

3 Inteligência Artificial

3.1. Introdução

Durante milhares de anos, temos tentado de entender *como pensamos*; é dizer, como entender como um simples punhado de matéria pode: perceber, entender, prever e manipular um mundo muito maior e complicado que ela mesma. O campo da Inteligência Artificial (IA) vai mais lá: não só tenta compreender, mas que também se esforça em construir entidades inteligentes. Nós temos na Tabela abaixo definições de I. A. se propõem algumas definições, não aceitas universalmente, mas simples de compreender. Na parte superior se referem a processos mentais/razoamento, enquanto que as da parte inferior aludem à conduta. Da esquerda medem o êxito em termos de fidelidade na forma de atuar dos humanos, enquanto que as da direita pegam um conceito ideal de inteligência chamado racionalidade. É racional se faz “o correto” em função de seu conhecimento.

Segundo Russell et al (2004) e Guerrero et al (2002), as técnicas de inteligência artificial têm três características principais que as diferenciam dos outros métodos.

1. A *busca*, para explorar as distintas possibilidades em aqueles problemas onde os passos a seguir não são claramente definidos.
2. O *emprego do conhecimento*, que permitem explorar a estrutura e relações do mundo ou domínio à que pertence o problema, e a redução do número de possibilidades por considerar, tal como fazemos os humanos.
3. A *Abstração*, que proporciona a maneira de generalizar para os passos intrinsecamente similares.

3.2. Escolas de pensamento

A Inteligência Artificial se divide em duas escolas de pensamento:

3.2.1. Inteligência Artificial Convencional

A inteligência artificial convencional tem que ver com métodos que atualmente se conhece como maquina de aprendizagem, se caracterizam por o formalismo e o analises estadístico. Alguns métodos de esta rama incluem:

- *Sistemas expertos*: aplicam capacidade de raciocínio para lograr uma conclusão. Um sistema experto pode processar um grão quantidade de informação conhecida e prover conclusões baseadas nesta.
- *Raciocínio baseada em casos*.
- *Rede bayesiana*, um modelo de representação do conhecimento baseado em teoria da probabilidade.
- *Inteligência Artificial baseada em comportamentos*: método modular para construir sistemas de Inteligência Artificial manualmente.

Inteligência Artificial Computacional

A inteligência Artificial Computacional implica desenvolvimento e aprendizagem iterativa (modificações iterativas dos parâmetros em sistemas conexionistas), baseando-se em dados empíricos. Alguns métodos de esta rama incluem:

- *Redes Neurais*, sistemas com grandes capacidades de reconhecimento de padrões.
- *Sistemas difusos*, técnicas para lograr o raciocínio baixo incerteza, Têm sido amplamente usada na indústria moderna e em produtos de consumo massivo, como as lavadoras.
- *Computação evolutiva* aplica conceptos inspirados na biologia, tais como população, mutação e sobrevivência do mais apto para gerar soluções sucessivamente melhores para o problema. Estes métodos a sua vez se dividem em algoritmos evolutivos (por exemplo, algoritmos genéticos) e inteligência coletiva (por exemplo, o algoritmo de formiga).

Nos últimos anos tem aparecido uma serie de disciplinas englobadas baixo o termo de I. A. que tentam emular o comportamento humano. Tecnologias como: as redes neurais e os algoritmos genéticos os quais nós vamos tratarmos mais adiante em neste capitulo.

3.3. Redes Neurais

As redes neurais (RNA em diante) são mais que outra forma de emular certas características próprias dos humanos, como são a capacidade de memorizar e de associar fatos. Sim se examinam com atenção aqueles problemas que não podem expressar se através de um algoritmo, se observará que todos eles têm uma característica em comum: **a experiência**. O homem é capaz de resolver estas situações acudindo à experiência acumulada. Assim, achamos claro que uma forma de aproximar se ao problema consiste na construção de sistemas que sejam capazes de reproduzir esta característica humana.

