

ESTUDO EXPERIMENTAL DOS PONTOS DE ALEATORIEDADE COMO ESTRATÉGIA PARA MELHORIA DE ÓTIMOS LOCAIS EM METAHEURÍSTICA

EXPERIMENTAL RESEARCH THROUGH RANDOMNESS POINTS AS METAHEURISTICS SUBOPTIMAL LOCAL RESPONSES IMPROVEMENT

Flávio Henrique Batista de Souza* - E-mail: flabasouza@yahoo.com.br

Diva de Souza e Silva Rodrigues** - E-mail: divasouz@gmail.com

Vladimir Alexei Rodrigues Rocha* - E-mail: vla.alexei@gmail.com

Renata Duarte Mellim** - E-mail: renatamellim16@gmail.com

Lucas Alberto Queiroz Marcatti** - E-mail: lucasmarcatti@gmail.com

Daniela Ferreira dos Santos** - E-mail: daani.santos01@gmail.com

Ana Gabriela Furbino Ferreira* - E-mail: gabriela.furbino@gmail.com

*Universidade Federal de Minas Gerais (UFMG), Belo Horizonte, Minas Gerais, MG

**Centro Universitário de Belo Horizonte (UNIBH), Belo Horizonte, Minas Gerais, MG

Resumo: Algoritmos de metaheurística são largamente empregados na otimização de problemas em diferentes áreas. Diversos estudos têm, por exemplo, aplicado esse método na otimização da logística de caminhões em mina a céu aberto. Esta pesquisa aborda uma análise experimental da metaheurística GRASP* por meio do deslocamento do ponto de aleatoriedade, com métricas ainda não exploradas na literatura, a fim de se verificar o desempenho do algoritmo em relação a respostas subótimas. Após a análise do comportamento do algoritmo com a alteração dos pontos de aleatoriedade, foi realizado um estudo de seu desempenho em relação a quantidade dos ciclos de processamento. Bases de dados já avaliadas em outras pesquisas, somadas a 10 outras bases de referência presentes na literatura, foram empregadas durante a análise exploratória do método GRASP*. Além disso, os resultados obtidos pelo algoritmo GRASP* foram comparados com a heurística construtiva NN*. Os resultados do presente estudo demonstram que as alterações aplicadas ao método GRASP* proporcionaram ganhos de mais de 24% em desempenho para determinados valores de deslocamento de pontos de aleatoriedade e ganhos de mais de 10% com a variação de números de ciclos. Tal estrutura pode ser implementada para a otimização de estratégias logísticas que podem conduzir negócios de milhões de dólares, como a mineração a céu aberto.

Palavras-chave: Metaheurística. GRASP. GRASP*. Roteamento. Minas a Céu Aberto.

Abstract: Metaheuristic algorithms are widely used in the optimization of problems in different areas. Several studies have, for example, applied this method to the optimization of truck logistics in open pit mining. This research approaches an experimental analysis of the GRASP* metaheuristic through the variation of the randomness point, with metrics not yet explored in the literature, in order to verify the performance of the algorithm in relation to suboptimal solutions. After the analysis of the algorithm convergency with the changes on the randomness points, a study of its performance in relation to the amount of processing cycles was performed. Databases already evaluated in other studies, added to 10 other reference databases present in the literature, were employed during the exploratory analysis of the GRASP* method. In addition, the results obtained by the GRASP* algorithm were compared with the NN* constructive heuristic. The results of this study demonstrate that the changes applied to the GRASP* method provided gains of more than 24% in performance for given values of randomness point and gains of more than 10% with varying numbers of cycles. Such a framework can be implemented for the optimization of logistical strategies that can drive million-dollar businesses, such as open pit mining.

Keywords: Metaheuristics. GRASP. GRASP*. Routing. Open-pit mines.

