

# Uma Ferramenta Para Geração de Grades Horárias Utilizando Algoritmos Genéticos<sup>1</sup>

José Luender de Lima Santos, Yuri de Almeida Malheiros Barbosa

Centro de Ciências Aplicadas e Educação  
Universidade Federal da Paraíba (UFPB) – Rio Tinto, PB – Brasil

{jose.luender, yuri}@dcx.ufpb.br

***Abstract.** Allocate class schedules is a recurring and laborious task in any institution of higher education. The allocation of timetables is done respecting a series of constraints, making it a complex problem and difficult to solve, being generally treated with heuristic techniques. This paper addresses the problem of time allocation faced by the Bachelor's courses in Information Systems and Degree in Computer Science of Campus IV of the Federal University of Paraíba. As a solution, a tool, based on Genetic Algorithms, was developed to meet the needs of the courses. Through tests with real data, it was possible to verify the applicability of the proposal.*

***Resumo.** Alocar horários de aulas é uma tarefa recorrente e bastante trabalhosa em qualquer instituição de ensino superior. A alocação dos horários é feita respeitando uma série de restrições, fazendo desta, um problema complexo e de difícil solução, sendo tratados, geralmente, com técnicas heurísticas. Este trabalho aborda o problema da alocação de horários enfrentado pelos cursos de Bacharelado em Sistemas de Informação e Licenciatura em Ciência da Computação do Campus IV da Universidade Federal da Paraíba. Como solução, uma ferramenta, baseada em Algoritmos Genéticos, foi desenvolvida para satisfazer as necessidades dos cursos. Através de testes com dados reais, foi possível verificar a aplicabilidade da proposta.*

## 1. Introdução

Em todo período letivo, as instituições de ensino enfrentam um desafio: preparar a grade horária de aulas. Os horários devem ser alocados seguindo uma série de restrições que podem ser conflitantes, o que torna esta tarefa bastante complexa [Vieira e Macedo 2011]. Buscar uma solução de forma manual para o problema pode exigir muito tempo e esforço de várias pessoas e, mesmo assim, não produzir uma solução satisfatória [Fucilini et al. 2008].

Os cursos de Bacharelado em Sistemas de Informação (BSI) e Licenciatura em Ciência da Computação (LCC) do Campus IV da Universidade Federal da Paraíba (UFPB) também enfrentam problemas para definir a grade horária de aulas em todos os

---

<sup>1</sup> Trabalho de Conclusão de Curso apresentado pelo aluno **José Luender de Lima Santos** sob a orientação do professor **Yuri de Almeida Malheiros Barbosa** como parte dos requisitos para obtenção do grau de Bacharel em Sistemas de Informação na UFPB Campus IV

períodos letivos. Seus coordenadores não utilizam nenhum software especializado para auxiliá-los nesta tarefa, tudo é feito de forma manual. Durante os últimos semestres, uma grande quantidade de tempo e esforço foi gasto na tentativa de produzir grades horárias que atendessem às restrições existentes.

Na literatura, este problema é comumente conhecido como *timetabling* e é um dos mais desafiadores em otimização combinatória. É normalmente tratado com técnicas heurísticas em virtude da sua complexidade e dificuldade em tratá-lo com algoritmos exatos [Ciscon et al. 2005]. Algumas técnicas da Inteligência Artificial são utilizadas para resolver o problema de alocação de horários. Uma das técnicas mais encontradas na literatura, e que tem gerado bons resultados, é a de Algoritmos Genéticos [Santos e Santos 2004], um algoritmo baseado nos mecanismos de evolução das espécies. Nesta técnica, uma população de indivíduos (que representam possíveis respostas para o problema) passa por um processo evolutivo que tem o objetivo de gerar uma solução ótima entre as soluções candidatas por meio da melhora progressiva dos cromossomos dos indivíduos. Durante esse processo, os operadores genéticos (seleção, cruzamento e mutação) são aplicados, sendo responsáveis pela transformação e diversificação da população ao longo das gerações [Borges 2003].

Várias ferramentas foram desenvolvidas com o objetivo de gerar grades horárias de forma automatizada, mas as suas utilizações não são frequentes devido às restrições específicas de cada instituição [Vieira e Macedo 2011], por serem soluções fechadas ou pagas. Por conta disso, diversos trabalhos têm sido feitos no intuito de propor ferramentas que solucionem os problemas de alocação de horários de instituições e cursos específicos.

O objetivo geral deste trabalho é propor uma solução automática, baseada em Algoritmos Genéticos, para a geração das grades horárias dos cursos de BSI e LCC da UFPB. Assim, foram feitas entrevistas com os coordenadores dos cursos para identificar como o processo de alocação de horários vem sendo realizado e quais as suas principais necessidades. A partir dessas informações, foi possível identificar os requisitos que guiaram o desenvolvimento da ferramenta. Com o objetivo de se obter um bom algoritmo, diferentes técnicas de operadores genéticos foram implementadas e testadas com dados reais. Com os resultados obtidos pôde-se averiguar a eficiência da ferramenta em um contexto real.

O restante do trabalho está organizado da seguinte forma: a seção 2 aborda trabalhos relacionados. A seção 3 apresenta o problema da alocação de horários. A seção 4 explana conceitos sobre Algoritmos Genéticos. A seção 5 descreve o processo atual de geração de grades horárias dos cursos de BSI e LCC. A seção 6 detalha a implementação da ferramenta e dos algoritmos propostos. A seção 7 aborda os testes e os resultados obtidos, enquanto a seção 8 conclui o trabalho.

## **2. Trabalhos Relacionados**

Muitos trabalhos sobre resolução de *timetabling* utilizando AGs podem ser encontrados na literatura. Vieira e Macedo (2011) desenvolveram uma solução baseada em AGs para o departamento de computação da Universidade Federal de Sergipe. A solução apresentou bons resultados, satisfazendo as restrições existentes. Em outros trabalhos,

como os de Borges (2003), Hamawaki (2005) e Berardino (2013), também foram desenvolvidas ferramentas para departamentos e cursos específicos.

