APLICAÇÃO DA METAHEURÍSTICA AGC PARA O PROBLEMA DE ALOCAÇÃO DE AULAS À SALAS

Rafael Bernardo Zanetti Cirino
Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo (ICMC-USP)
Avenida Trabalhador são-carlense, 400 - Centro CEP: 13566-590 - São Carlos - SP.
rafaelbzc@usp.br

Maristela Oliveira Santos
Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo (ICMC-USP)
Avenida Trabalhador são-carlense, 400 - Centro CEP: 13566-590 - São Carlos - SP.
mari@icmc.usp.br

Alexandre Cláudio Botazzo Delbem
Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo (ICMC-USP)
Avenida Trabalhador são-carlense, 400 - Centro CEP: 13566-590 - São Carlos - SP.
acbd@icmc.usp.br

RESUMO

Neste trabalho consideramos um problema de alocação de aulas às salas motivados por um caso real em uma universidade brasileira. O problema consiste em, dado um conjunto de aulas e um conjunto de salas, determinar em quais salas cada aula será alocada de modo a atender medidas de qualidade determinadas pelos gestores da universidade. Na alocação não pode haver sobreposições de horários das aulas nas salas e as aulas só podem ser alocadas em salas que as comportem por capacidade e recursos. Cada instituição de ensino possui suas características específicas, restrições e preferências, que devem ser atendidas. Para obter as alocações, propomos um modelo de programação inteira e um Algoritmo de Estimação de Distribuição (AED) - Algoritmo Genético Compacto (CGA). As duas abordagens proporcionam soluções promissoras. A meta-heurística obtém melhores soluções para instâncias maiores, enquanto a abordagem pela resolução do modelo é melhor para instâncias menores.

PALAVARAS CHAVE. Progamação Inteira, Algoritmo Genético Compacto, Alocação de Aulas à Salas, Algoritmo de Estimação de Distrituição.

ABSTRACT

This paper addresses the classroom assignment problem motivated by a real problem case in a Brazilian university. The problem can be described as: given a set of classes and a set of rooms, determine in which room each class should be assigned; not allowing the overlapping of classes in a room and classes can only be assigned to rooms with capacities and with resources to support them. Each educational institute has specific features, constraints and preferences, which should be attended. An Integer Programming Model and a Estimation of Distribution Algorithm (EDA) - Compact Genetic Algorithm (CGA) are proposed to obtain solutions to the problem. Both approaches are capable to obtain promising solutions. The meta-heuristics solutions are better for larger instances, on the other hand the mathematical model obtains better solutions for smaller-sized instances.

KEYWORDS. Integer Programming. Compact Genetic Algorithm. Classroom Assignment Problem, Estimation of Distribution Algorithm.

1. Introdução e Revisão Bibliográfica

Este trabalho propõe-se resolver o Problema de Alocação de Aulas à Salas (PAAS) por meio da metaheurística Algoritmo Genético Compacto (AGc), um algoritmo de estimação de distribuição proposto por Harik et al. (1999), e por meio de programação inteira. Segundo Mulvey (1982), o PAAS consiste em: dado um conjunto de atividades (aulas, práticas, campo, etc.) - com horários, docentes, discentes e requerimentos (acessibilidade para cadeirantes, vidraria, computadores, videoconferência, etc.) predefinidos - deve-se definir o local (laboratório, auditório, sala de aula, etc.) onde cada uma destas atividades irá ocorrer. De acordo com Carter e Tovey (1992) este é um problema NP-Completo, o que dificulta a sua resolução para alguns conjuntos de dados.

Waterer (1995) e Carter e Tovey (1992) destacam ainda que o PAAS não é um problema único, pois matematicamente é o mesmo problema de alocar quartos de hotel, alocação de navios em docas, alocação de portões de embarques para aeronaves e planejamento de tarefas para máquinas sem capacitação.

Este é um problema que atinge várias instituições de ensino superior do mundo no começo de seus períodos letivos, por exemplo: Waterer (1995) estuda-o na Universidade de Auckland; Mooney (1995) na Universidade de Purdue e Al-Yakoob e Sherali (2006) na Universidade do Kuwait onde o problema é estudado como parte integrante dos problemas de Programação de Horários Escolares (*School Timetabling*).

A primeira tentativa de resolução deste problema por meio de abordagem matemática foi de Dyer e Mulvey (1976) para a UCLA ((Universidade da Califórnia). Porém era somente um dos problemas abordados no artigo, cujo o foco principal era a automatização de serviços. Mais tarde foi estendido por Mulvey (1982) para o primeiro modelo de alocação de aulas à salas.

Na literatura é comum encontrar diversas abordagens de resolução para o problema. Destacam-se as meta-heurísticas: Recozimento Simulado (*Simulated Annealing*) nos trabalhos de Kripka e Kripka (2010), Souza et al. (2002), Matinez-Alfaro e Flores-Teran (1998), Nascimento et al. (2005) e Prado e Souza (2014); Busca Tabu em Souza et al. (2002), Subramanian et al. (2011) e Souza et al. (2004); além de abordagens por Coloração de Grafos de Silva e Silva (2009). Lopes e Schoeffel (2002) propõe uma heurística construtiva e Constantino et al. (2010) utilizam-se de técnicas de geração de colunas para aproximação da solução do modelo inteiro.

