

## HEURÍSTICAS HÍBRIDAS PARA O PROBLEMA DO CAIXEIRO VIAJANTE COM GRUPAMENTOS

Mário Mestria<sup>(1,2)</sup>, Luiz Satoru Ochi<sup>(1)</sup>, Simone de Lima Martins<sup>(1)</sup>

<sup>(1)</sup>Instituto de Computação - Universidade Federal Fluminense  
Niterói, RJ, Brasil

<sup>(2)</sup>Coordenadoria da Área de Informática – Campus Colatina - Instituto Federal do Espírito Santo

Avenida Arino Gomes Leal - 1700 - Bairro Santa Margarida ,Colatina, ES, Brasil

e-mails: {mmestria, satoru, simone}@ic.uff.br

**Resumo** - O artigo aborda o Problema do Caixeiro Viajante com Grupamentos (PCVG), uma generalização do Clássico Problema do Caixeiro Viajante, onde os vértices são particionados em grupos disjuntos e o objetivo é encontrar um ciclo hamiltoniano de custo mínimo que passa por todos os vértices e visita os vértices de cada grupo de forma contígua. Para o PCVG foram desenvolvidas quatro versões heurísticas híbridas utilizando-se GRASP e *Iterated Local Search* (ILS). Os resultados obtidos com as heurísticas híbridas são comparados entre si e com os resultados exatos obtidos pelo *software* CPLEX utilizando-se uma formulação matemática do problema. Estes resultados mostram que a heurística híbrida ILS-VND-IMPP apresenta desempenho melhor comparado aos outros métodos híbridos apresentados.

**Palavras-chave** – Heurísticas Híbridas, Pesquisa Operacional, Otimização Combinatória, Problema do Caixeiro Viajante, Inteligência Computacional.

### 1. Introdução

O Problema do Caixeiro Viajante com Grupamentos (PCVG) surgiu ao se estudar um problema de roteamento automático em sistemas de armazenagem (CHISMAN, 1975). Este problema é uma extensão do clássico Problema do Caixeiro Viajante (PCV), e seu objetivo é encontrar um ciclo hamiltoniano de custo mínimo que passe por todos os vértices do grafo associado e visita os vértices de cada grupo de forma contígua. Naturalmente quando o número de grupos for igual a um, o PCVG se reduz ao PCV. Assim, pode-se afirmar que o PCVG também pertence à classe NP-Difícil limitando com isso o uso exclusivo de métodos exatos. Neste contexto a resolução para o PCVG proposta neste trabalho utiliza conceitos de heurísticas híbridas.

Diversas aplicações para o PCVG encontram-se disponíveis na literatura, por exemplo, em roteamento automático em armazéns (Chisman, 1975), planejamento da produção (Lokin, 1979), despacho de veículos de emergência, (Weintraub *et al.*, 1999), sistemas de manufaturas (Laporte & Palekar, 2002) e desfragmentação de discos rígidos. Em (Ghaziri & Osman, 2003) é mostrada uma aplicação comercial envolvendo supermercados, lojas e fornecedores de mercadorias. Apesar de várias aplicações serem modeladas através do PCVG, na literatura este tipo de problema é pouco explorado. Duas versões são utilizadas para resolver o PCVG, uma no qual se define *a priori* a sequência dos grupos a serem visitados (Gendreau, Laporte & Potvin, 1994), (Potvin & Guertin, 1995), (Laporte, Potvin & Quilleret, 1996) e (Anily, Bramel & Hertz, 1999) e uma outra, denominada versão genérica, cuja sequência de visita dos grupos deve ser definida pelo algoritmo. Desta forma verificamos que a segunda versão é

mais complexa e geralmente mais próxima dos modelos reais, sendo portanto selecionada para estudo neste trabalho.

Uma das metaheurísticas utilizada neste trabalho para resolver o PCVG é o GRASP (*Greedy Randomized Adaptive Search Procedures*). O GRASP tem se destacado na literatura para resolver diferentes problemas de otimização combinatória (RESENDE & RIBEIRO, 2002). O algoritmo GRASP é constituído de duas fases: uma de construção e outra de busca local. Na fase de construção, uma solução viável é construída, e na fase de busca local sua vizinhança é investigada até uma solução ótima local ser encontrada. A melhor solução obtida ao final de um determinado número de iterações é retornada como a solução do algoritmo. Em Hernández-Pérez, Rodríguez-Martín & Salazar-González (2009) foi apresentado um GRASP combinado com *Variable Neighborhood Descent* (VND) (Hansen & Mladenović, 2003) para um problema de roteamento. A estrutura do VND não é complexa, de fácil implementação e seus componentes de vizinhanças podem ser alterados aumentando ou diminuindo sua complexidade para se atingir um compromisso entre *qualidade da solução e tempo computacional exigido*. Assim, propõe-se neste trabalho, utilizar o VND como módulo de busca local a ser incorporado nas heurísticas híbridas para resolução do PCVG. Em Mestria, Ochi & Martins (2009a) são apresentados diversos algoritmos GRASP para o PCVG com a incorporação de módulos mais sofisticados tais como a reconexão de caminhos e/ou VND mostrando que estes módulos são fundamentais no desempenho final destes algoritmos.

A metaheurística *Iterated Local Search* (ILS) (Lourenço, Martin & Stützle, 2002) também tem se mostrado promissora para diferentes problemas de natureza combinatória. Os fundamentos do ILS foram introduzidos por Martin, Otto & Felten (1992) e Martin & Otto (1996), que aplicaram ao PCV usando instâncias Euclidianas. Como busca local foi utilizada uma heurística *k-optimal* e para o método de perturbação o movimento *double bridge*. ILS possui princípios simples, é fácil de implementar e possibilita o armazenamento de informações históricas da estrutura da solução ao longo do algoritmo. A metaheurística ILS também é utilizada neste trabalho para solucionar o PCVG.

Neste trabalho foram desenvolvidas quatro heurísticas híbridas. A primeira heurística utiliza o GRASP (G-RC-VND-P) com reconexão de caminhos, uso do VND na fase de busca local e na fase de construção é usado a Inserção mais Próxima que penaliza as arestas intergrupos (IMPP). A segunda heurística GRASP (G-RC-VND-nP) é idêntica à primeira, exceto que na fase de construção é utilizado a Inserção mais Próxima que não penaliza as arestas intergrupos (IMPnP). A terceira (ILS-VND-P) e quarta (ILS-VND-nP) são heurísticas que utilizam o ILS combinado ao VND. A diferença entre as duas heurísticas ILS está na forma de construir as soluções: o ILS-VND-P utiliza o algoritmo construtivo IMPP e ILS-VND-nP o IMPnP.

