

Problema de alocação de salas de aulas: uma abordagem via algoritmos genéticos

Marco Aurélio Bueno Carone- thecoreh@gmail.com

Vinícius Tinti de Paula Oliveira- viniciustinti@gmail.com

Rogério Martins Gomes- rogerio@lsi.cefetmg.br

Marco Túlio Alves Nolasco Rodrigues- tulio@dri.cefetmg.br

Centro Federal de Educação Tecnológica de Minas Gerais, CEFET-MG

Laboratório de Sistemas Inteligentes. Avenida Amazonas 5253.

30510-000 - Belo Horizonte, MG, Brasil

Resumo.

O Problema de alocação de salas de aulas, conhecido na literatura como Timetabling Problem, é um problema do tipo NP-difícil. Essencialmente, o problema consiste em: dado um conjunto de professores, um conjunto de salas e um conjunto de turmas, procura-se encontrar a melhor distribuição possível de forma a otimizar os recursos existentes em uma determinada escola ou universidade. A solução ótima para esse tipo de problema é extremamente complexa e, na maioria das vezes, inviável. Sendo assim, o uso de heurísticas, como os algoritmos genéticos, pode ser uma excelente alternativa na busca por soluções aproximadas a um baixo custo computacional.

Keywords: Algoritmos genéticos, problema de alocação de salas, Timetabling problem

1. INTRODUÇÃO

O problema de alocação de salas diz respeito à distribuição de aulas em um conjunto de salas obedecendo a um conjunto de restrições. Esse problema é conhecido na literatura como NP-difícil, ou seja, não é conhecido um algoritmo que o resolva em tempo polinomial (Carter, 1986). Achar a solução ótima dessa classe de problemas é computacionalmente inviável, pois se trata de um problema com ordem de complexidade fatorial (Schuermann, 1996; Hornegger & Paulus, 1999). Sendo assim, diversos métodos baseados em heurísticas foram desenvolvidos na tentativa de se buscar uma solução que, se não ótima, se aproxime dela. Esses métodos possuem diversos pontos em comum e, em geral, apresentam boas soluções com tempo e custo razoáveis.

Basicamente, o problema de alocação de salas é constituído por 4 entidades que devem formar relacionamentos (Abranson, 1991). Essas entidades são formadas por professores (p), turmas (t), salas (s) e horários (h). Uma aula (a) consiste em uma tupla formada por horário (h), professor (p), sala (s), turma (t) que tem o seguinte significado: no horário h o professor p ministra uma aula a para a turma t no horário h . Além disso, o horário não necessariamente se limita a um dia podendo ser representado por uma tupla $\{horas, minutos, dia, mes, ano\}$.

Esse tipo de problema, por sua vez, possui um elevado número de restrições, tais como:

1. os horários são limitados;
2. o número de salas é limitado;

3. o número de aulas de um determinado professor ou turma é limitado;
4. um professor ou uma turma não pode ter aulas em um determinado horário;
5. algumas aulas não podem ocorrer em certas salas;
6. um professor p ministra aulas somente das disciplinas X , Y e Z ;
7. uma turma t tem aulas das disciplinas X e Y .

Além disso, o modelo a ser criado deve levar em consideração um conjunto de regras, quais sejam:

1. $p_{a_1} = p_{a_2}, h_{a_1} = h_{a_2}$

Um professor não pode ministrar duas aulas no mesmo horário.

2. $s_{a_1} = s_{a_2}, h_{a_1} = h_{a_2}$

Em uma sala não podem existir duas aulas no mesmo horário.

3. $t_{a_1} = t_{a_2}, h_{a_1} = h_{a_2}$

Uma turma não pode assistir duas aulas no mesmo horário.

É interessante observar, também, a existência de algumas restrições chamadas de flexíveis, ou seja, aquelas que são desejadas, mas é necessariamente uma obrigação. São estas as inúmeras restrições que tornam o problema complexo (Abranson, 1991; Bardadym, 1996), motivando a decisão de se usar heurísticas na tentativa de solucionar esse problema.

No desenvolvimento desse trabalho optou-se pelo uso de algoritmos genéticos (Goldberg, 1989; Koza, 1992; Russell & Norvig, 2003) devido ao seu baixo custo computacional, além de apresentar uma implementação rápida e fácil.

2. SISTEMA DE ALOCAÇÃO DE SALAS

Nesse trabalho, simularemos uma versão simplificada do problema de alocação de salas de aulas. Essa simulação consiste, basicamente, em alocar aulas para um conjunto de 5 turmas e 10 professores que podem ocorrer em cinco horários diários e em cinco dias por semana. Além disso, o escopo foi limitado ao considerar que o número de salas não pode ser maior que o número de aulas, ou seja, não poderá haver conflitos de salas.

De uma maneira geral, o problema tratado nesse trabalho pode ser definido pelas seguintes restrições:

- os horários são limitados (25 horários semanais);
- não podem existir conflitos;
- um professor pode ministrar somente um conjunto pré-estabelecido de disciplinas;
- uma turma necessita ter aulas de um conjunto fixo de disciplinas;
- o professor possuirá uma carga horária fixa semanal;
- a turma possuirá uma carga horária fixa semanal.

