

*Prof. Lucas de Souza Batista - DEE/EE/UFMG*

---

# Otimização de Redes

Iterated Local Search (ILS)

---

---

# ILS

---

- ❖ A ideia por trás do ILS foi inicialmente apresentada por J. Baxter (1981):
  - ❖ *Local optima avoidance in depot location. J. Oper. Res. Soc. (1981)*

---

# ILS

---

- ❖ Princípio simples;
- ❖ Fácil implementação;
- ❖ Vários resultados estado-da-arte;
- ❖ Longa história (proposto na literatura com diferentes nomes):
  - ❖ E.B. Baum, 1986;
  - ❖ D.S. Johnson, 1990;
  - ❖ O. Martin et al., 1991, 1996;
  - ❖ D. Applegate, 2003

---

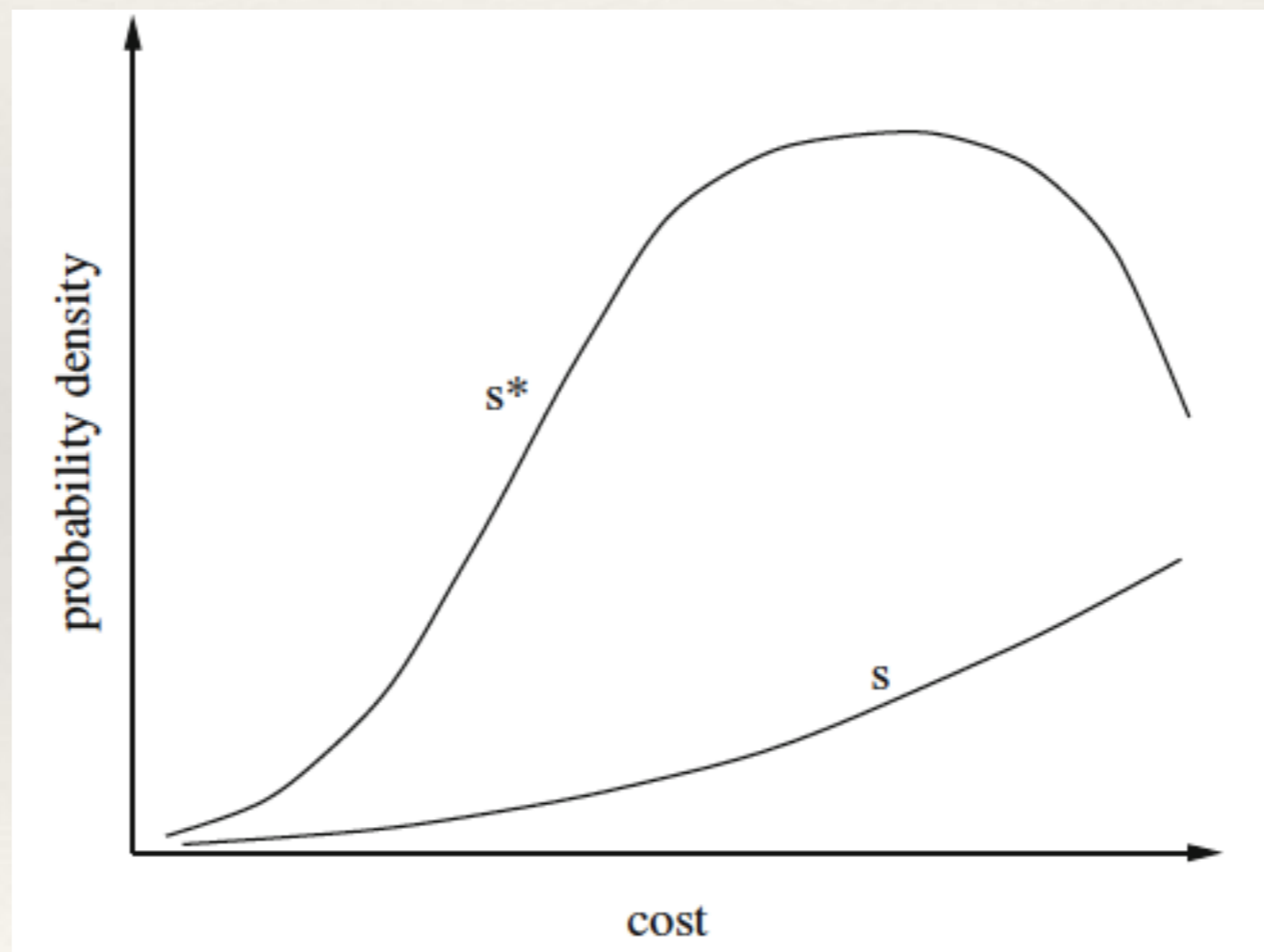
# ILS

---

- ❖ ILS gera uma **sequência** de soluções ótimas por meio de uma **heurística**;
- ❖ Conduz a soluções muito melhores do que se fosse considerado uma **heurística com recomeços**.

# Distribuição de Custos

- ❖ Suponha  $s \in S$  e  $s^* \in S^*$  tomados aleatoriamente



---

# Random Restart

---

- ❖ Gera múltiplas soluções ótimas  $s^*$  independentes;
- ❖ Não é muito efetivo na prática;
- ❖ Em instâncias grandes:
  - ❖ as soluções têm custo médio acima do ótimo, por um percentual fixo;
  - ❖ a distribuição dos custos das soluções tem um pico em torno da média, que fica mais pronunciado à medida em que o problema cresce.

