HEURÍSTICAS BASEADAS NO ALGORITMO DE COLORAÇÃO DE GRAFOS PARA O PROBLEMA DE ALOCAÇÃO DE SALAS EM UMA INSTITUIÇÃO DE ENSINO SUPERIOR

Douglas José da Silva
Geiza Cristina da Silva
Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas – Rua 37 nº. 115 – Loanda
35930-970 João Monlevade – Minas Gerais – Brasil
douglasjsilva@yahoo.com.br
geiza@decea.ufop.br

RESUMO

O problema de alocação de salas consiste em, a partir de horários de aulas pré-estabelecidos para cada turma, determinar a qual sala deve ser alocada uma turma em um dado horário. Este problema é classificado como NP-Dificil e, na busca por soluções, algoritmos heurísticos e baseados em metaheurísticas podem ser encontrados na literatura. Este tipo de metodologia não garante uma solução ótima, mas que, a um custo computacional razoável, soluções de boa qualidade sejam obtidas. Neste trabalho é proposta uma heurística baseada em um algoritmo de coloração de grafos para a geração de soluções iniciais que posteriormente são refinadas através dos métodos de busca local Primeiro de Melhora e Melhor Vizinho. A eficiência das heurísticas propostas é avaliada comparando-se os resultados obtidos com os reportados na literatura.

PALAVRAS CHAVE. Problema de programação de horários. Problema de Cursos Universitários. Coloração em grafos. Teoria e Algoritmos em Grafos (TAG). PO na Educação (EDU).

ABSTRACT

The classroom assignment problem consists of scheduling a set of class into a fixed number of rooms, given a fixed timetable. This scheduling problem is NP-Hard and many heuristic algorithms have been proposed to solve it. This methodology does not guarantee an optimal solution, but the computational cost reasonable, good quality solutions are obtained. In this paper we propose a heuristic based on a graph coloring algorithm for finds good initial solutions which are refined through two local search methods. The efficiency of the proposed heuristics is evaluated by comparing the results with the literature.

KEYWORDS. Timetabling. Course timetabling. Graph Coloring.

1. Introdução

Os problemas de programação de horários (*timetabling*) (PPH) vêem sendo estudados por diversos pesquisadores e muito tem sido publicado sobre assunto (Schaefer, 1999). Por se tratar de um assunto muito difundido na literatura, um grande número de variações do *timetabling* vem sendo proposto. Schaefer (1999) propõe três classificações para o problema de programação de horários:

- *School timetabling*: Nesta classificação se encontram os problemas da programação semanal de horários entre professor/turma em escolas/colégios. Aqui são consideradas que as disciplinas são fixas para cada turma e o objetivo principal é evitar que um professor esteja alocado a duas turmas simultaneamente ou que duas turmas tenham aulas com um determinado professor em um mesmo horário.
- *Course timetabling*: Tem em vista a programação semanal de horários de todas as disciplinas para todos os períodos dos cursos universitários determinando a relação (professor/turma/horário). Diferencia-se do *school timetabling*, pois nesta classificação os estudantes podem escolher as matérias (eletivas) que desejam se matricular. O objetivo também é o de minimizar a sobreposição de quaisquer das variáveis envolvidas.
- *Examination timetabling*: Trata a programação de exames para cursos universitários, evitando sobreposições de exames de disciplinas que possuem estudantes em comum e distanciando as datas dos exames dos estudantes o máximo possível.

O problema de alocação de salas (PAS) é parte integrante do Problema de Cursos Universitários (*course timetabling*) (Souza, 2007). O PAS considera que existem horários pré-estabelecidos de início e término das aulas e um conjunto de salas onde ocorrerão as aulas. O problema consiste em determinar uma alocação das turmas em salas de aulas de forma que requisitos considerados essenciais sejam respeitados, fazendo com que a solução seja viável. Além disso, requisitos não essenciais, mas desejáveis, devem ser obedecidos ao máximo de forma a melhorar a qualidade da solução.

Por se tratar de um problema altamente combinatório, e com o aumento de turmas nas instituições de ensino, a resolução manual desse problema pode demandar muito tempo e esforço por parte dos responsáveis e, mesmo feito assim, não se tem a garantia de que seja obtida uma solução que otimize a utilização do espaço (Silva, 2005).

