

Algoritmo Bioinspirado como heurística para solução do Problema do Caixeiro Viajante

Guilherme Fonseca e Guilherme Bittencourt

Novembro 2022

1 Introdução

Desde a origem dos primeiros computadores até os mais modernos sistemas digitais, o foco da ciência da computação é resolver problemas de maneira eficiente e automatizada. Porém, com a criação e o avanço da internet, as soluções dos mais diversos problemas se tornaram cada vez mais sofisticadas (Julia Gadelha, 2018). Por isso, atualmente há uma grande competição pelas melhores resoluções das problemáticas encontradas na computação.

Um dos principais grupos de problemas observados pelos cientistas são os chamados NP-Completo. A complexidade de um problema é denominado pelo tempo necessário para sua resolução no pior caso de entrada. Ou seja, no pior dos cenários, o algoritmo analisado demoraria determinada quantidade de tempo para encontrar a solução ótima. Na categoria dos NP-Completo, a função de complexidade de tempo dos mesmos seria, teoricamente, fatorial. Basicamente seria necessário testar todas as possibilidades possíveis para se encontrar a solução ótima do problema (Paulo Feofiloff, 2021).

Entretanto, a grande maioria dos problemas encontrados na realidade pertencem ao conjunto NP-Completo. Sendo assim, seria computacionalmente inviável implementar uma solução que testaria todas as possibilidades na busca por soluções ótimas.

Portanto, dado esse problema, foi inventado o conceito de heurística. Ao se provar que um problema é NP-Completo, não é necessário encontrar a melhor solução global. Ao invés disso, é buscado uma boa solução local. Logo, reduz-se o tempo e o esforço computacional consideravelmente, ao se buscar soluções ótimas locais (Willian Mاتيola, 2020).

Dito isso, o presente trabalho tem por objetivo utilizar uma heurística para a resolução de um problema NP-Completo, modelando o problema ao contexto inserido. Além de apresentar uma minuciosa análise de resultados, a qual exemplificará possíveis maneiras de otimizar os resultados com alterações nos parâmetros de entrada.

2 Problema do Caixeiro Viajante

O problema do caixeiro viajante é um dos problemas NP-Completo mais famosos na computação. Seja um grafo ponderado $G(V, A)$, onde V é o número de vértices e A é o número de arestas. O problema consiste em encontrar um caminho que passe por todos os vértices e tenha peso mínimo possível. De acordo com a definição apresentada acima, é matematicamente provado que se trata de um problema NP-Completo, pois para garantir que uma dada solução é ótima, é necessário testar todas as possibilidades possíveis. Então torna-se viável a obtenção de uma solução próxima da ótima, utilizando uma heurística (Volmir Eugênio, 2015).

3 Algoritmos Bioinspirados

Antes de modelar a solução para o PVC, convém explicar o que são algoritmos bioinspirados. Um algoritmo bioinspirado é uma abordagem computacional que visa resolver problemas utilizando métodos inspirados em eventos naturais observados. Como por exemplo, na ciência de dados, é comum se deparar

com problemas de previsões complexos e com muitas correlações entre os dados. Neste caso, foi criada uma abordagem de implementação chamada redes neurais, que imita o padrão de associatividade do cérebro, a fim de prever condições e eventos a partir de dados consumidos pelo modelo. Do mesmo modo, é possível reproduzir outros eventos naturais como a organização de uma colônia de formigas ou a entropia de uma nuvem de partículas em algoritmos que visam encontrar soluções satisfatórias para problemas (Guilherme Matos Passarini, 2020).

Dito isso, uma solução proposta para o problema do caixeiro viajante é a utilização de um algoritmo que imita a evolução genética das espécies, proposta por Darwin (Charles Darwin, 1859).

4 Modelagem e Solução

Dado um grafo fortemente conectado $G(V, A)$, onde os vértices são as cidades e as arestas são os caminhos que ligam as cidades, todos ponderados. O objetivo é encontrar o caminho de peso mínimo possível que liga todas as cidades. Para isso, foi implementado um algoritmo genético.