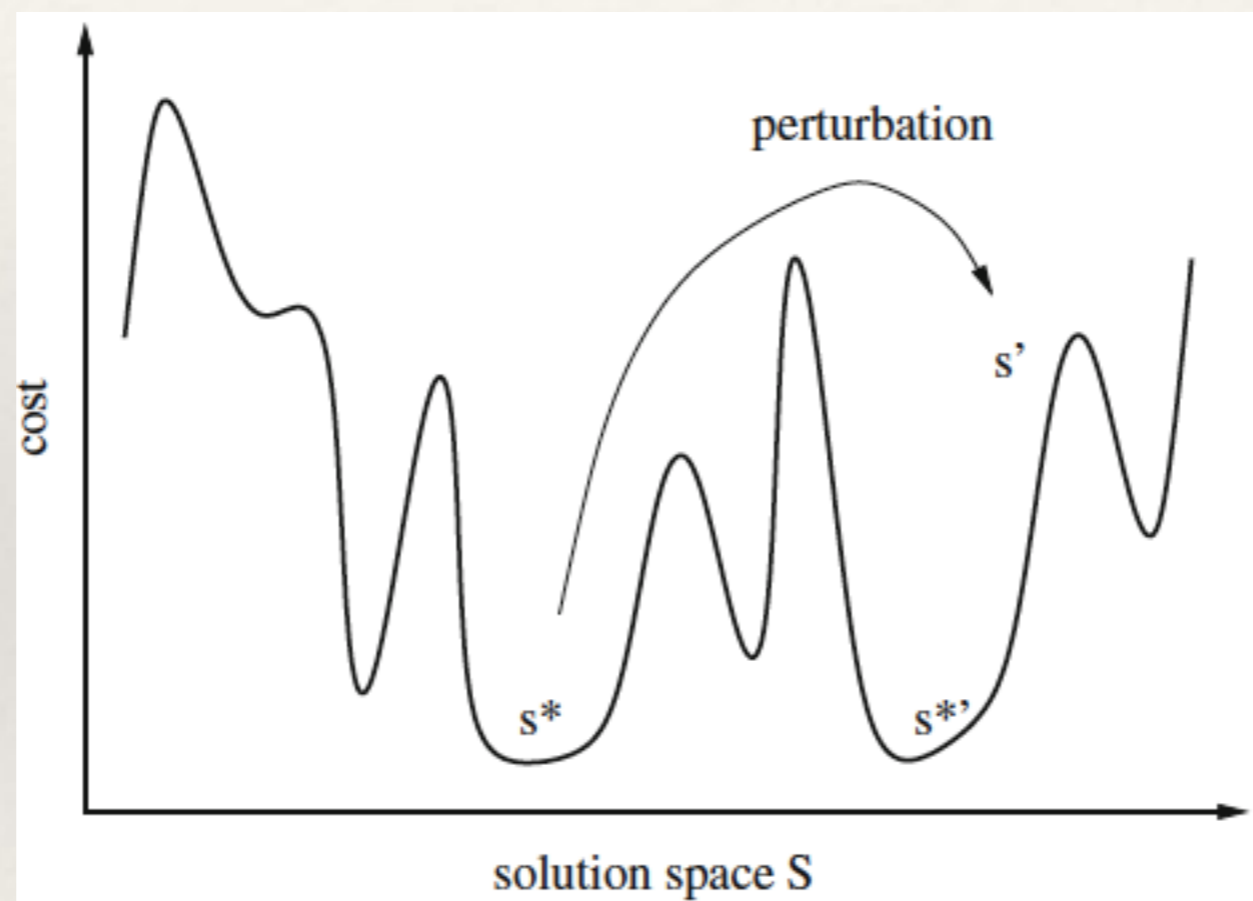
---

# Princípio do ILS

---

- ❖ Realiza buscas em  $S^*$
- ❖ *LocalSearch* conduz de um espaço grande  $S$  a um espaço reduzido  $S^*$ ;
- ❖ Define uma busca polarizada em  $S^*$ :
  - ❖ dada uma solução  $s^*$ , perturbe-a:  $s^* \longrightarrow s'$
  - ❖ aplique *LocalSearch*:  $s' \longrightarrow s^{*'}$
  - ❖ aplique critério de aceitação:  $s^*, s^{*'} \longrightarrow s_{new}^*$

# Princípio do ILS



---

# Procedimento Geral do ILS

---

*Algoritmo: Iterated local search*

$s_0 = \text{GenerateInitialSolution}$

$s^* = \text{LocalSearch}(s_0)$

*repeat*

$s' = \text{Perturbation}(s^*, \text{history})$

