

PROPOSTA E AVALIAÇÃO DE HEURÍSTICAS GRASP PARA O PROBLEMA DO CAIXEIRO VIAJANTE COM GRUPAMENTOS

Mário Mestria

Coordenadoria da Área de Informática - Campus Colatina - Instituto Federal do Espírito Santo

Avenida Arino Gomes Leal nº 1700 - Bairro Santa Margarida.

Colatina, ES, Brasil, CEP.: 29700-558

mmestria@ifes.edu.br, mmestria@ic.uff.br

Resumo – Neste artigo são propostas heurísticas GRASP com Reconexão de Caminhos para obter soluções no Problema do Caixeiro Viajante com Grupamentos (PCVG), uma generalização do Problema do Caixeiro Viajante (PCV). Neste problema, os vértices são particionados em grupos e todos os vértices de cada grupo têm que ser visitados de forma contígua. São propostas seis heurísticas GRASP para resolver o PCVG incluindo um GRASP tradicional e cinco GRASP com procedimentos de Reconexão de Caminhos. O desempenho dos algoritmos propostos é comparado entre eles e com um algoritmo exato usando o software CPLEX Paralelo. Resultados computacionais mostram que a heurística com Reconexão de Caminhos usando intensificação e fase de pós-otimização supera outras heurísticas no tocante a qualidade das soluções geradas.

Palavras-chave – Inteligência Computacional, Heurísticas, Reconexão de Caminhos, Problema do Caixeiro Viajante com Grupamentos.

Abstract – In this paper are proposed GRASP heuristics with Path Relinking for obtaining solutions in the Clustered Traveling Salesman Problem (CTSP), a generalization of the Traveling Salesman Problem (TSP). In this problem, the vertices are partitioned into clusters and all vertices of each cluster have to be visited contiguously. Six versions of GRASP heuristics are proposed to solve the CTSP including a traditional GRASP and five GRASP with Path Relinking (PR) procedures. The performance of the proposed algorithms is compared among them and then and with an exact algorithm using the Parallel CPLEX software. Computational results show that the heuristic with Path Relinking using intensification and post-optimization phase outperforms other heuristics concerning quality of the solutions.

Keywords – Computational Intelligence, Heuristics, Path Relinking, Clustered Traveling Salesman Problem.

1. INTRODUÇÃO

Uma extensão do Problema do Caixeiro Viajante (PCV), ainda pouco explorada embora aplicável em diferentes situações práticas, é o Problema do Caixeiro Viajante com Grupamentos (PCVG). Esta variante surgiu para resolver inicialmente um problema real envolvendo roteamento automático em sistemas de armazenagem [1].

O PCVG pode ser descrito da seguinte forma: seja $G = (V, A)$ um grafo completo, simétrico e não direcionado com um conjunto de vértices $V = \{v_1, v_2, \dots, v_n\}$ e um conjunto de arcos $A = \{(v_i, v_j) : v_i, v_j \in V, i \neq j\}$ ponderados satisfazendo a desigualdade triangular. O conjunto de vértices V é particionado em subconjuntos disjuntos V_1, V_2, \dots, V_m . Assumindo que um custo não-negativo c_{ij} é associado a cada arco (v_i, v_j) pertencente a A , o PCVG consiste em determinar um ciclo Hamiltoniano de custo mínimo em G , tal que os vértices de cada grupo são visitados contiguamente. Os custos utilizados para o PCVG correspondem às distâncias entre quaisquer vértices (v_i, v_j) . No caso particular onde só existe um subconjunto $V_i = V$, o PCVG se torna o PCV clássico. Desta forma o PCVG pertence à classe *NP-difícil* [2] e limita o uso exclusivo de métodos exatos. Neste sentido, o PCVG será tratado por heurísticas.

Aplicações para o PCVG encontram-se, por exemplo, em roteamento automático em armazéns [1], despacho de veículos [3], desfragmentação de discos rígidos, sistemas de manufaturas [4] e em transações comerciais envolvendo supermercados, lojas e fornecedores de mercadorias [5].

