

Perfil de Sistemas Inteligentes

Aprendizagem e Extração de Conhecimento

1º/4º Ano, 1º Semestre

Ano letivo 2017/2018

Trabalho Prático – 1º Parte

Outubro, 2017

**Grupo de Trabalho Nº6**

Ana Esmeralda Fernandes

A74321



Miguel Dias Miranda

A74726

Mestrado Integrado em Engenharia Informática

outubro de 17

# Conteúdo

[Conteúdo 2](#_Toc495853399)

[Índice figuras 3](#_Toc495853400)

[Algoritmos Genéricos 4](#_Toc495853401)

[Introdução 4](#_Toc495853402)

[Fases do algoritmo 4](#_Toc495853403)

[Codificação 4](#_Toc495853404)

[População inicial 5](#_Toc495853405)

[Seleção 5](#_Toc495853406)

[Reprodução 6](#_Toc495853407)

[Mutação 6](#_Toc495853408)

[Critério de paragem ? 6](#_Toc495853409)

[Vantagens 6](#_Toc495853410)

[Desvantagens 7](#_Toc495853411)

[Capacidade de aprendizagem 7](#_Toc495853412)

[Aplicações do algoritmo 7](#_Toc495853413)

[Ferramentas de desenvolvimento 7](#_Toc495853414)

[Soluções existentes no mercado 7](#_Toc495853415)

[Referencias 8](#_Toc495853416)

Índice figuras

[Figura 1 – Exemplos da técnica de reprodução por crossover 6](#_Toc495853503)

# Algoritmos Genéricos

## Introdução

Os algoritmos genéticos são métodos de procura de soluções, geralmente aplicados em problemas de otimização ou aprendizagem, que se baseiam nos conceitos da teoria da evolução, apresentada por Darwin.

De uma forma geral, a partir de uma população inicial, que representa um conjunto de possíveis soluções para o problema, os elementos com maior probabilidade de maximizar a solução do problema são selecionados e, a partir deles, são geradas novas gerações de soluções, cada vez melhores e mais adaptadas ao contexto do problema.

Como a cada iteração se gera uma nova geração de soluções, quando o algoritmo convergir será possível obter não só uma possível solução para o problema, mas sim um conjunto de soluções, representadas por cada elemento da geração final após a convergência do algoritmo. Esta é assim uma das principais vantagens do uso de algoritmos genéticos.

## Fases do algoritmo

### Codificação

Considerando que um algoritmo genético é independente da área e contexto dos problemas onde é aplicado, a forma das soluções por si encontradas é uma representação dos parâmetros que caracterizam uma solução real para um problema. Esta codificação das variáveis de uma solução num elemento capaz de ser interpretado e processado pelo algoritmo, dá origem a um cromossoma.

Cada cromossoma contem assim informação relativa á solução candidata que representa. Esta codificação num cromossoma é geralmente uma cadeia binária, de 0s ou 1s, podendo em algumas adaptações mais recentes do algoritmo ser também números inteiros, sequencias de *strings* ou até mesmo árvores de decisões. Tal como na biologia, o conteúdo de um cromossoma é designado por genótipo e interpretação semântica do seu conteúdo designa-se por fenótipo. Esta interpretação semântica será uma combinação do conteúdo de um cromossoma com o contexto do problema.

No caso de uma representação binária, um bit ou subconjuntos de bits podem representam alternativas sobre um determinado parâmetro da solução.

Considerando como exemplo um problema de otimização no embalamento de produtos, podemos ter quatro variáveis na decisão:

|  |  |  |  |
| --- | --- | --- | --- |
| Material a embalar | Espessura (cm) | Material lacrar caixa | Pegas laterais |
| 00 – Ferro  01 – Alumínio  10 – Madeira  11 – Papel | 001 – 1cm  010 – 2cm  …  110 – 6cm  111 – 7cm | 00 – Cola  01 – Fita cola  10 – Plástico | 1 – Sim  0 – Não |

Com o exemplo apresentado, qualquer cromossoma que represente uma solução candidata será uma sequencia binária de 8 bits.

Por exemplo, o embalamento de uma placa de alumínio, com 5cm de espessura, cuja caixa é fechada com fita cola e apresenta pegas laterais dará origem ao cromossoma cuja sequencia binária é **01011011**.

### População inicial

O primeiro passo na implementação do algoritmo é assim a criação de uma população inicial. Os elementos que constituam esta população inicial devem ser gerados de forma aleatória, criando assim alguma diversidade de soluções. Tal como no suporte da teoria da evolução biológica, este fator de diversidade entre as soluções é essencial para descrever diferentes hipóteses de solução e criar futuras gerações que gradualmente se vão adaptando e especializando como solução do problema. Se a dimensão da população inicial for reduzida, é cortada a hipótese de criar sucessivas gerações e o algoritmo converge prematuramente, gerando soluções limitadas e pouco ótimas. Em oposição, populações iniciais vastas e com muitos casos aumenta consideravelmente o tempo de execução do algoritmo.

Tendo um conjunto de soluções iniciais, o algoritmo pode assim aplicar os métodos de evolução: seleção e reprodução do elementos mais favoráveis e aplicação de mutações para manter a diversidade entre gerações.

### Seleção

A cada iteração, cada um dos elementos da população são analisados e avaliados a nível de qualidade enquanto possível solução para o problema tratar. Esta medida é obtida através de uma função de avaliação, que deve ter em conta o contexto do problema e privilegia os indivíduos melhor adaptados.

