

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

Otimização de Redes

Variable Neighborhood Search
(VNS)

Variable Neighborhood Search

- ❖ VNS é uma metaheurística proposta por N. *Mladenovic, P. Hansen (1997)*.
- ❖ Baseia-se na ideia de uma mudança sistemática de vizinhança:
 - ❖ na *fase de refinamento*, para encontrar um ótimo local; e
 - ❖ na *fase de perturbação*, para escapar de bacias de atração.

Estratégias Básicas

- ❖ VNS baseia-se em três fatos simples:
 - ❖ **Fato 1.** Um mínimo local relacionado a uma estrutura de vizinhança não é necessariamente mínimo local de outra estrutura;
 - ❖ **Fato 2.** Um mínimo global é um mínimo local relacionado a todas as possíveis estruturas de vizinhança;
 - ❖ **Fato 3.** Para muitos problemas, mínimos locais relacionados a uma ou várias estruturas de vizinhança (N_k) estão relativamente próximos um dos outros.

Estratégias Básicas

- ❖ Ao usar várias estruturas de vizinhança, os Fatos 1 a 3 podem ser aplicados de três formas diferentes:
 - ❖ determinística;
 - ❖ estocástica;
 - ❖ determinística e estocástica.

Variable Neighborhood Descent (VND)

Algoritmo: Variable neighborhood descent

Function $VND(x, k_{max})$

k $\leftarrow 1$

repeat

$x' \leftarrow \arg \min_{y \in N_k(x)} f(y)$ % Find the best neighbor in $N_k(x)$

$x, k \leftarrow NeighborhoodChange(x, x', k)$ % Change neighborhood

until $k = k_{max}$

return x

- ❖ No VND, a mudança de vizinhança é realizada de maneira determinística.
- ❖ Frequentemente, $k_{max} \leq 2$ em heurísticas de busca local.

Variable Neighborhood Descent (VND)

Algoritmo: Neighborhood change

Function NeighborhoodChange (x, x', k)

if $f(x') < f(x)$ then

$x \leftarrow x'$ % Make a move

$k \leftarrow 1$ % Initial neighborhood

else

$k \leftarrow k + 1$ % Next neighborhood

end

return x, k

Reduced VNS (RVNS)

Algoritmo: Reduced VNS

Function RVNS (x, k_{max}, t_{max})

repeat

k \leftarrow 1

repeat

x' \leftarrow Shake (x, k)

x, k \leftarrow NeighborhoodChange (x, x', k)

until k = k_{max}

t \leftarrow CpuTime()

until t > t_{max}

return x

- ❖ No RVNS, soluções aleatórias são selecionadas a partir de $N_k(x)$.
- ❖ Uma técnica de refinamento não é aplicada.

Basic VNS (BVNS)

- ❖ O método BVNS combina mudanças de vizinhança determinística e estocástica.
- ❖ A determinística é representada por uma heurística de busca local.
- ❖ A estocástica é representada por uma seleção aleatória de uma solução da k -ésima vizinhança.

Basic VNS (BVNS)

Algoritmo: Best improvement (steepest descent) heuristic

Function Best Improvement (x)

repeat

$x' \leftarrow x$

$x \leftarrow \arg \min_{y \in N(x)} f(y)$

until $f(x) \geq f(x')$

return x

Basic VNS (BVNS)

Algoritmo: First improvement (first descent) heuristic

Function First Improvement (x)

repeat

$x' \leftarrow x; i \leftarrow 0;$

repeat

$i \leftarrow i + 1$

$x \leftarrow \arg \min \{f(x), f(x^i)\}, x^i \in N(x)$

until $(f(x) < f(x'))$ or $i = |N(x)|$

until $f(x) \geq f(x')$

return x

Basic VNS (BVNS)

Algoritmo: Basic VNS

Function $BVNS (x, k_{max}, t_{max})$

$t \leftarrow 0$

while $t < t_{max}$ *do*

$k \leftarrow 1$

repeat

$x' \leftarrow Shake (x, k)$ % shaking

$x'' \leftarrow BestImprovement (x')$ % local search

$x, k \leftarrow NeighborhoodChange (x, x'', k)$

until $k = k_{max}$

$t \leftarrow CpuTime()$

end

return x

General VNS (GVNS)

- ❖ O método GVNS combina as técnicas VNS e VND.
- ❖ Esta estratégia está relacionada às aplicações de maior sucesso citadas na literatura sobre VNS.

General VNS (GVNS)

Algoritmo: General VNS

Function GVNS (x, l_{max}, k_{max}, t_{max})

repeat

k \leftarrow 1

repeat

x' \leftarrow Shake (x, k)

x'' \leftarrow VND (x', l_{max})

x, k \leftarrow NeighborhoodChange (x, x'', k)

until k = k_{max}

t \leftarrow CpuTime()

until t > t_{max}

return x

General VNS (GVNS)

Algoritmo: Variable neighborhood descent

Function $VND(x, k_{max})$

$k \leftarrow 1$

repeat

$x' \leftarrow \arg \min_{y \in N_k(x)} f(y)$ % Find the best neighbor in $N_k(x)$

$x, k \leftarrow NeighborhoodChange(x, x', k)$ % Change neighborhood

until $k = k_{max}$

return x

Reference

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