1 INTRODUÇÃO

Segundo Morais *et al.* (2020), o problema de roteamento de veículos (Vehicle Routing Problem - VRP) pode ser definido como um problema de otimização combinatória. A logística de transportes tem um grande impacto na cadeia de suprimentos de diversos setores e, devido a isso, tem sido o foco de diversos estudos que buscam estratégias para reduzir os custos operacionais desse processo.

Na mineração, o roteamento de caminhões é uma atividade de suma importância em minas a céu aberto, pois o custo logístico do transporte de materiais pode chegar a 60% da operação de extração. O custo do roteamento é um problema que envolve alguns fatores, tais como: consumo de combustível, desgaste dos pneus, emergências etc.

Segundo Souza (2011), a heurística pode ser definida como uma técnica inspirada em processos intuitivos que busca uma boa solução a um custo computacional aceitável, sem se comprometer a garantir a solução ótima como resultado. Por outro lado, a metaheurística consiste em procedimentos indicados a encontrar uma boa solução, porventura ótima, consistindo na aplicação, em cada passo, de uma heurística subordinada, onde deve ser configurada para cada problema específico (RIBEIRO, 1996).

Existem na literatura vários experimentos envolvendo a aplicação de heurística e metaheurística na otimização de rotas em minas a céu aberto, a fim de se obter o melhor desempenho nesse processo. Souza *et al.* (2017) apresentou uma heurística construtiva NN* variando o algoritmo do vizinho mais próximo (NN - *Nearest Neighbour*), cujo segundo vizinho mais próximo é obtido por meio de interferência de um ponto de aleatoriedade. Losqui e Souza (2019), propuseram uma alteração do ponto de aleatoriedade para o terceiro vizinho mais próximo resultando no modelo NNPR*, proporcionando um incremento de 24% no desempenho do algoritmo em relação às pesquisas anteriores.

Souza *et al.* (2017), também propôs uma alteração no modelo metaheurístico GRASP (*Greedy Randomized Adaptive Search Procedure*) para o mesmo problema, denominado como GRASP*, onde a construção aleatória da resposta não possui 100% de aleatoriedade. Uma condição de probabilidades (entre 0 e 100%) de escolha do vizinho mais próximo foi iniciada durante a construção da pesquisa.

Contudo, em ambos os estudos referenciados anteriormente, não foram analisadas todas as possibilidades que o algoritmo GRASP* proposto poderia assumir em um estudo de qualidade das respostas.

Portanto, o objetivo principal do trabalho é propor uma exploração do algoritmo GRASP*, proposto por Souza *et al.* (2017), de forma a demonstrar suas capacidades, explorando as bases utilizadas por Losqui e Souza (2019). Visa-se apresentar um processo de validação de desempenho com bases da literatura, que são relevantes para estudos de otimização de roteamento de veículos, com foco em contribuir com uma apresentação mais detalhada da metaheurística e suas capacidades de operação. Para tanto os objetivos específicos traçados são: descrever a estratégia de experimentação que foi considerada para a faixa de aleatoriedade do método GRASP*; e executar um processo de exploração do algoritmo, avaliando a qualidade de resposta com base no percentual de aleatoriedade proposto.

A importância do presente trabalho se fundamenta no controle e definição da aleatoriedade aplicada no algoritmo GRASP*, que permite que ótimos locais mais eficientes do que em outras propostas possam ser encontrados considerando bases de dados com diferentes perfis.

2 REFERENCIAL TEÓRICO

O Problema do Caixeiro Viajante (do inglês “*Traveling Salesman Problem*” ou TSP), é um problema de otimização combinatória, que consiste na busca de um trajeto com a menor distância entre dois pontos para reduzir os custos de transporte (SOUZA *et al.*, 2017; LOSQUI; SOUZA, 2019). Na formulação desse problema um caixeiro viajante tem de visitar n cidades, começando e terminando na mesma cidade. Além disso, todas as cidades devem ser visitadas seguindo a rota mais curta possível (caminho hamiltoniano) (CORMEN, 2015).

