

PROBLEMA DE ALOCAÇÃO DE SALAS EM CURSOS UNIVERSITÁRIOS: UM ESTUDO DE CASO

Alan Souza Prado, Sérgio Ricardo de Souza

Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG)
Av. Amazonas, 7675, CEP 30510-000, Belo Horizonte (MG), Brasil
alan.prado@gmail.com, sergio@dppg.cefetmg.br

RESUMO

Este trabalho apresenta um algoritmo heurístico, baseado em uma construção gulosa e na metaheurística *Simulated Annealing*, para resolver o Problema de Alocação de Salas (PAS) em cursos universitários. Este problema consiste em alocar turmas de disciplinas, com horários já previamente fixados, a salas de aulas distribuídas em vários prédios, levando-se em conta um conjunto de restrições. O algoritmo utiliza a fase de construção gulosa para gerar uma solução inicial e a metaheurística *Simulated Annealing* para fazer seu refinamento. A exploração do espaço de soluções é feita por meio de cinco movimentos. O algoritmo foi testado usando-se dados reais de alocação de salas de uma instituição de ensino superior.

PALAVRAS-CHAVE: *Problema de Alocação de Salas, Construção Gulosa, Simulated Annealing*

ABSTRACT

This paper presents a heuristic algorithm based on a greedy construction and the Simulated Annealing Metaheuristic in order to solve a Classroom Assignment Problem in university courses. This problem consists in allocating classes, with schedules previously defined, to classrooms distributed in several buildings, respecting a set of constraints. The algorithm uses the greedy construction phase to generate an initial solution and the Simulated Annealing to do its refinement. The exploration of the solution space is taken by five movements. The algorithm was tested using real data of an university.

KEYWORDS: *Classroom Assignment Problem, Greedy Construction, Simulated Annealing*

Área Principal: Metaheurísticas (MH) e Otimização Combinatória (OC)

1. INTRODUÇÃO

Semestralmente, inúmeras instituições de ensino superior necessitam distribuir as diferentes turmas de disciplinas a salas de aula, obedecendo a uma série de restrições, como, por exemplo, que as aulas de turmas de mesmo curso e período estejam em uma mesma sala.

Normalmente, essas instituições encontram muitas dificuldades nesta tarefa, visto a complexidade envolvida, o que inviabiliza sua solução de forma manual. De fato, dado o número elevado de combinações possíveis de alocações, uma solução manual normalmente não consegue contemplar muitos requisitos exigidos, gerando insatisfação por parte dos professores e alunos neste processo.

A distribuição de aulas previamente definidas com horários estabelecidos, atentando-se a diversas particularidades relacionadas a espaço físico, possibilidade de acesso, infraestrutura e recursos necessários, caracteriza, então, o Problema de Alocação de Salas (PAS) (Schaerf, 1999).

O PAS pode ser considerado como um subproblema do Problema de Programação de Horários de Cursos Universitários (*Course University*) (Bardadym, 1996), problema este que consiste em definir tanto os horários das turmas de disciplinas quanto as salas para essas turmas. Na prática, entretanto, é mais comum que cada departamento da instituição de ensino em tela defina previamente os horários de aula de suas turmas de disciplinas. Assim, a ocorrência do PAS é mais frequente que o do próprio Problema de Programação de Horários de Cursos Universitários.

Em vista do fato de o PAS ser um problema da classe NP-difícil (Even et al., 1976; Carter e Tovey, 1992), ele tem sido normalmente resolvido por meio de técnicas heurísticas, dentre as quais as metaheurísticas (Souza et al., 2002). Desta maneira, diversos trabalhos têm sido apresentados para a resolução do PAS por meio dessas técnicas, como, por exemplo, Algoritmo Genético (Ueda et al., 2001), Busca Tabu (Subramanian et al., 2011), *Simulated Annealing* (Beyrouthy et al., 2006). Além destes, citam-se algoritmos híbridos, que apresentam a combinação de técnicas, como: Grafo Bipartido mais Algoritmo Húngaro (Politano, 2006) e *Simulated Annealing* mais Busca Tabu (Souza et al., 2002).

Neste trabalho apresenta-se um algoritmo para solução do Problema de Alocação de Salas, que combina uma fase de construção gulosa para gerar uma solução inicial e a metaheurística *Simulated Annealing* para melhorar essa solução na fase de refinamento. Utilizou-se, para fins de validação da solução, um ambiente real, caracterizado pelos ambientes multiprediais da Universidade Federal de Ouro Preto (UFOP), abrangendo turmas distribuídas em dois prédios.

O restante deste trabalho está organizado como segue. Na seção 2 é apresentada a descrição do problema, em detalhes. Na seção 3, o algoritmo proposto é apresentado em detalhes. Na seção 4 são mostrados os resultados obtidos. A seção 5 conclui o trabalho.

