

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Instituto de Informática
INF01037 - COMPUTAÇÃO EVOLUTIVA
Prof. Dante Augusto Couto Barone

Time Table Problem

JEFFERSON RODRIGO STOFFEL
JOÃO LUIZ GRAVE GROSS

Porto Alegre, julho de 2013.

1 Problema

O trabalho consiste em uma implementação do problema de “time table” para alocação de disciplinas a uma turma de ensino médio. A aplicação tem como meta gerar uma tabela de horários para uma turma do ensino médio, respeitando soft e hard constraints, conforme definição do trabalho.

2 Solução

2.1 Método de solução

Para solucionar o problema de alocação de professores em turmas dada sua disponibilidade utilizaremos a heurística de algoritmos genéticos.

2.2 Condições de funcionamento

É necessário que o usuário tenha um banco de dados MySQL instalado no seu computador e que já tenha previamente criado um banco de dados chamado “computacaoevolutiva”. Neste banco de dados devem ser importadas as informações contidas no arquivo “quadrohorario_alterado.sql”, acessado pelo caminho <pasta raiz do projeto>/misc/.

2.3 Dados Utilizados

Os dados utilizados para a realização da tarefa vêm de um banco de dados sql fornecido aos alunos pelo moodle, porém seu provável autor é um aluno da PUC-RS que nos introduziu ao problema. Neste banco de dados, existem 4 turmas de ensino fundamental (5º, 6º, 7º e 8º séries) cuja a carga horária é de 25 períodos semanais e uma turma do ensino médio (1º ano do segundo grau) cuja a carga horária é de 28 períodos semanais.

2.4 Estrutura do Cromossomo

Nesta etapa alteramos o banco de dados diminuindo a carga horária da turma do ensino médio para 25 períodos semanais. Este procedimento foi adotado pelo fato de todos os professores só possuírem horários disponíveis de segunda a sexta-feira no período da manhã. Para a turma do segundo grau alcançar os 28 períodos semanais, esta deveria ter algumas aulas no período da tarde ou em sábados. Logo, mantendo-se a configuração original, seria

impossível atingir uma resposta válida.

O fato de todas as turmas possuírem a mesma quantidade de períodos semanais implica em uma simplificação na estrutura do cromossomo, que pode ser representado como uma matriz em que todos os lugares preenchidos, e uma simplificação no algoritmo de maneira geral, que não precisa se preocupar em verificar espaços vazios na matriz. Desta maneira, um cromossomo é representado através da matriz cujas colunas são as turmas e as linhas são os períodos ordenados cronologicamente dentro de uma semana, isto é, tem-se os períodos 0 ao 4 da Segunda-Feira seguido dos períodos 0 ao 4 da Terça-Feira, e assim por diante. A figura abaixo representa este cromossomo.

Turma 51	Turma 61	Turma 71	Turma 81	Turma 101
Segunda-feira 0	Segunda-feira 0	Segunda-feira 0	Segunda-feira 0	Segunda-feira 0
Segunda-feira 1	Segunda-feira 1	Segunda-feira 1	Segunda-feira 1	Segunda-feira 1
Segunda-feira 2	Segunda-feira 2	Segunda-feira 2	Segunda-feira 2	Segunda-feira 2
Segunda-feira 3	Segunda-feira 3	Segunda-feira 3	Segunda-feira 3	Segunda-feira 3
Segunda-feira 4	Segunda-feira 4	Segunda-feira 4	Segunda-feira 4	Segunda-feira 4
Terça-feira 0	Terça-feira 0	Terça-feira 0	Terça-feira 0	Terça-feira 0
Terça-feira 1	Terça-feira 1	Terça-feira 1	Terça-feira 1	Terça-feira 1
...
...
Sexta-feira 4	Sexta-feira 4	Sexta-feira 4	Sexta-feira 4	Sexta-feira 4

Figura 1: representação do cromossomo.

2.5 Função de Avaliação

A função de avaliação consiste em atribuir pontos a um dado escalonamento de maneira que é possível dizer qual é o melhor entre vários escalonamentos. A soma de todos os pontos dados por uma função de avaliação a uma solução é chamado de Valor de Fitness. Desta maneira, quando uma disciplina é atribuída a um horário para o qual o seu tutor tem disponibilidade, nossa função de avaliação atribui o valor 100 ao Valor de Fitness daquela solução. Assim, sendo 5 turmas, cada uma com 25 períodos semanais, uma solução válida terá um valor maior ou igual 12500.