O algoritmo genético é constituído das seguintes estruturas:

- Indivíduo - Vetor de V posições, o qual corresponde a um possível caminho.
- População - Conjunto de N indivíduos.

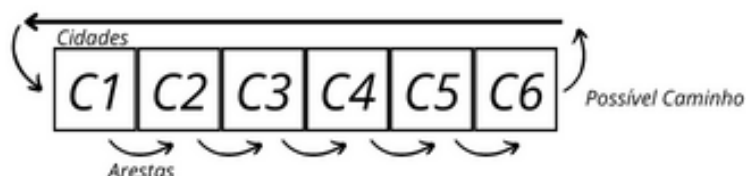


Figura 1: Estrutura do Indivíduo

O número de cidades será abordado na seção de análise de resultados.

O conceito de algoritmo genético se dá pela abordagem de cruzar, mutar e selecionar os melhores indivíduos da população a fim de melhorar constantemente o resultado a cada geração. É criada uma série de indivíduos aleatoriamente. Posteriormente, é avaliada a distância ao se percorrer o caminho que o indivíduo representa. Seleciona-se o melhor, que consiste no mínimo caminho, o qual será mantido nas gerações posteriores. Feita a seleção, cruzam-se os indivíduos com uma estratégia de crossover. Aplica-se uma mutação sobre cada vetor e é construída uma nova geração a partir do cruzamento dos melhores indivíduos da população. Portanto, o resultado nunca é piorado, somente estagnado.

4.1 Listagem de Rotinas

Para a implementação do sistema, foi criada uma rotina que realiza todo o procedimento presente em uma geração, chamada de 'Darwinismo'.

- Cada indivíduo tem as cidades do seu vetor criadas aleatoriamente.
- Logo após, é calculada a distância do caminho representado por cada vetor.
- O indivíduo com o menor caminho é armazenado para ser inserido diretamente na próxima população sem mutação.
- Os melhores indivíduos gerados naquela geração são escolhidos e compõem a próxima população.
- A partir dos novos indivíduos, ocorre o cruzamento entre eles de modo a gerar novas combinações.
- Ao final, os indivíduos são mutados para a ocorrência de diversidade da população.
- Feito o procedimento padrão, o melhor indivíduo da população anterior é recolocado na nova geração.

4.2 Seleção, crossover e mutação

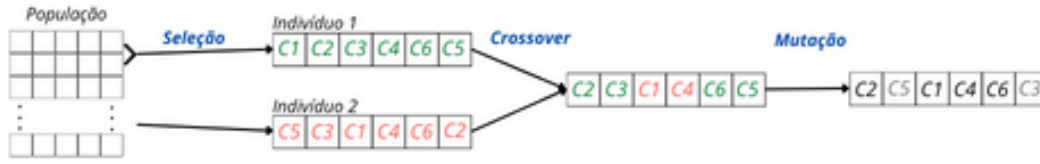


Figura 2: Caminho de Dados

O crossover (cruzamento) é feito da seguinte maneira:

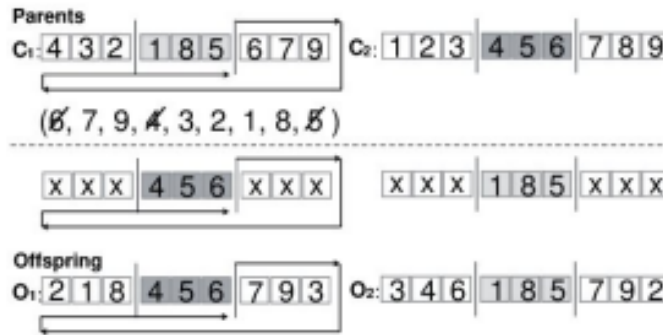


Figura 3: Cruzamento baseado em ordem

Na entrada é determinado o tamanho da população, a condição de parada, a qual se dá por uma tolerância de gerações que não ocorreram modificações no resultado (platô), e a porcentagem de probabilidade de mutação.

