

# Artigo Algoritmo Genético Timetabling Problem

Adriano Yuji Sato Vasconcelos<sup>1</sup>, Felipe Issamu Kuratomi<sup>1</sup>, Bianca Andréia da Silva Neto<sup>1</sup>

<sup>1</sup>Faculdade de Ciências Exatas e Tecnológicas - Universidade Federal da Grande Dourados

**Abstract.** *Reconciling the employees agenda and the institution's attributions is a highly complex task. To find the synchronism between resource and demand, we can use methods and tools that help in this task. Through the metaheuristic based on genetic algorithm and computer programs, in this case, javascript, it is possible to optimally solve the problem of time allocation taking into account its characteristics and constraints. This paper proposes the learning in the area of artificial intelligence, as well as the approach of the method employed in the article, contributing to the knowledge and academic growth when seeking the solution to the problem addressed.*

**Resumo.** *Conciliar a agenda dos colaboradores e as atribuições das instituições é uma tarefa de alta complexidade, para encontrar o sincronismo entre recurso e demanda pode se utilizar de métodos e ferramentas que auxiliam nesta tarefa. Através da metaheurística baseada no algoritmo genético e de programas computacionais, no caso, Javascript, é possível solucionar de forma otimizada o problema da alocação de horário levando em consideração suas características e restrições. Este trabalho propõe o aprendizado na área de inteligência artificial, tal como a abordagem do método empregado no artigo, contribuindo para o conhecimento e crescimento acadêmico ao buscar a solução para o problema tratado.*

## 1. Introdução

Estruturar de forma manual um horário acadêmico que respeite as atribuições da instituição, satisfaça as restrições do problema e atenda as necessidades dos docentes, torna-se inviável devido a complexidade do problema conforme o aumento das restrições dos indivíduos e o número do conjunto. Para otimizar o tempo e diminuir os recursos necessários, utiliza-se de soluções computacionais, sendo algumas, reflexos de outros campos do conhecimento.

Baseando-se na Biologia de Charles Darwin, os algoritmos genéticos tem o intuito de otimizar determinada solução, que partindo-se de uma "população"aléatoria e combinando-a iterativamente consigo mesma, possa gerar novas "populações"que representem resultados mais satisfatórios.

Muitos problemas computacionais pertencentes à classe dos NP-Difícil são sanados através dessa metaheurística, o problema da alocação de horário se enquadra neste meio de solução.

Portanto propõe-se neste trabalho a aplicação da metaheurística bio-inspirada na Teoria da Evolução, algoritmos genéticos, para a solução do problema de alocação de horários acadêmicos.

## 2. Algoritmo Genético

Diversos problemas computacionais são solucionados inspirados na natureza, observando cada indivíduo para chegar-se a um resultado coletivo, isto é como se define uma metaheurística. Sendo assim o algoritmo genético é baseado na teoria evolucionária de Charles Darwin, os seres sofrem mutações e os mais aptos irão sobreviver, este é um método bio inspirado desenvolvido por John Henry Holland em 1975.

Um algoritmo genético é constituído basicamente por, um método para gerar a população inicial, um operador de seleção, um operador de cruzamento e um operador de mutação. Portanto é necessário gerar uma população inicial aleatoriamente, definir como será feita a seleção dos indivíduos que passará os genes para a geração seguinte, "cruzar" os indivíduos selecionados e aplicar uma forma de mutação nos novos indivíduos, garantindo a variedade da solução, repetir esse processo até que se tenha a melhor solução.

Neste artigo será exposto o uso do algoritmo genético para solucionar o problema de alocação de horário.

## 3. Revisão da Literatura

O problema de geração de quadros de horários, também conhecido como *timetabling problem*, já foi muito revisado e estudado durante os anos e conta com diversos exemplos na literatura, como por exemplo, o escalonamento de enfermeiros [Cheang 2003], horários de aula em instituições de ensino [Burke 1997] [J. P. Caldeira 1997] e planejamento de transporte público [Loureço 2000].

Existem diversos casos de problemas de *timetabling* com suas restrições e exigências específicas. Durante as análises ficou evidente que não é conveniente, em termos computacionais, o uso de programação determinística para solucionar o *timetabling*. Muitos estudos são realizados para desenvolver métodos para assistir a descoberta de uma solução ótima, devido a sua complexidade computacional estimular o contínuo estudo sobre o tema [Burke 1997].