Se houverem repetições sequenciais da mesma disciplina em um dia para alguma das turmas, haverá o acréscimo em um do Valor de Fitness para cada repetição. Essa diferença tão grande de valores entre um escalonamento válido (100) e uma repetição (1) tem como explicação o fato de que se os valores fossem mais próximos, uma solução com várias repetições poderia ser erroneamente considerada uma solução válida. Assim, supondo que fosse gerado um escalonamento em que todos os períodos se repetem, para todas as turmas,

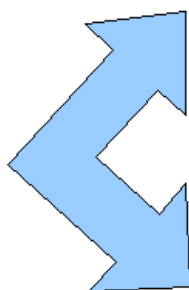
em todos os dias da semana teríamos: 4 (5 períodos tem no máximo 4 repetições) x 5 turmas x 5 dias = 100, o que na prática é impossível. Desta maneira, não há como repetições camuflarem uma resposta inválida.

2.6 Seleção

A seleção é utilizada para escolher quais dentre todos os cromossomos irão continuar a fazer parte do algoritmo genético. Para tal utilizamos o método mais comum, conhecido como algoritmo da roleta. O método consiste em sortear os cromossomos de acordo com sua aptidão, ou seja, cada cromossomo tem uma probabilidade de ser sorteado proporcional ao seu Valor de Fitness.

2.7 Crossover

Assim como o crossover estudado pela genética os algoritmos genéticos trocam material de seus cromossomos a fim de produzir melhores soluções. Para cada par de cromossomos selecionados pelo método da roleta, existe a probabilidade de ocorrer crossover. Esta probabilidade pode ser modificada pelo usuário através da Interface do programa. Sua implementação se dá da seguinte maneira: se for determinado que o par deve sofrer crossover, é então sorteada uma turma (coluna) e um intervalo de períodos (linhas) que serão trocados entre os dois cromossomos (matriz). Em seguida, se houver algum conflito entre a área trocada do crossover o restante do escalonamento para aquela turma, como, por exemplo, horários de determinadas disciplinas em excesso e(ou) disciplinas faltando, a parte conflitante **que não faz parte da área do crossover** é realocada de maneira a acabar com estes conflitos. As demais turmas (colunas) permanecem iguais ao que eram antes da operação de crossover. Na imagem a seguir podemos ver uma representação simples do crossover.



Turma 51	Turma 61	Turma 71	Turma 81	Turma 101
Segunda-feira 0	Segunda-feira 0	Segunda-feira 0	Segunda-feira 0	Segunda-feira 0
Segunda-feira 1	Segunda-feira 1	Segunda-feira 1	Segunda-feira 1	Segunda-feira 1
Segunda-feira 2	Segunda-feira 2	Segunda-feira 2	Segunda-feira 2	Segunda-feira 2
Segunda-feira 3	Segunda-feira 3	Segunda-feira 3	Segunda-feira 3	Segunda-feira 3
Segunda-feira 4	Segunda-feira 4	Segunda-feira 4	Segunda-feira 4	Segunda-feira 4
Terça-feira 0	Terça-feira 0	Terça-feira 0	Terça-feira 0	Terça-feira 0
Terça-feira 1	Terça-feira 1	Terça-feira 1	Terça-feira 1	Terça-feira 1
...
...
Sexta-feira 4	Sexta-feira 4	Sexta-feira 4	Sexta-feira 4	Sexta-feira 4

Turma 51	Turma 61	Turma 71	Turma 81	Turma 101
Segunda-feira 0	Segunda-feira 0	Segunda-feira 0	Segunda-feira 0	Segunda-feira 0
Segunda-feira 1	Segunda-feira 1	Segunda-feira 1	Segunda-feira 1	Segunda-feira 1
Segunda-feira 2	Segunda-feira 2	Segunda-feira 2	Segunda-feira 2	Segunda-feira 2
Segunda-feira 3	Segunda-feira 3	Segunda-feira 3	Segunda-feira 3	Segunda-feira 3
Segunda-feira 4	Segunda-feira 4	Segunda-feira 4	Segunda-feira 4	Segunda-feira 4
Terça-feira 0	Terça-feira 0	Terça-feira 0	Terça-feira 0	Terça-feira 0
Terça-feira 1	Terça-feira 1	Terça-feira 1	Terça-feira 1	Terça-feira 1
...
...
Sexta-feira 4	Sexta-feira 4	Sexta-feira 4	Sexta-feira 4	Sexta-feira 4

Figura 2: representação do crossover.