O presente trabalho tem por finalidade propor soluções para o PAS através de uma heurística baseada em um algoritmo de coloração de grafos para a geração de soluções iniciais e, em seguida, as soluções são refinadas. São propostas duas versões independentes de heurísticas para o PAS através da variação do algoritmo de busca local utilizado para refinar as soluções iniciais: a primeira versão utiliza a estratégia conhecida por Primeiro de Melhora e, a segunda, a do Melhor Vizinho.

O restante deste texto está organizando da seguinte forma. A Seção 2 apresenta o referencial teórico. Na Seção 3 é apresentada a metodologia para obtenção de soluções para o problema abordado. A Seção 4 apresenta e discute os resultados alcançados, e finalmente, a Seção 5 traz conclusões e sugestões para trabalhos futuros.

2. O Problema de Alocação de Salas

O PAS é classificado como um problema NP-Difícil (Even e Shamir, 1976), ou seja, trata-se de um problema para o qual ainda não é conhecido um algoritmo que o resolva em tempo satisfatório, dificultando com isso a utilização de métodos exatos. Este fato, justifica o emprego de técnicas heurísticas e metaheurísticas.

Em Souza et al. (2002b) é apresentada uma proposta para a alocação de salas do Instituto de Ciências Exatas e Biológicas da Universidade Federal de Ouro Preto (ICEB-UFOP). O método VNS (*Variable Neighborhood Search*) somado a dois métodos de busca local é aplicado ao problema. A primeira combinação é formada pela associação do algoritmo VNS com o VND (*Variable Neighborhood Descent*) e a outra, do VNS com o método de busca local Melhor Vizinho. A partir da solução inicial gerada por um procedimento que segue as idéias da fase de

construção do método GRASP, foi aplicado o método VNS fazendo o uso de três vizinhanças diferentes: a primeira, gerada por realocação; a segunda, por troca; e a última, correspondendo a todos os vizinhos da solução inicial gerados por um movimento de troca ou realocação. Os autores concluem, após a demonstração dos resultados, que foi possível obter uma melhor solução com a utilização da primeira combinação.

Em Souza et al. (2002a) os autores propõem uma técnica híbrida com a utilização das metaheurísticas *simulated annealing* e busca tabu. Uma solução inicial é construída por um procedimento que segue também as idéias da fase de construção do método GRASP. A solução gerada é submetida ao método *simulated annealing* e a solução resultante é então refinada por uma busca local. Como resultado, o procedimento híbrido mostrou-se capaz de gerar soluções finais com menor desvio em relação às melhores soluções encontradas para cada instância. Além disso, a técnica foi adotada pelo ICEB para a confecção da alocação de salas de aula no primeiro semestre de 2002.

Em Silva (2005) foram consideradas três instâncias do PAS: a primeira, um caso fictício; a segunda, um caso simplificado da Universidade Federal de Lavras (UFLA) e a terceira, o caso do ICEB-UFOP. Na resolução do problema primeiramente foi usada uma heurística para a geração de uma solução inicial. Para a geração da vizinhança foram utilizados movimentos de troca que corresponde a trocar uma turma por outra desde que horários coincidam ou hajam horários vagos que permitam esse movimento e de realocação, onde ocorre a realocação de turmas para salas cujos horários estejam vagos. A partir da solução inicial obtida e considerando a estrutura de vizinhança definida foi utilizada a metaheurística *simulated annealing* para o refinamento da solução. Os testes feitos mostraram que os resultados obtidos constituem uma diminuição do desvio percentual quando comparado ao resultado obtido em (Souza et al, 2002a). No entanto, é reportado que o algoritmo proposto permite soluções inviáveis, uma vez que turmas podem ser alocadas em salas de aula que não suportam a sua capacidade. Por este motivo, este trabalho não é considerado em nossas comparações.