Durante a pesquisa, a abordagem deste problema contemplou análises baseadas em Heurísticas Construtivas, a Metaheurística GRASP (*Greedy Randomized Adaptive Search Procedures*) e a distância de Haversine e roteamento de veículos.

2.1 Heurísticas Construtivas

Segundo Souza *et al.* (2019), uma heurística construtiva tem como objetivo construir uma solução passo a passo, ou seja, elemento por elemento. A organização e escolha de cada elemento a ser inserido varia de acordo com a função escolhida e com o objetivo do problema a ser abordado.

Os métodos construtivos, de um modo geral, são rápidos e seus resultados podem ser utilizados como ponto de partida para algoritmos de otimização e/ou metaheurística (ARROYO, 2002).

Nesta pesquisa as heurísticas construtivas serão empregadas como base de referência para avaliar o grau de contribuição das propostas por metaheurística. A heurística construtiva de referência foi o NN* explorado em Losqui e Souza (2019).

2.2 Metaheurística Greedy Randomized Adaptive Search Procedures (GRASP)

O GRASP (*Greedy Randomized Adaptive Search Procedures*) é uma metaheurística que foi desenvolvida por FEO e RESENDE (1995) como uma forma de obter melhores resultados para um problema de otimização combinatória. Trata-se de uma heurística construtiva que tem como objetivo gerar uma solução para um problema de otimização. O algoritmo é considerado um procedimento de busca guloso e adaptativo pois, a cada passo ele escolhe o melhor candidato e esse candidato fica estático em ótimos locais.

Para melhorar a eficiência do algoritmo, Hart e Shogan (1987) desenvolveram dois mecanismos. O primeiro baseado em cardinalidade, onde os melhores candidatos são adicionados à Lista Restrita de Candidatos (LCR) e o segundo, baseado em valor, onde todas as soluções que tiverem melhores valores do que α vezes o melhor valor, devem compor a LCR, em que $\alpha \in [0,1]$.

Segundo Resende e Ribeiro (2003), o algoritmo é uma metaheurística de multipartida que é composta de suas fases: a construção, onde o objetivo é gerar uma primeira solução por uma heurística construtiva gulosa aleatorizada; e a busca local, onde é efetuada a busca na vizinhança da primeira solução gerada. Essas etapas se repetem por um número máximo de iterações (definidos por um parâmetro) e, no final do processo, a melhor solução obtida na busca é retornada.

Assim como em Souza *et al.* (2017), o algoritmo GRASP tomado como referência (que também foi utilizado para alteração e construção do GRASP*) tem a estrutura mostrada com mais detalhes na Figura 1.

Pode-se observar na Figura 1, que os parâmetros analisados são baseados na função $f(\cdot)$, o grafo a ser analisado $g(\cdot)$, o número máximo de interações $MaxIterations$ e a $Seed$ de N (descrita como a resposta inicial a ser otimizada). No decorrer da execução do algoritmo, acontece o processo de solução gulosa e uma análise da possibilidade de x ser uma resposta aceitável é realizada. Caso a solução encontrada seja a melhor que o apresentado (linha 8), é executado o processo de atualização.

Figura 1 - Pseudocódigo GRASP

Algorithm 1 GRASP Pseudocode

```

1: function GRASP( $f(\cdot), g(\cdot), MaxIterations, Seed$ )
2:    $x_{best} \leftarrow \emptyset; f(x_{best}) \leftarrow +\infty;$ 
3:   for  $k = 1, 2, \dots, MaxIterations$  do
4:      $x \leftarrow ConstructGreedyRandomizedSolution(Seed, g(\cdot));$ 
5:     if ( $x$  not feasible) then
6:        $x \leftarrow repair(x);$ 
7:     end if
8:      $x \leftarrow LocalSearch(x, f(\cdot));$ 
9:     if ( $f(x) < f(x_{best})$ ) then
10:       $x_{best} \leftarrow x;$ 
11:    end if
12:  end for
13:  return  $x_{best};$ 
14: end function

```

Fonte: Souza *et al.* (2017).

