

## MÉTODOS HEURÍSTICOS USANDO BUSCA LOCAL ALEATÓRIA EM VIZINHANÇA VARIÁVEL PARA O PROBLEMA DO CAIXEIRO VIAJANTE COM GRUPAMENTOS

### HEURISTIC METHODS USING VARIABLE NEIGHBORHOOD RANDOM LOCAL SEARCH FOR THE CLUSTERED TRAVELING SALESMAN PROBLEM

Mário Mestria\* E-mail: [mmestria@ifes.edu.br](mailto:mmestria@ifes.edu.br)

\*Instituto Federal do Espírito Santo (IFES), Jucutuquara, Vitória, ES

**Resumo:** Nesse artigo, são propostos novos métodos heurísticos para resolver o Problema do Caixeiro Viajante com Grupamentos (PCVG). O PCVG é uma generalização do Problema do Caixeiro Viajante (PCV), onde os vértices são particionados em grupos disjuntos e o objetivo é encontrar um ciclo hamiltoniano de custo mínimo tal que os vértices de cada grupo são visitados de forma contígua. Nós desenvolvemos dois Métodos de Descida Aleatória em Vizinhaça Variável com *Iterated Local Search* para resolver o PCVG. Os métodos heurísticos propostos foram testados em tipos de instâncias com dados em níveis diferentes de granularidade para o número de vértices e grupos. Resultados computacionais mostraram que os métodos heurísticos se sobrepõem aos métodos recentes existentes na literatura e são competitivos com um método exato usando o software CPLEX Paralelo.

**Palavras-chave:** Pesquisa Operacional. Otimização Combinatória. Métodos Heurísticos. Método de Descida Aleatória em Vizinhaça Variável. *Iterated Local Search*.

**Abstract:** In this paper, we propose new heuristic methods for solver the Clustered Traveling Salesman Problem (CTSP). The CTSP is a generalization of the Traveling Salesman Problem (TSP) in which the set of vertices is partitioned into disjoint clusters and objective is to find a minimum cost Hamiltonian cycle such that the vertices of each cluster are visited contiguously. We develop two Variable Neighborhood Random Descent with *Iterated Local* for solver the CTSP. The heuristic methods proposed were tested in types of instances with data at different level of granularity for the number of vertices and clusters. The computational results showed that the heuristic methods outperform recent existing methods in the literature and they are competitive with an exact algorithm using the Parallel CPLEX software.

**Keywords:** Operations Research. Combinatorial Optimization. Heuristic Methods. Variable Neighborhood Randomized Descent. *Iterated Local Search*.

## 1 INTRODUÇÃO

O Problema do Caixeiro Viajante com Grupamentos (PCVG) é utilizado para modelar diversos problemas de otimização encontrados na área de Pesquisa Operacional. Podemos destacar: planejamento da produção (LOKIN, 1979),

despacho de veículos de emergência numa empresa de eletricidade, (WEINTRAUB; ABOUD; FERNANDEZ; LAPORTE; RAMIREZ, 1999), sistemas de manufaturas (LAPORTE; PALEKAR, 2002), desfragmentação de discos rígidos e aplicações comerciais envolvendo supermercados, lojas e fornecedores de mercadorias (GHAZIRI; OSMAN, 2003).

O PCVG possui a característica de ser pouco estudado na literatura e foi proposto por Chisman (1975) para um problema de roteamento automático em sistemas de armazenagem. O objetivo deste problema é encontrar ciclo hamiltoniano de custo mínimo que passe por todos os vértices do grafo e visita os vértices de cada grupo de forma contígua.

Como observado em Mestria (2013) ao solucionar o PCVG estaremos contribuindo para redução de custos na área de produção.

[...] em vários ambientes de produção existem diversas aplicações que podem ser modeladas através do PCVG. Neste sentido a solução para o PCVG contribuirá para reduzir os custos no transporte e produção, pois estaremos minimizando os custos envolvidos entre os caminhos dos vértices (MESTRIA, 2013, p. 1003).

O PCVG pertence à classe *NP*-Difícil restringindo o uso exclusivo de métodos exatos. Solução proposta neste artigo para solucioná-lo será através de métodos heurísticos.

Na literatura destacam-se alguns trabalhos para o PCVG. O de Jongens e Volgenant (1985) utilizando Relaxação Lagrangeana para obter limites inferiores. Foram gerados conjuntos de instâncias variando o número de vértices de 80 a 150 com diferentes tamanhos de grupos. A Busca Tabu de Laporte, Potvin e Quilleret (1996) no qual resolve o PCVG, mas a sequência de visita dos grupos é definida *a priori*. Por fim, o de Ding, Cheng e He (2007), onde foi desenvolvido um Algoritmo Genético, sem incorporar módulos de busca local, que usa o critério de visita dos vértices de cada grupo de forma aleatória sem estabelecer uma relação com as distâncias entre os vértices. A escolha da sequência de visita de cada grupo também é aleatória. Em Mestria, Ochi e Martins (2011) foram desenvolvidos dois algoritmos baseado no *Iterated Local Search* (ILS) (LOURENÇO; MARTIN; STÜTZLE, 2002) para o PCVG. O algoritmo que incorpora uma heurística construtiva que não aplica

uma penalização (P) nas arestas intergrupos foi capaz de obter soluções de boa qualidade. No trabalho de Mestria (2013) são propostos métodos heurísticos para o PCVG usando GRASP, Reconexão de Caminhos e *Variable Neighborhood Descent* (VND). Os resultados computacionais mostraram que o método heurístico híbrido usando VND sobressaiu em relação aos outros métodos.

O artigo de Anzanello e Fogliatto (2011) apresenta uma abordagem heurística cujo objetivo é minimizar o tempo demandado para término de tarefas caracterizadas pela necessidade de aprendizado dos trabalhadores em ambientes de Customização em Massa.

Em Nagano e Mesquita (2012) foram utilizados métodos heurísticos para solucionar o *problema flow shop permutacional* com tempos de preparação das máquinas separados dos tempos de processamento das tarefas, sendo dependentes da sequência de execução das tarefas. Este problema de produção é classificado como *NP*-difícil.

No trabalho de Ludwig, Anzanello e Vidor (2013) foram propostos duas heurísticas de sequenciamento com vistas à minimização do atraso total de entrega de tarefas em uma empresa de desenvolvimento de *softwares* personalizados. As heurísticas são inicialmente testadas em um banco de dados simulados e então aplicadas a dados reais.

Em Sobreiro e Nagano (2013) foi desenvolvida uma nova heurística para tratar do *problema de mix de produção*. A heurística proporcionou melhores soluções quando comparada com a principal heurística apresentada na literatura. As comparações foram realizadas através de simulações computacionais visando identificar o *mix de produção* que possibilitasse o maior ganho possível, considerando um bom tempo de processamento e as características do ambiente produtivo.

Como observado, em diversas áreas envolvem problemas de otimização nos quais são solucionados por métodos heurísticos. Soluções ótimas para estes tipos de problemas podem ser encontradas com tempos computacionais enormes. Isto prejudica nas tomadas de decisões que em oposição trabalha com curto prazo de tempo, e desta forma, métodos heurísticos são ferramentas úteis para realização deste propósito. Portanto, como afirma Mestria (2013), as heurísticas contribuem

para soluções de problemas de otimização buscando qualidade nas soluções geradas com tempo computacional baixo.

Para problemas de otimização combinatória, o ILS (LOURENÇO; MARTIN; STÜTZLE, 2002) foi utilizado com sucesso em diversos tipos de aplicações. Em Dong, Chen, Huang e Nowak (2013), o ILS foi usado para resolver o *problema flow shop de permutação* com minimização do tempo total de fluxo usando uma estratégia multipartida. Os autores Villegas, Prins, Prodhon, Medaglia e Velasco (2013) resolveram um *problema de roteamento* através de uma estratégia de duas fases que combina elementos de algoritmos de programação matemática e metaheurísticas de forma cooperativa, simples, mas eficaz. Na primeira fase, rotas de ótimos locais de um algoritmo híbrido (GRASP+ILS) são obtidas. Na segunda fase, estas rotas tornam-se colunas em um problema de particionamento de conjuntos. No artigo de Juan, Lourenço, Mateo, Luo e Castella (2014) são discutidos várias questões relacionadas com o uso de ILS para o desenvolvimento de algoritmos eficientes para o *problema flow shop de permutação*. ILS possui princípios simples, é fácil de implementar e possibilita o armazenamento de informações históricas da estrutura da solução ao longo do algoritmo (MESTRIA; OCHI; MARTINS, 2011).

