

ALGORITMOS GENÉTICOS

Problemas com restrições



ALGORITMOS GENÉTICOS

Restrições

A grande maioria dos problemas envolvem restrições;

Existem dois tipos de restrições:

- do tipo *soft*, que são aquelas desejáveis, mas que podem ser desobedecidas se necessário;
- do tipo *hard*, que são aquelas que obrigatoriamente devem ser obedecidas;



ALGORITMOS GENÉTICOS

Restrições

Como lidar com as restrições?

Descarte: as soluções que não respeitam as restrições são simplesmente descartadas, ou seja, são excluídos da população do algoritmo genético;

Penalização: as soluções que não respeitam as restrições tem suas avaliações penalizadas, a fim de manter as características desses indivíduos;



ALGORITMOS GENÉTICOS

Restrições

Penalização

Tipos de funções com relação ao grau de violação (desvio):

Logarítmica

$$\text{Pen}(x) = \log_n (1 + \rho \cdot (\text{desvio}))$$

Linear

$$\text{Pen}(x) = \rho \cdot (\text{desvio})$$

Quadrática

$$\text{Pen}(x) = (\rho \cdot (\text{desvio}))^2$$

Onde ρ = constante



ALGORITMOS GENÉTICOS

Restrições

Como lidar com as restrições?

Reparo da solução: as soluções que violam as restrições são corrigidas por um algoritmo de reparo específico;

Decodificadores: transforma os cromossomos em soluções válidas;

ALGORITMOS GENÉTICOS

Restrições

Como lidar com as restrições?

Genocop I: auxilia o processo de otimização considerando somente restrições lineares;

- Sua estratégia consiste em diminuir o espaço de busca, tentando encontrar uma solução inicial buscando regiões possíveis;

ALGORITMOS GENÉTICOS

Restrições

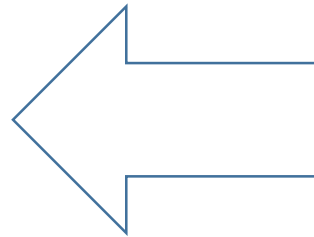
Como lidar com as restrições?

Genocop I

Exemplo: Otimizar f

$$f(x_1, x_2, x_3, x_4, x_5, x_6)$$

$$\begin{aligned} 2x_1 + x_2 + x_3 &= 6, \\ x_3 + x_5 - 3x_6 &= 10, \\ x_1 + 4x_4 &= 3, \\ x_2 + x_5 &\leq 120, \end{aligned}$$



$$\begin{aligned} -40 &\leq x_1 \leq 20 \\ 50 &\leq x_2 \leq 75 \\ 0 &\leq x_3 \leq 10 \\ 5 &\leq x_4 \leq 15 \\ 0 &\leq x_5 \leq 20 \\ -5 &\leq x_6 \leq 5. \end{aligned}$$

ALGORITMOS GENÉTICOS

Restrições

Como lidar com as restrições?

Genocop I

para cada x_k existe um intervalo possível $\langle \text{left}(k), \text{right}(k) \rangle$ onde as outras variáveis são fixas.

Assim, para o ponto possível $(x_4, x_5, x_6) = (10, 8, 2)$

$$\begin{array}{lll} X_4 \in [7.25, 10.375] & \text{enquanto } x_5=8 & \text{e } x_6=2 \\ X_5 \in [6, 11] & \text{enquanto } x_4=10 & \text{e } x_6=2 \\ X_6 \in [1, 2.666] & \text{enquanto } x_4=10 & \text{e } x_5=8 \end{array}$$



ALGORITMOS GENÉTICOS

Restrições

Como lidar com as restrições?

Genocop II: auxilia o processo de otimização considerando as restrições não lineares;

$$X_1^2 + X_4^5 + 0.78 \leq X_3^4$$

ALGORITMOS GENÉTICOS

Restrições

Como lidar com as restrições?