2. CARACTERIZAÇÃO DO PROBLEMA

O problema abordado é o do Instituto de Ciências Exatas e Biológicas (ICEB) da Universidade Federal de Ouro Preto (UFOP). Este instituto tem cursos próprios de graduação e pós-graduação, e oferta, também, aulas do ciclo básico de vários outros cursos de graduação da instituição. Neste instituto também são ministradas aulas de disciplinas de outros institutos, como, por exemplo, dos departamentos de letras e educação da mesma universidade, neste caso para os cursos de licenciatura ofertados pelo próprio ICEB. As aulas ocorrem em dois prédios, quais sejam, o prédio do ICEB e o prédio denominado Pavilhão Central de Aulas (PCA). Há 16 salas disponíveis no ICEB e 19 no PCA. As aulas ocorrem em três turnos (manhã, tarde e noite), ocorrendo, por sala, 16 horários diários de aula.

Estes horários referem-se aos períodos de início e fim de cada aula e estão distribuídos conforme a Tabela 1.

A Tabela 2 mostra a quantidade de salas disponíveis e a quantidade de vagas ofertadas em cada sala de cada um dos prédios nos quais as aulas do instituto são alocadas.

Tabela 1: Distribuição dos Horários das Aulas

Manhã	Tarde	Noite
De 07:30 às 08:20	De 12:00 às 13:30	De 18:00 às 18:50
De 08:20 às 09:10	De 13:30 às 14:20	De 19:00 às 19:50
De 09:20 às 10:10	De 14:20 às 15:10	De 19:50 às 20:40
De 10:10 às 11:00	De 15:20 às 16:10	De 21:00 às 21:50
De 11:10 às 12:00	De 16:10 às 17:10	De 21:50 às 22:40
-	De 17:10 às 18:00	-

Tabela 2: Características dos Blocos de Salas

Blocos de Salas ICEB		Bloco de Salas Pavilhão	
Total de Salas:	16	Total de Salas:	19
Nome	Capacidade	Nome	Capacidade
Sala 01	70	101	60
Sala 02	60	102	60
Sala 03	44	103	60
Sala 04	55	104	60
Sala 05	44	105	60
Sala 06	53	106	60
Sala 07	70	201	60
Sala 08	60	202	60
Sala 09	50	203	54
Sala 10	50	204	60
Sala 14	40	205	60
Sala 18	50	206	60
Sala 19	50	207	60
Sala 21	53	208	60
Sala 22	40	209	60
Sala 23	50	210	60
Total:	839	211	60
		213	60
		214	60
		Total:	1134

O problema estudado neste trabalho é o da alocação de aulas no primeiro semestre letivo de 2014, em que foram alocadas 571 (quinhentas e setenta e uma) turmas de disciplinas, que ocuparam 2280 (dois mil duzentos e oitenta) horários nos três turnos.

O sistema até então em uso considerava a existência de um único prédio. Esta era a situação do ICEB no ano em que foi desenvolvido o sistema. Como as aulas atualmente são ministradas em dois prédios, a solução encontrada para aplicar o sistema era definir *a priori* quais disciplinas devem ficar em um prédio e quais devem ficar em outro prédio. Naturalmente, essa solução não é de boa qualidade. De fato, alunos de um mesmo curso e período, por exemplo, podem ter uma aula em um prédio e a aula seguinte em outro prédio.

São os seguintes os requisitos a serem atendidos na alocação das turmas às salas:

1. Todas as aulas devem ser alocadas;
2. Não alocar turmas a salas que não comportem essas turmas;
3. Evitar a alocação de turmas pequenas a salas de maior capacidade;
4. Alocar turmas de disciplinas de mesmo curso e período preferencialmente em uma mesma sala, evitando, assim, o trânsito de alunos entre salas;

5. Preferencialmente, alocar aulas aos prédios requisitados pelos professores das turmas.
6. Alocar uma turma em um prédio específico, em caso de exigência.
7. Respeitar as disponibilidades das salas, já que algumas salas (ou horários de algumas salas) são previamente reservadas para cursos de outras unidades acadêmicas.

3. DESCRIÇÃO DO ALGORITMO PROPOSTO

3.1. Algoritmo Proposto

O algoritmo proposto possui duas fases, sendo uma de construção de uma solução inicial para o problema e outra de refinamento dessa solução. A construção é apresentada na Seção 3.2 e o refinamento na Seção 3.6.

3.2. Construção da Solução Inicial

Uma solução inicial para o problema é construída de forma gulosa, como a seguir se descreve. Inicialmente, as aulas das turmas são ordenadas pelo tamanho, de sorte que as turmas maiores são as primeiras da lista e as menores, as últimas. A seguir, começa-se a alocação com a turma de maior demanda. Em seguida, a partir desta turma, procura-se na lista das turmas, a(s) turma(s) de mesmo curso, período e turno em que a turma. Elas são alocadas à maior sala disponível no horário em que são ofertadas, independentemente de a turma caber ou não nesta sala. Se não existir sala nessas condições, então é criada uma sala fictícia, dita virtual, e essa(s) turma(s) são então alocadas nesta sala virtual. A partir de então passa-se para a alocação da segunda maior turma. Para esta turma, procura-se, então, a maior sala disponível. Este procedimento é repetido até que todas as aulas das turmas sejam alocadas a alguma sala, real ou virtual.