Em Oliveira (2006) é apresentada uma comparação de desempenho entre duas metaheurísticas para o PAS: um Algoritmo Genético e um *Simulated Annealing*. Estes métodos foram utilizados para a resolução de uma instância fictícia. Na modelagem do problema, além dos requisitos essenciais e não essenciais (Seção 2), o autor adiciona mais duas variáveis que são: a distância percorrida pelo aluno para ir de uma sala a outra e a alocação de aulas de conteúdo prático. Uma análise comparativa demonstrou que a heurística baseada em *Simulated Annealing* gerou melhores resultados do que o Algoritmo Genético, para a instância testada.

Pode-se ainda relacionar alguns trabalhos que abordam o problema de programação de horários utilizando por base para a solução a modelagem em grafos, mais especificamente, obtendo soluções a partir de algoritmos de coloração de grafos, método também utilizado neste trabalho.

Em Silva et al. (2006) é apresentada uma solução ao problema de horários do Curso de Engenharia de Produção da Poli/UFRJ. Neste trabalho, os autores fazem uso de algoritmos baseados em métodos aplicados ao problema de coloração de grafos para obter soluções para o problema.

Outra abordagem no mesmo sentido pode ser observada em Pereira et al. (2007). Neste trabalho, é proposta uma solução para o problema de alocação de horários na escola Estadual de Ensino Fundamental e Médio Ecoporanga (EEEFME) e é feita uma comparação entre dois métodos propostos: GRASP e GRASP Modificado. Este último, consiste de uma combinação do método GRASP e o método de coloração de arestas que participa da fase de construção do GRASP e tem como objetivo obter melhores soluções iniciais, forçando para que o máximo de restrições desejáveis seja atendido.

3. Metodologia

O problema aqui abordado considera que existem horários pré-estabelecidos de início e término das aulas e um conjunto de salas de aulas onde as turmas devem ser alocadas. Desta maneira, a confecção dos horários de aulas não faz parte do escopo deste trabalho.

Uma solução para o problema consiste em determinar uma alocação das turmas em salas de aulas considerando como requisitos essenciais que:

- uma sala deve ser destinada à apenas uma turma em determinado horário;
- uma turma deve ser alocada em apenas uma sala de aula em determinado horário;
- a sala de aula alocada a determinada turma deve ser capaz de comportar todos os seus estudantes.

Além disso, podem-se levar em consideração alguns requisitos não essenciais, mas que são desejáveis, como:

- utilizar o espaço das salas, de modo que turmas grandes fiquem em salas maiores e turmas menores em salas de menor capacidade;
- em algumas salas, como as que possuem equipamentos multimídias, preservar as restrições de uso e evitar sua utilização desnecessária sempre que possível;
- minimizar o deslocamento das turmas entre salas diferentes, diminuindo a troca de salas durante a mudança de horário.

3.1. Modelagem do problema

O modelo proposto considera as seguintes definições:

- T (i =1, ..., t), o conjunto que corresponde às turmas existentes no Instituto;
- n_i, número de horários de aulas de uma turma i;
- Hi (j = 1, ..., n_i), o conjunto de horários de aulas de uma turma i (i =1,..., t). Aqui temos t conjuntos distintos: H_1, ... H_t;
- $G = (V, X)$, o grafo (não orientado), onde V é o conjunto de vértices e X é o conjunto de arestas;

A cada vértice de V é associada uma dupla (T_i, Hi_j) (turma, horário). O número de vértices do grafo é dado por: $\sum_{i=1}^{t}\sum_{j=1}^{n_i} H_{ij}$.

Uma aresta é associada a um par de vértices u e v se e somente se tivermos uma coincidência de horário de aula de duas turmas u e v, ou seja, se $Hu_i = Hv_j$, garantindo que os dois primeiros requisitos essenciais sejam atendidos.

A Figura 1 é um exemplo da definição aplicada a três turmas e três horários de aulas.

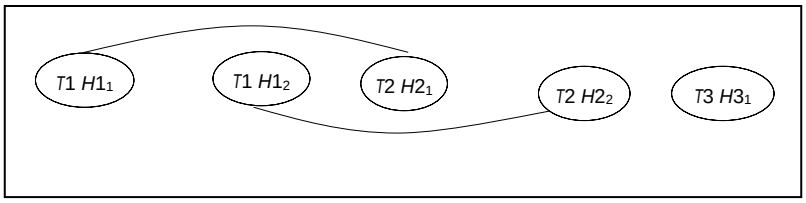

Figura 1 - Grafo associado ao exemplo da definição do modelo

3.2. Representação da Solução

O grafo é representado por meio de uma matriz de adjacências de dimensões $|V| \times |V|$, na qual a existência de um algarismo não nulo representa a adjacência entre dois vértices, que corresponde à coincidência de dois horários, como pode ser exemplificado na Tabela 1, de acordo com o grafo apresentado na Figura 1.