O VND (HANSEN E MLADENOVIC, 2003) foi utilizado em um dos níveis do algoritmo proposto de Tarantilis, Stavropoulou e Repoussis (2013) para resolver um *problema de roteamento de veículos*. Neste VND, a busca foi guiada em torno das soluções com distância mínimas percorridas utilizando vizinhanças do tipo intrarotas e interrotas. No trabalho de Jarboui, Derbel, Hanafi, Mladenović (2013), o VND também foi usado como busca local para resolver um *problema de roteamento e localização com vários depósitos capacitados e um veículo não capacitado por depósito*. O VND proposto incorpora cinco estruturas de vizinhança para explorar diferentes possibilidades de localização dos depósitos e melhorar as atribuições de clientes a cada depósito.

Como é destacado na literatura:

A estrutura do VND não é complexa, de fácil implementação e suas estruturas de vizinhanças (LAMB, 2012) podem ser alteradas aumentando ou diminuindo sua complexidade para se atingir um compromisso entre

qualidade da solução e tempo computacional empregado (MESTRIA, 2013, p. 1005).

Neste sentido, o objetivo deste trabalho é utilizar o método VND como busca local para as heurísticas ao PCVG, mas a implementação deste método utilizará uma busca aleatória em vizinhança variável, denominado de Método de Descida Aleatória em Vizinhança Variável (*Variable Neighborhood Random Descent - VNRD*).

Dois novos métodos heurísticos baseados no ILS e no VNRD são propostos neste trabalho. O primeiro (ILS-VNRD-P) utiliza ILS, o VNRD e um algoritmo construtivo que penaliza (P) as arestas intergrupos (IMPP). No segundo (ILS-VNRD-nP) troca-se o IMPP por IMPnP, que não penaliza (nP) as arestas intergrupos. Os algoritmos construtivos são baseados na Inserção mais Próxima (IMP).

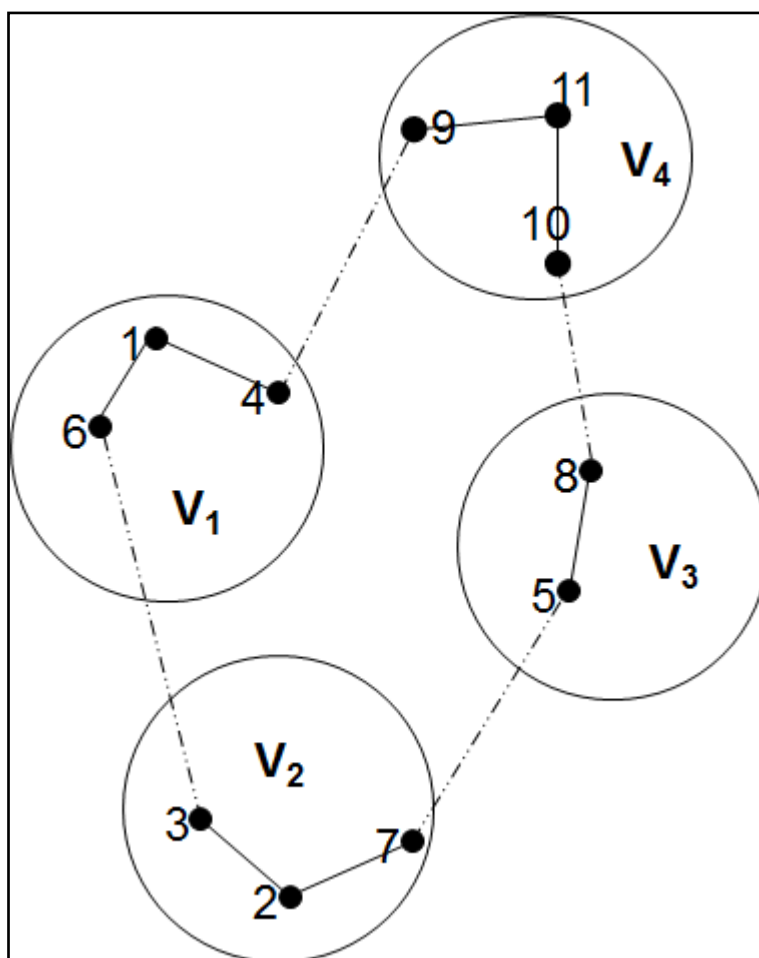
A estrutura do artigo é formada como segue: a segunda seção mostra uma definição formal para o PCVG, a terceira seção descreve os métodos heurísticos para solucionar o PCVG, a quarta mostra os testes computacionais, comparações e discussões com os métodos da literatura e na última seção são realizadas as considerações finais e conclusões.

## **2 O PROBLEMA DO CAIXEIRO VIAJANTE COM GRUPAMENTOS**

Em Mestria (2013) mostra uma definição para o PCVG utilizando grafos. Seja  $G = (V, E)$  um grafo completo com conjunto de vértices  $V = \{v_1, v_2, \dots, v_n\}$  e um conjunto de arestas  $E = \{(v_i, v_j) : v_i, v_j \in V, i \neq j\}$ . O conjunto de vértices  $V$  é particionado dentro de grupos disjuntos  $V_1, V_2, \dots, V_m$ .

Assumindo que um custo não negativo  $c_{ij}$  é associado com a aresta  $E = (v_i, v_j)$ , o PCVG consiste em determinar um ciclo Hamiltoniano de custo mínimo em  $G$ , tal que os vértices de cada grupo são visitados contiguamente.

**Figura 1** – Uma solução viável para um exemplo de uma instância do PCVG



Fonte: Autor

Uma solução viável para um exemplo de uma instância do PCVG é mostrado na Figura 1, onde a solução consiste de um ciclo Hamiltoniano tal que os vértices de cada grupo são visitados na forma contígua.

O grupo 1 é representado por  $V_1$ , o grupo 2 por  $V_2$ , o grupo 3 por  $V_3$  e Grupo 4 por  $V_4$ , contendo seus respectivos vértices:  $V_1=\{1,4,6\}$ ,  $V_2=\{2,3,7\}$ ,  $V_3=\{5,8\}$  e  $V_4=\{9,10,11\}$ . As linhas tracejadas (3, 6), (4, 9), (10, 8) e (5, 7) mostram as conexões das arestas intergrupos e as linhas cheias, as arestas intragrupo, que são ligações dentro do grupo.

### 3 MÉTODOS HEURÍSTICOS PARA SOLUCIONAR O PCVG

Nesta seção mostraremos os módulos de construção, o de busca local implementada, os de perturbação e o critério de aceitação para compor os métodos heurísticos propostos em solucionar o PCVG.

#### 3.1 Métodos Construtivos e o Método de Descida Aleatória em Vizinhança Variável

Para os módulos de construção temos: primeiro, o algoritmo construtivo que penaliza as arestas intergrupos (IMPP) e o segundo (IMPnP), que não realiza a penalização nas arestas intergrupos (MESTRIA; OCHI; MARTINS, 2011). Ambos os algoritmos são baseados na heurística de Inserção mais Próxima (IMP). A IMP necessita de três níveis de decisão sendo (i) a construção de um ciclo inicial, (ii) a escolha do próximo vértice a ser inserido na solução e (iii) a posição de inserção desse novo vértice no ciclo atual. O algoritmo construtivo que penaliza as arestas intergrupos (IMPP) é mostrado na Figura 2. No início do algoritmo IMPP são escolhidos três vértices do conjunto  $V=\{v_1, v_2, \dots, v_n\}$ , sendo o primeiro aleatório e os dois outros usando o conceito de vizinho mais próximo, passo 2 do algoritmo.

**Figura 2** - Pseudocódigo do algoritmo construtivo que penaliza as arestas intergrupos (IMPP)

1. Seja o conjunto de vértices  $V=\{v_1, v_2, \dots, v_n\}$ ;
2. Selecionar três vértices iniciais  $v_i, v_{i+1}$  e  $v_{i+2}$ , sendo o primeiro aleatório e os outros dois, serão os vizinhos mais próximos;
3.  $S \leftarrow \{v_i, v_{i+1}, v_{i+2}\}$ ;
4. Inicializar o conjunto de candidatos  $C=V \setminus \{v_i, v_{i+1}, v_{i+2}\}$ ;
5. *penalizar\_arestas*( );
6. **enquanto**  $C \neq \emptyset$  **faça**
7. Encontrar os  $k$  vértices mais próximos contidos em  $C$  em relação aos vértices pertencentes ao ciclo já formado (solução  $S$ ), calculando o custo incremental  $c(v_k)$  para cada  $v_k \in C$ ;
8.  $c_{\min} \leftarrow \min\{c(v_k) / v_k \in C\}$ ;
9.  $c_{\max} \leftarrow \max\{c(v_k) / v_k \in C\}$ ;
10.  $LRC \leftarrow \{v_k \in C / c(v_k) \leq c_{\min} + \alpha(c_{\max} - c_{\min})\}$ ;
11. Selecionar um elemento  $v_k$  da LRC de forma aleatória;
12.  $S \leftarrow S \cup \{v_k\}$  - conectar  $v_k$  aos vértices  $v_i$  e  $v_{i+1}$ , cujo custo (*price*)  
 $p(v_k)=\{p_{i,k} + p_{k,i+1} - p_{i,i+1}\}$  seja mínimo e atualizar o ciclo;
13. Atualizar o conjunto candidatos  $C=C \setminus \{v_k\}$ ;
14. **fim enquanto**

Fonte: Autor

A solução parcial ( $S$ ) e o conjunto de vértices candidatos ( $C$ ) são inicializados, passos 3 e 4. No passo 5 é realizada a penalização nas arestas intergrupos através do procedimento *penalizar\_arestas*( ). Em seguida, os vértices pertencentes ao conjunto de candidatos são avaliados pelos custos incrementais (passo 7), computando-se as distâncias destes vértices contidos no conjunto em relação aos vértices da solução parcial. O valor de  $c_{\min}$ , menor custo incremental e o valor de  $c_{\max}$ , maior custo incremental são armazenados para calcular a Lista Restrita de Candidatos (LRC), passo 10. A seguir, seleciona-se aleatoriamente um vértice da LRC, passo 11, que é conectado aos vértices do ciclo da solução parcial, onde o custo de inserção seja mínimo, passo 12. Atualiza-se o conjunto de candidatos, passo 13, e se repetem os passos 6 até 14 até que o algoritmo termina quando o conjunto de candidatos estiver vazio.