O pseudocódigo da construção de uma solução inicial para o problema é apresentado pelo Algoritmo 1.

Algoritmo 1: Construção Gulosa

```

Entrada:  $E = \{\text{conjunto de todos horários a se alocar}\}$ 
Saída: Solução  $S$ 
1 início
2    $S \leftarrow 0$ ;
3    $C \leftarrow E$  /* ordenado de forma decrescente pelo tamanho das turmas
   */;
4    $g \leftarrow 0$  /* armazena o valor das inserções na solução inicial */;
5   enquanto  $|C| > 0$  faça
6     Seleciona-se  $c^*$ ; /* Turma de maior demanda em  $C$  */;
7      $L(c) \leftarrow \{\text{Turma(s) de mesmo curso, período e turno que } c^*\}$ ;
8     Verifica a existência de sala disponível para alocar  $L(c)$ ;
9     se (Existe Sala Disponível) então
10      | Alocar  $L(c)$  à sala disponível;
11      | Atualizar a solução parcial:  $S \leftarrow S \cup \{L(c)\}$ ;
12      senão
13      | Criar Sala Virtual;
14      | Alocar  $L(c)$  à sala virtual criada;
15      | Atualizar a solução parcial:  $S \leftarrow S \cup \{L(c)\}$ ;
16      fim
17      $g \leftarrow g + \text{o valor do custo da inserção de } L(c) \text{ em } S$ ;
18     Retirar  $L(c)$  de  $C$ ;
19   fim
20   retorna  $S$ 
21 fim

```

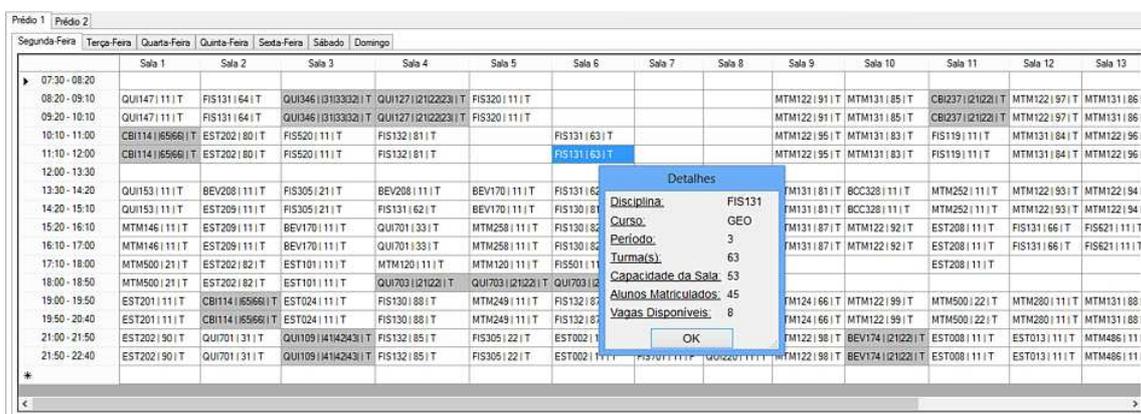
Como geralmente uma heurística de construção não produz uma solução final de boa qualidade (Barbosa, 2011), é aplicada sobre ela uma heurística de refinamento, no caso, a metaheurística *Simulated Annealing* (vide Subseção 3.6).

3.3. Representação da Solução

Uma solução x é representada por uma matriz multidimensional de dimensões $|T| \times |P| \times |S| \times |D| \times |H|$, em que T é o conjunto de turmas de disciplinas, P é o conjunto de prédios, no caso, dois (ICEB e PCA), S é o conjunto de salas $s \in S$ de um mesmo prédio, D é o conjunto de dias da semana (segunda-feira, terça-feira, quarta-feira, quinta-feira, sexta-feira, sábado) e H é o conjunto dos 16 horários diários disponíveis em cada sala.

A Figura 1 mostra um fragmento de uma solução $x_{tps dh}$, em que t é representação das turmas alocadas, p é o prédio do ICEB, s representa várias salas (Sala 1 a Sala 13), d representa o dia da semana (no exemplo, a segunda-feira) e h os horários disponíveis desse prédio nesse dia, no caso, das 7:30 às 22:40 h.

A capacidade de uma sala s de um prédio p é representada por cap_{ps} e a demanda de turma $t \in T$ por dem_t .



