

Algoritmo Branch-and-Bound para o Problema do Caixeiro Alugador

Vítor de Godeiro Marques¹

¹Instituto Metr pole Digital
Universidade Federal do Rio Grande do Norte (UFRN)
Av. Senador Salgado Filho, 3000 – 59078-970 – Natal – RN – Brasil

vitorgodeirom@gmail.com

Abstract. *In the Car Renter Salesman Problem, a set of cities should be visited, beginning and ending in the same city, using vehicles rented for transport. Has the goal to minimize the cost to perform the route. Several vehicles are available for rental, each one with its characteristics and operating costs. These costs include besides the amount paid for the rent of vehicles, the fuel consumption and toll. Additionally, there is a fee to return the vehicle to the city in which it was rented, if it went returned in a different city. In this paper we propose an exact algorithm based on the Branch-and-Bound method for the problem mentioned.*

Resumo. *No Problema do Caixeiro Alugador, um conjunto de cidades devem ser visitadas, comeando e terminando na mesma cidade, usando ve culos alugados para transporte. Possui como objetivo minimizar o custo para realizar o percurso. Diversos ve culos est o dispon veis para aluguel, cada qual com suas pr prias caracter sticas e custos operacionais. Esses custos incluem al m do valor pago pelo aluguel dos ve culos, o consumo de combust vel e ped gios. Adicionalmente, existe uma taxa para retornar um ve culo para a cidade em que ele foi alugado, se ele for entregue em uma cidade diferente. Nesse trabalho   proposto um algoritmo exato baseado no m todo Branch-and-Bound para o referido problema.*

1. Introdu o

O Problema do Caixeiro Viajante – PCV (do ingl s *Traveling Salesman Problem* – TSP)   um dos problemas cl ssicos da otimiza o combinat ria e um dos problemas de otimiza o combinat ria mais pesquisados na literatura. O PCV possui v rias e importantes aplica es pr ticas e diversas variantes [Gutin and Punnen 2002]. Uma das variantes do PCV   o Problema do Caixeiro Alugador (do ingl s *Car Renter Salesman Problem* – CaRS), recentemente descrito na literatura em [Goldbarg et al. 2011]. Esse modelo tanto pode representar importantes aplica es na  rea do aluguel de transporte para turismo quanto em manufatura flex vel [Goldbarg et al. 2012]. Por admitir o PCV como um subcaso elementar, o modelo representa uma variante de complexidade seguramente desafiadora [Goldbarg et al. 2011].

Neste trabalho, propomos atacar o Problema do Caixeiro Alugador de forma exata, para isso aplicamos a t cnica *Branch-and-Bound* [Balas and Toth 1983], pois ela tem sido aplicada com bastante frequ ncia em problemas de otimiza o combinat ria, incluindo o Problema do Caixeiro Viajante [Prestes 2006].

2. Caixeiro Alugador

Nos dias atuais existem mais de 90 companhias de aluguel de veículos com porte significativo no mercado mundial [Car 2014]. A importância do negócio de aluguel de veículos pode ser avaliada pelo faturamento do setor como pelo porte das companhias prestadoras de serviço. Algumas destas chegam a faturar bilhões de dólares [Conrad and Perlut 2006]. Contudo o aluguel de veículos é apenas uma fração dos custos de transporte, sobre esses custos adiciona-se pelo menos as despesas com combustível, seguros e pedágios [Goldbarg et al. 2011]. Apesar de alguns trabalhos serem dedicados a área de otimização combinatória para a indústria de aluguel de carros, tais como os revisados por [Yang et al. 2008], o ponto de vista do cliente não foi objeto de estudo nessas publicações. A abordagem do ponto de vista do usuário de carros alugados é introduzida no trabalho [Goldbarg et al. 2011], a partir do Problema do Caixeiro Alugador.