5 Análise de Resultados

Para analisar quais os parâmetros afetam mais a eficácia do algoritmo proposto realizamos testes nas bases de dados LAU15 e SGB128 (Ambas encontradas no site <https://people.sc.fsu.edu/~jburkardt/datasets/tsp/tsp.html>) que descrevem a distância entre cidades.

Os testes foram feitos variando os três parâmetros do algoritmo simultaneamente. O tamanho da população (POP) usada foi de 200, 400, 600, 800 e 1000 indivíduos, o platô (PLT) variou entre 100, 200, 300, 400 e 500 iterações sem modificação no melhor caminho, por fim a variação da mutação (MUT) foi de 0%, 5%, 10%, 15% e 20% para ocorrer mutação. As variações foram as mesmas nas duas bases de dados.

5.1 Resultados da base LAU15

A base de dados LAU15 descreve a distância entre 15 cidades abstratas. A solução para o PVC nesta base já está calculada e é igual a 291 unidades.

As tabelas 1 a 5 apresentam os resultados. para cada configuração de parâmetro foram realizados 5 testes e calculado a média (MD) e o desvio padrão (DP) dos mesmos. Para a base LAU15 consideramos um caminho menor que 340 unidades um caminho bom (células em verde na tabela) e um caminho maior que 400 unidades um caminho ruim (células em vermelho na tabela).

POP	PLT	MUT	MD	DP
200	100	0%	387	21,652
200	100	5%	389	10,640
200	100	10%	393	19,900
200	100	15%	420	8,741
200	100	20%	409	7,155
200	200	0%	381	21,288
200	200	5%	377	16,155
200	200	10%	378	30,751
200	200	15%	386	23,199
200	200	20%	397	10,488
200	300	0%	365	14,656
200	300	5%	337	21,647
200	300	10%	380	19,442
200	300	15%	397	29,155
200	300	20%	380	16,401
200	400	0%	371	23,774
200	400	5%	336	19,089
200	400	10%	370	28,678
200	400	150%	392	22,226
200	400	20%	355	31,315
200	500	0%	338	14,642
200	500	5%	361	11,883
200	500	10%	334	10,276
200	500	15%	360	13,107
200	500	20%	359	23,173

Tabela 1: População 200

POP	PLT	MUT	MD	DP
400	100	0%	390	24,249
400	100	5%	403	21,100
400	100	10%	393	18,254
400	100	15%	408	19,774
400	100	20%	404	15,994
400	200	0%	377	20,630
400	200	5%	369	14,758
400	200	10%	374	20,060
400	200	15%	382	13,587
400	200	20%	407	11,145
400	300	0%	340	43,516
400	300	5%	347	24,224
400	300	10%	367	11,171
400	300	15%	362	21,345
400	300	20%	392	21,138
400	400	0%	326	31,382
400	400	5%	341	25,725
400	400	10%	364	12,418
400	400	15%	374	13,349
400	400	20%	372	23,358
400	500	0%	368	18,072
400	500	5%	327	22,768
400	500	10%	362	28,733
400	500	15%	358	25,401
400	500	20%	383	9,381

Tabela 2: População 400

POP	PLA	MUT	MD	DP
600	100	0%	358	33,728
600	100	5%	389	20,871
600	100	10%	398	20,469
600	100	15%	410	4,604
600	100	20%	401	5,367
600	200	0%	376	30,712
600	200	5%	369	12,969
600	200	10%	374	31,765
600	200	15%	379	17,967
600	200	20%	380	10,060
600	300	0%	348	37,531
600	300	5%	369	13,077
600	300	10%	365	35,547
600	300	15%	361	15,767
600	300	20%	372	16,093
600	400	0%	357	16,162
600	400	5%	337	24,536
600	400	10%	365	13,907
600	400	15%	375	12,884
600	400	20%	374	12,466
600	500	0%	341	20,794
600	500	5%	331	25,207
600	500	10%	359	8,270
600	500	15%	363	25,985
600	500	20%	381	12,617