	T1 H11	T1 H12	T2 H21	T2 H22	T2 H23
T1 H1l	0	0	1	0	0
T1 H12	0	0	0	1	0
T2 H21	1	0	0	0	0
T2 H22	0	1	0	0	0
T2 H23	0	0	0	0	0

Tabela 1- Matriz de adjacências gerada através do grafo

3.3. Algoritmo para geração de solução inicial proposto

Neste trabalho, uma solução inicial é gerada com base na heurística proposta em Silva et al. (2006) que consiste de uma aplicação do algoritmo de coloração por classes (Boaventura-Netto, 2006).

Como dito anteriormente, a modelagem proposta para o problema faz com que a com os dois primeiros requisitos essenciais sejam obedecidos. Para que a terceira restrição seja atendida é adicionada ao algoritmo a verificação de capacidade das salas e de demanda de alunos de cada turma.

A partir do grafo representado por sua matriz de adjacências, o algoritmo constitui-se dos seguintes passos que são repetidos enquanto ainda há vértices não coloridos no grafo (turmas não alocadas à salas de aulas):

Passo 1. É aberta uma classe cor (sala de aula);
Passo 2. São introduzidos na classe cor, dentre os vértices não coloridos, aqueles que atendam às restrições do problema;
Passo 3. cor é incrementada.

A Figura 2 mostra o pseudocódigo do método descrito acima. São parâmetros de entrada: o grafo G e um vértice s, o primeiro a ser colorido. Y representa o conjunto de vértices não coloridos e DEMANDA é um vetor que armazena o número de alunos de cada turma.

```
Algoritmo Col_Classe_PAS (G (V, A), s)
    cor ← 1;
    Classe[cor] ← s;
    Y ← V – {s};
    Enquanto Y ≠ ∅ faça
        Para todo v | v ∈ V e v ∈ Y faça
            Se Classe[cor] não possui adjacente a v e capacidade de Classe[cor] >=
            DEMANDA (v) então
                Classe[cor] ← v;
                Y ← Y - {v};
            Fim-se
        Fim-Para
        Cor ← Cor + 1;
    Fim-Enquanto
    Retorna Classe;
Fim-Algoritmo.
```

Figura 2 - Pseudocódigo do algoritmo coloração por classe para o PAS com a adição da restrição de capacidade

A cada execução deste algoritmo, dependendo do vértice inicial s dado por parâmetro, podemos obter uma coloração diferente. Ao final de uma execução, o algoritmo retorna a coloração obtida na qual cada dupla (turma, horário) estará destinada a uma sala de aula (*cor* de *Classe*).

Devemos ainda ressaltar que a heurística tende a otimizar o número de salas utilizadas, uma vez que busca alocar o máximo de turmas em uma mesma sala de aula antes de iniciar a alocação em outra.

A Figura 3 exemplifica o funcionamento do algoritmo para o grafo apresentado na Figura 1. Para este grafo, com suas respectivas adjacências, a heurística proposta *Col_Classe_PAS* é capaz de obter uma coloração com duas cores: 1 e 2.

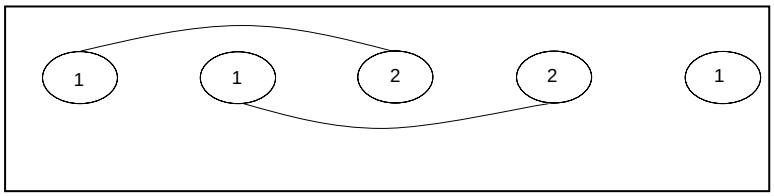

Figura 3 - Grafo do exemplo colorido

Interpretando o resultado obtido chega-se à seguinte alocação de salas, demonstrada na Tabela 2.