Considerando um grafo $G = (N, M)$ ponderado em arestas, com r carros disponíveis para alugar. Uma matriz de custos $C^r = [c_{ij}^r]$ indica o custo total de aluguel pela distância percorrida, combustível e possíveis pedágios entre quaisquer duas cidades i e j com o veículo r . A matriz $D^r = [d_{ij}^r]$ indica o custo total do carro r alugado na cidade i ser devolvido na cidade j , tal que $i \neq j$. A função objetivo é minimizar o custo total do percurso mais o custo de retorno dos veículos alugados. Neste trabalho, é considerada a variante do Caixeiro Alugador em que cada veículo pode ser alugado apenas uma única vez.

3. Branch-and-Bound na solução do Problema do Caixeiro Alugador

O cálculo do limitante inicial do problema é realizado a partir do algoritmo evolucionário da classe memética [Radcliffe and Surry 1994] descrito em [Goldbarg et al. 2011]. Nesse algoritmo a população inicial é gerada pelo procedimento do vizinho mais próximo adaptada ao CaRS. A etapa da busca local desse algoritmo é representada pela busca em vizinhança variável descendente (do inglês *Variable Neighborhood Descent - VND*), o procedimento explora as vizinhanças denominadas *InvertSol*, *Insert&Saving* e *2-Shift*.

Determinado o limitante superior inicial (melhor solução encontrada) do problema, a árvore de enumeração das soluções possíveis é percorrida em profundidade, existindo apenas duas regras para interromper a exploração de um ramo da árvore.

A primeira regra é se o ramo em análise (solução parcial) possui custo maior ou igual ao custo do limite superior (melhor solução encontrada), ou quando encontramos uma solução que possui custo maior ou igual ao limite superior. Caso o custo da solução encontrada for menor que o custo do limite superior, a melhor solução encontrada é atualizada e se ainda existir ramos em aberto na árvore de solução o algoritmo irá continuar o processo de *branching*.

A segunda e última regra ocorre na verificação se a solução parcial em análise é promissora ou não para solucionar o problema. Essa verificação é realizada a partir de uma função que apura se o custo da solução parcial encontrada até o momento acrescida dos menores custos de qualquer um dos carros disponíveis ir de uma cidade ainda não visitada para qualquer outra cidade é menor que o custo da melhor solução encontrada, se for menor essa solução é promissora, caso contrário o algoritmo realiza a poda.

4. Experimentos e Resultados

O presente tópico desenvolve um experimento computacional com o objetivo de validar o algoritmo exato proposto, e o faz por intermédio de uma comparação de desempenho com os resultados de um algoritmo *backtracking* apresentado no trabalho [Goldberg et al. 2012]. Os experimentos foram executados em uma plataforma Intel Core I5 3.20 GHz, 8GB RAM em Linux com C++, cuja implementação pode ser acessada em: <https://github.com/vitorgodeiro/CaixaViajanteAlugador>. São relatados resultados para **24** instâncias não-euclidianas que se encontram disponíveis em: <http://dimap.ufrn.br/lae/projetos/CaRS.php>. O número de cidades visitadas varia entre **9** e **16** e o número de veículos disponíveis para alugar varia entre **2** e **5**.

A Tabela 1 apresenta os resultados obtidos pelo algoritmo *Branch and Bound* descrito nesse trabalho e os resultados de um algoritmo *backtracking* apresentado no trabalho [Goldberg et al. 2012]. A coluna **Nome** contém o nome da instância que está sendo testada, a coluna **#Cidade** possui o número de cidades da instância em análise, a coluna **#Carro** possui o número de carros disponíveis para alugar na instância em análise, na coluna **T(s)** apresenta o tempo de execução em segundos do algoritmo *Branch and Bound* para o dado caso de teste, a coluna **T(s)'** apresenta o tempo de execução em segundos do algoritmo *backtracking* e a coluna **Opt** apresenta o valor das soluções ótimas.

Tabela 1. Resultados do Branch-and-Bound.

