

ANÁLISE DE PONTOS DE ALEATORIDADE COMO ESTRATÉGIA PARA MELHORIA DE ÓTIMOS LOCAIS EM UMA HEURÍSTICA CONSTRUTIVA

ANALYSIS OF RANDOM POINTS AS A STRATEGY FOR LOCAL OPTIMAL IMPROVEMENT IN A CONSTRUCTIVE HEURISTIC

Hugo Vinicius Freitas Losqui* E-mail: hugolosqui@gmail.com
Flávio Henrique Batista de Souza** E-mail: flabasouza@yahoo.com.br

*Centro Universitário de Belo Horizonte – UNIBH, Belo Horizonte, MG

**Universidade Federal de Minas Gerais – UFMG, Belo Horizonte, MG

Resumo: Esse artigo tem como objetivo fazer uma análise sobre o algoritmo NN*. Foi proposto um deslocamento no ponto de aleatoriedade de base por meio de duas propostas, a fim de obter uma otimização em relação ao algoritmo NN*. Após alteração, foram obtidos percentuais de ganhos na margem de até 24% sobre os valores da literatura.

Palavras-chave: Heurística. Problema do Caixeiro Viajante. Nearest Neighbour. Otimização.

Abstract: This article aims to make an analysis about the NN * algorithm. It was proposed a displacement at the point of base randomness by means of two proposals, in order to obtain an optimization with respect to the NN * algorithm. After the change, gains in the margin were obtained from 24% over what was already in the literature.

Keywords: Heuristic. Traveling Salesman Problem. Nearest Neighbor. Optimization

1 INTRODUÇÃO

Durante o experimento de Souza *et al.* (2017), foi apresentada uma heurística construtiva NN*, a qual faz uma variação do algoritmo vizinho mais próximo (NN – *Nearest Neighbour*), onde o segundo vizinho mais próximo é obtido com a interferência de um ponto de aleatoriedade. Tal algoritmo propõe (e obtém) melhorias nos valores obtidos do caminho ótimo em um processo de visitação, porém é possível que se encontre uma melhor resposta através do deslocamento do ponto de aleatoriedade?

Algoritmos heurísticos têm problema de ficar preso em ótimos locais, esse artigo propõe um deslocamento de pontos de aleatoriedade para obtenção de melhores respostas.

O objetivo geral é fazer uma avaliação rigorosa e robusta do algoritmo NN* desenvolvido por Souza *et al.* (2017), a fim de avaliar, através do deslocamento do

ponto de aleatoriedade se existe um ponto de aleatoriedade ótimo, que condicione melhor resposta e comparar com o que se tem na literatura.

Dentre os objetivos específicos tem-se: avaliar o NN* e entender seu funcionamento, realizar experimentos padronizados para uma avaliação de qualidade de respostas com datasets já preconizados na literatura, verificar se há um padrão nas respostas e se há qualidade nas respostas que variaram de acordo com a experimentação.

Esse experimento se justifica, pois, o problema do TSP é muito utilizado nos dias atuais, relacionando-se com a melhora do tempo de deslocamento, diminuição de gastos com tempo e combustível, planejar redes de telecomunicação. De acordo com Cordeiro (2015), sempre busca-se obter resultados melhores e mais rápidos.

Durante o processo de exploração feito por Souza *et al.* (2017), o algoritmo utilizou apenas o segundo vizinho mais próximo. A proposta deste trabalho é variar a partir do terceiro vizinho mais próximo para obter uma melhor resposta, já que não se sabe se o 2º vizinho é a melhor proposta para esse teste.

Assim, alguns questionamentos norteiam essa pesquisa: como funciona a abordagem do NN*? O deslocamento do ponto de aleatoriedade pode oferecer algum ganho? Há pontos de deslocamento ótimo? Qual o comportamento das respostas? Como o deslocamento se comporta em relação a tempo de processamento?

As seções subsequentes deste artigo estão organizadas da seguinte maneira: a segunda seção, demonstra os fundamentos teóricos deste artigo, com análise do algoritmo de referência NN* e proposta de otimização de soluções locais; a terceira seção demonstra a metodologia dos experimentos realizados; a quarta seção apresenta resultados experimentais; e a quinta seção apresenta a conclusão e referências.

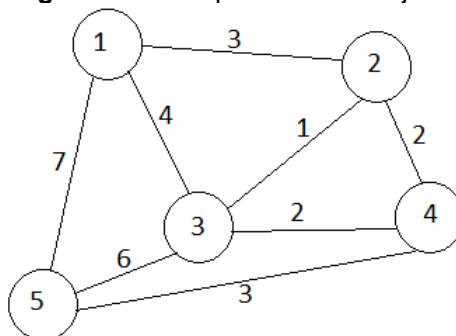
2 FUNDAMENTOS TEÓRICOS

2.1 Traveling Salesman Problem

O problema do TSP (*Traveling Salesman Problem*), caixeiro viajante é um problema NP- difícil que tenta definir a menor rota para visitar uma série de localizações (percorrendo uma vez por cada local) e regressando à localização de

origem (FERREIRA FILHO, 2016). A Figura 1 demonstra um exemplo de um grafo que representa um processo de visitação do TSP, onde deve ser feita uma rota que percorra todos os 5 vértices, sem repetições e retornando ao vértice inicial.

Figura 1 – Exemplo Caixeiro Viajante



Fonte: Autoria própria (2018)

Caixeiro viajante é um problema utilizado nos dias atuais em vários tipos de trabalhos. Ele consiste em encontrar o menor caminho com menor custo (CORMEN, 2015).

2.2 Otimização e Heurísticas

A otimização é um problema matemático que busca uma solução ótima para problemas de programação. O processo de otimização consiste em extrair o melhor resultado possível, reduzindo ou eliminando desperdício de tempo ou recursos, gastos desnecessários, gargalos e erros atingindo o objetivo determinado, podendo ser linear e não linear (ARENALES *et. al.*, 2007).

A heurística pode ser definida como um conjunto de regras, normas e procedimentos utilizados para obter uma melhor solução de um determinado problema (BELFIORE; FÁVERO, 2012). Elas são muito empregadas em problemas de otimização.

Um tipo específico de heurística é definido como heurísticas construtivas, que são as que buscam soluções, etapa por etapa em busca da solução mais viável utilizando critérios heurísticos (LUCENA; PONTES, 2007).

2.3 Trabalhos Relacionados

Como exemplos práticos, tem-se o processo de visitação de frente de lavras de caminhão em minas a céu aberto, como descrito por Souza *et al.* (2017) que implica na diminuição de gastos com combustível. Nesse mesmo ambiente Bastos *et al.* (2011) trabalhou sobre uma resolução de caminhões com um problema de comportamentos incertos em minas a céu aberto, entre eles a variação do consumo de combustível e falhas do equipamento (pneus, emergências) e operações duradouras, que encontramos com frequência no mundo real. A modelagem de resolução de caminhões por meio de uma abordagem probabilísticas torna-se crucial para atender e otimizar seus objetivos específicos.

Johnson (2003) usou colônia de formigas, como exemplo prático da aplicação dessa teoria: Elas encontram o caminho mais curto entre suas fontes de alimentos e sua colônia, este processo se ocasiona através de um hormônio chamado feromônio, onde fica marcado o caminho.

A partir desse processo, as demais formigas passam a segui-la, usando como caminho mais curto porque possui maior concentração do hormônio, pela passagem contínua da maior parte das formigas. O autor desenvolveu um algoritmo que usa formigas virtuais para poder trilhar a extensão por um tempo definido com informações limitadas. E também criou critérios de aptidão para determinar quais serão as formigas com permissão de reprodução. Esse programa desenvolvido se chama Tracker,

Rosen (2009) fez estudos de aplicação de algoritmos dentro da matemática discreta, como avaliação de um caminho mais curto, o autor, em uma das aplicações, usou um problema de viajar em determinados pontos através de vôo, na qual foi feita modelagem usando grafos com associação de pesos em suas arestas, sendo as cidades representada por vértices e os vôos por arestas, com isto, como escolher as combinações com o menor tempo de vôo. Outra aplicação apresentada pelo autor, como obter uma rede com baixo custo para uma melhor comunicação de dados e conexão entre computadores de determinadas cidades.

Assim como em Bastos *et al.* (2011), em Li *et al.* (2018) é analisado o roteamento para os caminhões da água, são usados em minas a céu aberto para a supressão da poeira e a manutenção da estrada do lanço. As estradas de transporte

bem conservadas estendem a vida do componente principal e do pneu e reduzem os custos de manutenção do caminhão de transporte. Ao longo do tempo, número de caminhões de água da frota aumenta para atender a perfis de transporte mais longo e operações de poço mais profundo. Isso exige despesas operacionais adicionais, nomeadamente as despesas com combustível e mão-de-obra. Também levanta a questão de saber se o tamanho da frota de caminhões de água pode ser mantido ou reduzido pela instalação de estações de reabastecimento de caminhões de água adicionais em locais estratégicos dentro ou ao redor da mina.

