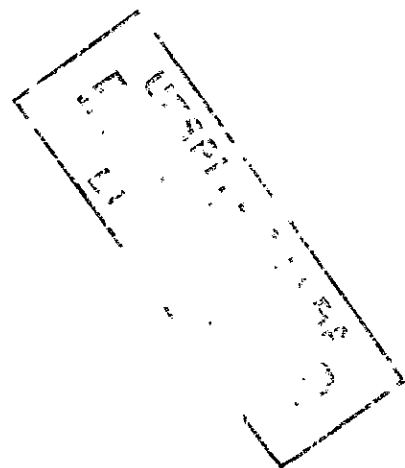
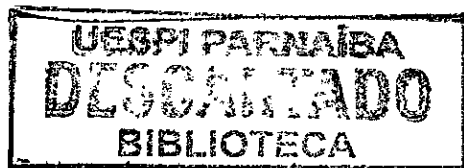


UNIVERSIDADE ESTADUAL DO PIAUÍ
CAMPUS PROFº ALEXANDRE ALVES DE OLIVEIRA
BACHARELADO EM CIÊNCIAS DA COMPUTAÇÃO

ANDERSON PASSOS DE ARAGÃO

UTILIZAÇÃO DE ALGORÍTMOS GENÉTICOS PARA A
ELABORAÇÃO DE UM PROGRAMA GERADOR DE GRADE
HORÁRIA.



PARNAÍBA – PI

2010

ANDERSON PASSOS DE ARAGÃO

**UTILIZAÇÃO DE ALGORITMOS GENÉTICOS PARA A
ELABORAÇÃO DE UM PROGRAMA GERADOR DE GRADE
HORÁRIA.**

Monografia apresentada ao curso de Bacharelado em Ciências da Computação da Universidade Estadual do Piauí – UESPI, Campus Prof. Alexandre Alves de Oliveira, como parte das exigências da disciplina de Estágio Supervisionado; requisito parcial para obtenção do título de Bacharel em Computação.

Orientador: Alessandro Saraiva Mendonça.

**Parnaíba-PI
2010**

FICHA CATALOGRÁFICA ELABORADA PELO BIBLIOTECÁRIO
HERNANDES ANDRADE SILVA CRB-3/936

A659u Aragão, Anderson Passos de

Utilização de algoritmos genéticos para a elaboração de um
programa gerador de grade horária / Anderson Passos de Aragão.
– Parnaíba: 2010.

66f : il.

Monografia apresentada ao Curso de Bacharelado em
Ciências da Computação, Universidade Estadual do Piauí -
UESPI, Parnaíba - 2010.

Orientador: Prof. Alessandro Saraiva Mendonça.

1. Algoritmos – Programação. 2. Algoritmo Genético. 3.
Programa (Computador). I. Título.

CDD – 005.1



Ata de Apresentação de Trabalho de Conclusão de Curso

Aos dez dias do mês de setembro de dois mil e dez, às 15h00, na Sala 201 do Campus Prof. Alexandre Alves Oliveira - UESPI, reuniram-se a Banca Examinadora – presidida pelo Prof. Alessandro Saraiva Mendonça e composta pelos membros efetivos os professores José Flávio Gomes Barros e Nécio de Lima Veras – e o aluno Anderson Passos de Aragão para a apresentação do Trabalho de Conclusão de Curso intitulado **Utilização de Algoritmos Genéticos para a Elaboração de um Programa Gerador de Grade Horária**, como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciência da Computação. Aberta a sessão pública, o candidato teve a oportunidade de expor o trabalho. Após a exposição, o aluno foi arguido oralmente pelos membros da Banca Examinadora. Em seguida, em sessão reservada e nos termos do Regulamento Geral dos Trabalhos de Conclusão de Curso da UESPI, os membros da Banca realizaram a avaliação do candidato, que obteve nota 9,0 (nove) sendo, portanto, **aprovado**. Nada mais havendo a tratar, eu, Prof. Alessandro Saraiva Mendonça, lavrei a presente Ata que será lida e assinada por mim, pelos membros da Banca Examinadora e pelo candidato. Parnaíba (PI), 10 de setembro de 2010.

Banca Examinadora


Prof. Esp Alessandro Saraiva Mendonça, UESPI
Orientador


Prof. Esp José Flávio Gomes Barros, UESPI


Prof. Esp Nécio de Lima Veras, UESPI

Candidato


Anderson Passos de Aragão

Aos meus pais, irmãos, tios, avós e amigos por todo incentivo, compreensão e apoio ao longo, não só desta caminhada, mais de toda a minha vida.

AGRADECIMENTOS

- Agradeço primeiramente a Deus pela força dada a mim para a conclusão desta longa caminhada;
- Aos meus pais Antonio Pereira e Rosa Passos e irmãos Helio e Sidnei Passos pelo amor, carinho, dedicação, incentivo, educação e apoio dedicados a mim ao longo de toda a minha vida;
- A minha avó paterna Maria Aragão *in memoriam*, e meus avôs maternos Joaquim e Francisca Passos pelo apoio, incentivo e amor;
- Aos meus tios, em especial a Regina Monteiro, Manoel Monteiro, Marcelo Pereira, Madalena Mendes, Sonia Passos e Celene Passos vocês me deram a base para esta conquista e acreditaram em mim. Meu muito obrigado;
- A minha namorada Caroline Mauriz por acreditar em mim, pela grande ajuda, apoio, companheirismo, amizade e amor;
- Aos meus amigos em especial a Alessandra Braúna, Conrado Machado, Danilo Borges, Josué Machado e Washington Costa pela grande ajuda, colaboração e parceria.
- Aos meus primos pela ajuda e apoio;
- Ao meu Orientador Prof^o Alessandro Saraiva Mendonça pela orientação e confiança;
- A todos os professores do curso de Bacharelado em Ciências da Computação da Universidade Estadual do Piauí;
- Aos amigos e colegas de turma pela convivência e aprendizado.

RESUMO

Um dos trabalhos árduo envolvendo diretamente os corpos discente, docente e funcionários de uma instituição de ensino superior é a elaboração da grade horária curricular. Isto é realizado pelas coordenações dos cursos da mesma a cada semestre e esta rotina consiste em alocar em um conjunto de intervalos de tempo, aulas de disciplinas de um curso, buscando satisfazer as preferências dos envolvidos e minimizando as restrições associadas a geração da grade horária escolar. O presente trabalho mostra o que são os algoritmos genéticos, a sua estrutura e o seu funcionamento para a construção de sistema gerador de grade horária para o curso de Bacharelado em Ciências da Computação da Universidade Estadual do Piauí – UESPI Campus Professor Alexandre Alves de Oliveira mostrando a complexidade dos problemas encontrados para a confecção de uma solução de qualidade, pois existem múltiplos objetivos, múltiplas restrições e um número grande de variáveis a serem analisadas e avaliadas. Este também descreve a problemática do estudo de caso e mostra os detalhes da implementação de cada estrutura do algoritmo juntamente com a análise dos resultados obtidos. O objetivo principal deste trabalho é desenvolver uma representação genética para o problema de geração de horário, utilizando algoritmos genéticos para alcançar uma solução viável para o problema, em relação à aplicação baseada no estudo de caso.

Palavras-chave: algoritmos genéticos. grade horária. sistema gerador de grade horária.

ABSTRACT

One of the hard work directly involving the student body, faculty and staff of an institution of higher education is to prepare the grid hourly curriculum. This is accomplished by coordination of the same courses each semester and this routine is to allocate a set of time intervals, lessons on subjects of a course, trying to satisfy the preferences of those involved and minimizing the restrictions associated with the generation of grid hourly. This paper shows that genetic algorithms are, their structure and functioning system for building grid hour generator for the course in Bachelor of Computer Science State University of Piauí - UESPI Campus Professor Alexandre Alves de Oliveira shows the complexity of the problems encountered for making a quality solution, because there are multiple objectives, multiple constraints and a large number of variables to be analyzed and evaluated. This also describes the problems of case study and shows the implementation details of each structure of the algorithm along with analysis of results. The main objective of this paper is to develop a genetic representation for the problem of generation scheduling using genetic algorithms to achieve a viable solution to the problem regarding the application based on case study.

Key-words: genetic algorithms, grid hourly, grid hourly system generator.

LISTA DE FIGURAS

Figura 1: Diagrama representando o funcionamento de um GA básico.	16
Figura 2: Pontos de Corte. As setas 1, 2, 3 e 4 representam pontos de corte de um cromossomo com 5 genes.	20
Figura 3: A figura a seguir mostra como o processo de crossover é semelhante ao que ocorre na natureza. O tipo de reprodução ao qual o crossover esta fazendo analogia corresponde à reprodução sexuada.	20
Figura 4: Crossover de dois pontos.	21
Figura 5: Crossover uniforme.	21
Figura 6: Exemplo de Mutação em um GA com representação cromossomial binária.	22
Figura 7: Tabela exemplando a grade Horária do 8º período – Bacharelado em Ciências da Computação Campus Profº Alexandre Alves de Oliveira.	28
Figura 8: Consulta efetuada no Banco de Dados do PEGH mostrando as disciplinas lecionadas pelo professor Alessandro Saraiva no semestre letivo 2010.1.	33
Figura 9: Consulta efetuada no Banco de Dados do PEGH mostrando uma lista com alguns professores cadastrados.	35
Figura 10: Tabela de Disponibilidade dos Professores dentro da Base de Dados do PEGH.	36
Figura 11: Tabela de Disciplinas utilizada pelo PEGH.	37
Figura 12: Matriz Grade aqui as colunas com os números de 1 a 12 representam os slots de tempo e as linhas representam os períodos.	39
Figura 13: Matriz Disciplinas as colunas contém os números que correspondem aos identificadores das disciplinas e as linhas representam os períodos da grade.	39
Figura 14: Matriz Professor que guarda o identificador do professor para a disciplina alocada na mesma posição da matriz disciplina.	40
Figura 15: Matriz Turno nesta matriz são mostrados os turnos referentes aos períodos. O valor 0 representa o turno vespertino e o valor 1 representa o turno noturno.	40
Figura 16: Ilustração de como ocorre a associação entre as matrizes para a codificação do indivíduo.	41
Figura 17: Roleta e a a parte de cada indivíduo dentro da população de acordo com sua avaliação.	44
Figura 18: Exemplo de como ocorre o crossover.	45

Figura 19: A mutação no PEGH.	47
Figura 20: Tela Inicial do PEGH – Programa de Escalonamento de Grade Horária.	49
Figura 21: Tela de Configuração dos parâmetros do GA.	50
Figura 22: Tela de Vinculação de Professor a Disciplina.	51
Figura 23: Tela de Cadastro de professor onde somente o nome do docente é preenchido pelo usuário. O PEGH gerará um valor para o id do docente.	51
Figura 24: Cadastro de Disciplinas, aqui o usuário cadastra as principais informações relativas a disciplina para a geração da grade horária.	52
Figura 25: Tela relativa ao cadastro das disponibilidades dos professores.	53
Figura 26: Tela de ajuda que mostrando como gerar uma nova grade horária.	54
Figura 27: Tabela de Comparativo de Desempenho.	55
Figura 28: Resultado da Execução do PEGH. Tabela referente ao 1º período.	56
Figura 29: Resultado da Execução do PEGH. Tabela referente ao 3º período.	56
Figura 30: Resultado da Execução do PEGH. Tabela referente ao 5º período.	56
Figura 31: Resultado da Execução do PEGH. Tabela referente ao 8º período.	56

SUMÁRIO

1 INTRODUÇÃO	11
2 ALGORITMOS GENÉTICOS	13
2.1 APRESENTAÇÃO	13
2.2 CARACTERÍSTICAS GERAIS	14
2.3 FUNCIONAMENTO	15
2.3.1 Codificação Genética	16
2.3.2 Inicialização	17
2.3.3 Avaliação	18
2.3.4 Seleção de Pais	18
2.3.5 Reprodução	19
2.3.5.1 O operador Genético Crossover	19
2.3.5.2 O operador Genético Mutação	22
2.3.6 Atualização	22
2.3.7 Finalização	23
3 PROBLEMÁTICA DE OTIMIZAÇÃO COMBINATORIAL E O PROBLEMA DA ELABORAÇÃO DE GRADES HORÁRIAS ESCOLARES	25
3.1 OTIMIZAÇÃO COMBINATORIAL	25
3.2 ORDEM DE GRANDEZA DA COMPLEXIDADE DO PROBLEMA	26
3.3 DEFINIÇÃO DO PROBLEMA	27
4 ESTUDO DE CASO: GERAÇÃO DA GRADE HORÁRIA PARA O CURSO DE CIÊNCIAS DA COMPUTAÇÃO DA UNIVERSIDADE ESTADUAL DO PIAUÍ – UESPI CAMPUS PROF. ALEXANDRE ALVES DE OLIVEIRA	29
4.1 ESTRUTURA	29
5 IMPLEMENTAÇÃO DO PEGH (PROGRAMA DE ESCALONAMENTO DE GRADE HORÁRIA)	31
5.1 INFORMAÇÕES RELATIVAS À BASE DE DADOS UTILIZADA PELO PEGH .	34
5.1.1 Tabela de Professores	35
5.1.2 Tabela de Disponibilidade dos Professores	36
5.1.3 Tabela de Disciplinas	37
5.2 O ALGORITMO GENÉTICO	37
5.2.1 Codificação Genética	38
5.2.1.1 A matriz Grade	39
5.2.1.2 A matriz Disciplina	39
5.2.1.3 A matriz Professor	40
5.2.1.4 A matriz Turno	40
5.2.2 Iniciação de População	41
5.2.3 A Função de Avaliação	42
5.2.3.1 Disponibilidade do Professor	42
5.2.3.2 Outras Penalizações	42
5.2.3.3 Cálculo total da Função de Avaliação	43
5.2.4 Método de Seleção: Roleta	43
5.2.5 Operadores Genéticos	45
5.2.5.1 Crossover (Cruzamento)	45
5.2.5.2 Mutação	46

6 ANÁLISE DE RESULTADOS	48
6.1 PLATAFORMAS DE HARDWARE E SOFTWARE	48
6.2 O APLICATIVO	49
6.3 ANÁLISE E RESULTADOS OBTIDOS	54
7 CONCLUSÃO E TRABALHOS FUTUROS	57
REFERÊNCIAS BIBLIOGRÁFICAS	58
ANEXO A: GRADE CURRICULAR DO CURSO DE BACHARELADO EM CIÊNCIAS DA COMPUTAÇÃO, DO SEMESTRE 2010.1.	60
APÊNDICE A: DIAGRAMA DE CASOS DE USO.	62
APÊNDICE B: DIAGRAMA DE CLASSES	63
APÊNDICE C: DIAGRAMA DE SEQUÊNCIA.	64

1 INTRODUÇÃO

Em qualquer instituição de ensino é adotada uma forma de dividir os horários das aulas ao longo da semana para que alunos e professores possam saber em que dia e em que horário do mesmo será ministrada uma aula. Essa forma de alocação é conhecida por grade horária e, uma vez criada, ela será adotada ao longo de todo o período letivo fazendo com que discentes e docentes tenham que se adequar a ela. A criação desta muitas vezes é um processo demorado e trabalhoso, principalmente se realizado manualmente, devido a grande quantidade de restrições a serem obedecidas, bem como, a preferência dos professores que precisa ser maximizada e a quantidade de variáveis envolvidas.