Justificativas para abordar o PCVG incluem o fato de existirem poucos trabalhos na literatura sobre este problema e, além disso, a maioria dos trabalhos partem do princípio que já se conhece uma sequência de visita dos grupos [6–8] e [9], limitando com isso seu uso em situações reais. Neste contexto, este trabalho trata do caso geral onde o PCVG não considera previamente nenhuma sequência de visitas aos grupos e para a sua solução propõem-se diferentes versões utilizando a metaheurística GRASP, que tem se mostrado como uma técnica muito competitiva para obter soluções aproximadas de boa qualidade para problemas de otimização combinatória [10].

O PCVG foi proposto por Chisman [1] para resolver um problema de roteamento automático em sistemas de armazenamento. O autor resolveu o problema, através de um método exato, transformando o PCVG em um PCV adicionando uma penalidade

constante de valor M aos custos dos arcos que ligam quaisquer dois vértices associados com grupos diferentes. Usando um algoritmo *branch-and-bound* resolveu o PCV resultante. Um outro método exato utilizando limites inferiores baseado em Relaxação Lagrangeana foram desenvolvidos em [11] para o PCVG.

Em [8] uma Busca Tabu combinada com uma fase de diversificação usando um Algoritmo Genético foi proposta para o PCVG, mas a sequência de visita dos grupos já era definida *a priori*. Um Algoritmo Genético (AG) tradicional (sem incorporar busca local intensiva) foi proposto para o PCVG em [12]. No trabalho de Ding, Cheng e He [13] um outro AG, também tradicional, foi desenvolvido no qual utiliza critério de visitas dos vértices de cada grupo de forma aleatória sem estabelecer uma relação com as distâncias entre estes vértices. A escolha da ordem de visita de cada grupo também foi aleatória. Na literatura encontram-se vários métodos heurísticos para o PCVG, mas que não incorporam módulos de memória adaptativa.

Observa-se que a inclusão de memória adaptativa [14, 15] e [16] em algoritmos heurísticos têm melhorado seu desempenho. Assim, neste trabalho coloca-se como um dos objetivos verificarem o impacto ao se incluir na metaheurística GRASP, memória adaptativa através de procedimentos de Reconexão de Caminhos (RC) [17].

Neste contexto foram desenvolvidas seis versões utilizando a metaheurística GRASP. Uma versão é um GRASP tradicional (G), sem uso de memória entre suas iterações. Nas demais versões incluem-se o uso de memória através da utilização e atualização dinâmica de um conjunto das k melhores soluções distintas geradas até o momento pelo GRASP, denotado por conjunto elite (CE), cujas informações são aproveitadas por um módulo de RC. A RC é utilizada em quatro versões: na primeira a busca local é mais intensiva onde a reconexão é feita entre cada par de soluções elite (RC1), após as iterações do GRASP. A segunda (RC2) é uma versão menos intensiva onde a reconexão é feita apenas entre a solução da iteração atual do GRASP com uma solução elite, a terceira (RC3) é idêntica a RC1, mas ela é ativada durante as iterações do GRASP sempre que o CE for totalmente renovado. A quarta (RC4) é idêntica a RC3, mas é ativada sempre que o CE conter uma solução nova.

Este artigo é estruturado conforme as seguintes seções. A segunda seção descreve os métodos heurísticos propostos para o PCVG. A terceira seção apresenta os resultados computacionais e as avaliações das heurísticas e na última seção descreve as conclusões.

2. MÉTODOS HEURÍSTICOS PARA O PCVG

Seis heurísticas GRASP foram desenvolvidas para o PCVG. A primeira versão (G) utiliza a estrutura do GRASP tradicional, onde a cada iteração executa-se uma fase de construção de uma solução e uma fase de busca local. Na etapa de construção para todas as heurísticas, utiliza-se o método de Inserção mais Próxima com penalização nas arestas cujos vértices extremos pertençam a subconjuntos V_i distintos.