Genocop III: incorpora duas populações separadas:

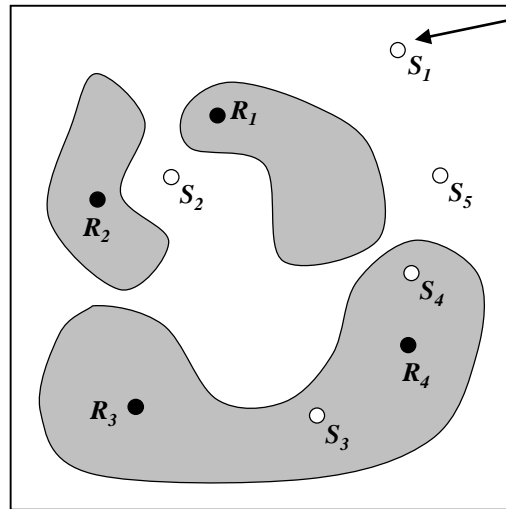
- população de busca: satisfaz as restrições lineares;
- população de referência: satisfaz todas as restrições;



ALGORITMOS GENÉTICOS

Genocop III

$Ps = \{S_1, S_2, S_3, S_4, S_5\}$ – satisfaz restrições lineares
 $Pr = \{R_1, R_2, R_3, R_4\}$ – satisfaz qualquer restrição



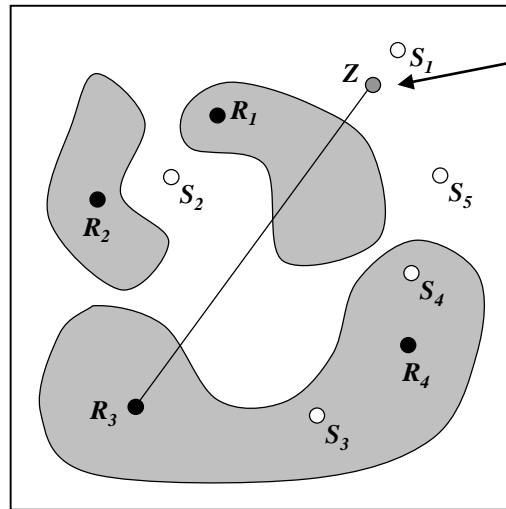
■ região viável
□ região não viável

```
begin
  P = p; // replacement probability
  if isfeasible(S) == false
    Z = aS + (1 - a)R; // a ∈ [0, 1]
    while isfeasible(Z) == false
      Z = aZ + (1 - a)R; // a ∈ [0, 1]
    end while
    if evaluation(Z) > evaluation(R)
      R = Z;
    end if
    if rand() ≤ P
      S = Z; // S is replaced by Z
    else
      evaluation(S) = evaluation(Z);
    end if
  end if
end
```

ALGORITMOS GENÉTICOS

Genocop III

$Ps = \{S_1, S_2, S_3, S_4, S_5\}$ – satisfaz restrições lineares
 $Pr = \{R_1, R_2, R_3, R_4\}$ – satisfaz qualquer restrição



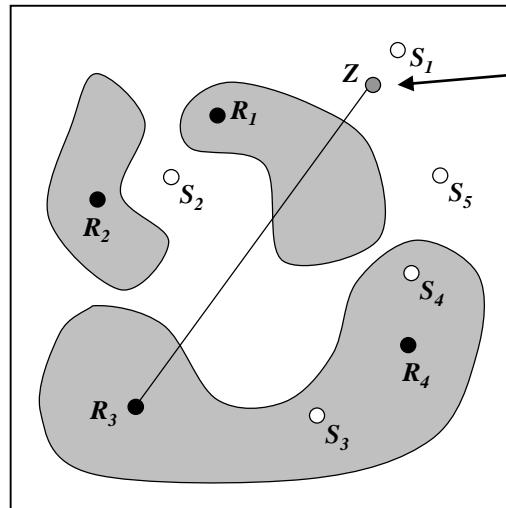
■ região viável
□ região não viável

```
begin
  P = p; // replacement probability
  if isfeasible(S) == false
    Z = aS + (1 - a)R; // a ∈ [0, 1]
    while isfeasible(Z) == false
      Z = aZ + (1 - a)R; // a ∈ [0, 1]
    end while
    if evaluation(Z) > evaluation(R)
      R = Z;
    end if
    if rand() ≤ P
      S = Z; // S is replaced by Z
    else
      evaluation(S) = evaluation(Z);
    end if
  end if
end
```

