

Um Método Exato para o Problema do Caixeiro Viajante com Grupamentos Euclidiano e Simétrico

Mário Mestria,

Coordenadoria de Engenharia Elétrica, Campus Vitória, IFES
Avenida Vitória, 1729 - Bairro Jucutuquara
29040-780, Vitória, ES
E-mail: mmestria@ifes.edu.br, mmestria@uol.com.br.

Resumo: *Nesse artigo, é proposto um método exato 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. A formulação para o método inclui a restrição de Chisman (1975) e uma formulação da literatura. Os testes foram realizados, através do software CPLEX Paralelo, em tipos de instâncias com diversas granularidades, com vértices e grupos de tamanhos variáveis. Resultados computacionais mostraram que o método encontrou diversas soluções ótimas e bons limites inferiores com tempo computacional relativamente baixo.*

Palavras-chave: *Método Exato, Formulação Matemática, Otimização Combinatória*

1 Introdução

O PCVG foi proposto por [2] para resolver um problema real de roteamento automático em sistemas de armazenagem. Neste exemplo particular de sistemas de armazenagem, as ordens de despacho de produtos devem ser respeitadas e estas ordens contêm várias subordens, mais conhecidas como *bills-of-lading*. Este tipo de logística foi modelado como um PCVG, onde os vértices representam as localizações dos estoques dos produtos, as arestas representam as distâncias a serem percorridas pelo veículo motorizado e os grupos representam as *bills-of-lading*.

Limites inferiores utilizando Relaxação Lagrangeana foram desenvolvidos em [7] para o PCVG. O método para obtenção do limite inferior baseou-se na árvore geradora mínima desenvolvida primeiro para o PCV. Foram criados vários conjuntos de instâncias variando o número de vértices de 80 a 150 com diferentes números de grupos. Os valores médios dos limites inferiores alcançaram *gap* igual a 0,35%.

Em [8] uma Busca Tabu combinada com uma fase de diversificação usando um Algoritmo Genético foi proposta para o PCVG, mas a ordem de visita dos grupos já era definida *a priori*. O algoritmo parte de um conjunto de soluções geradas aleatoriamente. A cada solução individual gerada aplica-se a Busca Tabu. Um Algoritmo Genético (AG) básico (sem componentes que realizam uma busca mais intensiva, tais como: reconexão de caminhos e/ou *Variable Neighborhood Descent*) foi proposto para o PCVG por [12]. O AG utilizou na reprodução o operador de seleção de classificação linear e a Recombinação de Arestas. O algoritmo tratou de forma independente o roteamento intergrupos e intragrupos, dando prioridade a solução intergrupos e em seguida a solução intragrupos.

No artigo apresentado por [5] um outro AG, também básico, 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 ordem de visita de cada grupo também é aleatória. Os algoritmos α -aproximados para o PCVG com diferentes variantes são encontrados em [1] e [6]. A maioria destes algoritmos necessita estabelecer em cada grupo os vértices de entrada e de saída e uma ordem pré-definida de visita dos grupos.

O objetivo deste artigo é propor um método exato para solucionar o PCVG na sua forma mais genérica, onde a ordem de visita dos grupos fica a cargo do próprio algoritmo exato. A estrutura do artigo é formada a seguir. A segunda seção mostra a definição formal para o PCVG, a terceira seção descreve a formulação proposta para o PCVG e a quarta seção mostra os resultados computacionais, comparações e discussões. Na última seção são mostradas as conclusões.