Nome	#Cidade	#Carro	T(s)	T(s)'	Opt
Egito9n	9	4	0,1195	–	610
Bolivia10n	10	3	0,1007	–	681
Etiopia10n	10	4	3,5524	–	666
Mauritania10n	10	2	0,0842	1	571
AfricaSul11n	11	3	8,3306	–	714
Colombia11n	11	2	1,3593	14	639
Mali11n	11	4	33,7877	–	777
Angola12n	12	2	24,6175	144	656
Chade12n	12	4	95,5035	–	930
Niger12n	12	3	17,8579	–	869
Ira13n	13	4	82,5575	–	909
Mongolia13n	13	3	86,7134	–	740
Peru13n	13	2	5,0023	1847	693
Arabia14n	14	5	442,0730	–	1026
BrasilRJ14n	14	2	475,8590	35263	167
Indonesia14n	14	3	307,3620	–	796
Mexico14n	14	4	1788,7200	–	902
Libia14n	14	2	816,5620	28331	760
Argelia15n	15	3	2574,3600	–	863
Congo15n	15	2	81,9783	412212	886
Cazaquistao15n	15	5	2303	–	1043
Sudao15n	15	4	11310,4000	–	1020
Argentina16n	16	2	10745,4000	7612310	894
BrasilRN16n	16	2	1238,2600	7613217	188

O tempo de execução do algoritmo *Branch and Bound* proposto nesse trabalho obteve tempos 5, 11, 6148 vezes menores que os resultados do algoritmo *backtracking*. Adicionalmente podemos observar que a complexidade do problema é seguramente de-

safiadora, pois com um pequeno aumento do número de cidades observamos um grande aumento no tempo de solução.

5. Conclusão

Neste trabalho, propomos uma versão de algoritmo exato baseado no método *Branch and Bound* para o CaRS e realizamos uma comparação com o único trabalho da literatura que apresenta um algoritmo exato para o CaRS.

A comparação com o outro trabalho da literatura demonstrou uma redução expressiva nos tempos de execução para encontrar a solução exata do Caixeiro Alugador, o que sugere que a metodologia proposta foi adequada ao contexto. Portanto estes resultados nos dão confiança para continuar atacando este problema com técnicas de programação matemática como por exemplo *branch-and-cut*, *branch-and-price*, *branch-cut-and-price*, *branch-and-prune*; uma vez que não existem outros estudos experimentais publicados sobre este problema para comparação direta.

Agradecimentos

Ao Programa de Educação Tutorial do Ministério da Educação.

Referências

- Balas, E. and Toth, P. (1983). Branch and bound methods for the traveling salesman problem. Technical Report DRC-70-22-83, Carnegie-Mellon University. Computer science. Pittsburgh (PA US).
- Car (2014). Car Rental Business - Global Strategic Business Report. http://www.researchandmarkets.com/reports/338373/car_rental_business_global_strategic_business. Accessed: 2016-08-29.
- Conrad, C. and Perlut, A. (2006). Enterprise Rent-A-Car Hits New Billion-Dollar revenue mark for 3rd consecutive year. https://www.enterpriseholdings.com/content/dam/ehicom/press-releases/Enterprise_FY06_Sept06.pdf. Accessed: 2016-08-29.
- Goldbarg, M. C., Asconavieta, P. H., and Goldbarg, E. F. G. (2012). Memetic algorithm for the traveling car renter problem: an experimental investigation. *Memetic Computing*, 4(2):89–108.
- Goldbarg, M. C., da Silva, P. H. A., and Goldbarg, E. F. G. (2011). Algoritmos Evolucionários na Solução do Problema do Caixeiro Alugador. chapter 14, pages 301–330.
- Gutin, G. and Punnen, A. P., editors (2002). *The traveling salesman problem and its variations*. Combinatorial optimization. Kluwer Academic, Dordrecht, London.
- Prestes, A. N. (2006). Uma Análise Experimental de Abordagens Heurísticas Aplicadas ao Problema do Caixeiro Viajante. Master's thesis, Universidade Federal do Rio Grande do Norte. Accessed: 2016-09-08.
- Radcliffe, N. J. and Surry, P. D. (1994). *Formal memetic algorithms*, pages 1–16. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Yang, Y., Jin, W., and Hao, X. (2008). Car rental logistics problem: A review of literature. In *2008 IEEE International Conference on Service Operations and Logistics, and Informatics*, volume 2, pages 2815–2819.