

Variable Neighborhood Search (VNS)

Busca na Vizinhança Variável

Prof. Ademir A. Constantino
Departamento de Informática
Universidade Estadual de Maringá
www.din.uem.br/~ademir

Variable Neighborhood Descent (VND)

- Proposto por Nenad Mladenovic 1995
- Método de busca local que explora o espaço de soluções através de trocas sistemáticas de vizinhança.
- Considera um conjunto de r estruturas de vizinhança.
- Este método explora a melhor solução de cada vizinhança.

VND

```
1  Seja  $s_0$  uma solução inicial e  $r$  o número de estruturas de
   vizinhança;
2   $s' \leftarrow s_0$ ;                                {Solução corrente}
3   $k \leftarrow 1$ ;                                {Tipo de estrutura de vizinhança}
4  enquanto ( $k \leq r$ ) faça
5      Encontre o melhor vizinho  $s'' \in N^{(k)}(s')$ ;
6      se (  $f(s'') < f(s')$  )
7          então  $s' \leftarrow s''$ ;  $k \leftarrow 1$ ;
8          senão  $k \leftarrow k + 1$ ;
9      fim-se;
10 fim-enquanto;
11 Retorne  $s'$ ;
fim VND;
```

Características

- Sugere-se que as vizinhanças sejam ordenadas de modo que primeiras sejam as menores, ou seja, que as primeiras envolvam movimentos mais fáceis que as últimas.
- Assim, geralmente, o gasto computacional cresce quando mudamos de vizinhança.
- Na prática, em cada vizinhança $N^k(x)$, podemos encontrar
 - a melhor solução, x^* ; (*best improvement*)
 - a primeira solução x' tal que $f(x') < f(x)$. (*first improvement*)

VNS básico

- Proposto por Nenad Mladenovic & Pierre Hansen em 1997
- Supor que exista r vizinhanças para o problema, isto é, N^1, N^2, \dots, N^r
- Método de busca local que explora o espaço de soluções através de trocas sistemáticas de vizinhança.
- Se a primeira solução vizinha não é aprimorante, então é a próxima vizinhança é pesquisada.

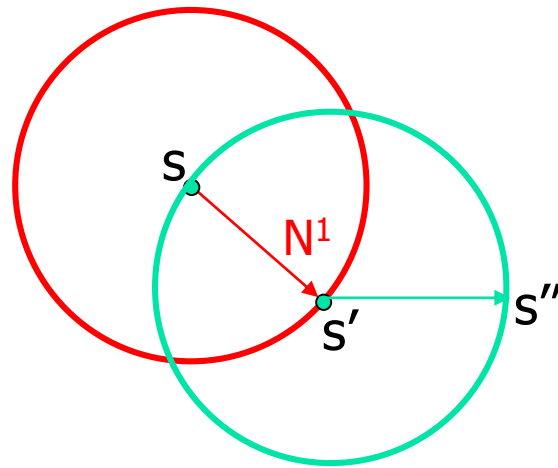
VNS básico

- Combina:
 - A seleção aleatória de um ponto de $N^k(x)$
 - A aplicação de um algoritmo de busca local
 - “*first improvement*” é o mais comum, mas
 - “*best improvement*” também pode ser usado.

VNS Básico

```
1  Seja  $s_0$  uma solução inicial e  $r$  o número de estruturas de
   vizinhança;
2   $s^* \leftarrow s_0$ ;           {Solução corrente}
3  enquanto (Critério de parada não satisfeito) faça
4       $k \leftarrow 1$ ;         {Tipo de estrutura de vizinhança}
5      enquanto ( $k \leq r$ ) faça
6          Gere um vizinho qualquer  $s' \in N^{(k)}(s^*)$ ;
7           $s'' \leftarrow \text{BuscaLocal}(s')$ ;
8          se (  $f(s'') < f(s^*)$  )
9              então  $s^* \leftarrow s''$ ;  $k \leftarrow 1$ ;
10             senão  $k \leftarrow k + 1$ ;
11             fim-se;
12     fim-enquanto;
13 fim-enquanto;
14 Retorne  $s$ ;
fim VNS;
```

