Problema de Alocação de Salas de Aula

Relatório Técnico

Maio, 2010

Flávia Shizue Kohiyama
Gizeli Ribas de Morais

Unioeste

Sumário

Capítulo 1

Introdução

Diante da necessidade de resolver o problema de alocação de salas de aulas no Centro de Engenharias e Ciências Exatas (CECE) da Universidade Estadual do Oeste do Paraná (Unioeste) que encontra-se localizado no Parque Tecnológico Itaipu (PTI), visando facilitar e melhorar a realização deste trabalho, que atualmente é realizado de forma manual, diminuindo a ocorrência de conflitos de horários.

1.1 Objetivo

Desenvolver um sistema de alocação de salas de aula para auxiliar no escalonamento de horários das aulas do Centro de Engenharias e Ciências Exatas da Unioeste.

1.2 Justificativa

Atualmente a tecnologia se faz presente em todas as áreas de trabalho, através da evolução dos mecanismos e processos utilizados para a realização das atividades afins de cada área. Diante disso, a partir de uma análise dos processos realizados pela área administrativa da Unioeste relacionados ao escalonamento de horários e turmas para o período letivo, verificou-se a necessidade de automação deste processo. Visando minimizar a possibilidade de ocorrência de conflitos de horários e também melhorar o processo de alocação de salas de aula, substituindo o processo manual. Desta forma gerou-se uma solução baseada em estudos de Algoritmos de Busca. Neste caso, serão utilizados os algoritmos de Busca em Profundidade junto com a Estratégia CSP e também o Algoritmo de Busca Gulosa (Greedy Search). Ao final será feita uma verificação dos resultados obtidos a partir da execução dos algoritmos citados

anteriormente, comparando seus resultados. Inicialmente é esperado como resultado que o Algoritmo de Busca Gulosa obtenha melhor performance em comparação com os resultados obtidos pela execução do outro algoritmo.

Capítulo 2

Análise do Problema

2.1 Descrição do Problema

O problema de alocação de salas (PAS) é muito comum. Diversas universidades e outras instituições utilizam técnicas de alocação de salas e recursos para a obtenção de melhores resultados ao final do processo. Sendo assim o problema apresentado pelo Centro de Engenharias e Ciências Exatas (CECE) da Unioeste campus de Foz do Iguaçu, fará parte destas estatísticas. A solução apresentada neste relatório está relacionada a probabilidade de ocorrência de erros e ao excesso de esforço manual para realizar a alocação de salas de aulas do CECE, localizado nas dependências do PTI. O desenvolvimento da solução está baseado no uso das seguintes informações fornecidas:

- Série e Quantidade de alunos matriculados em cada série;
- Nome da disciplina e quantidade média de matriculados na disciplina;
- Identificação da sala e sua capacidade;
- Nome do professor da disciplina e a sua disponibilidade de horários;

2.2 Descrição da Solução

Após pesquisas realizadas uma solução encontrada para solucionar o problema, é a utilização do algoritmo Busca Gulosa. Foi escolhido o algoritmo de Busca em Profundidade com a estratégia CSP para fazer uma comparação entre o algoritmo escolhido para a solução e uma outra opção que não seria uma boa opção. Para realizar uma comparação com os resultados destes algoritmos serão analisados os resultados de saída de cada um dos algoritmos.

Capítulo 3

Desenvolvimento da Solução

3.1 Algoritmos

3.1.1 Estratégia CSP

Segundo o livro Programação Lógica por Restrições na Resolução de Problemas, "a Programação Lógica por Restrições (PLR) compõe um paradigma computacional que combina a natureza declarativa da programação em lógica com a eficiência dos métodos de resolução de problemas que recorrem a restrições. Esta tecnologia tem-se mostrado eficaz na resolução de problemas que são intratáveis quando se recorrem a outras técnicas. Entre esses problemas encontram-se diversas classes de problemas de natureza combinatória, tais como os problemas de escalonamento, de planejamento e de atribuição de recursos. É de notar que uma das vantagens mais significativas deste modelo passa por oferecer um menor tempo de desenvolvimento de programas e uma eficiência comparável à das linguagens imperativas". Esta é uma estratégia onde são impostas algumas condições às quais o software criado deve atender. Neste caso são restrições que controlam o acesso aos dados dos cursos, dos horários das disciplinas, do turno de oferecimento da disciplina, entre outras. As restrições utilizadas para o desenvolvimento dos códigos deste trabalho estão descritas na seção referente à descrição do problema a ser resolvido.