De modo a minimizar a soma dos custos de atraso de funcionamento e de rega, o problema de programação é modelado como um problema de encaminhamento de arco capacitado com um horizonte infinito. Li *et al.* (2018) usa o fluxo de custo mínimo e as heurísticas baseadas em particionamento para resolver o problema de localizar as estações de reabastecimento de caminhões de água. Um estudo de caso é apresentado para uma grande mina de cobre / ouro a céu aberto na Austrália. Li *et al.* (2018) demonstra, em seus resultados que, sob certas circunstâncias, a construção de uma nova estação de recarga é uma opção melhor do que comprar um caminhão.

Propostas com Metaheurísticas também possuem representações na literatura, como as análises com algoritmos genéticos (AG), que foram realizados em Yu *et al.* (2011). Um AG híbrido adaptativo, juntamente com uma busca Tabu (GA-TS), foi utilizado para a resolução de roteamento ótimo. O algoritmo combina a computação paralela do AG e a otimização global com a habilidade da Busca Tabu (TS) para uma busca local mais rápida. Esta técnica foi experimentada no roteamento de veículos de transporte de material de minas na China Zhengzhou Coal Mine and Power Supply Co. Ltd. Através da experiência, o roteamento de transporte econômico ideal foi criado e os custos econômicos foram reduzidos.

Uma vertente do problema, como a alocação de caminhões, que afeta tanto o roteamento quanto a produtividade da mina, é abordado em Souza *et al.* (2010), que trata do problema do Planejamento Operacional de Minas Abertas com a alocação dinâmica de caminhões. O objetivo do trabalho é otimizar a extração mineral nas minas, minimizando o número de caminhões de mineração utilizados para atender às metas de produção e requisitos de qualidade. É apresentado um algoritmo híbrido que combina características de duas metaheurísticas: *Greedy Randomized Adaptive Search Procedures* e *General Variable Neighborhood Search*. O algoritmo proposto

foi testado usando um conjunto de problemas de dados reais e os resultados foram validados executando o otimizador CPLEX com os mesmos dados. Este solver usou um modelo de programação de inteiros mistos (também desenvolvido neste trabalho). As experiências computacionais mostram que o algoritmo proposto é muito competitivo, encontrando soluções quase ótimas (com um gap menor que 1%) na maioria dos casos, exigindo tempos de computação curtos.

Pode-se afirmar, diante do exposto, que vários ambientes e vertentes de operação assimilam o uso de heurísticas e metaheurísticas.

2.4 Análise do Algoritmo de Referência NN*

Como a versão inicial, foi abordada adicionando um ponto de aleatoriedade no segundo ponto onde o algoritmo NN* utiliza uma conexão de vértices com os vizinhos mais próximos com o nó analisado.

Na modelagem matemática apresentado abaixo, o V^* é um conjunto de vértices não visitados, o nó inicial é preenchido por nó 0 e arquitetado através dos nós, $\{0, i_1, \dots, i_j\}$, onde:

$$i_j = \arg \left(\min \{c_{i_{j-1}k} : k \in V^*\} \right) \quad (1)$$

Onde $u \geq d_{i_1} + d_{i_2} + \dots + d_{i_j}$, e qualquer outro nó $s \in V^*$ é tal que $u < d_{i_1} + d_{i_2} + \dots + d_{i_j} + d_s$.

Este processo será repetido até que aja visitação de todos os nós, de acordo com Souza et. Al. (2017, p5).

Para se obter um resultado satisfatório, esse algoritmo foi testado 100 vezes.

O artigo Souza et. Al. (2017) determina alguns experimentos para obter o menor custo e qualidade, utilizando os algoritmos *Clarke-Wright*, *Nearest Neighbour*, *GRASP (Greedy Randomized Adaptive Search Procedures)* e o NN* não se sabe a qualidade das respostas, são as melhores respostas sobre esse tipo de abordagem, sobre esse tipo de algoritmo pode ter, tanto é que não se sabe se o 2º vizinho for aleatório vai dar a melhor resposta que essa modificação pode fazer.

No experimento foram utilizadas 4 bases teóricas da literatura (berlin52, kroA100, kroA150, kroA200) que, como análise de distâncias, usaram a norma euclidiana para calcular a distância entre os pontos.

A saber a norma euclidiana é definida como: Silva *et al.* (2018) informa que a norma euclidiana é a distância mais curta entre dois pontos.

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (2)$$

Onde a distância d com vetor x, y e o somatório dos vetores onde $(x_i - y_i)^2$ é o resultado da distância total percorrida, percorrendo os pontos da base de dados.

Também foi utilizada base de dados onde consta exclusivamente a distância entre algumas minas do Reino Unido utilizando latitude e longitude reais. (Figura 2).

Figura 2 – Reino Unido (minas e pedreiras)



Fonte: USGS – United States Geological Survey (2018)

Souza *et al.* (2017) usou a distância de Haversine para o cálculo das distâncias reais. De acordo com Shumaker *et al.* (2011), o cálculo se baseia na fórmula:

$$\text{haversine}(d/R) = \text{haversine}(\Delta\phi) + \cos(\Delta\phi_1) + \cos(\Delta\phi_2) \text{haversine}(\Delta\lambda) \quad (3)$$

Onde esfera de raio R (Raio da Terra, em metros) com latitudes φ_1 e φ_2 , que geram uma separação de latitudes $\Delta\varphi$, onde $\Delta\varphi = \varphi_1 - \varphi_2$, e com longitudes λ_1 e λ_2 com a separação $\Delta\lambda = \lambda_1 - \lambda_2$, que apresentam os seus ângulos em radianos. Devido ao arredondamento, esta formula leva uma resposta imprecisa. Quando aplicada na terra, esta formula é usada por aproximação, pois o raio da terra não é uma esfera perfeita.

Analisando o primeiro teste (Tabela 1), feito por Souza *et al.* (2017), foram comparados dois algoritmos CW e NN onde o CW acaba tendo os resultados mais próximos do valor ótimo.

Tabela 1 - Resultados Benchmark

Dataset	Distância	Valor Ótimo	CW	NN
berlin52	Euclidiana	7542	8376	9251
kroA100	Euclidiana	21282	22568	26259
kroA150	Euclidiana	26524	28770	33530
kroA200	Euclidiana	29368	31895	36448
USGS	Haversine(Km)	-	4965	6207

Fonte: Souza *et al.* (2017)

No segundo teste foi adicionado o algoritmo NN*. Os algoritmos NN e NN* foram executados cem vezes para obter a resposta do melhor caminho. Nesse teste o NN* começa a obter um dos melhores resultados (Tabela 2, berlin52), utilizando um ponto de aleatoriedade a partir do segundo ponto do caminho. Apesar do seu custo, tem-se um resultado proveitoso.

Tabela 2 – Resultados de Testes NN*

Dataset	Distância	Valor Ótimo	CW	NN	NN*
berlin52	Euclidiana	7542	8376	9251	8206
kroA100	Euclidiana	21282	22568	26259	27342
kroA150	Euclidiana	26524	28770	33530	32680
kroA200	Euclidiana	29368	31895	36448	38115
USGS	Haversine(Km)	-	4965	6207	5989

Fonte: Souza *et al.* (2017)

No Terceiro teste foi adicionado o algoritmo GRASP* com 50%, que obteve um resultado péssimo comparado com os outros algoritmos adicionados. (Tabela 3).

No último teste (Tabela 4), foi utilizado o GRASP com (0%, 50% e 75% do controle de aleatoriedade), podemos reparar que demonstram valores muito altos,

mas nenhuma das respostas obteve o resultado mais próximo do valor ótimo quanto o CW.

