

Sistema Gerador de Grades horárias para Instituições de ensino

Thomás Augusto Preis

Orientador: Mauro Marcelo Mattos

Sumário

- Introdução
- Fundamentação teórica
- Desenvolvimento do trabalho
- Implementação
- Conclusão



Introdução

- Problemas na construção de grades horárias
 - Dificuldades
 - Complexidade
- Motivação
 - Desafio
 - Estudar funcionamento dos algoritmos genéticos



Objetivos gerais

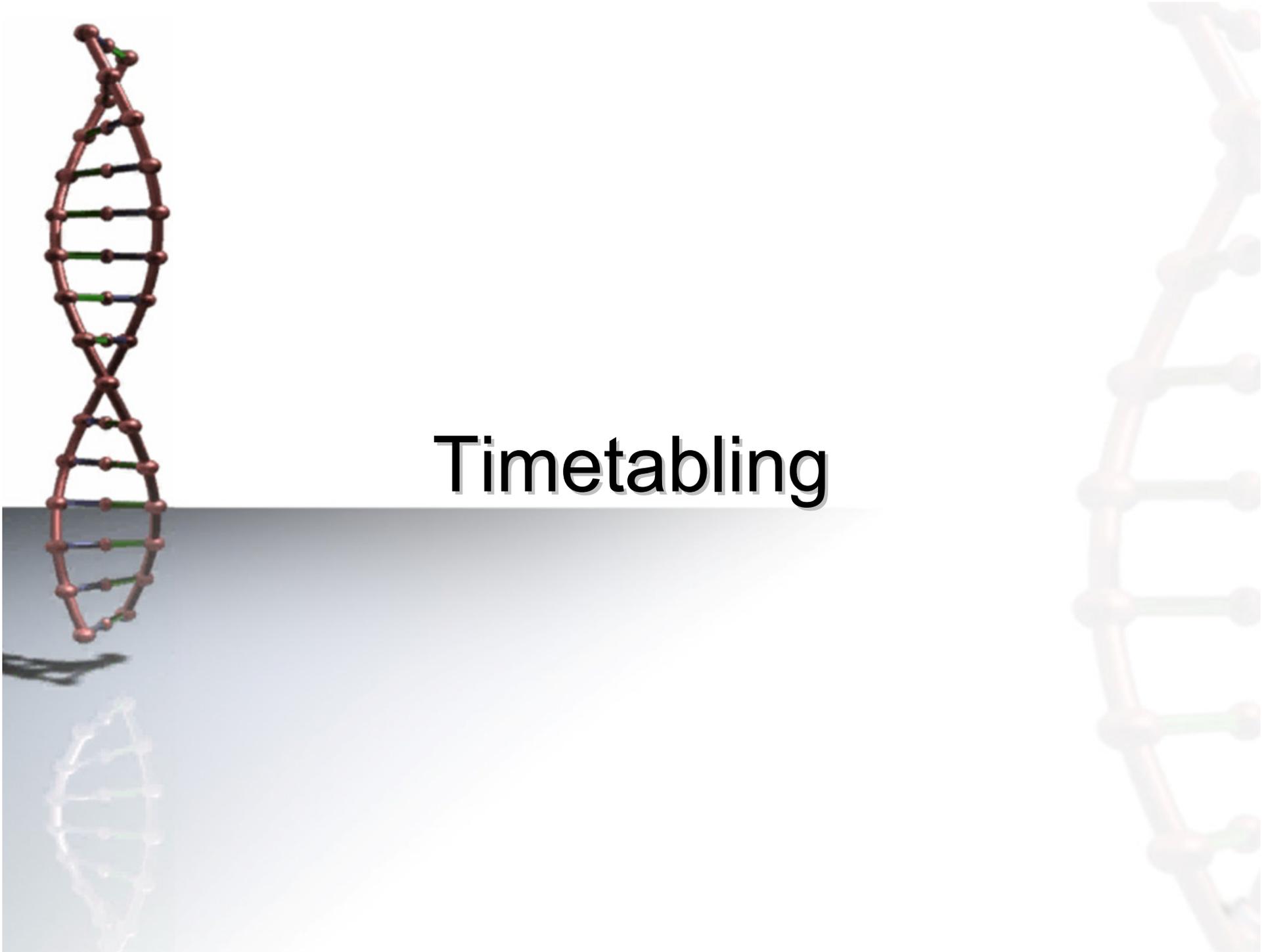
- Construção de um protótipo para gerar sugestões de grades horárias



Objetivos específicos

- Utilizar um algoritmo de alocação baseado em algoritmos genéticos para gerar as grades horárias
- Disponibilizar uma interface web





