TOPICOS EM COMPUTAÇÃO EVOLUTIVA

Recombinação e Algoritmos Evolutivos como Busca Paralela Adaptativa

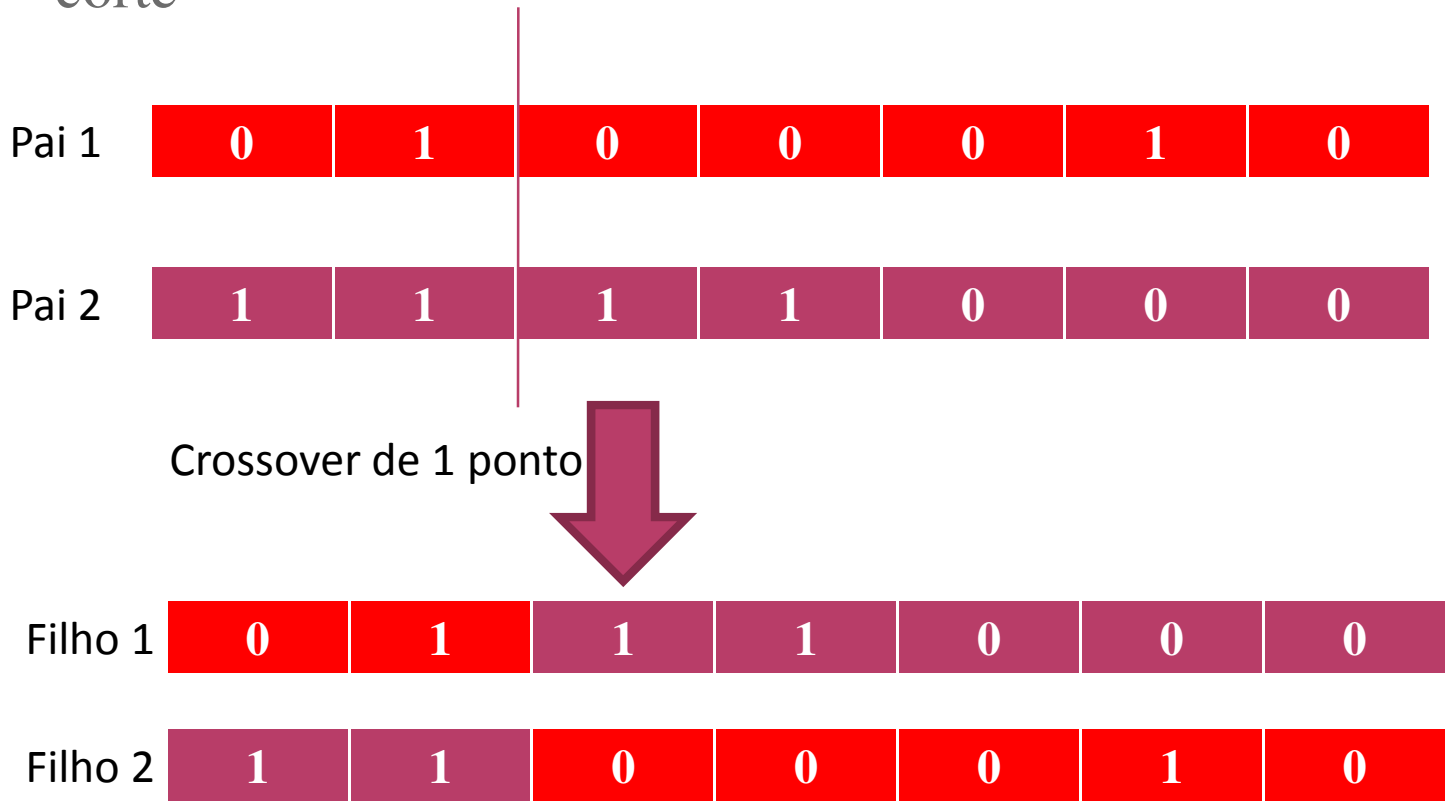
OPERADORES DE RECOMBINAÇÃO

- O mecanismo clássico de reprodução de 2 pais é a recombinação na qual subcomponentes do genoma dos pais são cloneados e remontados para criar um genoma descendente.
- Para genótipos de tamanho fixo a recombinação recebe o nome de crossover e possui 3 tipos básicos:
  - 1 ponto
  - 2 pontos
  - Multi-pontos ou uniforme