2 Definição do Problema

O Problema do Caixeiro Viajante com Grupamentos (PCVG) é uma extensão do Problema do Caixeiro Viajante (PCV), um dos problemas mais estudados na área de otimização combinatória. No caso particular do PCVG ter somente um grupo ou analogamente um vértice em cada grupo, o PCVG torna-se PCV. Neste sentido, o PCVG é também \mathcal{NP} -difícil [6]. Uma descrição do PCVG na estrutura de um grafo é dado a seguir: seja $G=(V, A)$ um grafo completo, direcionado, simétrico e ponderado com um conjunto de vértices $V = \{v_1, v_2, \dots, v_n\}$ e um conjunto de arcos $A = \{(v_i, v_j) : v_i, v_j \in V, i \neq j\}$ em ambas as direções com mesmo peso. O conjunto de vértices V é particionado em m grupos V_1, V_2, \dots, V_m , onde: $V = \bigcup_{i=1}^m V_i, \forall i$ e $V_i \cap V_j = \emptyset$ para todos i e $j, i \neq j$. Assumindo que um custo não-negativo c_{ij} é associado com o arco $(v_i, v_j) \in A$, o PCVG consiste em determinar um ciclo hamiltoniano de custo mínimo em G , tal que os vértices de cada grupo são visitados contiguamente, denominado de PCVG não-orientado.

O custo c_{ij} é a distância euclidiana que corresponde, neste trabalho, a distância geométrica d_{ij} entre dois vértices v_i e v_j no plano bidimensional. Neste contexto, instâncias euclidianas significam que quando os métodos utilizam estas instâncias para resolução do PCVG fazem o uso das distâncias euclidianas. Visto que as instâncias abordadas neste trabalho são euclidianas e simétricas, a rigor temos o Problema do Caixeiro Viajante com Grupamentos Euclidiano e Simétrico.

3 Formulação Matemática Proposta

Diversas formulações do Problema do Caixeiro Viajante (PCV) podem ser encontradas em [11]. Somente duas destas formulações (a de [4] e a de [10]) podem ser adaptadas para o Problema do Caixeiro Viajante com Grupamento (PCVG) incorporando as restrições de [2]. Estas restrições mantêm os vértices dos grupos a serem visitados de uma forma contígua, que é justamente uma das definições para o PCVG. As duas formulações são denominadas aqui de Dantzig/Chisman e Miller/Chisman (descrita a seguir).

Na formulação de Dantzig/Chisman suas restrições crescem exponencialmente na proporção de 2^n , enquanto que na formulação de Miller/Chisman suas restrições crescem na proporção polinomial (n^2). Na formulação Dantzig/Chisman demanda um tempo maior para encontrar a solução ótima do que a formulação de Miller/Chisman. Neste trabalho, nos testes computacionais foi observado que usando a formulação Dantzig/Chisman o tempo passa de menos de um segundo para 351 segundos, quando varia-se o número de vértices da instância de 10 para 20. Já na formulação de Miller/Chisman, o tempo para encontrar a solução ótima é sempre menor que um segundo para este mesmo conjunto de instâncias. Ambas as formulações foram executadas e implementadas num mesmo ambiente de *hardware* e *software* descritos na seção 4.

A partir de instâncias com número de vértices maior que 20 não foi possível encontrar nenhuma solução no modelo de Dantzig/Chisman, devido não ser possível armazenar o problema na memória utilizando esta formulação. Desta forma, o método exato proposto utiliza a formulação de Miller/Chisman para encontrar soluções ao PCVG.

A seguir é mostrado a formulação Miller/Chisman. Esta formulação para o PCVG utiliza programação inteira 0-1. Seja um caixeiro viajante partindo da origem da cidade 1 sem perda de generalidade. A função objetivo 1 requer que se minimize o custo total percorrido pelo caixeiro viajante. As restrições 2 garantem que de cada cidade i só se pode sair para uma única cidade

j . As restrições 3 significam que para cada cidade j há uma única origem em i . Estes dois conjuntos de restrições implicam que somente duas arestas podem ser conectadas a cada vértice formando um conjunto de ciclos possivelmente disjuntos. As restrições 4 garantem não haver subciclos desconexos da origem na solução.

$$\text{Min} \quad z = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad (1)$$

s.a.

$$\sum_{j=1}^n x_{ij} = 1, \quad \forall i \in V \quad (2)$$

$$\sum_{i=1}^n x_{ij} = 1, \quad \forall j \in V \quad (3)$$

$$u_i - u_j + (n - 1)x_{ij} \leq (n - 2), \quad 2 \leq i \neq j \leq n \quad (4)$$

$$\sum_{i \in V_k} \sum_{j \in V_k} x_{ij} = |V_k| - 1, \quad \forall V_k \subset V, |V_k| \geq 1, k = 1, \dots, m \quad (5)$$

$$u_i \geq 0 \quad 2 \leq i \leq n \quad (6)$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \in V \quad (7)$$