Este é conhecido por *class-teacher timetabling problem*. É um problema extremamente interessante e complexo onde cada período todas as coordenadas de uma instituição de ensino superior precisam definir os horários das disciplinas de acordo com as preferências dos professores, e uma série de restrições possíveis que podem ser de gerais, como disciplinas diferentes sendo lecionadas pelo mesmo professor não podem estar alocadas em um mesmo horário, e restrições específicas associadas a um estudo de caso, como a de que dois professores não podem estar lecionando em duas turmas ao mesmo tempo. Assim o espaço de busca é muito grande visto que o número de combinações turma(período)/professor/horário é proporcional ao fatorial do número de combinações. Porém quanto mais impomos restrições, sejam elas obrigatórias ou desejáveis, mais aumentamos o espaço de busca e mais complexo se torna o problema.

Várias formas de definir e solucionar esta problemática tem sido elaboradas, porém não existe nenhuma solução geral para ser aplicado nas instituições de ensino, pois para cada problema faz-se necessário estudar o tipo de escola e o seu sistema educacional. Isso para que se possa definir as preferências dos professores e as restrições a serem levadas em consideração para a elaboração da grade horária. (Hamawaki 2005 p.1) .

Este trabalho tem por objetivo principal criar um programa para o problema de geração automática de horário para o curso de Bacharelado em Ciências da Computação da Universidade Estadual – Campus Profº Alexandre Alves de Oliveira utilizando técnicas referentes a teoria dos algoritmos genéticos .

Nos capítulos à seguir serão mostrados os principais conceitos a respeito dos algoritmos genéticos, maiores detalhes sobre a problemática e sobre o estudo de caso proposto, será mostrado também os principais detalhes da implementação e análise dos

resultados obtidos.

O capítulo 2 aborda o histórico, características, definição e funcionamento dos algoritmos genéticos base para a compreensão e entendimento de toda a estrutura do sistema proposto.

O capítulo 3 define o que é a problemática da otimização combinatorial e descreve o problema de geração de grade horária mostrando os aspectos importantes do problema.

No capítulo 4 detalha o estudo de caso proposto, ou seja, a geração de grade horária para o curso de Bacharelado em Ciências da Computação da Universidade Estadual do Piauí – UESPI Campus Prof^o Alexandre Alves de Oliveira.

A implementação do Programa de Escalonamento de Grade Horária é descrita no capítulo 5, mostrando as restrições consideradas no estudo de caso, informações relativas a base de dados utilizada pelo sistema descrevendo o banco de dados e as tabelas utilizadas pelo PEGH e os principais detalhes da implementação do algoritmo genético utilizado pelo sistema, codificação genética, inicialização da população, função de avaliação, seleção dos pais, operadores genéticos (mutação e crossover) e finalização.

O capítulo 6 aborda a análise dos resultados obtidos, a plataforma de hardware e software, e o funcionamento do PEGH mostrando as telas do programa e suas funcionalidades.

A conclusão e trabalhos futuros esta definida no capítulo 7.

Como complemento dos capítulos resumidos acima são definidos os apêndices A, B e C mostrando os diagramas de clases, sequência e de casos de usos. Também segue no anexo A a grade horária do Curso de Bacharelado em Ciências da Computação UESPI campus Prof^o Alexandre Alves de Oliveira do período letivo 2010.1.

2 ALGORITMOS GENÉTICOS

A história dos algoritmos genéticos (GAs) começa na década de 40, pois foi nesta época que os cientistas inspirados no funcionamento e comportamento da natureza começaram a criar e desenvolver a inteligência artificial. O foco de estudo deles eram os ramos da pesquisa cognitiva no processo de raciocínio e no aprendizado. Porém foi no final da década de 50 que os pesquisadores buscavam modelos de sistemas genéticos, sistemas estes que pudessem gerar soluções para problemas computacionais muito complexos e que requisitavam de uma enorme quantidade de tempo e processamento (Linden 2008 p.34).

A primeira tentativa propriamente dita de associação entre evolução natural e problemas de otimização foi feita por Box em 1957. Este método era uma forma bem rudimentar de mutação que conhecemos hoje. No início da década de 60 Bledsoe e Bremmerman começaram estudos com genes desenvolvendo um modelo também rústico de recombinação (crossover).

Na história dos GAs Rechenberg pode ser considerado o pioneiro por ter introduzido a computação evolucionária em muitas aplicações práticas. Este fato ocorreu em meados de 60 com a criação das estratégias evolucionárias (*evolutionary strategies*) que consistia em uma população de indivíduos com cromossomos reais, onde um dos dois era filho do outro e era gerado por mutação. Em seu trabalho Rechenberg não incluiu crossover na população.

Rechenberg é o pioneiro na criação dos algoritmos genéticos, porém o homem considerado o pai dos GAs é conhecido por John Holland (Linden 2008 p. 34)

Holland tem essa denominação devido ao fato de ele ser o primeiro a propor um modelo heurístico computacional baseado em seus estudos sobre a evolução das espécies para a solução de problemas extremamente difíceis e insolúveis até aquela época. Em 1975 Holland publicou seu livro *Adaptation in Natural and Artificial Systems*, este livro contém estudos sobre os processos evolutivos, apresentando os GAs como uma metáfora para processos evolutivos de forma que ele pudesse estudar a adaptação e a evolução no mundo real, simulando-a no meio computacional e disseminando seu uso.

2.1 APRESENTAÇÃO

Os algoritmos genéticos representam um ramo da Inteligência Artificial e constituem uma série de técnicas de busca baseadas no princípio da seleção natural das espécies proposto

por Charles Darwin em seu livro *On the Origin of Species by Means of Natural Selection* (1859). Os GAs tem como patrono John Holland que em meados de 70 publicou seu livro *Adaptation in Natural and Artificial Systems*, considerado até hoje o manual de referência para a compreensão e desenvolvimento dos algoritmos genéticos.

Nos GAs é criada uma população de indivíduos que representam o conjunto de possíveis soluções do problema proposto, indivíduos estes que são codificados e competirão entre si pela reprodução. Assim os indivíduos que possuem um melhor resultado dentro de uma função de avaliação, terão uma maior chance de poder proliferar suas características, aumentando a qualidade dos indivíduos da próxima geração desenvolvendo assim uma heurística de busca.

Vale lembrar que os algoritmos genéticos não constituem métodos de busca de soluções ótimas, visto que a seleção natural ao qual o algoritmo se baseia consiste na competição de indivíduos pela perpetuação da espécie onde sobrevivem os mais aptos, neste contexto logicamente que os melhores tendem a sobreviver e reproduzir gerando filhos com avaliações possivelmente melhores, não impedindo em determinadas situações que indivíduos considerados menos aptos possam reproduzir.

GAs são técnicas probabilísticas e não técnicas determinísticas. Assim sendo, um GA com a mesma população inicial e o mesmo conjunto de parâmetros pode encontrar soluções diferentes a cada vez que é executado (Linden 2008 p.48),

Um dos pontos fortes que podemos ressaltar em relação aos GAs são o seu alto grau de adaptabilidade, onde um mesmo algoritmo genético pode ser utilizado em uma vasta gama de problemas bastando para isso que se mude apenas a função de avaliação e a codificação dos indivíduos caso necessário. Porém é importante ressaltar que um algoritmo criado especificamente para a resolução de um problema terá um melhor resultado que um algoritmo adaptado (genérico).

Os algoritmos genéticos são muito utilizados em problemas cujo espaço de busca é incrivelmente grande, exigindo um grande recurso de processamento, o que torna a resolução destes lenta ou incapaz de ser resolvido, lembrando que o tempo que se gasta para chegar a uma solução é um ponto crítico na avaliação da solução.

2.2 - CARACTERÍSTICAS GERAIS

Dentre as Principais características dos GAs destacam-se:

- a) Busca codificada: Segundo Lucas (2000 p.12) na busca codificada não utilizamos o elemento da população para ser avaliado na função de avaliação, mas sim uma representação deste indivíduo chamada de cromossomo. Os cromossomos são compostos por um conjunto de genes. Assim os GAs não trabalham no domínio do problema propriamente dito, é escolhido um método para a codificação da população para a representação de cada indivíduo para que este possa ser avaliado.
- b) Generalidade: Os GAs possuem um alto grau de adaptabilidade, este atributo se dá devido ao fato de que as únicas partes que requisitam de conhecimentos específicos do problema proposto é a representação (forma como serão apresentados cada indivíduo) e a função de avaliação, ou seja, como a representação dos dados será avaliada para que sejam escolhidos os melhores indivíduos da população. Com a alteração destas poderemos resolver outros problemas cujo uso do GA seria indicado. Logo um mesmo GA pode ser utilizado para resolver uma infinidade de problemas, bastando para isso que se mude tanto a representação quanto a função de avaliação (Lucas 2000 p.12).
- c) Busca estocástica: Igualmente ao mecanismo de seleção natural os GAs escolhem de forma aleatória o conjunto de elementos a serem avaliados, porém algoritmos genéticos não podem ser chamados de busca aleatória porque eles utilizam informações armazenadas relativas aos indivíduos para escolher a nova população.
- d) Paralelismo: Esta característica é devido ao fato de que os GAs mantêm uma população de soluções que são avaliadas simultaneamente (Linden 008 p.48, Goldberg 1989 p.56).
- e) Capaz de lidar com uma grande variedade de problemas: Segundo Linden (2008 p. 53), os GAs são capazes de lidar com funções reais, discretas, booleanas, e até mesmo categóricas (não-numéricas), sendo possível misturar as representações sem prejuízo na habilidade dos GAs para resolver problemas.

2.3 - FUNCIONAMENTO

Os algoritmos genéticos funcionam basicamente da seguinte maneira: A população é inicializada e sofre a atuação dos operadores genéticos que são a seleção, reprodução e mutação. Esse processo é realizado a cada nova geração até que se tenha alcançado um ponto de parada finalizando o GA e exibindo os resultados obtidos. A estrutura de seu funcionamento é mostrada à seguir:



Figura 01: Diagrama representando o funcionamento de um GA básico.

Fonte: Algoritmos Genéticos: um estudo de seus conceitos fundamentais e aplicação no problema de grade horária Lucas (2000). Disponível em: <www.ufpel.tche.br/prg/sisbi/bibct/acervo/info/2000/Mono-Diogo.pdf>.

2.3.1 Codificação Genética

A codificação genética é essencial em todo e qualquer algoritmo genético, isso porque, ela é a maneira básica de traduzir a informação do problema proposto em uma maneira viável

de tratamento pelo computador. Esta codificação é geralmente conhecida por cromossomo ou representação cromossomial é uma representação da possível solução do problema. Linden (2008 p.59) afirma que quanto mais ela for adequada ao mesmo, maior a qualidade dos resultados obtidos. A representação cromossomial segundo Goldberg (1989 p.65) pode ser definida da maneira arbitrária não havendo um tipo pré-determinado, embora muitos pesquisadores adotem a representação binária por sua simplicidade. Assim a representação a ser adotada depende do problema e deve ser escolhido a que mais se adequar ao mesmo, sendo assim Linden (2008 p.59) define algumas regras são adotadas entre elas:

- A representação deve ser a mais simples possível;
- Se houver soluções inválidas, então elas não devem possuir uma representação;
- Se o problema impuser algum tipo de condição, elas devem estar implícitas dentro da nossa representação;

2.3.2 – Inicialização

Segundo Reeves (2003 p.37) duas questões referentes a inicialização da população devem ser consideradas, a primeira é o seu tamanho e, em seguida, vem o método de como serão escolhidos os indivíduos. A escolha do tamanho da população deve ser considerada em relação a vários pontos de vista, e deve ser balanceada entre eficiência e eficácia. Agora dependendo da forma como será escolhida a população Linden (2008 p.65) afirma que a estratégia mais simples e mais utilizada na área de algoritmos genéticos é a inicialização dos indivíduos de forma aleatória, a escolha aleatória gera uma boa distribuição das soluções no espaço de busca, chamada em comparação com a natureza, de biodiversidade, fundamental para que se possa garantir uma busca bem mais abrangente no espaço de pesquisa. Vale ressaltar que cada um dos indivíduos criados constituem as possíveis soluções do problema proposto.

Goldberg (1989 p.78) define dois tipos comuns de inicialização, são elas:

- Inicialização randômica uniforme: onde cada gene do indivíduo receberá como valor um elemento do conjunto de alelos (conjunto de possíveis valores que um gene pode possuir), sorteado de forma aleatoriamente uniforme.
- Inicialização randômica não uniforme: aqui possíveis valores a que serão armazenados no gene tendem a ser escolhidos com uma maior frequência do que o

restante.

2.3.3 Avaliação

Nesta etapa o universo da população é avaliado. Em comparação a uma escola, por exemplo, cada indivíduo faz uma prova e assim como na escola, esta avaliação visa saber quão bom é um determinado indivíduo visto que cada indivíduo é um possível candidato a solução do problema proposto. Vale lembrar também que esta etapa é o ponto do algoritmo mais dependente do problema em si (Linden 2008 p.65) e faz-se necessário embutir o máximo de conhecimento do problema, onde deve ser inserido tanto restrições quanto seus objetivos de qualidade. Atualmente existem muitas formas de se fazer esta avaliação: caso a otimização seja de funções matemáticas, o próprio valor da imagem é o valor a ser escolhido como valor de retorno. A função de avaliação também é conhecida como função objetivo.

2.3.4 Seleção de Pais

Segundo a teoria da evolução das espécies de Darwin no mecanismo de seleção natural os pais mais desenvolvidos ou melhores adaptados são capazes de gerarem mais filhos, porém este fato não exclui que indivíduos menos adaptados também possam gerar descendentes. Um GA deve ser capaz de simular este mecanismo natural tal qual ocorre na natureza de forma que os indivíduos com função de avaliação alta sejam mais valorizados sem, contudo, desprezar os indivíduos com função de avaliação baixa. Não podemos desprezar os indivíduos com avaliação baixa, porque se considerarmos somente aqueles de avaliação boa, quando aplicarmos estes indivíduos aos operadores genéticos, estaremos nos encaminhado a uma convergência genética¹. Assim Linden (2008 p.46) afirma que não devemos ser muito seletivos só selecionando para pais indivíduos com excelente avaliação, pois poderemos estar correndo o risco de jogar fora bons esquemas presentes em indivíduos considerados ruins por possuírem uma avaliação baixa, pois assim estaremos comprometendo a evolução.

São alguns dos métodos utilizados para seleção dos pais:

- Seleção por Roleta Viciada: Neste tipo de seleção uma é criada uma roleta que corresponde à soma total das avaliações dos cromossomos, onde cada um desses

¹ Convergência Genética se traduz em uma população com baixa diversidade genética que, por possuir genes similares, não consegue evoluir, a não ser pela ocorrência de mutação.

recebe uma parte proporcional a sua avaliação em relação à soma total das avaliações de todos os cromossomos da população.

