

Q = População Intermediária
 ex: Pa 0101101101 Ca 0111110110
 Pb 0111110110 cb 0101101101

29.07.22

S T Q Q S S

Algoritmo do GA

POPSIZE \leftarrow Setar Tamanho da população
 P \leftarrow {}

FOR $i=0$ a POPSIZE
 P \leftarrow P U {move ind. aleatório}

BEST \leftarrow P[0]

REPEAT

FOR EACH $P_i \in P$

CALCULO Fitness(P_i)

IF Fitness(P_i) < Fitness(BEST)

BEST \leftarrow P_i

$\rightarrow Q = \{ \}$

FOR $j=0$ POPSIZE/2 DO

$P_a \leftarrow$ SelecionarREPOSICAO(P)

$P_b \leftarrow$ SelecionarREPOSICAO(P)

$C_a, C_b \leftarrow$ CROSSOVER(Copiar(P_a), Copiar(P_b))

$Q \leftarrow Q \cup$ (MUTACAO(C_a), MUTACAO(C_b))

APPEND

END FOR

P \leftarrow Q

ATEQUE ESTOURE A GERAÇÃO, ou ACHOU
 UM VALOR VARIÁVEL

GERAÇÃO IND ALEATORIA

```
 $\vec{v} \leftarrow \langle v_1 \dots v_l \rangle$   
FOR  $i$  de 1 a  $l$   
    if  $0.5 > \text{Número aleatório entre } [0, 1]$   
         $v_i \leftarrow 1$   
    ELSE  
         $v_i \leftarrow 0$ 
```

retorna \vec{v}

CROSSOVER E MUTAÇÃO

ALGORITMO BITFLIP(\vec{v}) \rightarrow mutação

```
 $p \leftarrow \text{prob de ocorrer}$  (geralmente  $p = 1/2$ )  
 $\vec{v} \leftarrow \langle v_1 \dots v_l \rangle$   
FOR  $i = 1$  até  $l$  do  
    if  $p \geq \text{Número aleatório uniforme } [0, 1]$   
         $v_i \leftarrow \text{not}(v_i)$   
retorna  $\vec{v}$ 
```

ALGORITMO ONEPOINT CROSSOVER

```
 $v \leftarrow \text{primeiro vetor } \langle v_1 \dots v_l \rangle$   
 $w \leftarrow \text{segundo vetor } \langle w_1 \dots w_l \rangle$   
 $c \leftarrow \text{Número aleatório entre } 1 \text{ e } l$ 
```


29.07.22

D S T Q Q S S

if $c \neq 1$ Then
for i de c até l do
 $\text{Swap}(v_i, w_i)$

retorna \vec{v}, \vec{w}

ALGORITMO CROSSOVER 2pts

$\vec{v} \leftarrow \langle v_1, \dots, v_l \rangle$
 $\vec{w} \leftarrow \langle w_1, \dots, w_l \rangle$
 $c \leftarrow \text{randomico}[1, l]$
 $d \leftarrow \text{randomico}[1, l]$
if $c > d$
 $\text{Swap}(c, d)$

if $c \neq d$ Then
 for i de c até $d = 1$ do
 $\text{SWAP}(v_i, w_i)$
retorna \vec{v} e \vec{w}

ALGORITMO SELEÇÃO (Proporcional a fitness)

A cada iteração

$\vec{P} \leftarrow$ Copia os indivíduos $\langle p_1, \dots, p_n \rangle$
 $f \leftarrow$ Calcular as fitness dos $\langle f_1, f_2, \dots, f_n \rangle$
 indivíduos em \vec{P}

for $i = 2^{\text{a}}$ até o último

$$f_i = f_i + f_{i-1}$$

f_n vai conter a soma
total das fitness

```
n ← número aleatório de 0 a fe
for i de 2 até l do
  if  $f_{i-1} < n < f_i$ 
    retorna  $p_i$ 
```

retorna P_1

Torneio

```
P ← População
T ← Tamanho Torneo
BEST ← pegado aleatoriamente de P
for i de 2 até T
  NEXT ← ind. aleatório de P
  if  $Fitness(NEXT) > Fitness(BEST)$ 
    BEST ← NEXT
```

retorna BEST

Se $T=1 \rightarrow$ busca aleatória

Se T for grande o Torneo tende a selecionar o melhor indivíduo toda hora