Tabela 3: População 600

POP	PLA	MUT	MD	DP
800	100	0%	390	18,499
800	100	5%	361	4,919
800	100	10%	383	26,340
800	100	15%	396	10,611
800	100	20%	402	12,822
800	200	0%	355	25,799
800	200	5%	352	17,607
800	200	10%	356	16,273
800	200	15%	385	12,704
800	200	20%	390	16,492
800	300	0%	335	12,075
800	300	5%	347	27,756
800	300	10%	348	33,550
800	300	15%	373	22,773
800	300	20%	391	7,694
800	400	0%	358	18,740
800	400	5%	368	13,638
800	400	10%	362	10,412
800	400	15%	360	17,338
800	400	20%	381	16,267
800	500	0%	340	17,181
800	500	5%	340	30,604
800	500	10%	333	34,412
800	500	15%	370	9,879
800	500	20%	368	18,644

Tabela 4: População 800

POP	PLA	MUT	MD	DP
1000	100	0%	378	32,323
1000	100	5%	375	20,620
1000	100	10%	400	15,748
1000	100	15%	377	12,498
1000	100	20%	388	18,740
1000	200	0%	352	15,375
1000	200	5%	339	23,191
1000	200	10%	374	20,115
1000	200	15%	358	28,726
1000	200	20%	383	8,798
1000	300	0%	346	22,018
1000	300	5%	342	32,951
1000	300	10%	360	36,856
1000	300	15%	353	12,696
1000	300	20%	371	9,980
1000	400	0%	361	21,223
1000	400	5%	346	13,914
1000	400	10%	350	18,352
1000	400	15%	372	26,623
1000	400	20%	374	15,447
1000	500	0%	346	17,135
1000	500	5%	320	21,541
1000	500	10%	369	12,538
1000	500	15%	344	13,100
1000	500	20%	379	9,359

Tabela 5

Ao analisar os resultados concluímos que a configuração que gerou os melhores resultados foi POP = 1000; PLA = 500 e MUT = 5% que gerou caminhos de, em média, 320 unidades. Já a configuração que gerou os piores resultados foi POP = 200; PLA = 100; MUT = 15% que gerou em médias caminhos de 420 unidades. Pela variação dos parâmetros, é notório que para a base de dados LAU15 quanto maior o Platô e a População melhores são os resultados do algoritmo, e quanto menor são estes parâmetros piores são os resultados. A Mutação tem comportamento contrario ao do Platô e ao da População, pois quanto menor a mutação melhores são os resultado e quanto maior a população piores são os resultados.

É notório, também, que o Platô e a Mutação são os parâmetros que mais influenciam no resultado do algoritmo, pois mesmo com a população pequena foi possível alcançar resultados bons quando o Plato é alto e a mutação baixa. Um exemplo disto é que a segunda melhor combinação de parâmetros (POP = 400; PLA = 500; MUT = 0%) mesmo tendo uma população pequena gera caminhos de em media 326 unidades, apenas 6 unidades a mais que a melhor configuração.