O algoritmo IMPnP não penaliza as arestas intergrupos, passo 5 do algoritmo IMPP, e a formação do conjunto de candidatos é diferente da utilizada no algoritmo IMPP, passo 7. Para o IMPnP, primeiro escolhem-se os vértices que irão compor o conjunto de candidatos sendo aqueles que pertencem ao mesmo grupo corrente dos vértices contidos na solução parcial. Se existem vértices pertencentes ao grupo corrente dos vértices do ciclo atual (solução parcial), consideram-se somente estes vértices para compor o conjunto de candidatos. Quando todos os vértices do grupo corrente já estão na solução parcial, todos os outros vértices restantes que ainda não foram selecionados são considerados candidatos.

Para o módulo de busca local é utilizado o VNRD cujo pseudocódigo é mostrado na Figura 3. É definido um conjunto de vizinhanças  $N_k$ ,  $k=1, 2, 3$  e  $4$ , e uma solução inicial  $S$  com seu valor da função objetivo  $f(S)$ . Estas quatro estruturas de vizinhanças estão definidas em Mestria, Ochi e Martins, (2011). O VNRD encontra a melhor vizinhança  $S'$  de  $S$ , ( $S' \in N_k(S)$ ) para cada estrutura, passos 5, 7, 9 e 11 da Figura 3, e calcula o valor da função objetivo desta vizinhança gerada  $f(S')$ . Se o valor  $f(S')$  for menor que  $f(S)$  atualiza-se  $S$  com  $S'$ , passos 13 e 14. Caso contrário escolhe-se uma nova estrutura de vizinhança na forma aleatória (diferente das anteriores) e incrementa o número de vizinhanças geradas, passos 16 e 17. Para reduzir o esforço computacional, as estruturas de vizinhanças  $N_1$  e  $N_3$  utilizam



*first improvement* e as estruturas  $N_2$  e  $N_4$ , *best improvement*, (MESTRIA; OCHI; MARTINS, 2011).

**Figura 3** - Pseudocódigo do *Variable Neighborhood Random Descent*

```
1.  $k \leftarrow$  sorteia um valor entre 1 e 4;  
2.  $contk \leftarrow 1$ ;  
3. enquanto  $contk \leq 4$  faça  
4. se  $k=1$  então  
5.    $S' \leftarrow 1\text{-Shift}(S)$ ;  
6. senão se  $k=2$  então  
7.    $S' \leftarrow 1\text{-DropCheap}(S)$   
8.   senão se  $k=3$  então  
9.      $S' \leftarrow 2\text{-Swap}(S)$   
10.    senão  
11.       $S' \leftarrow 2\text{-Optimal}(S)$   
12. fim se  
13. se  $f(S') < f(S)$  então  
14.    $S \leftarrow S'$ ;  
15. senão  
16.    $k \leftarrow$  sorteia um valor ainda não escolhido entre 1 e 4;  
17.    $contk \leftarrow contk + 1$ ;  
18. fim se  
19. fim enquanto  
20. retornar  $S$ ;
```

Fonte: Autor

### 3.2 Métodos Heurísticos utilizando o ILS para o PCVG

A seguir, são mostrados na Figura 4 os módulos do algoritmo ILS composto por: solução inicial, busca local, módulos de perturbação e de critério de aceitação (LOURENÇO; MARTIN; STÜTZLE, 2002). Seja a solução corrente  $S$  no qual é aplicada uma perturbação que permite um estado intermediário  $S'$ . Sobre  $S'$ , obtém-se  $S''$  através de uma busca local. Se a solução  $S''$  for considerada pelo critério de aceitação, esta se tornará a próxima solução a sofrer a perturbação, caso contrário é retornada a solução  $S$ . O algoritmo do ILS termina até que um critério de parada seja satisfeito.

**Figura 4** - Módulos do *Iterated Local Search*

```
1:  $S^0 \leftarrow \text{SolucaoInicial}$ ;  
2:  $S \leftarrow \text{BuscaLocal}(S^0)$ ;  
3: repita  
4:    $S' \leftarrow \text{Perturbacao}(S, \text{história})$ ;  
5:    $S'' \leftarrow \text{BuscaLocal}(S')$ ;  
6:    $S \leftarrow \text{CritérioAceitacao}(S, S'', \text{história})$ ;  
7: até critério de parada não satisfeito  
8: retornar  $S$ ;
```

**Fo**  
**nte**  
:  
Ada  
pta  
do  
de

Lourenço, Martin e Stützle (2002)

O primeiro método heurístico (ILS-VNRD-P) utiliza ILS, VNRD e o algoritmo construtivo IMPP. O segundo (ILS-VNRD-nP) utiliza também os módulos do ILS e do VNRD e substitui o algoritmo IMPP por IMPnP. Para ambos os métodos heurísticos propostos, a etapa de perturbação (*Pert*) consiste na realização do movimento *double-bridge* aplicado às arestas de um grupo escolhido aleatório. A sequência dos grupos não é alterada, somente a sequência dos vértices do grupo escolhido é modificada. Uma segunda perturbação (*Pertdg*) foi desenvolvida na qual se realiza uma troca aleatória no percurso de dois grupos sem alterar os vértices internos dos grupos.

O critério de aceitação (*CritAceit*), utilizado na etapa seguinte da busca local, considera três soluções: duas soluções ótimas locais,  $S$  e  $S''$ , descritas anteriormente e a solução  $S^*$ , a melhor solução até a etapa corrente do algoritmo. Para este critério de aceitação, a solução que tiver o menor custo entre as três soluções ( $S$ ,  $S''$ ,  $S^*$ ) será escolhida como solução para a etapa seguinte da perturbação (MESTRIA, OCHI; MARTINS, 2011).

O pseudocódigo do método heurístico ILS-VNRD-P é ilustrado na Figura 5. No passo 2, a distribuição de probabilidade dos alfas *inialfas*( ) é obtida e a melhor solução é inicializada no passo 4. Para cada iteração externa do ILS-VNRD-P a solução é construída no passo 6 pelo procedimento IMPP e, no passo 7, a busca local é executada pelo VNRD.

Para cada iteração interna do ILS-VND-P, a perturbação é aplicada no passo 9 e a busca local é executada novamente pelo VNRD no passo 10. No passo 11 aplica-se o critério de aceitação e a melhor solução é atualizada no passo 13. No

passo 17, a distribuição de probabilidade dos alfas *atualalfas*( ) é atualizada quando o parâmetro *iterprob* atingir o valor máximo igual a *maxprob*. Finalmente, no passo 22 a melhor solução é retornada.

**Figura 5** - Pseudocódigo para o método heurístico ILS-VNRD-P

```

1: iterprob ← 1;
2: inialfas( );
3:  $S^0 \leftarrow \text{IMPP}$ ;
4:  $S^* \leftarrow S^0$ ;
5: para i=1 até maxext faça
6:    $S^0 \leftarrow \text{IMPP}$ ;
7:    $S \leftarrow \text{VNRD}(S^0)$ ;
8:   para i=1 até maxint faça
9:      $S' \leftarrow \text{Pert}(S)$ ;
10:     $S'' \leftarrow \text{VNRD}(S')$ ;
11:     $S \leftarrow \text{CritAceit}(S, S'', S^*)$ ;
12:    se S for melhor que  $S^*$  então
13:       $S^* \leftarrow S$ ;
14:    fim se
15:  fim para
16:  se iterprob = maxprob então
17:    atualalfas( );
18:    iterprob ← 0;
19:  fim se
20:  iterprob ← iterprob + 1;
21: fim para
22: retornar  $S^*$ ;

```

**Fonte:** Adaptado de Mestria, Ochi e Martins (2011)

O pseudocódigo do método heurístico ILS-VNRD-nP é semelhante ao da Figura 5, substituindo o algoritmo construtivo IMPP por IMPnP nos passos 3 e 6. O número de iterações que o método heurístico reinicia a construção de soluções novas, depois de um número máximo de iterações internas (*maxint*), é considerado para o método como as iterações externas (*maxext*).

