

O Problema do Caixeiro Viajante com Limite de Calado: uma Abordagem usando Busca Tabu

André Eduardo Alessi¹, Marco Antônio de Castro Barbosa²

¹Acadêmico do curso de Engenharia de Computação
Universidade Tecnológica Federal do Paraná
Campus Pato Branco

²Professor do Departamento de Informática
Universidade Tecnológica Federal do Paraná
Campus Pato Branco.

andrealessi@alunos.utfpr.edu.br, mbarbosa@utfpr.edu.br

Abstract. *In this article, we propose an approach to the Traveling Salesman Problem with Draft Limit (TSPDL) using the Tabu Search metaheuristic, with initial greedy solution and local swap search. TSPDL is a variant of the classic Traveling Salesman Problem (TSP) but with applications fit for sea transportation. The draft of a ship is the distance between the water level and the bottom of the vessel. Several ports have restrictions on the draft limit so that ships can enter the port safely. This problem can be reduced to Asymmetric Traveling Salesman Problem, which is NP-Difficult and has great practical interest. The results of the Tabu Search metaheuristic are compared with the exact solution of the problem, the greedy solution, and a solution using the local swap search over the greedy solution, highlighting the superiority of the Tabu Search in relation to the heuristic solutions and its proximity to the best possible solution.*

Resumo. *Neste artigo é proposta uma abordagem para o Problema do Caixeiro Viajante com Limite de Calado (PCVLC) utilizando a meta-heurística Busca Tabu, com solução inicial gulosa e busca local swap. O PCVLC é uma variação do clássico Problema do Caixeiro Viajante (PCV), porém com aplicação para transporte marítimo. O calado de um navio é a distancia entre o nível da água e a base inferior da embarcação. Vários portos possuem restrições quanto ao limite do calado para que os navios possam entrar no porto de maneira segura. Este problema pode ser reduzido ao Problema do Caixeiro Viajante Assimétrico, que é NP-Difícil e possui grande interesse prático. Os resultados da meta-heurística Busca Tabu são comparados com a solução exata do problema, a solução gulosa, e uma solução usando a busca local swap em cima da solução gulosa, explicitando a melhoria da Busca Tabu em relação às soluções heurísticas e sua proximidade com a melhor solução possível.*

1. Introdução

O Problema do Caixeiro Viajante com Limite de Calado (PCVLC), proposto por [Glomvik Rakke et al. 2012], é uma variação do clássico Problema do Caixeiro Viajante (PCV), porém aplicado para transportes marítimos.

O calado de um navio é a distância entre o nível da água e a base inferior do navio, que tende a aumentar quando acrescenta-se carga à embarcação. Todo porto tem um limite de calado, e pode acontecer do navio ter um calado maior que o limite do porto, impedindo que o veículo consiga ancorar. A Figura 1, como vista em [Machado et al. 2015], ilustra o calado de um navio quando este está vazio e carregado, respectivamente.

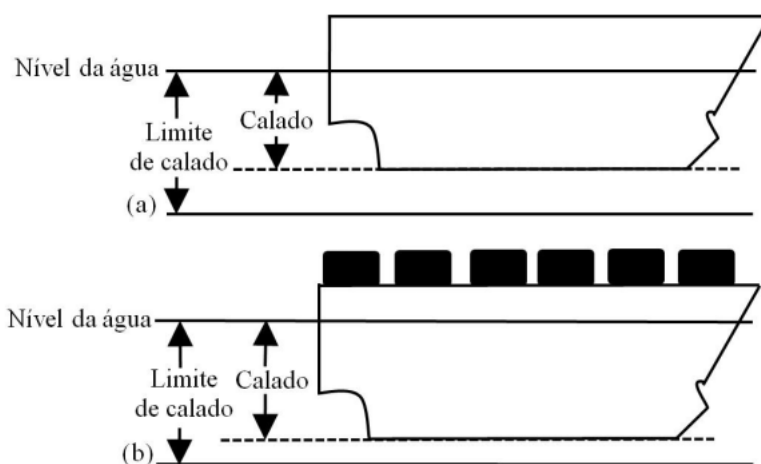


Figura 1. (a) Navio sem carga (b) Navio carregado. Fonte: [Machado et al. 2015]

O objetivo do PCVLC é fazer com que o navio percorra a rota de menor distância possível, saindo de um porto inicial, passando por todos os portos uma única vez e retornando ao porto de origem, além de obedecer o limite do calado de cada porto.

Segundo [Glomvik Rakke et al. 2012], o PCVLC pode ser reduzido ao Problema do Caixeiro Viajante Assimétrico, um problema da classe NP-Difícil, logo, o PCVLC encaixa-se na mesma categoria.

[Glomvik Rakke et al. 2012], ao apresentar o PVCLC, apresenta duas formulações matemáticas a ele, utilizando o método *branch-and-cut*, fornecendo soluções exatas ao problema. Em [Battarra et al. 2014], são apresentadas três novas formulações matemáticas para o PVCLC, e os autores também utilizaram o método *branch-and-cut* para obterem soluções exatas para o problema, aplicando-as para instâncias maiores.

Em [Todosijević et al. 2014], os autores utilizam duas novas soluções variantes da meta-heurística *Variable Neighborhood Search* (VNS), conseguindo assim resolver instâncias maiores. Para realizar seus experimentos, os autores propuseram novas instâncias de teste.