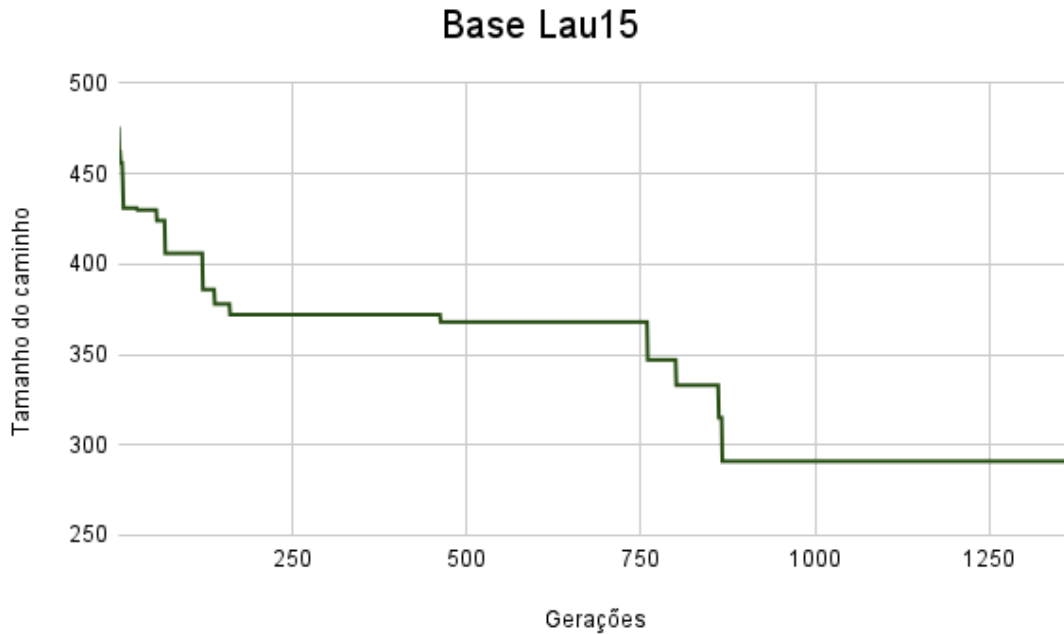


Figura 4: Gráfico da melhor configuração de parâmetros encontrada mostrando a evolução do melhor caminho com o decorrer das gerações

5.2 Resultados da base SGB128

A base de dados SGB128 descreve a distância entre 128 cidades dos EUA e do Canadá que começam com as letras R ou W. A Base de dados não vem com uma solução definitiva para o PVC como a LAU15.

As tabelas 6 a 10 apresentam os resultados encontrados. Como na outra base de dados, para cada configuração de parâmetro foram realizados 5 testes e calculado a média (MD) e o desvio padrão (DP) dos mesmos. Consideramos um caminho menor que 111.000 unidades um caminho bom (células em verde na tabela) e um caminho maior que 130.000 unidades um caminho ruim (células em vermelho na tabela).

Para esta base de dados a melhor configuração que gerou os melhores resultados foi POP = 600; PLA = 500 e MUT 0% que gerou caminhos com em media 105.759 unidades. A pior configuração foi POP = 200; PLA = 100 e MUT = 20% que gerou caminhos de 138.427 unidades.

Os parâmetros nesta base se comportam do mesmo modo que na LAU15. O valor da população e do platô melhoram os resultados quanto são maiores e pioram quando são pequenos. Já a mutação se comporta de maneira contraria, quanto menor o valor melhores são os resultados.

POP	PLA	MUT	MD	DP
200	100	0%	128.859	5.751,853
200	100	5%	135.517	2.405,010
200	100	10%	136.768	1.774,238
200	100	15%	136.156	1.388,696
200	100	20%	138.427	1.245,676
200	200	0%	119.732	9.084,203
200	200	5%	134.767	3.420,283
200	200	10%	133.758	4.707,984
200	200	15%	134.449	1.243,945
200	200	20%	137.073	962,918
200	300	0%	116.190	9.258,389
200	300	5%	129.828	2.313,131
200	300	10%	134.892	1.341,748
200	300	15%	133.298	715,549
200	300	20%	134.395	1.860,541
200	400	0%	110.996	3.426,162
200	400	5%	130.382	2.415,005
200	400	10%	133.689	3.072,980
200	400	15%	134.600	1.530,928
200	400	20%	135.433	2.352,649
200	500	0%	109.150	4.141,792
200	500	5%	127.974	2.677,552
200	500	10%	132.257	2.592,268
200	500	15%	134.022	2.230,921
200	500	20%	133.562	3.446,526

Tabela 6: População 200

POP	PLA	MUT	MD	DP
400	100	0%	128.776	4.397,503
400	100	5%	134.179	2.498,809
400	100	10%	137.111	864,404
400	100	15%	136.414	2.110,559
400	100	20%	137.618	1.115,229
400	200	0%	116.086	9.288,212
400	200	5%	129.977	4.042,865
400	200	10%	135.596	1.325,252
400	200	15%	134.748	2.301,021
400	200	20%	135.703	1.808,466
400	300	0%	111.542	3.237,054
400	300	5%	129.390	3.482,814
400	300	10%	133.713	2.707,603
400	300	15%	132.767	2.288,805
400	300	20%	134.957	1.869,553
400	400	0%	109.081	4.246,773
400	400	5%	127.361	3.157,476
400	400	10%	135.143	1.512,527
400	400	15%	133.081	2.475,667
400	400	20%	135.054	1.425,889
400	500	0%	106.829	5.243,552
400	500	5%	127.491	4.508,799
400	500	10%	132.082	3.340,123
400	500	15%	132.550	1.219,166
400	500	20%	132.237	2.217,530