Algoritmos que utilizam-se de técnicas evolutivas não são tão comumente utilizados para este problema, pois geralmente os operadores genéticos tradicionais de *crossover* e mutação geram soluções infactíveis. Neste caso, pode ser necessário que haja uma função de reparação para factibilizar a solução obtida, tornando-os assim menos atrativos (Fernandes et al. (1999)).

O PAAS possui três restrições fortes que são características do problema:

1. Todas as aulas devem ser alocadas em alguma sala;
2. As aulas não podem ser sobrepostas na sala, ou seja, duas aulas com sobreposição de horário não podem ser simultaneamente alocadas na mesma sala;
3. As aulas só poderão ser alocadas em salas que atendam suas demandas de requisitos e capacidade para acomodar todos os participantes.

Alguns trabalhos relaxam algumas destas restrições, por exemplo, Subramanian et al. (2011) relaxam as restrições de alocação de todas as aulas, porém, penalizam as violações com altíssimos custos na função objetivo, tornando-a virtualmente uma restrição forte. Estas três restrições caracterizam o problema básico de alocação de aulas à salas. enquanto as características peculiares de cada local em que o PAAS é resolvido determinam as medidas de qualidade das soluções. Para este trabalho um estudo de caso foi realizado no Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (ICMC-USP).

A Seção 2 descreve o caso de estudo e o modelo proposto, enquanto a Seção 3 apresenta a metaheurística AGc utilizada para resolver o problema. A Seção 4 expõe alguns testes computacionais e resultados. Por fim, a Seção 5 contém as conclusões do trabalho e perspectivas para trabalhos futuros.

2. Caso de Estudo e Modelo Matemático

O caso de estudo adotado para o problema foi o Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (ICMC-USP), em que somente as disciplinas da graduação são consideradas. O ICMC-USP conta com 23 salas de aula distribuídas em 4 blocos didáticos e ministra cerca de 260 aulas para, em média, 140 turmas.

Neste estudo cada turma esta associada a um ou mais perfis de alunos, que são conjuntos de discentes que dividem o mesmo curso e ano de ingresso e portanto estão cursando um conjunto pré-estabelecido de turmas. Estas turmas podem ter uma ou mais aulas semanais, que possuem associadas requerimentos de recursos e número de inscritos.

As salas possuem recursos disponíveis e capacidade conhecida. Assim para que uma aula possa ocorrer em uma sala, ela deve disponibilizar os recursos requeridos da aula e ter capacidade superior ao número de inscritos. Além disso, algumas salas são preferencialmente vazias, ou seja, seu uso deve ser evitado para aulas curriculares.

As aulas também estão associadas a um ou mais *slots* de horário, que são divisões de tempo dentro da semana, como mostra a Figura 1. Na figura os *slots* estão ordenados.

	Segunda	Terça	Quarta	Quinta	Sexta
07h30 às 08h20	**1**	**16**	**31**	**46**	**61**
08h20 às 09h10	**2**	**17**	**32**	**47**	**62**
09h10 às 10h00	**3**	**18**	**33**	**48**	**63**
10h20 às 11h10	**4**	**19**	**34**	**49**	**64**
11h10 às 12h00	**5**	**20**	**35**	**50**	**65**
13h30 às 14h20	**6**	**21**	**36**	**51**	**66**
14h20 às 15h10	**7**	**22**	**37**	**52**	**67**
15h10 às 16h00	**8**	**23**	**38**	**53**	**68**
16h20 às 17h10	**9**	**24**	**39**	**54**	**69**
17h10 às 18h00	**10**	**25**	**40**	**55**	**70**
18h00 às 18h50	**11**	**26**	**41**	**56**	**71**
19h00 às 19h50	**12**	**27**	**42**	**57**	**72**
19h50 às 20h40	**13**	**28**	**43**	**58**	**73**
21h00 às 21h50	**14**	**29**	**44**	**59**	**74**
21h50 às 22h40	**15**	**30**	**45**	**60**	**75**

Figura 1 - Representação de *slots* de horário

As medidas de qualidade adotadas pelos responsáveis pelo PAAS no ICMC-USP são:

1. Alocar aulas em salas que possuem capacidade o mais próximo o possível do número de inscritos da aula;
2. Alocar todas as aulas de uma mesma turma em uma mesma sala;
3. Reduzir o deslocamento dos perfis de aluno, ou seja, alocar todas as aulas dos pefis em salas próximas umas das outras;
4. Evitar uso de salas preferencialmente vazias;
5. Alocar as aulas dos perfis respeitando as preferências por salas dos mesmos.

Antes da construção do modelo matemático, faz-se um pré-processamento dos dados disponíveis para extrair alguns meta-dados para facilitar a modelagem do problema. Por exemplo, os conjuntos de conflitos de aula e a matriz de atendimento que indica quando uma sala é apta para alocar uma aula. Considerando então os cinco objetivos do ICMC-USP e as três restrições fortes características do problema o modelo proposto para o PAAS é dado por:

Conjuntos considerados:

H Conjunto de *slots* de Horário ($h \in \{1, \dots, H\}$);

R Conjunto de recursos ($r \in \{1, \dots, R\}$);

P	Conjunto dos perfis de turma ($p \in \{1, \dots, P\}$);
S	Conjunto de salas ($s \in \{1, \dots, S\}$);
T	Conjunto de turmas ($t \in \{1, \dots, T\}$).