Horário	Sala 1	Sala 2	Sala 3	Sala 4	Sala 5	Sala 6	Sala 7	Sala 8	Sala 9	Sala 10	Sala 11	Sala 12	Sala 13
07:30 - 08:20													
08:20 - 09:10	QUI147 111 T	FIS131 64 T	QUI346 0113332 T	QUI127 1212223 T	FIS320 111 T								
09:20 - 10:10	QUI147 111 T	FIS131 64 T	QUI346 0113332 T	QUI127 1212223 T	FIS320 111 T								
10:10 - 11:00	CBI114 05666 T	EST202 80 T	FIS520 111 T	FIS132 81 T		FIS131 63 T							
11:10 - 12:00	CBI114 05666 T	EST202 80 T	FIS520 111 T	FIS132 81 T		FIS131 63 T							
12:00 - 13:30													
13:30 - 14:20	QUI153 111 T	BEV208 111 T	FIS305 211 T	BEV208 111 T	BEV170 111 T	FIS131 62 T							
14:20 - 15:10	QUI153 111 T	EST209 111 T	FIS305 211 T	FIS131 62 T	BEV170 111 T	FIS130 81 T							
15:20 - 16:10	MTM146 111 T	EST209 111 T	BEV170 111 T	QUI701 331 T	MTM258 111 T	FIS130 82 T							
16:10 - 17:00	MTM146 111 T	EST209 111 T	BEV170 111 T	QUI701 331 T	MTM258 111 T	FIS130 82 T							
17:10 - 18:00	MTM500 211 T	EST202 82 T	EST101 111 T	MTM120 111 T	MTM120 111 T	FIS501 11 T							
18:00 - 18:50	MTM500 211 T	EST202 82 T	QUI703 121222 T	QUI703 121222 T	QUI703 121222 T	QUI703 121222 T							
19:00 - 19:50	EST201 111 T	CBI114 05666 T	EST024 111 T	FIS130 88 T	MTM249 111 T	FIS132 87 T							
19:50 - 20:40	EST201 111 T	CBI114 05666 T	EST024 111 T	FIS130 88 T	MTM249 111 T	FIS132 87 T							
21:00 - 21:50	EST202 90 T	QUI701 311 T	QUI109 1414243 T	FIS132 85 T	FIS305 221 T	EST002 1 T							
21:50 - 22:40	EST202 90 T	QUI701 311 T	QUI109 1414243 T	FIS132 85 T	FIS305 221 T	EST002 1 T							

Figura 1: Exemplo de Solução Inicial.

A Figura 1 apresenta uma alocação feita pelo Algoritmo 1: Construção Gulosa. Nesta alocação tem-se, por exemplo:

No Prédio 1, na sala 6, na segunda-feira, nos horários das 10:10 às 11:00 horas e das 11:10 às 12:00 horas, foi alocada:

- Disciplina: Física II representada pela sigla FIS131
- Curso: Geografia representado pela sigla GEO
- Período: 3^o
- Turma(s): 63
- Capacidade da Sala: 53
- Demanda: 45 alunos
- Vagas disponíveis: 8

Na Prédio 1, na sala 2, na segunda-feira, nos horários das 19:00 às 19:50 horas e das 19:50 às 20:40 horas, foi alocada:

- Disciplina: Anatomia Humana representada pela sigla CBI114
- Curso: Nutrição representado pela sigla NUT
- Período: 1^o

- Turma(s): junção de 65 e 66
- Capacidade da Sala: 60
- Demanda: 47 alunos
- Vagas disponíveis: 13

A cada horário está disponível um campo para a alocação das disciplinas e suas características. Eventualmente, em uma alocação, quando esses campos não apresentarem dados, ou seja, estiverem vazios, entende-se que não foi alocada nenhuma turma neste prédio, sala, dia da semana e horário. Essa situação pode ser identificada na Figura 1, por exemplo, nos horários das 10:10 às 11:00 horas e das 11:10 às 12:00 horas na segunda-feira no prédio 1 nas salas 5, 7 e 8 ($x(0, 1, 5, 2, 4)$, $x(0, 1, 5, 2, 5)$, $x(0, 1, 7, 2, 4)$, $x(0, 1, 7, 2, 5)$, $x(0, 1, 8, 2, 4)$ e $x(0, 1, 8, 2, 5)$ respectivamente).

É importante ressaltar que, para a solução inicial, alocações inviáveis podem acontecer, como por exemplo, a alocação de uma disciplina em uma sala com capacidade inferior ao número de alunos matriculados. Um exemplo é ilustrado na Figura 2.