Existem varias definições de redes neurais, aqui se apresentam algumas delas:

- As RNAs são modelos matemáticos inspirados em sistemas biológicos, adaptados e simulados em computadores convencionais.
- Uma RNA é um processador distribuído maciçamente paralelamente distribuído, constituído de unidades de processamento simples, que têm a propensão natural para armazenar conhecimento experimental e torná-lo disponível para o uso (Haykin, 2001).
- As RNAs são redes interconectadas massivamente em paralelo, de elementos simples e com organização hierárquica, as quais intentam interatuar com os objetos do mundo real do mesmo modo que o faz o sistema nervoso biológico.
- Uma RNA é um sistema que tem capacidade computacional adquirida por meio de aprendizado e generalização (Pelli, 2006).
- Uma RNA é um modelo computacional com um conjunto de propriedades específicas, com a habilidade de adaptar-se ou apreender, generalizar ou organizar a informação, todo elo baseado num processamento eminentemente paralelo.

3.3.1. Modelo físico

O cérebro humano continuamente recebe sinais de entrada de muitas fontes e as processa a maneira de criar uma apropriada resposta de saída. Este sistema é constituído de células, responsáveis pelo seu funcionamento, denominadas de *neurônios*. O cérebro humano apresenta aproximadamente 10

bilhões de neurônios e cerca de 60 trilhões de conexões entre eles, que se interconectam para elaborar “Redes Neurais”. Os neurônios estão formados por elementos chamados corpo, axônio e dendritos, como se mostra na Figura 10.

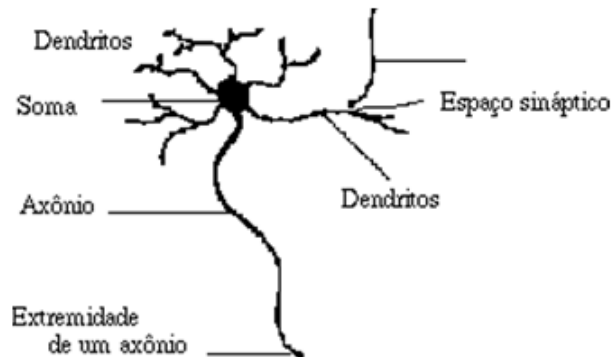


Figura 10. Neurônio biológico.

Fonte: Vieira et al (2004).

Segundo Vellasco (2008), existe multidão de modelos neuronais, cada um de eles especializado em determinadas funcionalidades. Um dos modelos mais utilizados é o *Perceptron Multicamada* cuja propriedade de aproximado universal lhe permite emular o comportamento da maioria de funções matemáticas em geral..

Para que uma RNA seja capaz aproximar uma função, deve passar previamente por um processo de aprendizagem que consiste em estabelecer um problema, obter uma possível solução e em função da perfeição alcançada, corrigir e repetir a ação até que a tarefa se leve a cabo de forma correta. Para que a aprendizagem seja efetiva, se deve proporcionar uma coleção de dados que sejam representativos do problema.

As RNAs caracterizam-se por possuírem elementos de processamento de estrutura bem simples, com conexões entre eles. Cada conexão na rede tem um peso associado e este peso representa a intensidade de interação ou acoplamento entre os elementos de processamento e sua natureza é excitatória ou inibitória (Haykin, 2001; Vellasco, 2008).

Um dos componentes da arquitetura de uma RNA é o número de nós em cada uma das camadas. O nó é também denominado de neurônio e “é à base de uma RNA”.

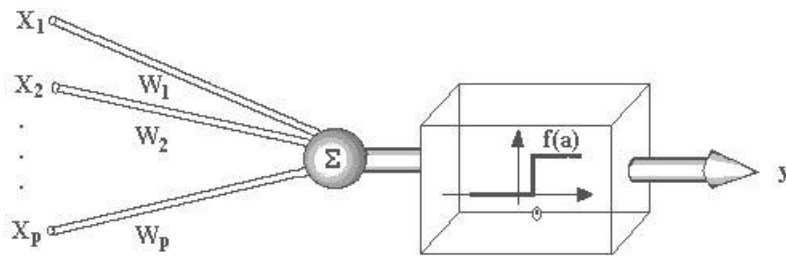


Figura 11. Neurônio Artificial.