Dados de entrada:

ρ_{th}	Matriz binária de turma t por horário h;
σ_{sr}	Matriz binária de salas s e recursos r;
τ_{tr}	Matriz binária de turmas t e recursos r;
υ_{tp}	Matriz binária de turma t por perfis p;
$\varphi_{ss'}$	Matriz de distâncias entre salas s e s';
χ_{ps}	Matriz de preterências dos perfis p por salas s;
CAP_s	Capacidade máxima da sala s;
PV_s	Vetor binário indicando se sala s é preferencialmente vazia;
TAM_t	Quantidade de alunos inscritos na turma t.

Dados de preprocessamento:

A	Conjunto de aulas ($a \in \{1, \dots, A\}$);
t_a	Turma $t \in T$ da qual a aula a faz parte;
$\theta_t \subseteq A$	Conjunto de aulas da turma t;
$\mu_p \subseteq A$	Conjunto de aulas do perfil p;
$CH^a \subseteq A$	Conjunto de aulas que possuem conflito com aula a;
$S^* \subseteq S$	Conjunto de salas preferencialmente vazias;
η_{as}	Matriz binária de aulas a por salas s, aonde $\eta_{as} = 1$ indica que aula a pode ser alocada na sala s;
c_{as}	Matriz de custo de alocação, $c_{as} \doteq \lfloor 100 \cdot (1 - TAM_{t_a}/CAP_s) \rfloor$.

Variáveis de decisão:

x_{as}	Variável binária indicando se aula a esta alocada na sala s;
y_t	Variável inteira indicando o número de trocas de sala da turma t;
w_{ps}	Variável binária indicando se perfil p possui aulas alocadas na sala s;
k_{psi}	Variável binária indicando se perfil p possui aulas alocadas nas salas s, i.

Modelo Proposto:

$$\min \quad z = \alpha\left(\sum_{a \in A}\sum_{s \in S} c_{as}x_{as}\right) + \beta\left(\sum_{t \in T} y_t\right) + \gamma\left(\sum_{p \in P}\sum_{s=1}^{|S|-1}\sum_{i=s+1}^{|S|} \varphi_{si}k_{psi}\right) + \delta\left(\sum_{a \in A}\sum_{s \in S^*} x_{as}\right) + \varepsilon\left(\sum_{p \in P}\sum_{s \in S} \chi_{ps}w_{ps}\right) \qquad \text{(I)}$$

s.a.

$$\sum_{s \in S} x_{as} = 1 \qquad a \in A \qquad \text{(II)}$$

$$x_{as} \leq \eta_{as} \qquad a \in A, \quad s \in S \qquad \text{(III)}$$

$$x_{as} + x_{a's} \leq 1 \qquad a \in A, s \in S, a' \in CH^a \qquad \text{(IV)}$$

$$\sum_{n=1}^{|\theta_t|} x_{a_n s_n} \leq 1 + y_t \qquad t \in T, a_n \in \theta_t, \{s_1, \dots, s_n\} \subseteq S \qquad \text{(V)}$$

$$x_{as} \leq w_{ps} \qquad a \in A, s \in S, p \in \mu_p \qquad \text{(VI)}$$

$$2k_{psi} \leq w_{ps} + w_{pi} \qquad p \in P, s \in S, i \in S \qquad \text{(VII)}$$

$$k_{psi} \geq w_{ps} + w_{pi} - 1 \qquad p \in P, s \in S, i \in S \qquad \text{(VIII)}$$

$$x_{as} \in \{0,1\} \qquad a \in A, s \in S \qquad \text{(IX)}$$
$$y_t \in \mathbb{Z}_+ \qquad t \in T \qquad \text{(X)}$$
$$w_{ps} \in \{0,1\} \qquad p \in P, s \in S \qquad \text{(XI)}$$
$$k_{psi} \in \{0,1\} \qquad \in P, s \in S, i \in S \qquad \text{(XII)}$$

A equação (I) é a função objetivo, em que α é o peso para o primeiro objetivo - minimizar o percentual de assentos vazios nas salas de aula -, β indica peso para o número de trocas de sala de uma mesma turma, γ esta associado à máximização da preferência dos perfis (minimizando a preterência) e δ esta associado ao uso de salas preferencialmente vazias e o parâmetro ε pondera o deslocamento das turmas, considerando as distâncias entre as salas.

