SIMULATED ANNEALING APLICADO AO PROBLEMA DE ALOCAÇÃO DE SALAS DE AULA

Hanniere de Faria Veloso dos Santos, Dinesh Atul Rodrigues Trivedi
Instituto de Ciência e Tecnologia
Universidade Federal de São Paulo (UNIFESP)
Rua Talim, 330, São José dos Campos, São Paulo,
CEP 12231-280

São José dos Campos, SP

Resumo

Este artigo vem com a proposta de tratar do problema de alocação de salas, modelando-o de forma a utilizar *Simulated Annealing* (SA) para sua resolução. Este problema é considerado como NP-Difícil e por isso muitos métodos heurísticos, como: *Simulated Annealing*, Busca Tabu, Algoritmo Genético, tem sido propostos para resolve-lo. Sendo assim, serão abordadas apenas algumas restrições do problema que possibilite a geração de resultados de qualidade. O objetivo é otimizar a utilização das salas de aula de uma instituição de acordo suas capacidades bem como eliminar inviabilidades, no caso, a alocação de turmas a salas cujas capacidades não a comportem. Este trabalho foi implementado utilizando C++ ANSI juntamente com a ferramenta Qt Creator 2.7.0 32 bit.

Palavras-Chave: Problema de Programação de Horários, Problema de Alocação de Sala, Simulated Annealing.

Abstract

This research project is aiming to address the Classroom Assignment Problem, modeling it as a graph that enable to apply an algorithm for graph coloring obtaining a feasible solution. This problem is considered NP-Hard, and so many heuristic methods, such as Simulated Annealing, Tabu Search, Genetic Algorithm has been proposed to solve it. Therefore, will be address only a few constraints of the problem to allow the use of the coloring algorithm with quality results, being possible to cover more restrictions in order to use more concepts of graphs to solve the problem.

Keywords: Timetabling, Graph Theory, Classroom Assignment Problem.

Sumário

1 Introdução

Problemas de Programação de Horários (*Timetabling*) , também conhecidos como (PPH) vêem motivando diversas pesquisas na área de Otimização Combinatória e Inteligência Artificial (SCHAEFER, 1999). Uma vez que estes problemas são classificados como NP-Difícil e não se conhece um algoritmo que os resolvam em tempo satisfatório, suas resoluções vêem sendo obtidas por heurísticas e metaheurísticas que não oferecem soluções ótimas, mas que possuem boa qualidade.

Na literatura há uma divisão de Problemas de Programação de Horários (PPH) em: Programação de Horários em Escolas (PPHE), Problemas Programação de Horários de Cursos em Universidades (PPHU) e Problemas Programação de Horários de Cursos em Universidades (PPHU). Essa classificação foi proposta por Schaefer (1999) sendo a mais utilizada na literatura.

Para essa classe de problemas inicialmente são definidos os requisitos que são expressos em forma de restrições do problema a ser atacado, que normalmente podem ser de dois tipos:

- Restrições fortes ou essenciais: restrições que devem ser satisfeitas a qualquer custo, sendo que não é possível uma solução factível que não as satisfaça.
- Restrições fracas ou não essenciais: restrições que são desejáveis de serem satisfeitas afim de que a solução obtida tenha melhor qualidade.

Segundo Souza (2007), o Problema de Alocação de Salas (PAS) é uma derivação dos Problemas de Programação de Horários de Cursos em Universidades (PPHU), ou seja, é tido como resolvido o problema PPHU desconsiderando a alocação de salas o que é resolvido por meio do PAS. Sendo assim, este artigo tem o objetivo de obter soluções factíveis para o Problema de Alocação de Salas por meio de sua modelagem e posterior utilização da metaheurística *Simulated Annealing.* Com o intuito de validar o sistema implementado, será utilizado dados reais obtidos da própria Universidade Federal de São Paulo.

2 Descrição do problema

2.1 O problema de alocação de Salas

Conhecido na literatura inglesa como *Classroom Assignment Problem* e aqui chamado de Problema de Alocação de Salas (PAS), consiste em alocar aulas, com horários previamente conhecidos à um numero fixo de salas de forma que requisitos considerados essenciais sejam respeitados (CARTER; TOVEY, 1992).

Na literatura são observadas várias restrições essenciais e não-essenciais, sendo os mais comuns listados a seguir. Como restrições fortes, considera-se:

- Em uma mesma sala e horário não pode haver mais de uma aula;
- Uma sala não pode receber uma turma cuja quantidade de alunos seja superior à sua capacidade;
- Uma turma só pode ser alocada somente em uma sala de aula em um determinado horário.