2.8 Mutação

Assim como o crossover, na mutação também existe a troca de material. No entanto, esta troca ocorre dentro do próprio cromossomo. Após todos os cromossomos passarem pela fase de crossover (em que podem ou não sofrer a operação de crossover), a cada um é aplicada uma mutação com uma dada probabilidade. Essa probabilidade pode ser modificada pela interface do programa. A implementação de mutação se dá da seguinte maneira:

“se um cromossomo for selecionado para sofrer mutação, é então selecionada uma turma (coluna) e dois períodos (linhas). Assim, estes dois períodos são trocados de lugar. Abaixo vemos uma imagem que representa a mutação.”

Turma 51	Turma 61	Turma 71	Turma 81	Turma 101
Segunda-feira 0	Segunda-feira 0	Segunda-feira 0	Segunda-feira 0	Segunda-feira 0
Segunda-feira 1	Segunda-feira 1	Segunda-feira 1	Segunda-feira 1	Segunda-feira 1
Segunda-feira 2	Segunda-feira 2	Segunda-feira 2	Segunda-feira 2	Segunda-feira 2
Segunda-feira 3	Segunda-feira 3	Segunda-feira 3	Segunda-feira 3	Segunda-feira 3
Segunda-feira 4	Segunda-feira 4	Segunda-feira 4	Segunda-feira 4	Segunda-feira 4
Terça-feira 0	Terça-feira 0	Terça-feira 0	Terça-feira 0	Terça-feira 0
Terça-feira 1	Terça-feira 1	Terça-feira 1	Terça-feira 1	Terça-feira 1
...
...
Sexta-feira 4	Sexta-feira 4	Sexta-feira 4	Sexta-feira 4	Sexta-feira 4

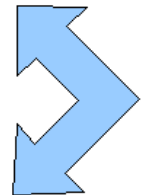


Figura 3: representação da mutação.

2.9 Elitismo

Inicialmente nossa solução não continha elitismo, o que fazia com que o Valor de Fitness ficasse oscilando de modo que nunca se chegava a uma resposta válida. A introdução do elitismo pressiona o algoritmo a produzir soluções com valores de Fitness cada vez mais altos. Sua implementação foi realizada da seguinte forma: sempre é guardado o cromossomo que possui o maior valor de Fitness. Após uma iteração do algoritmo genético, se todos os cromossomos produzidos possuírem valor de Fitness inferior ao valor de Fitness do cromossomo guardado, na próxima seleção será introduzida uma cópia do cromossomo guardado.

3 Pseudo-código

Pseudo-código

- gera_cromossomos aleatoriamente
- calcula fitness dos cromossomos

- gera_melhor aleatoriamente
- calcula fitness melhor

laço até condição de parada faça

- seleção pelo método da roleta de $n-1$ cromossomos
 - se usar melhor então introduz cópia do melhor
 - senão seleciona mais um pela roleta

- crossover nos pares $(0..1), (2..3)..(n-2..n-1)$ com probabilidade p_c

- aplica mutação em cada cromossomo com probabilidade p_m

- calcula fitness dos cromossomos gerados

- se algum possui fitness mais alto que o melhor
 - atualiza melhor

- se o fitness de todos for pior que o melhor
 - usa melhor na próxima iteração

fim laço

- se fitness melhor ≥ 12500
 - exibe fitness e tabela

4 Interface

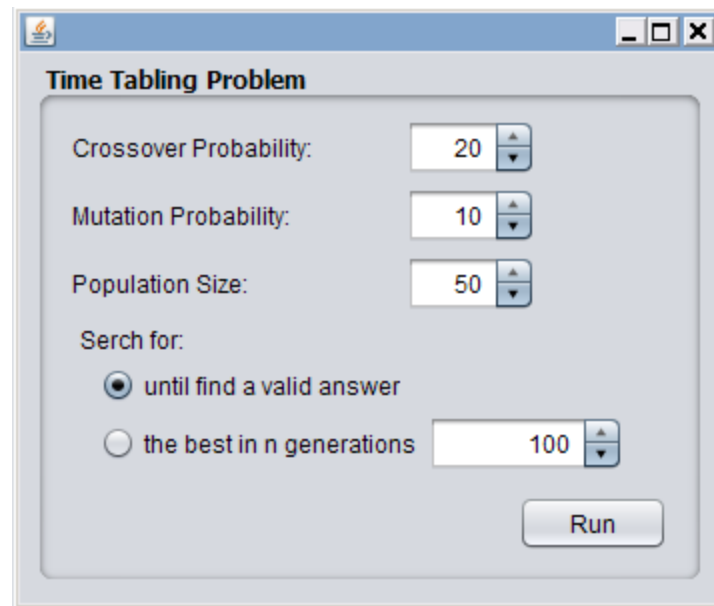


Figura 4: Interface.