Tabela 7: População 400

POP	PLA	MUT	MD	DP
600	100	0%	126.595	4.221,061
600	100	5%	134.098	2.769,271
600	100	10%	135.887	1.191,573
600	100	15%	136.297	517,587
600	100	20%	137.086	1.072,895
600	200	0%	113.117	4.347,099
600	200	5%	129.168	5.064,634
600	200	10%	133.747	1.781,145
600	200	15%	134.737	1.625,184
600	200	20%	135.266	435,602
600	300	0%	110.345	4.617,530
600	300	5%	131.600	4.001,067
600	300	10%	132.526	1.956,220
600	300	15%	135.078	1.828,348
600	300	20%	134.060	2.099,667
600	400	0%	112.038	3.659,594
600	400	5%	130.530	1.769,759
600	400	10%	129.470	2.695,999
600	400	15%	130.811	2.401,854
600	400	20%	133.927	1.124,466
600	500	0%	105.759	7.614,472
600	500	5%	127.398	6.042,812
600	500	10%	133.498	1.256,574
600	500	15%	132.384	2.456,157
600	500	20%	132.267	1.725,582

Tabela 8: População 600

POP	PLA	MUT	MD	DP
800	100	0%	123.780	8.340,644
800	100	5%	133.709	3.632,826
800	100	10%	135.494	1.644,373
800	100	15%	135.067	1.335,395
800	100	20%	136.549	1.395,337
800	200	0%	117.496	9.211,438
800	200	5%	127.770	4.065,952
800	200	10%	132.414	2.451,866
800	200	15%	133.315	2.104,119
800	200	20%	134.800	2.129,959
800	300	0%	115.579	5.150,436
800	300	5%	128.119	3.994,604
800	300	10%	132.331	2.557,422
800	300	15%	133.133	2.280,714
800	300	20%	134.918	976,819
800	400	0%	113.352	5.398,181
800	400	5%	129.365	2.515,406
800	400	10%	133.868	1.103,947
800	400	15%	134.350	1.529,395
800	400	20%	134.237	1.307,856
800	500	0%	109.968	4.549,065
800	500	5%	130.710	2.406,977
800	500	10%	131.225	2.466,346
800	500	15%	133.064	1.144,062
800	500	20%	132.887	1.118,613

Tabela 9: População 800

POP	PLA	MUT	MD	DP
1000	100	0%	125.065	2.707,436
1000	100	5%	131.808	1.904,680
1000	100	10%	133.644	2.613,967
1000	100	15%	135.600	2.012,171
1000	100	20%	135.444	1.529,584
1000	200	0%	114.851	5.405,415
1000	200	5%	132.484	1.706,532
1000	200	10%	132.630	2.744,014
1000	200	15%	134.477	2.111,274
1000	200	20%	134.676	1.063,693
1000	300	0%	111.210	4.644,779
1000	300	5%	131.575	1.349,044
1000	300	10%	131.635	1.159,538
1000	300	15%	133.441	1.837,982
1000	300	20%	134.698	543,968
1000	400	0%	115.051	7.566,568
1000	400	5%	127.402	5.204,101
1000	400	10%	132.138	2.903,112
1000	400	15%	132.166	2.704,764
1000	400	20%	133.639	1.763,470
1000	500	0%	110.977	5.822,487
1000	500	5%	127.881	3.593,558
1000	500	10%	130.391	2.652,531
1000	500	15%	132.017	2.496,144
1000	500	20%	131.889	2.525,883