O conjunto de restrições (II) garante que uma aula será alocada em uma, e apenas uma, sala. As restrições (III) garantem que os requisitos das aulas serão atendidos e que nenhuma aula será alocada em sala que não tenha espaço ou recursos necessários. As restrições (IV) evitam a sobreposição de horários dentro das salas, i.e., aulas com horário conflitante não podem ser alocadas simultaneamente na mesma sala. As restrições (V) fazem a ligação das variáveis de decisão x_{as} e y_t para medir o número de trocas de sala das turmas, considerando que o sub-índice n varia de 1 até o número de aulas da turma implica que $\{a_1, \dots, a_n\} = \theta_t$, ao passo que $\{s_1, \dots, s_n\}$ representa um suconjunto (portanto sem repetição) do conjunto de salas, i.e., são todos os arranjos de tamanho $|\theta_t|$ do conjunto S. Enquanto as restrições (VI) fazem a ligação das variáveis x_{as} e w_{ps}. Os conjuntos de restrições (VII) e (VIII) fazem a ligação, por meio de um e lógico, entre as variáveis k_{psi} e w_{ps}, fazendo $k_{psi} = 1$ se, e somente se, $w_{ps} = 1$ e $w_{pi} = 1$. Por fim os conjuntos de restrições (IX)-(XII) delimitam domínio de variáveis.

3. Algoritmo Genético Compacto (AGc)

O Algoritmo Genético Compacto é uma variação proposta por Harik et. al. (1999) do Algoritmo Genético proposto por Holland (1975). No AGc as populações explícitas são substituídas por modelos probabilísticos e os operadores genéticos tradicionais de *crossover* e mutação não existem explicitamente, caracterizando-o como um Algoritmo de Estimação de Distribuição (AED). A ideia geral do algoritmo é começar com um modelo probabilístico da população e amostrar dele algumas soluções que são confrontadas por meio de torneio. A solução vencedora faz com que o modelo probabilístico seja alterado para adicionar viés probabilísticos à soluções similares.

Uma solução do problema pode ser representada por uma matriz binária $\boldsymbol{X}$, tal que $X_{as} = 1$ significa que a aula a esta sendo ministrada na sala *s*, como no modelo proposto. Almejando soluções factíveis, deve-se garantir que para cada aula $a \in A$ somente uma sala será alocada e que sobreposições de horários das aulas não ocorram.

Propõe-se assim o modelo probabilístico adotado que terá a forma matricial dada por $\boldsymbol{MP}$. As entradas de MP_{as} serão valores reais entre 0 e 1, indicando a probabilidade de que a sala s seja escolhida para alocar a aula a. Este modelo inicial deverá possibilitar que qualquer solução factível seja amostrada, ao passo de que as soluções infactíveis podem ser ignoradas, portanto somente as salas que atendem as aulas serão consideradas.

Seja então $S^a \subseteq S$, o conjunto das salas que uma aula a pode ser alocada, i.e. as salas deste conjunto possuem recursos e capacidade suficiente para a realização da aula ($\eta_{as} = 1$). Define-se então o modelo probabilístico inicial pela expressão (1). Note-se que se para $a \in A$, e se $S^a = \emptyset$ então o problema não possui solução factível, portanto $|S^a| > 0$ e a expressão (1) está bem definida.

$$MP_{as}^0 = \begin{cases} 1/|S^a|, se\ s \in S^a \\ 0, c.c. \end{cases}, a \in A \qquad (1)$$

Visando obter somente soluções factíveis ou *quasi*-factíveis (com poucas violações de factibilidade) a técnica de amostragem do modelo probabilístico será uma heurística construtiva com viés probabilístico. Assim, em cada geração i, para cada aula $a \in A$, gira-se uma roleta assimétrica de salas, em que a probabilidade de cada sala ser escolhida será igual a MP_{as}^i.

Visando não permitir sobreposição de horário, quando a sala s' é escolhida para a aula a' esta mesma sala não pode ser escolhida para nenhuma aula que possua conflito direto de horário com a aula a' ($a' \in CH^a$). Neste caso retira-se a sala dos conjuntos de salas disponíveis para essas aulas.

Se em qualquer ponto da iteração, o conjunto $S^a = \emptyset$ implica que a solução que será amostrada será infactível, pois violará uma das 3 restrições fortes. Este é o caso da linha 8 do Algoritmo 1, em que o valor de π é a probabilidade conjunta de todas as salas que já não podem mais ser escolhidas para a aula a. Escolhe-se então uma sala qualquer que viole somente a terceira restrição, permitindo que a aula seja sobreposta nesta sala. Soluções infactíveis sofrem uma severa punição na função de *fitness*. Caso seja factível, será o valor da função objetivo dada pelo modelo. Um pseudo-código do algoritmo é dado pelo Algoritmo 1 e ilustrado pela Figura 2.

Algoritmo 1: Amostragem em Modelo Probabilístico

Entrada: $\boldsymbol{MP}$; $\boldsymbol{\eta}$; $CH^a, a \in A$
Saida: $\boldsymbol{X}$

1. Iniciar $X_{as} = \begin{cases} -1, se\ \eta_{as} = 1 \\ 0, se\ \eta_{as} = 0 \end{cases}$
2. $a = 1$;
3. **Enquanto** $a \leq |A|$ **faça:**
4. **Faça** $\pi = \sum_{s \in S} MP_{as}$, $x_{as} = 0$
5. **Se** $\pi < 1$ **então:**
6. **Escolha** s' por roleta assimétrica, dada por MP_{as}.
7. **Senão:**
8. **Escolha** s' aleatoriamente com $\eta_{as'} = 1$.
9. **Fim**
10. **Faça** $x_{as'} = 1$;
11. **Faça** $x_{as} = 0, s \in S, s \neq s'$;
12. **Faça** $x_{a's'} = 0, a' \in CH^a$;
13. $a = a + 1$;
14. **Fim**
15. **Retorne** X_{as} (Solução)

