



UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA  
ELÉTRICA E INFORMÁTICA INDUSTRIAL  
DISCIPLINA - TÓPICOS ESPECIAIS EM COMPUTAÇÃO PARALELA

Capítulo 11 – Parallel Variable Neighborhood search  
José A. Moreno-Pérez, Pierre Hansen, Nenad Mladenovic

Aluno: Chidambaram Chidambaram  
Prof. Heitor Silvério Lopes

Curitiba, 01 de Setembro de 2009

### **Metaheurística - Pesquisa Paralela na Vizinhança Variável**

*(Variable Neighborhood Search – VNS)*

#### **Objetivo:**

- ◆ Pesquisar sistematicamente por melhores soluções mudando os locais de vizinhança de forma paralela.
- ◆ Aplicar o paralelismo para reduzir o tempo computacional e/ou aumentar a exploração no espaço de busca.

#### **Métodos de VNS:**

- ◆ Variable Neighborhood Descent (VND)
- ◆ Reduced Variable Neighborhood Search (RVNS)
- ◆ Basic Variable Neighborhood Search (BVNS)

#### **Algoritmos Paralelos VNS:**

- ◆ Parallel Local Search Algorithm (PLS)
- ◆ Replicated Parallel Variable Neighborhood Search Algorithm( RPVNS)
- ◆ Cooperative Neighborhood VNS (CNVNS)

### **Definição:**

Considerando como um problema de otimização combinatorial, a função de minimização pode ser formulada da seguinte forma:

$$\min\{f(x) : x \in X\}$$

Onde  $X$  é espaço de soluções,  $x$  é uma possível solução e  $f$ , função objetiva do problema.

Mínimo Global (Solução Ótima)  $x^*$ :

$$x^* \in X \quad f(x^*) \leq f(x), \forall x \in X$$

Mínimo Local  $x'$  em relação a estrutura da vizinhança  $N$ :

$$x' \in X \quad f(x') \leq f(x), \forall x \in N(x)$$

### **Variable Neighborhood Search (VNS):**

É uma metaheurística que explora sistematicamente a idéia da mudança na vizinhança com o algoritmo de busca local tanto no momento da descida para um mínimo local quanto na saída de vales os quais contém os mínimos locais.

O VNS é baseado em três observações básicas:

- I - O mínimo local em uma estrutura da vizinhança não necessariamente é o mesmo de outras vizinhanças;
- II - O mínimo global é o mínimo local em relação a todas as possíveis estruturas da vizinhança.
- III - Para muitos problemas, o mínimo local em relação a uma ou a várias vizinhanças é relativamente próximo um ou outro.

## Variable Neighborhood Descent (VND)

- É uma versão determinística de VNS;
- É baseado na observação I:

“ O ótimo local para um tipo de mudança  $x \leftarrow x_1$  da vizinhança  $N_1(x)$  não necessariamente é o mesmo para um outro tipo de mudança como  $x \leftarrow x_2$  na vizinhança  $N_2(x)$ . ”

### Pseudocódigo VND:

1. **Inicialização:** **Selecione um conjunto de estruturas de vizinhança**  $N_l, l = 1, 2, \dots, l_{\max}$  **Encontre uma solução inicial**  $x$ ;
2. **Repita a seguinte seqüência até não encontrar mais melhorias:**
  - i)  $l \leftarrow 1$ ;
  - ii) **Repita os seguintes passos até que**  $l = l_{\max}$ ;
  - iii) **Exploração da vizinhança:**  
**Encontre a melhor solução-vizinho**  $x'$  de  $x$  ( $x' \in N_l(x)$ );
  - iv) **Mudança:**  
**Se a solução**  $x'$  **é melhor que**  $x$ ,  
**então**  $x \leftarrow x'$  **e**  $l \leftarrow l + 1$ ,  
**se não,**  $l \leftarrow l + 1$ ;

## Reduced Variable Neighborhood Search (RVNS)

- É um método estocástico;
- É baseado na observação III:

“ Soluções pré-selecionadas das vizinhanças ( $N_1(x), N_2(x), \dots, N_{k_{\max}}(x)$ ); são escolhidas de forma aleatória.”