Timetabling

Timetabling

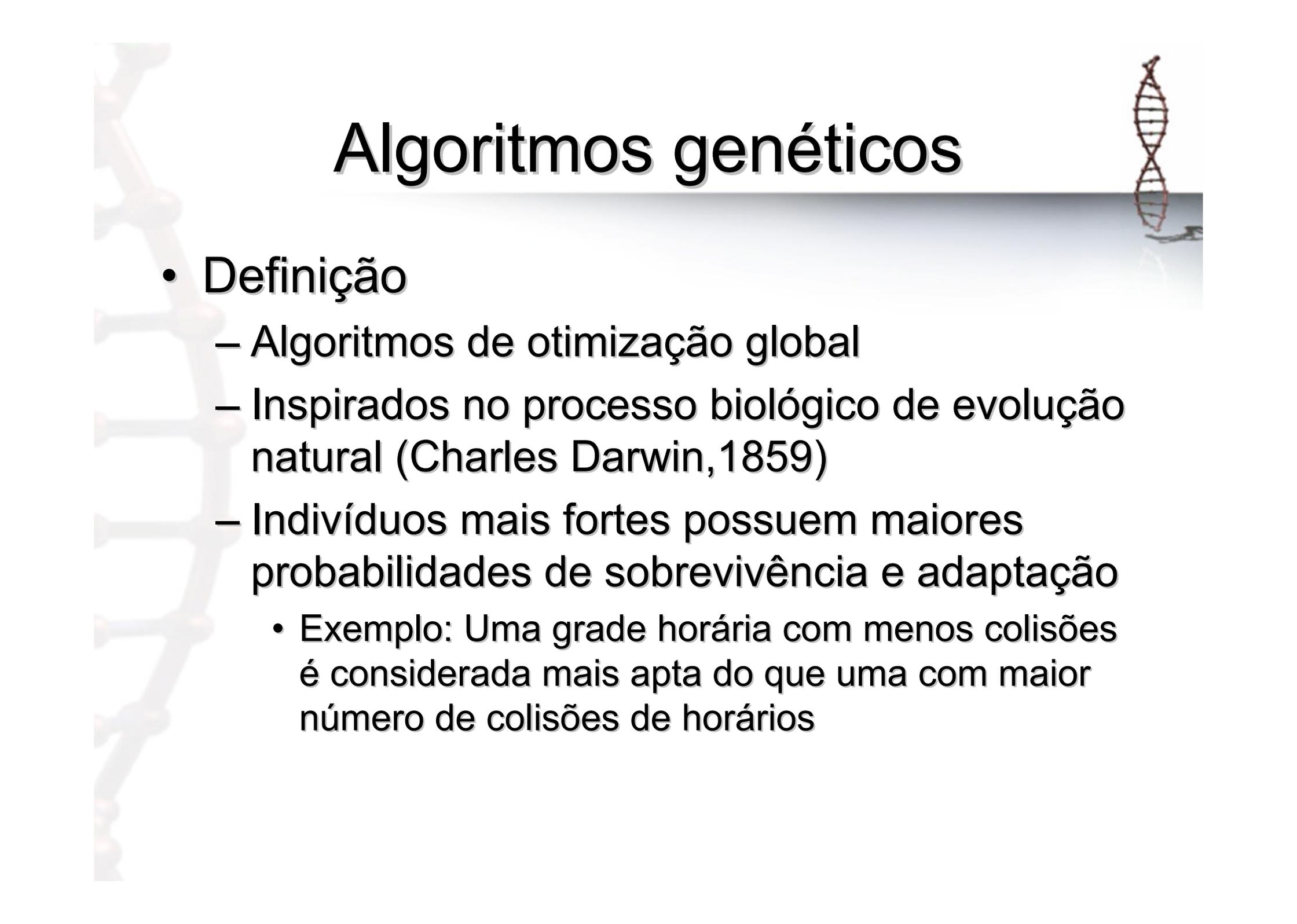


- Definição
 - O que é?
- Características
 - Utilizam heurísticas para buscar as soluções
 - Otimização combinatória
 - Não existem métodos universais para solucionar este tipo de problema
- Exemplos
 - Escalonamento de enfermeiros
 - Horário de aulas
 - Planejamento de transporte público



Algoritmos genéticos

Algoritmos genéticos



- Definição

- Algoritmos de otimização global
- Inspirados no processo biológico de evolução natural (Charles Darwin, 1859)
- Indivíduos mais fortes possuem maiores probabilidades de sobrevivência e adaptação
 - Exemplo: Uma grade horária com menos colisões é considerada mais apta do que uma com maior número de colisões de horários

Características



- Trabalham com um conjunto de soluções
- Avaliação de indivíduos é feita separadamente
- Busca cega
- Facilidade no uso de restrições

Termos utilizados



Linguagem natural	Algoritmos genéticos
Cromossomo	Individuo, <i>string</i> , cromossomo, árvore
Gen	Característica
Alelo	Valor
Locus	Posição
Genótipo	Estrutura
Fenótipo	Conjunto de parâmetros

Representação



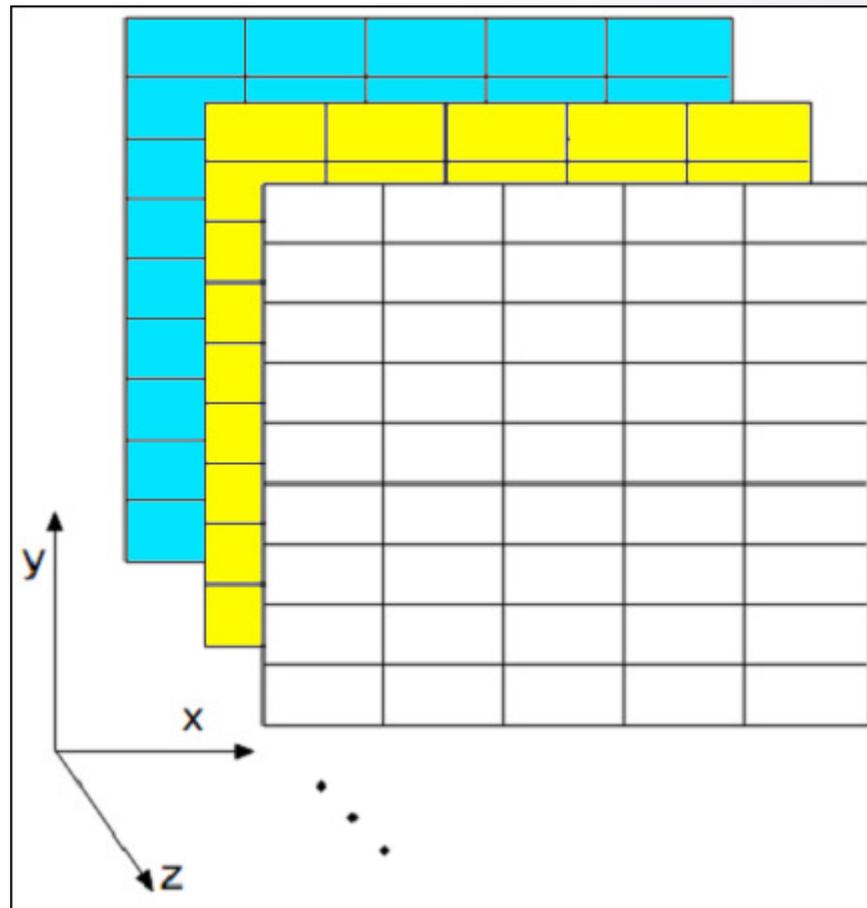
- Representação dos indivíduos
 - Criar estrutura de dados para representar os cromossomos
- Importância de uma boa representação
 - Desempenho
 - Otimizar e facilitar manipulações
- Características para uma boa representação
 - Completude
 - Possibilidade de representar todas as soluções possíveis
 - Coerência
 - Capacidade de gerar novos genótipos não pertencentes a população