Em [Machado et al. 2015], é apresentada uma solução para o PCVLC utilizando a meta-heurística *Greedy Randomized Adaptive Search Procedure* (GRASP).

[Morais et al. 2017] apresentam uma solução para o PCVLC utilizando a meta-heurística *Simulated Annealing*.

Os trabalhos apresentados constituem o estado da arte do PCVLC. Por conta da baixa quantidade de soluções encontradas na literatura, pretende-se contribuir com os estudos do PCVLC com uma solução utilizando a meta-heurística Busca Tabu.

Este artigo está estruturado como se segue. Na seção 2 são apresentados os principais conceitos básicos deste trabalho: a formulação matemática do PCVLC e uma breve explicação da meta-heurística Busca Tabu. Na seção 3 a solução proposta neste trabalho é detalhada. Na seção 4 pode-se encontrar os experimentos computacionais realizados e seus resultados. Na seção 5 é apresentada a conclusão deste trabalho.

2. Conceitos Básicos

Nesta seção será apresentada a formulação matemática original do PCVLC, proposta por [Glomvik Rakke et al. 2012]. Optou-se por esta formulação por ser a original e de melhor entendimento para o autor. Também será apresentada uma breve introdução à meta-heurística Busca Tabu [Glover 1986].

2.1. Formulação matemática do PCVLC

O PCVLC pode ser definido como um grafo direcionado $G = (V, A)$, onde o conjunto de vértices é $V = \{0, 1, \dots, n\}$, representando os portos, e o conjunto de arestas é $A = \{(i, j) \mid i, j \in V \text{ e } i \neq j\}$, que simboliza as conexões entre os portos. O nó 0 representa o porto de origem. Cada aresta (i, j) possui um custo c_{ij} associado. Os nós restantes $V \setminus \{0\}$ simbolizam os portos que devem ser visitados. Cada porto, exceto o de origem, tem uma demanda de carga representada por d_i , $i \in V \setminus \{0\}$, além de um limite de calado l_i , $i \in V \setminus \{0\}$. A carga inicial do navio é representada por $Q = \sum_{i \in V \setminus \{0\}} d_i$, cujo valor diminui a cada porto visitado. O navio não pode entrar no porto i se estiver com a carga atual maior do que o limite do calado l_i do porto, com risco de encalhar a embarcação.

A seguir será apresentada uma formulação matemática que levará a uma solução para o PCVLC. Considerar-se-á o uso de uma variável binária x_{ij} , que recebe o valor 1 se a aresta (i, j) é utilizada na solução e 0 caso contrário. A carga do navio na aresta (i, j) é representada pela variável y_{ij} . Define-se o modelo como:

$$\min \sum_{(i,j) \in A} c_{ij} x_{ij} \quad (1)$$

sujeito a:

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

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

$$\sum_{i \in V} y_{ij} - \sum_{i \in V} y_{ji} = d_j, \forall j \in V \setminus \{0\} \quad (4)$$

$$\sum_{i \in V} y_{0i} = \sum_{i \in V \setminus \{0\}} d_i \quad (5)$$

$$\sum_{i \in V} y_{i0} = 0 \quad (6)$$

$$0 \leq y_{ij} \leq l_j x_{ij}, \forall (i, j) \in A \quad (7)$$

$$x_{ij} \in \{0, 1\}, \forall (i, j) \in A \quad (8)$$

As restrições 2 e 3 estabelecem, respectivamente, os graus de entrada e saída de cada vértice, garantindo que cada porto seja visitado apenas uma vez. A restrição 4 garante que a demanda de cada porto seja satisfeita e impede a criação de subciclos. As restrições 5 e 6 garantem que o navio saia totalmente carregado da origem e retorne totalmente descarregado para a origem. A restrição 7 impõe que o navio respeite o limite do calado dos portos a serem visitados. Por fim, a restrição 8 impõe que, de forma binária, uma aresta esteja ou não presente em uma solução.

2.2. Busca Tabu

A Busca Tabu é uma meta-heurística proposta por [Glover 1986] com o objetivo de permitir que buscas locais escapem de ótimos locais por utilizar uma memória, chamada de *Lista Tabu*, que guarda o movimento mais recente da busca e proíbe que este movimento seja realizado novamente.

Segundo [Gendreau 2003], os dois elementos mais básicos da Busca Tabu são o *espaço de busca* e a *estrutura da vizinhança*. O espaço de busca é o espaço de todas as soluções que podem ser visitadas durante a busca. A estrutura da vizinhança $N(S)$ de uma solução S é o conjunto de todas as soluções obtidas ao se aplicar uma única transformação local em S .

Outro conceito importante da Busca Tabu é a *função de aspiração*. Segundo [Gendreau 2003], alguns movimentos que poderiam resultar em soluções melhores podem ser proibidos pela Lista Tabu, e a função de aspiração tem o papel de evitar isso. A abordagem mais comum, e utilizada neste trabalho, é de permitir que um movimento, mesmo que Tabu, seja realizado se ele melhorar a melhor solução conhecida.

Uma estratégia da Busca Tabu é a de *diversificação*. [Gendreau 2003] diz que um dos maiores problemas de métodos baseados em buscas locais como a Busca Tabu é que, como o nome implica, eles tendem a ser muito locais, e ficam presos em uma região do espaço de solução. A estratégia de diversificação utilizada neste trabalho foi utilizar uma técnica de *swap* aleatória, podendo buscar soluções em outras regiões do espaço de solução. Outra estratégia, não utilizada neste trabalho, é ter a lista tabu como uma lista de frequência, tendo armazenado quantas vezes uma movimentação é feita e de tempos em tempos realiza movimentos com menor frequência para buscar novas regiões do espaço de solução.