Este método seleciona um valor aleatório n compreendido no intervalo $0 < n <$ soma total das avaliações. Assim a roleta consegue selecionar indivíduos com qualquer tipo de avaliação privilegiando os indivíduos com maior avaliação (Linden 2008 p. 68, Lucas p. 17)

- Seleção por Torneio: Este método realiza uma competição entre alguns indivíduos da população e o que apresentar a melhor avaliação é selecionado. Neste método é usado o parâmetro **Tamanho do Torneio** que define quantos indivíduos participaram desta disputa (Linden 2008 p. 171, Lucas 200 p.17).

Aqui os indivíduos são escolhidos de forma totalmente aleatória e não há nenhum favorecimento ao melhor indivíduo.

- Seleção Uniforme: Neste tipo de seleção, assim como a seleção por torneio, todos os indivíduos possuem a mesma probabilidade de serem selecionados. Esse tipo de seleção não é muito utilizado, pois a evolução é muito demorada ou muitas vezes inexistente.

2.3.5 Reprodução

Depois da etapa de seleção dos pais o próximo passo é a reprodução onde nela atuarão os operadores genéticos crossover e mutação para a geração de novos indivíduos que serão posteriormente inseridos dentro da população.

2.3.5.1 O operador Genético Crossover

O crossover (cruzamento) é o responsável pela combinação dos genes dos pais. Genes são partes dos cromossomos, estes que correspondem à representação do indivíduo da população que, neste caso, são os pais. Existem vários tipos de crossover dentre eles estão:

- O crossover de um ponto (1x): este representa o operador mais simples e seu funcionamento ocorre segundo Reeves (2002 p. 50), da seguinte maneira:

Depois da seleção dos pais, selecionamos um ponto de corte que corresponde a uma posição, dentro do cromossomo entre dois genes, este ponto de corte é um numero

entre $m < p < \text{length}$, onde $m > 1$, p é o ponto de corte e length é o tamanho do cromossomo. Por exemplo, para um indivíduo que contém n genes, o número de pontos de corte corresponde a $n - 1$. Na figura 2 podemos ver um exemplo de pontos de corte.

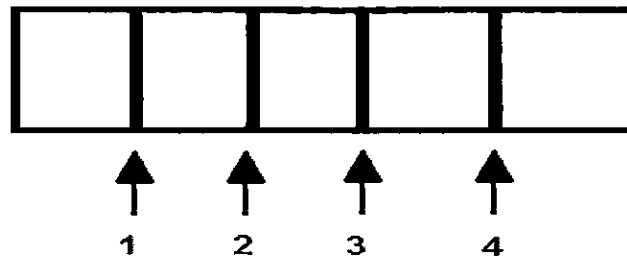


Figura 2: Pontos de Corte. As setas 1, 2, 3 e 4 representam pontos de corte de um cromossomo com 5 genes.
Fonte: Primária

Logo após o sorteio do ponto de corte, os pais têm sua representação (cromossomo) separada. Uma a esquerda do ponto de corte e outra a direita. Assim o primeiro filho é resultado da concatenação da parte esquerda do primeiro pai com a parte direita do outro pai. Logicamente o segundo filho é composto das outras partes restantes. A seguir um exemplo de crossover de um ponto.

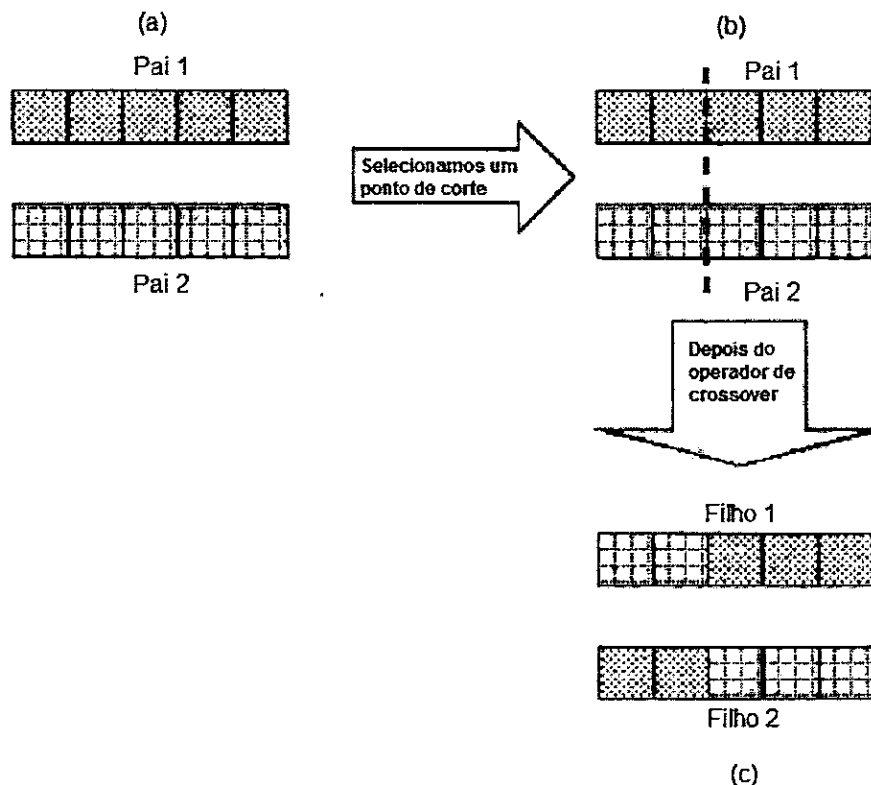


Figura 3: A figura a seguir mostra como o processo de crossover é semelhante ao que ocorre na natureza. Vale

lembrar que o tipo de reprodução ao qual o crossover esta fazendo a analogia corresponde à reprodução sexuada.
 Fonte: Algoritmos Genéticos: Uma importante ferramenta da Inteligência Computacional. Linden (2008).

- O crossover de dois pontos (2x): este operador funciona de forma semelhante ao crossover de um ponto tendo como grande diferencial a escolha de dois pontos de corte. Neste caso os filhos são gerados da seguinte maneira: o primeiro filho recebe as partes fora dos pontos de corte do primeiro pai e do segundo pai é recebida as partes dentro do ponto de corte. Com o segundo filho ocorre o inverso, ou seja, o segundo filho recebe as partes fora dos pontos de corte do segundo pai e do primeiro pai, recebe a parte dentro dos pontos de corte. A figura 4 a seguir tem-se um exemplo de crossover de dois pontos.

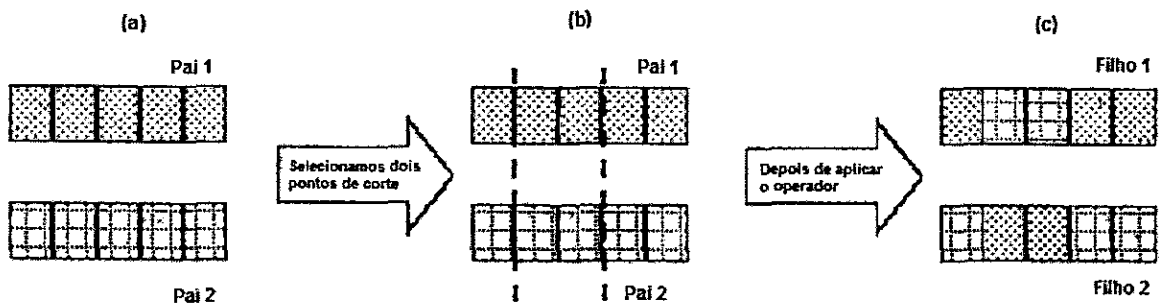


Figura 4: Crossover de dois pontos.

Fonte: Algoritmos Genéticos: Uma importante ferramenta da Inteligência Computacional. Linden (2008).

- O crossover uniforme: aqui todos os genes têm o sorteio de um número zero ou um. Caso o número sorteado for igual a um, o primeiro filho recebe o gene da posição atual do primeiro pai e o segundo filho recebe o gene da posição atual do segundo pai. Em contrapartida se o número sorteado for zero, o processo ocorre de forma inversa onde o primeiro filho recebe o gene atual do segundo pai e o segundo filho recebe a posição atual do primeiro pai.

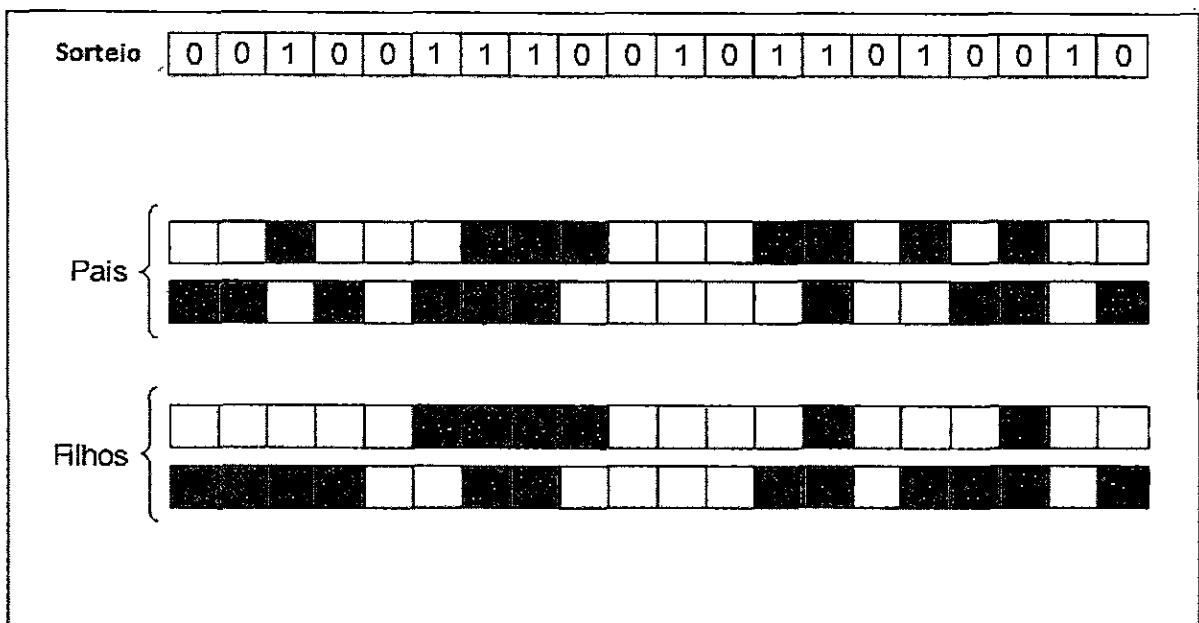


Figura 5: Crossover uniforme.

Fonte: Algoritmos Genéticos: um estudo de seus conceitos fundamentais e aplicação no problema de grade horária Lucas (2000). Disponível em: <www.ufpel.tche.br/prg/sisbi/bibct/acervo/info/2000/Mono-Diogo.pdf>.

2.3.5.2 - O operador Genético Mutação

Outro operador genético utilizado em GAs é o operador de mutação. Ele atua sempre depois da criação dos filhos pelo operador de crossover e têm como principal função evitar a convergência genética, ou seja, fazer com que a população não perca a diversidade genética com indivíduos muito semelhantes.

A mutação em algoritmos genéticos convencionais segundo Koza (1998 p.77) é um operador benéfico por reintroduzir a diversidade na população que está tendendo a convergência genética prematuramente.

Esta ocorre de várias formas, porém a mais comum consiste em percorrer todo o cromossomo (alelo por alelo) e sortear um número, se esse número for menor que a taxa de mutação altera-se o valor do alelo. No caso de cromossomos com representação binária troca-se o alelo com valor zero pelo valor um, e o alelo com valor um troca-se pelo valor zero.

A taxa de mutação tem que ser baixa, pois se essa taxa for muito alta estaremos determinando a solução de forma aleatória. Logo a taxa de mutação deve ser um valor pequeno o suficiente para gerar diversidade e atenuar o efeito da convergência genética. (Linden 2008 p.78)

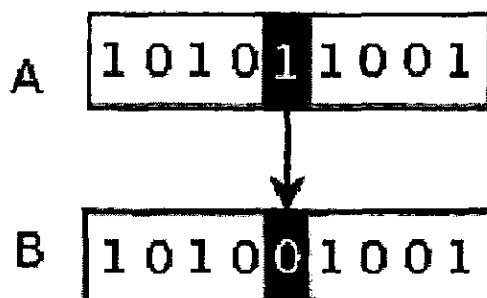


Figura 6: Exemplo de Mutação em um GA com representação cromossomial binária.
Fonte: Primária

2.3.6 Atualização

A atualização é a forma com que os algoritmos genéticos inserem os novos indivíduos (filhos criados pelo crossover) dentro da população. Nos GAs simples essa atualização é realizada apenas substituindo os pais pelos filhos criados. Neste caso o tamanho da população é fixa, pois em um crossover utilizamos dois pais que geram dois filhos e na mutação apenas é alterada uma parte do cromossomo que representa cada indivíduo. Outras formas de atualizar a população podem ser realizadas, por exemplo:

- **Elitismo:** Esta técnica consiste em não excluir totalmente os pais para inserção dos filhos na nova geração, os pais que possuem o melhor desempenho continuam na geração posterior preservando assim os indivíduos melhores avaliados. Assim garantimos que sempre o melhor indivíduo da geração $n + 1$ é pelo menos igual ao melhor indivíduo da população n . Linden (2008 p.141) afirma que esta pequena alteração muda de forma significativa o desempenho de execução de um GA. Logicamente isto ocorre porque estamos sempre mantendo na população os esquemas responsáveis pelas melhores avaliações. Vale lembrar que esta técnica por sempre escolher os melhores da geração anterior pode contribuir para a convergência genética.
- **Steady State:** Nos módulos da população citados anteriormente no momento em que uma população nova nasce à antiga população morre, ou seja, é substituída pela população que nasceu.

O steady state tenta simular a característica natural das populações biológicas onde indivíduos nascem aos poucos e os mais velhos morrem de forma lenta e há uma interação entre as gerações, isto implica que indivíduos da população $n + 1$ podem interagir com indivíduos da população n , por exemplo. Logo no steady state não é criada uma população de uma só vez, os filhos vão sendo criados aos poucos e substituirão os piores pais.

2.3.7 Finalização

Nesta fase o algoritmo genético já iniciou a população, avaliou a representação de cada indivíduo, selecionou os indivíduos para que eles possam ser submetidos aos operadores genéticos de crossover e mutação, foram criados os filhos e eles foram inseridos dentro da população. Este processo é realizado como o mostrado na figura 1 até que seja satisfeito um critério de parada que pode ser a quantidade de gerações criadas (que depende do problema específico), do grau de convergência atingindo pela população (caso os indivíduos possuem avaliações semelhantes). Caso nenhum desses critérios seja satisfeito o processo repete-se iniciando do item 2.2.

Este capítulo descreveu os principais conceitos referentes à teoria dos algoritmos genéticos, esse conhecimento é útil entendermos a estrutura e o funcionamento de um GA. Devido as suas características adaptabilidade, generalidade, busca codificada, entre outras, ele se mostra adequado para a resolução do problema de elaboração de grade horária. Nos próximo capítulo será abordada a problemática de otimização combinatorial e o problema da elaboração de grade horária.

3 PROBLEMÁTICA DE OTIMIZAÇÃO COMBINATORIAL E O PROBLEMA DA ELABORAÇÃO DE GRADE HORÁRIA ESCOLAR.

Os problemas de otimização combinatorial estão presentes em diversas áreas dentre elas podemos citar a alocação de recursos (como escalonamento de horários ou tarefas), escalonamento ou roteamento de veículos, planejamento de expansão e alocação de capacitores (setor elétrico) entre outros.