5 Considerações

Como não é garantido que o algoritmo irá gerar uma resposta válida, às vezes, quando está próximo de uma solução, o valor de Fitness parece não melhorar. Não sabemos se é por que o algoritmo caiu em um mínimo local, ou se simplesmente o crossover e a mutação não estão atingindo o local correto para melhorar a solução. Por este motivo, modificamos o programa para que o mesmo fique imprimindo o valor de Fitness do melhor cromossomo e também o cromossomo (aqui em sua forma de implementação, contendo os identificadores da tabela grade). Assim, se o usuário notar que o valor de Fitness do melhor cromossomo não melhora, este pode decidir se continua ou pára a execução. Vemos na figura abaixo como é essa impressão.


```

Fitness Value of Biggest of this iteration: 11735; Fitness Value of Best: 12138
45, 12, 25, 42, 37,
33, 12, 25, 48, 44,
23, 40, 47, 36, 44,
23, 40, 47, 36, 43,
11, 46, 41, 31, 43,
39, 46, 30, 42, 22,
39, 46, 30, 42, 20,
45, 17, 35, 5, 32,
39, 46, 35, 5, 32,
39, 34, 18, 48, 43,
45, 3, 52, 14, 38,
45, 3, 13, 53, 49,
50, 24, 13, 48, 49,
45, 24, 4, 42, 49,
28, 51, 47, 42, 55,
16, 40, 8, 26, 54,
16, 40, 8, 26, 22,
2, 40, 18, 9, 27,
33, 7, 41, 19, 27,
39, 34, 41, 19, 10,
2, 29, 47, 9, 1,
11, 29, 47, 31, 1,
6, 46, 41, 14, 21,
6, 17, 41, 48, 15,
28, 7, 4, 48, 15,

```

valor de Fitness do melhor cromossomo

melhor cromossomo

Figura 5: Impressão de dados.

6 Banco de dados

Nossa aplicação é integrada com o banco de dados MySQL, logo é necessário instalar um servidor MySQL no computador para que possam ser feitas leituras das informações presentes no banco de dados e também realizar comunicação entre o programa Java e o banco de dados.

Para instalar o MySQL utilizamos o Wamp Server para Windows, no qual além de servidor Apache e PHP há também é instalado um servidor MySQL e o phpmyadmin para gerência do banco de dados. E para a comunicação do programa com o banco de dados utilizamos a interface Java Database Connectivity (JDBC) e o driver JDBC para MySQL..

7 Como configurar e como executar

Desenvolvemos o projeto na IDE Eclipse, logo, para que a aplicação possa ser

executada é preciso criar um projeto Java no eclipse e importar todos os códigos fonte para a pasta scr/. Também é necessário configurar o projeto para realizar a comunicação com o banco de dados. O driver mysql_connector deve ser incluído no projeto através do seguinte caminho <botão direito>/Build Path/Configure Build Path/aba Libraries/Add External Jars/mysql-connector-java-5.1.25-bin.

Ao configurar o projeto para executá-lo, basta clicar com o botão direito sobre a pasta raiz e dar o comando Run as/Java Application.

8 Resultados

- Probabilidade crossover: 20
- Probabilidade mutação: 10
- Tamanho população: 50
 - Gerações: 8551
 - Fitness: 12538
- Probabilidade crossover: 20
- Probabilidade mutação: 10
- Tamanho população: 100
 - Gerações: 57429
 - Fitness: 12538

