



## ESTUDO DE UMA VARIAÇÃO DO PROBLEMA CaRS

**Matheus da Silva Menezes**  
Universidade Federal Rural do Semiárido  
Mossoró – RN, Brasil  
matheus@ufersa.edu.br

**Gabriela Colaço**  
Universidade Federal Rural do Semiárido  
Mossoró – RN, Brasil  
gabicolaco@yahoo.com.br

**Vanessa Elionara Souza Ferreira**  
Universidade Federal Rural do Semiárido  
Mossoró – RN, Brasil  
vanessaelionara@ufersa.edu.br

## ESTUDO DE UMA VARIAÇÃO DO PROBLEMA CaRS

### RESUMO

O Problema do Caixeiro Alugador ou *The Traveling Car Renter Problem* (CaRS) foi introduzido em [Goldberg et al, 2012], generaliza o Problema do Caixeiro Viajante quando apenas um veículo é utilizado e tem como objetivo minimizar os custos de visitar  $n$  cidades, viajando com carros alugados. Na literatura, todos os trabalhos relatados investigam a mesma variante do problema CaRS: o grafo que modela o problema possui todas as ligações (**completo**); qualquer carro pode ser alugado em qualquer cidade (**total**); o carro alugado pode ser devolvido em qualquer cidade (**irrestrito**); cada carro só pode ser locado uma vez (**sem repetição**); os custos de retorno independem da topologia das estradas ou restrições de rede (**livre**); os custos de utilizar um carro  $c$ , para ir de uma cidade  $i$  para uma cidade  $j$  são simétricos aos de  $j$  para  $i$  utilizando  $c$  (**simétrico**). Na prática, o usuário pode não ter uma cidade fixa para o início do trajeto, as empresas de aluguel não necessariamente aplicam restrições sobre alugar o mesmo tipo de carro e não temos garantia de que todas as cidades estão aptas a receber a devolução de veículos. Este trabalho apresenta um cenário ainda não analisado para o problema CaRS, onde o mesmo tipo de carro pode ser locado mais de uma vez durante o trajeto (**com repetição**), não existe uma cidade base pré-definida e são aplicadas restrições de devolução do carro alugado (**restrito**). Para tanto, uma formulação para o problema foi feita e implementada no *solver* GLPK, com tempo de processamento limitado a 10000 segundos. Um algoritmo memético, baseado em [Menezes, 2014] com três buscas locais (*invertSol*, *replaceSavingCar* e *2-opt*) adaptadas ao problema foi proposto. Essas buscas locais são aplicadas em seqüência no procedimento *buscaLocalMultiOper()*, substituindo a solução de entrada se a mesma for melhorada no procedimento. O procedimento de restauração foi adaptado. Uma investigação experimental foi realizada para verificar o potencial do algoritmo memético. A qualidade das soluções e o tempo de processamento para o GLPK e o algoritmo memético foram comparados. Um conjunto de 29 instâncias Euclidianas e 29 instâncias não Euclidianas foi considerado nesse experimento. Trinta e sete soluções exatas foram encontradas pelo GLPK, e o algoritmo memético estabeleceu valores superiores para o problema em outras dezessete instâncias. Os resultados do experimento computacional mostraram que, para as instâncias Euclidianas, o algoritmo memético encontrou quase todas as soluções que o *solver* alcançou. Para as instâncias não Euclidianas, o algoritmo memético encontrou soluções melhores que o GLPK em dezesseis instâncias. O experimento também mostrou que, quanto maior a instância, maior é a tendência de que o algoritmo memético produza melhores soluções utilizando menos tempo que o GLPK.

**PALAVRAS CHAVE.** Algoritmo Memético, Problema do Caixeiro Alugador.