[Gendreau 2003] ainda explicita a tendência de *hibridização* entre meta-heurísticas, ou seja, utilizar a Busca Tabu em conjunto com outras meta-heurísticas. Embora este conceito não seja explorado neste trabalho, estudos futuros podem obter resultados interessantes utilizando-se de hibridização.

Outros conceitos intermediários e avançados da Busca Tabu existem, contudo não serão explorados neste trabalho.

3. A Solução Proposta

Neste trabalho, propõe-se uma solução que realize a Busca Tabu a partir de uma solução inicial gulosa.

3.1. Construção da Solução Inicial

O algoritmo 1 ilustra a estratégia utilizada para gerar a solução inicial gulosa. Optou-se por um método construtivo para a solução inicial por apresentar melhores resultados na prática do que um método puramente aleatório.

Para cada instância de teste, teremos o número de portos n , a matriz $m[n][n]$ contendo a distância entre os portos, o vetor $limitecalado[n]$ contendo o limite do calado de cada porto, o vetor $demanda[n]$ contendo a demanda de carga de cada porto e o $peso$ sendo a soma da demanda de todos os portos.

Cria-se o vetor $visitados[n]$, onde para cada porto possui a informação se este já foi visitado ou não. A rota a ser gerada tem o porto 0 como inicial e final.

A cada iteração, escolhe-se o porto que tem a menor distância em relação ao atual, não foi visitado e cujo limite do calado é maior ou igual ao peso do navio.

Algoritmo 1: Método Guloso

Entrada: $n, M[n][n], limitecalado[n], demanda[n], peso$

Saída: $rota[n + 1]$

```
1 início
2    $rota[0] \leftarrow 0$ 
3    $visitados \leftarrow \emptyset$ 
4    $visitados[0] \leftarrow 1$ 
5    $anterior \leftarrow 0$ 
6   para  $i$  de 1 até  $n - 1$  faça
7      $menorDist \leftarrow$  menor valor em  $m[n][n]$ 
8     para  $j$  de 0 até  $n - 1$  faça
9       se  $(visitados[j] = 0)$  e  $(limitecalado[j] \geq peso)$  e
         $(m[anterior][j] \leq menorDist)$  e  $(anterior \neq j)$  então
10         $prox \leftarrow j$ 
11         $menorDist \leftarrow m[anterior][j]$ 
12      fim
13    fim
14     $anterior \leftarrow proximo$ 
15     $rota[j] \leftarrow anterior$ 
16     $visitados[anterior] \leftarrow 1$ 
17     $peso \leftarrow peso - demanda[anterior]$ 
18  fim
19   $rota[n] \leftarrow 0$ 
20 fim
21 retorna  $rota$ 
```

3.2. O método Busca Tabu

O algoritmo 2 ilustra a implementação da Busca Tabu, que recebe como solução inicial S_0 a rota gulosa encontrada pelo algoritmo 1. Neste algoritmo, S é a solução atual, S^* é a melhor solução encontrada, f é a função objetivo (neste problema, a distância total

percorrida), LT é a Lista Tabu, $iter$ é o contador de iterações, $melhorIter$ é a última iteração que obteve melhora na solução e $BTmax$ é o número de iterações sem melhoria permitidas.

Algoritmo 2: O Algoritmo Busca Tabu

Entrada: Solução inicial S_0

Saída: Melhor solução encontrada

```

1  início
2     $S \leftarrow S_0$ 
3     $S^* \leftarrow S_0$ 
4     $f^* \leftarrow f(S_0)$ 
5     $LT \leftarrow \emptyset$ 
6     $iter \leftarrow 0$ 
7     $melhorIt \leftarrow 0$ 
8     $BTmax \leftarrow 10000$ 
9    repita
10      $iter \leftarrow iter + 1$ 
11     Escolher o melhor vizinho  $S'$ , não Tabu ou que é aceito pela
        função de aspiração, entre 100 vizinhos aleatórios de  $N(S)$ 
12     se  $f(S') < f(S)$  então
13        $S \leftarrow S'$ 
14     fim
15     se  $f(S) < f^*$  então
16        $f^* \leftarrow f(S)$ 
17        $melhorIt \leftarrow iter$ 
18       Atualiza  $LT$ 
19       Atualiza função de aspiração
20     fim
21     até  $(iter - melhorIt) \leq BTmax$ ;
22 fim
23 retorna  $f^*$ 

```

4. Experimentos Computacionais

[Glomvik Rakke et al. 2012] apresentou 240 instâncias de teste, adaptadas das oito instâncias clássicas da biblioteca TSPLIB [Reinelt 1991], com o número de portos variando entre 14 e 48. As instâncias são: *burma14*, *ulysses16*, *ulysses22*, *fri26*, *bayg29*, *gr17*, *gr21* e *gr48*. Cada grupo de instância é dividido em 30 ocorrências separadas em 3 sub-grupos com 10 ocorrências cada, apresentando variações do limite de calado em 10%, 25% e 50% dos portos.