$s^{*'} = \text{LocalSearch}(s')$

$s^* = \text{AcceptanceCriterion}(s^*, s^{*'}, \text{history})$

*until termination condition met*

---

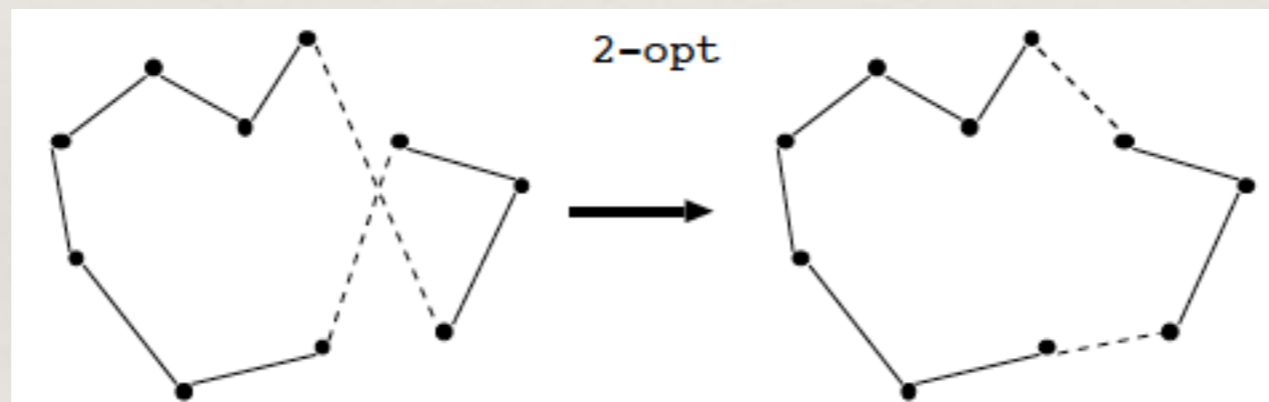
# ILS

---

- ❖ Desempenho depende da interação entre seus operadores;
- ❖ Versão básica do ILS:
  - ❖ *GenerateInitialSolution*: aleatório ou via heurística construtiva;
  - ❖ *LocalSearch*: inúmeras possibilidades (SA, VND);
  - ❖ *Perturbation*: movimento aleatório na estrutura de vizinhança;
  - ❖ *AcceptanceCriterion*: força o decrescimento do custo.
- ❖ ILS básico conduz frequentemente a um bom desempenho;
- ❖ ILS básico requer poucas linhas adicionais de código.