Santos e Santos (2004) propuseram uma ferramenta para a geração dos horários das aulas de uma escola de ensino médio. No trabalho foram encontradas soluções sub-ótimas. O trabalho de Ciskon et al. (2005) também apresentou um algoritmo genético voltado a escolas de ensino médio, mas com foco na eliminação de janelas e aulas isoladas nos horários dos professores.

Também é possível encontrar trabalhos que utilizam outras técnicas, como o de Souza et al. (2002) que propôs um algoritmo híbrido que utiliza a técnica GRASP para gerar a população inicial e um refinamento nos melhores indivíduos através do método de Busca Tabu.

Desse modo, os trabalhos citados não atendem totalmente as restrições específicas dos cursos de BSI e LCC. Assim, torna-se necessário o desenvolvimento de uma ferramenta própria para os cursos.

Na próxima seção é apresentado os conceitos referentes ao problema da alocação de horários.

### **3. O Problema da Alocação de Horários**

O problema da alocação de horários em universidades (*Course Timetabling*) consiste em combinar encontros entre professores e alunos em um período de tempo previamente determinado (uma semana, geralmente) que satisfaça um conjunto de restrições de diferentes tipos. Os tipos de restrições geralmente mudam de uma instituição para outra, assim, diversas variações do problema têm sido propostas na literatura [Ciskon et al. 2005].

A quantidade e a complexidade das restrições têm se tornado cada vez maiores e ainda não existe um algoritmo que seja capaz de testar todas as possibilidades de solução em tempo hábil, principalmente quando se tem muitos dados de entrada ou quando as restrições são muito complexas. Por estas dificuldades, observa-se na literatura que técnicas heurísticas são utilizadas para encontrar soluções aproximadas em menos tempo [Navarro e Coelho 2010]. Segundo Vieira e Macedo (2011), as técnicas que mais se destacam são as de algoritmos genéticos, satisfação de restrições, arrefecimento simulado, busca tabu e algoritmos de colônias de formigas.

As restrições podem ser divididas de acordo com seus graus de importância. As restrições *hard* são as que não podem ser violadas de modo algum, pois acarretam em soluções inválidas, como ter duas aulas de uma mesma turma acontecendo no mesmo dia e horário, por exemplo. Já as restrições *soft* são as que devem ser respeitadas, mas caso não sejam, não invalidam, necessariamente, a solução [Vieira e Macedo 2011]. Um exemplo frequente de restrição *soft* é a da não existência de horários vagos entre as aulas (janelas).

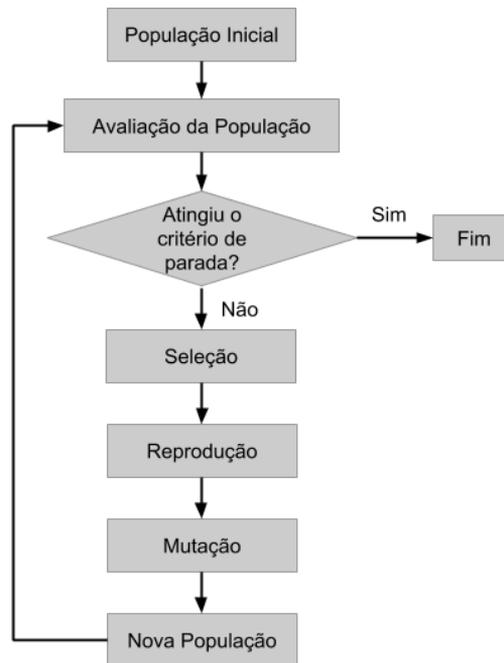
Na próxima seção serão abordados as características e o funcionamento dos Algoritmos Genéticos.

#### 4. Algoritmos Genéticos

Algoritmos Genéticos (AGs) são algoritmos heurísticos utilizados para solucionar problemas de otimização e busca, e fazem parte de um grupo conhecido como Algoritmos Evolutivos. Fundamentados por John Holland em 1975, os AGs são baseados nos mecanismos de evolução natural, propostos por Charles Darwin, e nos princípios da genética moderna, desenvolvidos por Gregor Mendel [Freitas et al. 2014].

Em AGs, os indivíduos de uma população disputam a oportunidade de se reproduzirem e propagarem seus genes para as próximas gerações. Cada indivíduo está associado a uma solução do problema e os mais adaptados possuem maiores chances de sobrevivência e reprodução [Santos e Santos 2004]. Segundo Vieira e Macedo (2011, p. 3): “O fundamento básico é criar indivíduos, avaliá-los, selecionar os mais aptos, realizar o cruzamento de seus ‘códigos genéticos’, e gerar novos indivíduos resultantes do processo, que possivelmente são mais adaptados.”.

A figura 1 demonstra o processo de execução de um AG. Inicialmente, uma população é gerada de forma aleatória ou através de alguma heurística. Em seguida, os indivíduos dessa população são avaliados por meio de uma função de avaliação, que define os seus *fitness* (valor que representa a qualidade do indivíduo). Uma condição de parada é testada para verificar se a solução foi encontrada, com base em um valor médio de *fitness*, ou se o algoritmo atingiu a quantidade de gerações especificada. A partir desta etapa, os operadores genéticos são aplicados. Na seleção, os indivíduos são escolhidos, a partir dos seus *fitness*, para participarem da etapa de cruzamento, onde irão trocar materiais genéticos e gerar os indivíduos que farão parte da próxima geração. Esporadicamente, ocorrem mutações nos indivíduos criados, onde acontecem alterações genéticas de forma aleatória, possibilitando o surgimento de novas soluções. E finalmente, uma nova população é formada e o processo recomeça [Vieira e Macedo 2011].



**Figura 1. Fluxo de execução de um Algoritmo Genético**

**Fonte: VIEIRA e MACEDO, 2011, p. 4**

Durante o processo de evolução, o cruzamento ou a mutação podem destruir o melhor indivíduo. Então, para evitar que uma possível melhor solução seja perdida, é utilizada a técnica de elitismo, onde os melhores indivíduos da geração atual são preservados na seguinte.