Tabela 3 – Resultados de Testes GRASP*

Dataset	Distância	CW	NN	NN*	GRASP*(50%)
berlin52	Euclidiana	8376	9251	8206	15166
kroA100	Euclidiana	22568	26259	27342	75346
kroA150	Euclidiana	28770	33530	32680	110127
kroA200	Euclidiana	31895	36448	38115	155662
USGS	Haversine(Km)	4965	6207	5989	11993

Fonte: Souza *et al.* (2017)

Tabela 4 – Resultados de Testes GRASP*

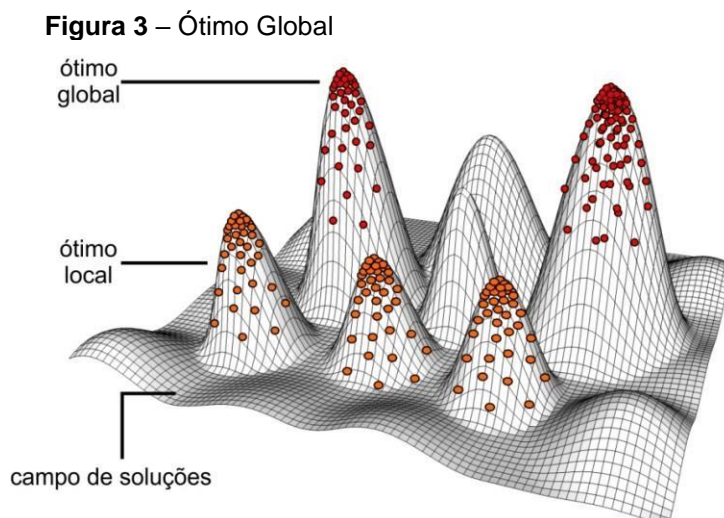
Dataset	Distância	GRASP	GRASP*(50%)	GRASP*(75%)
berlin52	Euclidiana	24931	15166	10640
kroA100	Euclidiana	149292	75346	47946
kroA150	Euclidiana	236565	110127	60376
kroA200	Euclidiana	308280	155662	90224
USGS	Haversine(Km)	24031	11993	8220

Fonte: Souza *et al.* (2017)

3 METODOLOGIA EXPERIMENTAL

Como já dito, durante o processo de exploração feito por Souza *et al.* (2017), o algoritmo utilizou apenas o segundo vizinho mais próximo. A proposta deste trabalho é variar a partir do terceiro vizinho mais próximo para obter uma melhor resposta, já que não se sabe se o 2º vizinho é a melhor proposta para esse teste.

Pode-se ver na Figura 3 o ótimo global e as variações do ótimo local.



Fonte: Scientific Figure on Research Gate, s.d.

Para a realização dos experimentos foram designadas as bases de dados; elaboradas regras de atuação e análise de distâncias; designado o número de ciclos de teste para cada regra e base de dados; definidas propostas de avaliação.

Para os testes são utilizadas as mesmas bases em Souza *et al.* (2017), adicionadas 10 outras bases de referência da literatura, no caso: berlin52, kroA100, kroA150, kroA200, kroB100, kroB150, kroB200, kroC100, kroD100, kroE100, lin105, pr76, pr107, pr124. Para calcular as distâncias d das referências será utilizado a norma euclidiana entre os vértices diferentes. A instância USGS foi utilizada e é baseada no cálculo da distância de Haversine.

Serão delineadas 3 regras de atuação: O primeiro teste será usar a partir do 3º vizinho mais próximo até o 20º vizinho (Regra 1: $v_3 - 20$); o segundo teste utilizando o vizinho que está no meio da base de dados (Regra 2: $v - n/2$) e o terceiro teste consiste em usar o penúltimo vizinho (Regra 3: $v(n - 1)$).

As distâncias serão avaliadas como o processo feito por Souza *et al.* (2017), utilizando a Norma Euclidiana e a distância de Haversine.

Para se obter um resultado satisfatório, esse algoritmo foi testado 100 vezes para cada base de teste.

A primeira proposta (P1) é obter o primeiro ponto fixo, semelhante a uma matriz que deve atender à filiais em uma cidade. A segunda proposta (P2) é obter manter primeiro ponto em aleatório, como no processo original do algoritmo NN, a fim de comparar com os resultados da literatura.

Totalizando uma análise de: $((17 \times 100) + (100) + (100)) \times 15 \times 2$, totalizando 57.000 experimentos. Para elaboração dos experimentos foi utilizado o software Matlab em um computador Intel® Core I5 7ª geração 2,5GHz com 8 Gigabytes de memória RAM e NVIDIA® GeForce® 940MX 2 Gigabytes dedicated Vram.

4 RESULTADOS

4.1 Análise de Respostas

4.1.1 Proposta P1 - (1º Vizinho Fixo)

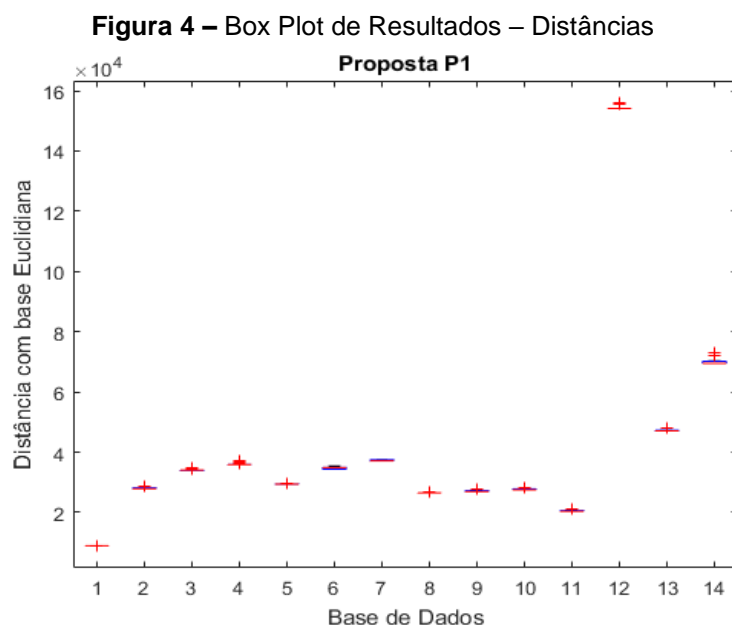
Na primeira proposta foi utilizado a 1ª base de dados fixa, na qual se define

como a primeira cidade berlin52 até USGS, na tabela V, pode-se observar que em destaque, está o menor caminho dentro de cada ponto aleatório. Para achar um caminho a partir do mesmo ponto.

4.1.1.1 Regra 1: $v_3 - 20$

A partir da 3ª vizinho mais próximo foi aplicada a aleatoriedade para a busca do ótimo local. Seguindo a regra, foi feito testes do ponto 3 ao 20, como está demonstrado na Tabela 5. Pra demonstrar a variância e a média das respostas obtidas, foi apresentado na Figura 4 um box plot com as primeiras 14 bases de dados. Pode ser percebido que não houve uma extrema diferença entre os resultados, poucos índices de outliers. Embora cada uma das bases apresentem uma média característica, os pontos máximos e mínimos não se distanciam tanto da média característica da base em questão.

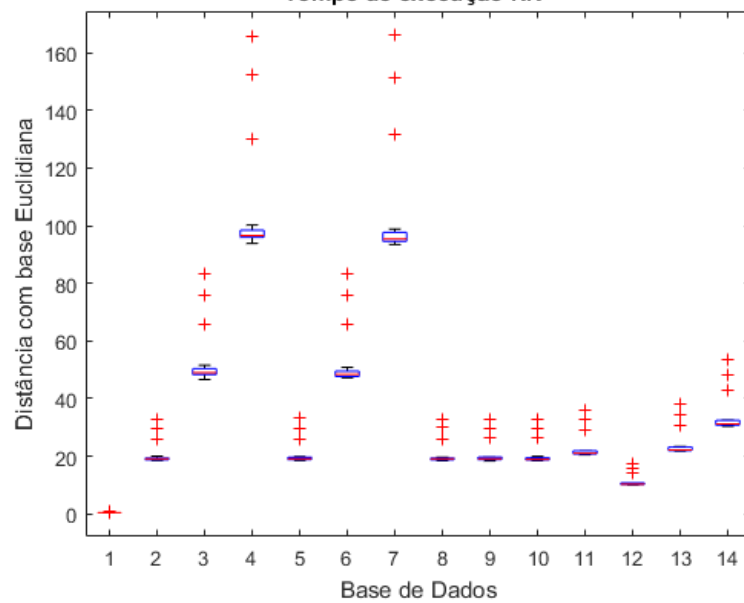
Na Figura 5 nota se que as bases 3, 4, 6 e 7 (kroA150, kroA200, kroB150, kroB200) tiveram um gasto de tempo diferenciado das demais bases, pois o número de execuções feitas em cada base são maiores.



Fonte: Autores

Na Tabela 6, observa-se a relação do tempo (milésimo por segundo) gasto para encontrar a aleatoriedade em cada base de dados, seguindo as regras estabelecidas na regra 1. Em destaque, o menor tempo em cada base de dados.