ALGORITMOS GENÉTICOS

Genocop III

$Ps = \{S_1, S_2, S_3, S_4, S_5\}$ – satisfaz restrições lineares
 $Pr = \{R_1, R_2, R_3, R_4\}$ – satisfaz qualquer restrição



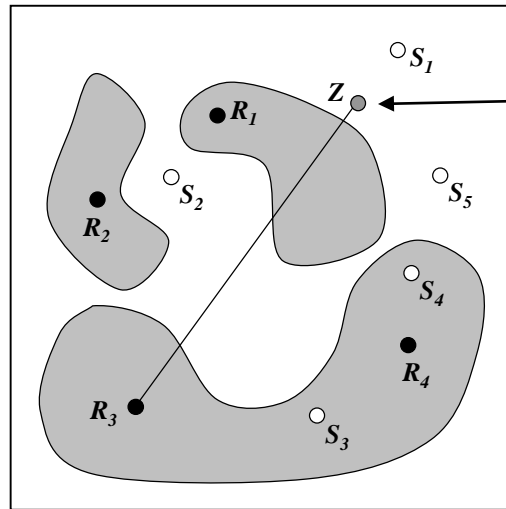
■ região viável
□ região não viável

```
begin
  P = p; // replacement probability
  if isfeasible(S) == false
    Z = aS + (1 - a)R; // a ∈ [0, 1]
    while isfeasible(Z) == false
      Z = aZ + (1 - a)R; // a ∈ [0, 1]
    end while
    if evaluation(Z) > evaluation(R)
      R = Z;
    end if
    if rand() ≤ P
      S = Z; // S is replaced by Z
    else
      evaluation(S) = evaluation(Z);
    end if
  end if
end
```

ALGORITMOS GENÉTICOS

Genocop III

$Ps = \{S_1, S_2, S_3, S_4, S_5\}$ – satisfaz restrições lineares
 $Pr = \{R_1, R_2, R_3, R_4\}$ – satisfaz qualquer restrição



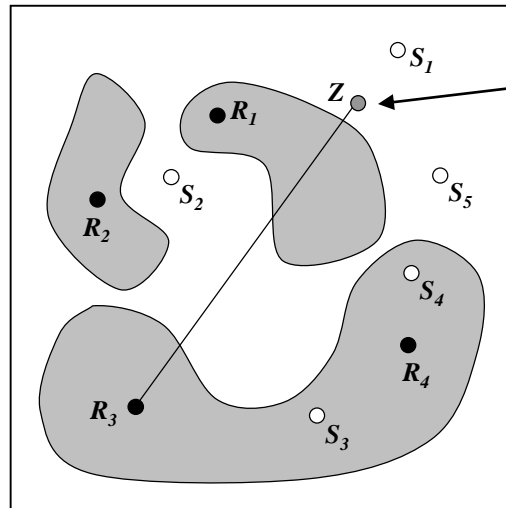
■ região viável
□ região não viável

```
begin
  P = p; // replacement probability
  if isfeasible(S) == false
    Z = aS + (1 - a)R; // a ∈ [0, 1]
    while isfeasible(Z) == false
      Z = aZ + (1 - a)R; // a ∈ [0, 1]
    end while
    if evaluation(Z) > evaluation(R)
      R = Z;
    end if
    if rand() ≤ P
      S = Z; // S is replaced by Z
    else
      evaluation(S) = evaluation(Z);
    end if
  end if
end
```

ALGORITMOS GENÉTICOS

Genocop III

$Ps = \{S_1, S_2, S_3, S_4, S_5\}$ – satisfaz restrições lineares
 $Pr = \{R_1, R_2, R_3, R_4\}$ – satisfaz qualquer restrição