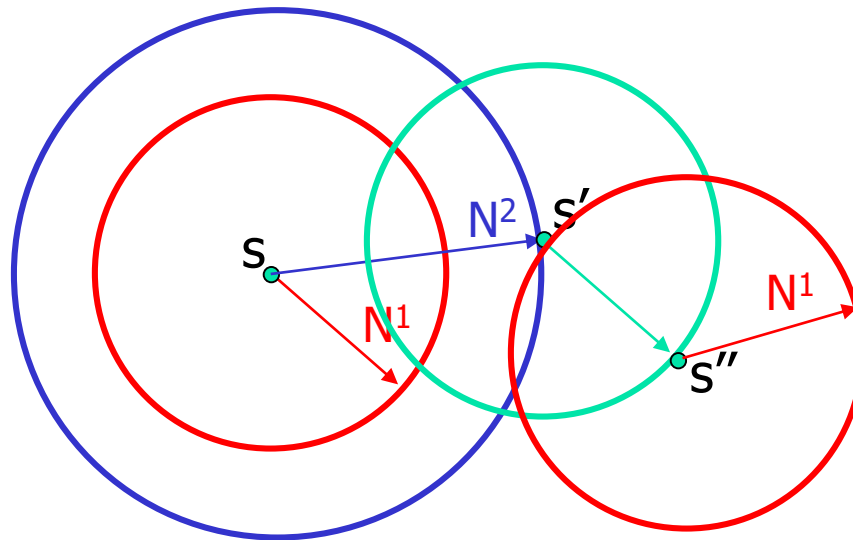
Variable Neighborhood Search (VNS)



s'' aceito se $f(s'') < f(s)$

Minimização

Variable Neighborhood Search (VNS)