Representação

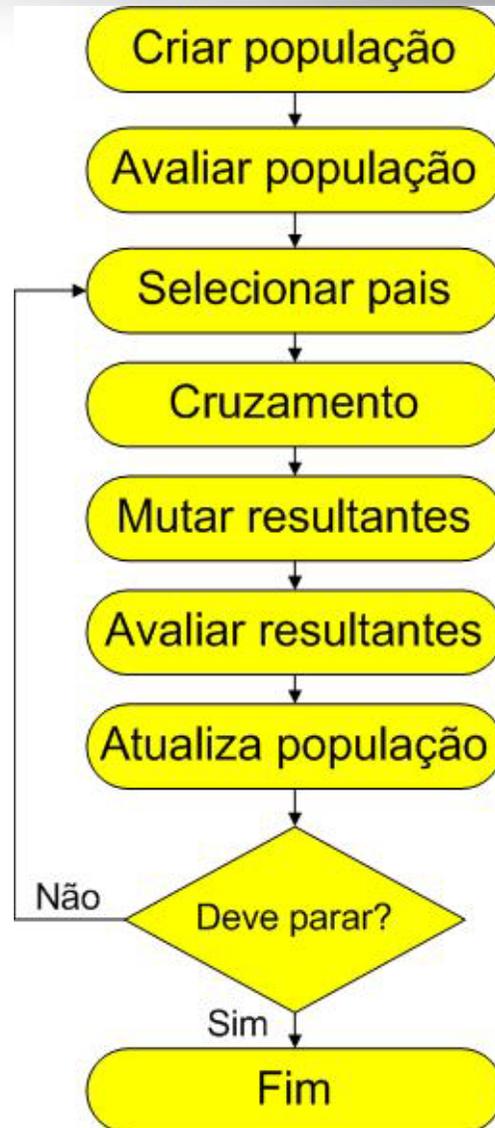


- Características para uma boa representação
 - Uniformidade
 - Fenótipos diferentes geram os mesmos genótipos
 - Simplicidade
 - Grau de complexidade de codificação e decodificação
 - Localidade
 - Pequenas alterações na estrutura dos indivíduos geram poucas alterações nos fenótipos

Representação do problema de timetabling



Funcionamento



Criar população



- Criação de um conjunto de possíveis soluções
 - Soluções válidas ou não
- Funções aleatórias
- Tamanho da população
 - Biodiversidade
 - Maior espaço de busca
 - Desempenho

Avaliar população



- Determinar o grau de aptidão dos cromossomos
- Funções de avaliação
 - Determinam a distância da solução do problema
- Principais tipos de avaliação
 - Distância máxima ou mínima
 - Valor máximo ou mínimo
 - Penalidades

Selecionar pais

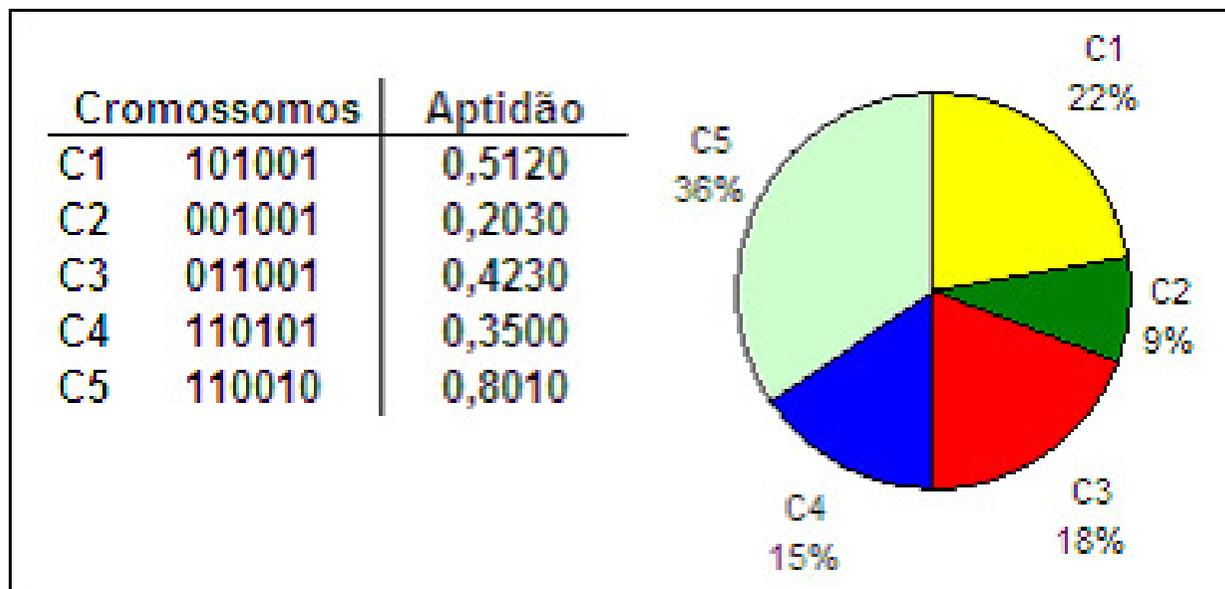


- Selecionar membros para reprodução
- Principais tipos de seleção
 - Seleção por ranking
 - Indivíduos ordenados conforme seus graus de aptidão
 - Torneio
 - Conjunto de soluções escolhidas sucessivamente e as mais aptas em cada um dos conjuntos são selecionadas
 - Uniforme
 - Todos os membros possuem a mesma probabilidade de serem escolhidos

Selecionar pais



- Principais tipos de seleção (continuação)
 - Roleta
 - Membros mais aptos possuem maiores chances de serem escolhidos



Cruzamento ou *crossover*

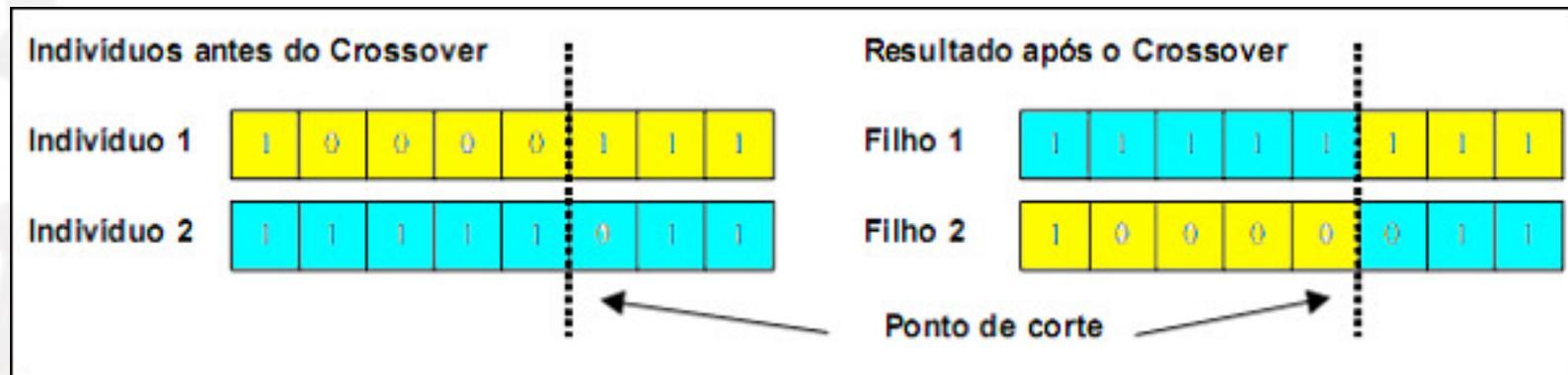


- Gerar novos indivíduos (filhos) através da troca do material genético dos membros selecionados
- Principais tipos
 - Ponto único
 - Multiponto
 - Segmentado
 - Uniforme