As restrições 5 indicam que a soma das variáveis x_{ij} na partição do grupo V_k , $k = 1, \dots, m$, sendo m o número total de grupos, alcançam o valor da cardinalidade $|V_k|$ deste grupo menos 1, forçando em conjunto com as restrições 2 e 3, a formação de um caminho dentro deste grupo. As variáveis binárias x_{ij} assumem o valor igual a 1 se a cidade i for conectada a cidade j na solução. Caso contrário, será igual a 0.

As restrições 4 garantem não haver subciclo na solução e que o caixeiro viajante parte da cidade de número 1 (vértice origem). As variáveis u_i , $i = 2, \dots, n$ representam a sequência das cidades i a serem visitadas tendo a relação $u_j - u_k \leq -1, \forall j, k \in V$ sendo k e j cidades contíguas na solução, $x_{jk} = 1$. Esta formulação envolve $n^2 - n + 2 + m$ restrições e n^2 variáveis inteiras e $n - 1$ variáveis contínuas.

Também é proposta a formulação 1 a 4 mais as restrições 6 e 7 com as arestas intergrupos penalizadas, denominada de Miller/Pen, para confrontar com a formulação Miller/Chisman.

4 Resultados Computacionais utilizando a Formulação Matemática

Nesta seção são descritos os resultados dos testes computacionais efetuados para a solução do PCVG. Serão mostrados resultados utilizando a formulação matemática da literatura incorporando as restrições de [2] e a formulação Miller/Pen.

Para avaliar e validar o método exato foram utilizadas as instâncias euclidianas de [9], disponíveis através de acesso em <http://labic.ic.uff.br/Instance/index.php>. Estes conjunto de instâncias são aplicadas ao PCVG, em sua versão genérica.

Estas seis formas diferentes de instâncias contidas neste conjunto, sem fixação da ordem de visitas aos grupos para o PCVG, são instâncias euclidianas e simétricas. Os resultados computacionais apresentados, a seguir, foram obtidos para este conjunto de instâncias para verificar o desempenho do método exato para instâncias de pequeno, médio e grande porte. Foram utilizadas as formulações matemáticas proposta ao PCVG e o *software* CPLEX [3] com a finalidade de encontrar soluções ótimas para maior parte das instâncias consideradas.

Para as instâncias mostradas, a seguir, o CPLEX Paralelo, versão 11.2, foi executado num computador com 4 núcleos Intel Core 2 Quad, 2.83 GHz com 8 GB de RAM num sistema operacional Linux Ubuntu versão 4.3.2-1. A partir de agora o CPLEX Paralelo, versão 11.2, será mencionado simplesmente por CPLEX. Os modelos de Miller/Chisman e Miller/Pen possibilitaram encontrar soluções através do CPLEX sem ocorrer estouro de memória. Foram realizados testes computacionais a diversos *tipos* de instâncias utilizando ambos modelos: Miller/Chisman (sem penalizar as arestas intergrupos) e Miller/Pen (com as arestas intergrupos penalizadas). A seguir, é mostrada na Tabela 1 a comparação entre os tempos computacionais usando as formulações dos modelos Miller/Chisman e Miller/Pen demandado pelo CPLEX para diversos

tipos de instâncias. Na primeira coluna da Tabela 1 é mostrado o número de vértices, na segunda o número de grupos, na terceira coluna as instâncias, na quarta coluna o *tipo* de instância e nas duas últimas colunas os tempos computacionais obtidos pelo CPLEX em segundos com a formulação sem penalizar as arestas inter-grupos e com a formulação que penaliza as arestas inter-grupos, respectivamente. Os valores alcançados pelo CPLEX na função objetivo foram iguais em ambas formulações para todas as instâncias e estão mostrados nas Tabelas 2, 3 e 4.