O artigo é estruturado conforme as seguintes seções. A segunda seção descreve os trabalhos mais importantes já existentes para o PCVG. A terceira seção descreve as heurísticas híbridas propostas para o PCVG. A quarta seção mostra os resultados computacionais e a finalmente na última seção são descritos conclusões e trabalhos futuros.

## 2. Trabalhos Relacionados

O PCVG foi proposto por Chisman (1975) para resolver um problema real de roteamento em sistemas de armazenagem, onde as ordens de despacho de produtos devem ser respeitadas e estas ordens contêm várias subordens. Cada subordem estabelece os produtos a serem despachados e quantidade de estoque dos produtos. Um veículo motorizado despachará através do armazém coletando os produtos estocados especificados em cada subordem. A restrição nos sistemas de armazenagem especifica que uma subordem deverá ser completamente despachada antes de a próxima ser iniciada. A ordem de coleta dos produtos estocados especificados em cada subordem e a ordenação das subordens é simultaneamente

otimizada. Este tipo de logística foi modelado como um PCVG, onde os vértices são as localizações dos estoques dos produtos, as arestas são as distâncias a serem percorridas pelo veículo motorizado e os grupos são as subordens. O PCVG foi transformado num PCV adicionando-se uma penalidade  $M$  aos custos das arestas que ligam quaisquer dois vértices pertencentes a grupos diferentes e resolvido por um algoritmo exato *branch-and-bound*.

Limites inferiores utilizando Relaxação Lagrangeana foram desenvolvidos em Jongens & Volgenant (1985) para o PCVG. O método para obtenção do limite inferior baseou-se na árvore geradora mínima desenvolvida primeiramente para o PCV. Foram criados vários conjuntos de instâncias variando o número de vértices de 80 a 150 com diferentes tamanhos de grupos. O algoritmo de Jongens & Volgenant (1985) obteve para instâncias testes, *gap* médio para os limites inferiores igual a 0,35%.

Uma Busca Tabu combinada com uma fase de diversificação usando um Algoritmo Genético foi proposta em Laporte, Potvin & Quilleret (1996) para o PCVG. Nesta Busca Tabu a sequência de visita dos grupos é definida *a priori* e os testes computacionais apresentados mostram que a Busca Tabu supera o Algoritmo Genético (AG) de Potvin & Guertin (1995) na qualidade da solução. A comparação no tempo computacional entre a Busca Tabu e AG não foi especificada. Os autores Potvin & Guertin (1996) desenvolveram um Algoritmo Genético para o PCVG sem a pré-fixação da sequência dos grupos, onde este algoritmo trata de forma independente a sequência dos grupos e dos vértices em cada grupo, dando prioridade à solução intergrupos e em seguida a solução intragrupos.

No artigo proposto em Ding, Cheng & He (2007) um outro AG, sem incorporar módulos de busca local, usa o critério de visitação 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. O AG foi dividido em dois níveis chamado de nível baixo e nível alto. No nível mais baixo, o AG acha um ciclo hamiltoniano para cada grupo. No nível mais alto, o algoritmo escolhe aleatoriamente uma aresta a ser excluída no ciclo de cada grupo e simultaneamente determina a sequência de visita de todos os grupos de forma aleatória. A desvantagem desta metodologia está na resolução do problema de forma separada. Primeiro, a metodologia resolve a sequência dos vértices de cada grupo e depois a visita de cada grupo.

Para o PCVG são encontrados diversos algoritmos  $\alpha$ -aproximados com diferentes variantes (Gendreau, Laporte & Potvin, 1994), (Arkin, Hassin & Klein, 1997), (Gendreau, Laporte & Hertz, 1997), (Anily, Bramel & Hertz, 1999) e (Guttmann-Beck *et al.*, 2000) que utilizam adaptações de Christofides (1976) e de Hoogeveen (1991) aplicadas inicialmente ao PCV. A maioria destes algoritmos necessita estabelecer em cada grupo os vértices de entrada e de saída e uma sequência pré-definida de visita dos grupos.

No trabalho de Mestria, Ochi & Martins (2009b) quatro heurísticas baseadas na metaheurística GRASP foram desenvolvidas para o PCVG onde a sequência dos grupos não é estabelecida em princípio. A primeira versão corresponde ao GRASP tradicional e as outras três utilizaram memória adaptativa através do uso de uma técnica denominada reconexão de caminhos. Dois tipos diferentes de reconexão de caminhos foram desenvolvidos, uma (RC2) realizada durante as iterações do GRASP e outra (RC1) realizada depois das iterações do GRASP. Na etapa de busca local foi utilizado somente o método *2-optimal*. As versões que utilizaram reconexão de caminhos (RC1) foram aquelas que conseguiram o melhor desempenho.

Em Mestria, Ochi & Martins (2009a) um dos objetivos foi verificar o impacto ao se incluir no GRASP, módulos de memória adaptativa e busca local mais eficiente utilizando conceitos da metaheurística *Variable Neighborhood Search* (VNS). A versão híbrida que incorpora GRASP, reconexão de caminhos e o método de Descida em Vizinhança Variável (VND)

obtiveram melhor desempenho em relação às outras versões GRASP. Uma única forma para construir soluções ao PCVG baseada na Inserção mais Próxima foi desenvolvida.

### 3. Proposta de Métodos Heurísticos Híbridos

As heurísticas apresentadas neste trabalho ao contrário da maioria dos trabalhos da literatura partem do princípio que não é pré-estabelecida nenhuma sequência de visita aos grupos e incorporam duas formas diferentes para construir soluções. A primeira forma constrói soluções penalizando as arestas intergrupos (IMPP) (neste caso convertendo o PCVG no PCV) e a segunda não estabelece a penalização (IMPnP) (resolvendo o PCVG na sua forma original). Ambas as formas de construir as soluções foram adaptadas do método de Inserção mais Próxima usando os conceitos do GRASP. Nesta implementação utilizou-se o GRASP reativo que tem apresentado bons resultados (RESENDE & RIBEIRO, 2002). Na fase de busca local, todas as heurísticas híbridas utilizam o VND que é caracterizado por trocas sistemáticas de vizinhanças. A medida que uma estrutura de vizinhança não consegue melhorar a solução corrente, permuta-se a estrutura de vizinhança. Caso haja melhoria na solução corrente, retorna-se a primeira estrutura de vizinhança.