Fonte: Luger, (2002).

O Perceptron, segundo Rich e Knight (1993), foi um dos primeiros modelos de uma rede neural, e modela um único neurônio que contém os seguintes elementos:

1. *Um conjunto de sinapses ou elos de conexão*, cada uma delas caracterizada por um peso ou força própria. Especificamente, um sinal x_j na entrada da sinapse j conectada ao neurônio k é multiplicado pelo peso sináptico w_{kj} . É importante notar que a maneira como são escritos os índices do peso sináptico w_{kj} , o primeiro índice se refere ao neurônio em questão e o segundo se refere ao terminal de entrada da sinapse à qual o peso se refere. O peso sináptico de um neurônio artificial pode estar em um intervalo que inclui valores negativos ou bem como positivos.
2. *Um somador*, para somar os sinais de entrada, ponderado pelas respectivas sinapses do neurônio, cujas operações descritas constituem um combinador linear.
3. *Uma função de ativação*, para restringir a amplitude da saída de um neurônio. A função de ativação é também referida como função restritiva já que limita o intervalo permissível de amplitude do sinal de saída a um valor finito. Tipicamente, o intervalo normalizado da amplitude da saída de um neurônio é escrito como o intervalo unitário fechado $[0,1]$ ou $[-1,1]$.

Rímolo e Villalta, (2004) definem como funções de ativação:

- a) **Função escalão.** A função escalão se utiliza quando a saída da rede é binária. A saída do neurônio se ativa somente quando o estado de ativação é maior ou igual ao valor de umbral.
- b) **Função lineal e mista.** A função linear corresponde à expressão

$f(x) = x$. É com função mista, se a soma das entradas é menor que um limite inferior, a ativação se define como zero, ao contrario a ativação é um. Se a soma de entrada esta compreendida entre ambos os limites a ativação se define como uma função linear da soma dos sinais de entrada.

- c) **Função continua (sigmóide).** Com a função sigmóide o valor de estímulo dado na entrada é perto para um dos valores assintóticos, este faz que o valor da saída este compreendido na zona alta ou baixa do sigmóide. Quando a pendente da função é elevada esta tende à função escalão. A importância da função sigmóide é que sua derivada sempre é positiva e perto a zero para valores grandes; ainda pega seu máximo quando x é zero.

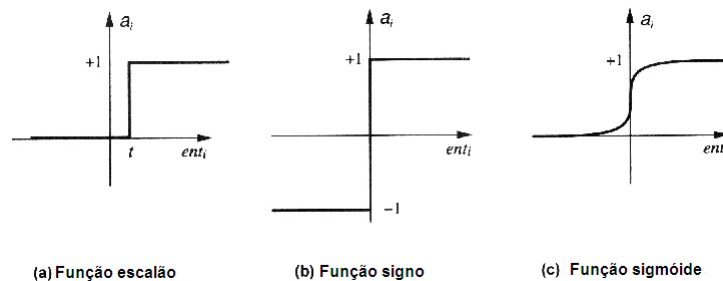


Figura 12. Algumas funções de ativação.

Fonte: Rímolo e Villalta (2004).

O modelo neural inclui também um **bias** aplicado externamente, representado por b_k . O bias b_k têm o efeito de aumentar ou diminuir a entrada líquida da função de ativação, dependendo se ele é positivo/negativo.

3.3.2. Número de Neurônios e as Camadas

Uma RNA é composta por neurônios artificiais e cada um dos neurônios possui um conjunto de elos de conexões, um somador e uma função de ativação. A conexão é uma linha única de comunicação que vai de um neurônio que recebe uma informação a outro que envia esta mesma informação, podendo ser de excitação ou inibição. Além das conexões, os neurônios da rede são agrupados em camadas e de acordo com o número de camadas, uma RNA pode ser:

- a) **De camada única**, quando só existe um nó entre qualquer entrada e

qualquer saída da rede.

- b) **De múltiplas camadas (*Multilayer Perceptron*–MLP)**, quando existe mais de um neurônio entre alguma entrada e alguma saída da rede. Esta camada intermediária entre a de entrada e a de saída, é denominada de camada oculta e pode existir uma ou mais dela.