Assim Hamawaki (2005 p.4) e Linden (2008 p.327) afirmam que uma grande variedade desses problemas podem ser modelados como problemas de máximo ou mínimos de uma função que apresenta variáveis que obedecem a tipos de restrições associadas a elas. Logo encontrar soluções ótimas, ou até mesmo aproximadas, para este tipo de problemática muitas vezes é um trabalho árduo e demorado de ser feito.

3.1 OTIMIZAÇÃO COMBINATORIAL

A Otimização Combinatorial estuda problemas de otimização onde estes são modelados computacional e matematicamente através de uma função que possui uma série de variáveis de decisão associadas a um tipo de restrição. É um ramo presente nas ciências da computação e consiste na maximização ou minimização da função também conhecida por função objetivo. Assim o ótimo da função ao qual o valor atribuído as variáveis não viole nenhuma restrição, ou viole o mínimo possível, representa a resposta ao problema proposto.

A geração de grade horária é um problema de otimização combinatorial que possui uma enorme quantidade de variáveis e a função objetivo busca minimizar conflitos e maximizar preferências e compactar horários de professores e alunos. Este problema pode ser formulado de diversas formas, logo para cada instituição, em particular, deve ser elaborada uma definição específica para a elaboração de uma grade horária mais eficiente (Hamawaki 2005 p.5, Linden 2008 p.51).

A grade horária no contexto de uma instituição de ensino superior é uma tabela onde os alunos e professores se informam sobre o horário das disciplinas ao longo da semana durante todo o período letivo, ela é algo tão importante que influi de forma significativa no cotidiano de todo corpo docente, discente e dos funcionários de uma instituição. Uma vez elaborada e posta em prática à grade horária será válida durante todo o período letivo, o que fará, por várias vezes, com que alunos e professores tenham que se adaptar a ela, o que não

necessariamente atende a todos os interesses e disponibilidades dos envolvidos sejam eles alunos, professores ou funcionários da instituição de ensino.

3.2 ORDEM DE GRANDEZA DA COMPLEXIDADE DO PROBLEMA

Já é sabido que o problema proposto representa um problema de otimização combinacional e segundo Hamawaki (2005 p.6) este problema pode ser expresso em uma ordem de grandeza de complexidade através de um fórmula criada por FRANGOULI (2002) e expressa o tamanho do espaço de pesquisa do problema de elaboração de grade de horários. A fórmula criada por ele é mostrada abaixo:

$$TEP = (NDS \times NAD \times NSA)^{(ND - NAP)}$$

onde:

TEP é o tamanho do espaço de pesquisa;

NDS é o número de dias da semana letiva. Habitualmente este valor será 5 (de segunda a sexta) ou 6 (de segunda a sábado, caso do curso de Bacharelado em Ciências da Computação da Universidade Estadual do Piauí - UESPI);

NAD é o número de aulas diárias para os turnos envolvidos na distribuição das aulas consideradas;

NSA é o número (ou média) de salas de aulas disponíveis para a utilização das aulas;

ND é o número de disciplinas pertencentes ao horário de aulas;

NAP é o número de aulas a serem distribuídos para cada uma das disciplinas.

Aplicando a fórmula ao estudo de caso do Curso de Ciências da Computação da UESPI sem levar em consideração a variável NSA (número de salas disponíveis). Temos:

NDS (número de dias da semana) = 6 (incluindo o sábado);

NAD (número de aulas diárias para os turnos) = 12;

ND (número de disciplinas pertencentes ao horário de aulas) = 22 (22 disciplinas dos quatro períodos do semestre 2010.1);

NAP (número de aulas a serem distribuídos para cada uma das disciplinas) = 2 (média);

$$TEP = (6 \times 12)^{(22 - 2)}$$

$$TEP = 14016833953562607293918185758734155776$$

Logo o tamanho do espaço de busca $TEP = 1,40 \times 10^{37}$

3.3 DEFINIÇÃO DO PROBLEMA

A elaboração de grades horárias para cursos é um problema muito conhecido no meio acadêmico, em especial pelos alunos de computação, tanto por estudos a ele dedicados, complexidade devido à necessidade de conciliar vários recursos como professores, disciplinas, salas, quanto por estar presente em toda instituição e a cada período letivo.

O problema consiste em alocar em um conjunto de intervalos de tempo, aulas de disciplinas de cada curso (Hamawaki 2005 p.7) minimizando as restrições possíveis e maximizando as preferências dos envolvidos na grade horária. Tal processo efetuado manualmente pode levar vários dias para uma conclusão devido a grande quantidade de restrições que, embora pareçam simples, precisam ser atendidas ou minimizadas. Por isso muitos métodos de otimização desenvolvidos para a resolução deste problema encontram dificuldades ao lidar com o grande número de restrições e a alta variedade de seu grau de importância, bem como falsos pontos ótimos.

As restrições para a resolução do problema são classificadas em fracas e fortes (Hamawaki 2005 p.7, Lucas 2000 p.46, Linden 2008 p. 328) assim:

- Violações às restrições fortes (como, por exemplo, a alocação de um professor em duas disciplinas diferentes em um mesmo horário) não servem como alocação e devem ser descartadas.
- Violações às restrições fracas geram alocações válidas, no entanto a qualidade da solução não é a melhor gerando uma solução de má qualidade (por exemplo, a preferência dos professores por determinados horários). Porém as restrições fracas geralmente são muito difíceis de serem obedecidas devido ao fato de que dois ou mais professores podem estar indisponíveis em um mesmo slot de tempo (exemplo segunda-feira no primeiro horário da tarde), fazendo com que um deles seja alocado em um espaço em que ele está indisponível.

Somado ao problema de alocação de professores onde um mesmo professor não possa lecionar em turmas diferentes em um mesmo horário, matérias de um mesmo período não podem ser alocadas no mesmo horário e a alocação do docente somente em horários definidos por sua preferência entre outras violações/restrições que devam ser consideradas pelo algoritmo além de que as instituições de ensino também devem garantir que seu espaço de ensino seja utilizado eficientemente.

A figura abaixo ilustra a grade horária do 8º período do curso de Bacharelado em Ciências da Computação, onde estão dispostas as disciplinas na semana e o seu horário correspondente.

Bacharelado em Ciências da Computação – 8º Período 2010.1							
Horário	SEG	TER	QUA	QUI	SEX	SAB	
18:20 ◆ 20:10	SEG & AUD	TOPE II	SIS DIST	TELEC	TELC	SIS DIST.	08:00 ◆ 09:50
20:20 ◆ 22:00	TEO GRA	SEG & AUD	GEST TI	TOPE II	TEO GRA	GEST TI	10:00 ◆ 12:00

Figura 7: Grade Horária do 8º período – Bacharelado em Ciências da Computação Campus Profº Alexandre Alves de Oliveira.

Fonte: Coordenação do curso de Bacharelado em Ciências da Computação Campus Profº Alexandre Alves de Oliveira.

A grade horária em instituições de ensino superior geralmente é elaborada por cada curso e a cada período onde os períodos do curso, em geral, correspondem a um semestre (no caso específico do curso de Ciências da Computação da UESPI, o curso apresenta oito períodos, totalizando quatro anos de duração). A elaboração da grade horária é uma tarefa individual de cada curso onde são conhecidos os docentes e a sua disponibilidade ao longo da semana para que o professor seja encaixado em horários pré-determinados procurando satisfazer as suas necessidades. O aluno também é parte importante nesta tarefa e o mesmo possui uma quantidade de horários disponíveis limitados ao longo do dia. O aluno possui um horário semanal de segunda-feira a sábado de tarde para alguns períodos, de noite para outros e de manhã nas aulas aos sábados num total de doze horários disponíveis por período na semana.

Definido o problemas passaremos a analisar o estudo de caso referente a Universidade Estadual do Piauí – UESPI Campus Profº Alexandre Alves de Oliveira no próximo capítulo.

4 ESTUDO DE CASO: GERAÇÃO DA GRADE HORÁRIA PARA O CURSO DE CIÊNCIAS DA COMPUTAÇÃO DA UNIVERSIDADE ESTADUAL DO PIAUÍ - UESPI CAMPUS PROF. ALEXANDRE ALVES DE OLIVEIRA

O curso de Ciências da Computação da Universidade Estadual do Piauí possui as suas disciplinas dispostas em oito períodos. O aluno graduar-se-á no curso somente depois de se matricular em todas as disciplinas e obtiver aprovação nas mesmas. Cabe também ao discente o cumprimento de trezentas horas de estágio supervisionado e a elaboração da monografia para a obtenção do título do Bacharel em Ciências da Computação.

As disciplinas são dispostas em períodos e a quantidade de delas nos diferentes períodos pode variar. Ao se matricular em um período o aluno vincula-se em todas as disciplinas correspondentes ao mesmo salvo em casos de pedido de dispensa (caso em que aluno já cursou esta em outro curso). Em anexo segue a figura ilustrativa da grade horária do curso de Bacharelado em Ciências da Computação onde estão dispostas as disciplinas, os professores e os horários da semana para cada disciplina.

Assim a elaboração da grade horária é de extrema importância, pois a partir de sua implantação na instituição ela será utilizada por todo o semestre letivo e fará com que a rotina dos docentes, discentes e funcionários sejam diretamente e indiretamente influenciada por ela.

Devido a grande quantidade de restrições a serem respeitadas a elaboração da grade de horários é um trabalho árduo e dentre as elas merecem destaque:

- A quantidade de aulas de cada disciplina;
- O desejo do professor em lecionar em determinados horários, ou seja, suas preferências em relação a cada slot de tempo;
- A ocorrência de duas ou mais disciplinas sendo lecionadas por um mesmo professor em um mesmo turno devem ocorrer em horários distintos.

4.1 ESTRUTURA

A estrutura do problema é essencial para a sua resolução e entendimento. Nela deve estar contida o máximo de informações e características relativas a criação da grade horária do curso de Ciências da Computação da UESPI dentre elas estão:

- A duração do curso é de quatro anos;

- O curso é dividido em 8 períodos;
- Possui um total de 45 disciplinas (contando com o estágio supervisionado);
- Possui quatro professores em seu quadro efetivo e oito professores substitutos que compõe o quadro de docente do curso;
- As aulas são ministradas no período da tarde (caso das turmas que cursam o 1º e o 5º período no semestre 2010.2) e da noite (caso das turmas que cursam o 3º e 8º período no semestre 2010.2) e de manhã (para todos os períodos aos sábados);
- Os horários das aulas estão compreendidos entre as 08h00 às 12h00 (caso das aulas aos sábados), 14h00 às 17h30 e das 18h30 às 22h00;
- Cada professor preenche uma ficha indicando o seu grau de preferência com relação a cada horário (slots de tempo) para lecionarem disciplinas.

Definido e estruturado o problema passaremos a analisar os detalhes da implementação do PEGH (Programa de Escalonamento de Grade Horária).

5 IMPLEMENTAÇÃO DO PEGH (PROGRAMA DE ESCALONAMENTO DE GRADE HORÁRIA)

O PEGH – Programa de Escalonamento de Grade Horária é um sistema criado para a geração da grade curricular para o curso de Bacharelado em Ciências da Computação da UESPI Campus Prof^o Alexandre Alves de Oliveira. Este sistema foi desenvolvido na linguagem de programação Java, utilizando também os principais conceitos dos algoritmos genéticos para a formulação, estruturação e a elaboração do mesmo. Tem por objetivo principal a elaboração de uma grade horária satisfatória para o curso maximizando as preferências dos professores e minimizando as restrições associadas ao problema e ao estudo de caso.

O funcionamento do PEGH segue o fluxo da execução dos algoritmos genéticos da seguinte maneira:

- a) Definida a quantidade de indivíduos para compor a população são criados os indivíduos das quatro espécies referentes as turmas que irão compor o semestre letivo. Na etapa de inicialização são feitas consultas a base de dados para a coleta de informações que serão utilizadas na criação destes indivíduos.
- b) Selecionam-se os pais. A escolha é feita baseada na aptidão dos mesmos para a atuação dos operadores genéticos (crossover, mutação).
- c) Criam-se os filhos para serem inseridos na população substituindo os pais.
- d) O processo repete-se iniciando novamente do item 2 até que seja satisfeita uma condição de parada definida, neste caso, que o limite de gerações foi alcançado pelo algoritmo.

Este é o princípio básico do fluxo de execução do programa, porém para esse fluxo seguir de forma correta e eficiente uma série de dificuldades são encontradas na criação do programa escalonador de grade horária e essas são definidas através de três naturezas:

- a) Múltiplos Objetivos: O programa precisa atender as preferências dos vários professores e garantir aos alunos de todas as turmas seus horários definidos para as aulas.
- b) Imposição de Restrições de Grau variado – já é sabido que na geração de uma grade horária um grande número de restrições são analisadas e devem receber uma penalização baseada em seu grau de restrição para que elas possam ser avaliadas de

- forma eficiente. Para isso utiliza-se a estratégia de pesos para cada uma;
- c) Quantidade de Variáveis – que corresponde a soma do total de Alunos, de Professores, de Disciplinas, Salas e Carga Horária.

As restrições descritas no item b acima são classificadas da seguinte maneira:

1. Leves: as restrições consideradas leves estão relacionadas às preferências dos horários dos professores onde estes, através de um formulário, classificam a sua preferência pelos horários disponíveis para as aulas. Este tipo de penalização é considerada fraca e representa a principal responsável pela avaliação do indivíduo;
2. Graves: as restrições que são consideradas graves no contexto do PEGH geram soluções inválidas e são descartas no momento em que são encontradas pelo programa. São muitas as restrições consideradas fortes não só no PEGH, mas sim, em qualquer programa que seja criado para o desenvolvimento de uma grade horária de qualquer instituição.

Dentre as restrições graves merecem destaque:

- 2.1 Cada disciplina tem uma carga horária semanal para que ao final do período letivo seja cumprida a carga horária total da disciplina (exemplo uma disciplina com carga horária total de noventa horas deve estar presente em três aulas de 1h40 cada aula, ou seja, a disciplina deve estar em três dos doze slots de tempo da semana). Esta relação deve garantir que os professores, em cada turma que leciona, devem estar presentes em um número pré-definido de horas para que não haja “janelas” que correspondem a espaços vazios entre as aulas;
- 2.2 Cada professor deve ser alocado em uma turma para um horário de aula, ou seja, não deve haver a ocorrência de dois professores lecionando em uma mesma turma no mesmo horário;
- 2.3 Do item acima se conclui também que nenhum professor pode estar em duas turmas ao mesmo horário;
- 2.4 Não deve haver a ocorrência de “janelas”, ou seja, horário em que nenhum professor foi especificado para uma turma. Esta ocorrência fará com que as aulas não fiquem distribuídas ao longo dos doze horários definidos para cada turma;

Essas restrições devem ser consideradas para os programas que foram designados para a criação de uma grade horária. Para o PEGH as seguintes restrições foram consideradas:

- a) O professor não pode lecionar em duas turmas ao mesmo tempo;
- b) Não pode haver a ocorrência de dois professores em uma mesma turma em mesmo horário;
- c) Não deve haver “janelas” entre as aulas;
- d) O número determinado de aulas por semana deve ocorrer;
- e) Disciplinas diferentes sendo ministradas pelo mesmo professor não podem estar alocadas em um mesmo horário;
- f) Os professores devem ter suas preferências maximizadas, ou seja, deve-se respeitar ao máximo a preferência dos professores por determinado período.