SALA 01	$T1\ H11_1$	SALA 02	$T2\ H21_1$
	$T1\ H11_2$		$T2\ H21_2$
	$T2\ H21_3$		-

Tabela 2 - Alocação das salas de aula

3.4. Função Avaliação

O problema de alocação de salas é um problema de decisão multicritério pois, para determinar a qualidade de uma alocação, é necessário considerar diferentes objetivos (Souza et al., 2002a).

Na avaliação da solução foram levadas em consideração as duas classes de requisitos, já mencionadas anteriormente, a dos requisitos essenciais que são aqueles que garantem a viabilidade da solução e a dos requisitos desejáveis, cuja função é melhorar a qualidade da solução mas que, no entanto, seu não atendimento não inviabiliza uma alocação.

Utilizando a definição apresentada em Souza et al. (2002a) temos que uma solução s é avaliada através de uma função $f(s)$, que contém duas componentes: a primeira é a componente de viabilidade ($g(s)$), que mede o não atendimento dos requisitos essenciais e, a segunda, é a componente de qualidade ($h(s)$) que tem como propósito avaliar o não atendimento dos requisitos desejáveis. Logo, para que seja encontrada uma solução satisfatória deve se buscar a minimização $f(s)$, descrita na Equação 1.

$$f(s) = g(s) + h(s) \tag{1}$$

A Equação 2 mede o nível de inviabilidade $g(s)$ da função objetivo $f(s)$. Deve-se ressaltar que uma solução só é considerada viável se esta componente for nula.

$$g(s) = \sum_{k=1}^{K} \alpha_k I_k \tag{2}$$

Onde:

- K: número de medidas de inviabilidade;
- I_k: valor da k-ésima medida de inviabilidade;
- α_k: peso associado à k-ésima medida de inviabilidade;

A componente $h(s)$, que demonstra a qualidade de uma solução s, é formulada como:

$$h(s) = \sum_{l=1}^{L} \beta_l Q_l \tag{3}$$

Onde:

- L: número de medidas de qualidade;
- Q_l: valor da l-ésima medida de qualidade;
- β_l: peso associado à l-ésima medida de qualidade.

Os pesos dados às diversas medidas da função $f(s)$ refletem a importância relativa de cada uma delas e, por isso, os pesos que representam a componente de viabilidade devem ser muito maiores do que os que representam a componente de qualidade, de modo a privilegiar a escolha de soluções viáveis, $\alpha_k >> \beta_l \ \ \forall k, l$.

3.5. Estrutura de vizinhança

Dada uma solução s, para atingir uma solução s', onde s' é dito vizinho de s, foi usado o movimento de troca proposto em Souza et al. (2002a). Este movimento consiste em trocar as turmas de sala de modo a obter uma alocação mais eficiente, atendendo as restrições desejáveis e minimizando assim o valor da função objetivo. A Figura 4 ilustra o movimento de troca.

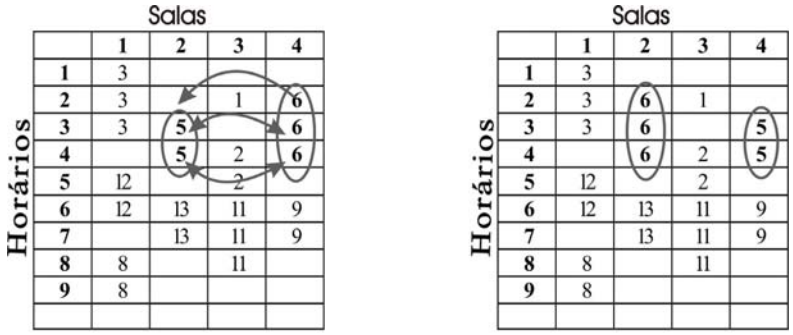

Figura 4 - Movimento de Troca. Fonte: Martins et. al (2002)

3.6. Heurísticas de refinamento

Dois métodos de busca local utilizados para percorrer a vizinhança de uma solução são consideradas neste trabalho: o Primeiro de Melhora e o do Melhor Vizinho, descritos nas seções seguintes.

