

# O Problema do Caixeiro Viajante com Coleta e Entrega Uma Abordagem via Variable Neighborhood Search(VNS)

Victor Alberti Costa<sup>1</sup>

<sup>1</sup>Departamento de Informática – Universidade Tecnológica Federal do Paraná (UTFPR)  
Via do Conhecimento, km 01 – 85.503-390 – Pato Branco – PR

victorcosta.2015@alunos.utfpr.edu.br

**Resumo.** *O Problema do Caixeiro Viajante com Coleta e Entrega compõe uma de inúmeras variações do Problema do Caixeiro Viajante, este artigo busca relacionar diferentes métodos de resolução do Problema do Caixeiro Viajante com Coleta e Entrega, para isso foi realizado uma abordagem do problema através da metaheurística Variable Neighborhood Search (VNS), e partir disso relacionar com a solução por meio da Heurística de Construção, Algoritmo Guloso, e também de algumas Heurísticas de Refinamento. A caráter de informação para aqueles que buscam saber a respeito do desempenho da metaheurística VNS e também sobre a restrição de Coleta e Entrega ao Problema do Caixeiro Viajante, este artigo pode servir como ajuda.*

**Abstract.** *The Traveling Salesman Pick-up Problem makes one of countless variations of the Traveling Salesman Problem, this article seeks to relate different methods of solving the Traveling Salesman's Pick-up and Delivery Problem, to this end, a problem approach was carried out through the Variable Neighborhood Search (VNS), and from this relate to the solution through the Construction Heuristic, Greedy Algorithm, and also some Refinement Heuristics. The informational nature for those seeking to know about the performance of the VNS meta heuristic and also about the restriction of collection and delivery to the traveling salesman problem, this article can help.*

## 1. Introdução

O Problema do Caixeiro Viajante (PCV) é um dos problemas mais estudados na pesquisa operacional, trata-se de de um problema que tenta determinar a menor rota para percorrer uma série de cidades (visitando uma única vez cada uma delas), retornando à cidade de origem. Ele é um problema de otimização NP-difícil inspirado na necessidade dos vendedores em realizar entregas em diversos locais (cidades) percorrendo o menor caminho possível, reduzindo o tempo necessário para a viagem e os possíveis custos com transporte e combustível. E a partir disso, posteriormente surgiram diferentes versões ao Problema do Caixeiro Viajante, com diferentes restrições e aplicações, dentre elas temos o Problema do Caixeiro Viajante com Coleta e Entrega envolvendo um único tipo produto, conhecido na literatura como *One-Commodity Pickup and Delivery Traveling Salesman Problem* (1-PDTSP) e uma variante do Problema do Caixeiro Viajante (PCV). O 1-PDTSP foi proposto por Hernandez-Pérez e Salazar-González (2004) e a principal diferença em relação ao PCV é que cada cliente é classificado de acordo com o tipo de serviço exigido, sendo este coleta e entrega.

## 2. Definição do problema

O problema tratado neste artigo é uma variação próxima de 1-PDTSP, que foi proposta por Mosheiov em 1994, antecedendo o popular 1-PDTSP de Hernandez-Pérez e Salazar-González, onde da mesma forma conta com dois tipos de clientes, onde cada um demanda de realizar uma coleta ou entrega ao veículo, que possui uma capacidade e originalmente parte de um depósito, para onde deve retornar ao fim do percurso. Contudo o Problema do Caixeiro Viajante com Coleta e Entrega de Mosheiov trabalha com dois produtos que devem ser transportados pelo veículo, onde o veículo parte do depósito com uma quantidade de carga inferior a sua carga máxima e este deve entregar o produto proveniente do depósito aos clientes que necessitam coletar carga do veículo e ao mesmo tempo deve coletar dos clientes que necessitam entregar carga ao do veículo, onde a carga coletada não é a mesma que será entregue aos clientes. A carga fornecida pelos clientes de entrega irá retornar ao depósito, uma analogia interessante para compreensão do problema é, um caminhão parte do depósito parcialmente carregado com garrafas cheias enquanto se move em direção de seus clientes, onde estes podem coletar as garrafas cheias reduzindo a carga total do caminhão e também podem entregar garrafas vazias ao caminhão e com isso aumentando a carga total do caminhão, sendo que ao fim do percurso todas as garrafas cheias devem ser entregues e todas as garrafas vazias que o depósito aguardava devem ser coletas.