Na literatura foi apresentado um algoritmo evolutivo híbrido para resolver o problema de *timetabling* de horários escolares. Uma população inicial é gerada por um procedimento construtivo adaptativo e parcialmente guloso e, a cada geração, o membro que mais se destaca é submetido a um refinamento pelo método de Busca Tabu [Costa 1994][Burke 1997]. Nos testes iniciais realizados esse procedimento híbrido conseguiu apresentar soluções de melhor qualidade final do que aquele sem a fase de refinamento por Busca Tabu. O algoritmo sem esta fase de refino tem enorme dificuldade para encontrar soluções viáveis [Souza 2002].

[Freitas et al. 2007] desenvolveram um software, para a Faculdade Ruy Babarboza, que utiliza uma técnica de programação metaheurística baseada na seleção natural das espécies, os Algoritmos Genéticos. Esta ferramenta se chama Kairós que recebe diversas regras e suas prioridades e com cálculos de gerações e mutações utilizando a técnica de elitismo, é então gerada um cronograma com as aulas alocadas.

Outro modelo adaptado para algoritmos genéticos foi o modelo de ilhas. Neste modelo, indivíduos são divididos em subpopulações, em processos concorrentes, cada qual responsável por suas operações locais. Através de um processo de migração, os indivíduos de melhor avaliação podem migrar de uma ilha para outra [Miranda Lobo 2004].

## 4. Descrição do problema

O problema abordado neste artigo se enquadra no conjunto de problemas denominado *Timetabling Problems*, onde se procura organizar recursos acadêmicos específicos (horários, professores e disciplinas) em um determinado período de tempo. A organização ocorre pela alocação de disciplinas aos horários e professores disponíveis, sendo assim, para que a solução seja satisfatória é necessário que certas restrições sejam impostas.

### 1. Restrições rígidas:

São necessárias para a admissibilidade da solução, o desvio às restrições impossibilita a construção de uma solução e inviabiliza a utilização dos horários em um cenário real.

- Cada turma não pode ter duas disciplinas ao mesmo tempo.

As disciplinas ofertadas a cada turma não podem ser alocadas à um mesmo horário.

- Cada professor não pode ministrar duas disciplinas no mesmo horário.

A alocação de um professor à duas disciplinas no mesmo horário impossibilita a ministração das mesmas, o professor deve possuir no máximo uma disciplina para cada horário.

- Todas as turmas têm que ser alocadas a professores.

As turmas devem conter um professor para a sua gestão.

- Cada professor pode ter dias da semana e horários em que não pode dar aula.

A restrição anterior tem como objetivo a aproximação da solução para com um cenário real, onde os professores não estarão disponíveis a todo momento.

- A alocação deve garantir uma carga horária semanal mínima (8h) e máxima (20h) para todos os professores.

Deve ser garantida a carga horária prevista no regulamento da universidade em questão.

### 2. Restrições flexíveis:

São restrições que não afetam a admissibilidade, porém possibilitam o melhoramento da solução. São necessárias se o objetivo é obtenção da melhor solução e servem como medida de desempenho e comparação com outros algoritmos que não utilizam tais restrições.

- Minimizar as aulas vagas das turmas entre aulas alocadas.

Minimizando as aulas vagas entre aulas garante um horário mais flexível, permitindo a aceitação por parte dos alunos e professores.

- Agrupar aulas de uma mesma disciplina (2 em 2), ou seja, não podem ser alocadas menos de duas aulas seguidas de uma mesma disciplina.

O agrupamento distribuído das disciplinas permite horários menos desgastantes, tanto para os professores como para os alunos e facilita no planejamento das aulas ofertadas.

- Sempre que possível, deve haver um intervalo de tempo entre aulas consecutivas de disciplinas distintas.

O intervalo de tempo proporciona aos alunos e professores uma pausa, com o intuito de aumentar a produtividade, assim como facilita a troca de professores entre horários.

- Minimizar o choque de horários das disciplinas com maior percentual de reprovações.

É preferível para a construção de um horário ideal, onde o choque de disciplinas, mesmo que com baixo percentual de reprovações, não gera uma solução eficaz.

- Disciplina prática após a teórica na semana.

Idealmente a alocação de disciplinas práticas devem vir após as disciplinas teóricas, já que as aulas práticas servem como meio de aplicar o conhecimento adquirido em aulas teóricas.

- Divisão mais equitativa de carga entre os professores.

A divisão balanceada da carga horária entre professores permite um ambiente pedagógico mais democrático, evitando diferenças e conflitos entre os professores, além do que, a alocação da carga horária para apenas um professor se distânciava de uma solução eficaz.