Apesar dos bons resultados obtidos pela metaheurística GRASP em diferentes problemas de otimização, uma das características deste método na sua forma tradicional (G) é a ausência de memória entre suas iterações. Ou seja, a cada nova iteração uma nova solução é obtida sem utilizar informação das soluções obtidas em iterações anteriores. Neste trabalho utiliza-se memória adaptativa no GRASP através do uso e atualização dinâmica de um conjunto elite (CE), uma técnica de busca denominada Reconexão de Caminhos (RC). A RC, proposta originalmente por Glover [18], consiste basicamente em explorar as soluções intermediárias entre duas soluções extremas de boa qualidade. São definidas duas soluções: origem s_o e destino s_d . Para gerar cada solução intermediária, movimentos são selecionados de forma a introduzir na solução corrente (inicialmente s_o), atributos presentes na solução destino (s_d). Reconexão de Caminhos é um método baseado em população que pode ocasionalmente ser considerado como algoritmo de Computação Evolucionária [19].

A segunda versão (GRC1) introduz ao (G) a RC1 ao final da execução de G na fase de pós-otimização. A terceira versão (GRC2) incorpora ao G a Reconexão de Caminhos (RC2) a cada iteração de G. A RC2 é realizada entre cada solução produzida pela busca local com uma solução extraída do CE. Uma solução gerada em uma iteração 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 ($diff_{min}$) na sua estrutura para cada solução presente no CE. Na quarta heurística (GRC12) introduz a RC2 durante a execução das iterações GRASP e ao final do GRASP ativa-se a RC1. A quinta heurística (GRC3) utiliza todos os módulos do GRC2 substituindo a Reconexão de Caminhos RC2 pela RC3 definida previamente e na sexta heurística (GRC4) a RC3 é substituída pela RC4. Na fase de busca local, para todas as heurísticas GRASP, foi utilizado o método *2-Optimal* desenvolvido originalmente para o PCV.

O parâmetro α para criar a Lista Restrita de Candidatos (LRC), na etapa de construção do GRASP, utiliza a versão reativa, onde seus valores variam no intervalo [0,1]. A estratégia reativa permite um melhor aproveitamento sobre a versão do GRASP com valor fixo de α em termos de robustez e qualidade de solução [10].

3. RESULTADOS COMPUTACIONAIS

Para avaliar os métodos aqui propostos ao PCVG genérico, sem pré-fixação da sequência de grupos a serem visitados, foram utilizadas as instâncias descritas em [20].

A Tabela 1 mostra as instâncias com os seus identificadores, números de vértices, número de grupos e tipo. Para analisar o desempenho empírico das heurísticas propostas, os resultados de G, GRC1, GRC2, GRC12, GRC3 e GRC4 foram comparados, quando possível, com a solução ótima obtida pelo método exato usando a formulação matemática da literatura proposta em [1] e o *software* CPLEX Paralelo versão 11.2 [21]. Devido ao elevado número de iterações ocorridas no CPLEX para a maioria das instâncias, determina-se um tempo limite de 25200 segundos para sua execução. Para a instância I_3 , a solução ótima foi

encontrada pelo CPLEX em 442s. Todas as outras execuções utilizaram o tempo limite de 25200s e neste caso o CPLEX não obteve solução ótima, ou ainda não se tem esta garantia.

Os principais parâmetros utilizados nas heurísticas são descritos a seguir. O valor da penalização dos custos das arestas que ligam vértices pertencentes a grupos distintos foi igual a $10 * \max\{c_{ij}\}$. A expressão $\max\{c_{ij}\}$ significa o maior custo entre todos os custos dos arcos (v_i, v_j) .

Instâncias	Id.	# vértices	# V_i	Tipo	CPLEX		
					Custo	Limite Inferior	(%)
10-lin318	I_1	318	10	1	531931	526559,70	1,01
10-pcb442	I_2	442	10	1	546157	536478,37	1,77
25-eil101	I_3	101	25	6	23671	23668,63	0,01
144-rat783	I_4	783	144	6	916103	913715,52	0,26
300-20-111	I_5	300	20	5	310590	308627,83	0,63
500-25-308	I_6	500	25	5	367509	364114,62	0,92
300-6	I_7	300	6	3	8956	8916,41	0,44
700-20	I_8	700	20	3	41615	41274,00	0,82
C1k.0	I_9	1000	10	2	133638625	131360206,30	1,70
C1k.1	I_{10}	1000	10	2	130563491	128540131,50	1,55
200-4-h	I_{11}	200	4	4	62835	62391,08	0,71
600-8-z	I_{12}	600	8	4	132767	128083,87	3,53
<i>gap</i> médio							1,11

Tabela 1: Instâncias com seus identificadores, número de vértices, número de grupos, tipo e soluções CPLEX.