Na grade horária presente no axeno podemos ver a presença de professores lecionando em turmas diferentes ou, até mesmo, nas mesmas turmas, porém lecionando disciplinas distintas. A figura abaixo mostra um exemplo de uma consulta na base de dados do sistema através do SGBD (Sistema Gerenciador de Banco de Dados) PostgreSQL mostrando um professor e as disciplinas por ele ministradas.

Query: pegh em postgres@localhost:5432

Arquivo Editar Consulta Favoritos Macros Visualizar Ajuda

SQL Editor | Graphical Query Builder

```
SELECT p.nome AS "Professor", d.nome AS "Disciplina", d.sigla AS "Sigla", d.período AS "Período"
FROM professor p, disciplina d
WHERE p.id = 19 and d.professor = 19
ORDER BY d.período;
```

Resultado da consulta:

	Professor character varying(200)	Disciplina character varying(200)	Sigla character	Período integer
1	ALESSANDRO SARAIVA MENDONÇA	INTELIGÊNCIA ARTIFICIAL	IA	3
2	ALESSANDRO SARAIVA MENDONÇA	ANÁLISE DE SISTEMAS II	AS II	3
3	ALESSANDRO SARAIVA MENDONÇA	GESTÃO DE TI	GTI	4
4	ALESSANDRO SARAIVA MENDONÇA	TELECOMUNICAÇÕES	TELECOM	4

Figura 8: Consulta efetuada no Banco de Dados do PEGH mostrando as disciplinas lecionadas pelo professor Alessandro Saraiva no semestre letivo 2010.1.

Fonte: Primária

A função de avaliação é uma função de maximização das preferências dos professores, pois é a partir das mesmas que avaliamos os indivíduos da população, assim o indivíduo que atender a maior quantidade de preferências de todos os professores que estão lecionando nas disciplinas do curso será o melhor resultado obtido pelo algoritmo genético.

Além das restrições essenciais para a geração da grade de horários algumas considerações e limitações para o desenvolvimento do PEGH tiveram de ser analisadas entre elas merecem destaque:

- Os alunos efetuam a matrícula por período e não por disciplina (salvo os casos de reprovação);
- A turma é composta por alunos que cursam um conjunto de disciplinas oferecidas por um período do curso;
- O período letivo é composto por um conjunto de disciplinas e tem geralmente a duração de um semestre e as disciplinas do mesmo estão distribuídas ao longo dos doze slots de tempo (de segunda à sábado duas aulas por dia) esses slots que formam a grade horária do período letivo.

Portanto depois de definidas e as restrições, considerações e limitações o conjunto de períodos com suas respectivas disciplinas associadas aos professores e alocadas nos doze slots de tempo constituem a grade horária total do curso.

5.1 INFORMAÇÕES RELATIVAS À BASE DE DADOS UTILIZADA PELO PEGH

Para a geração da grade horária proposta pela aplicação faz-se necessário uma série de informações referentes as disciplinas, docentes e disponibilidade dos professores do curso. Assim o PEGH busca estas informações em uma base de dados para o processamento das informações gerando a população inicial, base para as próximas gerações.

Os dados são armazenados em tabelas específicas em um banco de dados. Este foi criado e é gerenciado pelo SGBD (Sistema Gerenciador de Banco de Dados) PostgreSQL por apresentar algumas características e funcionalidades como:

- Suporte a Comandos Complexos;
- Chaves Estrangeiras;
- Gatilhos;
- Visões;
- Integridade Transacional;
- Controle de simultaneidade Multiversão;

O PostgreSQL não só oferece essas funcionalidades, como também, permite ser estendido de muitas formas pelo usuário como, por exemplo, adicionando novos

- Tipos de Dado;
- Funções;
- Operadores;
- Funções de Agregação;
- Linguagens Procedurais;

O PostgreSQL é um SGBD derivado do POSTGRES desenvolvido pelo Departamento de Ciência da Computação da Universidade da Califórnia em Berkeley. O POSTGRES foi o pioneiro em muitos conceitos que somente se tornaram disponíveis bem mais tarde em outros sistemas de banco de dados.

Também pelo fato de possuir código-aberto o PostgreSQL tem sua licença livre e pode ser utilizado, modificado e distribuído por qualquer pessoa para qualquer finalidade, seja privada, comercial ou acadêmica, livre de encargos.

As informações constantes nas tabelas são essências para o processamento do programa e serão detalhadas à seguir.

5.1.1 Tabela de Professores

A tabela de professores é uma tabela que contém um conjunto de tuplas com os campos id e nome. A tabela contém somente estas informações, pois apenas elas serão utilizadas pelo programa para processamento. A figura a seguir ilustra a tabela de professores.

	id	nome
	[PK] integ	character varying(200)
1	8	EYDER RIOS
2	11	DARIO CALÇADA
3	12	DARLENE SANTOS
4	13	FRANCISCO ROCHA
5	14	GIOVANE GOMES
6	15	JOSE ROBERTO
7	16	LEONAN MOURA
8	17	PABLO RODRIGUES
9	18	WINSTON SILVA
10	19	ALESSANDRO SARAIVA
*		

Figura 9: Consulta efetuada no Banco de Dados do PEGH mostrando uma lista com alguns professores cadastrados.

Fonte: Primária

5.1.2 Tabela de Disponibilidade dos Professores

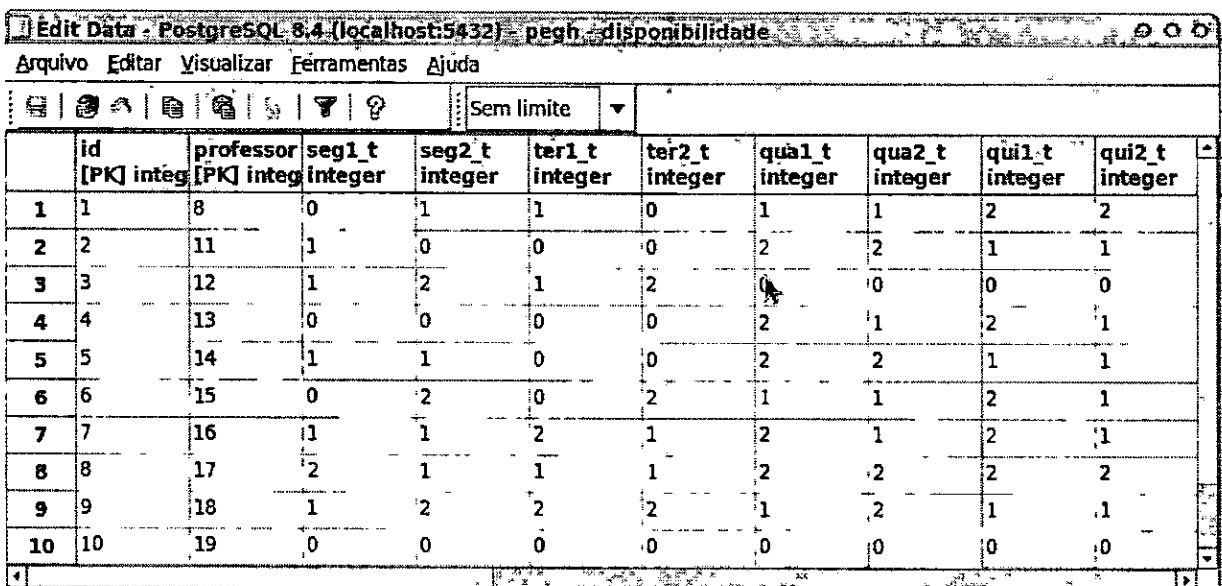
A tabela de disponibilidade dos professores é responsável pelo armazenamento das preferências dos professores por cada um dos horários do dia da semana. Assim o conjunto de tuplas dessa tabela contem o id da disponibilidade, o id do professor, e cada horário possível dentro da grade horária, ou seja, cada slot de tempo em que o professor poderá lecionar. Para ficar mais claro observe a codificação de uma coluna referente a um slot de tempo:

Slot: seg1_t

onde:

- seg = corresponde a segunda-feira;
- 1 = corresponde ao primeiro horário;
- t = indica que o horário é no turno da tarde;

Assim as colunas da tupla referentes aos slots de tempo estão assim representados seg1_t, seg2_t, ..., sab1, sab2, na base de dados. Cada coluna destas também recebe um valor que corresponde a preferência do professor por aquele horário, este valor x esta definido no intervalo $0 \leq x \leq 2$. A próxima figura ilustra a tabela de disponibilidade do professor.



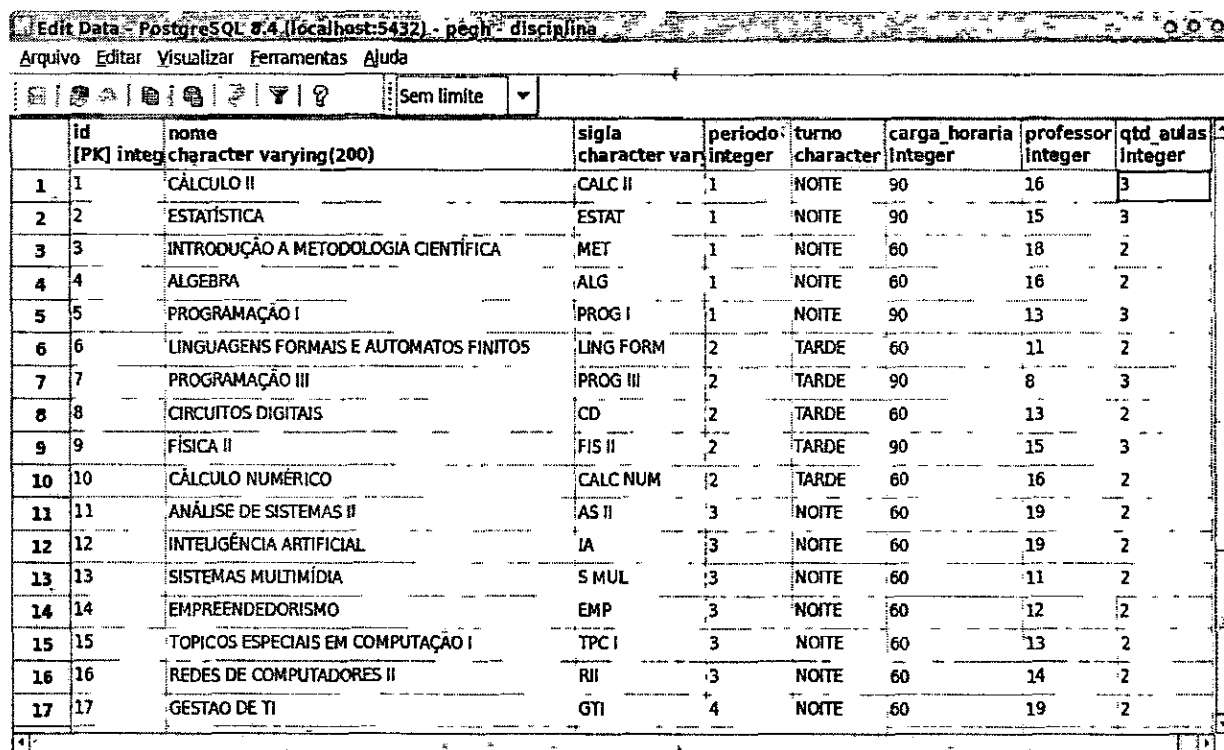
	id [PK] intég	professor [PK] intég	seg1_t integer	seg2_t integer	ter1_t integer	ter2_t integer	qua1_t integer	qua2_t integer	qui1_t integer	qui2_t integer
1	1	8	0	1	1	0	1	1	2	2
2	2	11	1	0	0	0	2	2	1	1
3	3	12	1	2	1	2	0	0	0	0
4	4	13	0	0	0	0	2	1	2	1
5	5	14	1	1	0	0	2	2	1	1
6	6	15	0	2	0	2	1	1	2	1
7	7	16	1	1	2	1	2	1	2	1
8	8	17	2	1	1	1	2	2	2	2
9	9	18	1	2	2	2	1	2	1	1
10	10	19	0	0	0	0	0	0	0	0

Figura 10: Tabela de Disponibilidade dos Professores dentro da Base de Dados do PEGH.

Fonte: Primária

5.1.3 Tabela de Disciplinas

A tabela de disciplinas apresenta um conjunto de tuplas com os campos id, nome, sigla, período, turno, carga horária, id do professor que a ministra e a quantidade de aulas na semana. A figura 11 mostra uma tabela de disciplinas.



	id [PK] integer	nome character varying(200)	sigla character var	periodo integer	turno character	carga_horaria integer	professor integer	qtd_aulas integer
1	1	CÁLCULO II	CALC II	1	NOITE	90	16	3
2	2	ESTATÍSTICA	ESTAT	1	NOITE	90	15	3
3	3	INTRODUÇÃO A METODOLOGIA CIENTÍFICA	MET	1	NOITE	60	18	2
4	4	ALGEBRA	ALG	1	NOITE	60	16	2
5	5	PROGRAMAÇÃO I	PROG I	1	NOITE	90	13	3
6	6	LINGUAGENS FORMAIS E AUTOMATOS FINITOS	LING FORM	2	TARDE	60	11	2
7	7	PROGRAMAÇÃO III	PROG III	2	TARDE	90	8	3
8	8	CIRCUITOS DIGITAIS	CD	2	TARDE	60	13	2
9	9	FÍSICA II	FIS II	2	TARDE	90	15	3
10	10	CÁLCULO NUMÉRICO	CALC NUM	2	TARDE	60	16	2
11	11	ANÁLISE DE SISTEMAS II	IAS II	3	NOITE	60	19	2
12	12	INTELIGÊNCIA ARTIFICIAL	IA	3	NOITE	60	19	2
13	13	SISTEMAS MULTIMÍDIA	S MUL	3	NOITE	60	11	2
14	14	EMPREENDEDORISMO	EMP	3	NOITE	60	12	2
15	15	TOPICOS ESPECIAIS EM COMPUTAÇÃO I	TPC I	3	NOITE	60	13	2
16	16	REDES DE COMPUTADORES II	RII	3	NOITE	60	14	2
17	17	GESTAO DE TI	GTI	4	NOITE	60	19	2

Figura 11: Tabela de Disciplinas utilizada pelo PEGH
Fonte: Primária

5.2 O ALGORITMO GENÉTICO

Nas seções anteriores foram explanados os conceitos referentes a teoria dos algoritmos genéticos juntamente com a problemática da geração de grade horária. Neste tópico serão apresentados os detalhes da implementação do PEGH. Vale ressaltar que o programa desenvolvido tem o objetivo de solucionar o problema da geração de grade horária do curso de Bacharelado em Ciências da Computação da Universidade Estadual do Piauí – UESPI Campus Prof^o Alexandre Alves de Oliveira. Portanto a aplicação pode não apresentar o mesmo desempenho se aplicada a outros problemas de geração de grade horária, apesar dos

algoritmos genéticos serem adaptáveis e genéricos.