Figura 5 – Tempo de execução NN*
Tempo de execução NN*



Fonte: Autores

Tabela 5 – Proposta P1 – Regra 1 - Respostas

P.A.	Distância Euclidiana											KM x 10 ³			
	berlin52	kroA100	kroA150	kroA200	kroB100	kroB150	kroB200	kroC100	kroD100	kroE100	lin105	pr76	pr107	pr124	USGS
3	8941	27893	34414	36777	29334	34797	37103	26525	27248	27529	21287	154085	47913	70169	62416
4	8941	27893	33919	35939	29437	34363	37103	26525	26951	27529	20418	155647	47080	69455	62393
5	8941	28160	34066	36809	29817	34363	37266	26525	26951	27529	20940	154085	47080	69455	62393
6	8941	28859	34066	35869	29437	35103	37103	26525	26951	27529	20418	154085	47080	69455	62393
7	8941	28160	34066	35939	29334	34960	37473	26525	27426	27529	20418	154085	47291	69455	62416
8	8941	27893	34066	35869	29817	35103	37103	26525	27248	27529	20654	154085	47080	70169	62393
9	8941	27893	34622	35869	29334	35126	37504	26525	26951	27529	20418	155955	47080	69455	62416
10	8941	28160	34066	35869	29334	34960	37103	26628	26951	27825	20418	154085	47080	69455	62393
11	8941	28160	34066	36351	29334	34797	37103	26525	26951	27529	20418	155647	47080	69455	62393
12	8941	27893	34066	35939	29823	35447	37103	26628	26951	27529	20654	154085	47291	69455	62393
13	9123	27893	34066	35869	29334	34960	37103	26628	26951	27529	20418	154085	47312	69455	62393
14	8941	27893	33919	35869	29437	34363	37463	26525	26951	28195	20418	154085	47080	70311	62393
15	8941	27893	33919	35869	29334	34960	37463	26525	26951	27529	20418	154085	47312	73112	62473
16	8941	27893	34066	35869	29334	34363	37463	26525	26951	28195	20654	154085	47080	71932	62393
17	8941	28652	33919	37140	29437	34960	37103	26525	26951	28353	20654	154085	47080	70169	62393
18	8941	27893	33919	35869	29334	34363	37103	26525	27700	27529	20418	154085	47080	70169	62393
19	8941	27893	33919	35869	29437	34363	37517	26525	27248	27529	20654	154085	47080	69455	62393
20	8941	27893	34365	35939	29437	34797	37266	26628	26951	28195	20940	154085	47080	69455	62393
Mínimos	8941	27893	33919	35869	29334	34363	37103	26525	26951	27529	20418	154085	47080	69455	62393

Fonte: Autores

Tabela 6 – Proposta P1 – Regra 1 - Tempo de Execução em Segundos

Tempo de Execução de NN*															
P.A.	berlin52	kroA100	kroA150	kroA200	kroB100	kroB150	kroB200	kroC100	kroD100	kroE100	lin105	pr76	pr107	pr124	USGS
3	0.5124	19.389	49.915	97.993	19.404	48.408	95.765	19.178	19.696	19.214	21.097	10.223	22.006	30.734	16.083
4	0.4551	18.764	48.823	96.340	18.553	47.616	93.425	18.456	18.390	18.539	20.673	10.041	21.852	30.808	16.120
5	0.4516	18.617	48.358	98.370	18.774	48.413	94.328	18.584	18.874	18.979	20.863	10.173	21.767	30.416	15.780
6	0.4635	19.076	48.660	95.966	18.878	47.624	94.127	18.536	18.584	18.705	20.947	10.330	22.387	31.666	16.500
7	0.4615	19.117	48.740	96.195	19.262	48.019	94.570	18.876	18.787	18.676	20.717	10.087	21.912	30.784	16.177
8	0.4646	19.254	49.228	95.474	18.872	48.772	94.971	18.973	18.952	18.887	21.030	10.215	22.047	31.075	16.310
9	0.4628	19.402	49.624	98.417	19.654	49.563	97.675	19.130	19.170	19.235	21.521	10.477	22.742	31.837	16.587
10	0.4644	19.273	49.363	96.683	19.027	47.694	94.694	19.412	19.619	19.432	21.304	10.441	22.268	31.232	16.470
11	0.4644	19.013	50.375	97.790	19.904	50.898	98.832	19.412	19.332	19.241	21.944	10.746	23.398	32.627	16.757
12	0.4622	19.146	48.340	96.039	19.040	47.799	93.720	18.774	18.941	18.866	21.279	10.391	22.374	31.506	16.259
13	0.4493	18.531	46.611	94.576	18.901	47.246	94.929	19.016	19.048	18.888	20.983	10.290	22.003	30.729	15.984
14	0.4525	18.629	47.674	95.791	19.298	48.493	96.943	19.357	19.064	18.708	20.641	10.154	22.110	30.845	16.084
15	0.6355	25.861	65.770	130.065	26.020	65.759	131.861	25.996	26.194	26.180	29.121	14.249	30.762	42.826	22.240
16	0.7892	32.563	83.118	165.678	33.099	83.457	166.106	32.684	32.737	32.654	36.200	17.642	38.001	53.568	27.946
17	0.7184	29.533	75.745	152.506	29.663	75.822	151.237	30.024	29.777	29.495	32.610	15.923	34.538	48.450	25.405
18	0.4805	20.030	51.612	100.181	19.339	48.894	96.319	19.687	19.807	19.957	22.007	10.744	23.171	32.507	17.137
19	0.4492	18.555	47.130	96.489	18.741	47.441	94.629	18.678	18.753	18.477	20.497	10.084	21.753	30.518	15.855
20	0.4627	18.914	47.256	93.832	18.937	48.259	96.195	19.262	19.380	18.869	20.791	10.079	21.960	31.138	16.245
Mínimos	0.4492	18.531	46.611	93.832	18.553	47.246	93.425	18.456	18.390	18.477	20.497	10.041	21.753	30.416	15.780

Fonte: Autores

4.1.1.2 Regra 2: $v = n/2$

Esta regra foi aplicada no vizinho $n/2$ que corresponde ao ponto aleatório da exata metade dos dados utilizados em relação a aleatoriedade para a busca do ótimo local. Seguindo a regra, foi feito testes do ponto $n/2$, como está demonstrado na Tabela 7. Também observa-se nesta tabela a relação do tempo gasto para encontrar a aleatoriedade em cada base de dados, seguindo as regras estabelecidas na regra 2.

Tabela 7 – Proposta P1 – Regra 2 – Resultados e Tempos de Execução

Base	Resultados	Tempo de Execução
berlin52	8941	0.0105
kroA100	27893	0.0282
kroA150	34066	0.0349
kroA200	36076	0.0409
kroB100	29334	0.0325
kroB150	34363	0.0402
kroB200	37103	0.0454
kroC100	26525	0.0290
kroD100	27700	0.0301
kroE100	28353	0.0292
lin105	20654	0.0247
pr76	154085	0.1846
pr107	47080	0.0596
pr124	71496	0.0860
USGS (km x10 ³)	6292	76.374

Fonte: Autores

4.1.1.3 Regra 3: $v(n - 1)$

Esta regra foi aplicada no vizinho $n - 1$ que corresponde ao penúltimo ponto dos dados utilizados em relação a aleatoriedade para a busca do ótimo local. Seguindo a regra, foi feito testes do penúltimo, como está demonstrado na Tabela 8. Na mesma tabela é vista a relação do tempo gasto para encontrar a aleatoriedade em cada base de dados, seguindo as regras estabelecidas na regra 3.

Tabela 8 – Proposta P1 – Regra 3 – Resultados e Tempos de Execução

Base	Resultado	Tempo de Execução
berlin52	8.941	0.0097
kroA100	27.893	0.0306
kroA150	34.365	0.0373
kroA200	36.777	0.0403
kroB100	29.334	0.0338
kroB150	34.363	0.0420
kroB200	37.103	0.0437
kroC100	26.525	0.0293
kroD100	26.951	0.0338
kroE100	27.529	0.0403
lin105	20.654	0.0260
pr76	155.955	0.1834
pr107	47.080	0.0617
pr124	69.455	0.0731
USGS (km x10 ³)	6.239	64.430

Fonte: Autores

4.1.2 Proposta P2 - 1º Vizinho Aleatório

Na segunda proposta foi utilizado a base em aleatoriedade do 1º vizinho, como é requerido no algoritmo NN original da literatura.