Cruzamento ou *crossover*



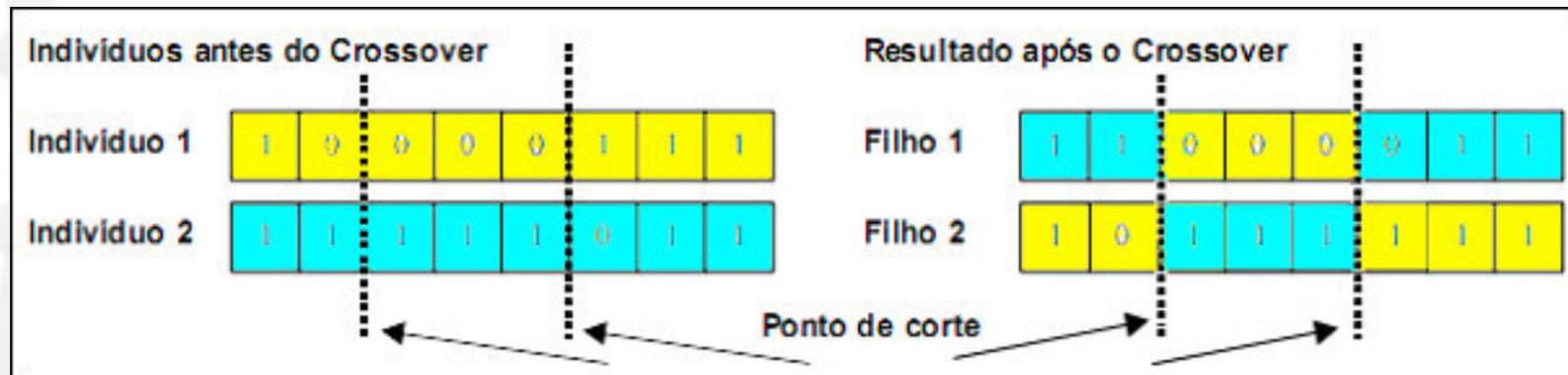
- Cruzamento de ponto único



Cruzamento ou *crossover*



- Cruzamento multiponto



Mutações



- Alterações nas estruturas dos indivíduos resultantes dos cruzamentos
 - Contribuem para não ocorrência de estagnação da população
 - Auxiliam no direcionamento da população para a solução desejável

Mutações



- Principais tipos de mutações
 - Aleatória
 - Genes recebem novos valores, dentro dos valores possíveis
 - Mutação por troca
 - Trocada de pares de genes
 - Mutação por *creep*
 - Alteração nos valores dos genes através de operações matemáticas

Avaliação, atualização e remoção



- Avaliação
 - Indivíduos resultantes submetidos as funções de avaliação
- Atualização
 - Novos membros na população são adicionados a população
- Remoção
 - Velhos membros da população são removidos
 - Não implementada por todos os algoritmos

Verificação de parada



- Verificar se o algoritmo chegou a solução desejável
 - Finalizar devido à um estouro de tempo ou número de gerações
 - Voltar para o passo onde é feita a seleção dos pais

Parâmetros do AG

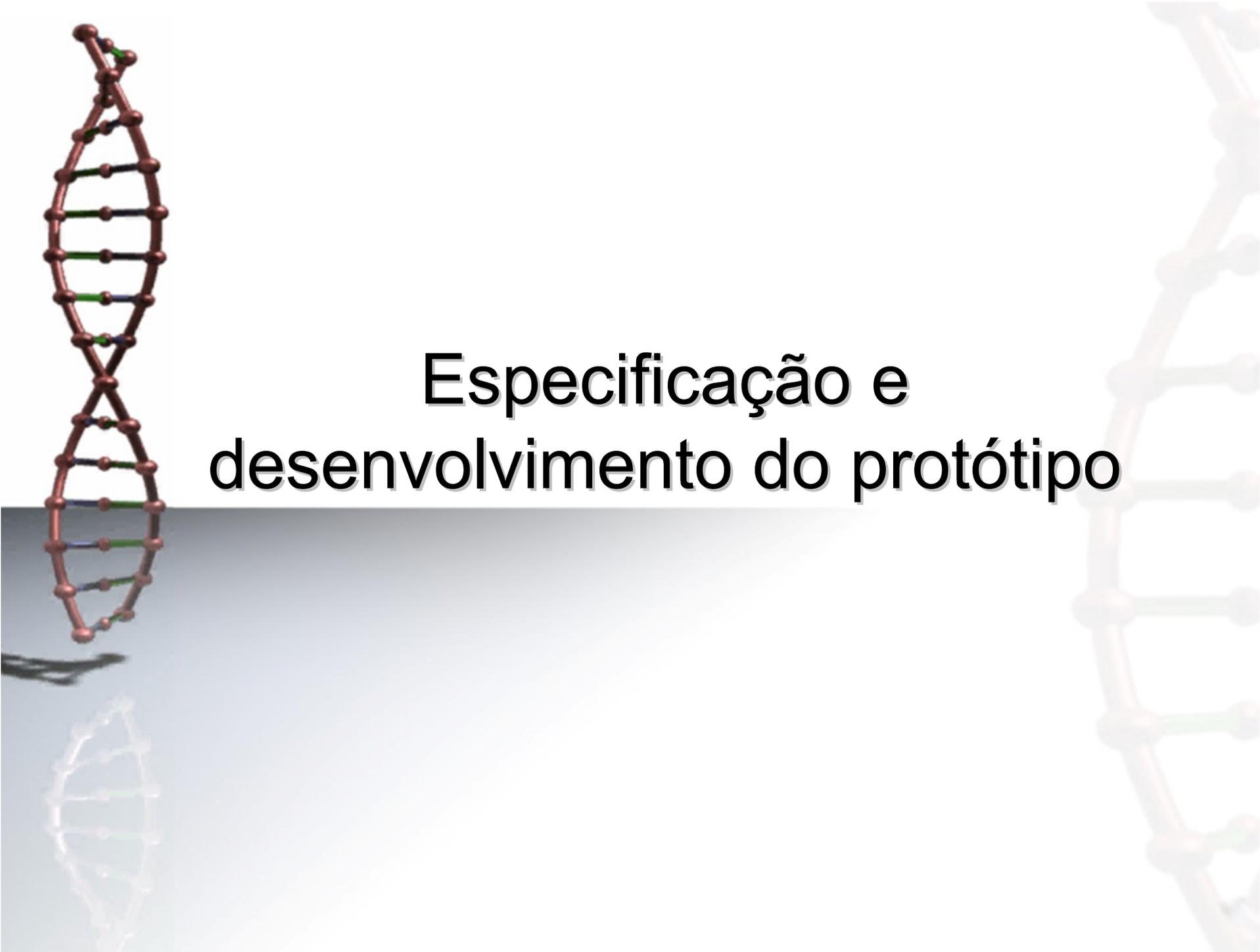


- Alteram o desempenho e capacidade do AG chegar a solução desejável
- Principiais parâmetros
 - Tamanho da população da população inicial
 - Taxa de cruzamento
 - Taxa de mutação
 - Intervalo de geração
 - Número máximo de gerações