O Algoritmo 1 terá como entrada o Modelo Probabilístico ($\boldsymbol{MP}$), a matriz de atendimento ($\boldsymbol{\eta}$) e para cada aula o conjunto de aulas conflitantes ($CH^a, a \in A$). A saída será uma solução para o problema ($\boldsymbol{X}$). A linha 1 inicia a solução $\boldsymbol{X}$ com base na matriz de atendimento $\boldsymbol{\eta}$ (Figura 2 a).

Da linha 2 até a linha 14 é a fase iterativa do método, na linha 4. calcula-se a variável auxiliar π, esta irá somar a probabilidade de todas as aulas que estão proibidas de serem alocadas, caso este somatório seja 1 implica que não existem salas disponíveis para aquela aula (linha 8) e portanto uma violação de factibilidade deverá ser feita; caso este somatório seja menor que 1, implica que existem salas ainda disponíveis e a linha 6. irá escolhe uma destas salas disponíveis com base no Modelo Probabilístico.

Por fim quando uma sala é escolhida para a aula em questão altera-se a solução, na linha 10 inclui-se a alocação aula na sala, enquanto na linha 11 proibe a alocação da aula para outras salas; na linha 12 proíbe que cada aula que possui conflito seja alocada nesta sala. O algoritmo então prossegue para a próxima aula, linha 13, e repete o processo de escolha de sala.

A atualização do modelo é feita segundo o parâmetro de tamanho da população n. Assim de um Modelo Probabilístico em uma geração i ($\boldsymbol{MP^i}$) amostram-se duas soluções e para cada calcula-se o *fitness*. O *fitness* será, no caso da solução ser factível, o valor da função objetivo do modelo; e caso contrário será o valor da função objetivo acrescida de um penalidade M. Após o torneio, as soluções são classificadas em vencedora ($\boldsymbol{x^v}$) e perdedora ($\boldsymbol{x^p}$) e o modelo é atualizado para a geração $i + 1$ conforme a equação (2).

$$MP_{as}^{i+1} = \begin{cases} MP_{as}^{i} + {}^{1}\!/_{n} \text{ }, se\ x_{as}^{v} = 1\ e\ x_{as}^{p} = 0 \\ MP_{as}^{i} - {}^{1}\!/_{n} \text{ }, se\ x_{as}^{v} = 0\ e\ x_{as}^{p} = 1 \\ MP_{as}^{i} \text{ }, c.c. \end{cases} \quad (2)$$

Portanto o AGc para o PAAS utiliza-se da equação (1) para iniciar o modelo probabilístico, amostra-se soluções do modelo por meio do Algoritmo 1 e por meio de um torneio atualiza o modelo segundo a equação (2). Repete o processo até que seja cumprida a condição de parada (que pode ser a convergência do modelo probabilístico, quando todas as entradas são 0 ou 1 para alguma geração, ou número de gerações). Um pseudo-código do AGc é dado pelo Algoritmo 2.

	Sala1	Sala 2	Sala3	Sala4	Sala5	Sala6	Sala7
Aula1	-1	-1	0	-1	-1	0	-1
Aula2	-1	-1	-1	0	-1	-1	-1
Aula3	-1	-1	-1	0	-1	-1	-1
Aula4	-1	-1	-1	0	-1	-1	-1
Aula5	-1	0	-1	-1	-1	-1	-1
Aula6	-1	-1	-1	-1	-1	-1	-1
Aula7	-1	-1	-1	-1	0	-1	-1
Aula8	-1	-1	-1	-1	-1	-1	-1
Aula9	-1	-1	-1	-1	-1	-1	0
Aula10	-1	-1	0	-1	-1	-1	0
Aula11	-1	-1	0	-1	-1	-1	0
Aula12	-1	-1	-1	-1	-1	-1	0
Aula13	-1	-1	-1	-1	-1	-1	0
Aula14	-1	-1	-1	-1	-1	-1	-1

Figura 2 a - Inicio

	Sala1	Sala 2	Sala3	Sala4	Sala5	Sala6	Sala7
Aula1	0	0	0	0	1	0	0
Aula2	-1	-1	-1	0	-1	-1	-1
Aula3	-1	-1	-1	0	0	-1	-1
Aula4	-1	-1	-1	0	-1	-1	-1
Aula5	-1	0	-1	-1	-1	-1	-1
Aula6	-1	-1	-1	-1	-1	-1	-1
Aula7	-1	-1	-1	-1	0	-1	-1
Aula8	-1	-1	-1	-1	-1	-1	-1
Aula9	-1	-1	-1	-1	0	-1	0
Aula10	-1	-1	0	-1	-1	-1	0
Aula11	-1	-1	0	-1	-1	-1	0
Aula12	-1	-1	-1	-1	-1	-1	0
Aula13	-1	-1	-1	-1	-1	-1	0
Aula14	-1	-1	-1	-1	-1	-1	-1