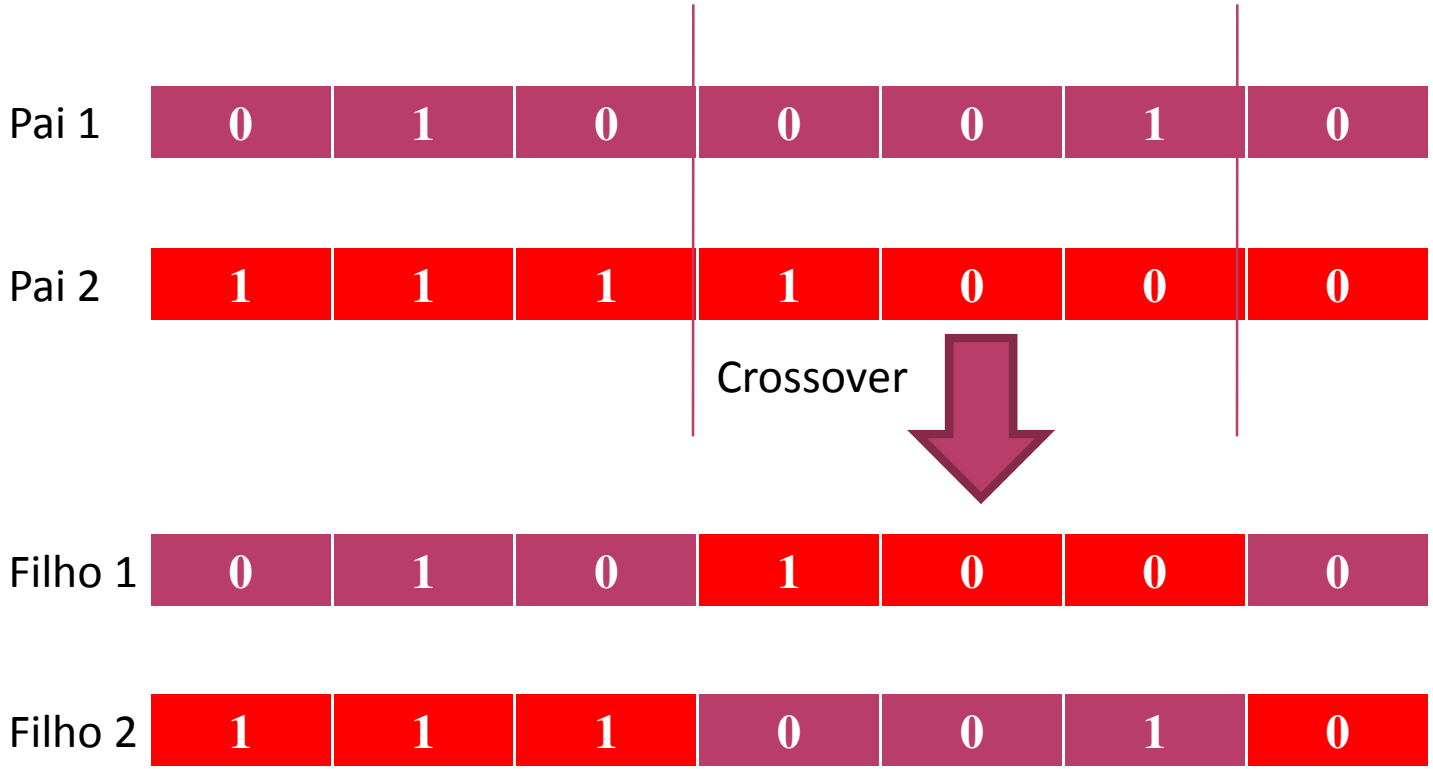
- 1 Ponto
  - consiste em definir um ponto de corte para posteriormente trocar os genes separados por esse ponto.

