



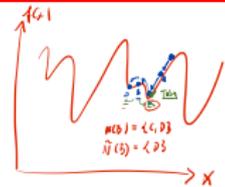
## HEURÍSTICAS 3

Marcus Ritt

INF 05010 – Otimização combinatória — <2020-06-17  
qua>

1. Heurísticas de busca local
2. Heurísticas de recombinação
3. Algoritmos genéticos

Vizinhança + solução inicial



### Ideias centrais:

- Seleciona o melhor vizinho, mas aceita soluções piores.
- Para evitar ciclagem: não visitar soluções visitadas recentemente ("tabu")

### Técnica central:

- Introduce uma memória de curta duração ("~~lista tabu~~")
- A memória contém
  - soluções recentes (raro, inefetivo)
  - atributos de soluções recentes

BuscaTabu( $s$ ) =

{ mantém a melhor solução  $s^*$  }

Inicialização:

$T := \emptyset$

while critério de parada não satisfeito

$s :=$  seleciona  $s' \in \tilde{N}(s)$  com  $\min f(s')$

insira movimento em  $T$  (a lista tabu)

end while

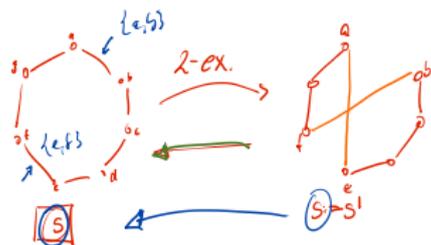
return  $s^*$

*melhor vizinho*

*Vizinhança reduzida*

$\tilde{N}(s) : N(s)$  vizinhos de  $s$   
sem soluções (nr.) tabu.

## Memória de curta duração em atributos



$$S = \{ \{a,b\}, \{b,c\}, \dots, \{e,a\} \}$$

Atributos: arestas.

⇒ Declarar atributos tabu, por um número fixo de iterações (duração tabu, tabu tenure).

## Tabela tabu

	a	b	c	d	e	f	g
a		At					
b							
c							
d							
e						At	
f							
g							

 $K = \text{tabu}$ 
 $i + \bar{t}$ 
 $\bar{T} = 5$  Dur. tabu

 $i = 12$ 
 $\{a, b\} = \text{tabu até it. } 17$   
 $\{c, d\} = \text{_____}$ 

Consequência: posso não ver novas soluções ótimas.

⇒ Critério de aspiração: exceções do tabu; exemplo: melhor solução já visitada.

**Ideais centrais:**

- Manter ~~uma~~ conjunto de soluções ("população de indivíduos")
- Consequência: possibilidade de recombinação para produzir novas soluções
- Exemplos: algoritmos genéticos e meméticos, busca dispersa

Scatter search

Técnica central:

- **Representação** de uma solução ("genótipo") e solução inteira ("fenótipo")
- Operador de **recombinação**
- Operador de **modificação** ("**mutação**")

**Otimização****Solução****Representação** de uma solução**Elemento** de uma solução**Valor** de um elemento**Conjunto** de solução**Recombinação** de solução**Modificação** de uma solução**Genética****Indivíduo/Fenótipo****Genótipo****Gene****Alelo****População****Cruzamento**

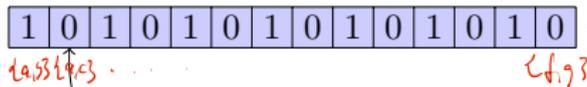
Crossover

**Mutação**

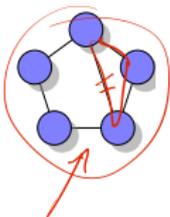
Mutation

$\{0,1\}^n$  $n$  genes / elementosRepresentação  $A^l$ 

cromossomo



Mapeamento

Solução  $S$ 

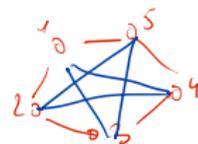
gene com alelos 0,1

Ex. (PCV)

Soluções = permutações de cidades

$$\left| \begin{array}{l} \pi_1 = (1 \ 2 \ 3 \ 4 \ 5) \\ \pi_2 = (3 \ 1 \ 4 \ 2 \ 5) \end{array} \right|$$

Soluções infactíveis: se permitir, preciso ou penalizar infactibilidade na função objetivo; ou incluir um mecanismo de „concertar“ soluções infactíveis.



$(\mu)$   
 # de sol. na pop.

for  $i = 1, \dots, \mu$ :  
 | cria sol.  $i$ .  
 Alg construtivo

- **Inicializar** a população

- **Repete:**

1. **Avalia** a população
2. **Seleciona pais** para recombinação
3. **Aplica operadores** de **recombinação** e **mutação**
4. **Seleciona** a **nova população**

Prob. prec

Prob. pmut

Ex.: somente mutação e  
 pop. de tamanho 1:  
evolution strategies.

≈ Busca local estocástica.

