação entre as soluções inicial, final e manual

As soluções inicial e final produzidas pelo método proposto foram todas viáveis, ao passo que a solução obtida manualmente apresentou algumas inviabilidades, isto é, alguns termos relativos às restrições do problema não foram zerados, mais especificamente aquelas referentes as restrições (ii), (iii), (v) e (vi) conforme ilustra a Tabela 4. Analisando a mesma, pode-se constatar, por exemplo, que 4,7% da demanda de aulas que deveriam ser ministradas em mesa, não foram alocadas em seu respectivo tipo, assim como 5,9% do total de horas-aula foram alocadas em salas com capacidade inferior à demanda de alunos.

Inviabilidades da Solução Obtida Manualmente

Restrição	Quantidade de Ocorrências (Em Horas-Aula)	Percentual
ii) Aulas de uma determinada turma não podem ser alocadas em mais de uma sala num mesmo horário	4	0,4%
iii) aulas que necessitam de mesa só podem ser alocadas em salas de seu respectivo tipo	11	4,7%
v) Não se pode alocar aulas cujas turmas apresentam uma demanda de alunos maior que a capacidade da sala	60	5,9%
vi) Aulas de turmas não pertencentes ao curso de EP não podem ser alocadas no Bloco F	17	1,9%

Tabela 4 – Inviabilidades da solução atualmente em vigor

5.3 Robustez

Por meio da Tabela 5 é possível visualizar algumas informações sobre o comportamento da função objetivo ao longo das 20 execuções e o tempo médio de execução do algoritmo proposto. O objetivo desta análise é verificar a consistência deste procedimento, isto é, se o percentual de melhora do valor da função objetivo da solução final em relação à solução inicial, apresenta baixa variabilidade.

Verifica-se que os valores da amostra apresentam uma baixa dispersão (vide desvio padrão), indicando que, para os valores dos pesos relativos aos requisitos de qualidade adotados, o procedimento BT se mostrou consistente.

Percentual Médio de Melhora	Desvio Padrão	Melhora de Maior Significância	Melhora de Menor Significância	Tempo Médio de Execução
79%	4,01	82%	69%	30min55s

Tabela 5 – Robustez do procedimento Busca Tabu

6. Conclusão

O procedimento desenvolvido para gerar a solução inicial obteve desempenho satisfatório, tendo gerado soluções que contemplaram todas as restrições de viabilidade com um adicional qualitativo, se comparado com a solução obtida manualmente. Esta, por sua vez, não atendeu plenamente aos critérios de viabilidade, nem foi capaz de alocar todas as aulas, deixando de contemplar 48 horas-aula (aproximadamente 4,7% do total).

A metaheurística BT mostrou-se adequada a este problema tendo sempre produzido melhorias significativas durante o processo de refinamento da solução inicial. Para o cenário teste, a metaheurística obteve redução média de 79% no valor da função objetivo da solução inicial. Além disso, o algoritmo apresentou-se robusto, uma vez que não houve grande variabilidade na solução final, e sempre gerou soluções satisfatórias, com baixo custo computacional.

Referências

- ALVAREZ-VALDÉS, R.; CRESPO, E. & TAMARIT, J. M. *Tabu Search: An efficient metaheuristic for university organization problems*. Revista Investigacion Operacional. Vol. 22, n. 2, p. 104-113, 2001.
- CARTER, M.W. & LAPORTE, G. *Recent Developments in Practical Course Timetabling*. Lecture Notes in Computer Science. Vol. 1408, p. 3-19, 1998.
- CARTER, M.W. & TOVEY, C.A. *When Is the Classroom Assignment Problem Hard?* Operations Research. Vol. 40, supplement 1, p. 28-39, 1992.
- COLORNI, A.; DORIGO, M. & MANIEZZO, V. *Metaheuristics for High School Timetabling*.

Computational Optimization and Applications. Vol. 9, n. 3, p. 275-298, 1998.

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

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

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

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

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

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

SOUZA, M. J. F. *Programação de horários em escolas: uma aproximação por metaheurísticas.* Rio de Janeiro, 2000. Tese (Doutorado), 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, 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. Vol. 153, p. 90-91, 2004.