## 5. Implementação

### 5.1. Representação de uma solução

Neste problema um cromossomo representa 5 quadros de horários semanais, visto que nossa solução tenta alocar horários para as 5 turmas presentes no curso. Cada quadro é uma matriz  $M_{ij}$  sendo  $i$  um número inteiro para representar a quantidade de aulas ministradas por dia, e  $j$  o número de dias na semana que haverá aula. Uma célula  $M_{ij}$  contém um objeto, este objeto contém uma variável professor e uma variável disciplina.

## 5.2. Geração da população Inicial

A população inicial é gerada de forma aleatória, uma população é composta por 100 indivíduos, cada indivíduo é composto por 5 semanas e cada semana é composta por 16 horários, cada horário recebe aleatoriamente dois valores inteiros, o primeiro indica o código do professor e o segundo o código da disciplina.

## 5.3. Operadores Genéticos

Com uma população de indivíduos estabelecida, são utilizados operadores genéticos até se obter uma melhor solução. Um valor é atribuído para a solução que define sua qualidade através de uma função *fitness*.

Esta função *fitness* avalia o indivíduo utilizando as restrições citadas na seção 4, Descrição do Problema, e retorna um *score*, quanto menor o *score* retornado, maior a qualidade da solução

### 5.3.1. Operador de Seleção

Com todos os indivíduos da população avaliados, o AG seleciona um grupo deles para participar do cruzamento (*crossover*). A seleção é realizada através da Seleção por torneio pelo fato de não ser um método extremamente elitista mas ainda sim consegue retornar indivíduos com um *score* satisfatório

Seleção por torneio: são sorteados aleatoriamente três indivíduos e faz-se uma comparação entre eles, o indivíduo com um menor *score* é selecionado, como mostra a Figura 1.

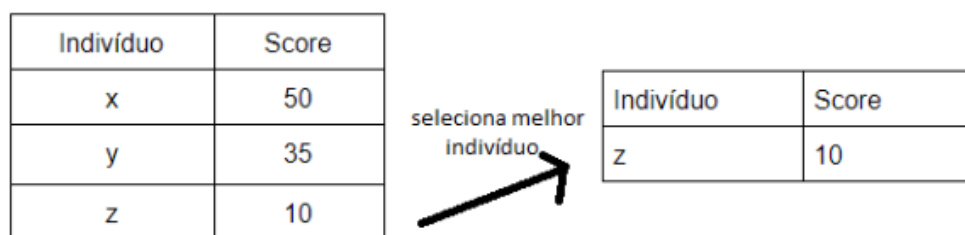


Figura 1. Demonstração do algoritmo de seleção por torneio

### 5.3.2. Operador de Cruzamento

A reprodução fica a cargo do cruzamento, mostrado na Figura 2, que mistura aos pares os genes remanescentes do processo de escolha e gera duas novas soluções, como mostrado na Figura 3. São escolhidos aleatoriamente um ponto de corte para efetuar o cruzamento dos indivíduos. Este método é conhecido como *Single Point Crossover*.

Segunda	Terça	Quarta	Quinta	Segunda	Terça	Quarta	Quinta
disciplina: 5 professor: 2	disciplina: 9 professor: 10	disciplina: 12 professor: 1	disciplina: 18 professor: 8	disciplina: 3 professor: 1	disciplina: 19 professor: 1	disciplina: 2 professor: 2	disciplina: 8 professor: 2
disciplina: 5 professor: 3	disciplina: 17 professor: 1	disciplina: 7 professor: 6	disciplina: 3 professor: 5	disciplina: 4 professor: 6	disciplina: 7 professor: 10	disciplina: 17 professor: 5	disciplina: 13 professor: 9
disciplina: 9 professor: 5	disciplina: 13 professor: 7	disciplina: 19 professor: 4	disciplina: 14 professor: 5	disciplina: 8 professor: 6	disciplina: 3 professor: 4	disciplina: 9 professor: 9	disciplina: 4 professor: 9
disciplina: 8 professor: 7	disciplina: 19 professor: 9	disciplina: 12 professor: 1	disciplina: 7 professor: 6	disciplina: 17 professor: 1	disciplina: 9 professor: 2	disciplina: 2 professor: 10	disciplina: 17 professor: 2