# Exemplo de Aplicação do ILS

- ❖ ILS básico para TSP
  - ❖ *GenerateInitialSolution*: heurística gulosa;
  - ❖ *LocalSearch*: 2-opt, 3-opt, LK, etc.



- ❖ *Perturbation*: 4-opt;
- ❖ *AcceptanceCriterion*: aceita  $s^{*'}$  somente se  $f(s^{*'}) \leq f(s^*)$ .

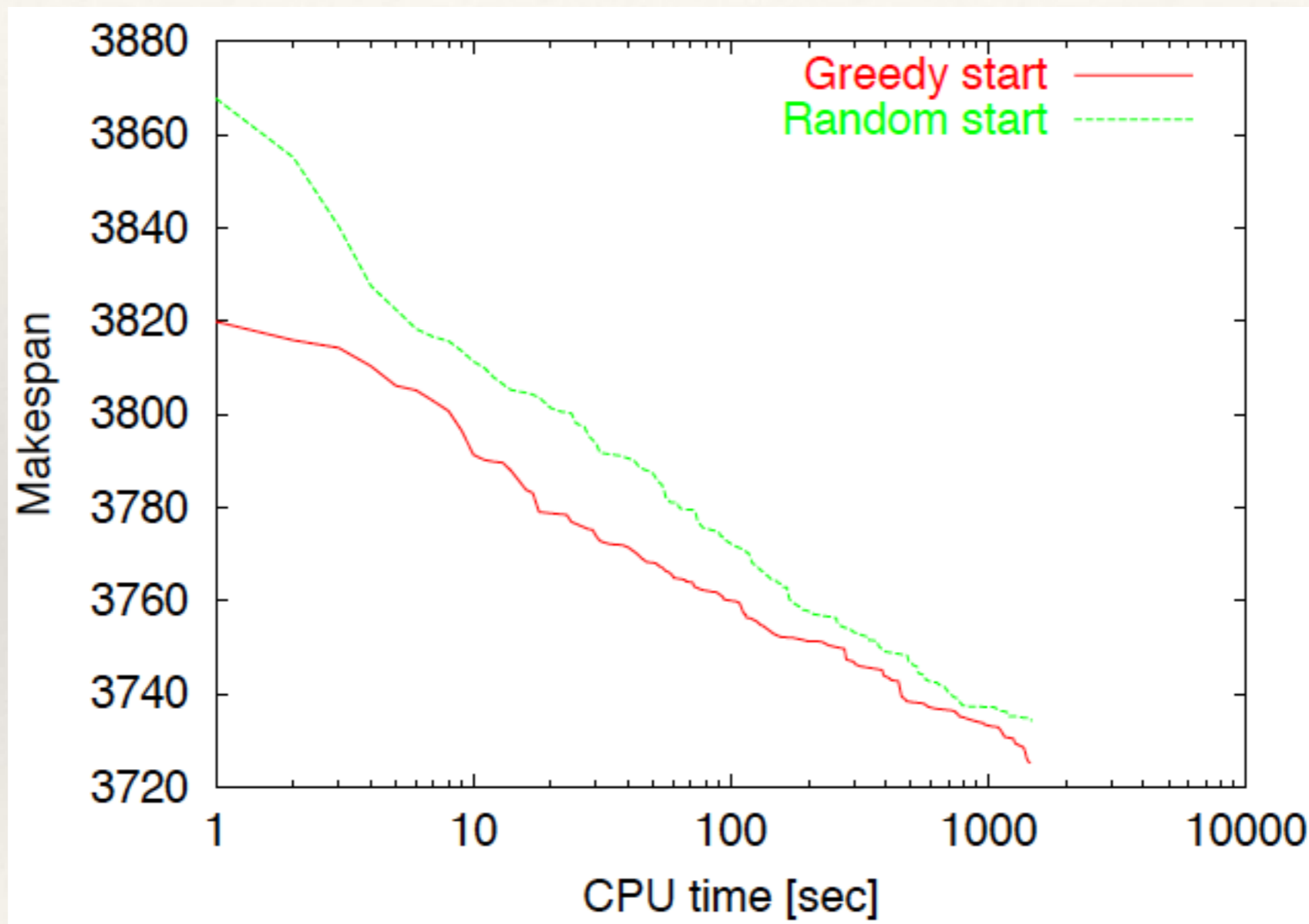
---

# ILS - Solução Inicial

---

- ❖ Determina o ponto inicial  $s_0^*$  de busca em  $S^*$ ;
- ❖ Solução inicial: aleatória  $\times$  gulosa;
- ❖ No geral, soluções gulosas são recomendáveis.