O número de iterações do GRASP foi fixado em 200 (critério de parada). A cardinalidade do conjunto de soluções elite (CE) foi igual a 10 e estabelecido após experimentos preliminares. A diferença mínima $diff_{min}$ na estrutura entre as duas soluções (a solução candidata e a solução presente no conjunto elite) foi igual a cinco. Devido à natureza aleatória do GRASP, executam-se dez vezes cada heurística aqui proposta para cada instância. Para as instâncias (**Id.**), na Tabela 2, são mostrados os tempos médios medidos em segundos com 200 iterações.

Id.	Tempos Médio (segundos)					
	G	GRC1	GRC2	GRC12	GRC3	GRC4
I_1	57,05	158,87	87,48	184,75	136,80	172,80
I_2	151,01	451,11	227,67	495,25	386,30	485,70
I_3	2,48	3,72	3,95	6,43	3,00	4,80
I_4	750,30	3718,32	1133,96	4132,28	3335,10	3735,10
I_5	51,68	121,30	75,65	142,81	106,00	158,90
I_6	225,93	720,47	331,15	822,49	631,60	860,40
I_7	49,44	99,47	75,97	119,72	86,30	132,80
I_8	593,80	1618,11	881,57	1837,78	1470,90	1870,30
I_9	1762,68	7592,92	2608,65	8749,56	7235,00	9435,00
I_{10}	1765,84	9421,46	2616,59	10297,08	8190,90	12390,00
I_{11}	15,75	33,95	25,07	42,37	29,90	49,30
I_{12}	364,78	1533,07	554,9	1652,09	1325,00	2201,80

Tabela 2: Tempos médios de execução das heurísticas com 200 iterações.

Observa-se que G e GRC2 são aquelas que consomem o menor tempo médio, fato que era esperado, pois estas não utilizam a RC (G) ou utilizam a RC menos intensiva RC2 (GRC2). Apesar de GRC1, GRC12, GRC3 e GRC4 apresentarem tempos médios maiores que G e GRC2, estes tempos ainda são bem menores que os tempos exigidos pelo CPLEX.

Na avaliação entre as melhores soluções obtidas pelas heurísticas GRASP e o CPLEX, compara-se a melhor solução de cada método com a melhor solução conhecida (ótima ou não). Neste contexto o *gap* médio do CPLEX ficou em 1,11% (Tabela 1), isso porque em muitas instâncias o CPLEX não conseguiu convergir para um ótimo no tempo limite estabelecido, G em 1,01%, GRC1 em 0,74%, GRC2 em 0,95%, **GRC12 em 0,72%**, GRC3 em 0,82% e GRC4 em 0,76%. O CPLEX obteve a melhor solução (solução ótima ou não) em três num total de 12 instâncias, as heurísticas GRC1, GRC3 e GRC4 obtiveram duas melhores soluções cada e **GRC12 seis melhores soluções**. Na comparação entre a melhor solução conhecida com as soluções médias obtidas pelas heurísticas (10 execuções de cada instância), o *gap* de G ficou em 1,47%, de GRC1 em 1,13%, de GRC2 em 1,21%, de **GRC12 em 1,04%**, de GRC3 igual a 1,22% e GRC4 em 1,16%.

De acordo com os resultados apresentados anteriormente, onde o critério de parada é um número máximo de iterações, ou seja, utiliza-se o mesmo número para todas as heurísticas, pode-se observar que GRC12 e GRC4 obtiveram os melhores resultados, mas exigiram tempos computacionais maiores do que as outras heurísticas.