CROSSOVER DE 1 PONTO

- Suponha que a posição 2 foi o sorteado como ponto de corte

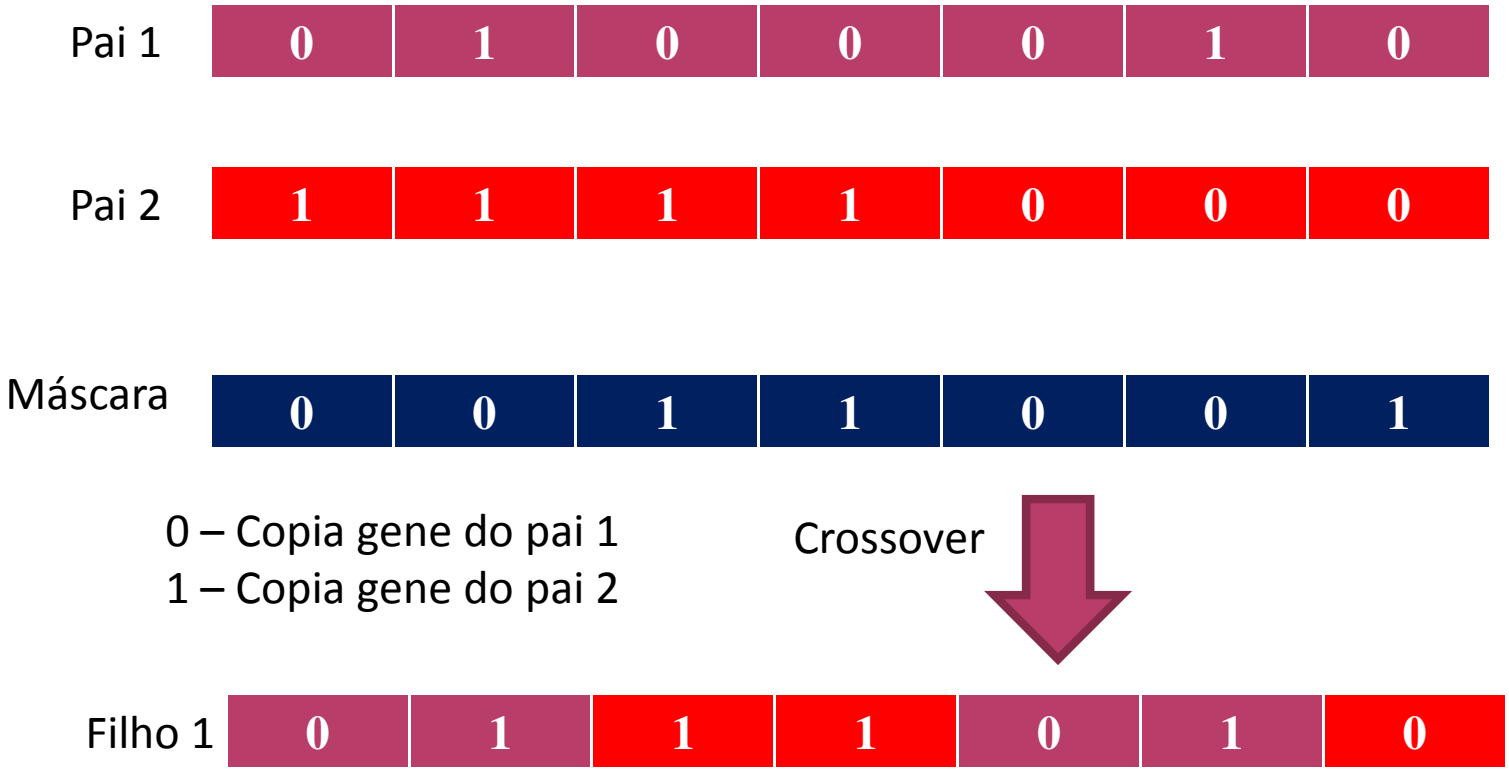


- Crossover dois-Pontos
  - Sortei duas posições de corte (Ex.: 4 e 6)



CROSSOVER MULTI-PONTO OU UNIFORME

- É definida uma máscara, como se jogasse uma moeda, que define de qual pai será copiado o gene.



ANÁLISE DA RECOMBINAÇÃO

- Para esses operadores a quantidade de variação introduzida quando produz descendentes é dependente de dois fatores:
  - Quantos pontos de crossover existem, e
  - Quão similar os pais são entre si
- Ao contrário da mutação, nessas operações de recombinação a quantidade de variação diminui com o tempo quando a população torna-se mais homogênea.
- Essa característica dificulta estimar o nível de variação induzida pelo crossover.
- Assim avalia-se os operadores de acordo com a quantidade de genes em seqüência que são copiados para o filho.