Numa MLP, o número de camadas intermediárias depende do número de exemplos de treinamento, da complexidade da função a ser aprendida e da distribuição estatística de treinamento. O número de nós nas camadas intermediárias, em geral, é definido empiricamente.

De acordo com o Teorema de Kolmogorov-Nielsen, em uma RNA de três (3) camadas, com dimensão n na camada de entrada, a camada oculta pode ser composta de até $2n + 1$ neurônios, e a camada de saída com m neurônios (Kovács, 2006).

3.3.3. Modo de Conexão

A conectividade de uma RNA pode ser:

- a. *Fracamente ou parcialmente conectada*, quando cada um dos nós de uma camada não está ligado a todos os nós da camada posterior a sua.
- b. *Fortemente ou completamente conectada*, quando cada um dos nós de uma camada (exceto, aos da camada de saída) está ligado a todos os nós da camada posterior a sua.

Segundo Haykin, (2001); Luger, (2002) e Leite (2006), além da definição da conexão para uma RNA, é necessário o estabelecimento da sua topologia que pode ser:

- **FEEDFORWARD OU ACÍCLICA** (a saída do neurônio na i -ésima camada da rede não pode ser usada como entrada de nós em camadas de índice menor ou igual a i) são redes cuja saída final (única) não possui qualquer ligação com as entradas.
- **FEEDBACK OU CÍCLICA** (a saída de algum neurônio na i -ésima camada da rede é usada como entrada de nós em camadas de índice menor ou igual a i), são redes cuja saída final (única) é ligada às entradas que se comportam como autômatos reconhecedores de cadeias, onde a saída, que é realimentada, fornece o estado de autômato.

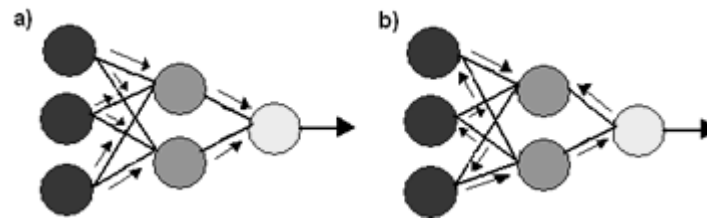


Figura 13. Conexão (a) Feedforward (b) Feedback.

3.3.4. Regra de Aprendizagem das RNA

A regra de aprendizagem de uma RNA é o conjunto preestabelecido de procedimentos ou regras bem-definidas para adaptar os parâmetros de uma RNA de modo que ela possa aprender uma determinada função e solucionar um problema de aprendizagem. Entretanto, não há um único algoritmo, mas sim um conjunto de ferramentas representadas por diversos algoritmos, cada qual com suas vantagens e desvantagens. Os algoritmos basicamente diferem pela maneira pela qual o ajuste dos pesos é feito (Leite, 2006; Vellasco, 2008).

3.3.5. Aprendizado

A principal propriedade de uma RNA é a sua capacidade de aprender a partir do seu ambiente e de aprimorar o seu desempenho através da aprendizagem. O desempenho é melhorado com o tempo de acordo com alguma medida preestabelecida. A RNA aprende através de um processo iterativo no qual ocorre ajuste nos pesos sinápticos e no nível do *bias*, tornando-se mais instruída no seu ambiente após cada iteração do processo de aprendizagem. A aprendizagem é:

1. O processo pelo qual os parâmetros livres de uma RNA são adaptados através do processo de estimulação pelo ambiente no qual a rede está inserida. Este tipo de aprendizagem é determinado pela maneira pela qual a modificação dos parâmetros ocorre (Haykin, 2001).
2. O processo pelo qual os parâmetros de uma RNA são ajustados através de uma forma continuada de estímulo pelo ambiente no qual a rede está operando, sendo o tipo específico de aprendizagem realizado definido pela maneira particular, como correm os ajustes realizados nos parâmetros (Luger, 2002; Leite, 2006).