Como restrições fracas mais comuns, têm-se:

- Algumas aulas precisam de recursos especiais, tais como : projetores, alto-falantes o que obriga alocá-las em salas que possuam esses recursos.
- Algumas salas possuem restrições de uso e a utilização delas deve ser evitada, por exemplo: auditórios, salas de defesa de tese e etc.
- Sempre que possível alocar alunos de um mesmo curso e período a uma mesma sala.
- Sempre que possível deixar diariamente um horário vazio em cada sala, para possibilitar sua limpeza.

O Problema de Alocação considera que existem horários pré-estabelecidos de inicio e término das aulas e suas respectivas demandas de alunos, bem como um conjunto de salas onde as turmas devem ser alocadas com suas respectivas capacidades. Sendo assim, neste artigo serão abordados os requisitos essenciais para a validação da modelagem aqui apresentada.

2.2 Modelagem do problema

A solução do problema é representada por uma matriz S_{mxn} com m representando o número de horários que as salas estão disponíveis para as aulas e n representando o numero de salas. Cada célula $S_{i,j}$ da matriz S representa um horário *i* disponível numa sala *j*, sendo que cada matriz representa um dia da semana. Sendo assim uma célula vazia indica que a sala *j* está desocupada no horário *i*.

Para exemplificar, é dada a seguinte tabela que possui seis salas (1, 2, 3, 4, 5,6) e sete horários (1, 2, 3, 4, 5, 6,7) e dez turmas.

Salas

Horários

	1	2	3	4	5	6
1						
2						
3						
4						
5						
6						
7						

Tabela 1 – Exemplo da modelagem

NÃO ESQUECER DE FALAR SOBRE A FUNCAO OBJETIVO

3 Metodologia

3.1 Simulated Annealing

Simulated Annealing é uma heurística de busca local que aceita movimentos que pioram o custo de forma a escapar de ótimos locais por meio de uma probabilidade. Este método foi proposto inicialmente por (Kirkpatrick et al., 1983), e tem como fundamento o principio da termodinâmica ao simular o resfriamento de um conjunto de átomos aquecidos.

Esta técnica inicia sua busca a partir de uma solução inicial qualquer, de forma que posteriormente em cada loop gera-se um vizinho de forma aleatória da solução corrente *s*.A cada vizinho gerado s' de s, testa-se o valor de Δ, ou seja, $\Delta = f(s') - f(s)$. Caso $\Delta < 0$, o método aceita a solução e o vizinho *s'*. *Caso* $\Delta \geq 0$, é calculado a probabilidade $e^{-\Delta/T}$ desse vizinho ser aceito ou não. T é um parâmetro do método, chamado de temperatura e regula a probabilidade de aceitação de soluções de pior custo.

Inicialmente a temperatura T assume um valor elevado T_0 . Após determinado número de iterações a temperatura é gradativamente reduzia por uma taxa de resfriamento α, ou seja, $T_n \leftarrow \alpha * T_{n-1}$ $(0 < \alpha < 1)$. Desta forma faz-se com que no inicio tenha-se uma chance de sair de mínimos locais e quando T aproxima-se de zero, o algoritmo comporta-se como o método de descida, dado que é diminuída a probabilidade de aceitar movimentos que pioram o custo.

O algoritmo para quando a temperatura chega a um valor próximo de zero e nenhuma solução de pior custo seja mais aceita.

Os parâmetros deste algoritmo são: a taxa de resfriamento α, o numero de iterações para cada temperatura (IT_{max}) e a temperatura inicial T_0.

```
        Algoritmo SA. (f(.), N(.), α,  ITmax, T0, s)
s* ← s;                          {Melhor solução obtida}
IterT ← 0;                       {Número de iterações na temperatura T}
T ← T0;                          {Temperatura corrente}
enquanto (T > 0) faça
   enquanto (IterT < ITmax ) faça
      IterT ← IterT + 1;
      Gere um vizinho qualquer s' ∈ N(s);
      Δ = f(s') – f(s);
      se (Δ < 0)
         então s ← s';
               se (f(s') < f(s*)) então s* ← s';
         senão Tome x ∈ [0,1];
               se (x < e-Δ/T) então s ← s';
      fim-se;
   fim-enquanto;
   T ← α * T;
   IterT ← 0;
fim-enquanto;
s ← s*;
Retorne s;
Fim S.A.:
```

Figura 1 – Algoritmo *Simulated Annealing*

4 Resultados

Referências

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

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, 2007.

M.W. CARTER; C.A. TOVEY. **When is the classroom assignment problem hard?** Operations Research Supplement 1, 40:2839, 1992.