ANÁLISE DA RECOMBINAÇÃO

- Aumentar o número de pontos de crossover aumenta a variação e simultaneamente diminui a probabilidade de que K genes consecutivos sejam copiados para o filho.
- Uma dificuldade dos operadores de 1 e 2 pontos de crossover é que eles introduzem uma tendência de distância de que genes juntos tem mais chance de aparecer no mesmo filho do que gene distantes.
- Por exemplo, no crossover de 1-ponto os genes das extremidades de um pai nunca aparecerão no mesmo filho.
- Com o objetivo de diminuir essa tendência é que foi desenvolvido o crossover de multi-pontos ou uniforme. Ele aumenta a probabilidade de que genes distantes de um pai apareçam no mesmo filho.

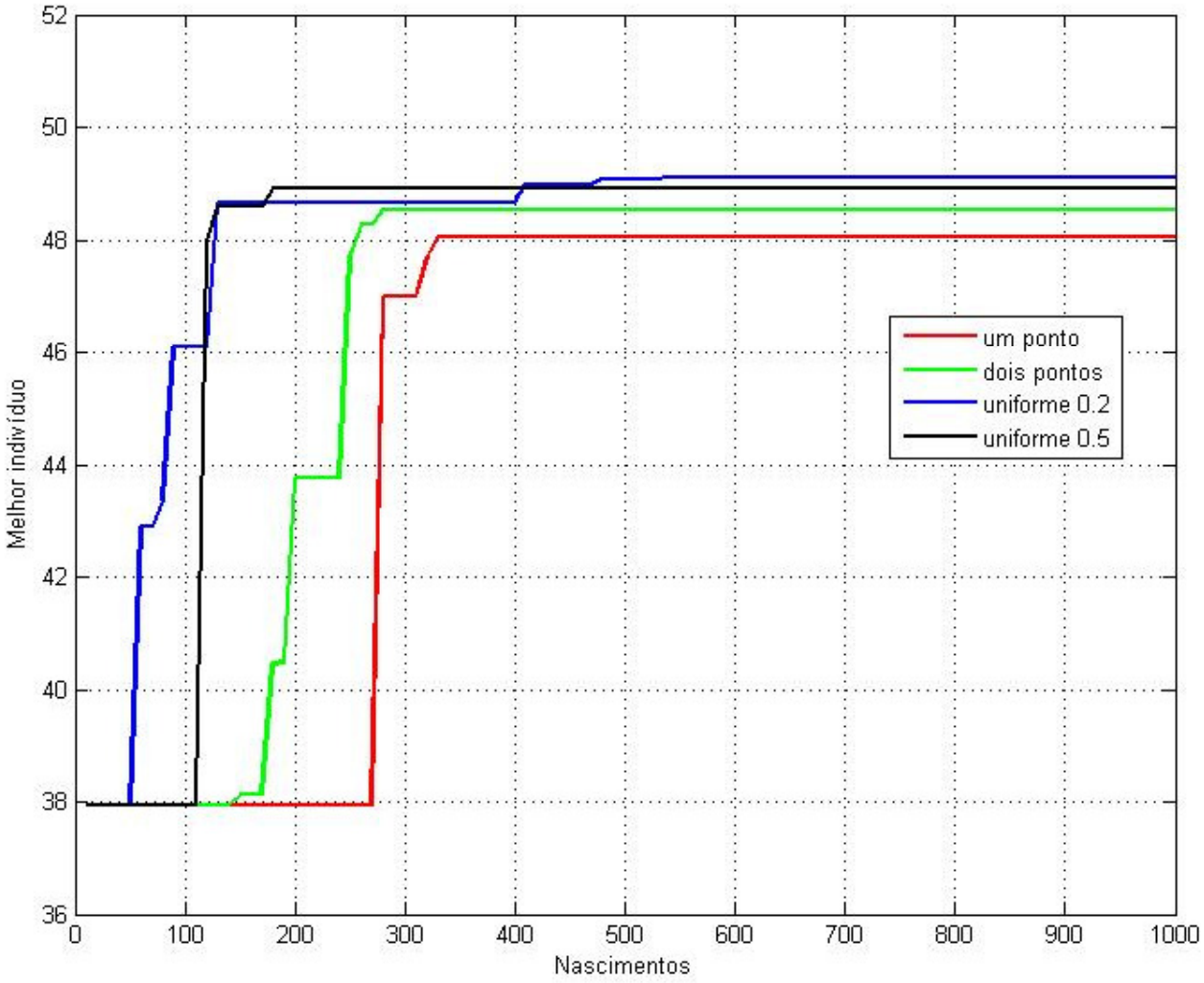
ANÁLISE DA RECOMBINAÇÃO

- Além disso, é possível determinar uma probabilidade para a escolha da máscara variando de 0 a 0,5 para que sejam produzidos mais 0s do que 1s.
- Se a probabilidade for 0 são produzidos 0s e 1s intercalados, se for 0,5 são produzidos seqüências de 0s e 1s.
- Em geral essa probabilidade é utilizada 0,2.
- Outro fator é que o crossover uniforme pode funcionar como os crossovers de 1 ou 2-pontos.