Foram utilizadas quatro estruturas de vizinhanças no VND. A primeira ( $N_1$ ) é obtida através de um movimento de deslocamento do vértice pertencente ao ciclo de uma posição para outra, denominada de *1-Shift*. Para definir a segunda estrutura ( $N_2$ ) precisam-se definir dois tipos de procedimentos: *Drop* e *Cheap*. O procedimento *Drop* varre o ciclo de uma solução e retira deste, o vértice que proporcionará a maior economia com sua retirada. O procedimento *Cheap* insere um vértice não pertencente ao ciclo parcial, entre dois vértices adjacentes pertencentes ao ciclo, cuja inserção permita um custo adicional mínimo a solução. O ciclo construído após a inserção do vértice deverá permanecer viável. Diante destas duas definições de procedimentos pode-se estabelecer a segunda estrutura de vizinhança ( $N_2$ ), *1-DropCheap*, cuja definição é a seguinte: uma sequência de procedimentos *Drop* e *Cheap* é realizada sobre uma solução até que nenhum melhoramento seja encontrado. A terceira ( $N_3$ ) é obtida através de permutas entre dois vértices de um mesmo grupo (intra *cluster*), *2-Swap*. A quarta ( $N_4$ ) utiliza a clássica heurística *2-Optimal*. Neste trabalho adotam-se as seguintes estratégias para reduzir o esforço computacional decorrente do uso das estruturas de vizinhanças. Quando se utilizam as vizinhanças  $N_1$  e  $N_3$ , adota-se a estratégia *first improvement* (ou seja, o primeiro movimento que melhora a solução corrente é atualizada), enquanto para  $N_2$  e a  $N_4$ , adota-se o método *best improvement* (analisam-se todas as soluções vizinhas e seleciona-se a melhor).

A primeira heurística híbrida GRASP (G-RC-VND-P) apresentada utiliza IMPP na fase de construção, o método de busca em vizinhança variável (VND) na fase de busca local e a reconexão de caminhos (RC) definida a seguir. A segunda heurística híbrida utilizando o GRASP (G-RC-VND-nP) é constituída de todos os módulos do (G-RC-VND-P), mas substituindo a IMPP por IMPnP na fase de construção. Mestria, Ochi & Martins (2009b) mostram que o GRASP tradicional obtém resultados inferiores para o PCVG quando comparado às heurísticas GRASP que incorporam memória adaptativa utilizando RC. Uma das desvantagens do GRASP tradicional é a ausência de memória entre suas iterações. A cada iteração a solução nova obtida não utiliza nenhuma informação das soluções encontradas nas iterações anteriores. As heurísticas híbridas (G-RC-VND-P e G-RC-VND-nP) incorporará memória adaptativa utilizando a RC através de um *pool* de soluções denotado *conjunto elite* (CE) de soluções obtidas durante suas iterações. A RC, proposta originalmente por Glover (1996), consiste basicamente em explorar as soluções intermediárias entre duas soluções extremas de boa qualidade. São definidas duas soluções extremas: origem  $s_0$  e destino  $s_d$ . Para gerar cada solução intermediária, movimentos são selecionados de forma a introduzir na solução corrente (inicialmente  $s_0$ ), atributos presentes na solução destino ( $s_d$ ) (RESENDE *et al.*, 2010).

Uma solução gerada em uma iteração do GRASP é inserida no CE caso seu custo da função objetivo seja menor que o custo da solução de pior qualidade do conjunto elite e apresente uma diferença mínima na sua estrutura para cada solução presente no CE. A RC proposta para o G-RC-VND-P e o G-RC-VND-nP realiza uma busca intensiva, onde a reconexão é feita entre cada par de soluções elite e ativada durante as iterações do GRASP sempre que o CE for totalmente renovado. Caso surjam novas soluções nesta recombinação que atualizem o CE, uma nova etapa será realizada entre as novas soluções obtidas. O processo termina quando o CE não for mais atualizado.

A metaheurística ILS vem se mostrando eficiente para obter soluções aproximadas de boa qualidade para problemas de otimização combinatória (LOURENÇO, MARTIN & STÜTZLE, 2002) e (SUBRAMANIAN *et al.*, 2010). Sua estrutura canônica é formada por quatro módulos principais: construção de solução, busca local, perturbação e critério de aceitação. ILS explora o espaço de soluções de ótimos locais definido a seguir. Seja a solução corrente  $S$  a qual aplica-se 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 espaço continua a ser explorado através da perturbação até que um critério de parada escolhido, geralmente número de iterações, seja encontrado.

A terceira heurística híbrida (ILS-VND-P) apresentada utiliza o ILS com VND e na fase de construção a IMPP e a quarta heurística (ILS-VND-nP) combina ILS, VND e IMPnP. Para as heurísticas híbridas ao PCVG utilizando ILS foram desenvolvidas perturbações sob três formas: a primeira denominada *Pert1* utiliza o movimento *double-bridge* aplicado às arestas que interligam grupos distintos. São selecionadas quatro arestas aleatórias, mas que conectam as extremidades de grupos diferentes. No final desta perturbação a solução contém uma nova sequência de grupos, sem alterar os vértices internos de cada grupo. Este tipo de perturbação *double-bridge* foi utilizado no PCV com sucesso, mas aplicada em todo o ciclo sem restringir quaisquer arestas (MARTIN, OTTO & FELTEN, 1992). A segunda perturbação (*Pert2*) realiza uma troca aleatória entre dois grupos sem alterar os vértices internos dos grupos. A terceira (*Pert3*) realiza o movimento *double-bridge* aplicado aos vértices em um grupo escolhido aleatório. A sequência dos grupos não é alterada, somente a sequência dos vértices do grupo escolhido é modificada. A perturbação que permitiu melhores resultados foi a *Pert3* inserida ao ILS-VND-P e ILS-VND-nP, seguida por *Pert2* e *Pert1*. Cabe ressaltar que a *Pert2* será utilizada, caso não seja possível aplicar a *Pert3*.

Para as heurísticas híbridas ILS-VND-P e ILS-VND-nP, o critério de aceitação (*CritAceit*), construído 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. Um outro critério de aceitação (*CritAceitAnt*) foi desenvolvido anteriormente envolvendo somente as duas soluções  $S$  e  $S''$ . Mas, nos testes preliminares observou-se que o critério de aceitação *CritAceit* mostrou-se com melhor desempenho. Nas heurísticas híbridas propostas utilizando a metaheurística ILS foram verificadas que a partir de um número fixo de iterações, denominado de *iterações internas*, os métodos não conseguem atualizar a melhor solução. Então a partir de um certo número de iterações internas sem melhorar a solução corrente, constroi-se uma solução nova e realizam-se todas as etapas de perturbação, busca local e critério de aceitação. O número de vezes que o algoritmo reinicia a construção de soluções novas, depois de um número máximo de iterações internas (*maxint*), é considerado para o algoritmo como as *iterações externas*. Estabeleceu-se um número máximo de iterações externas (*maxext*) para os algoritmos híbridos ILS-VND-P e ILS-VND-nP.

```

1.  $iterprob \leftarrow 1$ ;
2.  $inialfas( )$ ; {inicializar a distribuição de probabilidades dos alfas}
3.  $S^0 \leftarrow Insercao\_Mais\_Proxima\_X$ ;
4.  $S \leftarrow VND(S^0)$ ;
5.  $S^* \leftarrow S$ ;
6. para  $i=1$  até  $maxext$  faça
7.    $S^0 \leftarrow Insercao\_Mais\_Proxima\_X$ ;
8.    $S \leftarrow VND(S^0)$ ;
9.   para  $i=1$  to  $maxint$  faça
10.     $S' \leftarrow Pert3(S)$ ;
11.     $S'' \leftarrow VND(S')$ ;
12.     $S \leftarrow CritAceit(S, S'', S^*)$ ;
13.    se  $S$  for melhor do que  $S^*$  então
14.       $S^* \leftarrow S$ ;
15.    fim se
16.  fim para
17.  se  $iterprob = maxprob$  então
18.     $atualalfas( )$ ; {atualizar a distribuição de probabilidade dos alfa}
19.     $iterprob \leftarrow 0$ ;
20.  fim se
21.   $iterprob++$ ;
22. fim para
23. retornar  $S^*$ ;