# ILS - Solução Inicial



---

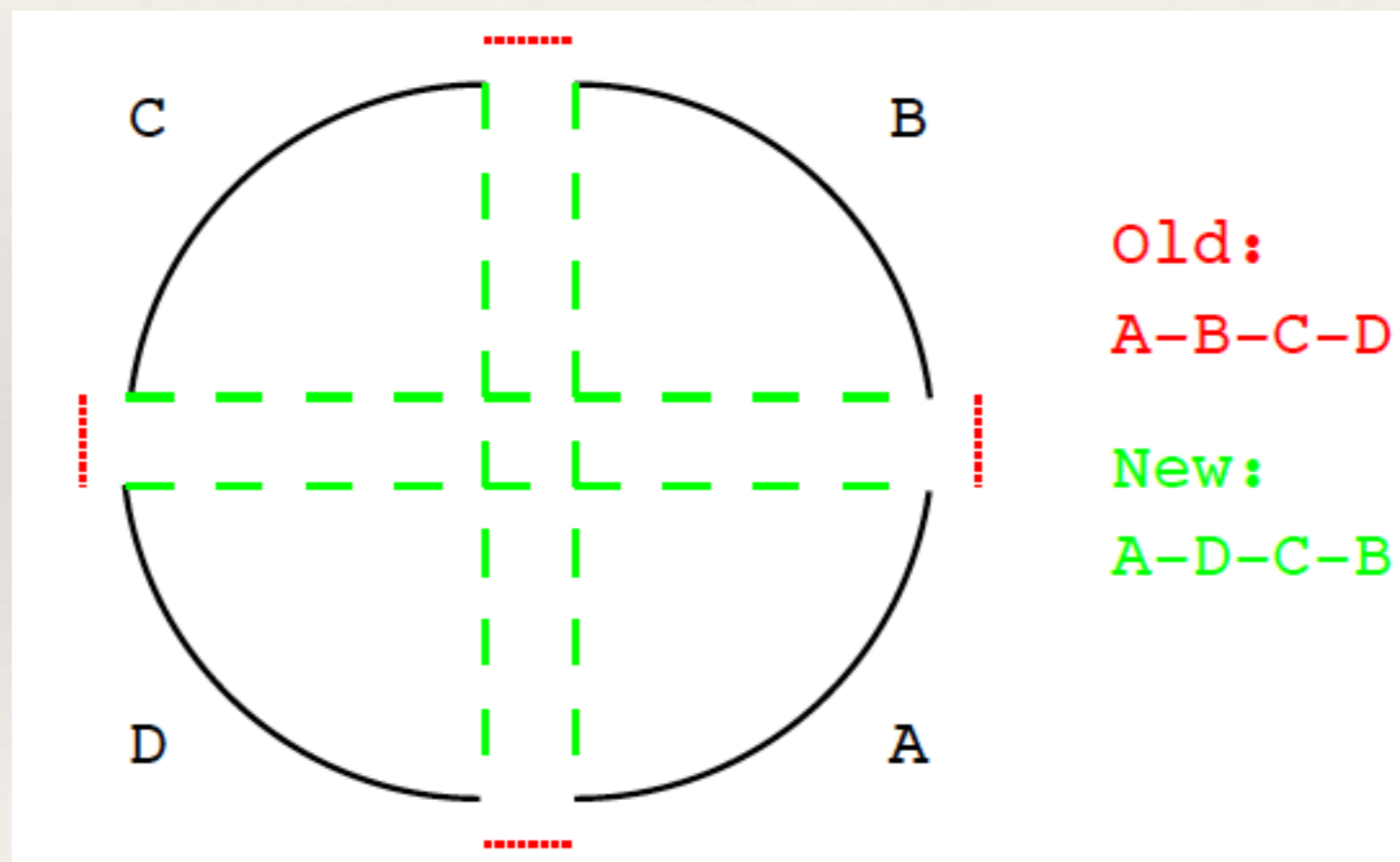
# ILS - Perturbação

---

- ❖ A **força** da perturbação é um fator importante:
  - ❖  **muito forte**: similar à reinicialização aleatória;
  - ❖  **muito fraca**: *LocalSearch* pode desfazer a perturbação.
- ❖ A força da perturbação pode variar ao longo do tempo;
- ❖ A perturbação deve ser complementar à *LocalSearch*.

# ILS - Perturbação

- ❖ Exemplo de perturbação muito comum no TSP:



---

# ILS - Perturbação

---

- ❖ Perturbações Adaptativas:
  - ❖ o tamanho da perturbação pode variar no tempo:
    - ❖ VNS básico
  - ❖ o tamanho da perturbação pode ser adaptado no tempo:
    - ❖ *reactive search*
- ❖ Mecanismos Complexos de Perturbação:
  - ❖ otimização de subproblemas;
  - ❖ modificação dos dados de entrada da instância.

# ILS - Critério de Aceitação

- ❖ *AcceptanceCriterion* possui forte influência na natureza e efetividade da busca em  $S^*$ ;
- ❖ Controla o balanço entre *intensificação* e *diversidade*;
- ❖ *Extreme intensification*:

$$\text{Better}(s^*, s^{*'}, \text{history}) = \begin{cases} s^{*'} & \text{if } \mathcal{C}(s^{*'}) < \mathcal{C}(s^*) \\ s^* & \text{otherwise.} \end{cases}$$

- ❖ *Extreme diversification*:

$$\text{RW}(s^*, s^{*'}, \text{history}) = s^{*'}.$$

# ILS - Critério de Aceitação

- ❖ Inúmeras escolhas intermediárias são possíveis:

$$\text{Restart}(s^*, s^{*'}, \text{history}) = \begin{cases} s^{*'} & \text{if } \mathcal{C}(s^{*'}) < \mathcal{C}(s^*) \\ s & \text{if } \mathcal{C}(s^{*'}) \geq \mathcal{C}(s^*) \text{ and } i - i_{\text{last}} > i_r \\ s^* & \text{otherwise.} \end{cases}$$

em que  $i_r$  representa um determinado número de iterações sem melhora, e  $s$  pode ser gerada de diferentes formas.

---

# ILS - Busca Local

---

- ❖ *LocalSearch* pode ser qualquer método de refinamento;
- ❖ Aconselha-se, entretanto, *otimizar* essa estratégia para a aplicação em questão.

---

# ILS - Busca Local

---

- ❖ Algoritmos complexos de busca local:
  - ❖ *variable depth local search*
  - ❖ *variable neighborhood descent* (VND)
  - ❖ vários possibilidades:
    - ❖ *tabu search*
    - ❖ *simulated annealing*
    - ❖ *dynamic local search*

---

# ILS - Busca Local

---

- ❖ Efetividade da busca local:
  - ❖ **frequentemente**, quanto mais efetiva é a busca local, melhor o desempenho do ILS;
  - ❖ **às vezes**, é preferível uma busca local rápida ao custo de uma menor efetividade.
- ❖ O **compromisso** entre efetividade e eficiência da busca local é um importante ponto a ser considerado na otimização do algoritmo ILS.

---

# Reference

---

- ❖ M. Gendreau, J.-Y. Potvin (eds.), *Handbook of Metaheuristics*, Springer, 2nd ed., 2010.