3.6.1. Método Primeiro de melhora

O método Primeiro de Melhora explora a vizinhança da solução até que um vizinho melhor seja encontrado. A solução obtida pela execução deste movimento é considerada a solução corrente para a próxima iteração do algoritmo. O método é encerrado quando, em uma iteração, toda vizinhança é investigada e nenhum vizinho é capaz de melhorar a solução corrente.

A vantagem da utilização deste método é que, só no pior dos casos toda a vizinhança é explorada, no entanto ele pode ficar preso no primeiro ótimo local encontrado (Souza, 2007).

3.6.2. Método do Melhor Vizinho

O método do Melhor Vizinho parte de uma solução inicial, analisando entre os possíveis vizinhos o que apresenta a maior melhoria na solução corrente. Este movimento é realizado e a solução obtida é considerada corrente para a próxima iteração. O método itera desta maneira até que, a partir de uma solução corrente, nenhuma melhora seja alcançada. Neste momento, chegamos a um ótimo local.

3.7. Heurísticas propostas para o PAS

O fluxograma apresentado na Figura 5 ilustra o funcionamento dos algoritmos propostos. Temos, como já dissemos antes, de acordo com a estratégia de busca local escolhida, duas versões independentes desta heurística.

Inicialmente ocorre a leitura dos dados da instância. Em seguida, o grafo é gerado. A partir

deste ponto a estrutura de repetição inicia fazendo com que os próximos passos sejam executados tantas vezes quanto for o valor de *n*, um parâmetro de entrada. Em nossa implementação, o valor de *n* é o número de vértices do grafo e, a cada execução, o vértice inicial *s* é um vértice do grafo que varia de 1 a *n*.

O *loop* gera uma solução inicial a partir do vértice inicial *s*. Esta solução é refinada por um dos algoritmos de busca local implementados. A melhor solução é atualizada a cada vez que uma solução de melhor custo que a corrente é encontrada.

Doravante, esta heurística quando utilizada com a busca Primeiro de Melhora é denotada por *Col_PAS + Primeiro de Melhora* e, quando utlizada com a busca local Melhor Vizinho por *Col_PAS + Melhor Vizinho*.

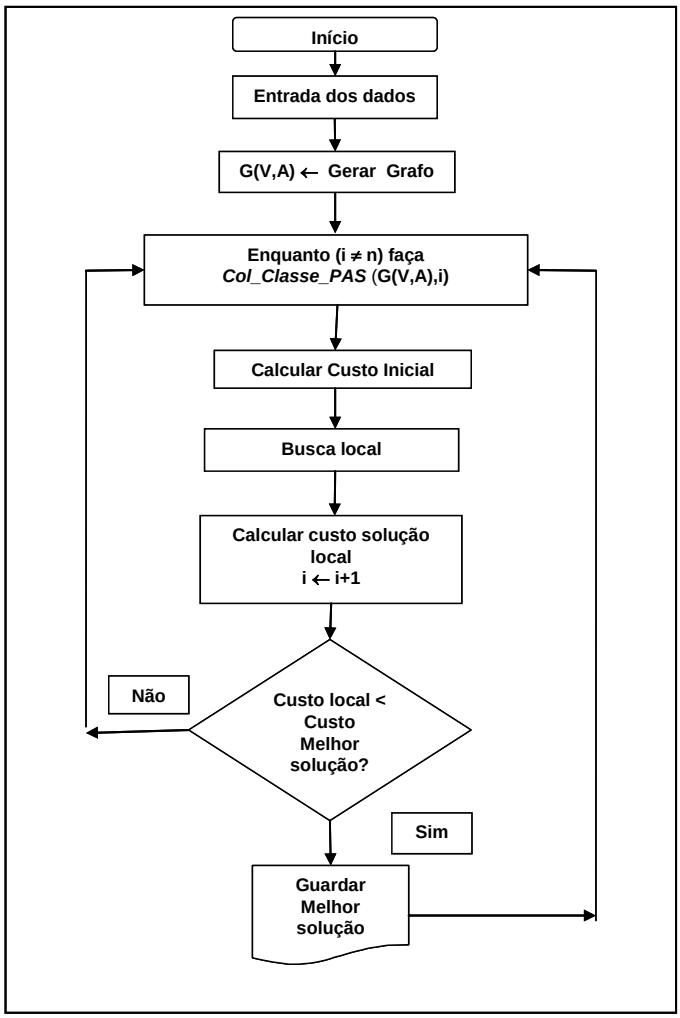

Figura 5 - Fluxograma do algoritmo proposto

4. Resultados computacionais

O algoritmo proposto tem sua implementação feita na Linguagem C e o código fonte gerado foi compilado utilizando o compilador Bloodshed Dev-C++. Os testes foram realizados em um computador Intel Celeron 2.13 GHz, com 2 GB de RAM sob o sistema operacional Windows XP.