O conjunto preestabelecido de regras bem-definidas para a solução de

um problema de aprendizagem é denominado de algoritmo de aprendizagem. Não há um único algoritmo de aprendizado, pois a diferença entre um algoritmo e outro está na maneira como é formulado o ajuste de um peso sináptico de um neurônio.

Há diversos métodos ou regras para treinamento de rede que foram desenvolvidos: Aprendizagem por correção; Aprendizagem baseada em memória; Aprendizagem Hebbiana; Aprendizagem competitiva; Aprendizagem de Boltzmann.

Haykin (2001); Vellasco (2008), afirmam que esses métodos listados anteriormente podem ser agrupados em dois paradigmas principais de aprendizagem:

1. *Aprendizado supervisionado ou aprendizagem com um professor.* O aprendizado supervisionado tem em sua estrutura uma espécie de instrutor que confere quanto à rede está próxima de uma solução aceitável.
2. *Aprendizado não-supervisionado ou aprendizagem sem um professor.* O aprendizado não-supervisionado não possui conhecimento a priori das saídas da rede e funciona de modo a distinguir classes de padrões diferentes dos dados apresentados à rede, através de algoritmos de aprendizado baseados geralmente em conceitos de vizinhança e agrupamento.

3.3.6.

Validação da rede neuronal

Segundo Vellasco (2008), depois do processo de treinamento, os pesos das conexões na rede neuronal ficam fixos. Com o propósito de validar a rede neuronal se requer de outro conjunto de dados, denominado conjunto de validação ou teste. Agora o problema é que tem que decidir quando a saída da rede neuronal será considerada como correta. Pelo tanto se tem que falar de generalização da rede neuronal. Generalização se entende como a capacidade da rede de proporcionar uma resposta correta frente a padrões que não foram usados em seu treinamento. Uma RNA corretamente treinada deve possuir uma boa capacidade de generalização, o que significa que tem apreendido corretamente e que respondera adequadamente em frente de padrões não vistos com anterioridade.

Neste trabalho focaremos apenas o tipo de aprendizado supervisionado,

apresentando o algoritmo de treinamento *backpropagation*, o qual é a alternativa mais utilizada para treinamento de redes neurais diretas.

3.3.7. Aprendizado Supervisionado

A denominação de aprendizado supervisionado é porque existe um supervisor (professor) externo que fornece a entrada e a saída desejadas para a rede com o objetivo de ajustar os parâmetros da rede, de forma a encontrar uma ligação entre os pares de entrada e saída fornecidos.

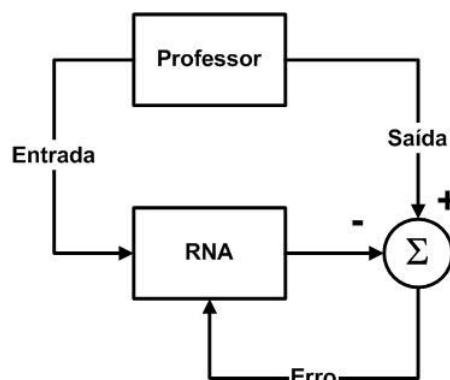


Figura 14. Aprendizado Supervisionado.

Fonte: Leite (2006).

Na aprendizagem o professor indica explicitamente um comportamento bom ou ruim para a rede, com o objetivo de direcionar o processo de treinamento. A rede tem sua saída comparada com a saída desejada, ajustando-se logo os pesos das conexões para minimizar o erro. A minimização da diferença é incremental, já que pequenos ajustes são feitos nos pesos a cada etapa de treinamento, de tal forma que caminhem para uma solução. A soma dos erros quadráticos de todas as saídas é normalmente utilizada como medida de desempenho da rede e também como função de custo a ser minimizada pelo algoritmo de treinamento. Os exemplos mais conhecidos de aprendizado supervisionado são a regra delta e sua generalização o algoritmo *backpropagation* (Haykin, 2001).

3.3.8. Algoritmo BACKPROPAGATION

O algoritmo *backpropagation* tem duas fases: a *Forward* e a *Backward*.