As tabelas a seguir ilustram um comparativo entre os resultados computacionais dos algoritmos propostos e a solução exata proposta por [Glomvik Rakke et al. 2012]. Na primeira coluna de cada tabela, tem-se o nome da instância que foi analisada. Em seguida, a solução exata e seu tempo de processamento. A próxima coluna oferece o resultado guloso, cujo tempo de processamento não é levado em conta pois as próximas

duas soluções possuem a solução gulosa como inicial. Logo após, vê-se o resultado de uma busca local, no caso deste trabalho o *swap*, que consiste no melhor resultado de cem trocas aleatórias de duas cidades na rota gulosa. Por fim, pode-se observar os melhores resultados da Busca Tabu e seu tempo de processamento.

Resultados em negrito na coluna da Busca Tabu indicam situações onde encontrou-se a melhor solução possível. Na tabela 1, tem-se os resultados para a instância *burma14*. Na tabela 2, da instância *ulysses16*. Na tabela 3, da instância *ulysses22*. Na tabela 4, da instância *fri26*. Na tabela 5, da instância *bayg29*. Na tabela 6, da instância *gr17*. Na tabela 7, da instância *gr21*. Por fim, na tabela 8, da instância *gr48*.

Os experimentos foram realizados em uma máquina com processador Inter(R) Core(TM) i5-5200U, 2.2 GHz, com 8GB de RAM e sistema operacional Ubuntu 18.10. O código fonte foi escrito em linguagem C.

Em instâncias com menor número de portos, como *burma14*, *ulysses16* e *gr17* a Busca Tabu obteve melhor desempenho, chegando à melhor solução em vários casos, e mesmo quando não chegava ao ótimo estava muito próximo deste.

Nas instâncias *ulysses22* e *bayg29* a Busca Tabu obteve desempenho mediano, melhorando a solução gulosa e solução de *swap*, mas chegando na melhor solução possível somente uma vez.

Nas demais instâncias, a Busca Tabu obteve seu pior desempenho, em alguns casos não conseguindo melhorar a solução gulosa ou a solução de *swap*. As melhorias que obteve foram pequenas e ainda estavam longe da melhor solução.

Instâncias	BCP		Método Guloso		Swap		Busca Tabu	
	Sol.	T(s)	Sol.	T(s)	Sol.	T(s)	Sol.	T(s)
burma14_10_1	3416	0.38	3814	-	3535	0.002	3416	0.297
burma14_10_2	3323	0.72	4048	-	3769	0.003	3416	0.306
burma14_10_3	3323	0.41	4820	-	4291	0.003	3346	0.302
burma14_10_4	3751	0.66	4048	-	3814	0.003	3814	0.297
burma14_10_5	3323	0.38	4863	-	4207	0.003	3611	0.299
burma14_10_6	3323	0.65	4048	-	3769	0.002	3416	0.302
burma14_10_7	3323	0.47	4048	-	3769	0.002	3416	0.307
burma14_10_8	3346	0.37	4291	-	3911	0.002	3513	0.294
burma14_10_9	3416	0.52	4048	-	3814	0.003	3416	0.294
burma14_10_10	3323	0.66	4048	-	3814	0.003	3416	0.307
burma14_25_1	4036	0.34	4217	-	4175	0.003	4175	0.277
burma14_25_2	3465	0.37	4841	-	4569	0.003	3692	0.292
burma14_25_3	3336	0.31	4144	-	4131	0.002	4131	0.278
burma14_25_4	3696	0.36	3839	-	3839	0.003	3794	0.272
burma14_25_5	3346	0.45	4048	-	3902	0.002	3416	0.298
burma14_25_6	3610	0.32	5086	-	4797	0.002	4362	0.285
burma14_25_7	3346	0.37	4291	-	3911	0.002	3620	0.294
burma14_25_8	3371	0.45	4214	-	3935	0.002	3478	0.298
burma14_25_9	3834	0.38	4296	-	4135	0.002	3909	0.286
burma14_25_10	3928	0.35	4080	-	4038	0.002	4025	0.277
burma14_50_1	4412	0.32	4687	-	4687	0.003	4425	0.260
burma14_50_2	3748	0.33	4250	-	4216	0.003	4169	0.278
burma14_50_3	3870	0.33	4644	-	4644	0.002	4161	0.271
burma14_50_4	3323	0.31	4107	-	4076	0.003	3942	0.284
burma14_50_5	3524	0.43	4164	-	3922	0.003	3524	0.270
burma14_50_6	3846	0.32	4425	-	4425	0.002	4276	0.279
burma14_50_7	3408	0.37	4040	-	4018	0.001	4018	0.290
burma14_50_8	3506	0.38	3988	-	3838	0.002	3506	0.232
burma14_50_9	4519	0.34	5466	-	5324	0.003	5292	0.222
burma14_50_10	4467	0.32	4716	-	4674	0.003	4674	0.215