## 4.1. Operadores Genéticos

O núcleo da execução dos AGs é formado pelos operadores genéticos: seleção, cruzamento e mutação. A função desses operadores é a de transformar a população ao longo das sucessivas gerações, ampliando a busca até atingir um resultado satisfatório. São fundamentais para que a população possa se diversificar e manter características de adaptação obtidas em gerações anteriores [Ramos 2002]. Neste trabalho foram implementados seis operadores genéticos diferentes e que são explanados a seguir.

### 4.1.1. Seleção

Depois que a população inicial é formada e seus indivíduos são avaliados, os operadores genéticos são aplicados até a resposta ser encontrada ou a quantidade de gerações ser atingida. O primeiro operador aplicado é o de seleção.

Existem vários métodos de seleção e neste trabalho foram implementados os de seleção por torneio, seleção por roleta giratória e seleção por truncamento.

- **Seleção por torneio:** a seleção por torneio funciona de maneira simples: pequenos grupos de indivíduos são escolhidos aleatoriamente e o que possui o maior *fitness* é selecionado [Lucas 2002].
- **Seleção por roleta giratória:** na seleção por roleta giratória, cada indivíduo recebe uma fatia da roleta imaginária proporcional ao seu *fitness*. Assim que a

roleta é girada, os indivíduos mais aptos tendem a ser selecionados com mais frequência, pois possuem as maiores fatias [Freitas et al. 2014]. De acordo com Borges (2003), este é o método de seleção mais utilizado nos AGs.

- **Seleção por truncamento:** na seleção por truncamento, um valor é utilizado para limitar a quantidade de indivíduos que poderão ser escolhidos: dado um valor T, a seleção é realizada entre os T melhores indivíduos. Cada um desses indivíduos possui a mesma probabilidade de seleção [Lima 2006].

#### 4.1.2. Cruzamento

Após a etapa de seleção dos indivíduos ser realizada, a estratégia de recombinação é aplicada com uma certa probabilidade, onde os indivíduos que farão parte da próxima geração serão formados a partir da troca de informações genéticas dos indivíduos selecionados. Os métodos de cruzamento implementados neste trabalho foram os de cruzamento em um ponto, cruzamento em dois pontos e cruzamento uniforme. Segundo Borges (2003), esses são os métodos de seleção mais utilizados.

- **Cruzamento em um ponto:** um ponto de corte é selecionado de forma aleatória nos cromossomos dos indivíduos e o cruzamento do material genético é realizado a partir dele [Freitas et al. 2014].
- **Cruzamento em dois pontos:** o cruzamento em dois pontos é semelhante ao anterior. No entanto, em vez de um ponto, dois pontos de corte são selecionados de forma aleatória [Borges 2003].
- **Cruzamento uniforme:** neste método, os genes dos descendentes são criados a partir da cópia dos genes dos pais, que são escolhidos de acordo com uma máscara de cruzamento. Esta máscara é uma sequência de zeros e uns gerados de forma aleatória. Quando houver o valor 1 na máscara de cruzamento, o gene a ser copiado deve ser o do primeiro pai e quando houver o valor 0, o gene a ser copiado deve ser o do segundo pai [Lucas 2002].

Na próxima seção será descrito o processo de alocação de horários utilizado nos cursos de BSI e LCC.

### 5. O Processo Atual

A partir de entrevistas realizadas com os coordenadores dos cursos, pôde-se compreender o funcionamento do processo de alocação de horários utilizado. Atualmente, a preparação das grades de horários é feita de forma manual e começa a ser realizada meses antes do início do período letivo. Toda essa antecipação se faz necessária devido a definição das disciplinas que serão ministradas por cada professor, bem como os seus horários disponíveis, a resolução de algum problema que venha a acontecer e, principalmente, a complexidade envolvida nesta tarefa.

Nos cursos existe o conceito de créditos. Um crédito representa uma hora-aula semanal e cada disciplina possui uma certa quantidade de créditos. Recentemente, um crédito passou a ter o tempo de 50 minutos para se adequar à realidade dos cursos. Todas as disciplinas do curso de BSI possuem quatro créditos. A maioria das disciplinas do curso de LCC também possuem quatro créditos e, algumas poucas, seis, três e dois créditos. Também existe a ideia de pré-requisitos. Um pré-requisito é uma disciplina que deve ter sido cursada com aprovação para que se possa matricular outras de

períodos subsequentes. Cada professor leciona, em média, duas ou três disciplinas. As aulas acontecem em turno diurno, mas existe um grande interesse de que os horários ocupados durante a manhã sejam maximizados. Geralmente, as aulas terminam, no máximo, às 15:00. Outro ponto importante se refere a algumas disciplinas que são ofertadas por outros departamentos e os horários das suas aulas são definidos previamente e são invariáveis.

A seguir, serão apresentados os detalhes da implementação da ferramenta e dos AGs propostos.

## **6. A Ferramenta Proposta**

A ferramenta proposta neste trabalho foi desenvolvida na linguagem de programação Java utilizando a plataforma Java Standard Edition (Java SE) com o Java Development Kit (JDK) 1.8. Outras tecnologias utilizadas foram: I) o ambiente de desenvolvimento NetBeans IDE 8.0.2; II) o framework de mapeamento objeto-relacional Hibernate 5.0.1; III) o banco de dados relacional PostgreSQL 9.4; IV) o Sistema Gerenciador de Banco de Dados (SGBD) pgAdmin III 1.20.0.

Para que o algoritmo seja executado, é preciso que os dados que serão utilizados sejam cadastrados através dos formulários de cadastros presentes na ferramenta. Os dados que o usuário deve registrar são referentes aos cursos, disciplinas, professores e suas disponibilidades de horários, e turmas.