3.1.2 Busca em Profundidade

Este tipo de busca inicia a partir de um nó localizado na raiz da árvore de dados, onde então suas fronteiras são analisadas e é feito a escolha pelo nó mais a direita até chegar em um nó raiz. A partir deste ponto caso a solução ainda não tenha sido encontrada, retorna-se um nível na árvore e uma nova escolha é realizada, de forma a busca a solução mais apropriada

para a resolução do problema apresentado. A técnica de retornar um nível na árvore recebe o nome de backtracking, que será explicada mais adiante com mais detalhes. O uso desta estratégia de busca permite sempre encontrar uma solução para o problema, portanto é considerado completa, porém nem sempre esta solução encontrada é considerada uma solução ótima, ou seja, a melhor solução possível.

3.1.3 Busca em Profundidade com Backtraking em conjunto com a Utilização da Estratégia CSP

Esta técnica utiliza o retorno ao nó anterior e neste caso está sendo utilizada em conjunto com a estratégia de satisfação por restrições. Segundo , [?] backtracking é um método para iterar (percorrer) todas as possíveis configurações em um espaço qualquer. É um algoritmo geral que poderá ser personalizado para cada tipo de aplicação.

3.1.4 Busca Gulosa (Greedy)

Esta estratégia de busca, também conhecida por Greedy Search considera apenas o fator heurístico h(n). Armazena o caminho que está percorrendo até encontrar uma solução aceitável, porém não armazena o quanto ele caminhou na árvore de dados. Não é considerado completo e nem ótimo, pois pode parar em um nó indesejável . Apresenta risco de entrar em loop infinito caso não faça tratamento de nós repetidos.

3.2 Metodologia Utilizada

Para atingir o resultado do trabalho de forma organizada utilizou-se o padrão de desenvolvimento de código MVC (Model View Control), onde o objetivo é separar o código -fonte em camadas onde respectivamente, na camada Model estão localizados os dados ou a lógica de negócio, na camada View está a interface com o usuário e a camada onde é descrito o fluxo de toda a aplicação chama-se Control . O objetivo de fazer uso desta técnica visa permitir que a lógica de negócio possa ser visualizada e acessada por diversas interfaces, como, por exemplo, celular, PDA, outros computadores conectados na rede entre outras. É também utilizada em aplicações interativas que necessitam de interfaces flexíveis. Por isso dá-se o nome de modelo a camada de negócio, pois não se pode ter acesso à quantidade e nem quais são as interfaces que a estão acessando. Para a resolução do problema apresentado, a geração automática de horários em salas de aula, foram utilizados dois algoritmos

para realizar esta tarefa, o Busca em Profundidade com o uso da técnica de CSP e o algoritmo de Busca Gulosa, também conhecido como Greedy.

3.2.1 Informações Técnicas

Os testes seriam realizados no computador que oferece as seguintes configurações:

- Sistema Operacional de 32 bits, Windows Vista Home Premium com Service Pack 2;
- Memória RAM de 3 Gb;
- Processador Intel Core 2 Duo T5450, 1,67 GHz;

Quanto a estratégia que seria utilizada para os testes, utilizariamos a comparação entre os resultados gerados pelos algoritmos desenvolvidos de forma a verificar se obtiveram o resultado esperado de acordo com os dados de entrada. Seriam utilizados diversos arquivos com diferentes quantidades de dados.

3.2.2 Comparações

Devido a dificuldades encontradas durante o desenvolvimento do trabalho não foi possível realizar testes, pois tivemos problemas com a geração da interface gráfica, porém ambos os algoritmos apresentam resultados após a sua execução.

Capítulo 4

Conclusão

Após a realização deste trabalho com todas as dificuldades encontradas durante o percurso de desenvolvimento, chegou-se à seria de grande importância para a realização da terefa de distribuição de horários dentre as salas de aulas, a execução de um software como era a intenção inicial de desenvolvimento.

=

Referências Bibliográficas

[1] Ana Estela Antunes da Silva. Tópico agentes de buscas - buscas cegas. http://www.unimep.br/ aeasilva/topicoBCEGA.pdf.

[2] Rodrigo Rebouças de Almeida. Model-view-controller (mvc). http://www.dsc.ufcg.edu.br/ jacques/cursos/map/html/arqu/mv.

[3] Josiane M. Pinheiro Ferreira. Problemas de satisfação de restrições. http://www.din.uem.br/ jmpinhei/SI/07csp.pdf, 2007.

[4] Marcio Suenaga and Marco A.S.Netto. Programação por restrições. http://grenoble.ime.usp.br/ gold/cursos/2005/mac5758/seminario-marcio-marco.pdf, 2005.