## 3. Algoritmos utilizados

Para resolução do problema proposto por Mosheiov, foi utilizado a metaheurística *Variable Neighborhood Search* (VNS) proposta por Mladenovic e Hansen em 1997, onde sobre uma solução inicial efetua-se uma sequência de buscas locais e que utiliza o conceito de estruturas de vizinhança para explorar o espaço de busca e procurar as melhores soluções, escapando assim de ótimos locais e garantindo uma maior possibilidade e encontrar a solução ótima global. Sobre este conceito de VNS foi desenvolvido o algoritmo para resolução do problema de Mosheiov, que utiliza a metaheurística VNS porém com as características descritas a seguir. As Heurísticas de Refinamento, também conhecido como Busca Local, utilizados foram o Swap, 2-OPT e 3-OPT, originalmente a metaheurística VNS trabalha com buscas locais baseada em n-OPT, portanto a primeira adaptação ao problema foi a adição da estratégia de busca local Swap, que é técnica que consiste em trocar elementos vizinhos. Das estratégias de refinamento, temos também o 2-OPT que é compreendido pela remoção de 2 arcos do ciclo e reconecta-los, similar a técnica 3-OPT que faz o mesmo com 3 arcos, disto temos uma possibilidade de rearranjo do ciclo na técnica 2-OPT e 7 possibilidades de rearranjo do ciclo na técnica 3-OPT, contudo o primeiro e o último ponto que representam o depósito não devem mudar e toda alteração deve ser validada, isto é, um movimento em que o veículo é requisitado a entrega de mais mercadoria que ele realmente tenha e que seja entregue ao veículo mais mercadoria que ele possa suportar, devem ser desconsiderados. As heurísticas de refinamento aplicadas sobre a metaheurística VNS foram utilizadas sobre o Método *First Improvement* ou Primeira Melhora, no qual o método percorre o espaço de buscas e na primeira solução vizinha que melhora o resultado atual a heurística de refinamento é finalizada. Para fins de testes e curiosidade, foi aplicado a metaheurística VNS sobre a solução inicial, proveniente de duas fontes, sendo elas a Heurística Construtiva Gulosa e outra puramente aleatória, e com isso foi realizado uma comparação entre as duas soluções iniciais. E também, sobre as mesmas

soluções iniciais foi aplicado isoladamente as três Heurísticas de Refinamento.

```
1  Definir um conjunto de  $k_{max}$  vizinhanças;  
2  Determinar uma solução inicial  $x$ ;  
3  Enquanto não é satisfeito um critério de parada,  
3.1   $k \leftarrow 1$ ;  
3.2  Enquanto ( $k \leq k_{max}$ ),  
3.2.1  Gerar aleatoriamente  $x' \in N_k(x)$ ;  
3.2.2  Se ( $f(x') < f(x)$ ),  
3.2.2.1  Então  $x \leftarrow x'$ ;  $k \leftarrow 1$ ;  
3.2.2.2  Senão  $k \leftarrow k + 1$ ;  
3.2.3  Fim-se;  
3.3  Fim-enquanto;  
4  Fim-enquanto;
```

Critérios de parada: número máximo de iterações ou tempo máximo.

**Figura 1. Algoritmo VNS**

A Figura 1 consiste na sequência de passos que compõe a metaheurística VNS, em que para a aplicação deste trabalho foi atribuído ao conjunto de vizinhanças o valor 3, pois é compreendido por Swap, 2-OPT e 3-OPT.

#### 4. Resultados Computacionais

Todas as instâncias foram executadas uma vez utilizando o critério de parada de número máximo de iterações do algoritmo, onde este critério é frequentemente utilizado para algoritmos de otimização na literatura. Para esse trabalho foi utilizado Critério de Parada (CP) igual a 5, porque o custo computacional de mais iterações sobre as instâncias testadas resultaria em um tempo de execução mais significativo. O computador onde foram executados os testes apresenta processador Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz, 4Gb de memória RAM, 1Gb de Disco Rígido com Sistema Operacional Linux Mint 19.1 Tessa e executando sobre o compilador gcc version 7.4.0.

#### 5. Conclusão

A Tabela 1 relaciona os custos das instâncias com relação a Solução Ótima (opt) calculado utilizando o algoritmo *Branch and Cut*, a Heurística de Construção gulosa (Guloso) de onde foi utilizado a sequência de cidades provenientes do algoritmo guloso como solução inicial dos métodos subsequentes, as Heurísticas de Refinamento Swap, 2-OPT e 3-OPT e por fim o resultado da aplicação da Metaheurística Variable Neighborhood Search (VNS) e o último campo Tempo, onde corresponde ao tempo de execução dos algoritmos citados e também de mais uma execução para uma instância puramente aleatória, que não foi representada na Tabela 1 pois a por sua característica puramente aleatória o seu custo usualmente é maior que o algoritmo Guloso, pois o mesmo já está submetido a um conjunto de regras de otimização. Contudo, durante a execução repetitiva das instâncias, houveram momentos em que o custo retornado pela metaheurística VNS superou a resposta Gulosa e assim aproximando-se mais da solução ótima, em minhas observações ao aplicar um fator estocástico maior a solução inicial a mesma tem um escape maior de ótimos locais e pode resultar em um resposta melhor, assim uma alteração na metaheurística VNS incluindo