ANÁLISE DA RECOMBINAÇÃO

- O exemplo a seguir apresenta a diferença de comportamento de um AG aplicado com a função de avaliação de Rosenbrock com os diferentes tipos de recombinação.

ANÁLISE DA RECOMBINAÇÃO



ANÁLISE DA RECOMBINAÇÃO

- Com os operadores de 1-ponto e 2-pontos não foi possível alcançar o ótimo pois a população se tornou homogênea e o crossover não conseguiu mais produzir variabilidade.
- O operador uniforme 0,2 produziu a quantidade de variabilidade necessária para que o algoritmo encontra-se a solução ótima.
- Já o operador uniforme com 0,3 produziu variabilidade excessiva o que dificultou o algoritmo a encontrar a solução ótima.

ANÁLISE DA RECOMBINAÇÃO

- Outra questão que existe é se a recombinação deve produzir um ou dois filhos.
- A produção de dois filhos em cada operação de recombinação resulta numa ligeira melhora no desempenho do algoritmo do que produzir somente 1.
- Visto que não é difícil produzir o segundo filho é recomendado a utilização da recombinação gerando sempre dois filhos.

CROSSOVER OU MUTAÇÃO?

- Uma questão que frequentemente é discutida diz respeito a qual dos operadores é melhor, crossover ou mutação.
- Na verdade os dois operadores se complementam e as aplicações tem mostrado que não é possível afirmar que um é melhor do que o outro.
- Cada operador executa um papel diferente na heurística de reprodução.

CROSSOVER OU MUTAÇÃO?

- Vamos utilizar uma representação geométrica para isso.
- Tendo o espaço de busca com os pontos representando os indivíduos.
- A mutação é como uma nuvem ao redor de um ponto que vai sumindo com a distância.
- E o crossover explora os vértices de um retângulo desenhado entre os pais.

REPRESENTAÇÃO

- Até agora a representação foi deixada de lado nas discussões apresentadas, considerou-se sempre o genótipo de tamanho fixo e linear.
- No entanto, para a definição dos mecanismos de reprodução é importante considerar a relação genótipo→fenótipo.
- Na formulação apresentada do algoritmo genético e dos operadores de recombinação e mutação para ele, é utilizado a representação binária como código universal.
- Porém, pode-se utilizar o algoritmo genético com outras representações, só que isso resulta na necessidade de propor, em alguns casos, operadores de reprodução específicos.
- A produção de operadores específicos as vezes pode ser trabalhosa mas, em geral, resulta num melhor desempenho do algoritmo.

REPRESENTAÇÃO

- Um dos problemas da representação binária ocorre, por exemplo, ao mapear problemas de número reais.
- Uma das formas de codificar números reais em binários consiste em discretizar o espaço dos números reais e a cada ponto atribuir um código binário.
- Com esse tipo de codificação pode ocorrer de que a alteração de um único gene da string binária resulte numa grande modificação no espaço dos números reais dependendo da ordem do gene.
- Outro caso é que números próximos no espaço dos reais possuem strings binárias de distância de Hamming máxima.
- Assim, deve-se tomar muito cuidado ao escolher a representação e os operadores que serão utilizados com ela.

RESUMINDO

- Um algoritmo evolutivo simples consiste em:
  - Uma população de pais de tamanho *m*
  - Uma população de descendentes de tamanho *n*
  - Um método para seleção dos pais
  - Um método para seleção dos sobreviventes
  - Um conjunto de operadores de reprodução
  - Um método interno para a representação dos indivíduos.
- Os algoritmos canônicos possuem a seguinte caracterização tradicional:

AE	*m*	*n*	Parent_sel	Survival_sel	mutação	crossover	Representação
EP	<20	*n=m*	uniforme	corte	sim	não	fenótipo
EE	<10	*n>=m*	uniforme	corte	sim	não	fenótipo
AG	>20	*n=m*	prop_aptidão	uniforme	sim	não	genótipo

