

Gerador Automático de Horários com o uso da Inteligência Computacional

Maicon D. S. Matos¹, Clarimundo M. Moraes Jr¹

¹Instituto Federal do Triângulo Mineiro (IFTM) – Campus Patrocínio
CEP 38.740-000 – Patrocínio – MG – Brasil

maicon.iftm@yahoo.com.br, clarimundo@iftm.edu.br

Abstract. *This project aims to solve the problem of generating schedules using techniques of Computational Intelligence (CI). The project is in the second year of the research voluntarily and has partial satisfactory results with the use of Genetic Algorithms (GA). The goal is to generate schedules perfect or nearly perfect, and so the search continues for the purpose of using the solutions obtained by AG and improve them using some kind of Artificial Neural Network (ANN) that is able to classify such solutions indicating the classes more likely to generate better schedules.*

Resumo. *Este projeto visa resolver o problema da geração de horários com o uso de técnicas da Inteligência Computacional (IC). O projeto encontra-se no segundo ano da pesquisa de forma voluntária e possui resultados parciais satisfatórios com o uso de Algoritmos Genéticos (AG). A meta é gerar horários perfeitos ou quase perfeitos, e para isso a pesquisa continuará com o propósito de utilizar as soluções obtidas pelo AG e melhorá-las com o uso de algum tipo de Rede Neural Artificial (RNA) que seja capaz de classificar tais soluções indicando as classes mais propícias a gerarem horários melhores.*

1. Descrição do Problema

O problema de geração de horários escolares, também conhecido na literatura como *Timetabling Problem*, consiste em alocar os horários das disciplinas do(s) curso(s) durante um período de tempo (normalmente, uma semana), de modo a satisfazer um conjunto variado de restrições. Segundo Souza (2000), a solução manual deste problema é uma tarefa difícil e que requer um tempo considerável, por exemplo, a elaboração manual de um quadro de horários pode demandar duas semanas de trabalho em uma escola secundária ou até um mês em uma universidade. O intuito, então, é fazer um quadro de horário semanal, de tal forma que: as cargas horárias de todas as disciplinas sejam cumpridas; um professor não dê aula para mais de uma turma em um mesmo horário; horários vagos sejam minimizadas ou não ocorram para uma mesma turma; as indisponibilidades de cada professor sejam respeitadas.

2. Objetivos

2.1. Objetivo Geral

Utilizar técnicas da IC para auxiliar na busca de soluções ótimas ou satisfatórias ao problema da geração de horários.

2.2. Objetivos Específicos

- Estudar e entender técnicas da IC;
- Aplicar técnicas da IC ao problema de geração de horários;
- Auxiliar a gestão dos coordenadores de cursos da Instituição no que se refere à elaboração efetiva de horários semestrais.

3. Referencial Teórico

Neste capítulo será abordado as técnicas de IC que estão sendo utilizadas no projeto.

3.1. Algoritmos Genéticos (AG)

Os AG foram criados por Holland (1975) e se baseiam na teoria da evolução natural descritos por Darwin, 1859. Um dos destaques para esta técnica é que ela tem se mostrado bem eficiente em resolver problemas de busca em que o espaço dessa busca é muito grande, além do fato de que a busca realizada é de forma global, o que é particularmente interessante para evitar que se caia em ótimos locais, [Srinivas et AL 1994].

Segundo Ribeiro Filho (2000), os AG são baseados num processo coletivo de aprendizagem dentro de uma população, cada um dos quais representando um ponto no espaço de busca de soluções para um dado problema. A população é inicializada e evolui através de gerações com o uso de operadores de *seleção*, *reprodução* e *mutação*.

3.2. Rede Neural Artificial (RNA)

As RNA são técnicas computacionais que apresentam um modelo matemático inspirado na estrutura neural de organismos inteligentes e que adquirem conhecimento através da experiência. Uma grande rede neural artificial pode ter centenas ou milhares de unidades de processamento; já o cérebro de um mamífero pode ter muitos bilhões de neurônios. O modelo proposto por McCulloch e Pitts (1943) pode ser generalizado culminando no modelo geral do neurônio. Sabe-se que com um modelo geral de neurônio e arquiteturas de redes mais complexas é possível resolver problemas que envolvem reconhecimento de padrões, classificação e aproximações.

O aprendizado das RNA pode ser do tipo supervisionado (quando se sabe quais são as saídas desejadas do problema) e do tipo não supervisionado (quando não se sabe quais serão as saídas da rede). Este segundo tipo pode ser indicado em problemas de classificação onde a priori não se sabe quais classes de padrões existem no escopo do problema a ser tratado. Um exemplo de RNA com aprendizado não supervisionado são as redes *Kohonen*, cujo princípio básico é fazer com que os pesos associados a cada conexão entre a camada de entrada e a camada de saída se aproximem cada vez mais dos seus respectivos valores de entrada [Kohonen 1989].