#nós	#V _i	Instâncias	Tipo	Tempo sem Pen.(s)	Tempo com Pen.(s)
51	5	5-eil51	1	12	13
51	10	10-eil51	1	74	70
51	15	15-eil51	1	2	1
52	5	5-berlin52	1	202	152
52	10	10-berlin52	1	89	56
52	15	15-berlin52	1	76	91
70	15	15-st70	1	884	605
76	5	5-eil76	1	84	91
76	10	10-eil76	1	254	188
76	15	15-eil76	1	50	35
76	5	5-pr76	1	99	301
76	10	10-pr76	1	238	286
76	15	15-pr76	1	262	163
99	10	10-rat99	1	651	800
99	25	25-rat99	1	351	343
99	50	50-rat99	1	2798	2305
100	25	25-kroA100	1	3514	367
100	50	50-kroA100	1	948	543
100	50	50-kroB100	1	2579	704
101	25	25-eil101	1	709	150
101	50	50-eil101	1	275	357
105	50	50-lin105	1	1577	2801
tempo médio (<i>tipo 1</i>) (s)				714,91	473,73
30	5	i-30-5-17	5	1	1
30	7	i-30-7-17	5	3	2
30	10	i-30-10-17	5	1	1
45	5	i-45-5-18	5	15	8
45	7	i-45-7-18	5	81	115
45	10	i-45-10-18	5	60	99
60	5	i-60-5-21	5	527	829
60	7	i-60-7-21	5	711	176
60	10	i-60-10-21	5	643	1220
65	5	i-65-5-21	5	1112	535
65	10	i-65-10-21	5	118	143
70	5	i-70-5-21	5	1994	250
70	10	i-70-10-21	5	701	2446
tempo médio (<i>tipo 5</i>) (s)				459,00	448,08
51	4	4-eil51-2x2	6	8	8
51	9	9-eil51-3x3	6	8	8
51	12	12-eil51-3x4	6	1	1
51	16	16-eil51-4x4	6	9	11
51	20	20-eil51-4x5	6	9	14
51	25	25-eil51-5x5	6	1	5
52	4	4-berlin52-2x2	6	226	210
52	6	6-berlin52-2x3	6	204	329
52	8	8-berlin52-2x4	6	343	488
52	10	10-berlin52-2x5	6	840	375
70	12	12-st70-3x4	6	2960	2167
70	20	20-st70-4x5	6	545	1733
76	4	4-eil76-2x2	6	110	280
76	9	9-eil76-3x3	6	204	165
76	12	12-eil76-3x4	6	65	45
76	16	16-eil76-4x4	6	26	35
76	20	20-eil76-4x5	6	48	28
76	25	25-eil76-5x5	6	28	27
76	4	4-pr76-2x2	6	3482	1751
76	9	9-pr76-3x3	6	197	200
76	12	12-pr76-3x4	6	34	42
76	15	15-pr76-3x5	6	430	3465
76	18	18-pr76-3x6	6	1019	931
99	25	25-rat99-5x5	6	1000	992
99	42	42-rat99-6x7	6	2521	2048
100	30	30-kroB100-5x6	6	368	561
101	9	9-eil101-3x3	6	1441	703
101	36	36-eil101-6x6	6	522	858
tempo médio (<i>tipo 6</i>) (s)				594,61	624,29
tempo médio total (s)				608,64	535,35

Tabela 1: Tempos alcançados pelo CPLEX.

Das 22 instâncias do *tipo 1*, 14 obtiveram tempos computacionais menores quando utilizam a formulação que penaliza as arestas inter-grupos, comparados aos tempos computacionais na formulação que não penaliza. O tempo computacional médio quando não se penalizam as arestas ficou em 714,91s e quando se penalizam em **473,73s**. Para as instâncias do *tipo 5*, seis obtiveram

tempos computacionais menores quando utilizam a formulação que não penaliza as arestas em um total de 13, comparados aos tempos computacionais na formulação que penaliza. Dois tempos computacionais foram iguais em ambas formulações. O tempo computacional médio quando não se penalizam as arestas ficou em 459,00s e quando se penalizam em **448,08s**. Os tempos médios para as instâncias do *tipo 5* foram bem próximos.