### **6.1. Implementação dos Algoritmos Genéticos**

Através das entrevistas foi possível especificar e classificar as principais restrições existentes na geração das grades horárias. Alguns pontos bastante específicos dos cursos podem ser observados, como alocar disciplinas que são pré-requisitos no mesmo horário. Isto se deve a tentativa de permitir que alunos reprovados em algumas disciplinas não sejam muito prejudicados com choques de horários entre disciplinas muito importantes e outras. Abaixo é possível ver as seis restrições *hard* e as quatro restrições *soft* levantadas.

Restrições *hard*:

- As aulas devem ser alocadas em dupla, ou seja, duas horas-aula seguidas;
- Duas aulas de disciplinas diferentes do mesmo período não podem estar alocadas no mesmo dia e horário;
- Os professores só devem ministrar suas aulas nos horários que os mesmos indicaram como disponíveis;
- As disciplinas ofertadas por outros departamentos possuem horários fixos;
- Um professor não pode lecionar duas disciplinas diferentes em um mesmo dia e horário;
- A quantidade de aulas de uma disciplina não pode ser excedida.

Restrições *soft*:

- Maximizar a quantidade de disciplinas por dia para o professor;
- Minimizar a quantidade de dias sem aula por período;
- Priorizar a alocação das aulas nos horários da manhã;

- As aulas das disciplinas que são pré-requisitos devem ser alocadas no mesmo horário.

### 6.1.1. Representação da Solução

Conforme Freitas et al. (2014), a representação da solução é de fundamental importância para o funcionamento correto dos AGs.

Neste trabalho, cada indivíduo é formado por todas as aulas que deverão ser ministradas no período letivo. O indivíduo é constituído por uma representação vetorial, onde cada gene (posição no vetor) armazena um objeto do tipo Aula, na qual é possível acessar as informações referentes a determinada aula, como: disciplina, professor, horário, etc.

A(1)	A(2)	A(3)	A(4)	A(5)	...	A(n)
------	------	------	------	------	-----	------

**Figura 2. Representação do indivíduo**

Como no trabalho de Fucilini et al. (2008), a granularidade do intervalo entre uma aula e outra são de dois créditos. Desse modo, o primeiro horário da manhã, por exemplo, equivale ao intervalo das 08:00 até 09:40.

### 6.1.2. População Inicial

Como cada aula (gene do indivíduo) que deverá ser alocada possui as informações da disciplina e do professor que irá lecioná-la, a população inicial é criada de forma simples: um horário, entre os horários disponíveis do professor, é atribuído de modo aleatório a cada uma das aulas que devem ser alocadas. A exceção são as aulas das disciplinas ofertadas por outros departamentos que já possuem seus horários cadastrados e que devem ser fixos.

Os horários foram representados através de números inteiros, por conta da facilidade de manipulação desses tipos de dados. Cada horário representa um determinado *slot* (posição na grade) que a aula deverá ocupar. Por exemplo, se o horário 4 for atribuído a uma aula, isso significa que ela será ministrada na terça-feira das 08:00 até 09:40. Dessa maneira, a restrição *hard* de alocação de aulas duplas já é completamente atendida. A Tabela 1 mostra os slots correspondentes aos horários disponíveis para as disciplinas. Essa representação também facilita a das aulas na exibição da grade na interface com o usuário, onde o horário da aula determina qual *slot* que deverá ser ocupado por ela.

**Tabela 1. Representação dos horários**

Horário	Segunda	Terça	Quarta	Quinta	Sexta
08:00 - 09:40	0	4	8	12	16
09:50 - 11:30	1	5	9	13	17
13:10 - 14:50	2	6	10	14	18
14:50 - 16:40	3	7	11	15	19

### 6.1.3. Função de Avaliação

Neste trabalho, os indivíduos são avaliados a partir das violações de restrições cometidas pelos mesmos. Quando uma restrição é violada, o indivíduo é penalizado. O valor da penalidade varia de acordo com o tipo de restrição violada (*soft* ou *hard*).

Como no trabalho de Simão (2013), o valor do *fitness* de cada indivíduo é dado pela seguinte função:

$$Fitness = \frac{100}{100 + \sum penalidades}$$

Através da função, é possível perceber que, caso o indivíduo não viole nenhuma restrição, o maior *fitness* que poderá ser alcançado é 1.

Os valores das penalidades de cada tipo de restrição podem ser vistos na tabela abaixo.

**Tabela 2. Valores das penalidades**

Tipo de restrição	Penalidade
Restrição <i>hard</i>	100
Restrição <i>soft</i>	10

Como pode-se notar, as restrições *hard* ganham um alto valor de penalidade, assim elas tendem a ser descartadas de forma natural através da etapa de seleção, não sobrevivendo em gerações futuras.

### 6.1.4. Seleção

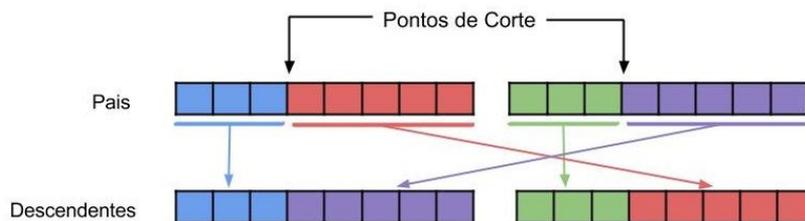
Nenhuma técnica de seleção foi adaptada, sendo implementadas como descritas na subseção 4.1.1.

Na seleção por torneio, três indivíduos são selecionados aleatoriamente e os dois com os maiores *fitness* são selecionados para terem seus genes propagados. Na seleção por truncamento, o valor do limiar utilizado foi 0.4, ou seja, apenas os 40% melhores indivíduos podem ser selecionados para participarem das próximas etapas.

### 6.1.5. Cruzamento

Assim como as técnicas de seleção, nenhuma técnica de cruzamento foi adaptada, sendo implementadas como descritas na subseção 4.1.1. e sem validação de indivíduos anômalos.

A figura 3 demonstra um exemplo de como é realizado o cruzamento em um ponto. Observa-se que dois novos indivíduos são gerados a partir da troca de informações genéticas de outros dois indivíduos (pais).