As instâncias testadas, propostas em Martins et al. (2002a), são descritas na Tabela 3. A primeira consiste de dados reais referentes à distribuição de salas do segundo semestre letivo do ano 2001 e as outras duas são instâncias geradas artificialmente.

Instância	Número de salas	Número de turmas	Número de horas/aula
Real ICEB	20	233	763
Teste17	17	214	713
Teste22	22	281	938

Tabela 1 – Características das instâncias

A Tabela 4 apresenta uma comparação entre os resultados obtidos pelas versões de algoritmos propostos e da literatura. Os melhores resultados presentes na literatura para este problema pertencem a Souza et al. (2002a) no qual os autores propõem implementações das metaheurísticas *Simulated Annealing* e Busca Tabu e outro algoritmo híbrido constituídos pelas duas.

	Algoritmos	**Melhor Solução**
Instância Teste 17	*Simulated Annealing*	7320
	Busca Tabu	7510
	Simulated Annealing + Busca Tabu	7144
	Col_PAS + Primeiro de Melhora	6042
	Col_PAS + Melhor Vizinho	**5849**
Instância Real ICEB	**Algoritmos**	**Melhor Solução**
	Simulated Annealing	10600
	Busca Tabu	10191
	Simulated Annealing + Busca Tabu	9208
	Col_PAS + Primeiro de Melhora	8268
	Col_PAS + Melhor Vizinho	**7739**
Instância Teste 22	**Algoritmos**	**Melhor Solução**
	Simulated Annealing	26329
	Busca Tabu	28677
	Simulated Annealing + Busca Tabu	25212
	Col_PAS + Primeiro de Melhora	13554
	Col_PAS + Melhor Vizinho	**10990**

Tabela 4 – Comparação entre os resultados da literatura e dos métodos propostos

A Tabela 5 mostra uma melhoria de custo da ordem de 18,1%, 16,0%, 56,4% em relação ao melhor algoritmo da literatura para as instâncias Teste 17, Real ICEB e Teste22 quando é utilizada a heurística *Col_PAS + Melhor Vizinho*. A melhoria de custos é de 15,4%, 10,2% e 46,2%, diante da heurística *Col_PAS + Primeiro de Melhora*, respectivamente.

Algoritmo	% de Melhora Teste 17	% de Melhora Real ICEB	% de Melhora Teste 22
Col_PAS + Primeiro de Melhora	15,4%	10,2%	46,2%
Col_PAS + Melhor Vizinho	18,1%	16,0%	56,4%

Tabela 5 – Melhoria dos resultados

A Tabela 6 apresenta um resumo dos resultados apurados com os testes feitos com os algoritmos propostos. A coluna *Solução Inicial* mostra o custo da solução inicial gerada pelo algoritmo Col_Classe_PAS e a coluna *Melhor Solução* apresenta o custo da solução após a busca local. A coluna *Custo Médio* apresenta a média dos custos obtidos nas execuções independentes das heurísticas. Os tempos reportados são em segundos.

Instância	Número de Execuções	Solução Inicial	Melhor Solução	Custo Médio	Tempo de CPU melhor solução	Tempo médio de CPU	Número de Salas Utilizadas
Teste 17	713	12851	6042	7344	55,0	59,8	17
		13186	5849	6801	67,0	73,7	17
Real ICEB	763	13934	8268	8682	64,0	66,4	20
		13869	7739	7785	79,0	82,0	20
Teste 22	938	24118	13554	18092	132,0	127,9	25
		26438	10990	14073	164,0	163,7	25

Tabela 6 – Resultados obtidos pelos métodos propostos

Comparando-se os resultados obtidos com o uso dos dois algoritmos de busca local implementados, podemos ver que o Método do Melhor Vizinho apresenta melhores soluções para o conjunto de instâncias testadas. Em média foi capaz de obter soluções de custos 11,8% menores que o Primeiro de Melhora. Verifica-se, no entanto, que o método Primeiro de Melhora necessita um menor tempo de computação, sendo em média 16,2% mais rápido.