A semente (ou $Seed$) é gerada pelo algoritmo de busca gulosa descrito na Figura 2, onde soluções são ordenadas e inseridas como resposta de x , retirando-as de uma lista de pontos candidatos RCL.

Segundo Festa e Resende (2011), há na literatura algumas variações que o GRASP pode apresentar, tais como:

1. **GRASP Reativo:** na fase de construção do GRASP básico, desconsidera-se um mecanismo de aprendizado em um processo sem memória.

Figura 2 - Pseudocódigo GREED

Algorithm 2 GREED Pseudocode

```
1: procedure CONSTRUCTGREEDYRANDOMIZEDSOLUTION(Seed,g(.))
2:    $x \leftarrow \emptyset$ ;
3:   Sort the candidate elements  $i$  according to their incremental cost  $g(i)$ ;
4:   while  $x$  is not a complete solution do
5:     RCL  $\leftarrow$  MakeRCL();
6:      $v \leftarrow$  SelectIndex(RCL,Seed);
7:      $x \leftarrow x \cup \{v\}$ ;
8:     Resort remaining candidate elements  $j$  according to their incremental cost  $g(j)$ ;
9:   end while
10:  return ( $x$ );
11: end procedure
```

Fonte: Souza *et al.* (2017)

2. **Perturbações de Custo:** uma interferência é imputada aos custos originais. Essas perturbações são mais eficientes em situações em que o algoritmo de construção não é muito sensível à aleatoriedade. Essas perturbações podem ocorrer devido a mudanças ou por eliminação dos custos dos nós.

3. **Bias Functions:** baseada na construção de uma LCR sem uma condição igualitária de probabilidade. Trata-se de uma família de funções probabilísticas que geram uma orientação para a seleção de candidatos.

4. **Construção POP:** O *Proximate Optimality Principle* (POP) consiste em uma ideia de que as boas soluções em um nível, provavelmente serão encontradas próximas a boas soluções em um nível adjacente. O objetivo dessa aplicação adicional da busca local é denominado *iron-out* da solução atual e seus componentes com baixo valor de contribuição. Porém, como mostra investigações feitas por experimentos na literatura, observa-se que a execução do POP em cada iteração de construção é extremamente demorada.

5. **Path-relinking:** A partir da conexão de soluções de elite obtidas pela busca de tabu ou busca por dispersão, são gerados e examinados caminhos no conjunto de soluções que encaminham as outras soluções de elite de orientação, na busca de melhores soluções. Isso ocorre por meio de uma seleção de movimentos que inserem atributos contidos nas soluções orientadas. A cada iteração, todas as ações que integram atributos da solução principal são verificados e o movimento de melhor desempenho (ou que menos prejudica) da solução inicial é selecionado.

6. **GRASP Paralelo:** Grande parte das implementações coexistentes do GRASP segue a estratégia de *thread* independente de múltiplos caminhos, tomando

como base a distribuição das iterações sobre os processadores. Duas abordagens podem ser citadas, estas são: a Decomposição por Espaço de Busca (quando o ambiente de pesquisa é dividido em várias áreas e o GRASP é aplicado em cada uma em paralelo) e a Paralelização de Iteração (onde as iterações GRASP são seccionadas e a cada uma das partes é atribuída a um processador).

7. **GRASP em Metaheurística Híbridas:** Diversas hibridizações de GRASP por meio de metaheurística têm sido estudadas e sugeridas na literatura como forma de melhorar a sua estrutura básica, como por exemplo: hibridização com a Pesquisa de Tabu, GRASP Reativo onde a busca local foi fortalecida pela Pesquisa Tabu e GRASP em conjunto com algoritmos genéticos.