1. Na *Forward*, o algoritmo executa os seguintes procedimentos:
 - a) A entrada é apresentada à primeira camada da rede, a camada inicial C_o .
 - b) Para cada camada C_i a partir da camada de entrada:

Após os nodos da camada C_i $i > 0$ calcular seus sinais de saída, estes servem como entrada para a definição das saídas produzidas pelos nodos da camada C_{i+1} . As saídas produzidas pelos nós da última camada são comparadas com as saídas desejadas.
2. Na *Backward*, o algoritmo executa os seguintes procedimentos:
 - a. A partir da última camada, até chegar à camada de entrada:
 - Os nós da camada atual ajustam seus pesos de forma a reduzir os erros.
 - O erro de um nó das camadas intermediárias é calculado utilizando os erros dos nós da camada seguinte conectados a eles, ponderada pelos pesos das conexões entre elas.

3.4. Algoritmos Genéticos

A evolução produza-se, em quase todos os organismos, como consequência, dos processos primários: a seleção natural e a reprodução sexual. A primeira determina que membros da população sobrevivam até reproduzir-se. A reprodução sexual avalia a mistura e recombinação de genes na descendência do organismo. Os Algoritmos Genéticos se encontram dentro desta classe de algoritmos certas analogias com os processos biológicos da natureza. Cujo propósito consiste em guiar uma busca estocástica fazendo evolucionar ao conjunto de estruturas e selecionando de jeito iterativo as mais adequadas. Sim bem não se garanta que encontre a solução ótima, existe evidencia empírica de que se encontram soluções de um nível aceitável.

Poderíamos definir os Algoritmos Genéticos (AG), de forma geral, como: Métodos estocásticos de busca cega de soluções quase-ótimas. Em eles se mantém uma população que apresenta um conjunto de possíveis soluções, a qual é submetida a certas transformações com as que se trata de obter novos candidatos, e um processo de seleção tendencioso em favor dos melhores candidatos (PACHECO et al,1999; PACHECO,(2008).

Dizemos que a busca é cega porque não se dispõe de nenhum conhecimento específico do problema, de maneira que a busca se baseia exclusivamente nos valores da função objetivo. É também uma busca codificada, já que não se trabalha diretamente sobre o domínio do problema, se não com representações de seus elementos; é múltipla porque busca simultaneamente entre um conjunto de candidatos; e é estocástica, referida tanto as fases de seleção como às de transformação, com o que se obtém controle sobre o fator de penetração da busca. Todo isto faz que os A. Gs proporcionem uma maior robustez, isto é, mais eficácia sem perder generalidade.

Os A. Gs ocupam um lugar central dentro da computação evolutiva. As razões são tanto teóricas como práticas. As razões mais importantes são:

- *Constituem o paradigma mais completo da computação evolutiva*, isto é, resumem de jeito natural, todas as idéias fundamentais de dito enfoque.
- *São muito flexíveis*, é dizer, podem adotar com facilidade novas idéias, gerais ou específicas, que surjam dentro do campo da computação evolutiva. Ademais se podem hibridar facilmente com outros paradigmas e enfoques, ainda que não tenham nenhuma relação com a computação evolutiva.
- *São o paradigma com maior base teórica* de entre os da computação evolutiva. Além disso, dita base teórica é simples em seu desenvolvimento e com grandes possibilidades de ampliação.
- *São os que menos conhecimento específico precisam pra seu funcionamento*, e em conseqüência, os mais versáteis. Mas podem além disso incorporar conhecimento específico com pouco esforço adicional.

3.4.1. Condições para usar um Algoritmo Genético

A aplicação mais comum dos A. Gs tinha sido a solução de problemas de otimização, em onde tem mostrado ser muito eficientes e confiáveis. Contudo, não todos os problemas puderam ser apropriados para a técnica, e se recomenda tomar em conta as seguintes características antes de intentar usa-a:

1. Seu espaço de busca deve estar delimitado dentro de certo rango.
2. Deve poder definir-se uma função de atitude que nos indique que tão bom ou mal é certa resposta.
3. As soluções devem codificar-se de uma forma que resulte relativamente mole de implementar no computador.