Esta função objetivo, também designada por função de fitness, pode envolver o uso de simuladores com a solução em análise e é geralmente complexa de calcular.

Os indivíduos com maiores medidas, têm assim maior probabilidade de serem escolhidos e se “reproduzirem”, originando uma discência que combina as suas características. Geralmente os cromossomas “pais” são substituídos pelos filhos que geram.

### Reprodução

Organizando os cromossomas selecionados por pares para reprodução, é aplicada a técnica de crossover através da definição de pelo menos um ponto de corte. A posição de corte representa assim a zona onde os dois cromossomas irão trocar as suas características, gerando assim uma descendência que os combine.

No caso de cromossomas com representação binária, a posição de corte pode tirar significado ao cromossoma, sendo assim um processo de verificação e correção dos cromossomas gerados.

A figura seguinte procura esquematizas a criação de duas gerações através da técnica de reprodução por crossover. No primeiro esquema a descendência é criada por um plano de corte e, no segundo exemplo, por dois planos de corte.

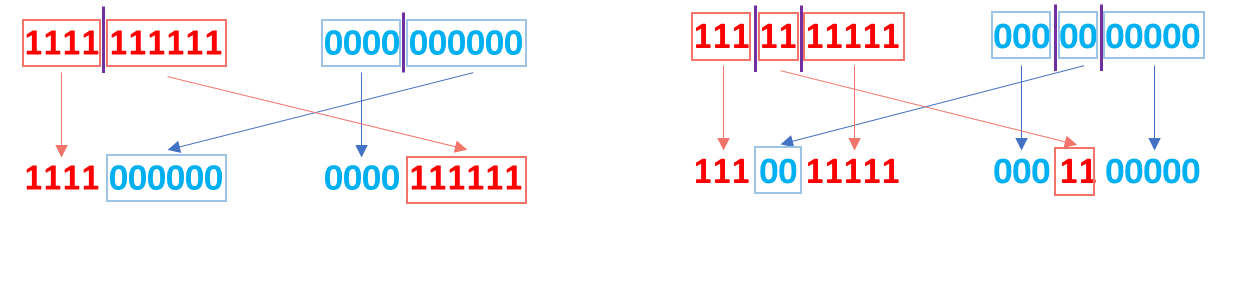


Figura 1 – Exemplos da técnica de reprodução por crossover

### Mutação

Além diversidade e seleção dos melhores cromossomas na fase de reprodução, existe ainda uma fase de mutação de alguns elementos da população gerada. Tal como no contexto biológico, as mutações ocorrem de forma extraordinária e rara e procuram dotar os indivíduos de novas características, possivelmente vantajosas para a sua adaptação ao meio.

No contexto dos algoritmos genéticos, a aplicação de mutações a cada cromossoma tem uma probabilidade baixa de acontecer, até porque poderia levar á perda de boas características herdadas previamente pelos cromossomas progenitores.

No caso de uma codificação binária, uma mutação corresponde á alteração aleatória do valor de um dos seus bits.

Por exemplo uma sequencia **11010110** pode mutar-se para **11010010**. Cada bit de um determinado cromossoma tem a mesma probabilidade de ser mutado.

### Critério de paragem ?

## Vantagens

Algoritmos intrinsecamente paralelos, pelo facto de trabalharem simultaneamente sobre várias soluções para o problema. Diferem de outros algoritmos tradicionais que operam as soluções de forma sequencial;

O Algoritmo em si não precisa de conhecimentos específicos sobre o problema que se encontra a resolver. A sua forma de iteração é igual para qualquer contexto, e é nas fases de codificação das soluções e definição da função objetivo que se tem em conta o contexto do problema a resolver.

Utiliza operadores probabilísticos em vez dos típicos operados determinísticos

## Desvantagens

Definir uma representação do problema. A codificação de uma solução num cromossoma deve ser robusta, e capaz de tolerar as alterações aleatórias induzidas pelos planos de corte, na fase de crossover e pelas possíveis alterações injetadas por mutações nos elementos da nova geração.

Dependendo do tamanho da população inicial e do número de gerações, o algoritmo pode demorar demasiado tempo a convergir, ou até nem o fazer.

Pode convergir prematuramente se logo de inicio surgir por acaso um individuo com boas características. A existência desta aparente solução ótima logo de inicio pode ocultar outras soluções viáveis para o problema, devido ao critério de seleção.

## Capacidade de aprendizagem

A capacidade de aprendizagem pode ser analisada na fase de reprodução do algoritmo. Considerando que um par de cromossomas selecionados representam duas possíveis soluções para o problema, os duas soluções geradas para a geração futuro vão herdar uma fusão das características dos seus progenitores, adquirindo assim o seu conhecimento.

## Aplicações do algoritmo

### Ferramentas de desenvolvimento

### Soluções existentes no mercado

# Referencias

Arranz de la Peña, Jorge and Parra Truyol, Antonio. (n.d.). Algoritmos Genéticos.

Departamento de Engenharia Telemática, Universidade Carlos III de Madrid.

[online] Available at: <http://www.it.uc3m.es/jvillena/irc/practicas/06-07/05.pdf>

[Accessed 16 Oct. 2017].

Goldberg, D. (2012). *Genetic algorithms in search, optimization, and machine learning*. Boston [u.a.]: Addison-Wesley.

Introduction to Genetic Algorithms. (n.d). [PDF] Available at: <http://www.egr.msu.edu/~goodman/GECSummitIntroToGA_Tutorial-goodman.pdf>

[Accessed 16 Oct. 2017].