#### 4 TESTES COMPUTACIONAIS

Para avaliar e validar os novos métodos heurísticos propostos foi utilizado instâncias apresentadas em (MESTRIA; OCHI; MARTINS, 2011). São seis tipos diferentes de instâncias com granularidade variada e com os vértices agrupados em diversas configurações.

Por exemplo, as instâncias do tipo 1 são adaptação de instâncias do PCV agrupadas usando o algoritmo de clusterização *k-means*, enquanto que a do tipo 6, a área plana retangular que compõe a instância do PCV é dividida em diversos quadriláteros e cada quadrilátero corresponde a um grupo. Estas instâncias são Euclidianas e estão disponíveis em <<http://labic.ic.uff.br/Instance/index.php>>.

Os novos métodos heurísticos foram comparados com as soluções ótimas ou os limites inferiores obtidos pelo método exato usando a formulação matemática de Chisman (1975) e o software CPLEX versão 11.2 Paralelo (CPLEX, 2009). O CPLEX foi executado num computador Intel Core 2 Quad, 2.83 GHz com 8 GB de RAM com 4 núcleos.

O sistema operacional utilizado foi Linux Ubuntu versão 4.3.2-1. Os algoritmos foram codificados em linguagem de programação C e executados no mesmo computador descrito anteriormente, mas utilizando somente um núcleo. Os algoritmos foram executados dez vezes para cada instância por incorporar elementos aleatórios, sendo seus parâmetros descritos a seguir.

O valor de penalização nos custos das arestas  $M$  foi igual a  $10 * \max\{c_{ij}\}$ , onde  $\max\{c_{ij}\}$  significa o maior custo entre todos os custos das arestas. Os valores de alfa ( $\alpha$ ) no intervalo de  $[0,1]$ , usa a estratégia reativa (FESTA; RESENDE, 2011).

O número máximo de iterações internas ( $maxint$ ) foi igual a 35 e  $maxext$  com 40. O valor  $maxprob$ , número de iterações para atualizar as probabilidades de alfa, ficou estabelecido em 10. Todos estes parâmetros foram escolhidos após testes preliminares utilizando as instâncias mostradas a seguir nas seções 4.1 e 4.2.

#### **4.1 Resultados e Comparações com um Método Exato**

A comparação entre os métodos heurísticos foi efetuada com método exato CPLEX sem limitação de tempo, mostrada na Tabela 1, onde o CPLEX encontrou valores ótimos para todas as instâncias.

**Tabela 1** - Valores do CPLEX e dos métodos heurísticos para instâncias do tipo 1 de pequeno porte

<b>Instâncias</b>	<b>CPLEX</b>	<b><math>t_{\text{CPLEX}}</math> (s)</b>	<b>ILS-VNRD-P (%)</b>	<b><math>t_{\text{ILS-VNRD-P}}</math> (s)</b>	<b>ILS-VNRD-nP (%)</b>	<b><math>t_{\text{ILS-VNRD-nP}}</math> (s)</b>
5-eil51	437	12,31	<b>0,00</b>	4,00	<b>0,00</b>	4,00
10-eil51	440	74,38	<b>0,00</b>	3,80	<b>0,00</b>	4,10
15-eil51	437	2,04	<b>0,00</b>	3,60	<b>0,00</b>	4,10
5-berlin52	7991	201,80	<b>0,00</b>	3,50	<b>0,00</b>	3,30
10-berlin52	7896	89,17	<b>0,00</b>	2,50	0,52	3,70
15-berlin52	8049	75,93	<b>0,00</b>	2,40	<b>0,00</b>	3,00
5-st70	695	13790,11	<b>0,00</b>	8,10	<b>0,00</b>	7,30
10-st70	691	4581,00	<b>0,00</b>	5,90	<b>0,00</b>	6,10
15-st70	692	883,50	<b>0,00</b>	5,10	<b>0,00</b>	5,60
5-eil76	559	83,70	0,36	8,30	0,18	8,20
10-eil76	561	254,30	<b>0,00</b>	6,50	<b>0,00</b>	7,40
15-eil76	565	49,66	0,18	6,00	0,35	7,40
5-pr76	108590	99,29	<b>0,00</b>	9,70	<b>0,00</b>	9,00
10-pr76	109538	238,13	<b>0,00</b>	6,70	<b>0,00</b>	7,90
15-pr76	110678	261,94	<b>0,00</b>	7,10	<b>0,00</b>	8,00
10-rat99	1238	650,67	<b>0,00</b>	14,50	<b>0,00</b>	14,20
25-rat99	1269	351,15	<b>0,00</b>	20,30	<b>0,00</b>	20,00
50-rat99	1249	2797,58	<b>0,00</b>	23,80	<b>0,00</b>	23,00
25-kroA100	21917	3513,57	<b>0,00</b>	20,40	<b>0,00</b>	20,90
50-kroA100	21453	947,55	<b>0,00</b>	22,40	<b>0,00</b>	23,30
10-kroB100	22440	4991,44	<b>0,00</b>	17,80	<b>0,00</b>	17,30
50-kroB100	22355	2579,22	<b>0,00</b>	24,50	<b>0,00</b>	24,90
25-eil101	663	709,45	<b>0,00</b>	22,20	<b>0,00</b>	21,50
50-eil101	644	275,33	0,16	23,70	0,16	23,20
25-lin105	14438	6224,55	<b>0,00</b>	21,50	0,03	19,30
50-lin105	14379	1577,21	<b>0,00</b>	27,70	<b>0,00</b>	28,70
75-lin105	14521	15886,77	<b>0,00</b>	29,50	<b>0,00</b>	30,20
Val. Médios	-	2266,73	<b>0,03</b>	13,02	0,05	13,17

**Fonte:** Autor

Na primeira coluna da Tabela 1 encontra-se a instância, na segunda o valor ótimo obtido pelo CPLEX e na terceira, o tempo demandado pelo CPLEX em segundos  $t_{\text{CPLEX}}(s)$ . Na quarta o  $gap_{hc}$  de ILS-VNRD-P entre a solução alcançada e o valor ótimo e na quinta o tempo médio de execução  $t_{\text{ILS-VNRD-P}}(s)$  de medido em segundos. Na sexta e sétima colunas são mostradas o  $gap_{hc}$  do melhor valor obtido por ILS-VNRD-nP e o tempo médio  $t_{\text{ILS-VNRD-nP}}(s)$ , respectivamente.

Os resultados apresentados mostram que o  $gap_{hc}$  médio dos melhores valores alcançados de ILS-VNRD-P ficou em 0,03% com o tempo médio total de 13,02s. O método heurístico ILS-VNRD-P encontrou 24 soluções ótimas de um total de 27. O método heurístico ILS-VNRD-nP conseguiu alcançar 22 soluções ótimas

com  $gap_{hc}$  médio dos melhores valores igual a 0,05% e tempo médio total igual a 13,17s.

A perturbação *Pertdg* foi ativada em algumas instâncias, porque estas não têm vértices suficientes nos grupos para realizar a perturbação *Pert*. Precisa no mínimo oito vértices em um grupo para realizar a *Pert*.

Um segundo experimento foi realizado e são mostrados na Tabela 2. A primeira coluna da Tabela 2 é mostrada as instâncias, seguida com os seus identificadores (coluna 2), número de vértices (nós), número de grupos e tipo. Mostram-se ainda as soluções e os limites inferiores (*lower bounds*) gerados pelo software CPLEX para as instâncias relacionadas. Na última coluna é mostrado o  $gap_i$  (%) entre valor da solução e o limite inferior.

Devido às longas iterações ocorridas no CPLEX para a maioria das instâncias, determinamos um tempo limite de 7200s para sua execução. Este tempo limite é o suficiente para encontrar bons valores para os limites inferiores. Para a instância  $J_{18}$ , a solução ótima foi encontrada pelo CPLEX em 442s.

Para estas instâncias (Tabela 2), os testes computacionais dos métodos heurísticos foram realizados utilizando como critério de parada um tempo máximo. Os resultados destes testes são mostrados na Tabela 3. Foi utilizado um tempo limite de 720 segundos para os métodos heurísticos, porque para cada instância, as heurísticas são executadas 10 vezes e a melhor solução encontrada nestas execuções por duas horas é retornada, assim como o valor médio. Somente a execução da instância  $J_{18}$  não foi limitada em duas horas, devido à solução ótima ser encontrada antes.