Tabela 10

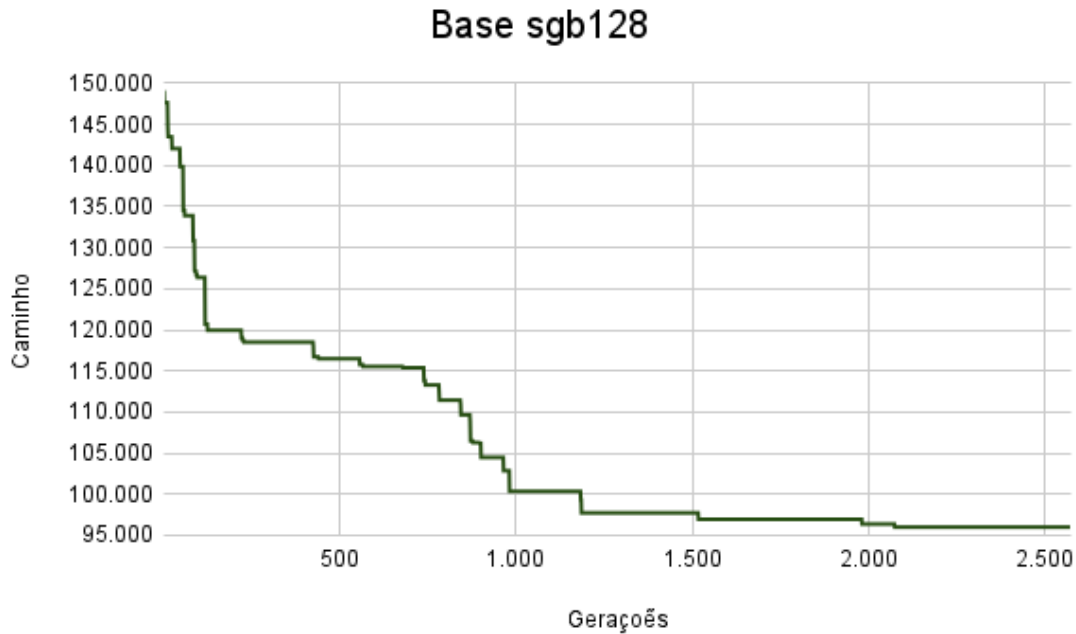


Figura 5: Gráfico da melhor configuração de parâmetros encontrada mostrando a evolução do melhor caminho com o decorrer das gerações

6 Conclusão

Neste trabalho implementamos um algoritmo bioinspirado para o problema do cacheiro viajante, além de fazer uma análise experimental em duas bases de dados (LAU15 e SGB128) para encontrar a combinação de parâmetros que deixa o algoritmo mais eficaz.

Da análise experimental concluímos que uma boa configuração de parâmetro, que faz o algoritmo alcançar melhores resultados, é com Platô 500, Mutação entre 0% a 5% e tamanho da população entre 600 e 1000.

Por fim, é notório que a utilização de um algoritmo bioinspirado para a resolução de problemas NP-Completos se torna eficiente de acordo com a modelagem e o ajuste de parâmetros.

7 Referências

Totas as referências encontradas no texto estão disponíveis em:

<http://www.ic.uff.br/~aconci/evolucao.html>: :text=Em%201946%2C%20surge

%20o%20Eniac,realiza%204.500%20c%C3%A1lculos%20por%20segundo.

https://www.ime.usp.br/~pf/analise_algoritmos/aulas/NPcompleto.html

<https://brasil.uxdesign.cc/o-que-s%C3%A3o-heur%C3%ADsticas-d16e5e8f47df>

https://docs.ufpr.br/~volmir/POI_I12_TSP.pdf

<https://iaexpert.academy/2020/09/08/algoritmos-bioinspirados/>

<http://ecologia.ib.usp.br/ffa/arquivos/abril/darwin1.pdf>

http://www.inf.ufsc.br/~mauro.roisenberg/ine5377/Cursos-ICA/CE-intro_a_post.pdf

<https://people.sc.fsu.edu/~jburkardt/datasets/tsp/tsp.html>