Além disso, as seguintes restrições foram definidas como flexíveis:

- os professores devem concentrar seu horário (não possuir horários vagos);
- os professores devem concentrar sua carga horária (minimizar o número de dias que devem ministrar aulas);
- as turmas devem concentrar seu horário (não possuir horários vagos);
- as turmas devem concentrar sua carga horária (minimizar o número de dias na escola).

O sistema foi construído para resolver um caso simplificado do problema de alocação de salas de aulas, servindo de base para o desenvolvimento de alternativas ou métodos que poderiam ser aplicados na resolução de restrições mais complexa do problema.

O algoritmo genético proposto consistiu, basicamente, de mutações, pois a realização de crossover entre as soluções tornou o problema ainda mais complexo (Timmis et al., 2000; V. et al., 2007; Garrett, 2005; McEwan & Hart, 2009; Lau et al., 2009). Dessa forma, para contornar a ausência de *crossover*, adotou-se a estratégia de manter vários indivíduos e mutar cada um deles de acordo com sua pertinência.

A função de *fitness* consiste em contar o número de horários vagos de professores e turmas e a minimização desse valor definiu o quão boa é a solução. Além disso, o sistema foi programado para levar em consideração um fator de ponderação (*FP*) entre beneficiar as turmas ou os professores. Dessa forma, a função de *fitness* adotada nesse trabalho foi:

$$\text{Fitness} = \text{FP} * \text{NHVP} + (1 - \text{FP}) * \text{NHVA}, \quad (1)$$

onde:

- FP: Fator de ponderação;
- NHVP: Número de horários vagos dos professores;
- NHVA: Número de horários vagos dos alunos.

No algoritmo proposto nesse trabalho, qualquer solução com conflito é desconsiderada e sofre mutação. Esse processo ocorre enquanto não houver um resultado factível ou o tempo de processamento alcance um limiar.

3. RESULTADOS EXPERIMENTAIS

No intuito de analisar a eficiência do algoritmo implementado e, assim poder comparar os resultados obtidos, foram realizados uma série de testes que apresentaram bons resultados.

Como foi criado o fator de ponderação, responsável pelo balanceamento entre turmas ou professores, uma situação de *tradeoff* foi acrescentada ao problema. Assim, melhorar os horários dos professores em geral significará piorar os horários das turmas.

As Figuras 1 e 2 indicam a configuração final do sistema quando o fator de ponderação é ajustado para privilegiar os alunos, isto é, nesse cenário eleva-se a alocação de aulas em determinados dias da semana de maneira a beneficiar os alunos. Por outro lado, as Figuras 3 e 4 indicam a configuração final do sistema quando o fator de ponderação é ajustado para privilegiar os professores, isto é, para esta configuração, o modelo computacional gera uma solução, na qual observa-se uma concentração das aulas de um mesmo professor.

Turma 1

	Segunda	Terça	Quarta	Quinta	Sexta
13h	AEDS2:Cristina	AEDS2:Cristina			IA:Cardeal
14h	AEDS2:Cristina	AEDS:Cristina			AEDS:Cristina
15h	CG:Cardeal	CG:Cardeal			AEDS2:Cristina
16h	AEDS:Cristina	IA:Cardeal		CG:Cardeal	CG:Cardeal
17h	AEDS2:Cristina	IA:Cardeal		AEDS:Cristina	IA:Cardeal

Turma 2

	Segunda	Terça	Quarta	Quinta	Sexta
13h	AEDS2:Cardeal	AEDS:Cardeal	IA:Rogério	AEDS:Cardeal	
14h	AEDS2:Cardeal	AEDS2:Cardeal	AEDS2:Cardeal	AEDS2:Cardeal	
15h	CG:André	SD:Jefferson	SD:Jefferson	CG:André	
16h	CG:André	SD:Jefferson	IA:Rogério	CG:André	
17h	IA:Rogério	IA:Rogério	AEDS:Cardeal	AEDS:Cardeal	

Figura 1: Exemplo de alocação de aulas das turmas 1 e 2 quando o fator de ponderação foi configurado para privilegiar os alunos

Fausto

	X			
X		X		
	X			
X				X
X		X		X

Albens

X				
X				
X	X			
X	X			

Figura 2: Exemplo de alocação de aulas de dois professores quando o fator de ponderação foi configurado para privilegiar os alunos

Turma 1

	Segunda	Terça	Quarta	Quinta	Sexta
13h	IA:Cardeal	AEDS2:Evandrino			IA:Cardeal
14h		AEDS2:Evandrino	CG:Cardeal		AEDS:Cardeal
15h	AEDS:Cardeal	AEDS2:Evandrino		CG:Cardeal	CG:Cardeal
16h	CG:Cardeal	AEDS2:Evandrino		AEDS:Cardeal	
17h		IA:Cardeal	IA:Cardeal	AEDS2:Evandrino	AEDS:Cardeal