4. Resultados e Testes

Os testes foram realizados em base nas avaliações feitas pelo sistema, sendo que cada restrição encontrada no horário era atribuído uma penalização neste indivíduo, ou seja, quanto maior a nota pior é o horário. É importante frisar que cada restrição tinha uma nota de penalização distinta, na figura 1.a, é possível notar que as disciplinas com *background* amarela, estão com restrições leves violadas.

Assim que a *implementação* do AG foi finalizado, teve início ao primeiro teste e

foi diagnosticado que os horários, se encontravam com um número de restrições muito elevado, em média de 374, conforme mostrado no Gráfico 1 (linha azul). Por este motivo teve início ao refinamento, que é a melhora da precisão do AG, com o objetivo de diminuir esta nota acarretando uma melhora nos horários gerados.

O primeiro refinamento foi realizado no método da seleção, onde era selecionado o primeiro pai entre os 25 melhores indivíduos e o segundo pai entre todos os indivíduos, ambos aleatórios, tendo uma pequena melhora nos horários cuja média obtida foi de 328 conforme mostrado no Gráfico 1 (linha laranja). Mas o resultado adquirido ainda estava longe do ideal, contudo realizou-se o segundo refinamento, desta vez na mutação, cuja probabilidade de ocorrência era de 10% e cuja troca de horário passou a ser de um único horário com outro, em cada período, ambos aleatórios (antes era selecionado um período e através deste período era trocado um dia com o outro). Com esta alteração, a nota passou para 215 conforme mostrado no Gráfico 1 (linha cinza).

No início, eram realizadas 100 (cem) gerações de novos indivíduos, como mostra o Gráfico 1, e o tempo para gerar os horários era inferior a 2 segundos, ou seja, os horários eram gerados em um tempo relativamente curto e seu resultado era de baixa qualidade. Por este motivo optou-se aumentar o número de gerações e assim buscar obter resultados melhores (nota menor para os indivíduos). Desta forma, o número máximo de gerações que foi adotado no sistema foi de 10.000 (dez mil) iterações, e assim obteve-se os melhores resultados com um tempo ainda relativamente pequeno (12 segundos) e com uma nota média valendo 64. Testes com o número máximo de gerações superior a dez mil, também foram realizados chegando ao patamar de 30.000 (trinta mil) gerações, mas os resultados não melhoraram significativamente (nota média valendo 59) e o tempo para gerar esses horários aumentou para 40 segundos. Em suma, ao termino de cada modificação, os resultados foram aperfeiçoados fazendo a média da nota cair de 374 para 64.