Neste contexto, realizam-se novos testes computacionais colocando como critério de parada um tempo máximo idêntico para as seis heurísticas. Coloca-se um tempo limite de 2 horas para o CPLEX e 720 segundos para as seis heurísticas GRASP (pois

as heurísticas são executadas 10 vezes). Somente a execução da instância I_3 não foi limitada por duas horas, devido à solução ótima ter sido encontrada antes pelo CPLEX. A comparação entre os melhores valores obtidos pelas heurísticas e pelo CPLEX com este novo critério de parada é mostrada na Tabela 3. Nesta tabela observa-se que o *gap* médio do CPLEX ficou em 1,36%, de G em 0,98%, de GRC1 em 0,83%, de GRC2 em 0,94%, de **GRC12 em 0,82%**, de GRC3 igual a 0,86% e GRC4 em 0,83%.

A heurística GRC12 obteve novamente o melhor desempenho nesta nova bateria de testes com **oito melhores soluções**, enquanto GRC4 obteve quatro, GRC1 com duas e GRC3 e CPLEX apenas uma. Observa-se desta forma a importância de usar uma RC de forma intensiva ao final do GRASP (GRC1 e GRC12). Além disso, pelos resultados anteriores (com o critério de parada definido pelas iterações) e os da Tabela 3 conclui-se mesmo de forma empírica que a melhor estratégia foi a de utilizar conjuntamente uma Reconexão de Caminhos (RC1) com a RC2 (GRC12). Nesta segunda bateria de testes, observa-se também que apesar de GRC1, GRC12 e GRC4 exigirem mais tempo por iteração, estas necessitam de menos iterações para atingir bons limites superiores de um valor ótimo.

Id.	CPLEX	Melhores Valores					
		G	GRC1	GRC2	GRC12	GRC3	GRC4
I_1	1,54	0,97	0,80	0,93	0,76	0,90	0,84
I_2	1,95	1,12	0,73	0,99	0,66	0,84	0,66
I_3	0,01	0,04	0,04	0,05	0,03	0,03	0,03
I_4	0,27	0,21	0,19	0,21	0,20	0,19	0,19
I_5	0,86	0,55	0,45	0,51	0,43	0,50	0,44
I_6	0,95	0,69	0,61	0,66	0,58	0,60	0,61
I_7	0,60	0,67	0,51	0,60	0,49	0,50	0,50
I_8	0,87	0,67	0,63	0,65	0,64	0,62	0,61
I_9	1,99	1,63	1,61	1,64	1,60	1,63	1,68
I_{10}	1,69	1,28	1,27	1,31	1,27	1,30	1,27
I_{11}	1,87	1,68	1,28	1,60	1,19	1,23	1,24
I_{12}	3,76	2,23	1,80	2,14	1,96	2,01	1,95
<i>gap</i> médio	1,36	0,98	0,83	0,94	0,82	0,86	0,83

Tabela 3: *Gaps* entre os melhores valores obtidos pelas heurísticas e pelo CPLEX com tempo limitado.

A Tabela 4 ilustra o desempenho médio das heurísticas quando utiliza um tempo limite como critério de parada. O *gap* médio do G ficou em 1,51%, de GRC1 em 1,15%, de GRC2 em 1,16%, de **GRC12 em 1,14%**, de GRC3 igual a 1,39% e de GRC4 em 1,35%. Novamente os resultados mostram a superioridade da versão GRC12 em relação às outras. No entanto ressalta-se o bom comportamento médio das seis heurísticas aqui propostas. Finalmente uma nova bateria de testes foi efetuada apenas entre GRC12 e o método exato CPLEX, para 27 instâncias de pequeno porte, mas sem limitação de tempo para o CPLEX. O número de iterações de GRC12 foi de 200. Para todas as instâncias, o CPLEX encontrou soluções ótimas, enquanto GRC12 encontrou 15 soluções ótimas com tempo médio 3,3s. O *gap* médio de GRC12 em relação ao ótimo foi de 0,25%. Este resultado reforça o potencial de GRC12 em encontrar soluções de boa qualidade em tempo computacional viável.