SimpleTableDemo					
Turma/Período	Segunda-Feira	Terça-Feira	Quarta-Feira	Quinta-Feira	Sexta-Feira
51 / 0	Gabriela/Inglês	Virginia/Espanhol	Jefferson/Educação Física	Fabiana/Matemática	Edilcia/Geografia
51 / 1	Gabriela/Inglês	Virginia/Espanhol	Jefferson/Educação Física	Fabiana/Matemática	Fernanda/Português
51 / 2	Daniel/Ciências	Fernanda/Português	Renato/Ensino Religioso	Fabiana/Matemática	Maria/Educação Artística
51 / 3	Daniel/Ciências	Fernanda/Português	Edilcia/Geografia	Fabiana/Matemática	Maria/Educação Artística
51 / 4	Fernanda/Português	Fernanda/Português	Marcelo/História	Fabiana/Matemática	Marcelo/História
61 / 0	Jefferson/Educação Física	Fabiana/Matemática	Fernanda/Português	Renato/Ensino Religioso	Marcelo/História
61 / 1	Jefferson/Educação Física	Fabiana/Matemática	Fernanda/Português	Edilcia/Geografia	Adir/Ciências
61 / 2	Gabriela/Inglês	Fabiana/Matemática	Fernanda/Português	Maria/Educação Artística	Fernanda/Português
61 / 3	Gabriela/Inglês	Fabiana/Matemática	Adir/Ciências	Virginia/Espanhol	Marcelo/História
61 / 4	Edilcia/Geografia	Fabiana/Matemática	Fernanda/Português	Virginia/Espanhol	Maria/Educação Artística
71 / 0	Daniel/Matemática	Fernanda/Português	Adir/Ciências	Daniel/Matemática	Maria/Educação Artística
71 / 1	Edilcia/Geografia	Fernanda/Português	Adir/Ciências	Daniel/Matemática	Marcelo/História
71 / 2	Jefferson/Educação Física	Virginia/Espanhol	Edilcia/Geografia	Daniel/Matemática	Marcelo/História
71 / 3	Jefferson/Educação Física	Virginia/Espanhol	Fernanda/Português	Daniel/Matemática	Fernanda/Português
71 / 4	Gabriela/Inglês	Gabriela/Inglês	Renato/Ensino Religioso	Maria/Educação Artística	Fernanda/Português
81 / 0	Fernanda/Português	Gislaine/Ciências	Edilcia/Geografia	Maria/Educação Artística	Fernanda/Português
81 / 1	Fernanda/Português	Gislaine/Ciências	Edilcia/Geografia	Maria/Educação Artística	Jefferson/Educação Física
81 / 2	Fernanda/Português	André/Matemática	André/Matemática	Edilcia/História	Jefferson/Educação Física
81 / 3	Fernanda/Português	André/Matemática	Renato/Ensino Religioso	Gabriela/Inglês	Virginia/Espanhol
81 / 4	André/Matemática	André/Matemática	Edilcia/História	Gabriela/Inglês	Virginia/Espanhol
101 / 0	André/Matemática	André/Matemática	Renato/Ensino Religioso	Edilcia/Sociologia	Henrique/Filosofia
101 / 1	André/Matemática	Henrique/História	Geferson/Literatura	Gislaine/Física	Maria/Educação Artística
101 / 2	Geferson/Português	Henrique/História	Adir/Biologia	Gislaine/Física	Daniel/Química
101 / 3	Geferson/Português	Gabriela/Inglês	Jefferson/Educação Física	Edilcia/Geografia	Daniel/Química
101 / 4	Geferson/Português	Virginia/Espanhol	Jefferson/Educação Física	Edilcia/Geografia	Adir/Biologia

Figura 6: Tabela de resultados.

9 Melhorias futuras

Por simplicidade, implementamos o algoritmo de maneira que uma turma possa ter em um único dia aulas somente de uma disciplina. Na vida real, esta alocação provavelmente não seria aceita, pois tanto os alunos quanto os professores teriam baixo rendimento devido ao cansaço que tal alocação provoca.

Outra melhoria, seria implementar o método da roleta de maneira que cada cromossomo recebesse uma fatia por ordem do valor de Fitness, e não proporcional ao mesmo como implementamos. Atualmente, um cromossomo com valor de Fitness muito mais alto que os demais domina quase toda a roleta.

Além disto, como foi dito no tópico “Considerações”, muitas vezes quando o valor de Fitness do cromossomo não se sabe se é por ter caído em um mínimo local ou se é por que o crossover e as mutações não estão sendo aplicados nos locais necessários. Desta maneira, uma melhoria proposta seria fazer crossovers e mutações de maneira dirigida quando o algoritmo estiver perto de uma solução válida, escolhendo os pontos que tornam o cromossomo inválido.

10 Referências

Integração de Java e MySQL. Disponível em: <http://www.vogella.com/articles/MySQLJava/article.html>. Acesso em: 5/6/2013.

Como instalar um servidor MySQL. Disponível em: <http://www.vogella.com/articles/MySQL/article.html>. Acesso em: 8/6/2013.

Wamp Server: Apache, PHP e MySQL. Disponível em: <http://www.wampserver.com/en/>. Acesso em: 8/6/2013.

MySQL Connector. Disponível em: <http://dev.mysql.com/downloads/connector/j>. Acesso em: 8/6/2013.