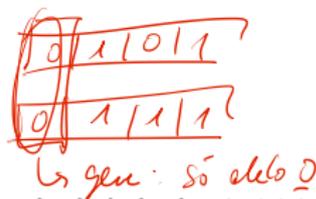
AG: meta-heurística com  
 mais parâmetros entre as  
 MH discutidas.

⇒ Mais difícil de calibrar.

- Heurísticas **constructivas** (e.g. ~~algoritmo guloso randomizado~~)
- Tamanho  $\mu$ : custo por iteração versus cobertura
- Critério de **Reeves** para soluções binários: para alcançar toda solução precisa todo alelo em todos genes. Probabilidade disso:

$$(1 - 2^{1-\mu})^n$$

com  $n$  genes.



- Exemplo: Com  $n = 50$  uma cobertura com probabilidade 0.999 precisa

$$\mu \geq 1 - \log_2 \left( 1 - \sqrt[50]{0.999} \right) \approx 16.61$$

• **Princípio:** selecionar os mais aptos

Menor valor na f.o. (min.)

Valor da f.o. da i-ésima sol.

1 • Proporcional à aptidão:  $\propto f_i$

2 • Proporcional à posição (transformada):  $\propto r(i)$ , e.g.  $r(i) = i^{-\tau}$

3 • Torneio: melhor de um  $k$ -conjunto aleatório

$\tau = 1$

1)  $f_1, f_2, \dots, f_n \leftarrow f.o.$

$$p_i = \frac{f_i}{\sum_{i \in [n]} f_i}$$

Max  $f_1 \quad f_2 \quad \dots \quad f_n$

$\frac{f_1}{99}$	$\frac{f_2}{1}$	$\Rightarrow$	$\frac{p_1}{0.99}$	$\frac{p_2}{0.01}$	$99 \times$
------------------	-----------------	---------------	--------------------	--------------------	-------------

2) Min.

$$f_1 \leq f_2 \leq f_3 \leq \dots \leq f_n$$

Max.

$$f_1 \geq f_2 \geq f_3 \geq \dots \geq f_n$$

$\frac{f_1}{99}$	$\frac{f_2}{1}$
$\frac{p_{2,1}}{99}$	$\frac{p_{2,2}}{1}$

$f_1$	$f_2$
$99$	$1$

$i$	$r_i$
$1 \ 2 \ 3$	$\frac{1}{1} \ \frac{1}{2} \ \frac{1}{3}$

$p_i$	$1$	$2$	$3$
	$\frac{1}{2}$	$\frac{1}{3}$	$\dots$

$\mu$  : # de sol. na pop.

$\lambda$  : # de descendentes criados

$$\lambda = \mu$$

$$\lambda \geq \mu$$

$(\mu, \lambda)$

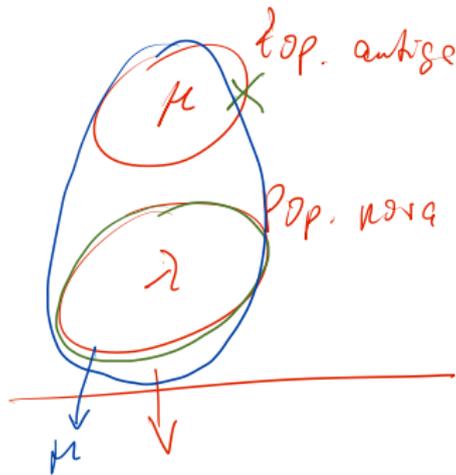
$(\lambda, \mu)$ : melhores entre os filhos

$(\lambda + \mu)$ : melhores entre os todos

• Elitismo

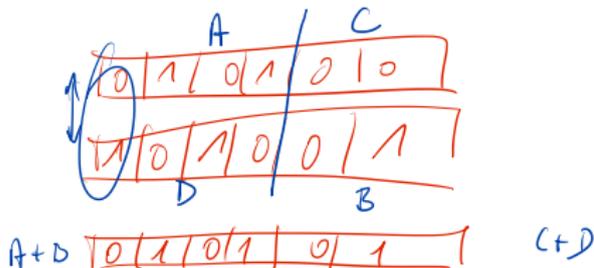
• Estado de equilíbrio

$$(\mu + 1) \quad \lambda = 1$$



Nova pop. na próx. iteração (geração)?

- Recombina **duas** (ou mais) soluções numa nova solução
- Muitos operadores conhecidos na **literatura**. Exemplos:
  - Representação binária: cruzamento em um ou mais pontos, cruzamento uniforme *uniform crossover*
  - Representação por permutação: cruzamento ordenado (OX), ciclico (CX).
    - OX: copia segmento, preenche resto em orden
    - CX: copia ciclo, preenche resto em orden



- Aplicar uma pequena modificação
- Em geral: movimento aleatório em alguma vizinhança
- Algoritmo memético: aplica uma busca local!