**Tabela 2** - Valores do CPLEX para um conjunto de instâncias com número de nós, número de grupos e tipos

Instâncias	Id.	# nós	# Vi	Tipo	CPLEX		
					Valor	Lim. Inf.	(%)
50-gil262	$J_1$	262	50	1	135529	135374,68	0,11
10-lin318	$J_2$	318	10	1	534640	526412,07	1,54
10-pcb442	$J_3$	442	10	1	547152	536478,33	1,95
C1k.0	$J_4$	1000	10	2	134025123	131354923,50	1,99
C1k.1	$J_5$	1000	10	2	130750874	128540131,50	1,69
C1k.2	$J_6$	1000	10	2	144341485	141501445	1,97
300-6	$J_7$	300	6	3	8969	8915,18	0,60
400-6	$J_8$	400	6	3	9117	9021,51	1,05
700-20	$J_9$	700	20	3	41638	41274,00	0,87
200-4-h	$J_{10}$	200	4	4	63429	62244,84	1,87
200-4-x1	$J_{11}$	200	4	4	60797	60242,96	0,91
600-8-z	$J_{12}$	600	8	4	132897	127901,75	3,76
600-8-x2	$J_{13}$	600	8	4	132228	127901,75	3,27
300-5-108	$J_{14}$	300	5	5	68361	67128,93	1,80
300-20-111	$J_{15}$	300	20	5	311286	308595,45	0,86
500-15-306	$J_{16}$	500	15	5	196001	193522,8	1,26
500-25-308	$J_{17}$	500	25	5	367586	364108,13	0,95
25-eil101	$J_{18}$	101	25	6	23671	23668,63	<b>0,01</b>
42-a280	$J_{19}$	280	42	6	130043	129560,53	0,37
144-rat783	$J_{20}$	783	144	6	916174	913715,52	0,27
						$gap_i$ médio=	1,36

Fonte: Autor

A comparação entre os melhores valores obtidos pelos métodos heurísticos e pelo CPLEX com este critério de parada é mostrada na Tabela 3, assim como os valores médios. Na primeira coluna são mostrados os identificadores das instâncias, na segunda e terceira, o valor de  $gap_i$  para os melhores valores encontrados pelos métodos heurísticos e em seguida o valor  $gap_i$  relacionado aos valores médios.

**Tabela 3** - Valores alcançados pelos métodos heurísticos

Id.	Melhores Valores		Valores Médios	
	ILS-VNRD-P	ILS-VNRD-nP	ILS-VNRD-P	ILS-VNRD-nP
$J_1$	<b>0,08</b>	<b>0,08</b>	0,11	<b>0,10</b>
$J_2$	<b>0,60</b>	0,79	<b>0,68</b>	0,84
$J_3$	<b>0,33</b>	0,34	0,47	<b>0,45</b>
$J_4$	<b>1,36</b>	1,43	<b>1,44</b>	1,60
$J_5$	<b>0,86</b>	1,03	<b>1,02</b>	1,15
$J_6$	1,37	<b>1,27</b>	1,50	<b>1,38</b>
$J_7$	0,23	<b>0,22</b>	0,31	<b>0,29</b>
$J_8$	0,39	<b>0,35</b>	0,50	<b>0,45</b>
$J_9$	<b>0,41</b>	0,44	<b>0,47</b>	0,52
$J_{10}$	<b>0,99</b>	<b>0,99</b>	<b>1,27</b>	1,29
$J_{11}$	0,83	<b>0,65</b>	1,02	<b>0,91</b>
$J_{12}$	<b>1,09</b>	1,21	<b>1,42</b>	1,51
$J_{13}$	1,20	<b>1,04</b>	1,44	<b>1,26</b>
$J_{14}$	1,06	<b>0,94</b>	1,16	<b>1,09</b>
$J_{15}$	<b>0,39</b>	0,50	<b>0,43</b>	0,54
$J_{16}$	0,74	<b>0,73</b>	<b>0,83</b>	0,86
$J_{17}$	0,51	<b>0,44</b>	0,56	<b>0,50</b>
$J_{18}$	0,04	0,02	<b>0,07</b>	<b>0,07</b>
$J_{19}$	<b>0,13</b>	<b>0,13</b>	0,19	<b>0,15</b>
$J_{20}$	<b>0,13</b>	0,15	<b>0,15</b>	0,16
$gap_i$ médio=	<b>0,64</b>	<b>0,64</b>	<b>0,75</b>	0,76

Fonte: Autor

Na avaliação entre as melhores soluções obtidas pelos métodos heurísticos e o CPLEX, compara-se a melhor solução de cada método com os limites inferiores obtidos pelo método exato. Neste contexto o valor médio de  $gap_i$  para o CPLEX foi igual a 1,36%, para **ILS-VNRD-P e ILS-VNRD-nP, ambos iguais a 0,64%**. Os valores em negrito na Tabela 3 mostram onde os métodos heurísticos alcançaram melhor desempenho.

O CPLEX obteve uma melhor solução no total de 20 instâncias, ILS-VNRD-P e ILS-VNRD-nP com oito cada e em três instâncias ambos os métodos heurísticos obtiveram o mesmo resultado. Na comparação entre as soluções médias obtidas pelos métodos heurísticos, o  $gap_i$  de ILS-VNRD-P ficou em 0,75% e de ILS-VNRD-nP igual a 0,76%.

#### 4.2 Resultados dos Métodos Heurísticos Propostos para Instâncias de Grande Porte



Foram realizados novos testes em instâncias de grande porte para comprovar a eficiência dos métodos heurísticos. Devido às instâncias serem de portes maiores, foi estabelecido um limite de 1080 segundos para cada execução do método heurístico ILS-VNRD-P e ILS-VNRD-nP.

A Tabela 4 mostra as instâncias com seus identificadores, número de nós, número de grupos, tipos e  $gap_h$  encontrados. O percentual do  $gap_h$  relativo aos **melhores valores** ou aos **valores médios** alcançados pelos métodos heurísticos é calculado em relação ao **melhor valor** ou ao **melhor valor médio** encontrado por ILS-VNRD-P ou ILS-VNRD-nP.

Observa-se que o método heurístico ILS-VNRD-nP encontrou oito soluções melhores num total de 15 e ILS-VNRD-P com sete. O valor médio de  $gap_h$  dos melhores valores encontrados por ILS-VNRD-P é 0,62% e de ILS-VNRD-nP igual a 0,46%. Para os valores médios alcançados pelos métodos heurísticos, o  $gap_h$  médio de ILS-VNRD-P foi igual a 0,65% e de ILS-VNRD-nP igual a 0,53%.

**Tabela 4** - Valores dos métodos heurísticos propostos para instâncias de grande porte

Instâncias	Id.	#nós	#Vi	Tipo	Melhores Valores		Valores Médios	
					ILS-VNRD-P	ILS-VNRD-nP	ILS-VNRD-P	ILS-VNRD-nP
49-pcb1173	$J_{21}$	1173	49	6	0,54	<b>0,00</b>	0,59	<b>0,00</b>
100-pcb1173	$J_{22}$	1173	100	6	<b>0,00</b>	1,57	<b>0,00</b>	1,28
144-pcb1173	$J_{23}$	1173	144	6	<b>0,00</b>	1,21	<b>0,00</b>	0,92
10-nrw1379	$J_{24}$	1379	10	6	0,27	<b>0,00</b>	<b>0,00</b>	0,27
12-nrw1379	$J_{25}$	1379	12	6	2,23	<b>0,00</b>	2,29	<b>0,00</b>
1500-10-503	$J_{26}$	1500	10	5	0,40	<b>0,00</b>	0,36	<b>0,00</b>
1500-20-504	$J_{27}$	1500	20	5	1,93	<b>0,00</b>	1,94	<b>0,00</b>
1500-50-505	$J_{28}$	1500	50	5	<b>0,00</b>	0,81	1,30	<b>0,00</b>
1500-100-506	$J_{29}$	1500	100	5	<b>0,00</b>	1,05	<b>0,00</b>	0,20
1500-150-507	$J_{30}$	1500	150	5	<b>0,00</b>	0,14	<b>0,00</b>	1,30
2000-10-a	$J_{31}$	2000	10	4	<b>0,00</b>	1,53	<b>0,00</b>	1,08
2000-10-h	$J_{32}$	2000	10	4	<b>0,00</b>	0,51	<b>0,00</b>	2,47
2000-10-z	$J_{33}$	2000	10	4	0,70	<b>0,00</b>	0,80	<b>0,00</b>
2000-10-x1	$J_{34}$	2000	10	4	2,84	<b>0,00</b>	2,41	<b>0,00</b>
2000-10-x2	$J_{35}$	2000	10	4	0,42	<b>0,00</b>	<b>0,00</b>	0,44
$gap_h$ médio =					0,62	<b>0,46</b>	0,65	<b>0,53</b>

Fonte: Autor

### 4.3 Comparações e Discussões com os Métodos da Literatura

Foram realizadas comparações entre os métodos propostos (ILS-VNRD-P e ILS-VNRD-nP) com os métodos da literatura (ILS-VND-P e ILS-VND-nP) de Mestria,

Ochi e Martins (2011). Utilizamos o *hardware* descrito no início da seção 4, tanto para os métodos propostos quanto para os da literatura, com a finalidade de realizar as comparações entre estes métodos e avaliar os tempos computacionais demandados e ainda, os valores de *gaps* alcançados.