**Figura 3. Exemplo de cruzamento em um ponto**

### 6.1.6. Mutação

A mutação implementada consiste em alterar, de forma aleatória, o horário atribuído a uma aula (gene) por outro. Da mesma maneira que na geração dos indivíduos da população inicial, o novo horário atribuído pertence a lista dos horários disponíveis do professor. Portanto, a troca é feita de forma simples: o horário antigo é excluído e o novo, obtido aleatoriamente, é alocado a aula. Assim, o algoritmo tem chances de explorar novas características e não permanecer em mínimos locais. Se a nova característica for ruim, será eliminada nas gerações seguintes, caso contrário, permanecerá. A ferramenta possibilita que o usuário defina o valor da taxa de mutação.

### 6.1.7. Finalização

A execução do algoritmo é finalizada quando o número máximo de gerações que deve ser percorrido é atingido ou quando se obtém o indivíduo que possui o maior *fitness* possível. O número máximo de gerações pode ser definido pelo usuário.

O algoritmo também utiliza a técnica de elitismo, preservando o melhor indivíduo da geração atual na próxima.

## 7. Testes e Resultados

Para a execução dos testes foram utilizados os dados referentes ao semestre 2015.1. Os dados foram obtidos através da análise da grade horária produzida no período citado, onde foi possível identificar as disciplinas ofertadas, os professores que as ensinaram, os dias em que cada professor ministrou suas aulas, etc. A partir desses dados foram formados dois grupos de testes. O grupo 1 composto por todas as aulas das disciplinas ofertadas do curso de BSI e o grupo 2 por todas as aulas das disciplinas ofertadas do curso de LCC. Os atributos dos dois grupos estão listados abaixo:

Grupo de testes 1:

- 25 disciplinas distribuídas entre 4 períodos diferentes;
- 50 aulas no total;
- 20 professores para ministrar as disciplinas;
- 1 disciplina possuía um pré-requisito;
- Todas as disciplinas possuíam 4 créditos;
- 4 disciplinas eram oferecidas por outros departamentos;
- Cada professor (exceto os que iriam ministrar disciplinas oferecidas por outros departamentos) possuía 6 horários disponíveis cadastrados (3 em um dia da semana e 3 em outro).

Grupo de testes 2:

- 46 disciplinas distribuídas entre 10 períodos diferentes;
- 85 aulas no total;
- 31 professores para ministrar as disciplinas;
- 7 disciplinas possuíam crédito 2;
- 2 disciplinas possuíam crédito 6;
- 37 disciplinas possuíam crédito 4;
- 8 disciplinas eram oferecidas por outros departamentos;
- Cada professor (exceto os que iriam ministrar disciplinas oferecidas por outros departamentos) possuía 6 horários disponíveis cadastrados (3 em um dia da semana e 3 em outro).

Como a ferramenta funciona apenas para disciplinas de créditos pares, as disciplinas de créditos ímpares foram cadastradas com um crédito a menos. Isso foi aplicado por dois momentos em disciplinas, originalmente, de crédito três.

Para definir os parâmetros utilizados pelos algoritmos, foram realizados testes prévios com diversos valores e, a partir da análise dos resultados gerados, foi possível estabelecer os seguintes parâmetros:

- População de 200 indivíduos;
- 800 gerações;
- 75% de taxa de cruzamento;
- 5% de taxa de mutação;
- Não existência de pré-processamento para remover indivíduos anômalos.

Os 6 operadores genéticos utilizados na implementação dos AGs foram combinados e geraram 9 configurações diferentes, como listado abaixo:

- Configuração 1 (C1): seleção por torneio e cruzamento em um ponto;
- Configuração 2 (C2): seleção por torneio e cruzamento em dois pontos;
- Configuração 3 (C3): seleção por torneio e cruzamento uniforme;
- Configuração 4 (C4): seleção por roleta giratória e cruzamento em um ponto;
- Configuração 5 (C5): seleção por roleta giratória e cruzamento em dois pontos;
- Configuração 6 (C6): seleção por roleta giratória e cruzamento uniforme;
- Configuração 7 (C7): seleção por truncamento e cruzamento em um ponto;
- Configuração 8 (C8): seleção por truncamento e cruzamento em dois pontos;
- Configuração 9 (C9): seleção por truncamento e cruzamento uniforme;

Com essas configurações, foi realizado um conjunto de testes para verificar a qualidade de suas respostas. Cada configuração foi executada 10 vezes e o *fitness* do melhor indivíduo da população final coletado para comparação posterior. A tabela 3 mostra os resultados obtidos para o grupo de testes 1.

**Tabela 3. Resultados do grupo de testes 1**

	C1	C2	C3	C4	C5	C6	C7	C8	C9
1ª	0.7143	0.5556	0.6667	0.0408	0.0348	0.0347	0.3125	0.6667	0.7143
2ª	0.6667	0.6250	0.7143	0.0413	0.0450	0.0383	0.5882	0.6250	0.6250
3ª	<b>0.7692</b>	0.6250	0.7143	0.0442	0.2941	0.0328	0.7143	0.6667	0.6250
4ª	<b>0.7692</b>	0.6667	0.5882	0.0375	0.8333	0.0377	0.3030	0.3030	0.5882
5ª	0.6250	0.7692	0.2857	0.0379	0.0439	0.0405	0.2857	0.6667	0.2857
6ª	0.5882	0.3030	0.7143	0.0435	0.0351	0.0407	0.6667	0.5556	0.7692
7ª	0.6250	0.2778	0.2941	0.0352	0.0407	0.0365	0.5882	0.5882	0.6250
8ª	0.6667	0.7692	0.2941	0.0382	0.0355	0.0376	0.7692	0.7143	0.2941
9ª	0.6667	0.2941	0.2857	0.0345	0.0457	0.0437	0.6667	0.3125	0.6667
10ª	0.3030	0.6667	0.7143	0.0348	0.0383	0.0397	0.2941	0.5556	0.3030
<b>Média</b>	<b>0.6394</b>	0.5552	0.5272	0.0388	0.1446	0.0382	0.5189	0.5654	0.5496

Como é possível observar, a configuração que apresentou a melhor média de resultados foi a C1 (seleção por torneio e cruzamento em um ponto), que, por sua vez, obteve os maiores valores nas execuções de número 3 e 4.