Id.	Valores Médios					
	G	GRC1	GRC2	GRC12	GRC3	GRC4
I_1	1,78	1,18	1,21	1,18	1,72	1,65
I_2	1,93	1,24	1,36	1,22	1,67	1,52
I_3	0,35	0,22	0,14	0,18	0,35	0,33
I_4	0,28	0,23	0,25	0,24	0,24	0,24
I_5	0,85	0,64	0,63	0,63	0,84	0,78
I_6	0,92	0,73	0,78	0,73	0,80	0,76
I_7	1,09	0,82	0,78	0,78	1,08	1,04
I_8	0,82	0,71	0,75	0,72	0,73	0,72
I_9	1,88	1,76	1,79	1,76	1,80	1,81
I_{10}	1,54	1,44	1,47	1,42	1,45	1,46
I_{11}	3,30	2,37	2,01	2,30	3,38	3,31
I_{12}	3,32	2,43	2,70	2,54	2,66	2,63
<i>gap</i> médio	1,51	1,15	1,16	1,14	1,39	1,35

Tabela 4: *Gaps* entre os valores médios obtidos pelas heurísticas com tempo limitado.

Análises gráficas das distribuições de probabilidade acumulativa foram realizadas para verificar o desempenho da introdução de módulos de Reconexão de Caminhos [22]. Analisa-se o desempenho de G com GRC12 e GRC4, porque estas duas últimas heurísticas obtiveram melhor desempenho. Para realizar a distribuição de probabilidade foi escolhida a instância I_{10} (instância com maior número de vértices) e foram estabelecidos dois alvos, um difícil com valor de 130280000 (0,31% acima do melhor valor encontrado) e um médio igual a 130340000 (0,36%). Para cada alvo os algoritmos foram executados 100 vezes com sementes diferentes, o que tornam estas execuções independentes. O limite de tempo para atingirem os alvos foi de 3600s, tempo

o suficiente para que o alvo difícil possa executar a fase de pós-otimização do GRC12 em diversas execuções.

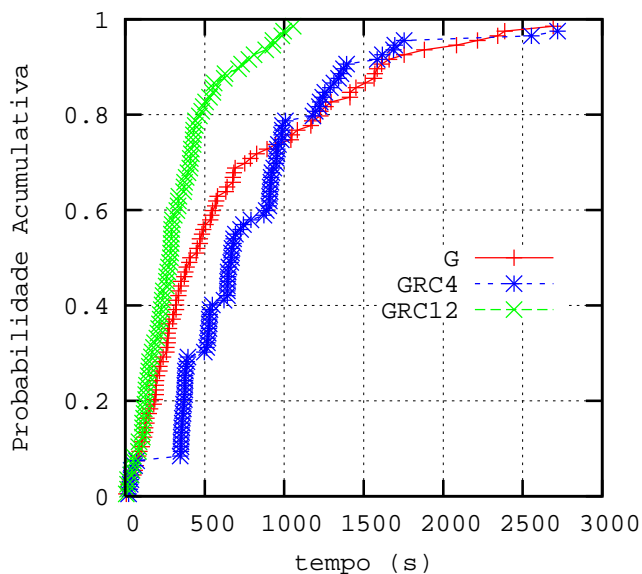


Figura 1: Distribuição de probabilidade para o alvo médio.

Na Figura 1 são mostrados o desempenho de G, GRC12 e GRC4 para o alvo médio. GRC12 atinge o alvo em 1056s, enquanto que o G atinge em 2692s e GRC4 em 2720s. A Figura 2 mostra o desempenho de G, GRC12 e GRC4 para o alvo difícil. GRC4 atinge o alvo em 3368s, GRC12 com 3478s conseguem atingir 0,95% do alvo, enquanto que G com 3488s só conseguem 0,46%. Para o alvo difícil, GRC12 atingiu o tempo limite estabelecido (3600s) somente em três execuções, G atingiu este tempo limite em 53 execuções e GRC4 em nenhuma execução atingiu o tempo limite. Nas distribuições de probabilidade empírica pode-se observar que GRC12 obtém melhor desempenho do que G.