Aplicações

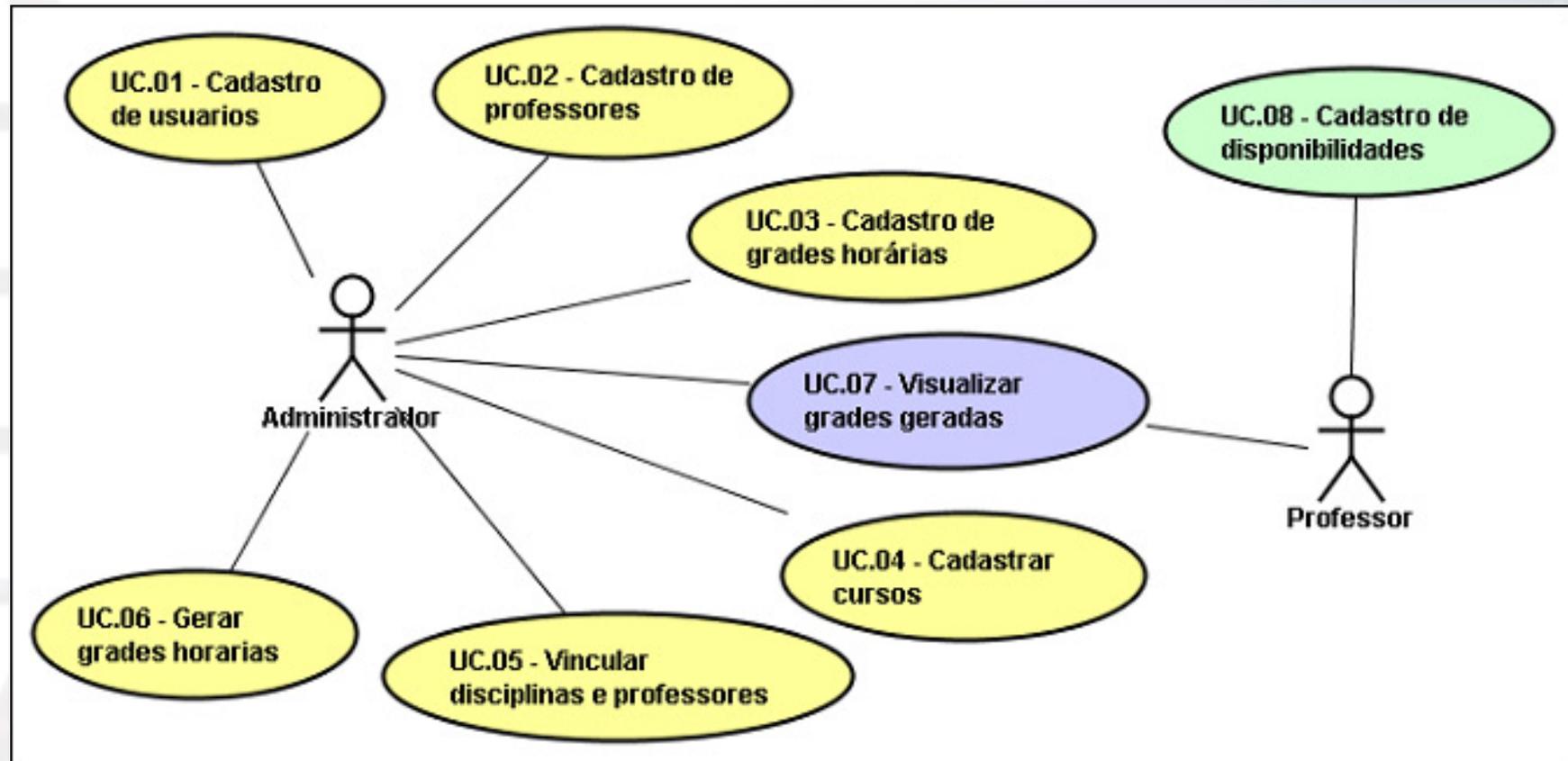
- Síntese de circuitos analógicos
- Gerenciamento de redes
- Computação evolutiva
- Problemas da otimização
- Ciências biológicas