Nos testes realizados com o grupo 2 foram obtidos os seguintes resultados:

**Tabela 4. Resultados do grupo de testes 2**

	C1	C2	C3	C4	C5	C6	C7	C8	C9
1ª	0.2222	0.3333	0.2381	0.0217	0.0197	0.0203	0.3846	0.1562	0.2174
2ª	0.3571	0.2273	0.4348	0.0205	0.0195	0.0224	<b>0.5000</b>	0.3846	0.3333
3ª	0.4167	0.3125	0.2128	0.0235	0.0183	0.0233	0.4167	0.2128	0.3571
4ª	0.4348	0.3704	0.4000	0.0199	0.0189	0.0233	0.2041	0.2381	0.4167
5ª	0.4167	0.4167	0.3704	0.0196	0.0206	0.0208	0.1562	0.3846	0.1493
6ª	0.2222	0.3571	0.2381	0.0204	0.0215	0.0237	0.4000	0.4000	0.1493
7ª	0.2222	0.3226	0.2083	0.0190	0.0224	0.0191	0.4348	0.2273	0.2128
8ª	0.2128	0.2326	0.2174	0.0200	0.0233	0.0202	0.4167	0.4545	0.1515
9ª	0.1515	0.3704	0.3846	0.0228	0.0218	0.0202	0.2174	0.3846	0.2041
10ª	0.2222	0.2083	0.2128	0.0232	0.0202	0.0203	0.3846	0.3704	0.3846
<b>Média</b>	0.2878	0.3151	0.2917	0.0211	0.0206	0.0214	<b>0.3515</b>	0.3213	0.2576

Desta vez, a configuração C7 (seleção por truncamento e cruzamento em um ponto) foi a que obteve a melhor média, tendo como valor máximo o obtido na segunda execução.

A partir dos resultados apresentados em ambos os testes, é possível perceber que as configurações dos métodos de seleção por torneio (C1, C2 e C3) e por truncamento (C7, C8 e C9) apresentaram médias de resultados relativamente próximas, mostrando que todos conseguem, em algum momento, encontrar soluções com certos níveis de qualidade. Os piores resultados foram obtidos pelas configurações que possuem a técnica da roleta giratória como método de seleção. As técnicas de cruzamento também não apresentaram diferenças significativas quando executadas com os mesmos métodos de seleção.

Os gráficos apresentados nas figuras 4 e 5 mostram a curva da evolução dos melhores e dos piores indivíduos de cada grupo ao longo das gerações. Do grupo de testes 1 foram utilizados os dados obtidos na terceira execução da configuração C1 e do grupo de testes 2 foram utilizados os dados da segunda execução da configuração C7. Nota-se que, no primeiro caso, há uma grande convergência até a geração de número 145 e no segundo até a geração de número 235, demonstrando que quanto mais aulas, mais gerações serão necessárias para que a evolução chegue no seu ponto máximo de evolução. Pode-se observar que não existe uma piora nos melhores indivíduos em nenhum momento, mostrando que as boas características são aproveitadas e propagadas pelo ambiente. Percebe-se, também, uma oscilação nos piores indivíduos, indicando que há variabilidade genética e chances de evolução da população.

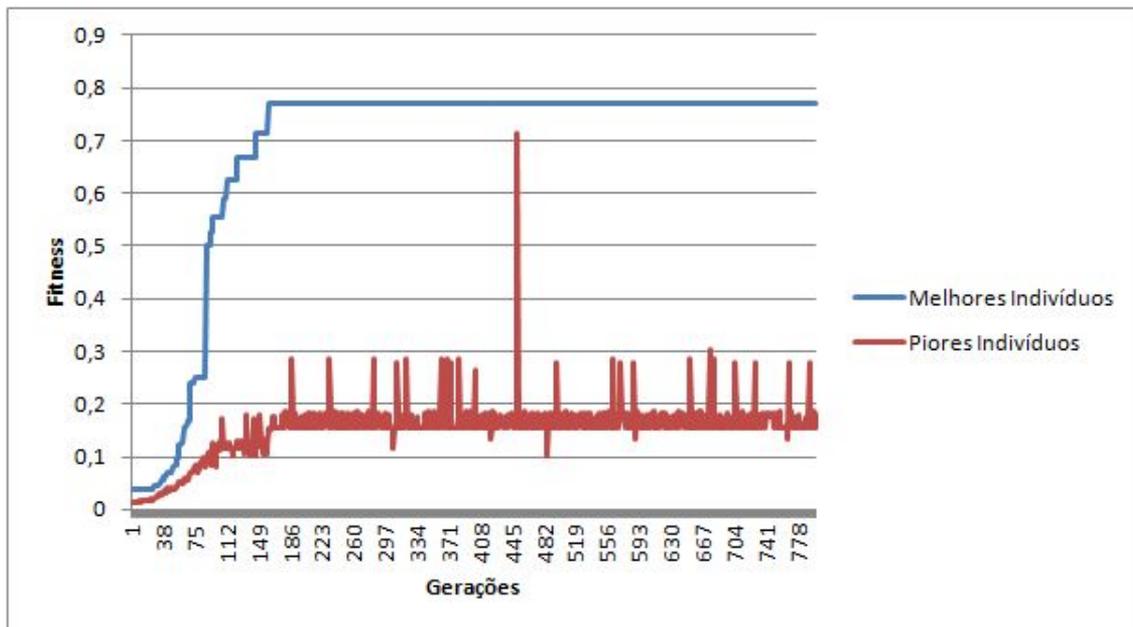
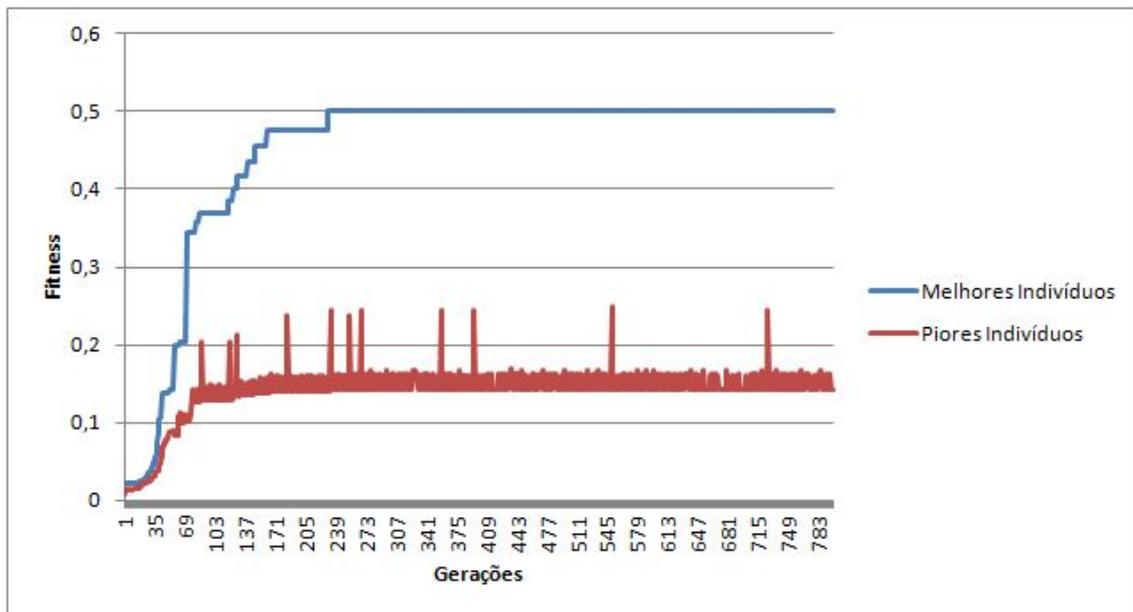