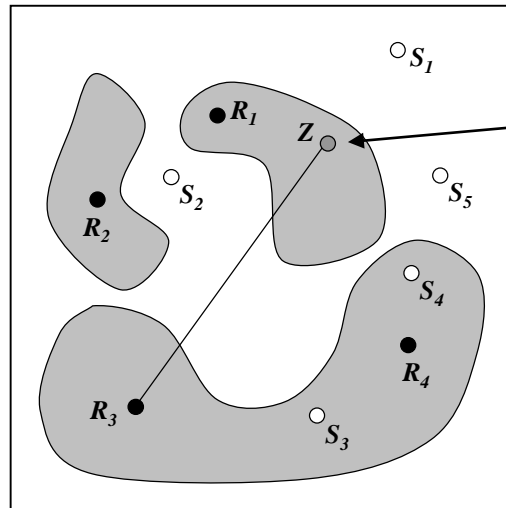
■ região viável
□ região não viável

```
begin
  P = p; // replacement probability
  if isfeasible(S) == false
    Z = aS + (1 - a)R; // a ∈ [0, 1]
    while isfeasible(Z) == false
      Z = aZ + (1 - a)R; // a ∈ [0, 1]
    end while
    if evaluation(Z) > evaluation(R)
      R = Z;
    end if
    if rand() ≤ P
      S = Z; // S is replaced by Z
    else
      evaluation(S) = evaluation(Z);
    end if
  end if
end
```

ALGORITMOS GENÉTICOS

Genocop III

$Ps = \{S_1, S_2, S_3, S_4, S_5\}$ – satisfaz restrições lineares
 $Pr = \{R_1, R_2, R_3, R_4\}$ – satisfaz qualquer restrição



■ região viável
□ região não viável

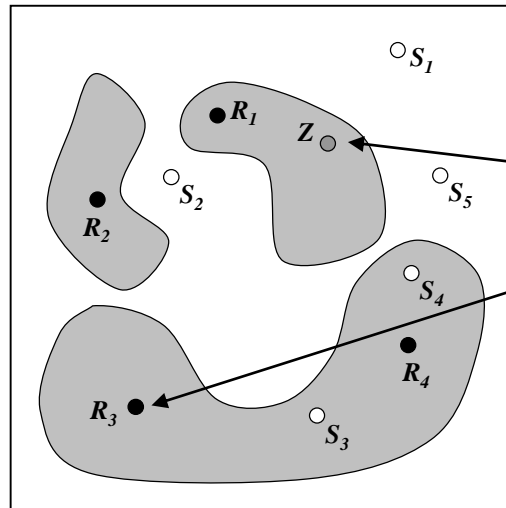
```
begin
  P = p; // replacement probability
  if isfeasible(S) == false
    Z = aS + (1 - a)R; // a ∈ [0, 1]
    while isfeasible(Z) == false
      Z = aZ + (1 - a)R; // a ∈ [0, 1]
    end while
    if evaluation(Z) > evaluation(R)
      R = Z;
    end if
    if rand() ≤ P
      S = Z; // S is replaced by Z
    else
      evaluation(S) = evaluation(Z);
    end if
  end if
end
```


ALGORITMOS GENÉTICOS

Genocop III

$Ps = \{S_1, S_2, S_3, S_4, S_5\}$ – satisfaz restrições lineares

$Pr = \{R_1, R_2, R_3, R_4\}$ – satisfaz qualquer restrição



■ região viável

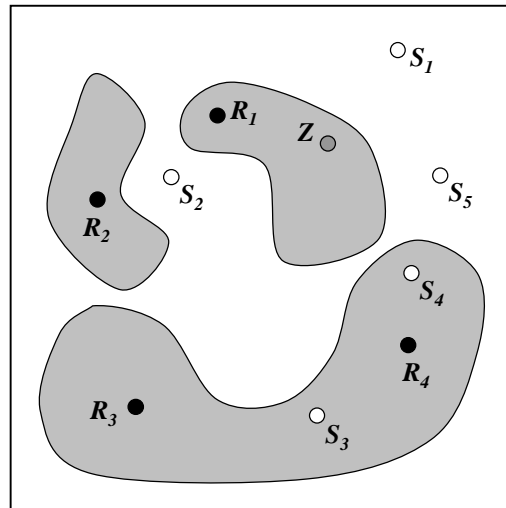
□ região não viável

```
begin
  P = p; // replacement probability
  if isfeasible(S) == false
    Z = aS + (1 - a)R; // a ∈ [0, 1]
    while isfeasible(Z) == false
      Z = aZ + (1 - a)R; // a ∈ [0, 1]
    end while
    if evaluation(Z) > evaluation(R)
      R = Z;
    end if
    if rand() ≤ P
      S = Z; // S is replaced by Z
    else
      evaluation(S) = evaluation(Z);
    end if
  end if
end
```