```

Figura 1 - Pseudocódigo para os algoritmos híbridos com ILS (ILS-VND-X)

O pseudocódigo dos algoritmos híbridos com ILS desenvolvidos neste trabalho para o PCVG são ilustrados na Figura 1 para o caso geral (ILS-VND-X), onde X representa a penalização (P) ou não (nP). No passo 2, a distribuição de probabilidade dos alfas  $inialfas( )$  é obtida e a melhor solução é introduzida no passo 5. Para cada iteração externa do ILS-VND-X a solução é construída no passo 7 pelo procedimento *Inserção\_Mais\_Proxima\_X* (IMPP ou IMPnP) e, no passo 8, a busca local é executada pelo VND. Para cada iteração interna do ILS-VND-X, a perturbação é aplicada no passo 10 e a busca local é executada de novo pelo VND no passo 11. No passo 12 é executado o critério de aceitação e a melhor solução é atualizada no passo 14. No passo 18, a distribuição de probabilidade dos alfas  $atualalfas( )$  é atualizada quando o parâmetro  $iterprob$  atingir o valor máximo igual a  $maxprob$ . Finalmente, no passo 23 a melhor solução é retornada.

#### 4. Resultados Computacionais

Na literatura são encontrados métodos exatos e heurísticos para o PCVG, mas não há registro de nenhuma biblioteca pública contendo instâncias para este tipo de problema, principalmente na versão genérica que não fixa a sequência de visitas aos grupos. Assim, foi necessário criar um conjunto de instâncias para o PCVG genérico para avaliar as heurísticas híbridas propostas neste trabalho. Seis tipos distintos de instâncias foram gerados: (*tipo 1*): Adaptação de instâncias do PCV (TSPLIB95, 2007) agrupadas usando o algoritmo de clusterização *k-means*; (*tipo 2*): Adaptação de instâncias disponíveis na literatura para o PCV geradas por Johnson & McGeoch (2002), agrupando-se os vértices em torno de um centro geométrico; (*tipo 3*): As instâncias são geradas através da interface Concorde de Applegate *et al.* (2007); (*tipo 4*): As instâncias são geradas usando *layout* proposto por Laporte, Potvin & Quilleret (1996); (*tipo 5*): Instâncias semelhantes aos do *tipo 2*, mas geradas com parâmetros diferentes de Johnson & McGeoch (2002); (*tipo 6*): Instâncias do PCV (TSPLIB95, 2007) onde a área plana retangular que compõe a instância é dividida em diversos quadriláteros e cada quadrilátero corresponde a um grupo.

Todas estas instâncias geradas são euclidianas e estão disponíveis para serem utilizadas através do endereço [ftp://ftp.cefetes.br/Teses\\_Dissertacoes/MarioMestria/instances](ftp://ftp.cefetes.br/Teses_Dissertacoes/MarioMestria/instances), onde

constam os valores ótimos e os melhores valores alcançados pela formulação exata ou pelas heurísticas utilizando GRASP ou ILS.

Instâncias	Id.	# nós	# $V_i$	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
						$gap_{li}$ médio	1,36

Tabela 1 - Instâncias com seus identificadores, número de nós, número de grupos, tipo e valores encontrados pelo CPLEX

O desempenho empírico das heurísticas híbridas G-RC-VND-P, G-RC-VND-nP, ILS-VND-P e ILS-VND-nP foram comparados às soluções ótimas ou aos limites inferiores obtidos pelo método exato usando a formulação matemática da literatura proposta em Chisman (1975) e o software CPLEX11.2 (CPLEX, 2009). O CPLEX foi executado num computador com 4 processadores Intel Core 2 Quad, 2.83 GHz com 8 GB de RAM. O sistema operacional utilizado foi Linux Ubuntu versão 4.3.2-1. A implementação das heurísticas híbridas foi codificada em linguagem de programação C e executada no mesmo computador descrito anteriormente, mas utilizando somente um processador. A Tabela 1 mostra as instâncias (coluna 1) com os seus identificadores (coluna 2), seguido do número de vértices (nós), número de grupos e tipo. Mostram-se ainda as soluções e os limites inferiores gerados pelo software CPLEX para as instâncias relacionadas. Na última coluna é mostrado o  $gap_{li}$  (%) 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 (duas horas) para sua execução. Para a instância  $J_{18}$ , a solução ótima foi encontrada pelo CPLEX em 442s. Todas as outras execuções utilizaram 7200s sendo, portanto, abortadas antes de se encontrar ou confirmar uma solução ótima.

Os principais parâmetros utilizados nas heurísticas são descritos a seguir. O valor de penalização dos custos das arestas  $M$  foi de  $10 * \max\{c_{ij}\}$ . A expressão  $\max\{c_{ij}\}$  significa o maior custo entre todos os custos das arestas. Os valores de alfa ( $\alpha$ ) na etapa de construção do GRASP estão no intervalo de  $[0,1]$  utilizados através da estratégia reativa. O número de iterações de G-RC-VND-P e G-RC-VND-nP foi fixado em 200. A cardinalidade do conjunto de soluções elites foi igual a 10 (estabelecido após experimentos preliminares). Para ILS-VND-P e ILS-VND-nP o número de iterações interna ( $maxint$ ) foi de 35, o número máximo de iterações externas ( $maxext$ ) igual a 40 e  $maxprob$  com valor igual a 10. Devido à natureza

aleatória das heurísticas híbridas, estas foram executadas dez (10) vezes para cada instância.