Turma 2

	Segunda	Terça	Quarta	Quinta	Sexta
13h		IA:Cardeal	IA:Cardeal	IA:Cardeal	SD:Jefferson
14h	AEDS2:Cardeal	CG:Cardeal		AEDS2:Cardeal	
15h		AEDS2:Cardeal	CG:Cardeal	AEDS:Evandrino	SD:Jefferson
16h		IA:Cardeal	CG:Cardeal	AEDS:Evandrino	CG:Cardeal
17h	AEDS2:Cardeal	AEDS:Evandrino	AEDS:Evandrino	AEDS2:Cardeal	SD:Jefferson

Figura 3: Exemplo de alocação de aulas das turmas 1 e 2 quando o fator de ponderação foi configurado para privilegiar os professores

Jefferson

	X			X
	X			X
	X			X
	X			X
	X			X

Figura 4: Exemplo de alocação de aulas de um professor quando o fator de ponderação foi configurado para privilegiar os professores

4. CONCLUSÃO

Neste artigo, foi apresentada uma proposta baseada em algoritmos genéticos para a solução do problema de alocação de salas de aulas. O algoritmo apresentou bons resultados com um baixo custo computacional.

Como foi criado o fator de ponderação, responsável pelo balanceamento entre turmas ou professores, uma situação de *tradeoff* foi acrescentada ao problema. Assim, melhorar os horários dos professores em geral significará piorar os horários das turmas. Dessa forma, torna-se necessário determinar o valor ótimo desse fator equilíbrio ou ponderação.

Um característica interessante verificada nas simulações foi com relação aos horários dos professores. No algoritmo desenvolvido houve uma tendência à concentração de horários em único professor. Por exemplo, se um professor p_1 leciona apenas duas aulas em um único dia e

um outro professor p_2 leciona aulas todos os dias exceto em dois horários o algoritmo convergia no sentido de deixar o professor p_2 ainda mais ocupado e o professor p_1 sem aulas.

Como perspectiva de trabalhos futuros deve-se procurar melhorar o algoritmo de forma a permitir que o mesmo possa trabalhar com mais variáveis e restrições, além de adicionar métricas de medidas de desempenho e comparação com outros métodos.

Agradecimentos

Os autores gostariam de agradecer ao Laboratório de Sistemas Inteligentes (LSI) pela infraestrutura disponibilizada e ao CEFET-MG e à FAPEMIG pelo suporte financeiro, sem o qual esse trabalho não teria sido viável.

Referências

- Abranson, D., 1991. Constructing school timetables using simulated annealing: Sequential and parallel algorithms. *Management Science*, vol. 113, n. 1.
- Bardadym, V. A., 1996. Computer-aided school and university timetabling: The new wave. *Management Science*, vol. 123, n. 1.
- Carter, M., 1986. A survey if practical applications of examination timetabling algorithms. *Operations Reserch*, vol. 34, n. 2.
- Garrett, S., 2005. How do we evaluate artificial immune systems? *Evolutionary Computation*, vol. 13, n. 2.
- Goldberg, D. E., 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional, 1 edition.
- Hornegger, J. & Paulus, D. W. R., 1999. *Applied Pattern Recognition: A Practical Introduction to Image and Speech Processing in C++ (2nd ed.)*. Morgan Kaufmann Publishers.
- Koza, J. R., 1992. *Genetic Programming: On the Programming of Computers by Means of Natural Selection (Complex Adaptive Systems)*. The MIT Press.
- Lau, H., Bate, I., & Timmis, J., 2009. An immuno-engineering approach for anomaly detection in swarm robotics. In *ICARIS*, pp. 136–150.
- McEwan, C. & Hart, E., 2009. On airs and clonal selection for machine learning. In *ICARIS*, pp. 67–79.
- Russell, S. J. & Norvig, P., 2003. *Artificial Intelligence: A Modern Approach*. Pearson Education.
- Schuermann, J., 1996. *Pattern Classification: A Unified View of Statistical and Neural Approaches*. Wiley.
- Timmis, J., Neal, M., & M.; Hunt, J., 2000. An artificial immune system for data analysis. *Byosystems*.
- V., C., Nicosia, G., M., P., & Timmis, J., 2007. An immune algorithm for protein structure prediction on lattice models. *Transactions on Evolutionary Computation*, vol. 11, n. 1, pp. 30–40.

Classrooms allocation problem: an approach via genetic algorithms

Abstract.

The classroom allocation problem, known in the literature as timetabling problem is a NP-hard problem. Essentially, the problem consists in finding the best distribution of classrooms as possible given a set of professors, rooms and classes. The optimal solution to this problem is extremely complex and, in most cases, unfeasible. Thus, the use of heuristics such as genetic algorithms can be an excellent alternative in the search for near-optimal solutions at relatively low computational cost.

Keywords: *Genetic algorithms, Timetabling problem*