Especificação e desenvolvimento do protótipo

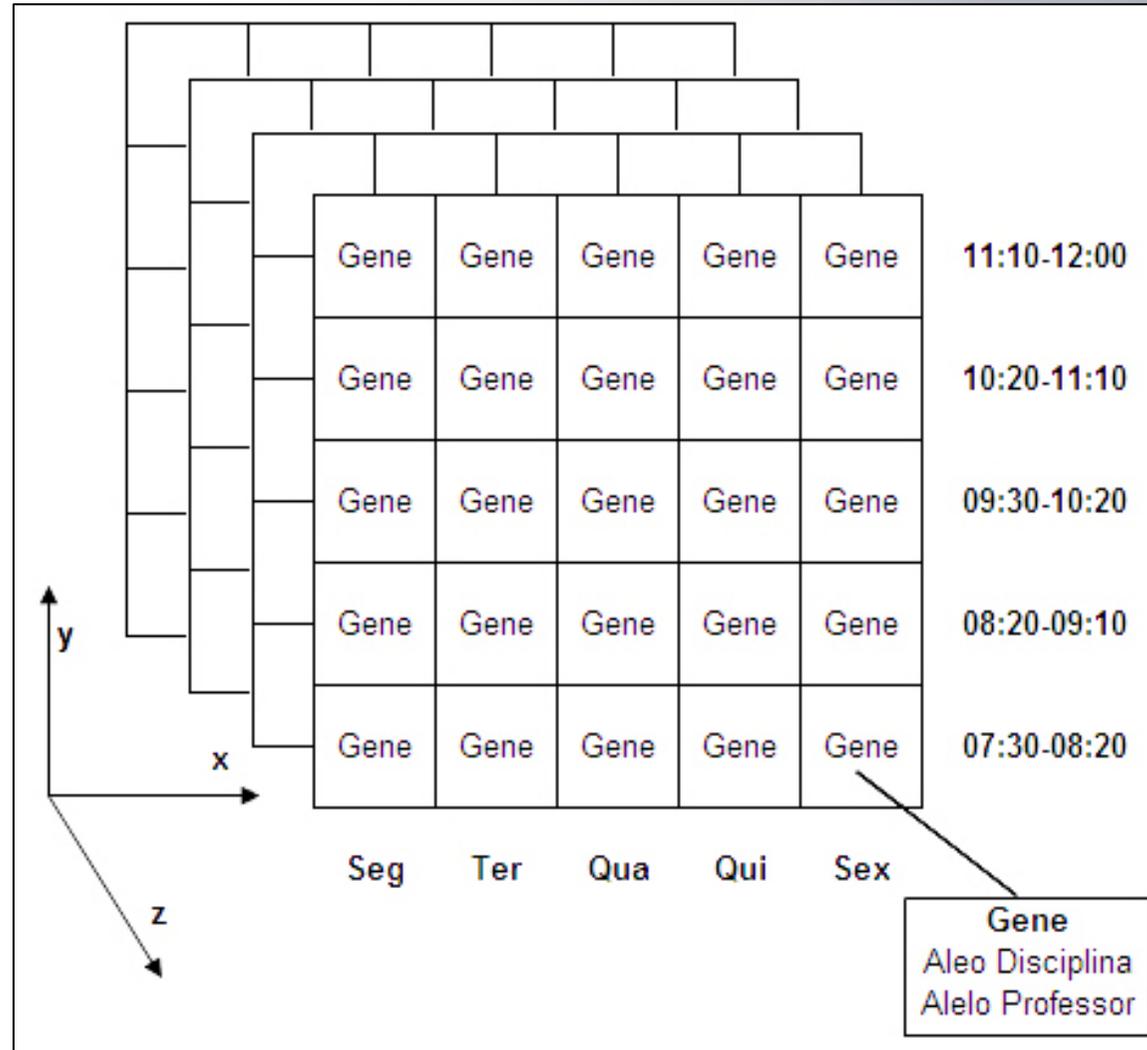
Casos de uso





Implementação do AG

Representação



Inicialização da população



- Inicialização parcialmente aleatória
 - Restrições de professores e disciplinas
 - Grade horária completa
 - Não ocorrência de repetição de disciplinas em um mesmo dia

Funções de avaliação



- Método baseado em penalidades
 - Caracterização de uma penalidade
 - Mesmo professor alocado em um determinado horário em mais de um semestre

Semestre 1					Semestre 2						
D2-P5	D4-P6	D2-P5	D1-P1	D4-P6	H4	D6-P2	D8-P7	D9-P3	D8-P7	D7-P5	H4
D2-P5	D4-P6	D2-P5	D1-P1	D4-P6	H3	D6-P2	D8-P7	D9-P3	D8-P7	D7-P5	H3
D1-P1	D3-P3	D4-P6	D3-P3	D2-P5	H2	D5-P1	D7-P5	D6-P2	D5-P1	D9-P3	H2
D1-P1	D3-P3	D4-P6	D3-P3	D2-P5	H1	D5-P1	D7-P5	D6-P2	D5-P1	D9-P3	H1
Seg	Ter	Qua	Qui	Sex		Seg	Ter	Qua	Qui	Sex	

Funções de avaliação



Função de avaliação:

$$G = \frac{1}{(1 + Np)}$$

tal que: $G = \text{Grau de aptidão}$
 $Np = \text{Número de penalidades}$

Exemplo:

$$G = \frac{1}{(1 + Np)} \quad \text{onde: } Np = 4$$
$$G = \frac{1}{(1 + 4)} \quad \text{então: } G = 0,2$$

Selecionar membros



- Método da roleta
 1. Soma dos graus de aptidão da população
 2. Atribuição das fatias da roleta para cada membro
 3. Sorteado número aleatório variando de zero à soma dos graus de aptidão
 4. Selecionar membro correspondente ao número sorteado

Operadores de cruzamento



- Métodos implementados
 - Cruzamento de ponto único
 - Relativamente ineficiente
 - Redução da biodiversidade
 - Produz diversos indivíduos geneticamente iguais
 - Cruzamento multiponto
 - Eficiente
 - Garante uma boa biodiversidade
 - Baixa ocorrência de indivíduos geneticamente iguais na população

Mutação



- Método implementado
 - Mutação por troca

Semestre						Semestre'						
D2-P5	D4-P6	D2-P5	D1-P1	D4-P6	H4	→	D3-P3	D4-P6	D2-P5	D1-P1	D4-P6	H4
D2-P5	D4-P6	D2-P5	D1-P1	D4-P6	H3		D3-P3	D4-P6	D2-P5	D1-P1	D4-P6	H3
D1-P1	D3-P3	D4-P6	D3-P3	D2-P5	H2	Mutações	D1-P1	D3-P3	D4-P6	D2-P5	D2-P5	H2
D1-P1	D3-P3	D4-P6	D3-P3	D2-P5	H1	→	D1-P1	D3-P3	D4-P6	D2-P5	D2-P5	H1
Seg	Ter	Qua	Qui	Sex			Seg	Ter	Qua	Qui	Sex	