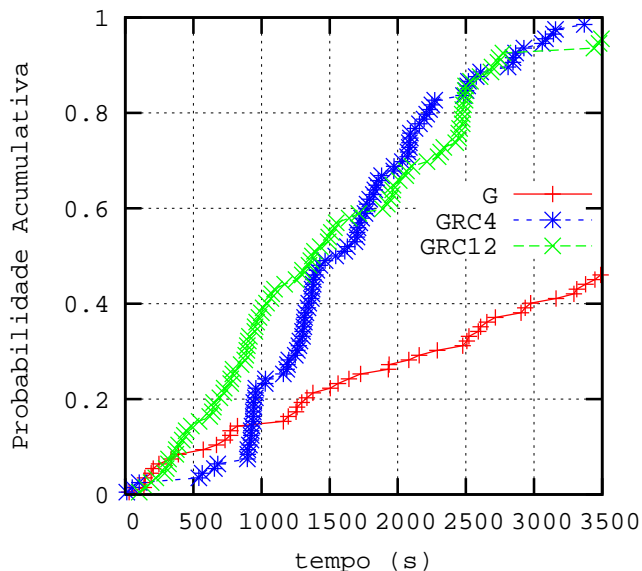


Figura 2: Distribuição de probabilidade para o alvo difícil.

4. CONCLUSÃO

Neste trabalho foram propostas seis heurísticas GRASP para a solução do PCVG. Ressalta-se que estas heurísticas tratam o PCVG sem estabelecer *a priori* a sequência de visitas dos grupos. Diversas avaliações foram realizadas e mostraram que a introdução de Reconexão de Caminhos na heurística GRASP melhora significativamente o desempenho da versão tradicional

(G), mesmo que de forma empírica. Nos resultados computacionais foi verificado que o uso conjugado de Reconexão de Caminhos durante e após as iterações do GRASP (GRC12) tende a produzir os algoritmos mais eficientes. Nas distribuições de probabilidade empírica foi observado que as heurísticas GRC12 e GRC4 apresentam uma probabilidade maior de encontrar uma solução de melhor qualidade em um tempo menor do que a heurística G. Os métodos heurísticos desenvolvidos neste trabalho alcançaram soluções de qualidade em tempo computacional razoável e são competitivos ao método exato.

5. AGRADECIMENTOS

O autor agradece a Jean-Yves Potvin, *Département d'informatique et de Recherche Opérationnelle, Université de Montréal* pelo material bibliográfico e a David S. Johnson, *AT&T Labs - Research*, pelos códigos para gerar instâncias do tipo 2. Este trabalho tem o apoio parcial da CAPES (Programa PIQDTec).