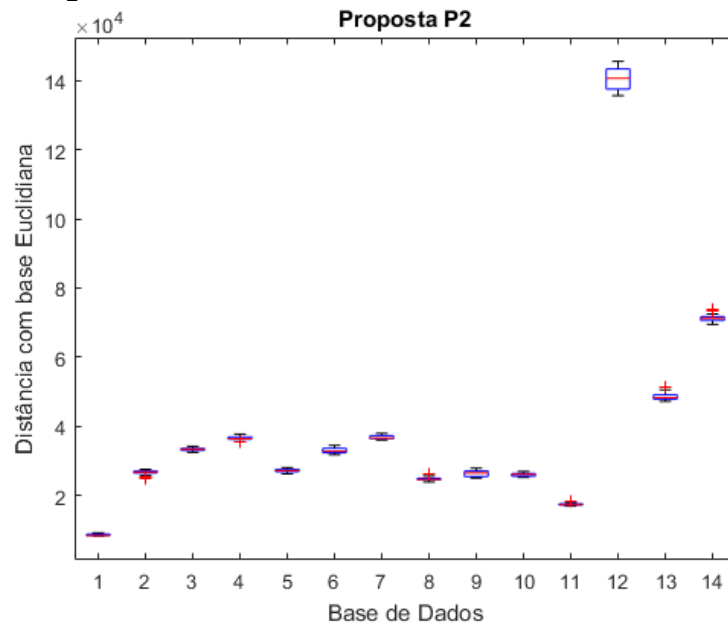
4.1.2.1 Regra 1: $v_3 - 20$

A partir da 3ª vizinho mais próximo foi aplicada a aleatoriedade para a busca do ótimo local. Seguindo a regra, foi feito testes do ponto 3 ao 20, como está demonstrado na Tabela 11.

Pra demonstrar a variância e a média das respostas obtidas, foi apresentado na Figura 6 um box plot com as primeiras 14 bases de dados. Pode ser percebido que não houve uma extrema diferença entre os resultados, a não ser a base de dados 12 onde nota-se que houve uma média com maior distância entre mínimos e máximos.

Na figura 6 nota-se que a base de dados 12 ela desponta não só em posição, porque a unidade de medida é maior, mas, também na variância onde percebe que há uma discrepância dela, não só na unidade de medida, mas em valores máximo e mínimos onde percebe que o box está mais expandido que os outros.

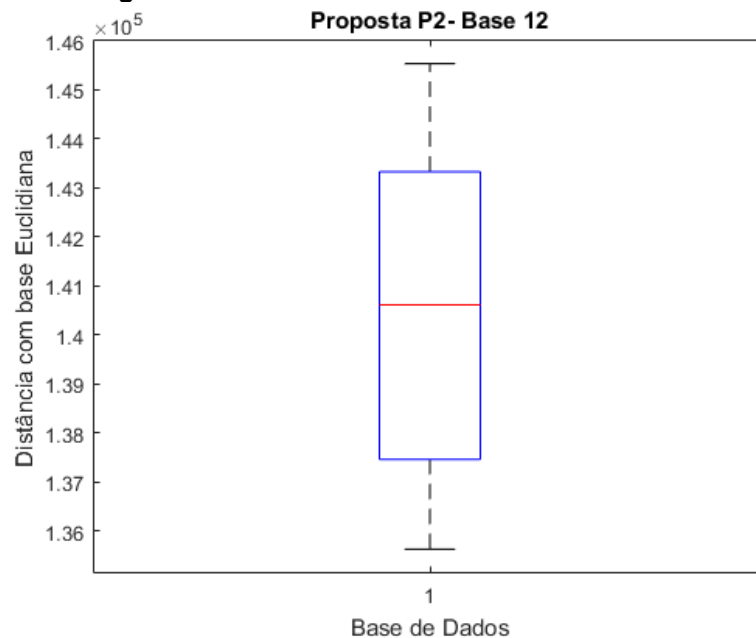
Figura 6 – Box Plot de Resultados



Fonte: Autores

Fazendo uma avaliação singular deles, nota-se o quanto ele é diferenciado dos outros, de acordo com a figura 7.

Figura 7 – Base de Dados 12

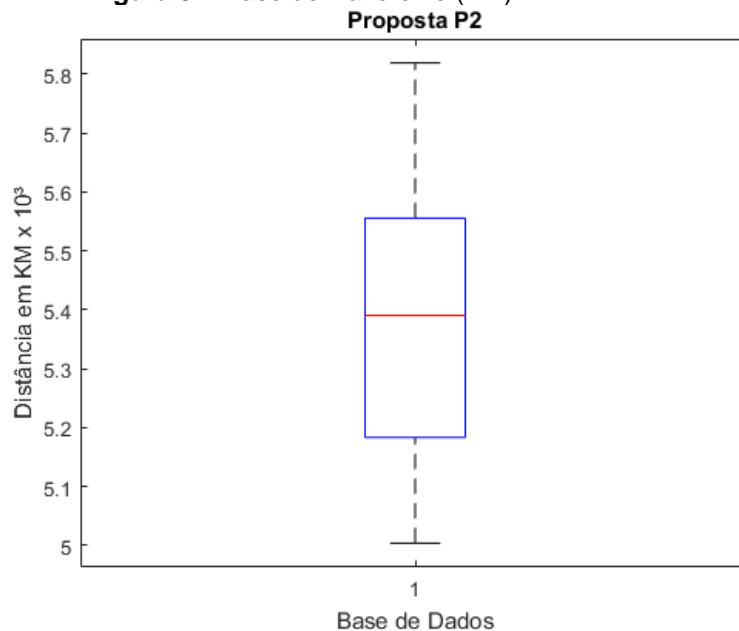


Fonte: Autores

Nota-se que na Tabela 9, a pior distância foi obtida em 5.800 KM e o menor (melhor) resultado foi obtido no ponto de aleatoriedade de número 19 com 5.000 KM, de acordo com a figura 8. Na tabela 10, observa-se a relação do tempo (milésimo por

segundo) gasto para encontrar a aleatoriedade em cada base de dados, seguindo as regras estabelecidas na regra 1.

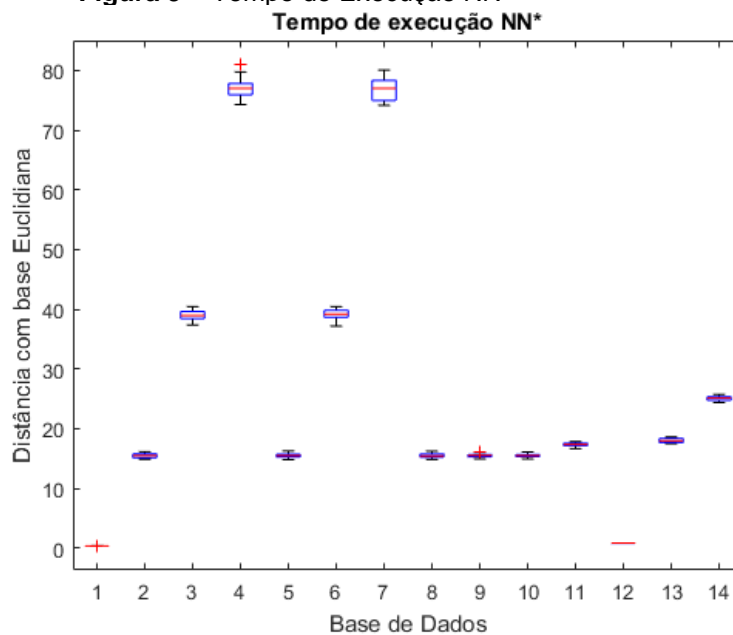
Figura 8 – Base de Haversine (KM)



Fonte: Autores

Em destaque, o menor tempo em cada base de dados. Na figura 9 nota-se que as bases 3, 4, 6 e 7, tiveram um gasto de tempo diferenciado das demais bases, pois o número de execuções feitas em cada base são maiores