Avaliação, remoção e inclusão

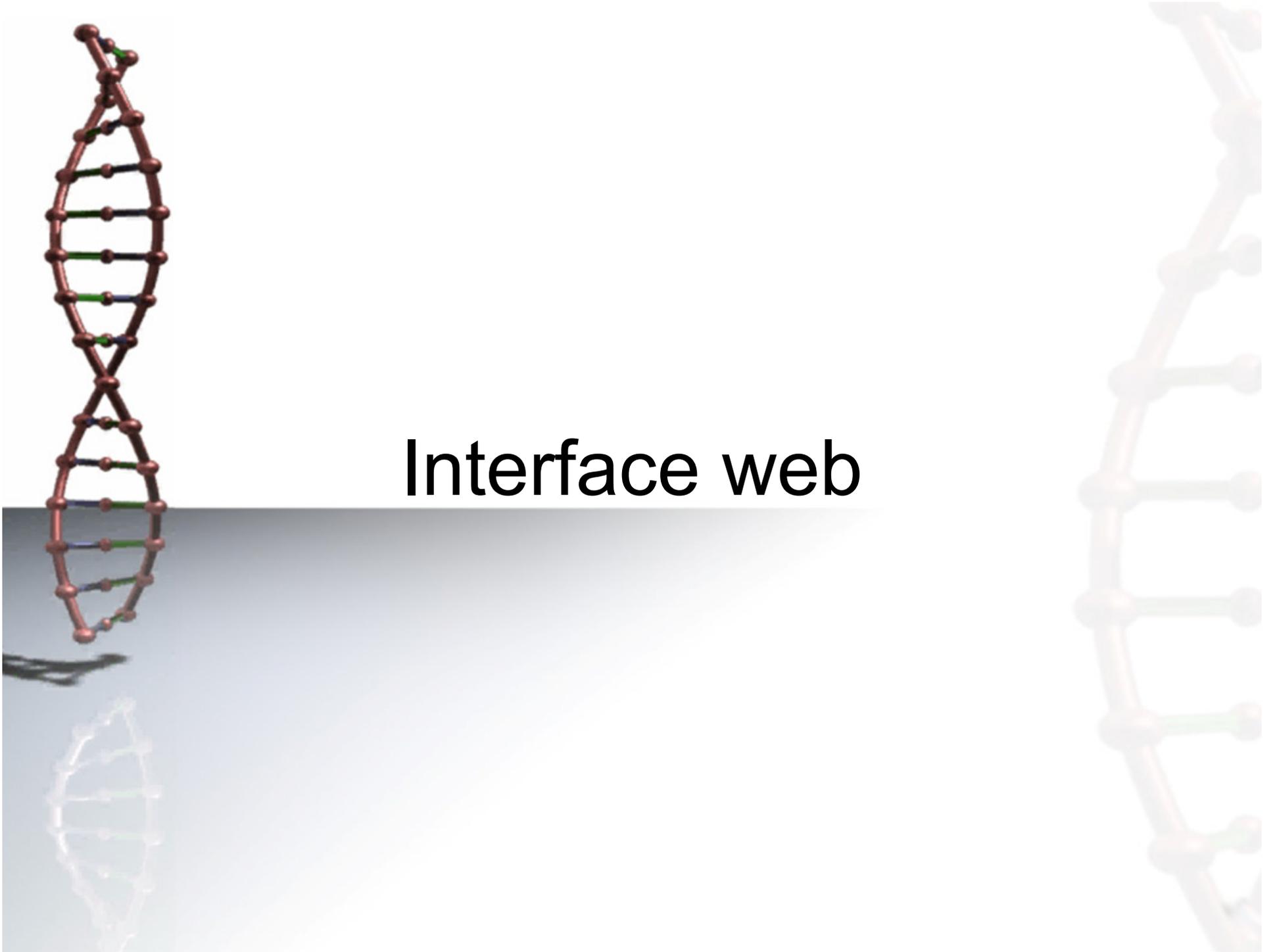


- Avaliação
 - Novo membro submetido as funções de avaliação
- Remoção
 - Feita uma varredura na população, removendo-se o membro menos apto
- Inclusão
 - Incluso novo membro na população

Verificação de parada



- Verifica se há um individuo com grau de aptidão igual a 1
 - Caso seja satisfeita a condição de parada, o algoritmo finaliza sua execução
 - Caso contrário o algoritmo reiniciará o processo na fase de avaliação da população

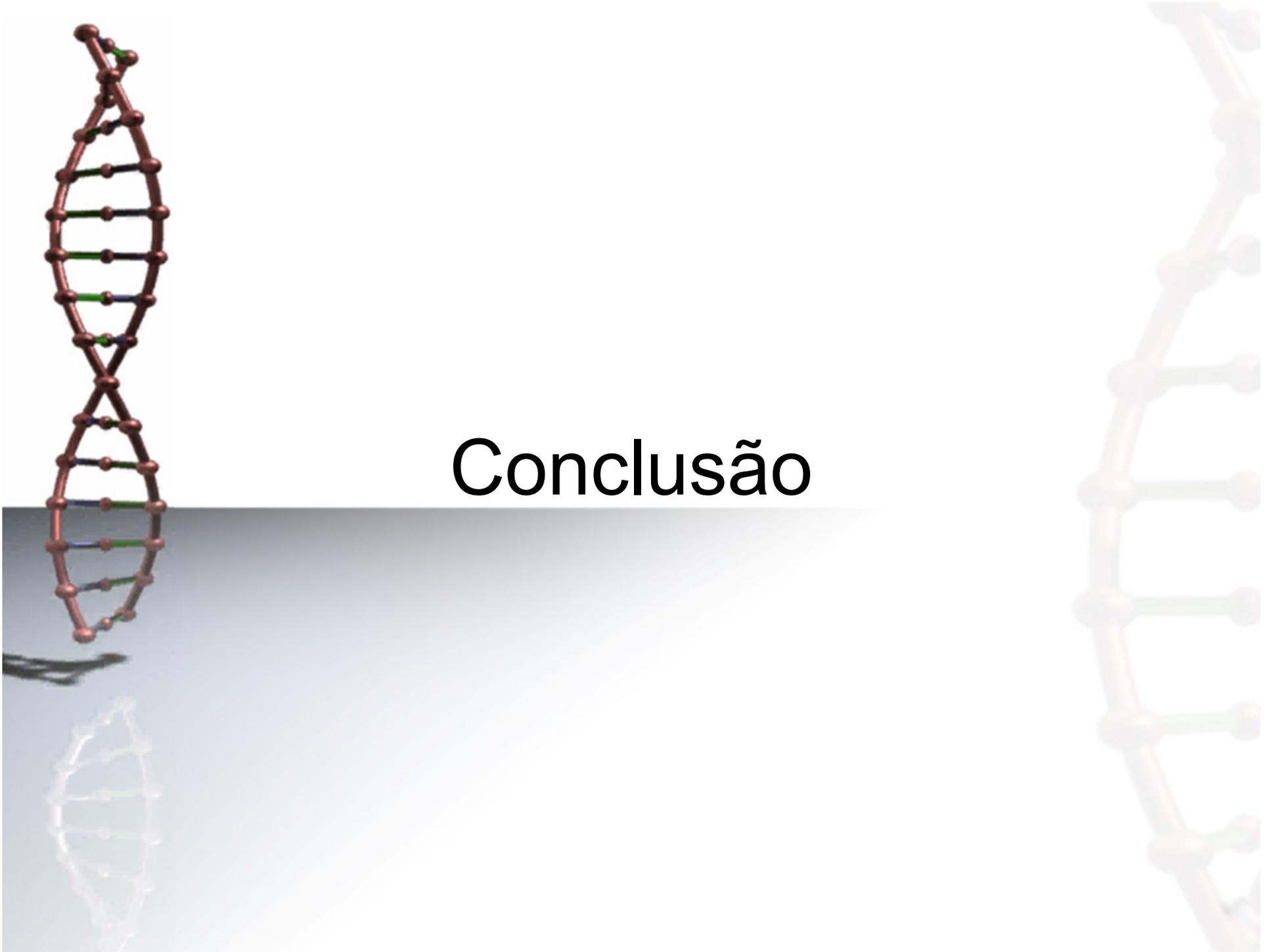


Interface web

Interface web

- Code Charge Studio 3.1
- Eclipse 3.2
- Case Studio
- MySQL 5.0
- Java 1.5
- Servlets