Marinakis *et al.* (2009), desenvolveram um estudo a partir da iteração do GRASP com o *Artificial Bee Colony* (ABC), a fim de se otimizar o ajuntamento de N objetos em K clusters. A performance do algoritmo estudado é contraposta com a de outros métodos metaheurísticos populares como algoritmos genéticos clássicos, pesquisa de tabu, GRASP, otimização de colônia de formigas, otimização de *particle swarm* e *honey bees mating*.

A partir destes estudos, Souza *et al.* (2017) propõe uma nova abordagem sobre esta estrutura, a qual denominou GRASP*. Este se fundamenta em um procedimento em que a concepção aleatória da resposta não contém 100% de aleatoriedade. Os autores experimentaram 5 bases de dados com análises pontuais de 50%, 75% e os outros percentuais de aleatoriedade analisados.

Como resultado, observou-se que para a quinta base de dados analisada, houve uma redução de custo da resposta para o problema de aproximadamente 32% para $p = 75%$ ao ser comparado com $p = 50%$. Desta forma, verificou-se que GRASP * (com $p = 75%$), representa uma metaheurística capaz de apresentar uma solução para o problema de visitação de minas a céu aberto com economia de aproximadamente 70% (em Km) com apenas 100 execuções em comparação com o GRASP original.

2.3 Distância de Haversine

Segundo Souza *et al.* (2017) a distância de Haversine é utilizada para realizar o cálculo de distâncias reais. Segundo Shumaker e Sinnott (1984), o cálculo é baseado na seguinte fórmula:

$$\text{hav}(d/R) = \text{hav}(\Delta\phi) + \cos(\phi_1) \cos(\phi_2) + \sin(\phi_1) \sin(\phi_2) \text{hav}(\Delta\lambda) \quad (1)$$

Onde R é a esfera do raio da terra em metros. As latitudes ϕ_1 e ϕ_2 geram a separação da latitude $\Delta\phi$, ou seja, $\Delta\phi$ é igual a $\phi_1 - \phi_2$. Já as longitudes são λ_1 e λ_2 , onde $\Delta\lambda$ é igual a $\lambda_1 - \lambda_2$, que representa seus ângulos em radianos.

A equação de cálculo da distância de Haversine, quando utilizada na terra é feita por aproximação pelo fato de o raio da terra não ser uma esfera totalmente perfeita. A seguir está representada a fórmula completa apresentada em Souza *et al.* (2017).

$$d = 2R \sin^{-1} \left(\sqrt{\sin^2 \left(\frac{\phi_2 - \phi_1}{2} \right) + \cos(\phi_1) \cos(\phi_2) + \sin^2 \left(\frac{\lambda_2 - \lambda_1}{2} \right)} \right) \quad (2)$$

Nesta avaliação, há um erro associado de aproximadamente 0,1% nas respostas obtidas. Contudo este erro pode ser considerado desprezível, uma vez que foi considerada a medida do raio da terra (aproximadamente 6.371 km).

2.4. Trabalhos Relacionados a Roteamento de Veículos

Segundo Alvarenga (2005), entende-se por roteamento de veículos uma ampla gama de problemas relacionados à escolha de rotas a serem percorridas por veículos, objetivando melhorar o custo relacionado a entregas e ou coletas de mercadorias. Cada encomenda deve ser coletada em uma localidade e entregue em outra.

Os veículos não estão necessariamente concentrados em um depósito ou garagem central, mas espalhados em qualquer localidade da região. Cada veículo tem um ponto de partida e um ponto de destino e durante a sua rota entre origem-destino, atenderá a um conjunto de coletas e entregas.