Horário	Sala 17	Sala 18	Sala 19	Sala 20	Sala 21	Sala 22	Sala 23	Sala 24	Sala 25	Sala 26	Sala 27	Sala 28	Sala 29
07:30 - 08:20	MED001 1 T	MED001 1 T	MED002 2 T	MED004 4 T					MED003 3 T				
08:20 - 09:10	CBI615 2 22 T				BCC701 5 6 T	CBI258 2 22 T	BEV177 2 22 T	BEV177 2 22 T	MED003 3 T	BCC444 1 1 T	BCC486 1 1 T	BCC341 1 1 T	
09:20 - 10:10				ED004 4 T	BCC701 5 6 T	CBI258 2 22 T	BEV177 2 22 T	BEV177 2 22 T	MED003 3 T	BCC444 1 1 T	BCC486 1 1 T	BCC341 1 1 T	
10:10 - 11:00				ED004 4 T	CBI607 2 22 T	BCC203 1 1 T	BCC324 1 1 T	BCC502 1 1 T	MED003 3 T	BCC342 2 1 T	BCC760 2 1 P	CBI126 1 1 T	
11:10 - 12:00				ED004 4 T	CBI607 2 22 T	BCC203 1 1 T	BCC324 1 1 T	BCC502 1 1 T	MED003 3 T	BCC701 5 6 T	BCC342 2 1 T	BCC760 2 1 P	CBI126 1 1 T
12:00 - 13:30													
13:30 - 14:20					CC321 1 1 T	BCC326 1 1 T	CBI143 1 1 T	BCC266 1 1 T	FIL211 2 1 T	BCC441 1 1 T	BCC102 1 1 T		
14:20 - 15:10					CC321 1 1 T	BCC326 1 1 T	CBI143 1 1 T	BCC266 1 1 T	FIL211 2 1 T	BCC441 1 1 T	BCC102 1 1 T		
15:20 - 16:10	BCC362 1 1 T				CC760 4 P	BCC466 1 1 T	BEV712 2 1 P						
16:10 - 17:00	BCC362 1 1 T				CC760 4 P	BCC466 1 1 T	BEV712 2 1 P						
17:10 - 18:00	CBI234 2 22 T				B1156 1 1 T	CBI156 1 1 T	BCC449 1 1 T						
18:00 - 18:50	EDF001 1 1 T				EDF004 4 T	EDF005 5 T	EDF006 6 T		EDF007 7 T				EDF008 8 T
19:00 - 19:50	EDF001 1 1 T	EDF002 2 T	EDF003 3 T	EDF004 4 T	EDF005 5 T	EDF006 6 T	BCC701 5 6 T	EDF007 7 T	BCC721 1 1 T	EDU209 5 4 T	BCC760 7 T	CBI139 1 1 T	EDF008 8 T
19:50 - 20:40	EDF001 1 1 T	EDF002 2 T	EDF003 3 T	EDF004 4 T	EDF005 5 T	EDF007 7 T	BCC701 5 6 T	EDF007 7 T	BCC721 1 1 T	EDU209 5 4 T	BCC760 7 T	CBI139 1 1 T	EDF008 8 T
21:00 - 21:50	EDF001 1 1 T	EDF002 2 T	EDF003 3 T	EDF004 4 T	EDF005 5 T	EDF006 6 T	BCC701 5 6 T	EDF008 8 T	CBI614 2 22 T	EDU303 3 7 T	BCC760 8 T		
21:50 - 22:40	EDF001 1 1 T	EDF002 2 T	EDF003 3 T	EDF004 4 T	EDF005 5 T	EDF006 6 T	BCC701 5 6 T		CBI614 2 22 T	EDU303 3 7 T	BCC760 8 T		

Figura 2: Exemplo de Solução Inicial com inviabilidade

A alocação inviável mostrada na Figura 2 apresenta, por exemplo, que no Prédio 2, na sala 17, na terça-feira, nos horários das 08:20 às 9:10 horas e das 9:20 às 10:10 horas, foi alocada:

- Disciplina: Bioquímica Celular A representada pela sigla CBI615
- Curso: Farmácia representado pela sigla FAR
- Período: 3^o
- Turma(s): junção de 21 e 22
- Capacidade da Sala: 59
- Demanda: 69 alunos
- Vagas disponíveis: -10

ou seja, o número de alunos matriculados é maior que a capacidade da sala em 9 alunos.

3.4. Vizinhanças

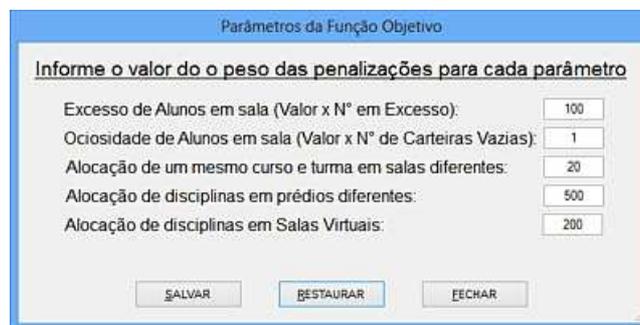
Para explorar o espaço de solução do PAS foram implementados cinco movimentos, definidos a seguir, cada qual dando origem a uma vizinhança. O conjunto de todas essas vizinhanças de uma solução $s \in S$ define a vizinhança $N(s)$.

1. Realocação: Consiste em realocar uma disciplina da alocação atual para outra sala no mesmo prédio, dia da semana e horário, vazia.
2. Troca: Consiste em trocar uma disciplina por outra já alocadas no mesmo prédio, no mesmo dia da semana e horário.
3. Realocação entre Prédios: Consiste em realocar uma disciplina da alocação atual para outra sala em prédio diferente, mesmos dia da semana e horário, vazia.
4. Troca entre Prédios: Consiste em trocar uma disciplina por outra já alocadas em prédio diferente, para o mesmo dia da semana e horário.
5. Disciplina/Turno/Curso/Período para mesma sala: Consiste em efetuar realocações e trocas, independentemente do prédio, visando, desde que em mesmo turno (manhã, tarde ou noite), que todas as disciplinas de um mesmo curso e período fiquem em uma mesma sala.