**Figura 2. Ponto de corte selecionado para cruzamento**

Segunda	Terça	Quarta	Quinta	Segunda	Terça	Quarta	Quinta
disciplina: 5 professor: 2	disciplina: 9 professor: 10	disciplina: 12 professor: 1	disciplina: 8 professor: 2	disciplina: 3 professor: 1	disciplina: 19 professor: 1	disciplina: 2 professor: 2	disciplina: 18 professor: 9
disciplina: 5 professor: 3	disciplina: 17 professor: 1	disciplina: 7 professor: 6	disciplina: 13 professor: 8	disciplina: 4 professor: 6	disciplina: 7 professor: 10	disciplina: 17 professor: 5	disciplina: 3 professor: 5
disciplina: 6 professor: 5	disciplina: 13 professor: 7	disciplina: 19 professor: 4	disciplina: 4 professor: 9	disciplina: 8 professor: 6	disciplina: 3 professor: 4	disciplina: 9 professor: 9	disciplina: 14 professor: 5
disciplina: 8 professor: 7	disciplina: 19 professor: 8	disciplina: 12 professor: 1	disciplina: 17 professor: 2	disciplina: 17 professor: 1	disciplina: 9 professor: 2	disciplina: 2 professor: 10	disciplina: 7 professor: 6

**Figura 3. Filhos gerados após o cruzamento**

### 5.3.3. Operador de Mutação

A mutação existe para garantir uma variedade maior de indivíduos e explorar melhor o espaço de soluções possíveis. Toda vez após o cruzamento existe um chance de 10% dos filhos sofrerem uma mutação. O procedimento escolhe um gene do indivíduo e gera aleatoriamente outros valores para a célula, como mostra a Figura 4.

Segunda	Terça	Quarta	Quinta
disciplina: 5 professor: 2	disciplina: 9 professor: 10	disciplina: 12 professor: 1	disciplina: 18 professor: 9
disciplina: 5 professor: 3	disciplina: 17 professor: 1	disciplina: 7 professor: 6	disciplina: 3 professor: 5
disciplina: 9 professor: 5	disciplina: 13 professor: 7	disciplina: 19 professor: 4	disciplina: 14 professor: 5
disciplina: 8 professor: 7	disciplina: 19 professor: 9	disciplina: 12 professor: 1	disciplina: 7 professor: 6

Matéria →

### Figura 4. Mutação

## 5.4. Seleção

Após executar o processo de seleção e cruzamento 50 vezes (visto que a o tamanho da população é 100, ou seja o algoritmo de seleção e cruzamento acontece  $PopulationSize/2$  vezes) serão gerados novos indivíduos, então passam pelo processo de seleção e cruzamento novamente, para assim estabelecer uma nova geração da população cada vez mais refinada com *scores* mais próximos da solução ideal.

## 5.5. Resultados

Os resultados obtidos podem ser analisados observando os *scores* da primeira geração com o da última geração, apresentados na Figura 5. É perceptível que a última geração possui *scores* mais próximos do desejável (0) e a primeira possui escores mais altos que são indesejáveis.

Primeira Geração		Última Geração	
INDIVÍDUO	SCORE	INDIVÍDUO	SCORE
1	45	1	0
2	60	2	0
3	65	3	0

Figura 5. 3 melhores indivíduos da primeira e última geração

## 6. Conclusão

Foi apresentado um algoritmo genético para resolver o problema de alocação de horário em universidades. O algoritmo gera uma população inicial, utiliza seleção por torneio para escolher os indivíduos que são cruzados por meio de *Single Point Crossover* para gerar membros mais próximos da solução a cada geração. Nos testes práticos foram geradas soluções que respeitam as restrições implementadas.

Como trabalhos futuros, é visado a implementação de mais restrições para um refinamento maior dos resultados, além da implementação de um objeto com três elementos (Horário, disciplina e professor) para maior flexibilidade na alocação dos professores.

## Referências

- Burke, K. E. (1997). Automated timetabling: the state of the art.
- Cheang, B. (2003). Nurse rostering problems: a bibliographic survey.
- Costa, D. (1994). A tabu search algorithm for computing an operational timetable.
- Freitas, C. C., Guimarães, P. R., Neto, M. C., and BARBOZA, F. J. (2007). Uma ferramenta baseada em algoritmos genéticos para a geração de tabela de horário escolar. *SÉTIMA ESCOLA REGIONAL DE COMPUTAÇÃO Bahia-Sergipe. Vitória da Conquista: [sn]*.
- J. P. Caldeira, C. R. (1997). *Dynamic school timetabling using genetic search*. INTERNATIONAL CONFERENCE ON THE PRACTICE AND THEORY OF AUTOMATED TIMETABLING.
- Lourenço, J. P. Paixão, R. P. (2000). The crew-scheduling module in the gist system.
- Miranda Lobo, O. (2004). Uma solução do problema de horário escolar via algoritmo genético paralelo.
- Souza, Costa, G. (2002). Um algoritmo evolutivo híbrido para o problema de programação de horários em escolas.