### Pseudocódigo RVNS:

1. **Inicialização:** **Selecione um conjunto de estruturas de vizinhança**  $N_k, k = 1, 2, \dots, k_{\max}$ .  
**Encontre uma solução inicial**  $x$ ; **Escolhe uma condição de parada**;
1. **Repita a seguinte seqüência até que a condição de parada é alcançada:**
  - i)  $k \leftarrow 1$ ;
  - ii) **Repita os seguintes passos até que**  $k = k_{\max}$ ;
    - a) *shake*. **Escolhe aleatoriamente uma solução**  $x'$  de  $N_k(x)$ ;
    - b) **Se a solução**  $x'$  **é melhor que**  $x$ ,  
**então**  $x \leftarrow x'$  **e continue a busca com**  $N_1(k \leftarrow 1)$ ;  
**se não,**  $k \leftarrow k + 1$ ;

### Basic Variable Neighborhood Search (BVNS)

- É um método VND + RVNS;
- É chamado também como "ILS" (*Iterated Local Search*);
- É uma combinação de busca local com mudanças sistemática de vizinhanças ao redor de um ótimo local encontrado.

#### Pseudocódigo BVNS:

1. **Inicialização:** **Selecione um conjunto de estruturas de vizinhança  $N_k$ ,  $k = 1, 2, \dots, k_{\max}$ .**  
**Encontre uma solução inicial  $x$ ; Escolhe uma condição de parada;**
1. **Repita a seguinte seqüência até que a condição de parada é alcançada:**
  - 1)  $k \leftarrow 1$ ;
  - 2) **Repita as seguintes etapas até que  $k = k_{\max}$ :**
    - a) **shake.** **Gera aleatoriamente um ponto  $x'$  da vizinhança  $k$  de  $x$  ( $x' \in N_k(x)$ );**
    - b) **Busca Local:**  
**Aplique um método de busca local com  $x'$  como solução inicial;**  
**Encontre a solução ótimo local  $x''$ ;**
    - c) **Mudança:**  
**Se a solução ótima local  $x''$  é melhor que a solução inicial  $x$ ,**  
**então move para  $x \leftarrow x''$  e continue a busca com  $N_1(k \leftarrow 1)$ ;**  
**se não,  $k \leftarrow k+1$ ;**

### Propriedades de Metaheurística<sup>1</sup>:

- ✓ **Simplicidade** – baseada em princípios simples e claras que devem ter aplicação ampla;
- ✓ **Coerência** – etapas de heurística devem ser naturais;
- ✓ **Eficiência** – gerar soluções ótimas ou próxima a ótimas para todas ou pelo menos para instâncias reais;
- ✓ **Efetividade** – gerar soluções ótimas ou próxima a ótimas em tempo moderado de CPU;
- ✓ **Robustez** – fornecer boas soluções a uma variedade de instâncias;
- ✓ **Amigável** – poucos ou quase sem parâmetros;
- ✓ **Inovação** – conduzir para novos tipos de aplicações;
- ✓ **Generalidade** – boas soluções para uma grande variedade de problemas;
- ✓ **Interatividade** – inclusão de conhecimento do usuário;
- ✓ **Multiplicidade** – apresentar várias soluções ótimas.

<sup>1</sup> Editorial – VNS, Hansen, P., Mladenovic, N., e Pérez J.A.M.

### Heurística Paralela de VNS:

- Reduz o tempo computacional particionando programas seqüenciais;
- Aumenta exploração no espaço de busca aplicando *threads* independentes de busca;
- Codificação feita em C usando OpenMP;
- Paralelização de VNS consiste em:
  - Busca local – *Synchronous Parallel VNS (SPVNS)*;
  - Replicação do VNS inteira em cada processador – *Replicated Parallel VNS (RPVNS)*;

### Synchronous Parallel VNS (SPVNS)

#### Algoritmo SPVNS:

```
Initialize (Best_solution);
K=0;
While(k<k_max){
  K++;
  cur_sol = shake(best_sol, k);
  par_local_search(cur_sol);
  if improved(cur_sol, best_sol) {
    best_sol = cur_sol;
    k = 0;
  }
}
```