O primeiro ponto é muito importante é tentar resolver problemas que tenham espaços de busca discretos, porém estes sejam muito grandes.

3.4.2.

Estrutura de um Algoritmo Genético

O algoritmo genético consiste de um programa de computador que:

- A menor unidade de um AG é chamada gene. Um gene representa uma unidade de informação do domínio do problema, um valor de um atributo.
- Uma série de genes, ou um cromossoma, representa uma possível solução completa para o problema.
- Aceitará dados de entrada correspondente aos cromossomas originais ou os gerará ao azar e aos cromossomas alvos ou dados do entorno.
- Terá uma função de convergência que meda o grão de assimilação do cromossoma ao entorno.
- Elaborará “gerações” de cromossomas compostas por vários cromossomas atuando ao mesmo momento. A geração tem um tempo estabelecido de vida. Os espécimes das gerações competirão entre si utilizando os operadores genéticos. Só os mais aptos se reproduzem.
- Entrará em convergência. Isto significa que não se percebe incremento nos exemplares das gerações seguintes. Cada novo exemplar tem os mesmos genes. Em este ponto o processo evolutivo termina.

3.4.3.

Algoritmo Genético Simples

Tabela 2. Estrutura de um Algoritmo Genético Simples.

Algoritmo
INICIO
T = 0;
INICIAR P(T);
FAZER MENTRAS T < TMAX
T = T + 1;
SELECIONAR P(T) DESDE P(T-1)
CRUZAMENTO P(T)
MUTAÇÃO P(T)
FIM DE FAZER
FIM

Fonte: Pacheco (2008).

A estrutura básica de um A. G. simples inicia com uma geração aleatória de uma população inicial com as potenciais soluções. Em uma iteração qualquer t , o vetor de cromossomas $P(t) = x_1t, \dots, x_nt$ é avaliado de acordo a uma função de ajuste ou *fitness* pra cada x_it . Logo se forma uma nova população, P_{t+1} com os indivíduos que obtiveram melhor ajuste. A nova população é submetida a alterações mediante operadores (cruzamento e mutação) para formar novas soluções que volverão a competir entre elas mediante a aplicação da função de ajuste, até que, depois da aplicação de um critério de detenção de algoritmo finalize.

3.4.4. Funcionamento do Algoritmo Genético

Pacheco (2008) define: Um A. G. inicia pela definição dos genes, onde uma série de genes, ou um cromossoma, representa uma possível solução completa para o problema, logo se continua com a criação da população inicial (em geral, aleatória), tem inicio um processo iterativo de refinamento ou evolução das soluções iniciais, ela emprega uma população de cromossomas, executando, assim, uma busca de forma paralela. O A. G. cria novas soluções através da combinação e refinamento das informações dos cromossomas usando três operações: seleção, *crossover* e mutação. Essas operações produzem novas soluções que formam uma nova população. Cada nova população é chamada de geração.

Durante a seleção, o A. G. escolhe os cromossomas, privilegiando aqueles com maiores aptidões para permanecer e se multiplicar na população. Uma forma comum de seleção (método da roleta) é onde cada cromossoma tem a probabilidade de permanecer na próxima geração proporcional à sua aptidão. Cromossomas com maiores aptidões possuem mais espaço na roleta e conseqüentemente possuem maiores chances de serem escolhidos para o cruzamento e a mutação. Durante o *crossover*, dois cromossomas basicamente trocam algumas de suas informações gene a gene, ou seja, permite a combinação dos elementos de uma solução com os de outra; recombinando as informações, porém não é capaz de gerar diversidade numa população, essa diversidade é obtida através do operador de mutação, alternando o valor de um gene por outro valor qualquer no domínio da aplicação, permitindo explorar novas áreas a procura de melhores soluções.