**Tabela 5** – Comparações entre CPLEX, métodos heurísticos propostos e os da literatura

Instâncias	CPLEX	ILS-VNRD-P (%)	ILS-VNRD-nP (%)	ILS-VND-P (%)	ILS-VND-nP (%)
5-eil51	437	0,00	0,00	0,00	0,00
10-eil51	440	0,00	0,00	0,00	0,00
15-eil51	437	0,00	0,00	0,00	0,00
5-berlin52	7991	0,00	0,00	0,00	0,11
10-berlin52	7896	0,00	0,52	0,00	0,52
15-berlin52	8049	0,00	0,00	0,00	0,00
5-st70	695	0,00	0,00	0,00	0,00
10-st70	691	0,00	0,00	0,00	0,43
15-st70	692	0,00	0,00	0,00	0,87
5-eil76	559	0,36	0,18	0,00	0,00
10-eil76	561	0,00	0,00	0,36	1,07
15-eil76	565	0,18	0,35	0,00	0,88
5-pr76	108590	0,00	0,00	0,00	0,00
10-pr76	109538	0,00	0,00	0,00	0,00
15-pr76	110678	0,00	0,00	0,49	0,00
10-rat99	1238	0,00	0,00	0,00	0,00
25-rat99	1269	0,00	0,00	0,00	0,00
50-rat99	1249	0,00	0,00	0,00	0,00
25-kroA100	21917	0,00	0,00	0,00	0,00
50-kroA100	21453	0,00	0,00	0,00	0,00
10-kroB100	22440	0,00	0,00	0,16	0,91
50-kroB100	22355	0,00	0,00	0,00	0,00
25-eil101	663	0,00	0,00	0,45	1,06
50-eil101	644	0,16	0,16	1,09	0,16
25-lin105	14438	0,00	0,03	0,00	1,38
50-lin105	14379	0,00	0,00	0,00	0,00
75-lin105	14521	0,00	0,00	0,15	0,00
Val. Médios	-	0,03	0,05	0,10	0,27

Fonte: Autor

Para as instâncias da Tabela 1, o critério de parada é limitado pelo número de iterações a cada execução e para as instâncias da Tabela 2 e da Tabela 4, o critério de parada foi estabelecido por um tempo limite igual a 720s e 1080s, respectivamente a cada execução.

Para as instâncias da Tabela 1 os resultados mostraram que **ILS-VNRD-P encontrou 24 soluções ótimas de um total de 27 instâncias com valor médio do  $gap_{hc}$ , para os melhores valores, igual a 0,03% e tempo médio total de 13,02s.**

O valor médio de  $gap_{hc}$  dos melhores valores de *ILS-VNRD-nP* foi 0,05% com o tempo médio de 13,17s e encontrou 22 soluções ótimas. O método ILS-VND-P obteve valor médio de  $gap_{hc}$  dos melhores valores igual a 0,10% e tempo médio igual a 11,37s e ILS-VND-nP com valor médio de  $gap_{hc}$  de 0,27% e tempo médio de 12,05. O ILS-VND-P encontrou 21 soluções ótimas e o método ILS-VND-nP com 17 soluções ótimas.

A Tabela 5 mostra em detalhes os valores alcançados pelos métodos para as instâncias da Tabela 1.

Para as instâncias da Tabela 2, o valor médio do  $gap_i$  dos melhores valores de **ILS-VNRD-P** foi de **0,64%**, assim como o de **ILS-VNRD-nP** que ficou com **0,64%**, de ILS-VND-P em 0,65% e de ILS-VND-nP com 0,67%.

A Tabela 6 mostra os melhores valores alcançados pelos métodos heurísticos propostos e os da literatura para as instâncias da Tabela 2.

A seguir, na Tabela 7, são mostradas as comparações entre os métodos para as instâncias de grande porte da Tabela 4. Na primeira coluna da Tabela 7 são mostrados os identificadores, na segunda e terceira, os melhores valores encontrados pelos métodos propostos. Na quarta e quinta colunas o valor do  $gap$  para os melhores valores encontrados pelos métodos da literatura.

**Tabela 6** – Melhores valores alcançados pelos métodos heurísticos propostos e os da literatura

Id.	Melhores Valores			
	ILS-VNRD-P	ILS-VNRD-nP	ILS-VND-P	ILS-VND-nP
$J_1$	<b>0,08</b>	<b>0,08</b>	0,11	0,10
$J_2$	<b>0,60</b>	0,79	0,73	0,73
$J_3$	<b>0,33</b>	0,34	0,56	0,46
$J_4$	1,36	1,43	<b>1,30</b>	1,45
$J_5$	<b>0,86</b>	1,03	0,88	0,99
$J_6$	1,37	1,27	1,29	<b>1,24</b>
$J_7$	0,23	0,22	0,24	<b>0,21</b>
$J_8$	0,39	0,35	<b>0,33</b>	<b>0,33</b>
$J_9$	<b>0,41</b>	0,44	0,51	0,43
$J_{10}$	0,99	0,99	<b>0,89</b>	<b>0,89</b>
$J_{11}$	0,83	0,65	<b>0,55</b>	1,13
$J_{12}$	1,09	1,21	<b>1,05</b>	1,17
$J_{13}$	1,20	<b>1,04</b>	1,43	<b>1,04</b>
$J_{14}$	1,06	<b>0,94</b>	1,03	1,01
$J_{15}$	<b>0,39</b>	0,50	0,46	0,52
$J_{16}$	0,74	<b>0,73</b>	0,80	0,86
$J_{17}$	0,51	<b>0,44</b>	0,51	0,47
$J_{18}$	0,04	<b>0,02</b>	<b>0,02</b>	0,04
$J_{19}$	0,13	0,13	<b>0,12</b>	<b>0,12</b>
$J_{20}$	<b>0,13</b>	0,15	0,16	0,16

$gap_i$ médio=	<b>0,64</b>	<b>0,64</b>	0,65	0,67
----------------	-------------	-------------	------	------

Fonte: Autor

O percentual do *gap* (%) é calculado em relação ao melhor valor encontrado pelo ILS-VND-P, obtendo assim um *gap* médio igual 0,00%. O *gap* médio de ILS-VNRD-P foi igual a -0,25%, de ILS-VNRD-nP em **-0,41%** e de ILS-VND-nP em 0,12%. Na maioria das instâncias de grande porte (foram 11 de um total de 15) o método heurístico ILS-VNRD-nP obteve *gaps* negativos. Isto significa que o método ILS-VNRD-nP alcançou um valor melhor do que o encontrado pelo ILS-VND-P. Nestas comparações observamos nos valores médios do *gap* de ILS-VNRD-P e do ILS-VNRD-nP que estes obtiveram melhores resultados do que ILS-VND-P e ILS-VND-nP.