5.2.1 Codificação Genética

A codificação genética é fundamental para o algoritmo genético, pois através dela que conseguimos traduzir o nosso problema em uma forma de ser tratada pelo computador. Uma boa codificação é feita embutindo o máximo de informação a respeito da problemática, por isso deve ser o primeiro aspecto a ser pensado na implementação de um GA.

O cromossomo é um possível candidato a solução e deve apresentar as informações a respeito da grade horária assim para a representação cromossomial utilizamos os dados constantes e variáveis para não correremos o risco de gerar soluções inválidas. As informações que são essenciais para a resolução do problema são:

Informações constantes:

- As disciplinas
- A quantidade de aulas por semana de cada disciplina
- O turno da disciplina
- Os professores, cada um vinculado a(s) disciplina(s) que ministra(m)
- A tabela de disponibilidade de cada professor

Informações Variáveis:

- Os Horários das aulas

Uma observação importante é que as informações constantes listadas acima estão armazenadas em uma base de dados e são acessados pela aplicação para que o GA possa trabalhar em cima desses dados gerando os dados variáveis.

Na codificação do algoritmo proposto foi utilizada uma matriz chamada grade onde as linhas correspondem aos períodos do curso e as colunas representam os slots de tempo.

A associação entre os slots de tempo que formam a grade horária com as disciplinas, professores e turnos é realizada através de outras três matrizes que representam os dados constantes ao longo da execução do GA. São elas a matriz de disciplina, professor e turno todas com a mesma dimensão da matriz cromossomo.

A seguir detalhes de cada uma das matrizes e um exemplo de como ocorre a

associação entre elas.

5.2.1.1 A matriz Grade

Esta é uma matriz de ordem 4x12, onde as linhas representam os períodos do curso, no caso do curso de Bacharelado em Ciências da Computação – UESPI todo semestre letivo possui quatro turmas em quatro períodos distintos e as colunas representam os doze slots de tempo disponíveis para a alocação das aulas. A figura a seguir ilustra um exemplo de uma matriz grade.

$$\begin{pmatrix} 6 & 3 & 8 & 5 & 2 & 10 & 7 & 12 & 4 & 11 & 1 & 9 \\ 1 & 6 & 5 & 2 & 3 & 12 & 9 & 4 & 11 & 10 & 8 & 7 \\ 5 & 3 & 1 & 12 & 2 & 8 & 11 & 7 & 9 & 10 & 4 & 6 \\ 12 & 3 & 5 & 2 & 8 & 1 & 10 & 7 & 7 & 11 & 9 & 6 \end{pmatrix}$$

Figura 12: Matriz Grade aqui as colunas com os números de 1 a 12 representam os slots de tempo, as linhas representam os períodos.

Fonte: Primária

5.2.1.2 A matriz Disciplina

Na matriz disciplina as linhas representam os períodos e as colunas as os códigos das disciplinas referentes ao período. As disciplinas aparecem na matriz na mesma quantidade em que aparecem ao longo da semana, ou seja, uma disciplina, por exemplo Inteligência Artificial, que possui uma carga horária total de sessenta horas, terá ao longo da semana duas aulas distribuídas para a linha que corresponde ao período em que ela esta inserida. Estas duas aulas, por questão de conveniência, estão dispostas sequencialmente como mostrado na figura abaixo:

$$\begin{pmatrix} 1 & 1 & 1 & 2 & 2 & 2 & 3 & 3 & 4 & 4 & 5 & 5 \\ 11 & 11 & 12 & 12 & 13 & 11 & 14 & 14 & 15 & 15 & 15 & 15 \\ 22 & 22 & 22 & 23 & 23 & 24 & 24 & 25 & 25 & 25 & 26 & 26 \\ 40 & 40 & 41 & 41 & 42 & 42 & 43 & 43 & 44 & 44 & 45 & 45 \end{pmatrix}$$

Figura 13: Matriz Disciplinas as colunas contém os números que correspondem aos identificadores das

disciplinas e as linhas representam os períodos da grade.

Fonte: Primária

5.2.1.3 A matriz Professor

Os professores precisam de uma codificação devido ao fato de um professor poder lecionar em mais de uma disciplina. A matriz de professor esta diretamente relacionada a matriz disciplina. Ela também apresenta a ordem 4x12 e as linhas, assim como as outras tabelas, são constituídas pelos períodos, já as colunas recebem o id do professor associado a disciplina que ocupa a mesma posição na matriz de disciplinas. Através da figura podemos perceber como é formada a matriz de professores.

$$\begin{pmatrix} 22 & 22 & 22 & 16 & 16 & 16 & 0 & 8 & 11 & 11 & 13 & 13 \\ 22 & 22 & 23 & 23 & 12 & 12 & 15 & 15 & 15 & 13 & 13 & 13 \\ 23 & 23 & 23 & 13 & 13 & 8 & 8 & 0 & 0 & 0 & 23 & 23 \\ 12 & 12 & 11 & 11 & 11 & 11 & 8 & 8 & 0 & 0 & 16 & 16 \end{pmatrix}$$

Figura 14: Matriz Professor que guarda o identificador do professor para a disciplina alocada na mesma posição da matriz disciplina.

Fonte: Primária

5.2.1.4 A matriz Turno

Como os períodos do curso estão divididos nos turnos vespertino e noturno faz-se necessária também uma representação através de uma matriz para o turno das disciplinas e, assim como, a tabela de professores está relacionada com a tabela de disciplinas onde a matriz de turno possui ordem de 4x12 sendo que as linhas correspondem aos períodos. As colunas possuem valor 0 ou 1, onde o valor 0 representa o período vespertino e o valor 1 representa o período noturno. As colunas do matriz turno associam-se com as colunas da matriz de disciplinas onde a coluna receberá o valor 0 ou 1 dependendo do turno da disciplina na mesma posição na tabela de disciplinas conforme a figura abaixo.

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Figura 15: Matriz Turno nesta matriz são mostrados os turnos referentes aos períodos. O valor 0 representa o turno vespertino e o valor 1 representa o turno noturno.

Fonte: Primária

Definida e detalhada a codificação do indivíduo é mostrado um exemplo da codificação de um indivíduo através de um exemplo na figura 15 onde temos duas linhas de cada matriz mostrando a associação entre elas.

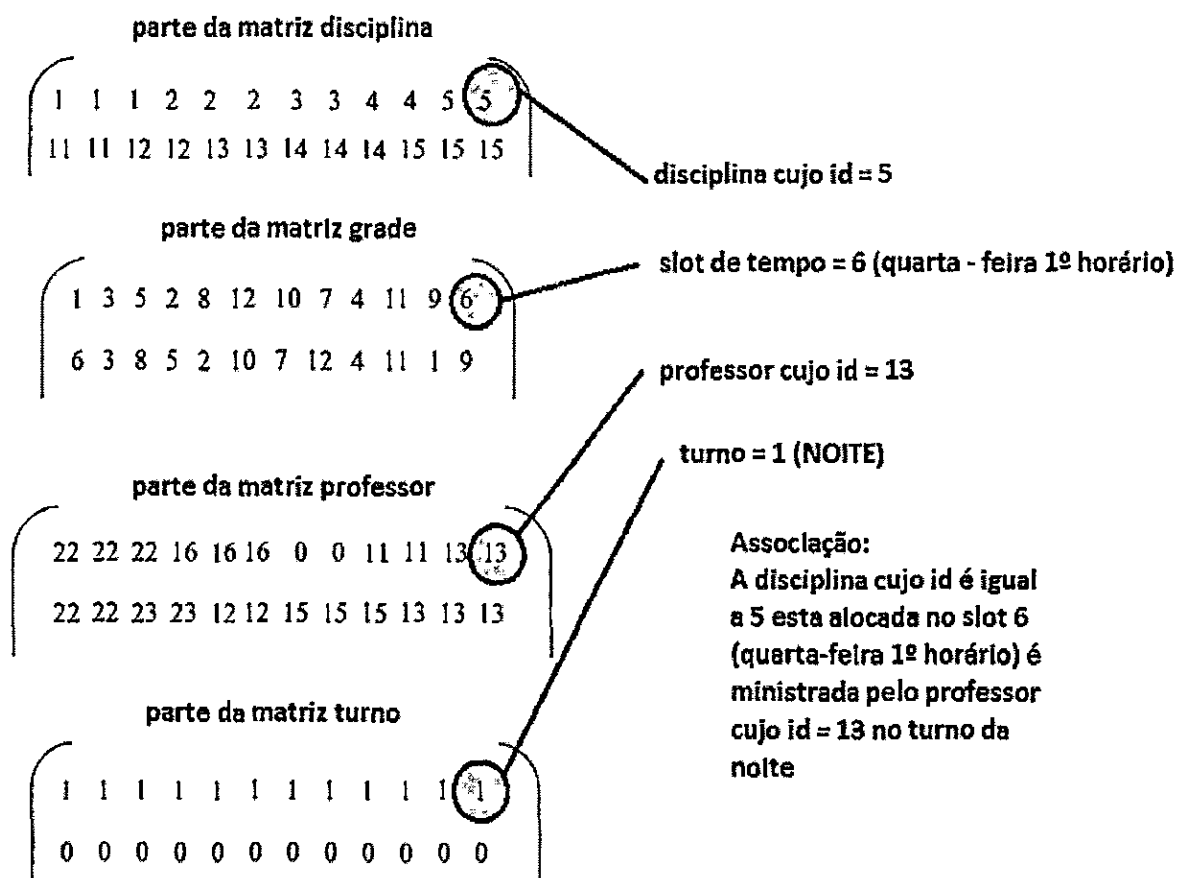


Figura 16: Ilustração de como ocorre a associação entre as matrizes para a codificação do indivíduo.
Fonte: Primária

5.2.2 Inicialização da População

Os Algoritmos genéticos trabalham em cima de uma população e, para tanto, deve criar uma população inicial criando um conjunto de cromossomos. Um detalhe muito importante a ser considerado na inicialização da população é o seu tamanho, esse parâmetro é muito difícil de se definir, pois cada problema possui uma modelagem de acordo com suas necessidades. Logo encontrar um valor considerado satisfatório só é possível através de testes.

A inicialização da população é realizada, atribuindo valores aleatórios aos elementos da matriz cromossomo de cada indivíduo. A inicialização é aleatória, porém ela não pode gerar soluções consideradas inválidas devido ao fato de que a população inicial é a base para as outras gerações e uma solução inválida gerada na população inicial comprometeria todo o

desenvolvimento do GA.

5.2.3 A Função de Avaliação

Para podermos determinar a qualidade de um indivíduo como solução do problema proposto utilizamos a função de avaliação. Além disso, ela deve embutir todas as restrições do problema aplicando punições para os indivíduos que as desrespeitarem.

5.2.3.1 Disponibilidade do Professor

O cálculo da aptidão é responsável por descobrir qual indivíduo tem em cada slot de sua matriz cromossomo mais próxima da disponibilidade de todos professores para os slots em que eles estão alocados. O aplicativo possui em sua base de dados uma tabela onde está armazenada a disponibilidade de cada um dos professores. Os docentes preenchem o quadro horário atribuindo valores inteiros a cada um dos doze slots de acordo com a seguinte convenção 0: horário não disponível; 1: horário condicionalmente disponível; 2: horário disponível.

Portanto a disponibilidade total do cromossomo, mostrada na equação abaixo, é calculada através do somatório do peso correspondente de cada slot para o professor que o está alocando.

Assim matematicamente temos:

$$D_{(Total)} = \sum_{i=0}^n Disponibilidade(slot)$$

no qual:

n = representa o número total de genes;

$disponibilidade (slot)$ = disponibilidade do professor para o slot cuja disciplina lecionada por ele foi alocada.

5.2.3.2 - Outras Penalizações

A função objetivo do PEGH considera somente com critério de avaliação a soma das preferências dos professores para os slots em que eles foram alocados, porém outras penalizações são analisadas para a escolha da solução do problema proposto. Dentre as restrições que não possuem uma fórmula para o cálculo da função de avaliação destacam-se:

- Penalização por ocorrência de janelas e ocorrência de disciplinas por dia além do permitido – estas duas penalizações não ocorrem devido ao fato de que os períodos ocorrem somente em um turno (pela tarde ou pela noite), exceto as aulas aos sábados, e em horários suficientes para a alocação de todas as disciplinas. Logo a carga horária de todas as disciplinas dos períodos se encaixarem em todos os slots disponíveis, graças também, as aulas aos sábados não ocorrendo as chamadas “janelas”.
- Penalização pela ocorrência de professor lecionando disciplinas diferentes em mesmo horário - Por esta ser uma das restrições que gera um indivíduo inválido possui a penalização mais grave. Pelo fato de que esta é uma condição impossível de se ter em uma grade horária funcional, o cromossomo não pode nem se quer ser inserido na população inicial.

5.2.3.3 Cálculo total da Função de Avaliação

O cálculo da função de avaliação determinará o quão bom o cromossomo é como candidato a solução do problema. Sendo assim para o cálculo da função de avaliação é estabelecida a seguinte fórmula:

$$D_{(Total)} = 3 \times \sum_{i=0}^n Disponibilidade(slot)$$

O número inteiro 3 multiplicado pela soma das disponibilidades dos professores foi o que melhor se adequou ao problema proposto, principalmente pela escolha dos pais assunto para a próxima seção.

5.2.4 Método de Seleção: Roleta

Qualquer indivíduo criado na população inicial dificilmente será a solução do algoritmo genético por geralmente apresentar baixa avaliação, por isso faz-se necessário a atuações dos operadores genéticos. Esses operam em cima dos elementos da população para a geração de novos indivíduos, possível e provavelmente, melhores que os pais.

Como os operadores genéticos trabalham em cima da população precisamos escolher os cromossomos que irão participar destas operações. A escolha desses cromossomos é feita através do método da roleta. Esse método cria em uma roleta virtual como mostra a figura X formada pelas avaliações dos cromossomos presentes na população, sendo que cada cromossomo possui uma parte da roleta proporcional ao resultado da sua função de avaliação. Dessa forma, esse método permite a seleção de indivíduos com qualquer tipo de avaliação, privilegiando logicamente os mais aptos.

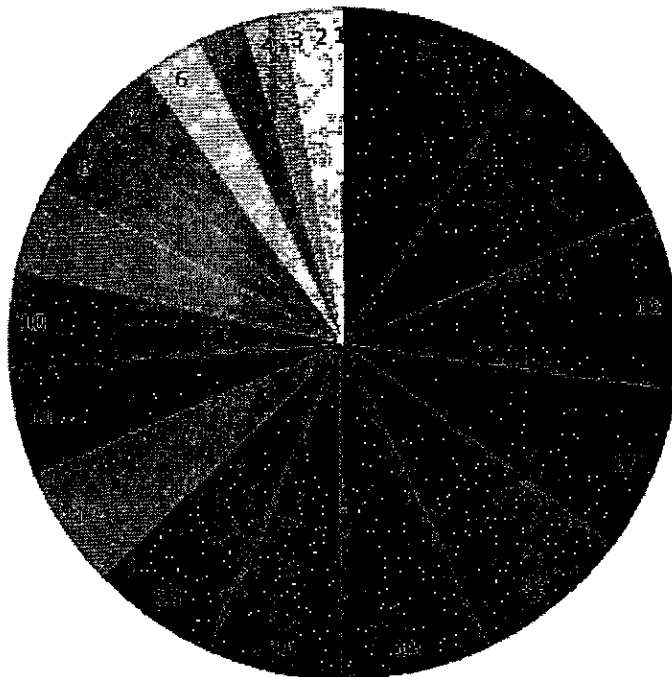


Figura 17: Roleta e a parte de cada indivíduo dentro da população de acordo com sua avaliação.