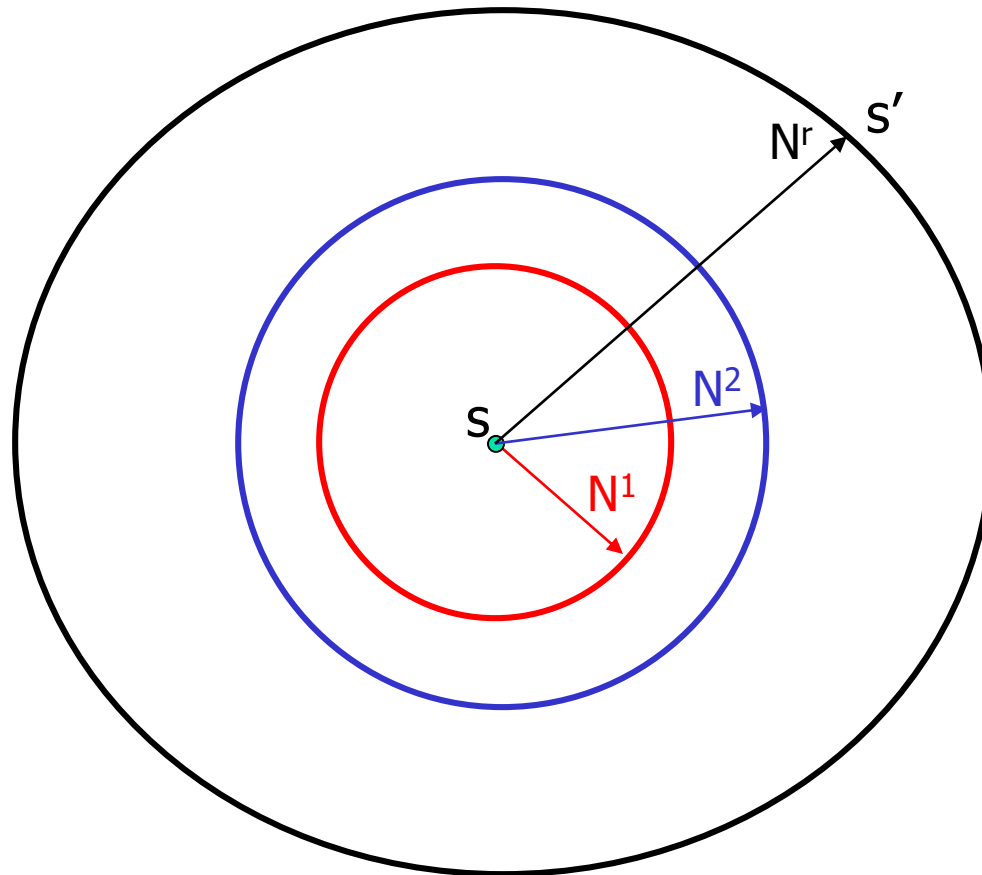


Se f não melhora na vizinhança de x' em N^1 , tenta-se novo x' em N^2 .

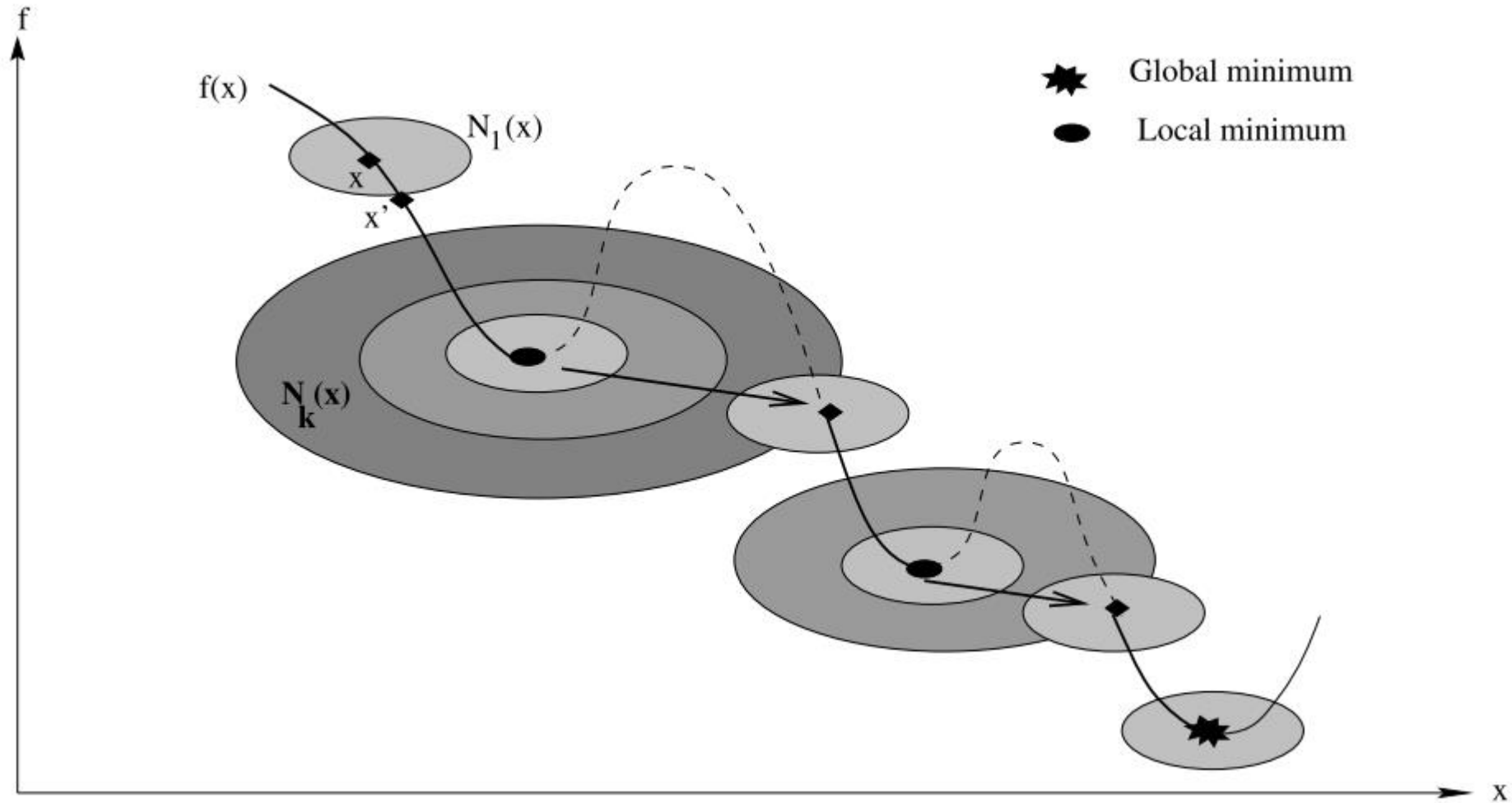
Nova busca local é feita na vizinhança de x' de N^2 .