**Tabela 7** - Comparações entre os métodos heurísticos propostos com os da literatura para instâncias de grande porte

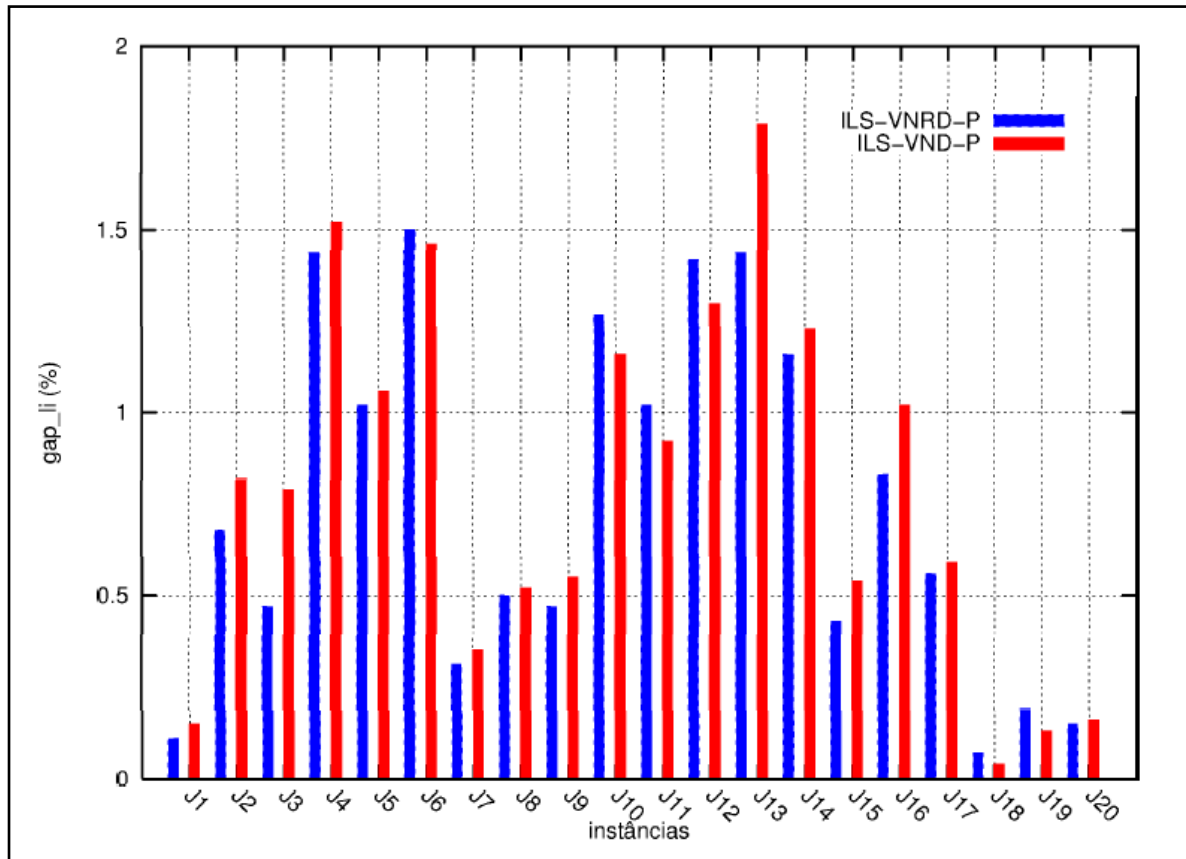
Id.	Melhores Valores			
	ILS-VNRD-P	ILS-VNRD-nP	ILS-VND-P	ILS-VND-nP
$J_{21}$	-2,42	<b>-2,95</b>	0,00	-1,04
$J_{22}$	<b>-1,86</b>	-0,32	0,00	-0,21
$J_{23}$	0,92	2,14	0,00	<b>-0,22</b>
$J_{24}$	-0,32	-0,58	0,00	<b>-1,10</b>
$J_{25}$	0,67	<b>-1,53</b>	0,00	-0,48
$J_{26}$	0,21	<b>-0,19</b>	0,00	-0,05
$J_{27}$	0,58	<b>-1,33</b>	0,00	1,21
$J_{28}$	0,26	1,08	<b>0,00</b>	3,67
$J_{29}$	0,93	1,99	<b>0,00</b>	1,32
$J_{30}$	<b>-0,63</b>	-0,49	0,00	0,31
$J_{31}$	<b>-1,67</b>	-0,16	0,00	0,25
$J_{32}$	-2,14	-1,63	0,00	<b>-2,59</b>
$J_{33}$	-0,84	<b>-1,53</b>	0,00	-0,88
$J_{34}$	0,87	-1,92	0,00	<b>-2,01</b>
$J_{35}$	1,70	1,27	0,00	<b>-0,02</b>
$gap$ médio =	-0,25	<b>-0,41</b>	0,00	-0,12

Fonte: Autor

Nos resultados apresentados, tanto para instâncias de pequeno porte quanto para as instâncias de grande porte, observamos que a introdução do **Método de Descida Aleatória em Vizinhança Variável** permite melhores resultados do que o tradicional Método de Descida em Vizinhança Variável. Realizar a busca local aleatória em vizinhança variável permite escapar de mínimos locais e encontrar

melhores valores. Desta forma, não é possível alcançar este resultado através das buscas locais em vizinhanças sequências determinísticas.

**Figura 6** – Comparações entre os métodos heurísticos penalizados para os valores médios



Fonte: Autor

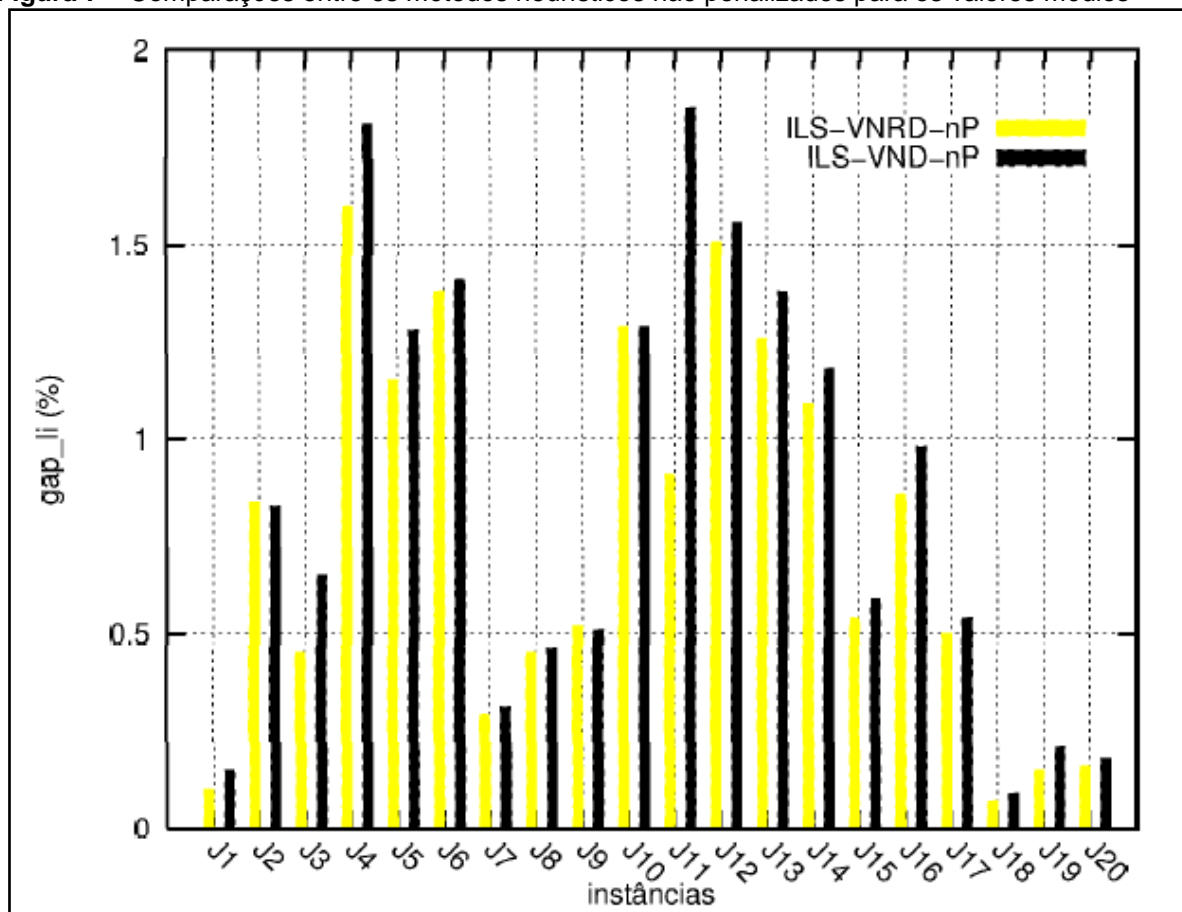
#### 4.4 Comparações entre os Métodos Heurísticos para os Valores Médios através de Análises Gráficas

Para as instâncias da Tabela 2 realizamos comparações, considerando os valores médios alcançados, entre os métodos heurísticos propostos neste trabalho e os métodos da literatura utilizando análises gráficas. Os *gaps* médios das soluções médias alcançadas foram: para o **ILS-VNRD-P o  $gap_i$  ficou em 0,75%**, o de *ILS-VNRD-nP* igual a 0,76%, contra 0,80% de ILS-VND-P e 0,86% de ILS-VND-nP.

Nas comparações entres os métodos, utilizamos o mesmo construtivo (IMPP ou IMPnP), mas com busca local diferente, tanto para os métodos penalizados (P)

(ILS-VNRD-P e ILS-VND-P), como para os não penalizados (nP) (ILS-VNRD-nP e ILS-VND-nP).

**Figura 7** – Comparações entre os métodos heurísticos não penalizados para os valores médios



Fonte: Autor

A Figura 6 mostra o  $gap_i$ , para cada instância da Tabela 2, relacionado aos valores médios alcançados pelos métodos heurísticos penalizados e a Figura 7 para os métodos não penalizados. Das 20 instâncias analisadas, em 14 o método ILS-VNRD-P obteve melhor resultado do que o ILS-VND-P. Entretanto, **o ILS-VNRD-nP alcançou 18 melhores resultados** quando comparado ao ILS-VND-nP.