Id.	Melhores Valores				Valores Médios			
	G-P	G-nP	ILS-P	ILS-nP	G-P	G-nP	ILS-P	ILS-nP
$J_1$	<b>0,07</b>	0,08	0,11	0,10	0,28	0,32	<b>0,15</b>	<b>0,15</b>
$J_2$	0,76	0,81	<b>0,73</b>	<b>0,73</b>	1,08	1,64	<b>0,82</b>	0,83
$J_3$	0,70	0,76	0,56	<b>0,46</b>	1,08	1,63	0,79	<b>0,65</b>
$J_4$	1,38	1,47	<b>1,30</b>	1,45	1,58	1,94	<b>1,52</b>	1,81
$J_5$	1,12	1,07	<b>0,88</b>	0,99	1,24	1,60	<b>1,06</b>	1,28
$J_6$	1,48	1,48	1,29	<b>1,24</b>	1,82	1,83	1,46	<b>1,41</b>
$J_7$	0,52	0,53	0,24	<b>0,21</b>	0,83	1,32	0,35	<b>0,31</b>
$J_8$	0,82	0,79	<b>0,33</b>	<b>0,33</b>	1,39	1,53	0,52	<b>0,46</b>
$J_9$	<b>0,42</b>	0,63	0,51	0,43	<b>0,50</b>	0,82	0,55	0,51
$J_{10}$	1,35	1,36	<b>0,89</b>	<b>0,89</b>	2,46	3,45	<b>1,16</b>	1,29
$J_{11}$	1,04	1,04	<b>0,55</b>	1,13	2,97	3,21	<b>0,92</b>	1,85
$J_{12}$	1,83	1,67	<b>1,05</b>	1,17	2,29	2,63	<b>1,30</b>	1,56
$J_{13}$	1,83	1,81	1,43	<b>1,04</b>	2,81	2,70	1,79	<b>1,38</b>
$J_{14}$	1,47	1,33	1,03	<b>1,01</b>	2,18	2,46	1,23	<b>1,18</b>
$J_{15}$	<b>0,43</b>	0,47	0,46	0,52	0,64	1,02	<b>0,54</b>	0,59
$J_{16}$	0,95	1,01	<b>0,80</b>	0,86	1,33	1,49	1,02	<b>0,98</b>
$J_{17}$	0,56	0,57	0,51	<b>0,47</b>	0,74	0,90	0,59	<b>0,54</b>
$J_{18}$	<b>0,01</b>	0,04	0,02	0,04	0,06	0,36	<b>0,04</b>	0,09
$J_{19}$	0,17	0,17	<b>0,12</b>	<b>0,12</b>	0,45	0,48	<b>0,13</b>	0,21
$J_{20}$	<b>0,14</b>	0,15	0,16	0,16	0,17	0,24	<b>0,16</b>	0,18
$gap_{li}$ médio	0,85	0,86	<b>0,65</b>	0,67	1,30	1,58	<b>0,80</b>	0,86

Tabela 2 - Os melhores valores e valores médios obtidos pelas heurísticas híbridas

Os testes computacionais foram estabelecidos com o critério de parada num tempo máximo idêntico para as quatro heurísticas híbridas. Foi utilizado um tempo limite de 720 segundos para as heurísticas, 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 por duas horas, devido à solução ótima ser encontrada antes. A comparação entre os melhores valores obtidos pelas heurísticas híbridas e pelo CPLEX com este critério de parada é mostrada na Tabela 2, assim como os valores médios. Na avaliação entre as melhores soluções obtidas pelas heurísticas híbridas e o CPLEX, compara-se a melhor solução de cada método com os limites inferiores ou com a solução ótima conhecida. Neste contexto o  $gap_{li}$  médio do CPLEX ficou em 1,36%, G-RC-VND-P (G-P) em 0,85%, G-RC-VND-nP (G-nP) em 0,86%, **ILS-VND-P (ILS-P) em 0,65%** e ILS-VND-nP (ILS-nP) em 0,67%. Os valores em negrito na Tabela 2 mostram onde os métodos alcançaram o melhor valor. O CPLEX obteve uma melhor solução no total de 20 instâncias, G-RC-VND-P obteve cinco melhores soluções, ILS-VND-P dez e ILS-VND-nP onze. Na comparação entre as soluções médias obtidas pelas heurísticas híbridas (10 execuções de cada instância), o  $gap_{li}$  de G-RC-VND-P ficou em 1,30%, de G-RC-VND-nP em 1,58%, de **ILS-VND-P em 0,80%** e de ILS-VNDn-P igual a 0,86%. Os valores médios das heurísticas híbridas marcados em negrito representam melhor desempenho.

Uma comparação entre G-RC-VND-P e ILS-VND-P foi realizada estabelecendo um alvo a ser atingido por ambas heurísticas. Cada heurística foi executada 10 vezes para uma determinada instância utilizando-se como critério de parada atingir um determinado valor alvo ou um tempo limite em cada execução. Como as duas heurísticas foram comparadas na Tabela 2 num tempo de duas horas (720 segundos a cada execução), estabelecemos o tempo limite de quatro horas (1440 segundos a cada execução). O alvo a ser atingido foi o melhor valor encontrado entre as duas heurísticas e estão mostrados na Tabela 3.



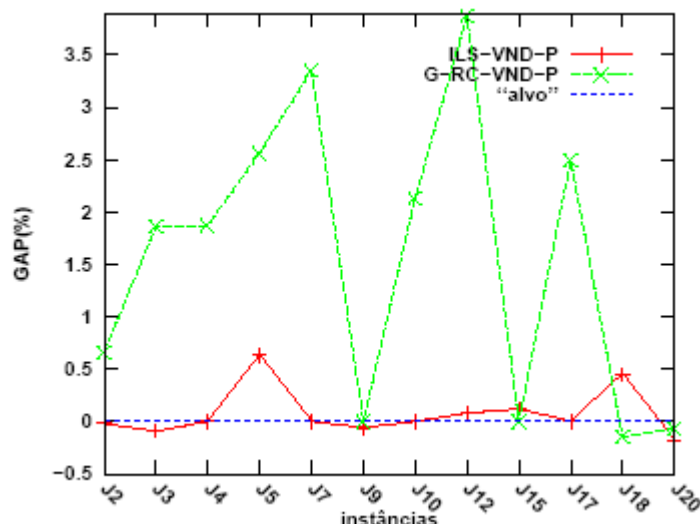


Figura 2 - Gaps de G-RC-VND-P e ILS-VND-P alcançados para encontrar o alvo

A Figura 2 mostra os *gaps* de G-RC-VND-P e ILS-VND-P das soluções encontradas estabelecendo o alvo e o tempo limite. No eixo *x* são mostrados os identificadores das instâncias da Tabela 2 e no eixo *y* o *gap* entre a solução alcançada e a melhor solução (Tabela 3) para ambas as heurísticas híbridas. Valores abaixo e inclusive sobre a linha tracejada com descrição “alvo”, mostrados na Figura 2, significam que as heurísticas híbridas atingiram o alvo. Observa-se que G-RC-VND-P atingiu quatro alvos e **ILS-VND-P atingiu oito alvos**. Os desvios mínimo e máximo de G-RC-VND-P em relação aos alvos foi de -0,15% e 3,87%, respectivamente e de ILS-VND-P iguais a -0,18% (mínimo) e 0,63% (máximo).