Figura 4. Curva da evolução dos melhores e dos piores indivíduos do grupo de testes 1



**Figura 5. Curva da evolução dos melhores e dos piores indivíduos do grupo de testes 2**

Por fim, foi investigado se as melhores soluções encontradas atendiam as restrições especificadas. Os dados encontrados podem ser vistos na tabela 5 para o grupo de testes 1 e na tabela 6 para o grupo de testes 2. Podemos observar que os resultados alcançados foram bastante positivos. Nos dois casos não ocorreram violações de restrições *hard*. Em nenhuma solução houve choques de horários entre aulas e nem entre as aulas ministradas por cada professor, que também não ensinariam em horários indisponíveis. Todas as disciplinas tiveram a sua quantidade certa de aulas alocadas e nenhuma aula de disciplinas oferecidas por outros departamentos teve seu horário alterado. Mas podemos ver que nem todas as restrições *soft* foram completamente satisfeitas. Na solução do grupo de testes 1, em três ocasiões, mais de uma aula da mesma disciplina foi alocada no mesmo dia. Na solução do grupo de testes 2, seis dias não tiveram aulas alocadas, um horário poderia ser alocado durante a manhã, mas não foi e três disciplinas tiveram mais de uma aula alocadas no mesmo dia.

**Tabela 5. Ocorrências das restrições no melhor indivíduo da melhor configuração do grupo de testes 1**

Restrição	Tipo	Ocorrências
Choques de horários entre aulas	<i>Hard</i>	0
Choques de horários entre as aulas do professor	<i>Hard</i>	0
Aulas alocadas em horários não disponíveis do professor	<i>Hard</i>	0
Quantidade de aulas excedidas de uma mesma disciplina	<i>Hard</i>	0
Horários das aulas das disciplinas ofertadas por outros departamentos alterados	<i>Hard</i>	0
Dias sem aula por período	<i>Soft</i>	0
Horários que poderiam ser ocupados durante a manhã, mas não foram	<i>Soft</i>	0

Mais de uma aula da mesma disciplina sendo ministrada no mesmo dia	<i>Soft</i>	3
Horários das aulas de disciplinas pré-requisitos diferentes	<i>Soft</i>	0

**Tabela 6. Ocorrências das restrições no melhor indivíduo da melhor configuração do grupo de testes 2**

Restrição	Tipo	Ocorrências
Choques de horários entre aulas	<i>Hard</i>	0
Choques de horários entre as aulas do professor	<i>Hard</i>	0
Aulas alocadas em horários não disponíveis do professor	<i>Hard</i>	0
Quantidade de aulas excedidas de uma mesma disciplina	<i>Hard</i>	0
Horários das aulas das disciplinas ofertadas por outros departamentos alterados	<i>Hard</i>	0
Dias sem aula por período	<i>Soft</i>	6
Horários que poderiam ser ocupados durante a manhã, mas não foram	<i>Soft</i>	1
Mais de uma aula da mesma disciplina sendo ministrada no mesmo dia	<i>Soft</i>	3
Horários das aulas de disciplinas pré-requisitos diferentes	<i>Soft</i>	0

A figura 6 ainda mostra um exemplo de um período de uma grade gerada. O exemplo se refere ao primeiro período do curso de BSI, que possuía seis disciplinas e doze aulas que deveriam ser alocadas. As disciplinas de Sociologia e Administração I (ADM I na grade) foram oferecidas por outros departamentos e não houve alteração nos seus horários previamente definidos. É possível perceber o total aproveitamento dos horários matutinos e a variação das aulas durante a semana.