## 5 CONSIDERAÇÕES FINAIS E CONCLUSÕES

Através dos resultados computacionais alcançados nós constatamos que os métodos heurísticos propostos com *Iterated Local Search* e uma busca local

aleatória em vizinhança variável obtêm soluções de boa qualidade para o PCVG. Observamos ainda que os métodos heurísticos encontram soluções para o PCVG em tempo computacional relativamente baixo tanto para as instâncias de pequeno porte como para as de grande porte.

Nas comparações com a literatura, os métodos heurísticos propostos neste trabalho mostraram-se com melhor resultado e foram competitivos com um método exato. Desta forma, concluímos que o método proposto de descida aleatória em vizinhança variável pode ser utilizado para solucionar o PCVG e ainda ser estendido para outros problemas de natureza combinatória.

Nós destacamos neste trabalho a vantagem dos métodos heurísticos desenvolvidos formados por módulos independentes. Neste sentido, os módulos propostos podem ser modificados para balancear qualidade na solução e tempo computacional. E ainda, podemos desenvolver outros módulos independentes incorporando maiores complexidades a cada um, com a finalidade de obter soluções de alta qualidade para o PCVG.

Trabalhos futuros sugerem construir métodos heurísticos baseados no GRASP, *Variable Neighborhood Search* ou Algoritmos Evolutivos, incorporando como busca local o algoritmo proposto, de descida aleatória em vizinhança variável, para resolver o PCVG.

## AGRADECIMENTOS

*O autor agradece ao Instituto Federal de Educação, Ciência e Tecnologia do Espírito Santo pelo apoio e fomento.*

## REFERÊNCIAS

- ANZANELLO, M. J.; FOGLIATTO, F. S. Programação de Tarefas Baseada em Curvas de Aprendizado para Linhas de Produção Customizadas. **Revista Produção Online**, v.11, n. 3, p. 851-870, 2011. <http://dx.doi.org/10.14488/1676-1901.v11i3.955>
- CHISMAN, J. A. The Clustered Traveling Salesman Problem. **Computers & Operations Research**, v. 2, n.2, p.115-119, 1975. [http://dx.doi.org/10.1016/0305-0548\(75\)90015-5](http://dx.doi.org/10.1016/0305-0548(75)90015-5)

CPLEX. ILOG CPLEX 11.2 **User's Manual and Reference Manual**. ILOG S.A, 2009.

DING, C.; CHENG, Y.; HE, M. Two-Level Genetic Algorithm for Clustered Traveling Salesman Problem with Application in Large-Scale TSPs. **Tsinghua Sci. and Technology**, v. 12, n.4, p.459-465, 2007. [http://dx.doi.org/10.1016/S1007-0214\(07\)70068-8](http://dx.doi.org/10.1016/S1007-0214(07)70068-8)

DONG, X.; CHEN, P.; HUANG, H.; NOWAK, M. A multi-restart iterated local search algorithm for the permutation flow shop problem minimizing total flow time. **Computers & Operations Research**, v. 40, n. 2, p. 627-632, 2013. <http://dx.doi.org/10.1016/j.cor.2012.08.021>

FESTA, P.; RESENDE, M. GRASP: basic components and enhancements. **Telecommunication Systems**, v.46, n.3, p. 253-271, 2011. <http://dx.doi.org/10.1007/s11235-010-9289-z>

GHAZIRI, H.; OSMAN, I. H. A Neural Network for the Traveling Salesman Problem with Backhauls. **Computers & Industrial Engineering**, v. 44, n.2, p.267-281, 2003. [http://dx.doi.org/10.1016/S0360-8352\(02\)00179-1](http://dx.doi.org/10.1016/S0360-8352(02)00179-1)

HANSEN, P.; MLADENOVIĆ, N. Variable Neighborhood Search. In: Glover, F. W.; Kochenberger, G. A., editores, **Handbook of Metaheuristics**, p. 145-184, 2003.

JARBOUI, B.; DERBEL, H.; HANAFI, S.; MLADENOVIĆ, N. Variable neighborhood search for location routing. **Computers & Operations Research**, v. 40, n. 1, p. 47-57, 2013. <http://dx.doi.org/10.1016/j.cor.2012.05.009>

JONGENS, K.; VOLGENANT, T. The Symmetric Clustered Traveling Salesman Problem. **European Journal of Operational Research**, v. 19, n.1, p. 68-75, 1985. [http://dx.doi.org/10.1016/0377-2217\(85\)90309-1](http://dx.doi.org/10.1016/0377-2217(85)90309-1)

JUAN, A. A.; LOURENÇO, H. R.; MATEO, M.; LUO, R.; CASTELLA, Q. Using iterated local search for solving the flow-shop problem: Parallelization, parametrization, and randomization issues. **International Transactions in Operational Research**, v. 21, n. 1, p. 103-126, 2014. <http://dx.doi.org/10.1111/itor.12028>

LAMB, J. D. Variable neighbourhood structures for cycle location problems. **European Journal of Operational Research**, v. 223, n. 1, p. 15-26, 2012. <http://dx.doi.org/10.1016/j.ejor.2012.05.038>

LAPORTE, G.; PALEKAR, U. Some Applications of the Clustered Travelling Salesman Problem. **Journal of the Operational Research Society**, v. 53, n.9, p. 972-976, 2002. <http://dx.doi.org/10.1057/palgrave.jors.2601420>



LAPORTE, G.; POTVIN, J.-Y.; QUILLERET, F. A Tabu Search Heuristic using Genetic Diversification for the Clustered Traveling Salesman Problem. **Journal of Heuristics**, v. 2, n.3, p.187-200, 1996. <http://dx.doi.org/10.1007/BF00127356>

LOKIN, F. C. J. Procedures for Travelling Salesman Problems with Additional Constraints. **European Journal of Operational Research**, v. 3, n.2, p.135-141, 1979. [http://dx.doi.org/10.1016/0377-2217\(79\)90099-7](http://dx.doi.org/10.1016/0377-2217(79)90099-7)

LOURENÇO, H. R.; MARTIN, O. C.; STÜTZLE, T. Iterated local search. In: Glover, F.; Kochenberger, G., (eds), **Handbook of Metaheuristics**, Kluwer Academic Publishers, p.321-353. 2002.

LUDWIG, I. P.; ANZANELLO, M. J.; VIDOR, G. Minimização dos tempos de atraso na programação de tarefas em uma empresa de desenvolvimento de softwares. **Revista Produção Online**, v.13, n. 2, p. 479-499, 2013. <http://dx.doi.org/10.14488/1676-1901.v13i2.1069>

MESTRIA, M.; OCHI, L. S.; MARTINS, S. L. Iterated Local Search para o Problema do Caixeiro Viajante com Grupamentos. In: SIMPÓSIO DE PESQUISA OPERACIONAL & LOGÍSTICA DA MARINHA, 14., 2011. Rio de Janeiro, RJ: **Anais do XIV SPOLM** (CD-ROM), v.1, ISSN 1806-3632, 2011.

MESTRIA, M. Métodos Heurísticos usando GRASP, reconexão de caminhos e busca em vizinhança variável para o problema do caixeiro viajante com grupamentos. **Revista Produção Online**, v.13, n.3, p.1002-1033, 2013. <http://dx.doi.org/10.14488/1676-1901.v13i3.1350>

NAGANO, M. S.; MESQUITA, M. S. Métodos Heurísticos para o Problema de Programação *Flow Shop* com Tempos de *Setup* Separados. **Revista Produção Online**, v.12, n. 2, p. 499-521, 2012. <http://dx.doi.org/10.14488/1676-1901.v12i2.939>

SOBREIRO, V. A.; NAGANO, M. S. Um novo método heurístico para a otimização de mix de produção baseado na teoria das restrições e no problema da mochila. **Revista Produção Online**, v.13, n. 2, p. 520-543, 2013. <http://dx.doi.org/10.14488/1676-1901.v13i2.1125>

TARANTILIS, C. D.; STAVROPOULOU, F.; REPOUSSIS, P. P. The Capacitated Team Orienteering Problem: A Bi-level Filter-and-Fan method. **European Journal of Operational Research**, v. 224, n. 1, p. 65-78, 2013. <http://dx.doi.org/10.1016/j.ejor.2012.07.032>

VILLEGAS, J. G; PRINS, C.; PRODHON, C.; MEDAGLIA, A. L.; VELASCO, N. A matheuristic for the truck and trailer routing problem. **European Journal of Operational Research**, v. 230, n. 2, p. 231-244, 2013. <http://dx.doi.org/10.1016/j.ejor.2013.04.026>

WEINTRAUB, A.; ABOUD, J.; FERNANDEZ, C.; LAPORTE, G.; RAMIREZ, E. An Emergency Vehicle Dispatching System for an Electric Utility in Chile. **Journal of the**

**Operational Research Society**, v. 50, p. 690-696, 1999.  
<http://dx.doi.org/10.2307/3010322>



Artigo recebido em 31/12/2013 e aceito para publicação em 14/04/2014  
DOI: <http://dx.doi.org/10.14488/1676-1901.v14i4.1721>