3.5. Avaliação de uma Solução

Uma solução é avaliada por uma função baseada em penalidades pelo não atendimento às restrições do problema. Para tal, a cada restrição i é contabilizada quantas vezes ela não é respeitada. Essa quantidade de vezes é, então, multiplicada por um peso α_i . Valores maiores para α_i indicam uma importância maior da respectiva restrição. Essa função deve ser minimizada, o que significa que menores valores para ela indicam um maior atendimento às restrições.

No sistema desenvolvido foram implementadas cinco restrições, conforme mostrado na Figura 3.



Parâmetros da Função Objetivo

Informe o valor do o peso das penalizações para cada parâmetro

Excesso de Alunos em sala (Valor x N° em Excesso):	100
Ociosidade de Alunos em sala (Valor x N° de Carteiras Vazias):	1
Alocação de um mesmo curso e turma em salas diferentes:	20
Alocação de disciplinas em prédios diferentes:	500
Alocação de disciplinas em Salas Virtuais:	200

SALVAR RESTAURAR FECHAR

Figura 3: Parametrização das Restrições

A formulação da função de avaliação se dá da seguinte forma:

1. Excesso de Alunos em sala: Determinam-se todas as salas onde a demanda de alunos é superior à capacidade da sala. Em seguida, encontra-se, para cada uma dessas salas, a diferença entre a demanda de alunos e a capacidade da sala. Essa diferença é somada até que todas as salas com excesso de alunos sejam verificadas. Ao final, multiplica-se a quantidade de alunos em excesso em todas as salas pelo peso α_1 .
2. Ociosidade em sala: Encontram-se todas as salas onde a demanda de alunos é inferior à capacidade da sala. Posteriormente, encontra-se, para cada uma dessas salas, a diferença entre a capacidade da sala e a demanda de alunos. Esta diferença é somada até que todas as salas com ociosidade sejam verificadas. Ao final, multiplica-se a quantidade de carteiras ociosas em todas as salas pelo peso α_2 .

3. Turmas de disciplinas de mesmo curso e período em salas diferentes: Determinam-se todas as turmas de disciplinas de mesmo curso e período que estejam alocadas em salas diferentes. Soma-se a quantidade de turmas nessas condições e, ao final, multiplica-se pelo peso α_3 .
4. Turmas de disciplinas em prédios diferentes: Encontram-se todas as turmas de disciplinas que, inicialmente, tinham como preferência a alocação em determinado prédio e, posteriormente, foi alocada em outro. Somam-se todas as turmas nessa condição e, ao final, multiplica-se o resultado pelo peso α_4 .
5. Turmas de disciplinas alocadas em salas virtuais: Encontram-se todas as turmas de disciplinas que foram alocadas em salas virtuais. Soma-se a quantidade de turmas nessa condição, multiplicando o resultado pelo peso α_5 .

O valor da função de avaliação é dado pelo somatório da quantidade de não atendimentos às restrições multiplicado pelo respectivo peso.

3.6. Algoritmo da Fase de Refinamento

O refinamento da solução gerada na Subseção 3.2 é feita pelo algoritmo *Simulated Annealing* – SA (Kirkpatrick et al., 1983). Pereira e Vasconcelos (2012) caracterizam o SA como uma metaheurística inspirada no processo físico de recozimento de um sólido para obtenção de estados de baixa energia na área da física da matéria condensada. Esse processo consiste em aquecer o sólido até atingir sua temperatura de fusão, para que a matéria passe do estado sólido para o líquido; posteriormente, a temperatura deve ser lentamente diminuída para evitar estados metaestáveis e obter a matéria no estado cristalino, ou seja, com energia mínima.

O algoritmo SA consiste em um laço principal que gera, a cada iteração, um único vizinho s da solução corrente s aleatoriamente. Se s for melhor que s segundo a função de avaliação, ele é aceito e passa a ser considerado a solução corrente, assumindo assim o lugar de s . Caso s represente uma solução de piora quando comparado a s por uma quantidade Δ , ele pode ser aceito se satisfizer uma probabilidade, dada por $e^{-\Delta/T}$. Nessa expressão, T , denominado Temperatura, tem valor decrescido gradualmente conforme o progresso do algoritmo. O critério de parada para esse processo é o valor de T . Quando T assume um valor pequeno, entende-se a obtenção de um ótimo local, possivelmente global.

A aceitação de movimentos de piora no algoritmo SA visa a escapar dos ótimos locais. No início da aplicação do algoritmo essa aceitação de movimentos de piora é mais intensa, mas passa a ser menos frequente no final do algoritmo.

O Algoritmo 2 mostra o pseudocódigo do SA Dowsland (1993). Nesse algoritmo, s_0 é uma solução inicial, T_0 a temperatura inicial, α a taxa de resfriamento e SAm_{max} o número máximo de iterações para se atingir o equilíbrio térmico na temperatura T . Os valores desses parâmetros é um dado de entrada do sistema, conforme mostra a Figura 4.