Figura 9 – Tempo de Execução NN*



Fonte: Autores

Tabela 9 - Proposta P2 – Regra 1 - Resultados

P.A.	Distância Euclidiana												USGS		
	berlin52	kroA100	kroA150	kroA200	kroB100	kroB150	kroB200	kroC100	kroD100	kroE100	lin105	pr76		pr107	pr124
3	8419	26678	32506	37309	27442	31951	37211	24630	25468	26716	17480	137459	47365	73430	5.508
4	8250	24822	33752	36291	27524	32239	37439	24660	27059	25303	17261	140922	49100	71711	5.114
5	8309	27506	33543	36383	27885	33735	36321	23872	25848	26189	18107	143518	47915	71246	5.158
6	8705	26551	33479	37771	27247	33002	38010	24570	26607	26394	17610	145129	49125	71397	5.442
7	8402	26710	33544	37034	26601	34149	36458	23897	25068	26169	17079	143330	48262	71263	5.511
8	8434	27089	32570	36531	27046	33143	37457	25025	26531	25566	17343	140308	48660	69466	5.263
9	8371	27063	32980	35847	27290	32345	36591	24989	27567	26626	17138	138589	51457	71959	5.556
10	8321	27562	33282	36393	27183	33112	36314	24068	26219	26014	17212	141646	48218	70897	5.183
11	8905	26893	34254	35322	27521	32361	37299	24534	26063	26169	18334	135966	47233	70156	5.819
12	9222	26895	33770	36906	26816	34050	36547	24717	27128	26014	18131	144019	47906	69758	5.564
13	8395	27139	33625	36021	26948	32321	36547	24851	26557	26171	17498	145536	48307	71293	5.368
14	8705	25934	33270	37379	27794	33641	37831	26469	27647	25600	17670	140922	48998	73681	5.204
15	8795	26793	34121	36414	26979	34507	36466	24552	27890	25272	17293	136975	50587	70662	5.761
16	8411	27258	33334	36304	27124	32539	36643	25119	26557	26290	17393	143155	48933	71118	5.183
17	8795	26683	33065	36414	28075	32728	36666	25645	25372	25851	17079	135629	47467	72503	5.616
18	8402	25746	33460	36911	26927	32899	36336	24895	25473	25566	17442	139840	48153	70598	5.414
19	8373	26776	33304	35551	27772	31803	36056	24832	26636	26962	16966	139840	50162	69979	5.004
20	9067	25386	32569	36684	26240	32616	36952	24246	25468	26520	17345	135730	47754	71612	5.241
Mínimos	8250	24822	32506	35322	26240	31803	36056	23872	25068	25272	16966	135629	47233	69466	5004

Fonte: Autores

Tabela 10 – Proposta P2 – Regra 1 – Tempos de Execução

P.A.	Tempo de Execução de NN*														
	berlin52	kroA100	kroA150	kroA200	kroB100	kroB150	kroB200	kroC100	kroD100	kroE100	lin105	pr76	pr107	pr124	USGS
3	0.4233	15.821	38.449	78.136	15.841	38.973	76.866	15.455	15.607	15.505	17.165	0.8355	17.516	24.917	13.250
4	0.3923	15.834	39.795	77.801	15.721	40.024	75.350	15.070	15.226	15.209	17.042	0.8481	18.214	25.681	13.751
5	0.3945	16.103	39.776	75.913	15.577	39.707	79.152	16.022	15.653	15.507	16.911	0.8299	17.621	25.212	13.647
6	0.3821	15.704	39.529	77.535	15.570	39.149	75.449	15.479	15.560	15.522	17.369	0.8485	18.041	24.757	13.149
7	0.3845	15.741	40.208	79.734	15.679	40.009	79.617	16.172	16.046	15.831	17.415	0.8497	17.693	25.351	13.360
8	0.4000	16.144	40.451	77.783	15.405	39.225	77.560	15.760	15.708	15.723	17.559	0.8728	18.626	25.673	13.653
9	0.3665	15.002	38.566	75.053	15.489	39.031	74.962	15.325	15.391	15.249	16.896	0.8350	17.634	24.372	12.585
10	0.3739	15.345	38.920	76.430	14.823	37.195	74.156	15.174	15.374	15.520	17.427	0.8424	17.723	24.418	12.996
11	0.3779	15.590	39.455	76.128	15.792	40.422	78.945	15.389	15.190	14.943	16.663	0.8192	17.627	24.549	13.145
12	0.3764	15.129	38.131	76.839	15.274	38.673	77.145	15.804	15.538	15.414	17.356	0.8422	17.933	25.001	13.194
13	0.3690	14.894	37.361	75.636	15.449	39.807	78.316	15.510	15.759	15.653	17.231	0.8476	18.045	25.176	12.935
14	0.3714	14.972	37.838	74.270	14.996	38.899	75.450	15.275	15.497	15.662	17.607	0.8598	18.397	25.497	12.952
15	0.3875	15.777	39.645	77.417	15.531	39.274	77.272	15.515	15.530	15.563	17.689	0.8707	18.329	25.163	13.415
16	0.3729	15.256	39.006	76.164	15.216	37.884	74.280	14.864	14.950	15.191	17.414	0.8659	18.359	25.109	13.037
17	0.3836	15.474	39.280	80.974	16.257	40.428	80.068	16.244	16.218	16.097	17.867	0.8678	18.466	25.354	13.144
18	0.3793	15.191	38.452	77.145	15.356	38.417	74.167	15.077	15.191	15.351	17.266	0.8395	17.833	25.007	13.147
19	0.3719	14.930	38.288	77.978	15.806	39.842	77.279	15.356	15.580	15.549	17.208	0.8358	17.696	24.729	12.981
20	0.3804	15.499	38.736	75.893	14.843	37.390	74.672	15.241	15.305	15.494	17.637	0.8695	18.455	25.732	13.454
Mínimos	0.3665	14894	37361	74270	14823	37195	74156	14864	14950	14943	16663	0.8192	17516	24372	12585

Fonte: Autores

4.1.2.2 Regra 2: $v = n/2$

Esta regra foi aplicada no vizinho $n/2$ que corresponde ao ponto aleatório da exata metade dos dados utilizados em relação a aleatoriedade para a busca do ótimo local. Seguindo a regra, foi feito testes do ponto $n/2$, como está demonstrado na Tabela 11. Na tabela, observa-se a relação do tempo gasto para encontrar a aleatoriedade em cada base de dados, seguindo as regras estabelecidas na regra

Tabela 11 – Proposta P2 – Regra 2 – Resultados e Tempos de Execução

Base	Resultado	Tempo de Execução
berlin52	8705	0.0095
kroA100	26551	0.0362
kroA150	33479	0.0372
kroA200	37771	0.0461
kroB100	27247	0.0346
kroB150	33002	0.0431
kroB200	38010	0.0455
kroC100	24570	0.0331
kroD100	26607	0.0377
kroE100	26394	0.0295
lin105	17610	0.0220
pr76	145129	0.1883
pr107	49125	0.0569
pr124	71397	0.0908
USGS (km x10 ³)	5.441	65.980

Fonte: Autores

4.1.2.3 Regra 3: $v(n - 1)$

Esta regra foi aplicada no vizinho $n - 1$ que corresponde ao penúltimo ponto dos dados utilizados em relação a aleatoriedade para a busca do ótimo local. Seguindo a regra, foi feito testes do penúltimo, como está demonstrado na tabela 12. Na tabela, observa-se a relação do tempo gasto para encontrar a aleatoriedade em cada base de dados, seguindo as regras estabelecidas na regra 3

Tabela 12 – Proposta P2 – Regra 3 – Resultados e Tempos de Execução

Base	Resultado	Tempo de Execução
berlin52	8.309	0.0095
kroA100	27.506	0.0366
kroA150	33.543	0.0419
kroA200	36.383	0.0387
kroB100	27.885	0.0316
kroB150	33.735	0.0366
kroB200	36.321	0.0399
kroC100	23.872	0.0349
kroD100	25.848	0.0286
kroE100	26.189	0.0293
lin105	18.107	0.0230
pr76	143.518	0.1627
pr107	47.915	0.0832
pr124	71.246	0.0927
USGS (km x10 ³)	5.158	70.401

Fonte: Autores

4.2 Avaliação Comportamental dos Experimentos

Na Tabela 13 encontra-se a consolidação de todas as propostas e regras analisadas nos tópicos anteriores, onde se observa que a proposta 2 ganha destaque na maioria dos casos, sendo a regra 1 que proporcionou esse resultado, com exceção do pr107 e pr124 que a proposta 1 se destacou.

Na Tabela 13, observa-se a relação do tempo gasto para encontrar a aleatoriedade em cada base de dados. Pode se reparar que os melhores tempos ficaram na proposta 2 da regra 3. E os piores tempos ficaram na proposta 1 da regra 1. Percebe-se que a relação entre o resultado e tempo não está relacionado, pois enquanto o resultado da tabela 13 se destaca na proposta 2 da regra 1, o tempo de acordo com a tabela 14 não segue o mesmo padrão, se destacando a proposta 2 da regra 3 que apresentou o tempo mínimo mais vezes.