ALGORITMOS EVOLUTIVOS COMO BUSCA PARALELA ADAPTATIVA

INTRODUÇÃO

- Os algoritmos vistos foram desenvolvidos como modelos de simulação de sistemas evolutivos simples.
- Eles não foram desenvolvidos com nenhuma aplicação de ciência da computação ou engenharia em mente ou até mesmo de que eles poderiam produzir alguma resposta.
- Por outro lado, não é necessário muito esforço para interpretar os algoritmos evolutivos como procedimentos de busca adaptativa paralela.
- Indivíduos iniciais aleatórios com movimentos aleatórios dão caminhos de busca, não como resultado de uma busca planejada pré-definida.
- Mas preferencialmente através de reorganizações dinâmicas como sinais estimando onde a comida pode ser encontrada.

INTRODUÇÃO

- Matematicamente, indivíduos são pontos do espaço dando dicas de onde se encontram as regiões de alta aptidão.
- As dinâmicas evolutiva simuladas produzem uma exploração adaptativa e guiada por aptidão do espaço de busca.
- Quando o processo evolutivo é encerrado, os resultados da busca podem ser vistos como a "resposta" a ser retornada pelo algoritmo evolutivo.
- Visualizando algoritmos evolutivos como mecanismos de busca paralela imediatamente abre uma rede de potenciais áreas de aplicação na ciência e engenharia.

INTRODUÇÃO

- Com o objetivo de obter soluções para problemas complexos com representações não-lineares, algumas decisões precisam ser tomadas entre duas opções:
  - Fazer suposições de linearidade dos problemas , ou
  - Desenvolver procedimentos efetivos de busca para sistemas não-lineares?
- Neste sentido algoritmos evolutivos tem sido vistos como paradigmas independentes do problema para desenvolver procedimentos de busca efetivos.

INTRODUÇÃO

- Os algoritmos evolutivos para serem procedimentos de busca eficientes necessitam de algumas decisões de projeto:
  - O que o indivíduo na população representa
  - Providenciar meios para calcular a função de aptidão
  - Decidir como descendentes são produzidos a partir dos pais
  - Especificar tamanho das populações de dinâmicas
  - Definir um critério de parada para o processo evolutivo
  - Retornar uma resposta
- Escolhas apropriadas para cada item é uma função de ambos, independente de domínio e específica do problema.
- Primeiramente veremos essas escolhas como independentes de domínio.

REPRESENTAÇÃO

- A primeira pergunta a ser feita quando consideramos utilizar um algoritmo evolutivo para solucionar problemas é: Em qual espaço eu quero que o algoritmo efetue a busca?
- A resposta mais simples é o espaço de soluções e a função de aptidão medir a distância para a solução esperada.
- Por exemplo, para o rendimento máximo de um sistema de manufatura, os indivíduos podem representar os muitos caminhos alternativos que o processo de manufatura pode ser configurado. E a aptidão de uma configuração particular pode ser determinada pela estimação de seu rendimento máximo por meio de uma simulação do processo.

REPRESENTAÇÃO

- Por simplicidade nosso foco será em algoritmos evolutivos que buscam no espaço de soluções.
- Mas, isso ainda deixa uma questão em aberto, como representar o espaço de soluções no algoritmo.
- Em muitos casos os problemas podem ser representados em mais de uma forma e uma ou mais delas podem ser mais adequadas para as técnicas evolutivas do que outras.
- Existem duas abordagens principais:
  - Abordagem fenotípica que utiliza o próprio fenótipo e precisa de operadores de reprodução específicos.
  - Abordagem genotípica que utiliza um mapeamento genótipo→fenótipo e em geral utiliza os operadores convencionais.

REPRESENTAÇÃO

- Ambas as abordagens tem suas vantagens e desvantagens.
- Uma abordagem fenotípica em geral permite uma maior utilização das propriedades específicas do problema, mas o tempo de desenvolvimento do AE é maior.
- Uma abordagem genetípica encoraja prototipagem rápida de novas aplicações, mas torna mais difícil tirar vantagens do conhecimento do domínio.
- A escolha de qual abordagem seguir depende fortemente de propriedades particulares do espaço de soluções a ser pesquisado.