Figura 4: Parametrização do Simulated Annealing

Algoritmo 2: *Simulated Annealing (SA)* Dowsland (1993)

```

procedimento
   $s \leftarrow s_0$ ; {Solução Corrente}
   $s^* \leftarrow s$ ; {Melhor solução obtida até então}
   $T \leftarrow T_0$ ; {Temperatura corrente}
   $T_f \leftarrow T_i$ ; {Temperatura limitante inferior}
   $IterT \leftarrow 0$ ; {Número de iterações na temperatura  $T$ }
  enquanto  $T > T_f$  faça
    enquanto  $IterT < SAm_{ax}$  faça
       $IterT \leftarrow IterT + 1$ ;
      Gerar vizinho qualquer  $s' \in N(s)$ ;
       $\Delta = f(s') - f(s)$ ;
      se  $\Delta < 0$  então
         $s \leftarrow s'$ ;
        se  $f(s') < f(s^*)$  então
           $s^* \leftarrow s'$ ;
        fim
      senão
        Escolha aleatoriamente um valor  $x$  real no real  $[0, 1]$ ;
        se  $x < e^{-\Delta/T}$  então
           $s \leftarrow s'$ ;
        fim
      fim
    fim
     $T \leftarrow \alpha \times T$ ;
  retorna  $S^*$ 
fim procedimento
    
```

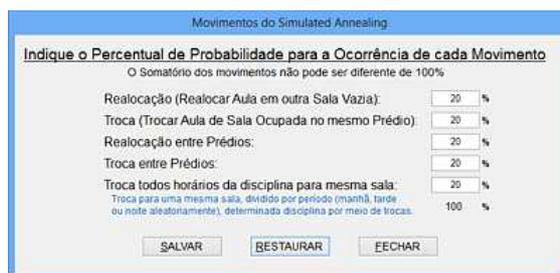
No algoritmo SA, cada uma das cinco vizinhanças implementadas (vide Subseção 3.4) é escolhida aleatoriamente a cada iteração. Por padrão, a probabilidade de escolha da vizinhança é a mesma para todas as vizinhanças. No entanto, ao usuário é dada a oportunidade de estabelecer uma diferenciação entre essas probabilidades, conforme mostra a Figura 5. Isso pode ser interessante se um movimento for considerado melhor que outro na exploração do espaço de busca.

4. RESULTADOS OBTIDOS

O algoritmo desenvolvido foi implementado na linguagem C#, no ambiente IDE Microsoft Visual Studio 2013. Os testes foram realizados em um microcomputador com processador Intel CORE I5-2450M CPU 2,49GHz, com 4 GB de memória RAM, sob o sistema operacional Windows 8 64 bits. Utilizou-se, como teste, a instância real fornecida pela UFOP, caracterizada conforme seção 2.

Dado o caráter estocástico do Algoritmo, ele foi executado trinta vezes nessa instância. Os pesos para a função de avaliação foram fixados nos valores especificados na Tabela 3.

A solução inicial foi gerada de forma totalmente gulosa. Assim, o valor inicial para a



Movimentos do Simulated Annealing

Indique o Percentual de Probabilidade para a Ocorrência de cada Movimento

O Somatório dos movimentos não pode ser diferente de 100%

Realocação (Realocar Aula em outra Sala Vazia): 20 %

Troca (Trocar Aula de Sala Ocupada no mesmo Prédio): 20 %

Realocação entre Prédios: 20 %

Troca entre Prédios: 20 %

Troca todos horários da disciplina para mesma sala: 20 %

Troca para uma mesma sala, dividido por período (manhã, tarde ou noite aleatoriamente), determinada disciplina por meio de trocas: 100 %

SALVAR RESTAURAR FECHAR

Figura 5: Parametrização dos Movimentos do Simulated Annealing

Tabela 3: Penalizações da Função Objetivo

Penalizações	Valor
Excesso de Alunos	100
Ociosidade de Carteiras	1
Mesma Turma/Curso em Salas Diferentes	20
Turma/Curso em Prédios Diferentes	500
Turma/Curso em Salas Virtuais	200

função objetivo foi sempre 767902. O movimento aplicado a cada iteração do *Simulated Annealing* foi escolhido dando-se a mesma chance a cada um dos cinco movimentos possíveis. Os valores dos parâmetros do *Simulated Annealing* foram fixados conforme Tabela 4.

Tabela 4: Parametrizações do *Simulated Annealing*

Parâmetros	Valor
Iterações por Temperatura	2500
Taxa de Resfriamento	0,98
Temperatura Inicial	5000
Temperatura de Congelamento	0,1

A Tabela 5 apresenta algumas características das soluções geradas. Como pode ser observado, a solução inicial foi melhorada em 15%, em média, com a aplicação da fase de refinamento.