Tabela 13 – Comparação de Resultados

Base	Proposta			Proposta			Mínimo
	1			2			
	Regra			Regra			
	1	2	3	1	2	3	
berlin52	8941	8941	8941	8250	8705	8309	8250
kroA100	27893	27893	27893	24822	26551	27506	24822
kroA150	33919	34066	34365	32506	33479	33543	32506
kroA200	35869	36076	36777	35322	37771	36383	35322
kroB100	29334	29334	29334	26240	27247	27885	26240
kroB150	34363	34363	34363	31803	33002	33735	31803
kroB200	37103	37103	37103	36056	38010	36321	36056
kroC100	26525	26525	26525	23872	24570	23872	23872
kroD100	26951	27700	26951	25068	26607	25848	25068
kroE100	27529	28353	27529	25272	26394	26189	25272
lin105	20418	20654	20654	16966	17610	18107	16966
pr76	154085	154085	155955	135629	145129	143518	135629
pr107	47080	47080	47080	47233	49125	47915	47080
pr124	69455	71496	69455	69466	71397	71246	69455
USGS (km x10 ³)	6293	6292	6239	5004	5441	5158	5004

Fonte: Autores

Tabela 14 – Tempo em segundos

Base	Proposta			Proposta			Mínimo
	1			2			
	Regra			Regra			
	1	2	3	1	2	3	
berlin52	0.4492	0.0105	0.0097	0.3665	0.0095	0.0095	0.0095
kroA100	18.531	0.0282	0.0306	14.894	0.0362	0.0366	0.0282
kroA150	46.611	0.0349	0.0373	37.361	0.0372	0.0419	0.0349
kroA200	93.832	0.0409	0.0403	74.270	0.0461	0.0387	0.0387
kroB100	18.553	0.0325	0.0338	14.823	0.0346	0.0316	0.0316
kroB150	47.246	0.0402	0.0420	37.195	0.0431	0.0366	0.0366
kroB200	93.425	0.0454	0.0437	74.156	0.0455	0.0399	0.0399
kroC100	18.456	0.0290	0.0293	14.864	0.0331	0.0349	0.0290
kroD100	18.390	0.0301	0.0338	14.950	0.0377	0.0286	0.0286
kroE100	18.477	0.0292	0.0403	14.943	0.0295	0.0293	0.0292
lin105	20.497	0.0247	0.0260	16.663	0.0220	0.0230	0.0220
pr76	10.041	0.1846	0.1834	0.8192	0.1883	0.1627	0.1627
pr107	21.753	0.0596	0.0617	17.516	0.0569	0.0832	0.0569
pr124	30.416	0.0860	0.0731	24.372	0.0908	0.0927	0.0731
USGS (km x10 ³)	15.780	76.374	64.430	12.585	65.980	70.401	64.430

Fonte: Autores

Na tabela 15, contém todos os algoritmos apresentados anteriormente, foi adicionado o algoritmo NN*PR (NN* Baseado em Proposta e Regra), onde constituiu na utilização da aleatoriedade a partir do 3º vizinho.

Os algoritmos NN, NN* e NN*PR foram executados cem vezes para obter a resposta do melhor caminho.

Verificando os resultados apresentados, percebeu-se uma melhora em torno de 24% do NN*PR sobre a literatura NN em todas as bases berlin52, kroA100, kroA150, kroA200 e USGS. Há uma melhora em torno de 20% sobre o NN*.

Com a nova aleatoriedade feito com o NN*PR, percebeu-se uma melhora em torno de 1,5% sobre berlin52 e observou-se que os valores ficam bem próximos do algoritmo CW, em alguns casos ficando apenas menos de 1% de conseguir um melhor resultado como se pode observar no USGS.

Tabela 15 – Resultados

Dataset	Distância	Valor Ótimo	CW	NN	NN*	NN* PR	Resultados
berlin52	Euclidiana	7542	8376	9251	8206	8250 Proposta (2) Regra (1)	Venceu: CW (2%) e NN (12%); Perdeu: NN* (1%)
kroA100	Euclidiana	21282	22568	26259	27342	24822 Proposta (2) Regra (1)	Venceu: NN (6%) e NN* (10%); Perdeu: CW (10%);
kroA150	Euclidiana	26524	28770	33530	32680	32506 Proposta (2) Regra (1)	Venceu: NN (3%) e NN* (1%); Perdeu: CW (13%);
kroA200	Euclidiana	29368	31895	36448	38115	35322 Proposta (2) Regra (1)	Venceu: NN (3%), NN* (8%); Perdeu: CW (11%)
USGS	Haversine (Km)	-	4965	6207	5989	5004 Proposta (2) Regra (1)	Venceu: NN (24%) e NN*(20%); Perdeu: CW (1%)

Fonte: Autores

4.3 Avaliação de aplicação em situações Reais

O problema de minas ao céu aberto apresentado por Souza *et al.* (2017), como mostra a Figura 10, aplica uma solução para o melhor caminho, entretanto pela análise desse artigo, demonstrou-se que pode haver uma otimização, maximizando do desempenho do caminho.

Figura 10 – Minas ao Céu Aberto

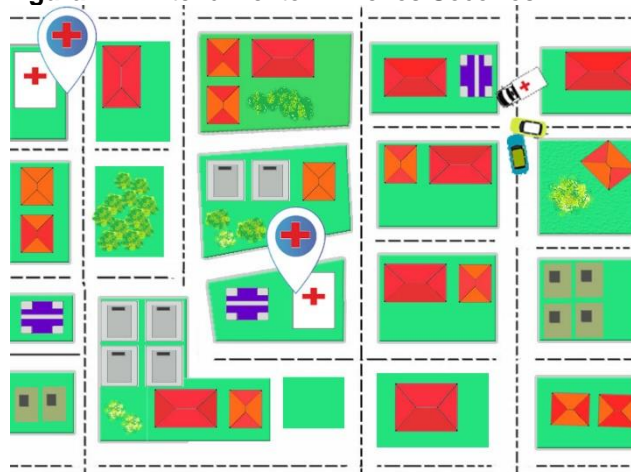


Fonte: Autores

Algumas aplicações se mostraram viáveis com o algoritmo proposto, considerando a lógica das propostas e regras:

Atendimento Médico: Esse algoritmo pode ser utilizado em áreas da saúde como, por exemplo as unidades de atendimento móvel, na qual escolhendo a ambulância que estiver mais próxima do local do acidente e após prestar os primeiros socorros, recalculando o melhor caminho para a unidade atendimento de urgência que mais se aproxima do acidente para levar a vítima.

Figura 11 – Atendimento Primeiros Socorros

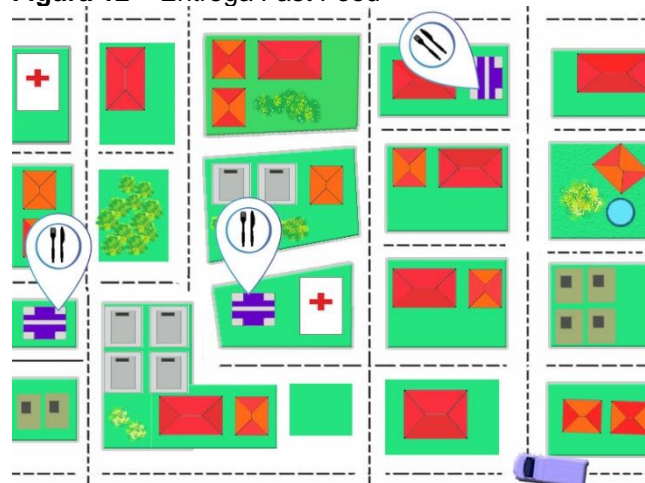


Fonte: Autores

Setor Alimentício: Para restaurante que tem apenas a matriz e há várias entregas a serem feitas pelo mesmo entregador, o cálculo inicial de rota, será refeita a partir do momento que as entregas estão sendo feitas, aplicando uma rota otimizada e agilizando a entrega do pedido. Já para uma rede de fast food, pode ser utilizado

para reabastecimentos de mercadorias. Quando o veículo de entrega sai do centro de distribuição, é calculado uma rota para o mesmo, facilitando e agilizando a entrega em todas as unidades da rede de fast food.

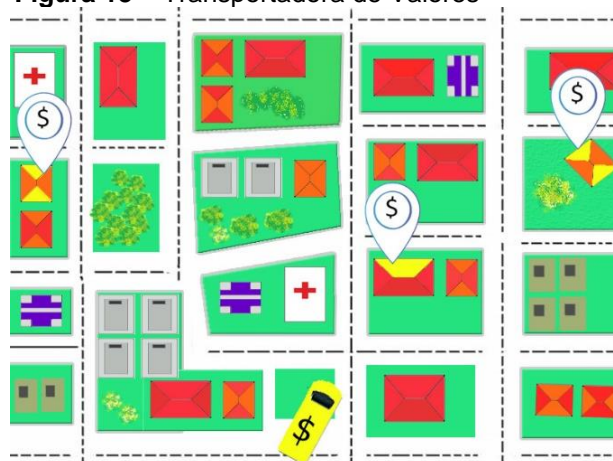
Figura 12 – Entrega Fast Food



Fonte: Autores

Setor Bancário: As transportadoras de valores, prestam a maior parte do serviço para instituições financeiras onde tem que fazer o reabastecimento de valores em vários terminais eletrônicos bancários.