Fonte:

A roleta é uma função implementada no PEGH que retorna o índice do indivíduo dentro da população. O algoritmo abaixo foi utilizado como base para a implementação da função.

- 1) Some todas as avaliações para uma variável soma
- 2) Selecione um número s entre 0 e soma e armazene em uma variável chamada limite

- 3) $i = 0^2$
- 4) $aux =$ avaliação do indivíduo
- 5) faça
- 6) $i = i + 1$
- 7) $aux = aux +$ avaliação do indivíduo i
- 8) enquanto ($aux < s$)
- 9) retorne i

5.2.5 Operadores Genéticos

Os indivíduos selecionados pela roleta sofrerão a atuação dos operadores genéticos tanto para a geração de possíveis descendentes com melhores avaliações, tanto para gerar mais diversidade na população. Os operadores genéticos utilizados no PEGH foram o crossover e a mutação que serão definidos com mais detalhes nas seções à seguir.

5.2.5.1 Crossover (Cruzamento)

O cruzamento dos cromossomos chamados de pais realiza-se pela troca do material genético entre os mesmos. Devido ao fato de os operadores genéticos estarem trabalhando com cromossomos representados por matrizes, a operação de crossover foi definida como uma união entre o cruzamento monoponto com o cruzamento multiponto. A figura ilustra de forma mais clara como ocorre o cruzamento entre os cromossomos.

² $i = 0$, pois os indivíduos estão armazenadas em uma lista dinâmica onde o primeiro elemento possui o índice 0.

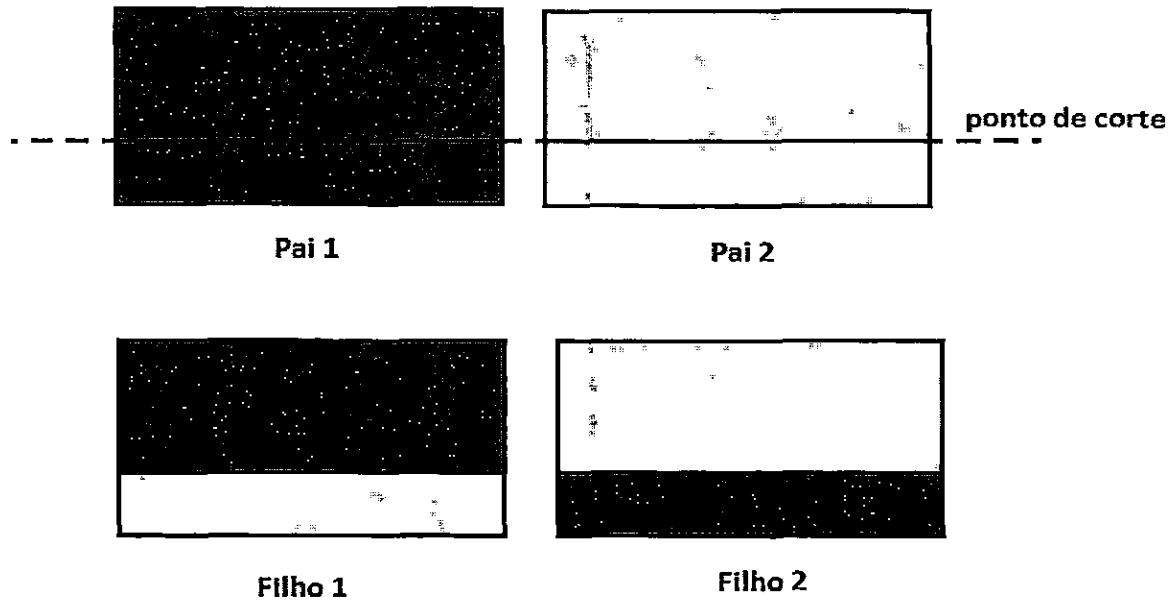


Figura 18: Exemplo de como ocorre o crossover.
Fonte: Primária

No crossover é escolhido aleatoriamente um ponto p denominado ponto de corte. Depois desta etapa os cromossomos recebem um corte horizontal como o mostrado na figura acima e troca-se o material genético dos dois pais gerando dois novos indivíduos chamados de filhos. O cruzamento pode não gerar filhos válidos, caso em que os pais ao trocarem material genético geram filhos que possuem professores com lecionando em duas turmas distintas no mesmo horário. Por isso foi implementada uma função no PEGH que analisa se o filho gerado é um filho válido (que gera horários sem choque), caso sim o indivíduo é inserido a nova população, caso não novos pais são escolhidos para serem submetidos ao cruzamento até que o tamanho da nova população seja igual ao tamanho da população inicial.

O cruzamento apesar de gerar dois filhos somente o melhor é escolhido para ser inserido na população, isso para garantir que a nova população tenha somente os bons resultados oriundos do crossover. Um detalhe importante a ser lembrado é que o crossover possui uma probabilidade de ocorrer de 90% e no momento da seleção dos pais é gerado um valor n aleatório entre $0 < n < 100$. Assim caso n possua um valor menor ou igual a 85% o crossover ocorrerá, essa probabilidade pode ser alterada configurando os parâmetros do GA no PEGH.

5.2.5.2 Mutação

O cruzamento é necessário nos algoritmos genéticos para que o filho gerado seja o resultado da troca de material genético entre os pais. Porém quando se alcança um determinado número de gerações a população pode começar a convergir para um cromossomo específico consequentemente a função a avaliação começa a possuir um mesmo valor. Para tornar o GA mais eficaz devemos aplicar o operador de mutação. Esse é responsável por introduzir variabilidade genética dentro da população e o faz da seguinte maneira:

- 1) Seleciona-se um valor aleatório que representará a linha da matriz grade que será alterada;
- 2) É gerado um array com 12 slots dispostos em ordem aleatória;
- 3) Troca-se a linha da matriz grade sorteada no item 1 pelo array gerado no item 2;
- 4) Depois disso verifica se o filho gerado pela mutação é válido ou não;
- 5) O filho é submetido a análise de validade. Se for válido, o filho é inserido na nova população, senão é descartado;

Como ocorre no cruzamento, a mutação também possui uma probabilidade de ocorrer, neste caso de 10% e no momento da seleção dos pais é gerado um valor n aleatório entre $0 < n < 100$ assim caso n possua um valor menor ou igual a 10% a mutação irá ocorrer, essa probabilidade pode ser alterada configurando os parâmetros do GA na aplicação. A figura ilustra o processo de mutação ocorrido por um indivíduo.

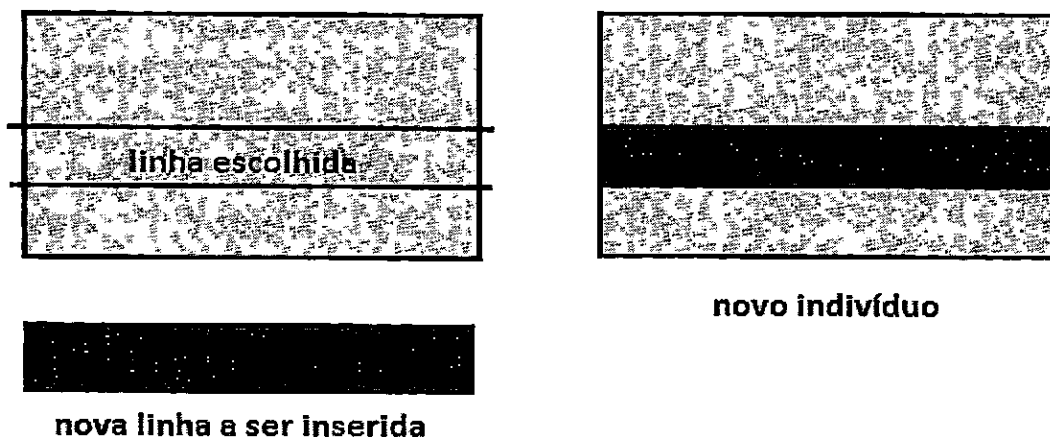


Figura 19: A mutação no PEGH.
Fonte: Primária

6 ANÁLISES DE RESULTADOS

O programa de escalonamento de grade horária foi desenvolvido para a elaboração da grade horária do curso de Bacharelado em Ciências da Computação – UESPI utilizando para a sua construção os conceitos de algoritmos genéticos. O problema da geração de grade horária é conhecido por sua complexidade devido ao fato de ter que atender a múltiplos objetivos, por isso fez-se necessário a utilização dos GAs para a automação desse problema.

A escolha do melhor indivíduo gerado na execução do algoritmo genético é baseada na função de avaliação que busca maximizar as preferências dos professores pelos slots por eles alocados. Assim o indivíduo dentro da população que tiver a melhor avaliação será a solução do problema. No programa proposto indivíduos considerados inaptos por apresentarem algumas restrições considerada graves, não são inseridos na população sendo esta também uma forma de seleção.

Nesta seção apresentaremos a análise dos resultados obtidos pelo PEGH – Programa Escalonador de Grade Horária mostrando também as plataformas de hardware e software, as telas do programa e seu funcionamento.

6.1 PLATAFORMAS DE HARDWARE E SOFTWARE

As plataformas de hardware e software utilizadas para desenvolvimento e bateria de testes da aplicação foi mantida constante no intuito de garantir que os resultados obtidos na execução não sofressem discrepância causada pela mudança de hardware. Assim na plataforma de hardware foi utilizado um notebook com a seguinte configuração:

- Processador Intel(R) Core(TM)2 Duo CPU T5550 com 1.83GHz de frequência.
- Placa-Mãe modelo JFW01 fabricada pela SiS com chipset 671/671DX/671FX.
- Memória Principal de 1792 MB DDR2.
- Disco Rígido de tecnologia IDE, modelo WDC WD1600BEVS-00UST0 com capacidade de 160 GB.
- Drive DVD-RAM UJ-850S fabricado pela MATSHITA
- Placa de vídeo SiS 351 Mirage 3 IGP

A plataforma de software é descrita abaixo:

- Sistema Operacional Ubuntu 10.04;

- OpenOffice.org 3.2;
- NetBeans IDE 6.8;
- SGBD PostgreSQL 8.4;

6.2 O APLICATIVO

O PEGH - Programa de Escalonamento de Grade Horária dispõe de uma série de menus para cadastramento, visualização e configuração dos parâmetros necessários para a geração da grade horária. Em sua tela principal são encontrados os seguintes componentes:

- Barra de Menu: Cadastro, Opções, Exibir, Ajuda;
- Controle de Execução: Gerar Grade, Cancelar Execução;
- Botão Sair;

A tela inicial do PEGH é ilustrada na figura abaixo.

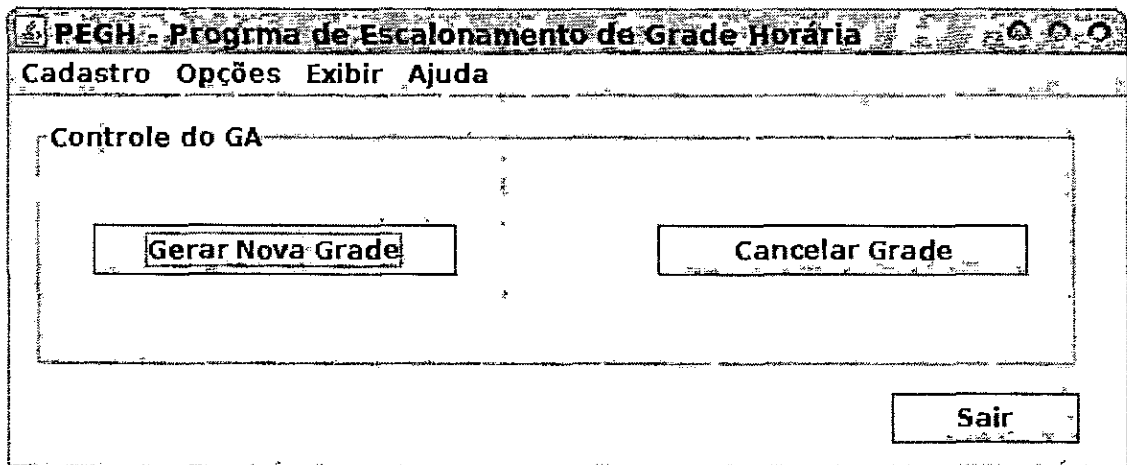


Figura 20: Tela Inicial do PEGH – Programa de Escalonamento de Grade Horária.
Fonte: Primária

Para gerar uma grade horária é necessária a configuração de alguns parâmetros que serão inseridos dentro do algoritmo genético. No momento na inicialização do PEGH o usuário tem a opção de utilizar as configurações da última grade gerada ou inicializar uma grade utilizando novos parâmetros. Caso o usuário escolha a primeira opção os professores continuam vinculados às disciplinas, as disciplinas continuam sendo lecionadas no mesmo turno e será gerada a grade para os períodos definidos na última configuração do GA. Caso a opção desejada pelo usuário seja a segunda é realizado um update na tabela de disciplinas no banco de dados onde as disciplinas perdem o vínculo com os professores e o campo correspondente ao id, identificador do de professor, na tabela de disciplinas recebe o valor

zero que corresponde ao professor “SEM PROFESSOR” na tabela de professores. Já o campo turno de todas as disciplinas é atualizado para “SEM TURNO”, isso para que o usuário comece com novos parâmetros, definindo os professores para as disciplinas e também o turno para estas.

Os parâmetros necessários para a configuração da nova grade horária são:

- A quantidade de indivíduos para a população do algoritmo genético;
- A quantidade de gerações que serão geradas;
- A probabilidade de ocorrer o crossover;
- A probabilidade de ocorrer a mutação;
- Os períodos que irão fazer parte da nova grade horária;

Os dados relativos do item (a) ao item (d), já vem com um parâmetro default que corresponde a quantidade mínima de cada item e pode ser alterado de acordo com a preferência do usuário e respeitando um limite máximo pré-estabelecido. Já no item (e) o usuário define os períodos para compor o semestre letivo do curso. A figura a seguir mostra a tela de configuração dos parâmetros do GA disponível em Menu > Opções > Configuração dos Parâmetros do GA.

Configuração de Parâmetros do PEGH

Parâmetros de Configuração do GA

No. Indivíduos: 150

No. Gerações: 10

Prob. Crossover: 85 %

Prob. Mutaçao: 15 %

Parâmetros Relativos a Grade

Períodos do curso: 1, 2, 3, 4, 5, 6, 7, 8

Períodos da Grade: 1, 3, 5, 8

>>

Remover da Grade

OK

Figura 21: Tela de Configuração dos parâmetros do GA.
Fonte: Primária

Também se faz necessário para a geração da grade horária pelo PEGH, a atualização das disciplinas onde serão definidos os turnos dos períodos e a vinculação dos professores nas disciplinas em que eles irão lecionar. Esta opção realiza um update na tabela de disciplinas e pode ser utilizada tanto para os usuários que queiram optar por utilizar a última configuração da grade quanto para os que optaram por gerar uma nova a partir de suas preferências. Disponível em Menu > Opções > Vincular os Professores as Disciplinas. A figura mostra a tela relativa a este item.