Analisando o custo médio das soluções, percebemos que ao longo de sua execução o algoritmo gera soluções viáveis e de boa qualidade.

5. Conclusões e Sugestões de Trabalhos Futuros

O objetivo deste estudo foi o de apresentar uma heurística baseada no algoritmo de coloração de grafos a fim de propor soluções viáveis para problema de alocação de salas. Variando-se o algoritmo de busca local chegamos a duas combinações de métodos diferentes.

Pode-se constatar que os algoritmos se mostram promissores para gerar soluções iniciais viáveis de menores custos que a literatura para o problema.

Como já era esperado, o Método do Melhor Vizinho foi capaz de produzir soluções de custos médios menores do que o Primeiro de Melhora. No entanto, este último se mostrou um algoritmo de tempo computacional mais baixo. Desta forma, deixamos neste trabalho, a possibilidade de escolha entre um algoritmo com menor tempo de resposta e outro onde a solução é de melhor qualidade.

Como sugestão para trabalhos futuros propõe-se que o algoritmo para geração de soluções iniciais aqui proposto seja utilizado juntamente com metaheurísticas como GRASP, Busca Tabu ou *Simulated Annealing*, de forma a se estudar o efeito destas sobre o método.

Agradecimentos

Os autores são gratos ao CNPq/(CT-Info e Universal) e FAPEMIG pelo apoio a este trabalho.

Referências Bibliográficas

Boaventura Netto, P.O., *Grafos: Teoria, modelos, algoritmos*. Ed. Blucher, São Paulo, 2006.

Even, S., Itai, A. and Shamir, A. (1976), On the complexity of timetabling and multicommodity flow problems, *SIAM Journal of Computation*, 5, 691-703.

Oliveira, A.C., Uso do Algoritmo Genético e Recozimento Simulado para o problema de alocação de salas. *Monografia*, Universidade Federal de Lavras, Departamento de Ciência da Computação, 2006.

Pereira, R.S, Boaventura Netto, P.O e Lacruz. A.J. (2007) O método GRASP aplicado a um problema de coloração: estudo de caso em uma instituição de ensino fundamental e médio**.** In: Simpósio de Pesquisa Operacional e Logística da Marinha, *Anais do SPOLM*,.

Schaefer, A. (1999), A survey of automated timetabling, *Artificial Intelligence Review*, 13, 87-127.

Silva, A.S.N. Estudo e Implementação, Mediante Recozimento Simulado, do Problema de Alocação de Salas. *Monografia*, Universidade Federal de Lavras, Departamento de Ciência da Computação, 2005.

Silva, G. C., Pereira, R., Boaventura Netto, P.O., Jurkiewicz, S. e Meirelles, L. A. (2006), Programação de horários com reservas no curso de graduação em Engenharia de Produção da UFRJ, *Anais do XXXVIII SBPO*, 409-419.

Souza, M.J.F. Notas de aula da disciplina Inteligência Computacional para Otimização, Notas de aulas, Universidade Federal de Ouro Preto, 2007.

Souza, M.J.F., Martins, A.X. e Araújo, C.R. (2002a), Experiências com a utilização de Simulated Annealing e Busca Tabu na resolução do Problema de Alocação de Salas. In: XXXIV Simpósio Brasileiro de Pesquisa Operacional - SBPO, Instituto Militar de Engenharia, Rio de Janeiro, Brasil. *Anais do XXXIV SBPO*, 1100-1110.

Souza, M.J.F., Martins, A.X., Araújo, C.R e Costa, F.W.A. (2002b), Métodos de Pesquisa em Vizinhança Variável aplicados ao Problema de Alocação de Salas. In: XXII Encontro Nacional de Engenharia de Produção, *Anais do ENEGEP 2002*, 1-8.