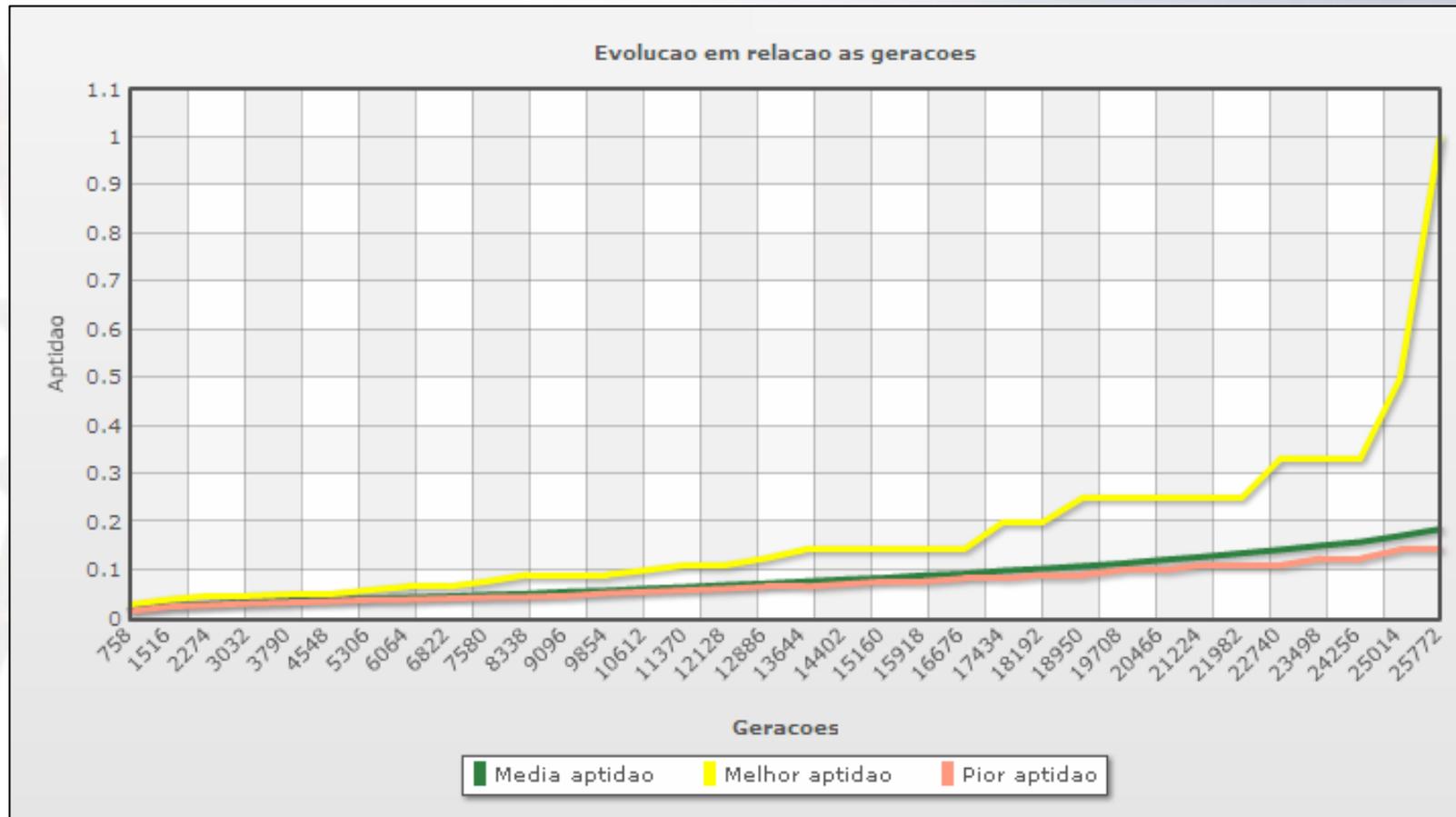
Conclusão

Resultados e discussão



Gerações		Taxa mutações	Tamanho população	Melhor solução	Colisões	Tempo
Máximo	Realizadas					
10000	5064	0,0015	100	1,000000	0	00:19
10000	10000	0,0015	300	0,200000	4	00:34
10000	10000	0,0015	500	0,083333	11	00:38
20000	20000	0,0015	100	0,500000	1	00:59
20000	20000	0,0015	300	0,500000	1	00:58
20000	19316	0,0015	500	1,000000	0	01:02
30000	30000	0,0015	100	0,500000	1	01:40
30000	15645	0,0015	300	1,000000	0	00:50
30000	20311	0,0015	500	1,000000	0	01:03
10000	5619	0,0500	100	1,000000	0	00:20
10000	10000	0,0500	300	0,166667	5	00:29
10000	10000	0,0500	500	0,142857	6	00:35
20000	4149	0,0500	100	1,000000	0	00:15
20000	14967	0,0500	300	1,000000	0	00:56
20000	20000	0,0500	500	0,500000	1	01:11
30000	4680	0,0500	100	1,000000	0	00:18
30000	7791	0,0500	300	1,000000	0	00:27
30000	30000	0,0500	500	0,500000	1	01:32
10000	10000	0,1500	100	0,500000	1	00:50
10000	10000	0,1500	300	0,166667	5	00:51
10000	10000	0,1500	500	0,090909	10	00:52
20000	20000	0,1500	100	0,500000	1	01:50
20000	20000	0,1500	300	0,333333	2	01:44
20000	20000	0,1500	500	0,142857	6	01:53
30000	19168	0,1500	100	1,000000	0	01:48
30000	30000	0,1500	300	0,500000	1	02:40
30000	30000	0,1500	500	0,500000	1	03:13

Resultados e discussão



Conclusão



- Sistema gerador de grades horárias implementado
- Requisitos levantados foram alcançados
- Geração de grades horárias
 - Eficiente na geração de grades sem colisão
 - Pouco eficiente na geração de grades obedecendo restrições de horários dos professores

Sugestões e extensões



- Sugestões
 - Alterações nos métodos responsáveis pelas mutações
- Extensões
 - Distribuição de salas de aula
 - Laboratórios
 - Equipamentos



Obrigado!