Id.	Melhor Valor	Heurística	Tempo (s)	
			ILS-VND-P	G-RC-VND-P
$J_2$	43709	ILS-VND-P	<b>964</b>	14400
$J_3$	55388	ILS-VND-P	<b>1501</b>	14400
$J_4$	12704785	ILS-VND-P	<b>8801</b>	14400
$J_5$	12530603	ILS-VND-P	14400	14400
$J_7$	777	ILS-VND-P	<b>3676</b>	14400
$J_9$	1723	G-RC-VND-P	<b>427</b>	<b>3557</b>
$J_{10}$	10763	ILS-VND-P	<b>139</b>	14400
$J_{12}$	19260	ILS-VND-P	14400	14400
$J_{15}$	13737	G-RC-VND-P	14400	<b>3039</b>
$J_{17}$	16461	ILS-VND-P	<b>2805</b>	14400
$J_{18}$	672	G-RC-VND-P	890	<b>212</b>
$J_{20}$	10673	G-RC-VND-P	<b>858</b>	<b>6424</b>

Tabela 3 - Tempos alcançados por G-RC-VND-P e ILS-VND-P para encontrar o alvo

O tempo limite para a instância  $J_{18}$  foi considerado de 890 segundos (89s a cada execução), pois o CPLEX encontrou a solução ótima em menos de duas horas e a comparação entre G-RC-VND-P e ILS-VND-P estabelecido na Tabela 2 foi realizado num tempo total de 445 segundos para as 10 execuções. A Tabela 3 mostra o tempo atingido para encontrar o alvo de cada instância das heurísticas G-RC-VND-P e ILS-VND-P. Os valores em negrito mostram que as heurísticas alcançaram o alvo antes do tempo limite. Na primeira coluna da Tabela 3 são mostrados os identificadores das instâncias (Id.), na segunda coluna o melhor valor encontrado entre as duas heurísticas, considerado como alvo a ser alcançado, na terceira coluna a heurística que encontrou o melhor valor (critério de parada 720s) e nas últimas duas colunas os tempos alcançados pelas heurísticas expressos em segundos para atingirem os

alvos. Como se pode observar pela Figura 2 e Tabela 3, que o ILS-VND-P encontra mais alvos do que o G-RC-VND-P e quando ambos alcançam o alvo, o método heurístico ILS-VND-P demanda um tempo menor (ver as duas situações nas instâncias  $J_9$  e  $J_{20}$  da Tabela 3). Considerando as Tabelas 2 e 3 e a Figura 2 conclui-se que ILS-VND-P apresenta melhor desempenho do que G-RC-VND-P.

Observando a Tabela 2 verifica-se que as heurísticas híbridas que penalizam as arestas intergrupos obtiveram melhor  $gap_{li}$  médio tanto para os melhores valores como para os valores médios. Novos testes foram efetuados, mostrados na Tabela 4, somente com as heurísticas híbridas que utilizam a IMPP e o método exato CPLEX, restritos às instâncias menores, mas sem limitação de tempo para o CPLEX. Para este conjunto de instâncias, o CPLEX encontrou valores ótimos para todas as instâncias. A heurística G-RC-VND-P foi executada com número de iterações fixo igual a 200 e a heurística ILS-VND-P com os valores das iterações iguais a 35 para  $maxint$  e 40 para  $maxext$ . Na primeira coluna da Tabela 4 encontra-se a instância, na segunda o valor ótimo obtido pelo CPLEX e na terceira o tempo demandado pelo CPLEX em segundos  $t_{CPLEX}(s)$ . Na quarta coluna o  $gap_{hc}$  de G-RC-VND-P, representado por G-P (%), entre a solução alcançada e o valor ótimo e na quinta o tempo médio de execução  $t_{G-P}(s)$  de G-P medido em segundos. Na sexta e sétima colunas são mostradas o  $gap_{hc}$  do melhor valor obtido por ILS-VND-P (ILS-P (%)) e o tempo médio demandado pelo ILS-P em segundos  $t_{ILS-P}(s)$ , respectivamente. Na última coluna o tipo de perturbação realizada pelo ILS-P.

Instâncias	CPLEX	$t_{CPLEX}(s)$	G-P (%)	$t_{G-P}(s)$	ILS-P (%)	$t_{ILS-P}(s)$	perturbação
5-eil51	437	12,31	<b>0,00</b>	1,50	<b>0,00</b>	5,20	Pert3
10-eil51	440	74,38	<b>0,00</b>	1,30	<b>0,00</b>	4,40	Pert2
15-eil51	437	2,04	<b>0,00</b>	1,40	<b>0,00</b>	4,90	Pert2
5-berlin52	7991	201,80	<b>0,00</b>	1,80	<b>0,00</b>	2,90	Pert3
10-berlin52	7896	89,17	<b>0,00</b>	1,60	<b>0,00</b>	2,50	Pert3
15-berlin52	8049	75,93	<b>0,00</b>	1,60	<b>0,00</b>	4,30	Pert3
5-st70	695	13790,11	<b>0,00</b>	3,40	<b>0,00</b>	5,40	Pert3
10-st70	691	4581,00	<b>0,00</b>	2,50	<b>0,00</b>	6,60	Pert3
15-st70	692	883,50	<b>0,00</b>	2,90	<b>0,00</b>	5,40	Pert3
5-eil76	559	83,70	0,54	3,80	<b>0,00</b>	6,70	Pert3
10-eil76	561	254,30	0,71	3,20	0,36	5,50	Pert3
15-eil76	565	49,66	0,53	3,60	<b>0,00</b>	6,90	Pert3
5-pr76	108590	99,29	<b>0,00</b>	5,50	<b>0,00</b>	8,20	Pert3
10-pr76	109538	238,13	<b>0,00</b>	4,00	<b>0,00</b>	8,40	Pert3
15-pr76	110678	261,94	0,15	4,40	0,49	10,00	Pert3
10-rat99	1238	650,67	<b>0,00</b>	9,70	<b>0,00</b>	13,40	Pert3
25-rat99	1269	351,15	<b>0,00</b>	9,80	<b>0,00</b>	23,80	Pert2
50-rat99	1249	2797,58	0,80	10,50	<b>0,00</b>	23,40	Pert2
25-kroA100	21917	3513,57	<b>0,00</b>	8,60	<b>0,00</b>	18,40	Pert2
50-kroA100	21453	947,55	<b>0,00</b>	9,60	<b>0,00</b>	19,90	Pert2
10-kroB100	22440	4991,44	<b>0,00</b>	9,30	0,16	9,40	Pert3
50-kroB100	22355	2579,22	0,54	9,50	<b>0,00</b>	17,70	Pert2
25-eil101	663	709,45	1,21	7,80	0,45	21,20	Pert2
50-eil101	644	275,33	1,09	9,10	1,09	21,10	Pert2
25-lin105	14438	6224,55	<b>0,00</b>	10,50	<b>0,00</b>	8,10	Pert3
50-lin105	14379	1577,21	<b>0,00</b>	12,40	<b>0,00</b>	21,50	Pert2
75-lin105	14521	15886,77	0,23	14,10	0,15	21,90	Pert2
valores médios	-	2266,73	0,21	<b>6,05</b>	<b>0,10</b>	11,37	-