REFERÊNCIAS

- [1] J. A. Chisman. “The Clustered Traveling Salesman Problem”. *Comput. Oper. Res.*, vol. 2, no. 2, pp. 115–119, 1975.
- [2] N. Guttmann-Beck, R. Hassin, S. Khuller e B. Raghavachari. “Approximation Algorithms with Bounded Performance Guarantees for the Clustered Traveling Salesman Problem”. *Algorithmica*, vol. 28, no. 4, pp. 422–437, 2000.
- [3] A. Weintraub, J. Aboud, C. Fernandez, G. Laporte e E. Ramirez. “An Emergency Vehicle Dispatching System for an Electric Utility in Chile”. *J. Oper. Res. Soc.*, vol. 50, pp. 690–696, 1999.
- [4] G. Laporte e U. Palekar. “Some Applications of the Clustered Travelling Salesman Problem”. *J. Oper. Res. Soc.*, vol. 53, no. 9, pp. 972–976, 2002.
- [5] H. Ghaziri e I. H. Osman. “A Neural Network for the Traveling Salesman Problem with Backhauls”. *Comput. & Ind. Eng.*, vol. 44, no. 2, pp. 267–281, 2003.
- [6] M. Gendreau, G. Laporte e J. Y. Potvin. “Heuristics for the Clustered Traveling Salesman Problem”. Technical Report CRT-94-54, Centre de recherche sur les transports, Universidade de Montreal, Montreal, Canada, 1994.
- [7] J.-Y. Potvin e F. Guertin. “A Genetic Algorithm for the Clustered Traveling Salesman Problem with an A Priori Order on the Clusters”. Technical Report CRT-95-06, Centre de recherche sur les transports, Université de Montréal, Canada, 1995.
- [8] G. Laporte, J.-Y. Potvin e F. Quilleret. “A Tabu Search Heuristic using Genetic Diversification for the Clustered Traveling Salesman Problem”. *J. Heuristics*, vol. 2, no. 3, pp. 187–200, 1996.
- [9] S. Anily, J. Bramel e A. Hertz. “A 5/3-approximation Algorithm for the Clustered Traveling Salesman Tour and Path Problems”. *Oper. Res. Lett.*, vol. 24, no. 1-2, pp. 29–35, 1999.
- [10] M. Resende e C. Ribeiro. “Greedy Randomized Adaptive Search Procedures”. In *Handbook of Metaheuristics*, editado por F. Glover e G. Kochenberger, pp. 219–249. Kluwer Academic Publishers, 2002.
- [11] K. Jongens e T. Volgenant. “The Symmetric Clustered Traveling Salesman Problem”. *Eur. J. Oper. Res.*, vol. 19, no. 1, pp. 68–75, 1985.
- [12] J.-Y. Potvin e F. Guertin. “The Clustered Traveling Salesman Problem: A Genetic Approach”. In *Meta-heuristics: Theory & Applications*, editado por I. H. Osman e J. Kelly, pp. 619–631. Kluwer Academic Publishers, Norwell, MA, USA, 1996.
- [13] C. Ding, Y. Cheng e M. He. “Two-Level Genetic Algorithm for Clustered Traveling Salesman Problem with Application in Large-Scale TSPs”. *Tsinghua Sci. Technol.*, vol. 12, no. 4, pp. 459–465, 2007.
- [14] M. Mestria, L. S. Ochi e S. L. Martins. “GRASP com Memória Adaptativa para o Problema do Caixeiro Viajante com Grupamentos”. In *XXIX Encontro Nacional de Engenharia de Produção*, volume 1, Salvador, BA, Brazil, 2009. Anais do XXIX ENEGEP, ABEPRO, RJ (CD-ROM).
- [15] M. Mestria, L. S. Ochi e S. L. Martins. “Heurísticas Híbridas para o Problema do Caixeiro Viajante com Grupamentos”. In *XXX Encontro Nacional de Engenharia de Produção*, volume 1, São Carlos, SP, 2010. Anais do XXX ENEGEP, ABEPRO, RJ (CD-ROM).
- [16] D. Aloise e C. Ribeiro. “Adaptive memory in multistart heuristics for multicommodity network design”. *Journal of Heuristics*, vol. 17, no. 2, pp. 153–179, 2011.
- [17] M. G. Resende, R. Martí, M. Gallego e A. Duarte. “GRASP and Path Relinking for the Max-Min Diversity Problem”. *Comput. & Oper. Res.*, vol. 37, no. 3, pp. 498–508, 2010.

- [18] F. Glover. “Tabu Search and Adaptive Memory Programing - Advances, Applications and Challenges”. In *Interfaces in Computer Science and Operations Research*, editado por R. S. Barr, R. V. Helgason e J. L. Kennington, pp. 1–75. Kluwer Academic Plublishres, Dordrecht, MA, EUA, 1996.
- [19] C. Blum e A. Roli. “Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison”. *ACM Computing Surveys*, vol. 35, no. 3, pp. 268–308, 2003.
- [20] M. Mestria, L. S. Ochi e S. L. Martins. “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)*, volume 1, Ouro Preto, MG, Brazil, 2009. Anais do IX CBRN/Inteligência Computacional, SBRN, RJ (CD-ROM).
- [21] CPLEX. “ILOG CPLEX 11.2 User’s Manual and Reference Manual, 2009”. ILOG S.A., 2009. Acesso em: 06 jan. 2009.
- [22] R. Aiex, M. Resende e C. Ribeiro. “TTT plots: a Perl Program to Create Time-to-Target Plots”. *Opt. Lett.*, vol. 1, no. 4, pp. 355–366, 2007.