Tabela 1. Instância *burma14*

Instâncias	BCP		Método Guloso		Swap		Busca Tabu	
	Sol.	T(s)	Sol.	T(s)	Sol.	T(s)	Sol.	T(s)
ulysses16_10_1	6859	33.88	9600	-	8073	0.003	6878	0.274
ulysses16_10_2	6859	46.83	9988	-	8425	0.003	6900	0.280
ulysses16_10_3	6859	72.6	9988	-	9056	0.003	6900	0.282
ulysses16_10_4	6859	33.04	9876	-	8033	0.003	6859	0.282
ulysses16_10_5	6859	60.21	9998	-	8435	0.003	6895	0.281
ulysses16_10_6	6951	32.58	9954	-	8360	0.003	7029	0.278
ulysses16_10_7	6859	62.46	9988	-	8168	0.003	6900	0.285
ulysses16_10_8	6859	40.66	10008	-	8481	0.003	7001	0.277
ulysses16_10_9	6859	75.11	9876	-	8033	0.003	6859	0.287
ulysses16_10_10	6859	48.59	9988	-	8168	0.003	6952	0.282
ulysses16_25_1	6890	49.57	8173	-	8110	0.002	7234	0.275
ulysses16_25_2	6859	3.98	10041	-	9430	0.002	7987	0.265
ulysses16_25_3	6859	21.95	9929	-	8581	0.003	7702	0.272
ulysses16_25_4	7401	3.52	9550	-	8784	0.003	8264	0.268
ulysses16_25_5	7671	11.1	10021	-	9121	0.003	7768	0.272
ulysses16_25_6	7029	10.16	9704	-	8908	0.003	8731	0.262
ulysses16_25_7	7446	9.81	9988	-	9851	0.003	8457	0.273
ulysses16_25_8	6859	43.05	10068	-	8609	0.003	6977	0.272
ulysses16_25_9	6859	11.32	9876	-	8033	0.003	6859	0.264
ulysses16_25_10	7781	21.31	9984	-	9347	0.003	8209	0.262
ulysses16_50_1	7264	2.09	10356	-	9343	0.003	8577	0.250
ulysses16_50_2	7715	2.6	10107	-	8680	0.003	7781	0.257
ulysses16_50_3	9612	1.2	10690	-	10661	0.003	9614	0.237
ulysses16_50_4	7313	1.26	10466	-	9084	0.003	7313	0.243
ulysses16_50_5	6909	9.93	9876	-	8033	0.003	6909	0.256
ulysses16_50_6	7301	3.66	10992	-	10341	0.003	7444	0.255
ulysses16_50_7	8118	3.06	11295	-	9906	0.003	8213	0.251
ulysses16_50_8	7065	2.4	10040	-	9986	0.002	9628	0.246
ulysses16_50_9	6900	11.81	8490	-	7724	0.003	7503	0.260
ulysses16_50_10	7706	7.97	10143	-	9861	0.003	9508	0.267

Tabela 2. Instância *ulysses16*

Instâncias	BCP		Método Guloso		Swap		Busca Tabu	
	Sol.	T(s)	Sol.	T(s)	Sol.	T(s)	Sol.	T(s)
ulysses22_10_1	7013	35.76	10562	-	8746	0.003	7652	0.359
ulysses22_10_2	7013	26.54	10945	-	9930	0.003	7478	0.355
ulysses22_10_3	7013	28.06	10586	-	9707	0.003	7482	0.363
ulysses22_10_4	7013	19.77	10586	-	8789	0.003	7540	0.357
ulysses22_10_5	7013	39.26	10899	-	9804	0.003	7404	0.351
ulysses22_10_6	7250	40.55	11383	-	10310	0.003	7639	0.353
ulysses22_10_7	7246	40.2	10293	-	10070	0.003	7375	0.351
ulysses22_10_8	7181	55.88	11035	-	10482	0.003	7540	0.343
ulysses22_10_9	7047	18.25	9331	-	8426	0.003	7395	0.341
ulysses22_10_10	7087	17.92	10627	-	8830	0.003	7627	0.355
ulysses22_25_1	7083	30.63	10657	-	9676	0.003	7698	0.350
ulysses22_25_2	7415	19.25	10971	-	10069	0.003	7415	0.333
ulysses22_25_3	8177	21.95	10563	-	9342	0.003	8791	0.342
ulysses22_25_4	7385	28.7	10406	-	8469	0.003	7418	0.338
ulysses22_25_5	7449	23.58	10956	-	10669	0.003	7783	0.343
ulysses22_25_6	7589	32.37	11307	-	10191	0.003	7924	0.335
ulysses22_25_7	7729	23.91	10792	-	9606	0.003	7813	0.334
ulysses22_25_8	7123	17.45	10860	-	10453	0.003	7680	0.340
ulysses22_25_9	7176	27.08	8757	-	8683	0.003	7725	0.341
ulysses22_25_10	7961	22.77	8892	-	8883	0.003	8030	0.340
ulysses22_50_1	8290	24.18	10907	-	10141	0.003	9199	0.328
ulysses22_50_2	7538	16.58	11026	-	10011	0.003	7852	0.324
ulysses22_50_3	8833	21.75	11110	-	11003	0.003	9514	0.312
ulysses22_50_4	9324	38.53	9437	-	9415	0.003	9356	0.304
ulysses22_50_5	8284	46.11	10171	-	10044	0.003	9119	0.314
ulysses22_50_6	7570	10.6	8911	-	8911	0.003	7835	0.315
ulysses22_50_7	7897	25.41	10547	-	8890	0.003	8297	0.322
ulysses22_50_8	9558	20.68	10259	-	9932	0.003	9732	0.296
ulysses22_50_9	9021	48.15	10521	-	10246	0.003	10084	0.310
ulysses22_50_10	7941	15.13	11787	-	10953	0.003	8849	0.316