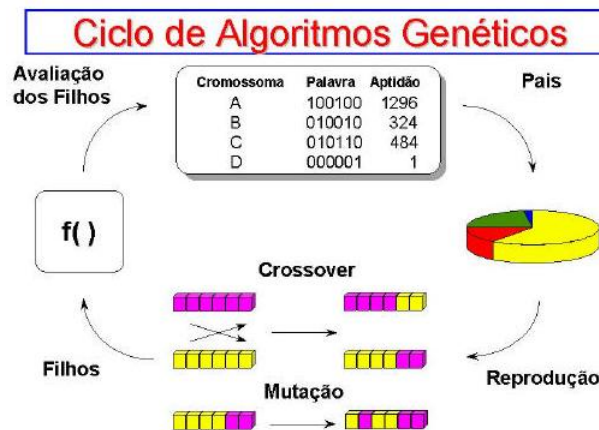


Figura 15. Ciclo básico do método de Algoritmo Genético.

Fonte: Pacheco (2008).

3.4.5. Operadores Genéticos

Os algoritmos genéticos trabalham em base aos operadores genéticos. Um operador genético consiste em uma serie de passos estruturados que se aplicam aos cromossomos e formam resultados concretos. Os mais usuais são:

a. Função Fitness

Este operador mede e opera com a adaptação ao entorno ou competitividade. Mediante o uso de funções como a função de convergência o sistema reconhece o grão de cercania ou adaptabilidade que o cromossoma possui com o que se espera de ele. Os mais parecidos são selecionados pra sobreviver e os demais pra morrer.

b. Seleção

A seleção de pares constitui o método em que se selecionará baixo certo critério dos cromossomos pra sua futura reprodução e substituir a população atual. Independentemente de que se todos os pares se reproduzem. Existem vários métodos que podem utiliza-se pra selecionar os pares que deverão posteriormente cruza-se com uma probabilidade determinada (Pacheco, 2008).

c. Cruzamento ou Crossover

Este operador trata de sacar vantagens da reprodução sexual das espécies superiores. O raciocínio é o seguinte: Sim um macho tem alguns genes que o tornam mais competitivo, e se emparelha com uma fêmea que tem outros genes igualmente importantes, algumas de suas crias terão ambos os genes. Pelo qual se tornarão mais competitivos que seus respectivos pais.

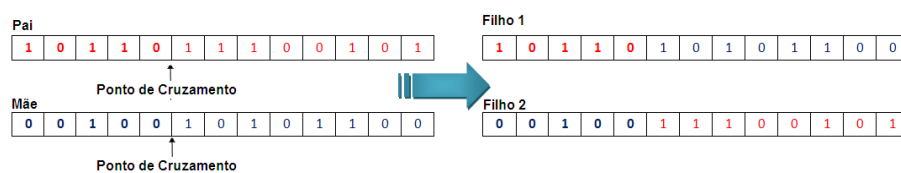


Figura 16. Cruzamento de um só ponto

d. Mutação

Os genes podem sofrer mutação por influência de fatores naturais ou artificiais. O operador de mutação consiste no cambio de valor de uma ou mais bits por outro. Sim bem a porcentagem de mutação é baixa, a acumulação ao longo dos séculos pode produzir efeitos acumulativos importantes. No âmbito dos algoritmos genéticos este efeito se acelera para não esperar este tempo, porém se faz de maneira de que sua incidência não seja um fator preponderante, já que de fato mudaria o conceito de evolução acumulativa pela evolução ao azar.

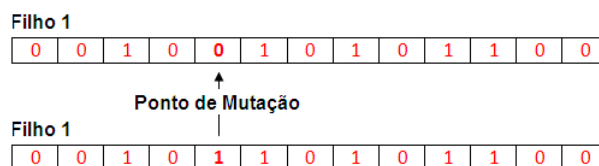


Figura 17. Mutação

e. Clonagem ou elitismo

Consiste na duplicação da estrutura genética de um cromossomo para a geração seguinte. Isto faz que esse exemplar sobreviva intato para competir na nova geração. A clonagem se pode produzir numa de estas duas formas:

- *Explícita*: Quando algum software baseado num algoritmo genético prevê que pra certas circunstancias um cromossoma se duplique pra sim mesmo.
- *Implícita*: Quando o programa não proíbe que no processo de cruzamento um cromossoma seja selecionado como pares de se mesmo pra gerar descendência.