Figura 13 – Transportadora de Valores

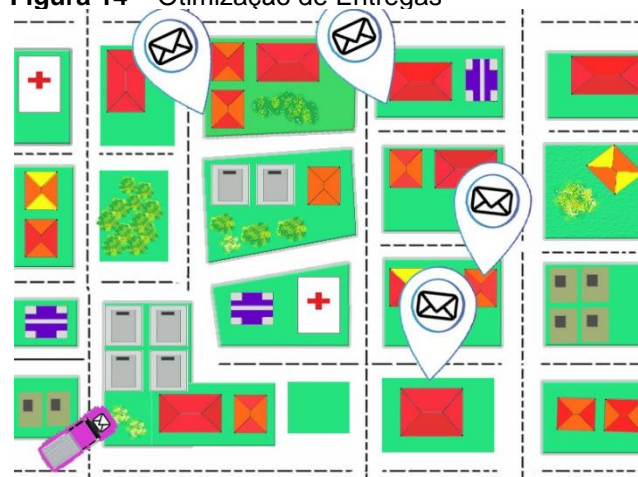


Fonte: Autores

Sempre precisam de um bom caminho para que possam fazer as entregas de um modo mais rápido e seguro. Por isso antes de saírem para fazer a distribuição nos terminais, as transportadoras planejam suas rotas. Com a implementação do algoritmo pode-se escolher um menor caminho para distribuição dos valores.

Transportadora de encomendas e carga fracionada: Atendem usuários que enviam pequenas quantidades de remessas/encomendas produtos que não utilizariam todo o veículo para ser enviada. Normalmente as transportadoras separam os itens por cidades/bairros para aproveitar o mesmo veículo para fazer as entregas, com isso há otimização de rotas e recursos.

Figura 14 – Otimização de Entregas



Fonte: Autores

As extensões das possibilidades desse algoritmo são baseadas no fato de que ele possui um custo de tempo computacional e processamento baixo, não dependente da plataforma. Se tiver as posições previamente definidas, pode-se fazer um aplicativo estratégico para a organização.

3 CONSIDERAÇÕES

No artigo apresentado por Souza *et al.* (2017) que consiste em análise do problema de minas ao céu aberto que é aplicar uma otimização de caminho, o mesmo usou o segundo ponto para análise como o melhor caminho, entretanto este estudo demonstrou que pode haver um melhor caminho a partir do terceiro ponto de aleatoriedade, maximizando o desempenho do caminho de minas.

Após a análise da otimização pôde-se concluir que há um melhor desempenho de ganho em relação a proposta do Souza *et al.* (2017) entre 1% e 24%, demonstrando a interferência nos resultados obtidos. Este ganho de desempenho se deu contra o algoritmo NN da literatura e NN* de Souza *et al.* (2017), além de uma melhora de 2% em relação o algoritmo CW (Clarke-Wright).

Este artigo, veio a contribuir com o estudo e a demonstração que se pode melhorar a otimização de menor caminho, que se pode aplicar essa otimização em vários cenários do cotidiano como empresas de transporte, alimentação e financeiro. Cabe a ressalva de que o algoritmo, em um computador comum, gastou-se cerca de 10 minutos fazendo 57 mil experimentos, levando em consideração que pode ser aplicado em situações reais na qual haverá quantidades maiores de pontos, pode haver a necessidade de um computador com nível de processamento maior ou um servidor dedicado que possa prover um resultado mais rápido, como análises com 10 a 15 mil pontos, ou com mais de 100 ciclos de análises por base de dados.

REFERÊNCIAS

- ARENALES, M., ARMENTANO, V., MORABITO, R., YANASSE, H. **Pesquisa Operacional: para cursos de engenharia**. 2. ed. Rio de Janeiro: Elsevier Editora Ltda, 2007. 564 p. ISBN 978-85-352-7161-4.
- BASTOS, G. S., SOUZA, L. E., RAMOS, F. T., RIBEIRO, C. H. A single-dependent agent approach for stochastic time-dependent truck dispatching in open-pit mining. *In: INTERNATIONAL IEEE CONFERENCE ON INTELLIGENT TRANSPORTATION SYSTEMS (ITSC)*, 14., 2011. *Proceedings [...]*, 2011. p. 1057-1062. <https://doi.org/10.1109/ITSC.2011.6082902>
- BELFIORE, P.; FÁVERO, L. P. **Pesquisa operacional: para cursos de administração, contabilidade e economia**. 1. ed. Rio de Janeiro: Elsevier Editora Ltda, 2012. 277 p. ISBN 978-85-352-3421-3.
- CORDEIRO, J. **Desculpability: elimine de vez as desculpas e entregue resultados excepcionais**. 1. ed. São Paulo: Évora, 2015. 196p. ISBN 978-85-8461-048-8.
- CORMEN, T. **Desmistificando algoritmos**. 1. ed. São Paulo: Elsevier Academic, 2015. 940 p. ISBN 978-0-262-51880-2.
- FERREIRA FILHO, V. J. M. **Gestão de operações e logística na produção de petróleo**. 1. ed. Rio de Janeiro: Elsevier Editora Ltda, 2016. 520 p. ISBN 978-85-352-8037-1
- JOSNSON, S. **Emergência: a vida integrada de formigas, cérebros, cidades e softwares**. 1. ed. Rio de Janeiro: Jorge Zahar Editor, 2003. 231p. ISBN 978-85-7110-739-7.
- LI, J. Q., MIRCHANDANI, P. B., KNIGHTS, P. Water truck routing and location of refilling stations in open pit mines. *In: AUSTRALIAN MINING TECHNOLOGY CONFERENCE SUNSHINE COAST. AUSTRALIA*, 2008. *Proceedings [...]*, 2008.
- LUCENA, A.; PONTES, R. **Aviação comercial controlada por máquinas inteligentes**. 1. ed. Rio de Janeiro: e-papers, 2007. 132 p. ISBN 978-85-765-0110-7.
- PORTAL RESEARCH GATE. **Algoritmos evolutivos como método para desenvolvimento de projetos de arquitetura - Scientific Figure on Research Gate**.S.D.

Disponível em: https://www.researchgate.net/Figura-410-Otimo-local_fig64_310464951

Acesso em: 31 mar. 2018.

ROSEN, K. H. **Matemática discreta e suas aplicações**. 6. ed. Rio de Janeiro: Mc-Graw Hill Brasil, 2009. 982 p. ISBN 978-85-7726-036-2

SHUMAKER, B. P., SINNOTT, R. W. Astronomical computing: 1. Computing under the open sky. 2. Virtues of the haversine. **Sky and telescope**, v. 68, p. 158-159, 1984.

SILVA, C.; RIBEIRO, B. **Aprendizagem computacional em engenharia**. 1. ed. Imprensa Da Universidade de Coimbra: Coimbra University Press, 2018. 290 p. ISBN 978-989-26-1507-3.

SOUZA, F. H. B., LISBOA, A. C., MAIA, C. A., SALDANHA, R. R. Randomization Control in Heuristics and Metaheuristics Applied to the Optimal Path Search in Open Pit Mines. SBPO – SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL, 49. Blumenau, *Proceedings* [...], 2017.

SOUZA, M. J., COELHO, I. M., RIBAS, S., SANTOS, H. G., , L. H. D. C. A hybrid heuristic algorithm for the open-pit-mining operational planning problem. **European Journal of Operational Research**, v. 207, n. 2, p. 1041-1051, 2010.

<https://doi.org/10.1016/j.ejor.2010.05.031>

STEIN, J. **Como a matemática explica o mundo: o poder dos números no cotidiano**. 2. ed. Rio de Janeiro: Elsevier Editora Ltda, 2008. 288 p. ISBN 978-85-352-2945-5

USGS – United States Geological Survey. **Mineral Resources On-Line Spatial**. Disponível em: <https://mrdata.usgs.gov/mineral-operations/map-us.html#home>

Acesso em: 26/03/2018.

YU, S., DING, C., ZHU, K. A hybrid GA–TS algorithm for open vehicle routing optimization of coal mines material. **Expert Systems with Applications**, v. 38, n. 8, p. 10568-10573, 2011.

<https://doi.org/10.1016/j.eswa.2011.02.108>



Artigo recebido em: 06/08/2018 e aceito para publicação em: 16/08/2019

DOI: <http://dx.doi.org/10.14488/1676-1901.v19i3.3336>