P1	Segunda	Terça	Quarta	Quinta	Sexta
08:00 - 09:40	<b>IC</b> Rafael	<b>Elementar</b> Juliana A.	<b>Metodologia</b> Gabriela	<b>IC</b> Rafael	<b>Elementar</b> Juliana A.
09:50 - 11:30	<b>IP</b> Rodrigo	<b>ADM I</b> Eliane	<b>ADM I</b> Eliane	<b>Metodologia</b> Gabriela	<b>IP</b> Rodrigo
Almoço					
13:10 - 14:50	<b>Sociologia</b> Osicleide				
14:50 - 16:40	<b>Sociologia</b> Osicleide				

**Figura 6. Exemplo de um período de uma grade horária gerada**

## 8. Conclusão

Neste trabalho foi proposta uma solução automática, baseada na técnica de Algoritmos Genéticos, para o problema da geração das grades horárias dos cursos de Bacharelado

em Sistemas de Informação e Licenciatura em Ciência da Computação do campus IV da Universidade Federal da Paraíba.

Os resultados obtidos demonstram a viabilidade da proposta e as análises feitas mostram a evolução dos resultados ao longo das gerações. Dentre as nove configurações diferentes testadas, as que apresentaram os melhores resultados foram as que possuíam as técnicas de seleção por torneio e seleção por truncamento como métodos de seleção. Os resultados também comprovam que é possível gerar as grades horárias dos cursos em menos tempo e de modo bem menos trabalhoso. Apesar dos bons resultados, a solução proposta possui algumas limitações, como a alocação de disciplinas de créditos pares, apenas, e a geração da grade de um curso por vez.

Como sugestões de trabalhos futuros ficam a geração das grades de vários cursos em uma única execução, a utilização de outras técnicas de otimização para compará-las aos Algoritmos Genéticos e a disponibilização deste serviço através da internet.

## Referências

- BERARDINO, D. R. D. *Software para Geração da Grade Horária do Curso de BSI da UNIRIO*. 2013. 52 f. Monografia (Bacharelado em Sistemas de Informação) - Escola de Informática Aplicada da Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2013.
- BORGES, S. K. *Resolução de timetabling utilizando algoritmos genéticos e evolução cooperativa*. 2003. 104 f. Dissertação (Mestrado em Informática) - Departamento de Informática da Universidade Federal do Paraná, Curitiba, 2003.
- CISCON, L. A.; OLIVEIRA, A. C.; HIPÓLITO, T. R.; ALVARENGA, G. B.; ROULLIER, A. C.. O Problema de Geração de Horários: um Foco na Eliminação de Janelas e Aulas Isoladas. In: XXXVII Simpósio Brasileiro de Pesquisa Operacional. 2005.
- FREITAS, Cherze C.; GUIMARÃES, Priscilla R. B.; NETO, Manoel C. M.; BARBOZA, Frederico J. R. Uma Ferramenta Baseada em Algoritmos Genéticos para a Geração de Tabela de Horário Escolar. Disponível em: <[https://www.researchgate.net/publication/265922481\\_Uma\\_Ferramenta\\_Baseada\\_em\\_Algoritmos\\_Geneticos\\_para\\_a\\_Geracao\\_de\\_Tabela\\_de\\_Horario\\_Escolar](https://www.researchgate.net/publication/265922481_Uma_Ferramenta_Baseada_em_Algoritmos_Geneticos_para_a_Geracao_de_Tabela_de_Horario_Escolar)>. Acesso em: 17 jul. 2016.
- FUCILINI, T. P.; C. S. P., MARUANI, E.; REBONATTO, M. T. Timetabling com algoritmos genéticos: resultados, restrições e exploração do paralelismo. *Hifen*, Rio Grande do Sul, v. 32, n. 62, II Semestre. 2008. Disponível em: <<http://revistaseletronicas.pucrs.br/ojs/index.php/hifen/article/view/4594>>. Acesso em: 02 mar. 2015.
- HAMAWAKI, C. D. L. *Geração Automática de Grade Horária Usando Algoritmos Genéticos: O Caso da Faculdade de Engenharia Elétrica da UFU*. 2005. 104 f. Dissertação (Mestrado em Engenharia Elétrica) - Departamento de Engenharia Elétrica da Universidade Federal de Uberlândia, Uberlândia, 2005.

- LIMA, M. R. C. *Algoritmos Genéticos na Formação de Grupos Para Aprendizagem Cooperativa Apoiada por Computador*. 2006. 101 f. Dissertação (Mestrado em Engenharia Elétrica na área de Ciência da Computação) - Universidade Federal do Maranhão, São Luís, 2006.
- LUCAS, D. C. Algoritmos Genéticos: uma Introdução. Disponível em: <<http://www.inf.ufrgs.br/~alvares/INF01048IA/ApostilaAlgoritmosGeneticos.pdf>>. Acesso em: 17 jul. 2016.
- NAVARRO, F.; COELHO, F. Estudo de Coloração Aplicado ao Problema de Alocação de Horário de Professores. Disponível em: <<http://www.unipac.br/site/bb/tcc/tcc-5486411b6e467ad999a2a853d986d947.pdf>>. Acesso em: 17 jul. 2016.
- RAMOS, P. S. *Sistema Automático de Geração de Horários para a UFLA utilizando Algoritmos Genéticos*. 2002. 87 f. Monografia (Bacharelado em Ciência da Computação) - Departamento de Ciência da Computação da Universidade Federal de Lavras, Lavras, 2002.
- SANTOS, C. N.; SANTOS, Roberto Jefferson da Silva. Implementação de um Algoritmo Genético Para Construção Automática de Horários em Uma Escola de Ensino Fundamental e Médio. In: VII Simpósio de Pesquisa Operacional e Logística da Marinha. 2004.
- SIMÃO, T. D. *Utilização de Algoritmos Genéticos para otimização de soluções para o timetabling escolar*. 2013. 72 f. Monografia (Bacharelado em Ciência da Computação) - Departamento de Ciência da Computação da Universidade Federal de Lavras, Lavras, 2013.
- VIEIRA, F.; MACEDO H. Sistema de Alocação de Horários de Cursos Universitários: Um Estudo de Caso no Departamento de Computação da Universidade Federal de Sergipe. *Scientia Plena*, Sergipe, v. 7, n. 3, mar. 2011. Disponível em: <<https://www.scientiaplenu.org.br/sp/article/view/163/119>>. Acesso em: 17 jul. 2016.