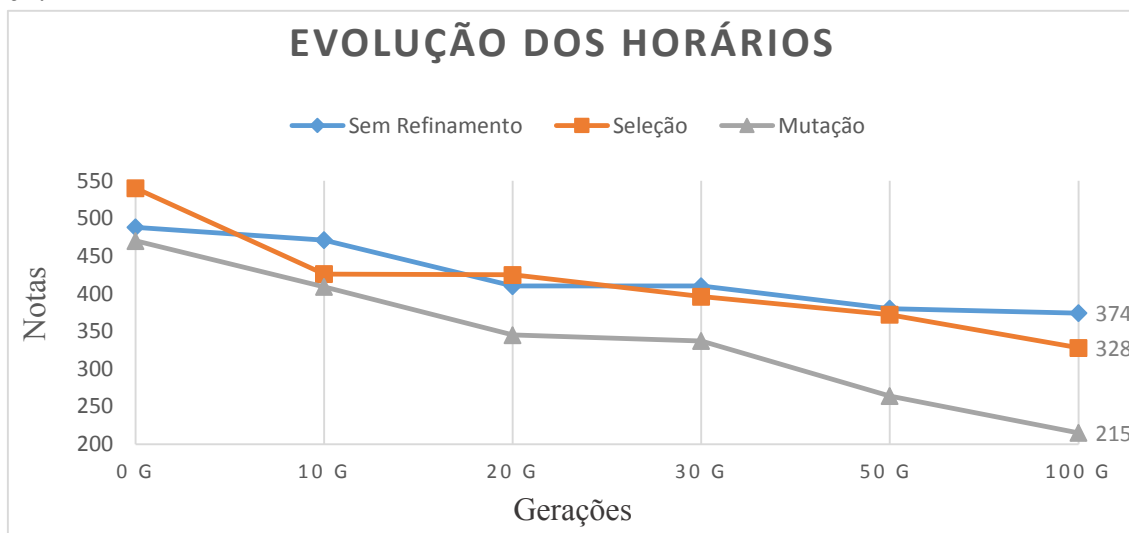


Gráfico 1. Refinamento do AG

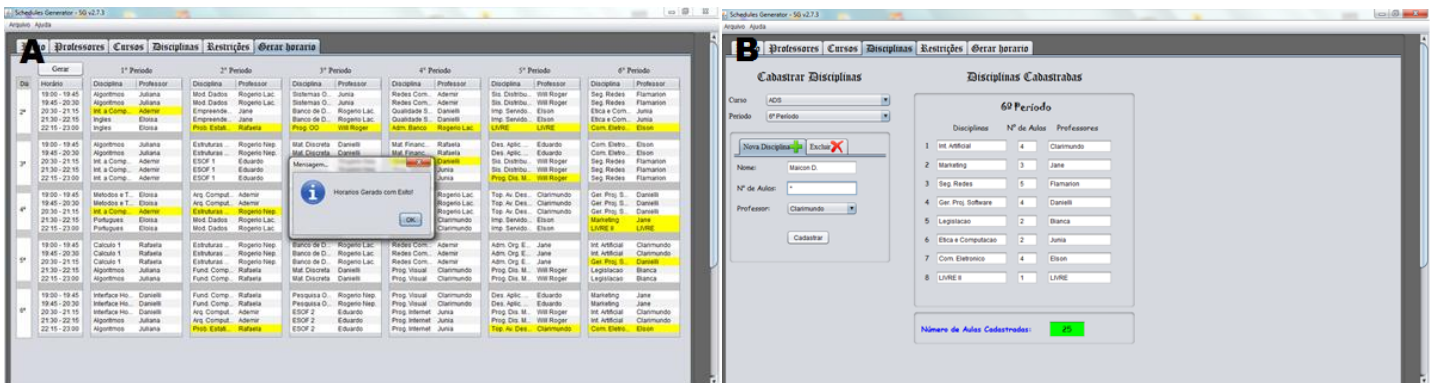


Figura 1. Telas do Sistema: a) Horário Gerado, b) Cadastro de Disciplinas.

5. Conclusões e Trabalhos Futuros

Com o uso do refinamento dos parâmetros e operadores é possível melhorar a precisão dos AG de forma significativa e aperfeiçoando os seus resultados.

Nos testes foi diagnosticado que aumentando as gerações, os resultados melhoravam gradativamente, porém chega um momento que o AG para de convergir, tendo perda de tempo e processamento.

Com a utilização dos AG é possível lidar com as restrições do problema no sentido de otimizá-lo com a minimização das violações delas. Espera-se que a obtenção de soluções ótimas (perfeitas) seja obtida aliando o uso dos AG com outras técnicas da IC (redes neurais artificiais, por exemplo).

O projeto não obteve resultados perfeitos, contudo será almejado o mesmo, sendo porvindouro a utilização de RNA, com o intuito desta perfeição.

Referencias

- Holland, J. H. “**Adaptation in natural and artificial systems**”. MIT Press, Cambridge, MA, USA, 1975.
- Kohonen, T. “**Self-Organization and Associative Memory**”. 3. ed. Springer, 1989.
- McCulloch, W. S.; Pitts, W. “**A logical calculus of the ideas immanent in nervous activity**”. Bulletin of Mathematical Biophysics. Vol. 5. (p. 115-133), 1943.
- Ribeiro Filho, G. In: Melhoramentos no Algoritmo Genético Construtivo e Novas Aplicações em Problemas de Agrupamento. INPE, São José dos Campos, SP, 2000. Disponível em: <<http://www.lac.inpe.br/~lorena/geraldo/tese-geraldo.pdf>>. Acesso em: 19 MAR. 2013.
- Souza, M. J. F. **Programação de horários em escolas: uma aproximação por metaheurísticas**. Tese de Doutorado. Programa de Engenharia de Sistemas e Computação. UFRJ, Rio de Janeiro, RJ, 2000.
- Srinivas, N; Deb, K. In: "Multiobjective optimization using nondominated sorting in genetic algorithms". Tese de Mestrado. Indian Institute of Technology, Kanpur, India, 1994. Disponível em: <<http://www.politespider.com/papers/general/Multiobjective%20Optimization%20Using%20Nondominated%20Sorting%20in%20Genetic%20Algorithms.pdf>>. Acesso em: 24 MAI. 2013.