#### Algoritmo PLS:

```
Void par_local_search (sol cur_sol)
{
  init_sol = cur_sol
  while (improved()) {
    load = (n-p) div (num_proc);
    parallel (pr=0; pr<num_proc; pr++) {
      tmp_sol(pr) = init_sol;
      low = pr * load;
      high = low + load;
      for(i=low; i < high; i++)
        for(j=0; j<p; j++) {
          exchange (init_sol, new_sol, i,j)
          if improve(new_sol, tmp_sol(pr))
            tmp_sol(pr) = new_sol;
        }
      critical
      if improve(tmp_sol(pr), cur_sol)
        cur_sol = tmp_sol(pr);
    } // Parallel
  } // par_local_search
}
```

### Replicated Parallel VNS:

- Tenta explorar uma região ampla de espaço de soluções inicializando várias soluções na mesma vizinhança ou vizinhanças diferentes;
- Cada busca local é substituída por um VNS;

#### Algoritmo RPVNS:

```
Initialize (Joint_best_solution);
Parallel pr = 0, num_proc-1 {
  initialize (best_sol(pr));
  k(pr) = 0;
  while (k(pr) < k_max) {
    k(pr)++;
    cur_sol(pr) = shake(best_sol(pr), k(pr));
    local_search(cur_sol(pr));
    if improved(cur_sol(pr), best_sol(pr)) {
      best_sol(pr) = cur_sol(pr);
      k(pr) = 0;
    }
  } // while
critical
  if improve(best_sol(pr), joint_best_sol)
    joint_best_sol = best_sol(pr);
} // parallel
```

### Cooperative Neighborhood VNS (CNVNS):

- Abordagem Master-slave usando o mecanismo de cooperação assíncrona;
- Aplica o método de pesquisa múltipla cooperativa baseado em mecanismo de memória central (master);

#### Processo Master:

- Executa *Replicated Parallel Variable Neighborhood Search* (RVNS);
- Envia as soluções iniciais para processos individuais de VNS;
- Atualiza a solução ótima global através de comunicações recebidos dos processos individuais de VNS;
- Atende as requisições dos processos VNS enviando a ótima global;
- Verifica a condição de parada.

#### Processo VNS:

- Recebe a solução inicial, seleciona aleatoriamente a vizinhança e explora através do *shaking e busca local*;
- Se a solução é melhorada, a busca ocorre da primeira vizinhança: *shake and busca local*;
- Se a solução não pode ser melhorada, então o processo:
  - \* comunica a solução encontrada ao *master*;
  - \* solicita a solução ótima global do *master*;
  - \* continua a busca na vizinhança do ótimo global.

### **Aplicação:**

➤ **A new hybrid metaheuristics for the vehicle routing problem with simultaneous pick-up and delivery** by Lijun Meng and Xiaochai Guo;

➤ **Variable Neighborhood Tabu Search for Capacitor Placement in Distribution Systems** by Hiroyuki Mori and Shingo Tsunokawa

➤ **A neighborhood search heuristic for pre-marshalling export containers** by Yusin Lee and Shih-Liang Chao

➤ **General variable neighborhood search for the continuous optimization** by Nenad Mladenovic, Milan Draz'ic, Vera Kovac'evic-Vujc'ic and Mirjana C'angalovic'

➤ **Variable Neighborhood Search as Genetic Algorithm Operator for DNA Fragment Assembling Problem** by Gabriela Minetti and Gabriel Luque - Enrique

Alba  
➤ **A variable neighborhood-based heuristic for the heterogeneous fleet vehicle routing problem** by Arif Imran, Said Salhi and Niaz A. Wassan

➤ **A Novel Variable Neighborhood Particle Swarm Optimization for Multi-objective Flexible Job-shop Scheduling Problems** by Hongbo Liu Ajith Abraham and Crina Grosan

➤ **Scheduling Jobs on Parallel Machines with Sequence-Dependent Setup Times, Precedence Constraints, and Ready Times Using Variable Neighborhood Search** by Rene Drießel, Lars Moench

### **Referências:**

[1] E. Alba. Parallel Metaheuristics – A New Class of Algorithms, A John Wiley & Sons, Inc., Publications, pp. 247-264, 2005.

[2] P. Hansen, N. Mladenović, J. A. M. Pérez. Variable neighborhood search, European Journal of Operational Research, Vol.191, No. 3, pp. 593-595, Dec2008.

[3] P. Hansen, N. Mladenović, J. A. M. Pérez. Variable neighborhood search: Principles and applications, European Journal of Operational Research, Vol. 130, No. 3, pp 449-467, May2001.