OBJETOS LINEARES DE TAMANHO FIXO

- A mais simples e mais natural representação interna para um algoritmo evolutivo envolve indivíduos que consistem de vetores de genes de tamanho fixo.
- Então, espaços de soluções que são definidos como parâmetros N-dimensionais são simples e fáceis para representar internamente em um AE visto que soluções são descritas por vetores de tamanho fixo com comprimento N, e simples representações internas consideram cada gene um parâmetro.
- A única decisão que resta é se internamente os genes são representados fenotipicamente ou genotipicamente.
- No entanto, existem vários problemas cujas soluções não são naturalmente expressas como lineares, vetores de parâmetros de comprimento fixo. Por exemplo, os parâmetros de uma rede neural ou o agendamento de um trabalho.

OBJETOS NÃO-LINEARES DE COMPRIMENTO FIXO

- Também deve-se decidir se a representação é fenotípica ou genotípica.
- A abordagem genotípica pensa em maneira de linearizar a solução não-linear para poder aplicar a representação interna padrão.
- Por exemplo, um objeto matricial MxN pode ser linearizado em um vetor com comprimento M*N colocando as colunas uma após a outra. Neste caso operadores de mutação padrão podem ser aplicados sem problemas, mas a recombinação precisa ser adaptada, pois aplicar a recombinação no meio de uma coluna pode causar a perda de informação do problema.
- Por exemplo, se a matriz representa um grafo. Cada coluna indica todas as ligações chegando a um nó, e a linha todas as ligações saindo do nó.
- Assim, a recombinação deve preservar o significada das informações.

OBJETOS NÃO-LINEARES DE COMPRIMENTO FIXO

- Alternativamente, pode se utilizar abordagens fenotípicas onde os indivíduos são representados em seu espaço original.
- No caso das matrizes como uma matriz.
- Nesse caso as decisões de projeto devem se concentrar em desenvolver operadores de reprodução adequados para a representação escolhida.

OBJETOS DE COMPRIMENTO VARIÁVEL

- Soluções de comprimento variável também necessitam de cuidadoso pensamento de como melhor representá-las internamente.
- Geralmente objetos que são vistos naturalmente como listas de itens de comprimento variável, como conjuntos de regras ou seqüências de ações são mais fáceis de lidar.
- Nesse caso, cada item da lista pode ser visto como um gene e claramente extensões naturais dos operadores de mutação e crossover podem ser desenvolvidas.
- No caso da mutação, em adição a modificação de valores dos genes, pode-se remover ou adicionar itens.
- O crossover padrão pode ser utilizado levando em consideração que os indivíduos podem ter tamanhos diferentes.

OBJETOS DE COMPRIMENTO VARIÁVEL

- Um complicação, que será discutida futuramente, é o significado semântica dos itens na solução final do problema.
- Por exemplo, se a lista é lida da esquerda para a direita então a posição e a ordem dos genes é semanticamente importante.
- Isto significa que até pequenas mudanças no genoma podem resultar em grandes mudanças arbitrárias no comportamento e aptidão de um objeto.
- Nesse caso deve se ter muito cuidado ao desenvolver os operadores de reprodução para preservar a correlação de aptidão.
- Quando a ordem dos itens não interessa não são necessários cuidados especiais.

OBJETOS NÃO-LINEARES DE COMPRIMENTO VARIÁVEL

- Os aspectos mais difíceis de representação envolvem objetos que são não-lineares e de comprimento variável.
- Suponha, por exemplo, que deseja-se evoluir estruturas de grafos com formas e tamanhos variáveis.
- Adicionar e remover ligações entre nós é direto. Mas imagine que deseja-se remover e adicionar nós.
- Para representações por matrizes de adjacências é preciso remover uma linha e uma coluna para remover um nó, e para adicionar um nó é preciso adicionar uma linha e uma coluna.
- O ponto difícil é projetar operadores que façam isso e mantenham a correlação de aptidão.

OBJETOS NÃO-LINEARES DE COMPRIMENTO VARIÁVEL

- Conforme aumenta a complexidade dos objetos a serem evoluídos, torna-se mais difícil projetar operadores de reprodução eficientes que produzam somente soluções válidas.
- Uma solução é não se preocupar com isso e deixar que o processo evolutivo elimine essas soluções.
- Essa estratégia funciona bem para problemas onde as soluções precisam se adequar a um pequeno número de restrições.
- Mas pode ser um problema para estruturas com mais restrições, visto que a maioria das soluções será inválida e pouco progresso, ou nenhum, é feito para encontrar estruturas mais adequadas.