Tabela 3. Instância *ulysses22*

Instâncias	BCP		Método Guloso		Swap		Busca Tabu	
	Sol.	T(s)	Sol.	T(s)	Sol.	T(s)	Sol.	T(s)
fri26_10_1	937	11.99	1090	-	1055	0.003	1042	0.408
fri26_10_2	937	11.35	1099	-	1076	0.003	1062	0.408
fri26_10_3	1009	14.54	1160	-	1160	0.003	1097	0.404
fri26_10_4	955	12.61	1074	-	1074	0.002	1026	0.406
fri26_10_5	997	21.95	1308	-	1279	0.003	1174	0.407
fri26_10_6	937	8.54	1047	-	1024	0.003	1024	0.407
fri26_10_7	1039	14.24	1224	-	1224	0.003	1224	0.401
fri26_10_8	953	8.33	1247	-	1241	0.003	1169	0.407
fri26_10_9	937	11.18	1102	-	1079	0.003	1002	0.400
fri26_10_10	937	9.98	1102	-	1079	0.003	966	0.415
fri26_25_1	1055	14.7	1226	-	1226	0.003	1226	0.378
fri26_25_2	1201	26.49	1420	-	1406	0.003	1363	0.397
fri26_25_3	1139	24.34	1349	-	1335	0.003	1232	0.387
fri26_25_4	1233	24.35	1404	-	1404	0.003	1304	0.377
fri26_25_5	1017	14.96	1227	-	1216	0.003	1196	0.386
fri26_25_6	1172	14.79	1270	-	1270	0.003	1270	0.374
fri26_25_7	1101	19.01	1475	-	1424	0.003	1396	0.387
fri26_25_8	955	6.46	1224	-	1224	0.003	1168	0.391
fri26_25_9	1081	10.24	1295	-	1207	0.003	1103	0.377
fri26_25_10	1093	9.22	1151	-	1140	0.003	1140	0.385
fri26_50_1	1273	16.64	1486	-	1474	0.003	1458	0.344
fri26_50_2	1045	22.21	1201	-	1201	0.003	1201	0.374
fri26_50_3	1035	12.52	1236	-	1236	0.003	1205	0.386
fri26_50_4	1185	12.13	1457	-	1457	0.002	1448	0.350
fri26_50_5	1185	12.57	1433	-	1428	0.003	1419	0.358
fri26_50_6	1158	9.9	1366	-	1366	0.003	1343	0.360
fri26_50_7	1150	14.74	1296	-	1296	0.003	1273	0.348
fri26_50_8	1441	22.36	1754	-	1742	0.003	1621	0.341
fri26_50_9	1267	28.57	1449	-	1435	0.003	1406	0.343
fri26_50_10	1048	11.96	1097	-	1088	0.003	1083	0.365

Tabela 4. Instância *fri26*

Instâncias	BCP		Guloso		Swap		Busca Tabu	
	Sol.	T(s)	Sol.	T(s)	Sol.	T(s)	Sol.	T(s)
bayg_29_10_1	1610	10.98	1948	-	1920	0.001	1864	0.453
bayg_29_10_2	1654	9.9	2135	-	2135	0.003	2007	0.446
bayg_29_10_3	1753	11.73	2156	-	2114	0.003	1995	0.448
bayg_29_10_4	1622	8.32	2162	-	2105	0.003	1959	0.450
bayg_29_10_5	1645	12.59	2101	-	2088	0.003	2050	0.450
bayg_29_10_6	1622	10.16	2005	-	1942	0.003	1842	0.457
bayg_29_10_7	1833	11.08	2258	-	2232	0.003	2205	0.448
bayg_29_10_8	2114	17.69	2654	-	2614	0.003	2522	0.439
bayg_29_10_9	1628	14.96	1956	-	1943	0.003	1769	0.450
bayg_29_10_10	1655	13.51	1991	-	1991	0.003	1893	0.455
bayg_29_25_1	2027	10.53	2139	-	2139	0.003	2038	0.414
bayg_29_25_2	1655	14.43	2298	-	2238	0.003	2181	0.435
bayg_29_25_3	1827	16.52	2156	-	2128	0.003	2056	0.412
bayg_29_25_4	1799	10.17	2179	-	2175	0.003	2142	0.419
bayg_29_25_5	1709	15.12	2110	-	2110	0.003	1948	0.429
bayg_29_25_6	1841	9.79	2198	-	2198	0.003	2111	0.419
bayg_29_25_7	1805	9.08	2112	-	2084	0.001	2071	0.423
bayg_29_25_8	1718	10.97	2180	-	2152	0.003	1928	0.431
bayg_29_25_9	1683	8.05	1999	-	1999	0.003	1927	0.423
bayg_29_25_10	1862	6.99	2220	-	2220	0.003	2133	0.428
bayg_29_50_1	1928	14.16	2459	-	2459	0.003	2390	0.401
bayg_29_50_2	2255	11.27	2788	-	2777	0.003	2680	0.369
bayg_29_50_3	2093	11.04	2316	-	2316	0.003	2117	0.397
bayg_29_50_4	2019	9.18	2299	-	2299	0.003	2183	0.406
bayg_29_50_5	1785	8.65	2101	-	2084	0.003	2042	0.398
bayg_29_50_6	2340	11.19	2790	-	2758	0.003	2715	0.388
bayg_29_50_7	2400	11.32	2630	-	2612	0.003	2492	0.377
bayg_29_50_8	2204	34.68	2908	-	2890	0.003	2637	0.383
bayg_29_50_9	1987	12.99	2225	-	2225	0.003	2162	0.395
bayg_29_50_10	1899	10.16	2007	-	1993	0.003	1959	0.395