#nós	# V_i	Instância	Valor	gap (%)	status
30	5	i-30-5-17	5193	0,01	sol. ótima
30	7	i-30-7-17	7940	0,01	sol. ótima
30	10	i-30-10-17	6772	0,00	sol. ótima
45	5	i-45-5-18	6879	0,01	sol. ótima
45	7	i-45-7-18	7478	0,01	sol. ótima
45	10	i-45-10-18	8258	0,01	sol. ótima
60	5	i-60-5-21	7824	0,01	sol. ótima
60	7	i-60-7-21	8623	0,01	sol. ótima
60	10	i-60-10-21	9317	0,01	sol. ótima
65	5	i-65-5-21	7680	0,01	sol. ótima
65	7	i-65-7-21	8732	0,10	sol. encontrada
65	10	i-65-10-21	9740	0,01	sol. ótima
70	5	i-70-5-21	7793	0,01	sol. ótima
70	7	i-70-7-21	9013	0,07	sol. encontrada
70	10	i-70-10-21	9746	0,01	sol. ótima
75	5	i-75-5-22	7327	1,06	sol. encontrada
75	10	i-75-10-22	9278	0,40	sol. encontrada
90	5	i-90-5-33	6678	0,58	sol. encontrada
90	10	i-90-10-33	8727	0,31	sol. encontrada
120	5	i-120-5-46	7890	1,23	sol. encontrada
120	10	i-120-10-46	9734	0,21	sol. encontrada

Tabela 2: Valores da função objetivo para as instâncias do *tipo 5*

Nas instâncias do *tipo 6*, de um total de 28, 13 instâncias obtiveram tempos computacionais menores ao utilizar a formulação que não penaliza as arestas inter-grupos. Em três instâncias os tempos computacionais foram iguais. O tempo computacional médio quando não se penalizam as arestas ficou em **594,61s** e quando se penalizam em 624,29s. Observa-se que para as instâncias do *tipo 6* a penalização aplicada nas arestas inter-grupos não é uma estratégia promissora. Considerando todas as instâncias da Tabela 1, o tempo computacional médio total ficou em 608,64s ao utilizar a formulação que não penaliza as arestas inter-grupos e de **535,35s** quando se penalizam.

#nós	# V_i	Instâncias	Valor
51	5	5-eil51	437
51	10	10-eil51	440
51	15	15-eil51	437
52	5	5-berlin52	7991
52	10	10-berlin52	7896
52	15	15-berlin52	8049
70	5	5-st70	695
70	10	10-st70	691
70	15	15-st70	692
76	5	5-eil76	559
76	10	10-eil76	561
76	15	15-eil76	565
76	5	5-pr76	108590
76	10	10-pr76	109538
76	15	15-pr76	110678
99	10	10-rat99	1238
99	25	25-rat99	1269
99	50	50-rat99	1249
100	25	25-kroA100	21917
100	50	50-kroA100	21453
100	10	10-kroB100	22440
100	50	50-kroB100	22355
101	25	25-eil101	663
101	50	50-eil101	644
105	25	25-lin105	14438
105	50	50-lin105	14379
105	75	75-lin105	14521

Tabela 3: Valores da função objetivo para as instâncias do *tipo 1*

Na Tabela 2 apresentam-se os resultados obtidos para as instâncias do *tipo 5* geradas variando o número de vértices de 30 a 120 e o número de grupos de 5 a 10 vértices. Na Tabela 2 é mostrado na primeira coluna o numero de vértices, na segunda o número de grupos, na terceira a instância, na quarta o valor alcançado pelo CPLEX, na quinta o percentual do *gap* calculado pelo CPLEX e na última coluna o *status* da solução. Para a instância i-65-7-21 com 65 vértices,

para a instância i-70-7-21 com 70 vértices e todas as instâncias contendo 75, 90 e 120 vértices não foram encontradas soluções ótimas. Para as demais, o CPLEX alcançou soluções ótimas. O tempo máximo estabelecido ao CPLEX para encontrar soluções ótimas ou não, foi de 7200 segundos. Na Tabela 3 apresentam-se as instâncias do *tipo 1* com o número de vértices na faixa 51 a 105 e número de grupos variando entre cinco e 75 vértices. Na primeira coluna da Tabela 3 é mostrado o número de vértices, na segunda coluna o número de grupos, na terceira as instâncias e na quarta o valor obtido pelo CPLEX. Para todas instâncias o CPLEX atingiu soluções ótimas.