A imagem mostra uma janela de software com o título "Vincular Professor". O formulário contém os seguintes elementos:

- Um campo rotulado "Seledone o Perído:" com o valor "2" selecionado em uma caixa de lista suspensa.
- Um campo rotulado "Turno:" com o valor "NOITE" selecionado em uma caixa de lista suspensa.
- Um campo rotulado "Disciplinas:" com o valor "PROGRAMAÇÃO I" selecionado em uma caixa de lista suspensa.
- Um campo rotulado "Professor(a):" com o valor "FRANCISCO ROCHA" selecionado em uma caixa de lista suspensa.
- Dois botões de ação: "Vincular" e "Sair".

-Figura 22: Tela de Vinculação de Professor a Disciplina.
Fonte: Primária

O usuário também tem a opção de cadastrar um novo professor, uma nova disciplina ou uma nova disponibilidade para um professor. Estas opções encontram-se no Menu > Cadastro > Cadastro de Professor, Cadastro de Disciplina e Cadastro de Disponibilidade. Cada uma destas opções realizam uma inserção em suas respectivas tabelas no banco de dados.

Na opção Cadastro de Professor é gerado um id para o novo professor, e o usuário preenche o campo relativo ao nome do professor. A figura abaixo mostra a tela de cadastro de professores.

Figura 23: Tela de Cadastro de professor onde somente o nome do docente é preenchido pelo usuário. O PEGH gerará um valor para o id do docente.

Fonte: Primária

A opção Cadastro de Disciplinas gera um id para a nova disciplina e os seguintes campos precisam ser preenchidos:

- Nome: nome da disciplina;
- Sigla: seu apelido;
- Período: em que período a disciplina é ofertada;
- Turno: turno em que a disciplina é ministrada;
- Professor: O nome do professor que ministrará a disciplina. O usuário escolhe o professor e no momento do cadastro da disciplina na base de dados o id referente ao professor escolhido é inserido no campo professor desta disciplina. A tela de cadastro de disciplinas é ilustrada na figura abaixo.

Figura 24: Cadastro de Disciplinas, aqui o usuário cadastra as principais informações relativas a disciplina para a geração da grade horária.

Fonte: Primária

O Cadastro de Disponibilidade é onde inserimos as preferências dos professores pelos horários das aulas. O cadastro das disponibilidades é feito escolhendo o professor e atribuindo as preferências deste pelos horários de alocação das aulas. As preferências são os valores 0, 1 e 2 significando:

- 0: Horário Indisponível;
- 1: Horário Parcialmente Disponível;
- 2: Horário Disponível;

No caso das disponibilidades dos professores, caso já possua no banco de dados uma disponibilidade para o professor ao qual se quer cadastrar uma disponibilidade na tabela de disponibilidade é realizado um update nesta, atualizando os dados referentes ao professor ao qual se deseja cadastrar a disponibilidade. A figura abaixo mostra detalhes do cadastro de disponibilidade dos professores.

Cadastro de Disponibilidade

Professor:

Tarde						Sábado	
	Segunda	Terça	Quarta	Quinta	Sexta		
14:00 / 15:40	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	08:00 / 10:00	<input type="text"/>
15:50 / 17:30	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	10:00 / 12:00	<input type="text"/>

Noite						Legenda:	
	Segunda	Terça	Quarta	Quinta	Sexta	0 - Hora Indisponível	1 - Hora Parc. Disponível
18:30 / 20:10	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	2 - Hora Disponível	
20:20 / 22:00	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>		

Figura 25: Tela relativa ao cadastro das disponibilidades dos professores.

Fonte: Primária

Depois de realizados as inserções de parâmetros, cadastrados ou vinculados professores, disponibilidades e disciplinas, o programa esta pronto para gerar a grade horária, bastando que para isso seja clicado o botão gerar grade. Caso queira abortar a geração da grade é necessário e suficiente clicar o botão cancelar geração.

O PEGH também apresenta a opção de exibir a relação dos professores, disponibilidades e as disciplinas cadastradas no banco de dados do programa. Esta opção está disponível no Menu > Exibir > Relação de Professores, Relação de Disponibilidades e Relação de Disciplinas. As figuras X, Y e Z Mostram as telas de exibição de Professor, Disponibilidade e Disciplinas respectivamente.

Caso o usuário possua alguma dúvida de como gerar a grade horária o PEGH dispõe do item Ajuda encontrado no Menu onde ele será orientado a respeito dos passos a seguir para a elaboração de uma nova grade horária. A tela de ajuda é mostrada abaixo:

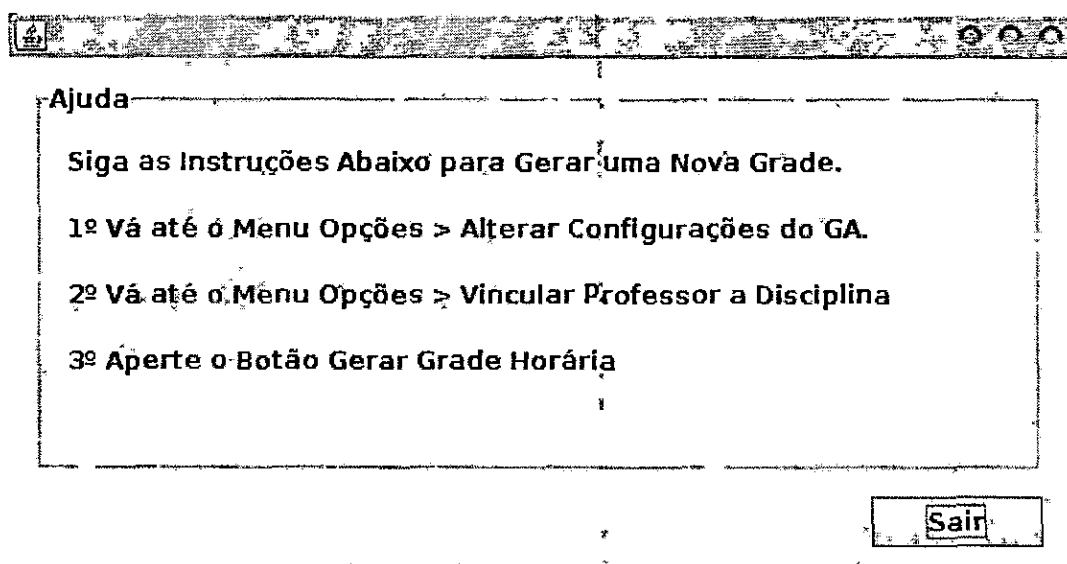


Figura 26: Tela de ajuda que mostrando como gerar uma nova grade horária.

Fonte: Primária

6.3 ANÁLISE E RESULTADOS OBTIDOS

Durante a implementação do PEGH uma série de testes foram realizados para verificar a eficiência das funções utilizadas no algoritmo. As principais funções testadas foram:

- A inicialização e codificação da população itens essenciais para o bom desempenho de todo o programa;
- No cruzamento foi avaliado o corte horizontal e o corte vertical, sendo o segundo descartado por ser inviável devido a grande quantidade de choques de horários;
- A avaliação do indivíduo;
- Teste de seleção por roleta;
- Quantidade ideal de indivíduos, gerações, probabilidade de crossover e mutação;

Inicialmente ao período de implementação os dados usados para os testes eram compostos de dados reais (disciplinas e professores) e fictícios (disponibilidade dos docentes), onde os dados reais eram referentes ao semestre letivo de 2010.1. Após a conclusão do algoritmo o mesmo foi testado com dados reais do curso de Bacharelado de Ciências da Computação semestre letivo 2010.2, dados estes cedidos pela coordenação do curso. Com os dados reais o PEGH foi executado com diferentes populações, gerações, probabilidade de crossover e mutação no intuito de se obter a melhor quantidade dos respectivos itens. Depois de cada execução o resultado era armazenado para comparações futuras com o objetivo de analisar o desempenho do programa com os parâmetros teste para que fosse encontrado o conjunto de valores satisfatórios para cada item.

O Programa de Escalonamento de Grade Horária foi executado no mesmo computador ao qual foi criado. Com os parâmetros considerados os mais eficientes o programa demorou em média 00:35 minutos para a geração da grade horária atual (semestre letivo 2010.2) do curso de Bacharelado em Ciências da Computação – UESPI. As quantidades dos parâmetros foram:

- Número de Indivíduos: 150
- Numero de Gerações: 10
- Probabilidade de Crossover: 85%
- Probabilidade de Mutação 5%

O programa alcançou melhores resultados aumentando a população e o número de gerações, o aumento é considerado logicamente em relação ao valor da função de avaliação. Porém esse aumento não é muito significativo e o tempo de processamento é muito maior em comparação aos parâmetros satisfatórios. A Figura a seguir ilustra o comparativo de desempenho com o tamanho da população e geração variáveis.

Comparativo de Desempenho				
Nº de Indivíduos	Nº de Gerações	Quantidade de Testes	Avaliação Média	Tempo Médio
150	10	10	247,5	00h37min
180	10	10	249	01h01min
200	15	10	249,5	01h13min

Figura 27: Tabela mostrando o Comparativo de Desempenho.

Fonte: Primária

A seguir são ilustrados os resultados do PEGH para a grade atual de cada período do curso de Bacharelado em Ciências da Computação – UESPI Campus Prof^o Alexandre Alves

de Oliveira.

1º Período							
	Segunda	Terça	Quarta	Quinta	Sexta	Sábado	
14:00/15:40	INGL TEC	LOG	INTR COMP	AL&GEO A	CALC I	CALC I	08:00/09:40
16:00/17:40	AL&GEO A	LOG	INGL TEC	AL&GEO A	INTR COMP	CALC I	10:00/11:40

Figura 28: Resultado da Execução do PEGH. Tabela referente ao 1º período
Fonte: Primária

3º Período							
	Segunda	Terça	Quarta	Quinta	Sexta	Sábado	
18:20/20:00	FIS I	FIS I	OS&M	LING II	LING II	LING II	08:00/09:40
20:20/22:00	EQ DIF	FIS I	ESTR	ESTR	EQ DIF	OS&M	10:00/11:40

Figura 29: Resultado da Execução do PEGH. Tabela referente ao 3º período.
Fonte: Primária

5º Período							
	Segunda	Terça	Quarta	Quinta	Sexta	Sábado	
14:00/15:40	BD I	SO I	ARQ	SO I	COM I	PESQ E ORD	08:00/09:40
16:00/17:40	ARQ	ARQ	BD I	PESQ E ORD	COM I	COM I	10:00/11:40

Figura 30: Resultado da Execução do PEGH. Tabela referente ao 5º período.
Fonte: Primária

8º Período						
	Segunda	Terça	Quarta	Quinta	Sexta	Sábado
18:20/20:00	SD	TEO GRA	SD	SEG E AUD	TELECOM	TEPC II
20:20/22:00	SEG E AUD	TELECOM	GTI	GTI	TEO GRA	TEPC II

08:00/09:40
10:00/11:40

Sair

Figura 31: Resultado da Execução do PEGH. Tabela referente ao 8º período.

Fonte: Primária

7 CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho propôs a implementação de um programa gerador de grade horária para o curso de Bacharelado em Ciências da Computação – UESPI Campus de Parnaíba utilizando para este fim os principais conceitos referentes aos algoritmos genéticos. Mostrou-se as principais características relativas ao GAs e complexidade do problema para a geração de uma solução viável e satisfatória devido a grande quantidade de objetivos, restrições e preferências a serem atendidas e levadas em consideração.

O aplicativo criado tem por objetivo reduzir o tempo e o esforço requerido para a elaboração da grade horária específica do curso e depois da análise dos resultados obtidos pelo programa usando dados reais, pode-se concluir que a utilização dos algoritmos genéticos para a implementação do programa favorece a confecção de uma solução viável e eficiente para o problema de alocação de horários para os professores do curso, automatizando o processo e o tornando mais simples e eficiente se comparado ao trabalho manual.

Como contribuição deste trabalho, destacam-se os principais conceitos relativos aos algoritmos genéticos, ramo da inteligência artificial bastante estudado e preferido para a resolução de problemas de otimização e a elaboração de um programa que utiliza esses conceitos e que busca a resolução de um problema de restrições, que trabalha com uma grande quantidade de variáveis e que deve respeitar uma quantidade significativa de restrições impostas, mostrando essa tecnologia se bem estruturada e implementada constituem uma solução bastante viável para a resolução do problema proposto.

Como trabalho futuro, sugere-se a implementação de um programa de geração de grade horária genérico para a UESPI podendo este ser utilizado pelas coordenações de todos os cursos da instituição. Para isso deve ser analisada a problemática específica de cada curso como foi feito com o estudo de caso deste trabalho.

Os resultados oriundos da execução do PEGH são satisfatórios não apresentando restrições graves e infringindo uma pequena quantidade de restrições leves.

REFERÊNCIAS BIBLIOGRÁFICAS

- DEITEL, Harvey M.; DEITEL, Paul J. **Java: Como Programar**. 6 ed. Brasil.[s.n.], Prentice Hall, 2005.
- GOLDBERG, David E. **Genetic Algorithms in Search, Optimization and Machine Learning**. 1 ed. Boston: Addison-Wesley Longman Publishing Co., Inc., 1989.
- HAMAWAKI, Cristiane Divina Lemos. **Geração Automática de Grade Horária Usando Algoritmos Genéticos: O Caso da Faculdade de Engenharia Elétrica da UFU**. Uberlândia: Universidade Federal de Uberlândia. 2005, 90 p. Disponível em: <http://www.bdtd.ufu.br/tde_busca/arquivo.php?codArquivo=230>. Acesso em 17 de junho de 2010.
- HOLLAND, John H. **Adaptation in Natural and Artificial Systems**. 1 ed. Boston: The MIT Press, 1992.
- KOZA, John R. **Genetic Programming: On the Programming of Computers by Means of Natural Selection**. 6 ed. Boston: The MIT Press, 1998.
- LINDEN, Ricardo. **Algoritmos Genéticos: Uma importante ferramenta da Inteligência Computacional**. 2 ed. Rio de Janeiro: Brasport, 2008.
- LUCAS, Diogo Correa. **Algoritmos Genéticos: um estudo de seus conceitos fundamentais e aplicação no problema de grade horária**. Pelotas: Universidade Federal de Pelotas. 2000, 66 p. Disponível em: <www.ufpel.tche.br/prg/sisbi/bibct/acervo/info/2000/Mono-Diogo.pdf>. Acesso em 21 de junho de 2010.
- NETO, Álvaro P. **PostgreSQL – Técnicas Avançadas: Versões Open Source 7.x e 8.x. Soluções para Desenvolvedores e Administradores de Banco de Dados**. 4 ed. São Paulo: Érica, 2007.
- REEVES, Colin R.; ROWE, Jhonathan E. **Genetic Algorithms: Principles and Perspectives. A Guide to GA Theory**. New York: Kluwer Academic Publishers, 2003.
- SIERRA, Kathy; Bates, Bert. **Use a Cabeça Java**. 2 ed. Brasil.[s.n.], Alba Books, 2005.