Tabela 4 - Comparação do CPLEX com as heurísticas híbridas para instâncias do *tipo 1* de pequeno porte

Os resultados apresentados na Tabela 4 mostram que G-RC-VND-P encontrou soluções ótimas em 18 de um total de 27 instâncias com  $gap_{hc}$  médio dos melhores valores obtidos pelo G-RC-VND-P em relação ao ótimo de 0,21% e *tempo médio total de 6,05s*. O  $gap_{hc}$  médio dos melhores valores alcançados de *ILS-VND-P ficou em 0,10%* com o tempo médio total de 11,37s. O método híbrido *ILS-VND-P encontrou 21 soluções ótimas*. Estes resultados reforçam o potencial das heurísticas híbridas em encontrar soluções de boa qualidade em tempo computacional viável. A razão para executar a perturbação *Pert2* em algumas instâncias deve-se ao fato que estas não têm vértices suficientes nos grupos para realizar a perturbação *Pert3*, pois se devem ter oito vértices no mínimo em um grupo para realizá-la. A *Pert2* foi realizada em 11 instâncias do total de 27. A necessidade do algoritmo realizar a *Pert2* torna-se importante mesmo em instâncias de grande porte contendo muitos grupos e não tendo vértices o suficiente em cada grupo para realizar a perturbação *Pert3*. Isto pode acontecer em situações práticas e reais.

Testes foram realizados com a heurística ILS-VND-nP utilizando as instâncias da Tabela 4. Este método conseguiu alcançar 17 soluções ótimas com  $gap_{hc}$  médio dos melhores valores igual a 0,27% e tempo médio total igual a 12,05s. O tipo de perturbação utilizada a cada instância para a heurística ILS-VND-nP foi idêntica a da Tabela 4. Observa-se que o tempo médio de 12,05s do ILS-VND-nP foi um pouco maior do que o tempo médio de 11,37s do ILS-VND-P. Isto ocorre porque em IMPnP, a Lista de Candidatos que auxilia a elaboração da solução parcial considera em cada etapa todos os vértices pertencentes ao mesmo grupo dos vértices já contidos na solução parcial. Em oposição, IMPP considera somente os  $k=10$  vértices mais próximos independente que pertençam ou não ao mesmo grupo dos vértices da solução parcial já construída. Para as instâncias da Tabela 4, o  $gap_{hc}$  médio de 0,27% de ILS-VND-nP foi maior em relação a ILS-VND-P (0,11%) e ultrapassou o  $gap_{hc}$  médio de G-RC-VND-P com 0,21%. Pelos resultados apresentados constata-se que as heurísticas híbridas para o PCVG com o método construtivo IMPP permitem alcançar resultados melhores do que ao utilizar o construtivo IMPnP. Neste sentido, tratar o PCVG transformando-o em PCV é uma alternativa eficaz. Cabe ressaltar que várias instâncias do *tipo 1* e do *tipo 3* alcançaram melhores resultados utilizando o construtivo IMPnP. Nas instâncias do *tipo 1* e do *tipo 3* os vértices estão agrupados bem próximo uns dos outros e relativamente distantes dos vértices de grupos distintos. Isto facilita a construção de soluções parciais de qualidade em torno dos vértices de cada grupo, o que torna o IMPnP mais eficaz para estes tipos de instâncias.

## 5. Conclusões

Neste trabalho foi abordado o Problema do Caixeiro Viajante com Grupamentos, uma generalização do Problema do Caixeiro Viajante. Quatro heurísticas híbridas foram desenvolvidas para o PCVG utilizando GRASP e ISL e dois tipos diferentes de algoritmos construtivos, o IMPP e o IMPnP. De nosso conhecimento, esta é a primeira vez que o ILS é utilizado para a solução do PCVG genérico onde não é previamente fixada a sequência de visitas aos grupos. Os métodos híbridos que utilizaram IMPP foram os que obtiveram melhor desempenho quando comparados aos métodos com IMPnP. Observa-se que o uso de uma busca local intensiva baseado no VND é fundamental para o desempenho do G-RC-VND-P e o do ILS-VND-P. Constata-se que as metaheurísticas híbridas alcançam melhor desempenho do que sua forma clássica. Trabalhos futuros incluem utilizar a metaheurística *Variable Neighborhood Search* (VNS) e novas formas híbridas reunindo RC, GRASP e ILS.

## Agradecimentos

*O primeiro autor agradece ao Instituto Federal de Educação, Ciência e Tecnologia do Espírito Santo. Os autores também agradecem ao apoio parcial dos seguintes órgãos de fomento: CNPq (CT-INFO & UNIVERSAL & Bolsa de Produtividade); CAPES (Pró-Engenharia e PROCAD-NF); FAPERJ (PENSA RIO, Prioridade Rio & Cientista do Estado).*

