

Estudantes:

Caio Phillipe Mizerkowski – GRR20166403

Caio Filus – GRR20172172

João Marcelo – GRR20172162

Questões Teóricas:

1. What is evolutionary computation?

Computação evolutiva é inspirada pela evolução natural e tem como objetivo replicar processos da evolução natural. Os sistemas que usam a computação evolutiva funcionam com geração de soluções aleatórias viáveis, com base nos parâmetros fornecidos, porém não necessariamente boas, imposição de variações aleatórias a essas soluções e seleção dos indivíduos mais relevantes para os parâmetros desejados, repetindo o processo com os novos indivíduos para ser repetido o processo e cada vez se aproximar do resultado desejado.

2. Explain some basic concepts and terms related to genetic algorithm with binary representation i.e. population, chromosome, gene, allele, fitness function, and genetic operators.

População:

É um subconjunto de todas as soluções possíveis (indivíduos) que serão usados para criar uma geração de indivíduos para análise.

Cromossomo:

Cada uma das possíveis soluções para o problema dado. Representando as características de cada indivíduo.

Gene:

Cada posição de elemento no cromossomo.

Alelo:

Cada valor que um gene assume. Representado pelo valor atribuído a uma variável.

Fitness Function:

A função de aptidão associa uma pontuação para cada fenótipo na população.

Operadores Genéticos:

Operações a serem feitas para modificar a população, criando mais variedade para as gerações, podendo assim abranger uma maior variedade de possíveis soluções, por exemplo, crossover single-point, two-points, n-points, uniforme e mutações.

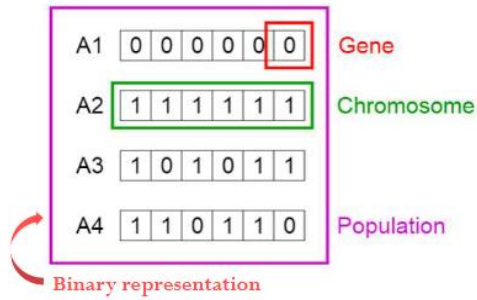


Figura 1 - Binary representation

3. Compare the single-point and two-point crossover in a genetic algorithm with binary representation.

Single-point vs Two-points:

Enquanto no single-point é selecionado um ponto aleatório para separar as duas cadeias de bits e haver a troca, enquanto no two-points, são dois pontos selecionados aleatoriamente, aumentando a variedade genética disponível para ser selecionada para a próxima geração.

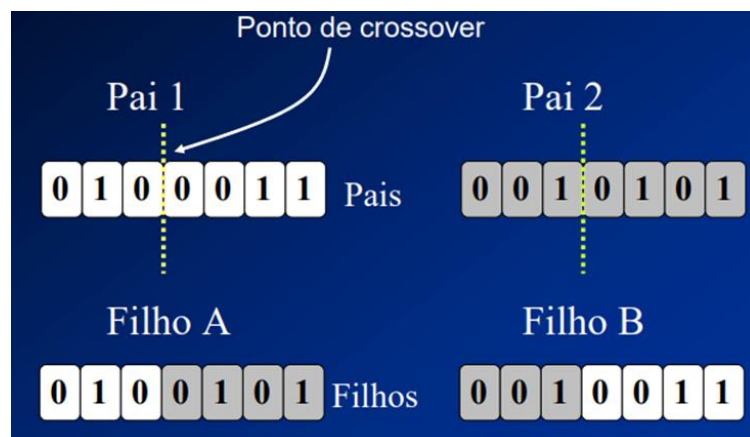


Figura 2 - Crossover single-point

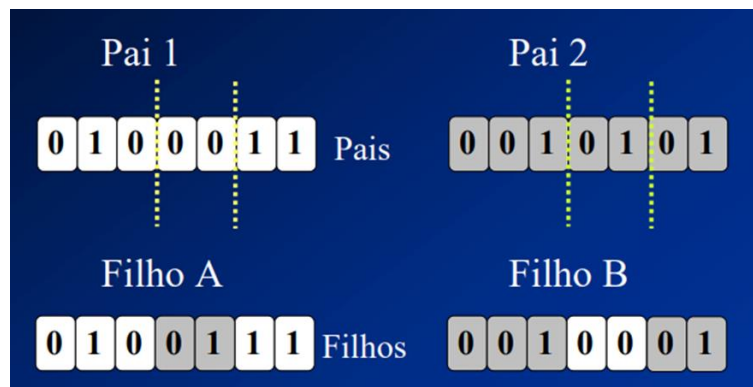


Figura 3 - Crossover two-point

4. What are some potentialities and disadvantages of genetic algorithms?

Vantagens:

- São robustos e aplicáveis a uma grande variedade de problemas;
- Apresentam um bom desempenho para uma grande escala de problemas;
- São de fácil implementação e proporcionam maior flexibilidade no tratamento do problema a ser resolvido;
- Não usam apenas informação local, logo, não ficam presos, necessariamente, a ótimos locais como determinados métodos de busca;
- São técnicas extremamente adequadas para funções multimodais e de comportamento complexo;

Desvantagens:

- Dificuldade de achar o ótimo global exato;
- Requerem muitas avaliações de função de aptidão;
- Grandes possibilidades de configurações que podem complicar a resolução do problema tratado;

5. Describe the crossover (recombination) operation in a classical differential evolution approach.

É a criação de novos indivíduos por meio da combinação de dois ou mais indivíduos. Realiza-se a troca de informações entre as soluções candidatas pais, sendo feita pela seleção aleatória de um local ou mais, em um cromossomo, para ser feita divisão e combinação dos pedaços separados de cada cromossomo com os pedaços de outro cromossomo criando soluções candidatas filhos.

6. Describe the mutation operation in a classical differential evolution approach.

A mutação modifica aleatoriamente um ou mais genes de um cromossomo, regulado por uma taxa de mutação que regula a probabilidade de ocorrer mutação em cada gene. Essa taxa de mutação tende a ser pequena, para apenas introduzir uma variabilidade extra para os cromossomos sem que desviem do valor desejado devido a essas mutações. Isso permite contornar algumas situações em que a função converge rápido demais, chegando a uma solução que não é a buscada.

## DE - Tabelas

Member of population	Genotype (binary representation) Base 2			Phenotype (floating point representation) Base 1 (maximization) f			
	g1	g2	g3	x1	x2	x3	f = x1+x2+x3
1	001.110	001.100	010.000	1.750	1.500	2.000	5.250
2	000.001	000.010	001.111	0.125	0.250	1.875	2.250
3	001.000	010.010	000.110	1.000	2.250	0.750	4.000
4	110.010	010.110	001.000	6.250	2.750	1.000	10.000
best member	110.010	010.110	010.000	6.250	2.750	2.000	10.000

Selected members to match (parents)	Genotype (binary representation) Base 2			Phenotype (floating point representation) Base 1		
	g1'	g2'	g3'	x1'	x2'	x3'
1': 1 and 2	000.001	000.010	011.111	0.125	0.250	3.875
2': 1 and 2	001.110	001.100	000.000	1.750	1.500	0.000
3': 2 and 3	001.000	000.010	000.110	1.000	0.250	0.750
4': 2 and 3	000.001	010.010	001.111	0.125	2.250	1.875

Members to apply the mutation (offspring)	Genotype (binary representation) Base 2			Phenotype (floating point representation) Base 1		
	g1''	g2''	g3''	x1''	x2''	x3''
1': yes	001.001	000.101	010.011	1.125	0.625	2.375
2': no	001.110	001.100	000.000	1.750	1.500	0.000
3': yes	011.000	101.010	010.010	3.000	5.250	2.250
4': no	000.001	010.010	001.111	0.125	2.250	1.875

Member of population	Genotype (binary representation) Base 2			Phenotype (floating point representation) Base 1 (maximization) f			
	g1''	g2''	g3''	x1''	x2''	x3''	f = x1+x2+x3
1''	001.001	000.101	010.011	1.125	0.625	2.375	4.125
2''	001.110	001.100	000.000	1.750	1.500	0.000	3.250
3''	011.000	101.010	010.010	3.000	5.250	2.250	10.500
4''	000.001	010.010	001.111	0.125	2.250	1.875	4.250
best member	110.010	010.110	010.00	3.000	5.250	2.375	10.500