s'' aceito se $f(s'') < f(s)$ e retorna para N^1

Variable Neighborhood Search (VNS)



Variable Neighborhood Search (VNS)



VNS Reduzido

VNS sem busca local.

```
1  Seja  $s_0$  uma solução inicial e  $r$  o número de estruturas de
   vizinhança;
2   $s^* \leftarrow s_0$ ;           {Solução corrente}
3  enquanto (Critério de parada não satisfeito) faça
4       $k \leftarrow 1$ ;         {Tipo de estrutura de vizinhança}
5      enquanto ( $k \leq r$ ) faça
6          Gere um vizinho qualquer  $s' \in N^{(k)}(s^*)$ ;
7          se (  $f(s'') < f(s^*)$  )
8              então  $s^* \leftarrow s''$ ;  $k \leftarrow 1$ ;
9              senão  $k \leftarrow k + 1$ ;
10         fim-se;
11     fim-enquanto;
12 fim-enquanto;
13 Retorne  $s$ ;
fim VNS;
```

VNS Geral= VNS Básico + VND

```
1  Seja  $s_0$  uma solução inicial e  
    $r$  o número de estruturas de vizinhança,  
    $r'$  o número de estrutura de vizinhança para o VND;  
2   $s^* \leftarrow s_0$ ;           {Solução corrente}  
3  enquanto (Critério de parada não satisfeito) faça  
4       $k \leftarrow 1$ ;           {Tipo de estrutura de vizinhança}  
5      enquanto ( $k \leq r$ ) faça  
6          Gere um vizinho qualquer  $s' \in N^{(k)}(s^*)$ ;  
7           $s'' \leftarrow \text{VND}(s', r')$ ;  
8          se (  $f(s'') < f(s^*)$  )  
9              então  $s^* \leftarrow s''$ ;  $k \leftarrow 1$ ;  
10             senão  $k \leftarrow k + 1$ ;  
11             fim-se;  
12     fim-enquanto;  
13 fim-enquanto;  
14 Retorne  $s$ ;  
fim VNS;
```

VNS Geral= VNS Básico + VND

```
Function GVNS( $x, k'_{\max}, k_{\max}, t_{\max}$ );  
1 repeat  
2    $k \leftarrow 1$ ;  
3   repeat  
4      $x' \leftarrow \text{Shake}(x, k)$ ;  
5      $x'' \leftarrow \text{VND}(x', k'_{\max})$ ;  
6     NeighborhoodChange( $x, x'', k$ );  
   until  $k = k_{\max}$ ;  
7    $t \leftarrow \text{CpuTime}()$   
until  $t > t_{\max}$ ;
```

Algorithm 8: Steps of the general VNS

Variações de VNS

- Variable neighborhood descent (VND)
- Reduced VNS (RVNS)
- Skewed VNS (SVNS)
- General VNS (GVNS)
- VN Decomposition Search (VNDS)
- Parallel VNS (PVNS)
- Primal Dual VNS (P-D VNS)
- Reactive VNS
- Backward-Forward VNS
- Best improvement VNS
- Exterior point VNS
- VN Simplex Search (VNSS)
- VN Branching

Hansen, P., Mladenović, N. and Moreno Pérez, JA., Variable neighbourhood search: Methods and applications, *Annals of Operations Research* 175 (1) : 367- 407, 2010.