ALGORITMOS GENÉTICOS

Genocop III

$Ps = \{S_1, S_2, S_3, S_4, S_5\}$ – satisfaz restrições lineares
 $Pr = \{R_1, R_2, R_3, R_4\}$ – satisfaz qualquer restrição



■ região viável
□ região não viável

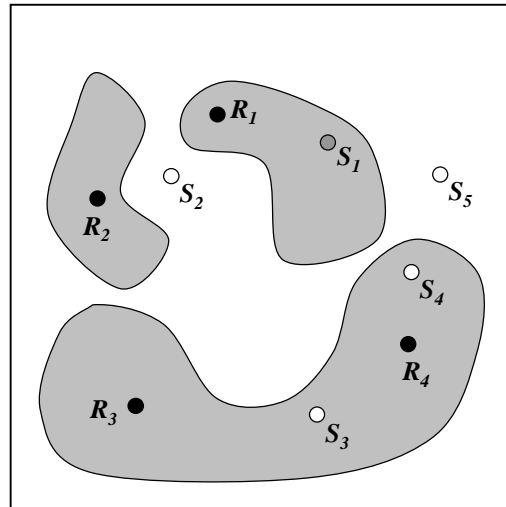
```
begin
  P = p; // replacement probability
  if isfeasible(S) == false
    Z = aS + (1 - a)R; // a ∈ [0, 1]
    while isfeasible(Z) == false
      Z = aZ + (1 - a)R; // a ∈ [0, 1]
    end while
    if evaluation(Z) > evaluation(R)
      R = Z;
    end if
    if rand() ≤ P
      S = Z; // S is replaced by Z
    else
      evaluation(S) = evaluation(Z);
    end if
  end if
end
```

ALGORITMOS GENÉTICOS

Genocop III

$Ps = \{S_1, S_2, S_3, S_4, S_5\}$ – satisfaz restrições lineares

$Pr = \{R_1, R_2, R_3, R_4\}$ – satisfaz qualquer restrição



■ região viável

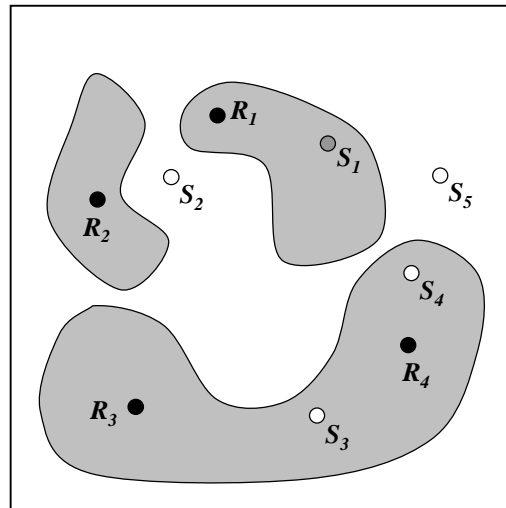
□ região não viável

```
begin
  P = p; // replacement probability
  if isfeasible(S) == false
    Z = aS + (1 - a)R; // a ∈ [0, 1]
    while isfeasible(Z) == false
      Z = aZ + (1 - a)R; // a ∈ [0, 1]
    end while
    if evaluation(Z) > evaluation(R)
      R = Z;
    end if
    if rand() ≤ P
      S = Z; // S is replaced by Z
    else
      evaluation(S) = evaluation(Z);
    end if
  end if
end
```

ALGORITMOS GENÉTICOS

Genocop III

$Ps = \{S_1, S_2, S_3, S_4, S_5\}$ – satisfaz restrições lineares
 $Pr = \{R_1, R_2, R_3, R_4\}$ – satisfaz qualquer restrição



■ região viável
□ região não viável

```
begin
  P = p; // replacement probability
  if isfeasible(S) == false
    Z = aS + (1 - a)R; // a ∈ [0, 1]
    while isfeasible(Z) == false
      Z = aZ + (1 - a)R; // a ∈ [0, 1]
    end while
    if evaluation(Z) > evaluation(R)
      R = Z;
    end if
    if rand() ≤ P
      S = Z; // S is replaced by Z
    else
      evaluation(S) = evaluation(Z);
    end if
  end if
end
```