Com esta perspectiva, várias pesquisas já foram conduzidas. Pode ser analisado nos trabalhos já realizados, que tem-se o procedimento de visitação de frente de lavras de caminhões em minas a céu aberto, assim como retratado por Souza *et al.* (2017). Nesse estudo, análises e demonstrações de heurísticas construtivas e metaheurísticas já consolidadas pela literatura foram abordadas e

duas propostas nos algoritmos de Nearest Neighbour, ou vizinho mais próximo (NN) e GRASP, que foram denominados NN* e GRASP*, foram sugeridas. Nesta primeira pesquisa, apenas foram avaliadas 5 bases e com avaliações pontuais de 50 e 75% de aleatoriedade para o GRASP*, o que figura como uma oportunidade de exploração do mesmo algoritmo.

Losqui e Souza (2019), propõem uma análise rigorosa do algoritmo NN* proposto por Souza *et al.* (2017), com objetivo de verificar se com deslocamento do ponto de aleatoriedade há um ponto de aleatoriedade ótimo. Os testes foram realizados nas mesmas bases analisadas em Souza *et al.* (2017), que também serão as bases utilizadas neste trabalho. Após os testes, verificou-se que houve um ganho entre 1% a 24% em relação ao estudo base. Este ganho se deu em função do desempenho comparativo dos algoritmos NN e NN*, além de uma otimização de 2% em relação ao algoritmo Clarke e Wright.

Souza *et al.* (2019), propõe uma análise e implementação de heurísticas para roteirização em uma indústria alimentícia. Os autores realizaram uma comparação dos resultados obtidos com a rota fornecida pela empresa e com a gerada a partir das simulações nas heurísticas NN e Clarke e Wright. Como resultado desta comparação, constatou-se que a implementação do recurso proposto acarretaria em uma redução de aproximadamente 25% nos custos logísticos da empresa.

Aloise *et al.* (2002), propõe a aplicação de modelagem matemática e a metaheurística *Ant Colony Optimization* ou otimização por Colônia de Formigas (ACO) para otimizar o problema de alocação de sondas de produção terrestre - SPT, utilizadas para execução de trabalhos de intervenção em poços de petróleo. O problema proposto no estudo mostrou-se como um caso peculiar de problema de k-Servos e, portanto, referente à classe dos problemas NP-árduos. Os autores ainda realizaram estudos comparativos dos resultados obtidos pela metodologia escolhida aos resultados da aplicação das metaheurísticas, como: algoritmo genético, GRASP e variações do ACO e do GRASP por meio do incremento do mecanismo de *path-relinking* (PR).

3 METODOLOGIA

Para a realização dos experimentos, foram utilizadas quinze bases de dados de referência na literatura, extraídas da TSPLIB, que é uma biblioteca com um

compilado de problemas relacionados ao caixeiro viajante, de diversas fontes e tipos.

Como em Losqui e Souza (2019), as bases utilizadas foram: berlin52, kroA100, kroA150, kroA200, kroB100, kroB150, kroB200, kroC100, kroD100, kroE100, lin105, pr76, pr107, pr124 e USGS. As bases caracterizam um problema do caixeiro viajante com 52, 100, 150, 200, 100,150, 200, 100, 100, 100, 105, 76, 107 e 124 cidades, respectivamente e estavam em medida de distância euclidiana, que é o cálculo da diferença entre dois pontos projetados em um plano.

A base USGS é uma base composta de dados reais de uma mineradora localizada na Inglaterra. Ela foi utilizada e calculada, através da distância de Haversine, que como mencionada, é utilizada para realizar o cálculo de distâncias reais.

Primeiramente, foi realizada a avaliação do GRASP* em todas as quinze bases da literatura que foram utilizadas em Losqui e Souza (2019). Os experimentos realizados, levaram em consideração os pontos de aleatoriedade entre 51% e 90%, resultando em um total de quarenta pontos de aleatoriedade para cada uma das 15 bases de dados. Tal faixa de aleatoriedade se diferencia dos experimentos avaliados por Souza *et al.* 2017, onde foram avaliados especificamente os pontos 0%, 50% e 75% para o controle de aleatoriedade. Com base nisso, observa-se uma margem de exploração para avaliação de possíveis pontos de atração para mínimos locais, uma vez que foram parametrizados apenas 100 ciclos.