<b>Instância</b>	<b>Opt</b>	<b>Guloso</b>	<b>Swap</b>	<b>2-OPT</b>	<b>3-OPT</b>	<b>VNS</b>	<b>Tempo</b>
n20mosA	3816	5038	4956	4913	4913	4738	0,31s
n20mosB	3942	5024	4858	4858	4765	4601	0,34s
n20mosC	4048	4700	4648	4648	4648	4301	0,25s
n20mosD	4151	4425	4216	4216	4216	4200	0,22s
n20mosE	4387	6187	6086	6033	5819	5663	0,36s
n20mosF	4262	6331	6272	5655	5501	5024	0,30s
n30mosA	4686	5290	5225	5225	5225	4973	0,17s
n30mosB	4805	6050	6046	5991	5990	5755	0,13s
n30mosC	4503	4771	4659	4659	4659	4618	0,13s
n30mosD	5047	5819	5727	5635	5559	5328	0,14s
n30mosE	4929	6356	6322	6322	6322	6230	0,15s
n30mosF	4500	5567	5535	5495	5495	5512	0,19s
n40mosA	5192	5825	5825	5791	5791	5791	0,50s
n40mosB	5416	7333	7274	7181	7180	7035	0,45s
n40mosC	5079	7237	7193	6885	6885	6529	0,58s
n50mosA	5826	7103	7054	7038	7038	6979	1,27s
n50mosB	6080	8836	8757	8757	8743	7972	1,30s
n50mosC	6203	8691	8645	8613	8515	8422	1,15s
n60mosA	6279	8088	8047	8041	8041	8041	3,21s
n60mosB	6561	8002	7906	7906	7895	7758	2,81s
n60mosC	6295	9492	9478	8810	8585	7826	3,00s
n100mosA	7785	11203	11186	11102	11102	10519	25,3s
n100mosE	7933	19334	10168	10168	10168	9562	22,51s
n200mosA	10572	12803	12760	12742	12739	12043	13,44m
n300mosA	12789	16030	15991	15966	15861	15534	114,92m
n400mosA	14703	18247	18196	18115	18109	17350	249,25m
n500moH	16577	————	————	————	————	————	> 586m

**Tabela 1. Relação dos resultados para CP = 5**

um laço de repetição em toda sua estrutura e com isso armazenar a melhor função objetivo poderia reduzir a distância até a Solução Ótimo o que como consequência implicaria em maior custo de computacional. Como o fator aleatoriedade permite uma movimentação sobre o espaço de soluções, uma alteração que pode resultar na obtenção de resultados melhores é referente a implementação, a implementação do algoritmo 3-OPT apresenta 7 casos possível de rearranjo, o código realizado faz testes em cada uma das possibilidades de maneira sequencial, onde é testado o rearranjo do primeiro caso, verificado se a sequência não desrespeita nenhuma regra da variação de Mosheiov e o custo que a nova sequência apresenta, se este custo for menor o algoritmo 3-OPT chega ao fim e é aplicado as características do Algoritmo VNS, caso contrário é realizado o mesmo processo para o segundo caso do Algoritmo 3-OPT, portanto creio que a aplicação de aleatoriedade nos casos do Algoritmo 3-OPT podem resultar em uma maior liberdade no espaço de soluções e consequentemente a possibilidade de se obter melhores resultados.

## 6. Referências

<https://hhperez.webs.ull.es/PDsite/#XM94> .Acesso em: 15 de dez. de 2019

[https://pt.wikipedia.org/wiki/Problema\\_do\\_caixeiro-viajante](https://pt.wikipedia.org/wiki/Problema_do_caixeiro-viajante) .Acesso em: 15 de dez. de 2019

<http://www2.ic.uff.br/~satoru/conteudo/artigos/SBPO2010-Bruno.pdf> .Acesso em: 15 de dez. de 2019

G. Mosheiov. **The traveling salesman problem with pickup and delivery**. European Journal of Operational Research, 79:299-310, 1994.

H. Hernández-Pérez and J. J. Salazar-González. **The one-commodity pickup-and-delivery travelling salesman problem**. In M. Jünger, G. Reinelt, and G. Rinaldi, editors, Combinatorial Optimization-Eureka, You Shrink!, volume 2570, pages 89-104. Lecture Notes in Computer Science, Springer, 2003.

<http://www.ime.unicamp.br/~chico/mt852/slidesvns.pdf> .Acesso em: 15 de dez. de 2019

<http://tsp-basics.blogspot.com/2017/03/3-opt-move.html> .Acesso em: 15 de dez. de 2019

<https://paginas.fe.up.pt/~mac/ensino/docs/OR/CombinatorialOptimizationHeuristicsLocalSearch.pdf> .Acesso em: 15 de dez. de 2019

<https://www.marinha.mil.br/spolm/sites/www.marinha.mil.br.spolm/files/88242.pdf> .Acesso em: 15 de dez. de 2019