Figura 2 b - Primeira Iteracao

	Sala1	Sala 2	Sala3	Sala4	Sala5	Sala6	Sala7
Aula1	0	0	0	0	1	0	0
Aula2	1	0	0	0	0	0	0
Aula3	0	1	0	0	0	0	0
Aula4	-1	-1	-1	0	-1	-1	-1
Aula5	-1	0	-1	-1	-1	-1	-1
Aula6	-1	-1	-1	-1	-1	-1	-1
Aula7	-1	-1	-1	-1	0	-1	-1
Aula8	-1	-1	-1	-1	-1	-1	-1
Aula9	-1	0	-1	-1	0	-1	0
Aula10	-1	-1	0	-1	-1	-1	0
Aula11	-1	-1	0	-1	-1	-1	0
Aula12	-1	-1	-1	-1	-1	-1	0
Aula13	-1	-1	-1	-1	-1	-1	0
Aula14	-1	-1	-1	-1	-1	-1	-1

Figura 2 c - Algumas Iterações

Figura 2- Processo de obtenção de Solução (Quasi-)Factível

A Figura 2 ilustra o processo de obtenção de uma solução quasi-factível. As células alaranjadas são as alocações proibidas, o azul indica a escolha de sala para aula. As aulas marcadas com cores distintas do branco são aulas que possuem conflito de horário entre si, ou seja, não podem ser alocadas na mesma sala. Portanto entre a Figura 2 a e Figura 2 b a Aula 3 e a Aula 9 são alteradas com base na escolha feita para a Aula 1, ilustrando a linha 12. do Algoritmo 1. A Figura 2 c ilustra o estado do algoritmo após algumas iterações, notando que a Aula 9 foi alterada 2 vezes, uma pela escolha da Aula 1 e outra pela escolha da Aula 3.

O Algoritmo 2 recebe de entrada um tamanho de população virtual ($\boldsymbol{n}$), um número máximo de iterações ($\boldsymbol{maxIt}$), a matriz de atendimento ($\boldsymbol{\eta}$), o conjunto de salas disponíveis para cada aula ($S^{a}, a \in A$) e o conjunto de aulas conflitantes para cada aula ($CH^{a}, a \in A$). A saída é uma solução do problema ($\boldsymbol{X}^{*}$).

Primeiramente deve-se iniciar o Modelo Probabilístico inicial ($\boldsymbol{MP^{0}}$) usando a equação (1), ou seja, cada uma das salas possíveis para cada aula deverá ter inicialmente a mesma probabilidade de ser amostrada (linha 1). Inicia-se então a melhor solução encontrada até o momento ($\boldsymbol{X}^{*}$) por amostragem utilizando-se do Algoritmo 1.

A linha 3 inicia o contador de gerações e enquanto este contador for menor que o número máximo de iterações serão executados da linha 5 até linha 14, sendo que a linha 12 irá conferir se o Modelo Probabilístico convergiu, i.e., todas as entradas 1 ou 0 e em caso positivo termina a execução retornando a melhor solução obtida até o momento. Na linha 5 por meio do Algoritmo

1 faz-se a amostragem de 2 soluções do Modelo Probabílistico da geração atual, assim na linha 6 classifica as soluções em vencedora e perdedora segundo o *fitness*. Caso a melhor solução seja melhor que a melhor até o momento salva-a em X_{as}^{*}, como mostra as linhas 7 e 8.

Algoritmo 2: Algoritmo Genético Compacto (AGc)

Entrada: n ; $maxIt$; η_{as} ; $S^{a}, a \in A$; $CH^{a}, a \in A$
Saida: X_{as}^{*}
1. Iniciar MP_{as}^{0} usando equação (1);
2. **Faça** X_{as}^{*} solução amostrada de MP_{as}^{0} utilizando o Algoritmo 1;
3. $g = 0$;
4. **Enquanto** $g \leq maxIt$ **faça:**
5. | **Amostre** X_{as}^{1} e X_{as}^{2} de MP_{as}^{g};
6. | **Ordene** X_{as}^{1} e X_{as}^{2} por *fitness* e **classifique-as** por X_{as}^{v} e X_{as}^{p};
7. | **Se** $fitness(X_{as}^{v}) \leq fitness(X_{as}^{*})$ **então:**
8. | | $X_{as}^{*} = X_{as}^{v}$;
9. | **Fim**
10. | **Atualizar** MP_{as}^{g+1} segundo X_{as}^{v} e X_{as}^{p} pela equação (2);
11. | $g = g + 1$;
12. | **Se** MP_{as}^{g} convergiu **então:**
13. | | **Retorne** X_{as}^{*};
14. | **Fim**
15. **Fim**
16. **Retorne** X_{as}^{*}

A linha 10 faz então a atualização do Modelo Probabilístico para a próxima geração tomando por base as duas soluções amostradas e aplicando a equação (2). Adiciona-se então uma geração ao contador de gerações e faz a verificação se o modelo convergiu, caso tenha convergido retorne a melhor solução obtida até o momento, caso contrário volte para a linha 5 e recomeça o processo. Ao fim a melhor solução obtida é retornada como solução do método.