## Referências

- ANILY, S.; BRAMEL, J. & HERTZ, A.** *A 5/3-approximation Algorithm for the Clustered Traveling Salesman Tour and Path Problems*. Operations Res. Letters. Vol. 24 n.1-2, p.29-35, 1999.
- APPLEGATE, D.; BIXBY, R.; CHVÁTAL, V. & COOK, W.** *Concorde TSP Solver*. School of Industrial and Systems Engineering, Georgia Tech, <http://www.tsp.gatech.edu/concorde/index.html>, acesso em 22/12/2007, 2007.
- ARKIN, E. M.; HASSIN, R. & KLEIN, L.** *Restricted Delivery Problems on a Network*. Networks. Vol. 29, n.4, p.205-216, 1997.
- CHISMAN, J. A.** *The Clustered Traveling Salesman Problem*. Computers & Operations Research. Vol. 2, n.2, p.115-119, 1975.
- CHRISTOFIDES, N.** *Worst-case Analysis of a new Heuristic for the Travelling Salesman Problem*. Relatório Técnico, n.388. Graduate School of Industrial Administration, Carnegie-Mellon University, 1976.
- 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. Vol. 12, n.4, p.459-465, 2007.
- GENDREAU, M.; LAPORTE, G. & HERTZ, A.** *An Approximation Algorithm for the Traveling Salesman Problem with Backhauls*. Operational Research. Vol. 45, n.4, p.639-641, 1997
- GENDREAU, M.; LAPORTE, G.; & POTVIN, J. Y.** *Heuristics for the Clustered Traveling Salesman Problem*. Relatório Técnico n.CRT-94-54, Centre de recherche sur les transports, Universidade de Montreal, Montreal, Canadá, 1994.
- GHAZIRI, H. & OSMAN, I. H.** *A Neural Network for the Traveling Salesman Problem with Backhauls*. Computers & Industrial Engineering. Vol. 44, n.2, p.267-281, 2003.
- GLOVER, F.** *Tabu Search and Adaptive Memory Programing - Advances, Applications and Challenges*. In: Barr, R.S., Helgason, R.V., & Kennington, J.L., (eds), Interfaces in Computer Science and Operations Research, Kluwer Academic Publishers, Dordrecht, MA, EUA, p.1-75, 1996.
- GUTTMANN-BECK, N.; HASSIN, R.; KHULLER, S. & RAGHAVACHARI, B.** *Approximation Algorithms with Bounded Performance Guarantees for the Clustered Traveling Salesman Problem*. Algorithmica. Vol. 28, n.4, p.422-437, 2000.
- HANSEN, P. & MLADENOVIC, N.** *Variable Neighborhood Search*. In: Glover, F. W. & Kochenberger, G. A., (eds), Handbook of Metaheuristics, p. 145-184, 2003.
- HERNÁNDEZ-PÉREZ, H.; RODRÍGUEZ-MARTÍN, I. & SALAZAR-GONZÁLEZ, J. J.** *A hybrid GRASP/VND heuristic for the one-commodity pickup-and-delivery traveling salesman problem*. Computers & Operations Research. Vol. 36, n.5, p.1639-1645, 2009.
- HOOGVEEN, J. A.** *Analysis of Christofides' Heuristic: Some Paths are more Difficult than Cycles*. Operations Research Letters. Vol. 10, n.5, p. 291-295, 1991.
- JOHNSON, D. S. & MCGEOCH, L. A.** *Experimental Analysis of Heuristics for the STSP*. In: G. Gutin & A. Punnen (eds), The Traveling Salesman Problem and its Variations, Kluwer Academic Publishers, Dordrecht, Holanda, p. 369-443, 2002.
- JONGENS, K. & VOLGENANT, T.** *The Symmetric Clustered Traveling Salesman Problem*. European Journal of Operational Research. Vol. 19, n.1, p. 68-75, 1985.
- LAPORTE, G. & PALEKAR, U.** *Some Applications of the Clustered Travelling Salesman Problem*. Journal of the Operational Research Society. Vol. 53, n.9, p. 972-976, 2002.
- LAPORTE, G.; POTVIN, J.-Y. & QUILLERET, F.** *A Tabu Search Heuristic using Genetic Diversification for the Clustered Traveling Salesman Problem*. Journal of Heuristics. Vol. 2, n.3, p.187-200, 1996.
- LOKIN, F. C. J.** *Procedures for Travelling Salesman Problems with Additional Constraints*. European Journal of Operational Research, Vol. 3, n.2, p.135-141, 1979.
- 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.

- MARTIN, O. C. & OTTO, S. W.** *Combining Simulated Annealing with Local Search Heuristics*. Annals of Operations Research, Vol. 63 n.1, p.57–75, 1996.
- MARTIN, O.; OTTO, S. W. & FELTEN, E. W.** *Large-Step Markov Chains for the TSP Incorporating Local Search Heuristics*. Operations Research Letters, Vol. 11, n.4, p.219–224, 1992.
- MESTRIA, M.; OCHI, L. S. & MARTINS, S. L.** *Desenvolvimento e Análise Experimental de Heurísticas GRASP para o Problema do Caixeiro Viajante com Grupamentos*. In: IX Congresso Brasileiro de Redes Neurais/Inteligência Computacional (IX CBRN). Ouro Preto, MG: Anais do IX CBRN/Inteligência Computacional, SBRN, RJ (CD-ROM), Vol. 1, ISSN 2177-1200, 2009a.
- MESTRIA, M.; OCHI, L. S. & MARTINS, S. L.** *GRASP com Memória Adaptativa para o Problema do Caixeiro Viajante com Grupamentos*. In: XXIX Encontro Nacional de Engenharia de Produção. Salvador, BA: Anais do XXIX ENEGEP, ABEPRO, RJ (CD-ROM), Vol. 1, 2009b.
- POTVIN, J.-Y. & GUERTIN, F.** *A Genetic Algorithm for the Clustered Traveling Salesman Problem with a Priori Order on the Clusters*. Relatório Técnico n.CRT-95-06. Centre de recherche sur les transports, Universidade de Montreal, Montreal, Canadá, 1995.
- POTVIN, J.-Y. & GUERTIN, F.** *The Clustered Traveling Salesman Problem: A Genetic Approach*. In: I. H. Osman & J. Kelly (eds), Metaheuristics: Theory & Applications, Kluwer Academic Publishers, Norwell, MA, EUA, cap. 37, p.619-631, 1996.
- RESENDE, M. & RIBEIRO, C.** *Greedy Randomized Adaptive Search Procedures*. In: F. Glover & G. Kochenberger (eds), Handbook of Metaheuristics, Kluwer Academic Publishers, p. 219-249, 2002.
- RESENDE, M. G.; MARTÍ, R.; GALLEGO, M. & DUARTE, A.** *GRASP and Path Relinking for the Max-Min Diversity Problem*. Computers & Operations Research. Vol. 37, n.3, p. 498-508, 2010.
- SUBRAMANIAN, A.; DRUMMOND, L. M. A.; BENTES, C.; OCHI, L. S. & FARIAS, R.** *A Parallel heuristic for the Vehicle Routing Problem with Simultaneous Pickup and Delivery*. Computers & Operations Research. Vol. 37, p.1899-1911, 2010.
- TSPLIB95.** *Traveling Salesman Problem*. Gerhard Reinelt, Universität Heidelberg, Institut für Informatik, Heidelberg, Alemanha, <http://www.iwr.uniheidelberg.de/groups/comopt/software/TSPLIB95/index>, acesso em 17/09/2007.
- 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. Vol. 50, p. 690-696, 1999.