Tabela 5. Instância bayg29

Instâncias	BCP		Método Guloso		Swap		Busca Tabu	
	Sol.	T(s)	Sol.	T(s)	Sol.	T(s)	Sol.	T(s)
gr17_10_1	2153	21.86	2885	-	2876	0.003	2864	0.284
gr17_10_2	2165	28.69	2402	-	2373	0.002	2288	0.288
gr17_10_3	2085	34.6	2199	-	2187	0.003	2085	0.288
gr17_10_4	2590	18.36	2885	-	2873	0.003	2864	0.283
gr17_10_5	2085	23.95	2199	-	2187	0.002	2085	0.289
gr17_10_6	2085	24.9	2199	-	2170	0.003	2085	0.285
gr17_10_7	2085	26.33	2525	-	2428	0.002	2377	0.293
gr17_10_8	2085	2.71	2224	-	2195	0.003	2119	0.287
gr17_10_9	2085	25.06	2199	-	2190	0.003	2085	0.292
gr17_10_10	2085	46.16	2199	-	2170	0.003	2085	0.299
gr17_25_1	2265	3.18	2370	-	2370	0.003	2265	0.271
gr17_25_2	2505	16.48	3233	-	3233	0.003	2814	0.269
gr17_25_3	2270	4.3	2946	-	2934	0.003	2877	0.269
gr17_25_4	2103	20.1	3032	-	2942	0.003	2551	0.278
gr17_25_5	2088	22.16	3046	-	2812	0.003	2154	0.280
gr17_25_6	2160	9.34	2602	-	2503	0.003	2322	0.271
gr17_25_7	2085	15.93	2386	-	2289	0.003	2280	0.283
gr17_25_8	2088	10.47	2224	-	2195	0.003	2119	0.274
gr17_25_9	2138	1.96	2443	-	2405	0.003	2304	0.269
gr17_25_10	2675	8.25	2950	-	2909	0.003	2782	0.268
gr17_50_1	2743	9.46	3196	-	3196	0.003	2916	0.262
gr17_50_2	2216	6.67	2365	-	2365	0.003	2268	0.259
gr17_50_3	3000	1.77	3219	-	3175	0.003	3035	0.248
gr17_50_4	2946	4.73	3723	-	3482	0.003	2948	0.250
gr17_50_5	2205	19.37	2823	-	2726	0.003	2718	0.266
gr17_50_6	2579	2.63	3587	-	3327	0.003	2892	0.261
gr17_50_7	2812	2.01	3516	-	3495	0.003	3141	0.253
gr17_50_8	3014	1.14	3254	-	3140	0.003	3014	0.242
gr17_50_9	3454	1.73	3823	-	3617	0.003	3570	0.248
gr17_50_10	2134	6.52	2319	-	2296	0.002	2134	0.264

Tabela 6. Instância *gr17*

Instâncias	BCP		Método Guloso		Swap		Busca Tabu	
	Sol.	T(s)	Sol.	T(s)	Sol.	T(s)	Sol.	T(s)
gr21_10_1	2707	18.77	3134	-	3104	0.001	3104	0.336
gr21_10_2	3002	8.58	3877	-	3870	0.002	3870	0.326
gr21_10_3	2851	11.62	3408	-	3408	0.003	3401	0.335
gr21_10_4	2760	10.25	3874	-	3836	0.002	3679	0.339
gr21_10_5	2707	8.56	3098	-	3098	0.003	3068	0.338
gr21_10_6	2760	7.76	3465	-	3465	0.003	3465	0.329
gr21_10_7	3093	12.58	3531	-	3501	0.003	3501	0.329
gr21_10_8	2962	15.88	3357	-	3357	0.003	3357	0.331
gr21_10_9	2787	11.9	3164	-	3134	0.003	3134	0.336
gr21_10_10	2707	11.32	3098	-	3098	0.003	3068	0.343
gr21_25_1	2788	9.32	4077	-	3667	0.003	3616	0.323
gr21_25_2	2946	7.43	3713	-	3713	0.003	3713	0.319
gr21_25_3	3109	12.53	3417	-	3417	0.003	3417	0.320
gr21_25_4	2707	6.01	3098	-	3098	0.003	3068	0.328
gr21_25_5	3159	23.31	3431	-	3431	0.003	3431	0.316
gr21_25_6	3159	15.72	3788	-	3758	0.003	3758	0.313
gr21_25_7	2921	9.45	3487	-	3457	0.003	3457	0.317
gr21_25_8	3421	21.77	3835	-	3805	0.003	3640	0.319
gr21_25_9	2709	3.74	3062	-	3062	0.003	3062	0.315
gr21_25_10	2709	6.26	3404	-	3404	0.003	3270	0.333
gr21_50_1	3115	8.34	3438	-	3438	0.003	3438	0.304
gr21_50_2	4041	30.7	4218	-	4218	0.003	4213	0.291
gr21_50_3	3892	7.82	4600	-	4600	0.003	4220	0.293
gr21_50_4	3570	10.64	4571	-	4291	0.003	3882	0.289
gr21_50_5	4132	5.32	4512	-	4512	0.003	4453	0.301
gr21_50_6	3417	7.43	4249	-	4209	0.003	4209	0.300
gr21_50_7	4249	7.71	4680	-	4680	0.003	4460	0.286
gr21_50_8	3296	5.85	4509	-	4464	0.003	3725	0.299
gr21_50_9	4186	8.88	4676	-	4676	0.003	4676	0.288
gr21_50_10	3483	11.47	4676	-	4638	0.003	4502	0.299