#nós	# V_i	Instâncias	Valor
51	4	4-eil51-2x2	429
51	9	9-eil51-3x3	445
51	12	12-eil51-3x4	441
51	16	16-eil51-4x4	438
51	20	20-eil51-4x5	441
51	25	25-eil51-5x5	437
52	4	4-berlin52-2x2	8232
52	6	6-berlin52-2x3	8236
52	8	8-berlin52-2x4	8412
52	10	10-berlin52-2x5	9267
70	6	6-st70-2x3	706
70	9	9-st70-3x3	703
70	12	12-st70-3x4	701
70	16	16-st70-4x4	714
70	20	20-st70-4x5	718
76	4	4-eil76-2x2	557
76	9	9-eil76-3x3	560
76	12	12-eil76-3x4	562
76	16	16-eil76-4x4	562
76	20	20-eil76-4x5	562
76	25	25-eil76-5x5	557
76	4	4-pr76-2x2	112071
76	6	6-pr76-2x3	112573
76	9	9-pr76-3x3	109123
76	12	12-pr76-3x4	109164
76	15	15-pr76-3x5	115960
76	18	18-pr76-3x6	117406
99	25	25-rat99-5x5	1377
99	42	42-rat99-6x7	1318
100	28	28-kroA100-4x7	25372
100	25	25-kroB100-5x5	24626
100	30	30-kroB100-5x6	25244
101	9	9-eil101-3x3	659
101	36	36-eil101-6x6	658
105	2	2-lin105-2x1	14627
105	16	16-lin105-4x4	16516

Tabela 4: Valores da função objetivo para as instâncias do *tipo 6*

A Tabela 4 mostra as soluções ótimas alcançadas pelo CPLEX para as instâncias do *tipo 6*. O número de vértices mostrado na primeira coluna da Tabela 4 variaram de 51 a 105 vértices e o número de grupos (segunda coluna) entre quatro e 42 vértices. Na terceira coluna desta tabela são mostradas as instâncias e na quarta coluna o valor obtido pelo CPLEX.

Instâncias	Id.	#nós	# V_i	Tipo	CPLEX		
					Valor	Lim. Inf.	(%)
i-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	0,01
42-a280	J_{19}	280	42	6	130043	129560,53	0,37
144-rat783	J_{20}	783	144	6	916174	913715,52	0,27
<i>gap_{li}</i> médio =							1,36

Tabela 5: Instâncias com seus identificadores (J_k), número de nós, número de grupos, tipo e valores alcançados pelo CPLEX com limite de duas horas.

Os testes computacionais, usando a formulação Miller/Chisman e Miller/Pen, realizados nas instâncias do *tipo 1*, *5* e *6* alcançaram valores ótimos num total de 76 instâncias. Somente em

oito casos as instâncias não atingiram a solução ótima dentro do tempo estabelecido. Toda estas instâncias (Tabelas 2, 3 e 4) são de pequeno porte com até 120 vértices. Para os outros tipos de instâncias não foram realizados testes computacionais utilizando estas formulações, pois as instâncias do *tipo 2, 3 e 4* contém acima de 1000, 300 e 200 vértices, respectivamente. Isto inviabiliza encontrar soluções ótimas para estes últimos tipos de instâncias usando o *software* CPLEX executado no *hardware* descrito acima. Para as instâncias do *tipo 2, 3 e 4* foram realizados testes computacionais através do CPLEX utilizando-se o modelo de Miller/Pen para se encontrar bons limites inferiores para estas instâncias, com número maior e igual a 200 vértices. Estes testes serão mostrados a seguir, na Tabela 5, assim como os testes computacionais para obter limites inferiores para as instâncias de porte maior do *tipo 1, 5 e 6* contendo número de vértices acima de 100. A Tabela 5 mostra as instâncias (coluna 1) com os seus identificadores (coluna 2), seguido do número de vértices (nós), número de grupos e tipo. Na Tabela 5 são mostradas as soluções, os limites inferiores gerados pelo software CPLEX e na última coluna mostra-se o percentual do gap_i (%) entre valor da solução e limite inferior calculado da seguinte forma:

$$gap_i = 100 * \left(\frac{best - li}{best + \epsilon} \right), \quad (8)$$

onde, $best$ é o melhor valor encontrado pelo CPLEX, li o limite inferior dado pelo CPLEX e ϵ igual 10^{-10} . Devido às longas iterações ocorridas no CPLEX para a maioria das instâncias, um tempo limite de 7200s foi determinado 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.

5 Conclusão

Com o método exato proposto utilizando a formulação de Miller/Chisman foi possível encontrar soluções ótimas para diversas instâncias do PCVG e bons limites inferiores. O tempo computacional despendido pelo método exato foi relativamente baixo usando a formulação penalizada ou não-penalizada de Miller. Observamos que para algumas instâncias a formulação penalizada não é promissora. Concluimos desta forma que o método com formulação de Miller/Chisman para o PCVG tem bom desempenho computacional validando desta forma a metodologia utilizada.

Referências

- [1] ANILY, S.; BRAMEL, J.; HERTZ, A. A 5/3-approximation Algorithm for the Clustered Traveling Salesman Tour and Path Problems. *Operations Research Letters*, v. 24, n. 1-2, p. 29–35, 1999.
- [2] CHISMAN, J. A. The Clustered Traveling Salesman Problem. *Computers & Operations Research*, v. 2, n. 2, p. 115–119, 1975.
- [3] CPLEX. ILOG CPLEX 11.2 User's Manual and Reference Manual, 2009. In: ILOG S.A.: [s.n.], 2009. Disponível em: <<http://www.ilog.com>>. Acesso em: 06/01/2009.
- [4] DANTZIG, G.; FULKERSON, R.; JOHNSON, S. Solution of a Large-Scale Traveling-Salesman Problem. *Journal of the Operations Research Society of America*, v. 2, p. 393–410, 1954.
- [5] DING, C.; CHENG, Y.; HE, M. Two-Level Genetic Algorithm for Clustered Traveling Salesman Problem with Application in Large-Scale TSPs. *Tsinghua Science and Technology*, v. 12, n. 4, p. 459–465, 2007.
- [6] GUTTMANN-BECK, N.; HASSIN, R.; KHULLER, S.; RAGHAVACHARI, B. Approximation Algorithms with Bounded Performance Guarantees for the Clustered Traveling Salesman Problem. *Algorithmica*, v. 28, n. 4, p. 422–437, 2000.
- [7] JONGENS, K.; VOLGENANT, T. The Symmetric Clustered Traveling Salesman Problem. *European Journal of Operational Research*, v. 19, n. 1, p. 68–75, 1985.
- [8] 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.
- [9] MESTRIA, M. Métodos Heurísticos usando GRASP, Reconexão de Caminhos e Busca em Vizinhança Variável para o Problema do Caixeiro Viajante com Grupos. *Revista Produção Online*, Florianópolis, SC, v. 13, n. 3, p. 1002–1033, 2013.
- [10] MILLER, C. E.; TUCKER, A. W.; ZEMLIN, R. A. Integer Programming Formulation of Traveling Salesman Problems. *Journal of the ACM*, ACM Press, Nova York, NY, EUA, v. 7, n. 4, p. 326–329, 1960.
- [11] ORMAN, A.; WILLIAMS, H. *A Survey of Different Integer Programming Formulations of the Travelling Salesman Problem*. Relatório Técnico, n. LSEOR 04.67. Department of Operational Research, London School of Economics and Political Science, Londres, Grã Bretanha, 2004.
- [12] POTVIN, J.-Y.; GUERTIN, F. The Clustered Traveling Salesman Problem: A Genetic Approach. In: OSMAN, I. H.; KELLY, J. (Ed.). *Meta-heuristics: Theory & Applications*. Norwell, MA, EUA: Kluwer Academic Publishers, 1996. cap. 37, p. 619–631.