Tabela 5: Resultados das Execuções

Item	Valor
Valor inicial da função de avaliação	767902
Tempo Médio de Execução	98 min
Valor médio da função de avaliação	649738
Melhor valor da função de avaliação	643570
Salas virtuais criadas no Pavilhão	3
Salas virtuais criadas no ICEB	3

A ocorrência de turmas alocadas em salas virtuais, apesar de gerar uma solução inviável, é comum em problemas de alocação de salas. Este fato acontece porque há concentração de aulas em poucos dias, normalmente às terças, quartas e quintas-feiras. Na solução apresentada, essas são as menores turmas. A solução encontrada mostra que a diretoria do ICEB tem a tarefa de encontrar salas em outras unidades acadêmicas, ou, caso isso não seja possível, alocar essas turmas pequenas a outros espaços do Instituto, em geral, em laboratórios.

5. CONCLUSÃO

Este trabalho apresentou um algoritmo heurístico para resolver o Problema de Alocação de Salas do Instituto de Ciências Exatas e Biológicas da Universidade Federal de Ouro Preto (UFOP). O algoritmo parte de uma solução inicial construída de forma gulosa e a refina pela metaheurística *Simulated Annealing*.

A fase de refinamento utiliza cinco movimentos para explorar o espaço de soluções, a saber: Realocação de turmas, Troca de turmas entre salas, Realocação de turmas entre Prédios, Troca de turmas entre Prédios e Disciplina/Turno/Curso/Período para mesma sala.

O sistema desenvolvido incluiu duas novas restrições importantes ao sistema anteriormente usado, no caso, turmas de disciplinas de mesmo período e curso procuram ser alocadas em uma mesma sala, e alocação de uma turma a um prédio previamente fixado.

Os resultados mostraram que a fase de refinamento foi capaz de melhorar a solução inicial em até 15%. Além disso, o sistema desenvolvido inclui facilidades ao usuário para manipular a solução, evitando a ocorrência de erros caso a solução fosse gerada manualmente.

Agradecimentos

Os autores agradecem à CAPES, à FAPEMIG e ao CEFET-MG, pelo apoio ao desenvolvimento deste trabalho.

Referências

- Barbosa, S. H. D. (2011). Resolução do problema de programação de cursos universitários baseada em currículos via meta-heurísticas. Dissertação de mestrado, Programa de Pós-Graduação em Modelagem Matemática e Computacional, Centro Federal de Educação Tecnológica de Minas Gerais - CEFET/MG, Belo Horizonte.
- Bardadym, V. A. (1996). Computer-aided school and university timetabling: The new wave. *Lecture Notes in Computer Science*, v. 1153, p. 22–45.
- Beyrouthy, Camille; Burke, Edmund K.; Landa-Silva, J. Dario; McCollum, Barry; McMullan, Paul e Parkes, Andrew J. (2006). Towards improving the utilisation of university teaching space. *PATAT*, v. , p. 103–122.
- Carter, Michael W. e Tovey, Craig A. (1992). When is the classroom assignment problem hard? *Operations Research*, v. 40, p. 28–39.
- Dowsland, Kathryn A. (1993). Simulated annealing. Reeves, Colin R., editor, *Modern heuristic techniques for combinatorial problems*, p. 20–69. John Wiley & Sons, Inc., New York, NY, USA.
- Even, S.; Itai, A. e Shamir, A. (1976). On the complexity of time table and multi-commodity flow problems. *SIAM Journal of Computation*, v. 5, p. 691–703.
- Kirkpatrick, S.; Gelatt, C. D. e Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, v. 220, p. 671–680.
- Pereira, Marconi A. e Vasconcelos, Joao A. (2012). *Manual de Computação Evolutiva e Metaheurística*, Capítulo 8, *Recozimento Simulado*, p. 163–176. UFMG, Belo Horizonte.
- Politano, Gustavo Lico Cunha. (2006). Modelo e algoritmo para alocação de espaço físico. Monografia - (Ciência da Computação) Faculdade de Ciência da Computação, Universidade Federal de Maringá.
- Schaerf, A. (1999). A survey of automated timetabling. *Lecture Notes in Computer Science*, v. 13, p. 87–127.



- Souza, Marcone Jamilson Freitas; Martins, Alexandre Xavier e de Araujo, Cássio Roberto. (2002). Experiências com simulated annealing e busca tabu na resolução do problema de alocação de salas. *XXXIV Simpósio Brasileiro de Pesquisa Operacional*, v. 34, p. 1100–1110.
- Subramanian, Anand; Medeiros, José Maurício Fernandes; Cabral, Lucídio Formiga e Souza, Marcone Jamilson Freitas. (2011). Aplicação da metaheurística busca tabu ao problema de alocação de aulas a salas em uma instituição universitária. *Revista Produção Online*, v. 11, p. 54–75.
- Ueda, Hiroaki; Ouchi, Daisuke; Takahashi, Kenichi e Miyahara, Tetsuhiro. (2001). A co-evolving timeslot/room assignment genetic algorithm technique for university timetabling. *Lecture Notes in Computer Science*, v. 2079, p. 48–63.