4. Testes Computacionais

Todos os testes foram realizados em um computador Intel Core i7 com clock de 2,6 GHz e 8 GB de memória RAM. Os códigos para o AGc foram implementados em JAVA utilizando-se da NetBeans IDE 8.0.2 e os códigos para utilização da biblioteca comercial ILOG IBM CPLEX 12.4 (para resolução exata do modelo) foram feitos utilizando-se da tecnologia concert para c++ implementados em Microsoft Visual C++ 2008.

Como a incorporação de um banco de dados específico está em fase final de implementação a obtenção de instâncias reais de dados toma muito tempo. Para contornar este fato foi implementado um gerador de instâncias, em que o número de *slots* de horário (H), turmas(T), salas(S), recursos(R) e perfis(P) são fornecidos. Os dados de entrada do modelo são amostrados aleatoriamente tomando como base as instâncias reais, foi gerado por este método 3 instâncias de testes com tamanhos variados. Além disso, as instâncias geradas tem como objetivo verificar a escalonabilidade dos métodos. Uma instância real também foi obtida, referente ao primeiro semestre de 2015 do ICMC. As dimensões destas instâncias estão descritas na Tabela 1.

Tabela 1 - Descrição das Instâncias

Nome	H	T	S	R	P
Ficticia1	60	50	10	3	5
Ficticia2	60	200	40	4	40
Ficticia3	60	250	40	3	50
Real	85	143	23	5	39

Foi também definido 3 conjuntos de parâmetros (valores de α, β, γ, δ, ε) para o modelo, de forma a corresponder às prioridades quanto aos objetivos da instituição. Para cada instância e cada parametrização o modelo matemático foi executado por 10 minutos, com ênfase em obtenção de soluções factíveis, e a melhor solução factível obtida é considerada. Os resultados encontram-se na Tabela 2, em que a coluna de GAP é referente ao melhor limitante dual encontrado. Note-se que para a instância Ficticia3 não foi encontrada nenhuma solução factível durante o tempo de execução para duas parametrizações.

Tabela 2 - Resultados do Modelo Matemático

Instância	Parametrização					Tempo	Função Objetivo	GAP
	α	β	γ	δ	ε			
Ficticia1	0,1	10000	10	1000	100	1,58s	24805,4	0%
	1	5000	5	2000	500	3,27s	53884	0%
	10	20000	100	3000	1000	2,75s	218920	0%
Ficticia2	0,1	10000	10	1000	100	600s	273216	83,09%
	1	5000	5	2000	500	600s	400841	74,57%
	10	20000	100	3000	1000	600s	3298730	82,87%
Ficticia3	0,1	10000	10	1000	100	600s	1017180	87,98%
	1	5000	5	2000	500	600s	-	-
	10	20000	100	3000	1000	600s	-	-
Real	0,1	10000	10	1000	100	600s	17600,8	0,75%
	1	5000	5	2000	500	600s	92184	0,10%
	10	20000	100	3000	1000	600s	238970	0,92%

Para a resolução por meio da meta-heurística Algoritmo Genético Compacto proposta foi escolhido o tamanho da população virtual igual a 3000. O AGc foi executado 5 vezes para cada instância e parametrização e os resultados para melhor solução, pior solução e solução mediana encontram-se na Tabela 3. A coluna do GAP foi calculada apartir do limitante inferior encontrado pela resolução do modelo matemático comparado com o valor mediano encontrado. Para duas parametrizações da instância Ficiticia3 não foi encontrado sequer limitante inferior pelo método exato e portanto o GAP não foi calculado para estas instâncias.

A principal diferença de performance entre o AGc e a resolução do modelo encontra-se pelo tamanho das instâncias. Para a instância Ficticia1 e Real a resolução exata tem mostrado bons resultados, considerando que para Ficticia1 conseguiu provar otimalidade em pouco tempo. No entretanto para instâncias maiores a própria construção do modelo já toma muito tempo e o AGc obteve resultados melhores, incluindo duas parametrizações da instância Ficitica3 em que obteve resolutados factíveis em todas as execuções contra nenhum resultado encontrado pela resolução do modelo.

Tabela 3 - Resultados para o AGc

Instância	Parametrização					Mínimo	Máximo	Mediana	GAP
	α	β	γ	δ	ε				
Ficticia1	0,1	10000	10	1000	100	42080,3	54352,5	47980,6	48,3%
	1	5000	5	2000	500	68074	79444	76712	29,8%
	10	20000	100	3000	1000	314720	381400	344050	36,4%
Ficticia2	0,1	10000	10	1000	100	273701,4	344225,2	291487	84,1%
	1	5000	5	2000	500	336145	587939	402252	74,7%
	10	20000	100	3000	1000	1510590	3685680	2540040	77,8%
Ficticia3	0,1	10000	10	1000	100	761374,7	1785870,1	1245325,9	90,2%
	1	5000	5	2000	500	525465	853809	681710	-
	10	20000	100	3000	1000	2706190	6088620	5726990	-
Real	0,1	10000	10	1000	100	38298,3	49158,4	41564,6	57,9%
	1	5000	5	2000	500	119406	143380	136554	32,6%
	10	20000	100	3000	1000	362170	436320	383570	38,3%

5. Conclusões e Perspectivas

Neste trabalho foi abordado o Problema de Alocação de Aulas à Salas (PAAS) modelado segundo os critérios do Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (ICMC-USP). Além do modelo matemático, foi proposta uma abordagem utilizando-se de uma meta-heurística baseada em Algoritmo Genético Compacto (cGA). A baixa demanda de memória deste algoritmo, por substituir populações por representações estatísticas, é um diferencial em sua utilização em comparação ao genético convencional.