Tabela 7. Instância *gr21*

Instâncias	BCP		Método Guloso		Swap		Busca Tabu	
	Sol.	T(s)	Sol.	T(s)	Sol.	T(s)	Sol.	T(s)
gr48_10_1	5046	813.29	5901	-	5889	0.003	5800	0.734
gr48_10_2	5542	341.27	7518	-	7385	0.004	6936	0.705
gr48_10_3	5300	3314.02	6949	-	6910	0.003	6531	0.711
gr48_10_4	5293	7057.36	6236	-	6236	0.003	6119	0.715
gr48_10_5	5679	496.29	7109	-	7108	0.003	7030	0.707
gr48_10_6	5610	41.04	7524	-	7524	0.003	7372	0.703
gr48_10_7	5063	22.79	6201	-	6192	0.003	6070	0.710
gr48_10_8	5103	188.77	7508	-	7508	0.003	7183	0.715
gr48_10_9	5153	892.87	6385	-	6385	0.003	6268	0.714
gr48_10_10	5055	359.47	6204	-	6190	0.003	6082	0.720
gr48_25_1	5524	1073.43	6421	-	6421	0.003	6340	0.673
gr48_25_2	5895	361.9	7273	-	7261	0.004	6961	0.684
gr48_25_3	5754	47.85	7365	-	7365	0.003	7164	0.682
gr48_25_4	5588	126.05	6673	-	6673	0.003	6425	0.669
gr48_25_5	6159	1793.96	7238	-	7238	0.003	6866	0.671
gr48_25_6	5760	3053.47	6625	-	6584	0.003	6415	0.682
gr48_25_7	5955	172.08	7838	-	7838	0.004	7516	0.668
gr48_25_8	5562	1070.86	7124	-	7124	0.003	6946	0.680
gr48_25_9	5792	184.24	7298	-	7298	0.004	7159	0.678
gr48_25_10	6014	730.82	7274	-	7274	0.003	7013	0.673
gr48_50_1	6096	157.47	7372	-	7372	0.004	7031	0.626
gr48_50_2	6629	21.44	8187	-	8178	0.003	7795	0.608
gr48_50_3	5896	132.45	7633	-	7633	0.003	7448	0.622
gr48_50_4	6404	20.75	8528	-	8528	0.004	7706	0.605
gr48_50_5	6617	19.41	8351	-	8351	0.003	8211	0.624
gr48_50_6	8533	66.09	10541	-	10541	0.003	9522	0.611
gr48_50_7	6166	1520.1	7914	-	7914	0.003	7700	0.615
gr48_50_8	6535	20.93	9489	-	9489	0.003	9243	0.595
gr48_50_9	7150	42.73	9398	-	9398	0.003	9269	0.624
gr48_50_10	6331	112.13	7467	-	7467	0.003	7139	0.635

Tabela 8. Instância *gr48*

5. Conclusões

O PCVLC é uma nova variação do clássico PCV aplicado a transportes marítimos, impondo novas restrições de limite de calado, impedindo que uma embarcação fique encalhada. Até o presente momento encontra-se poucas abordagens na literatura para a solução deste problema.

Neste trabalho foi apresentada a abordagem da Busca Tabu com solução inicial gulosa e *swap* como busca local. Os resultados encontrados foram variados, porém foi possível obter melhoria em relação ao método guloso na maioria dos casos.

Sugere-se como trabalhos futuros refinamentos no algoritmo aqui proposto, como substituição da busca local e solução inicial, experimentar uma Lista Tabu baseada em

frequência e outros métodos de diversificação. Incentiva-se também a hibridização entre a Busca Tabu e outras meta-heurísticas.

Referências

- Battarra, M., Pessoa, A. A., Subramanian, A., and Uchoa, E. (2014). *Exact algorithms for the traveling salesman problem with draft limits*. *European Journal of Operational Research*, 235:115–128.
- Gendreau, M. (2003). An introduction to tabu search. In Glover, F. and Kochenberger, G. A., editors, *Handbook of Metaheuristics*, chapter 2. Kluwer Academic Publishers.
- Glomvik Rakke, J., Christiansen, M., Fagerholt, K., and Laporte, G. (2012). *The Traveling Salesman Problem with Draft Limits*. *Computers & Operations Research*, 39:2161–2167.
- Glover, F. (1986). *Future paths for integer programming and links to artificial intelligence*. *Computers & Operations Research*, 13:533–549.
- Machado, V. M. C., Ochi, L. S., and Neves, T. A. (2015). Grasp para o problema do caixeiro viajante com limite de calado. *Anais do 12º Congresso Brasileiro de Inteligência Computacional*, pages 1–6.
- Morais, W. B., Rosa, M., Teixeira, M., and Barbosa, M. A. (2017). O problema do caixeiro viajante com limite de calado: Uma abordagem usando *Simulated Annealing*. *XLIX Simpósio Brasileiro de Pesquisa Operacional*.
- Reinelt, G. (1991). *TSPLIB—A Traveling Salesman Problem Library*. *INFORMS Journal on Computing*, 4(3):376–384.
- Todosijević, R., Mjirda, A., Mladenović, M., and Hanafi, S. (2014). *A general variable neighborhood search for the travelling salesman problem with draft limits*. *Optimization Letters*, 11(6):1–10.