Considere ainda que diferentemente de outras abordagens evolutivas em que a população inicial determina um viés no espaço de soluções por onde o algoritmo irá trafegar, pois as soluções herdarão as características dos progenitores. No AGc, no entretanto, o espaço inteiro é oferecido para a amostragem no inicio e o viés estatístico é adicionado, restringindo o espaço de busca.

Alguns ajustes devem ser feitos ainda para uma melhor aplicação desta meta-heurística ao PAAs. Alguns tópicos que podem ser vistos para aperfeiçoamento do AGC:

- Modificação na técnica de amostragem, ordenando as aulas antes de começar o processo iterativo de alocação com viés estatístico;
- Testes paramêtricos para identificar como o parâmetro de tamanho da população afeta a convergência e qualidade das soluções obtidas;
- Investigações a respeito do torneio e possibilidades de expansão o mesmo para mais do que dois indíviduos.

Agradecimentos

Os autores gostariam de agradecer à CAPES pelo financiamento.

Referências

Al-Yakoob, S. M. e Sherali, H. D. (2006), Mathematical programming models and algorithms for a class-faculty assignment problem, European Journal of Operational Research, 173(2), 488-507.

Carter, M. W. e Tovey, C. A. (1992), When is the classroom assignment problem hard?, Operations Research, 40(1), S28-S39.

Constantino, A. A., Marcondes Filho, W. e Landa-Silva, D. (2010), Iterated heuristic algorithms for the classroom assignment problem, Proceedings of the 8th International Conference on the Practice and Theory of Automated Timetabling-PATAT, 152-166.

Dyer, J. S. e Mulvey, J. M. (1976), An integrated optimization/information system for academic departmental planning, Management Science INFORMS, 22(12), 1332-1341.

Fernandes, C., Caldeira, J. P., Melicio, F. e Rosa, A. (1999), High school weekly timetabling by evolutionary algorithms, Proceedings of the 1999 ACM symposium on Applied computing (SAC '99), 344-350.

Harik, G. R., Lobo, F. G. e Goldberg, D. E. (1999), The compact genetic algorithm, IEEE Transactions on Evolutionary Computation, 3(4), 287-297.

Holland, J. H., Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence, U Michigan Press, Oxford, England, 1975.

Krikpa, R. M. L. e Kripka, M. (2010), Simulated annealing aplicado na otimização da alocação de salas em instituição de ensino superior, Mecânica Computacional, XXIX, 9317-9325.

Lopes, M. C. e Schoeffel, P. (2002), Um método de alocação para o problema de reservas de sala de aula, II Congresso Brasileiro de Computação.

Martinez-Alfaro, H. e Flores-Teran, G. (1998), Solving the classroom assignment problem with simulated annealing, IEEE International Conference on Systems, Man, and Cybernetics, 4, 3703-3708.

Mooney, E. L. (1995), Large scale classroom scheduling, IIE transactions, 28(5), 369-378.

Mulvey, J. M. (1982), A classroom/time assignment model, European Journal of Operational Research, 9, 64-70.

Nascimento, A. S., Sampaio, R. M. e Alvarenga, G. B. (2005) Uma aplicação de simulated annealing para o problema de alocação de salas, INFOCOMP Journal, 59-66.

Prado, A. S. e Souza, S. R. (2014), Problema de alocação de salas em cursos universitários: um estudo de caso, Anais do XLVI Simpósio Brasileiro de Pesquisa Operacional, 2054-2065.

Silva, D. J. e Silva, G. C. (2009), Heurísticas baseadas no algoritmo de coloração de grafos para o problema de alocação de salas em uma instituição de ensino superior, Anais do XLII Simpósio Brasileiro de Pesquisa Operacional, 2839-2849.

Souza, M. J. F., Martins, A. X. e Araújo, C. R. (2002), Experiências com simulated annealing e busca tabu na resolução do problema de alocação de salas, Anais do XXXIV Simpósio Brasileiro de Pesquisa Operacional, 34, 1100-1110.

Souza, M. J. F., Maculan, N. e Ochi, L. S. (2004), A grasp-tabu search algorithm for solving school timetabling problems, Metaheuristics: Computer decision-making, 659-672.

Subramanian, A., Medeiros, J. M. F., Formiga, L. A. e Souza, M. J. F. (2011), Aplicação da metaheurística busca tabu ao problema de alocação de aulas a salas em uma instituição universitária, Revista Produção Online, 11(1), 54-75.

Waterer, H. A. (1995), Zero-one integer programming model for room assignment at the university of auckland, Proceedings of the 1995 ORSNZ Conference.