Assim como em Souza *et al.* (2017), as primeiras análises foram realizadas através da metaheurística GRASP e em seguida foram realizadas análises quantitativas e qualitativas das respostas com avaliação gráfica do comportamento do algoritmo e comparada com os resultados que foram obtidos no algoritmo NN*, desenvolvido por Souza *et al.* (2017).

Os experimentos começaram a ser executados em 7 de abril de 2021 e finalizados no dia 22 de maio de 2021, levando em média 18 horas por dia de execução. O algoritmo foi testado uma vez para cada base de dados, totalizando uma análise de 600 experimentos. Por fim, as bases que apresentaram melhor performance em tempo de processamento, foram rodadas novamente com o dobro de cliques da análise anterior, a fim de se verificar o impacto do número de ciclos na qualidade das respostas.

Para elaboração dos experimentos foi utilizado o software para cálculo numérico e científico, Matlab® (MATrix LABoratory), em um computador Intel Core I3 2,5GHz com 8 Gigabytes de memória RAM.

4 RESULTADOS

Os resultados da pesquisa foram segmentados em uma primeira análise experimental onde a estrutura foi testada, e então foi realizada uma avaliação crítica com análises quantitativas das respostas obtidas.

4.1. Análise Experimental

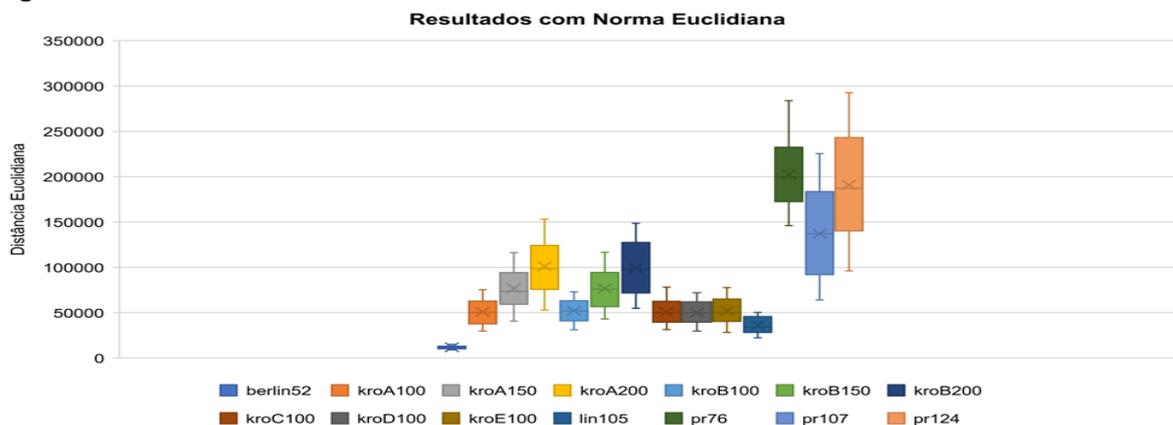
Uma análise experimental foi executada em todas as quinze bases. Pelo fato de as quatorze primeiras bases estarem utilizando a norma euclidiana e a da USGS utilizar a distância de Harversine, esta primeira análise foi dividida em dois gráficos de *boxplot* (caixa estreita). A Figura 3 apresenta a análise das quatorze primeiras bases e a Figura 4 mostra a Base USGS.

Na análise inicial da Figura 3 observa-se que não há a presença de outliers em nenhuma das bases. Sobre a dispersão dos dados, a base berlin52, em que a distância entre os máximos, médias e mínimos (além de sua dispersão) dos valores das respostas é qualitativamente pequena.

As bases kroA100, kroB100, kroD100 e pr107 apresentaram uma simetria na dispersão dos dados como se pode observar por meio da linha da mediana ou segundo quartil que se encontra localizada no centro do gráfico.

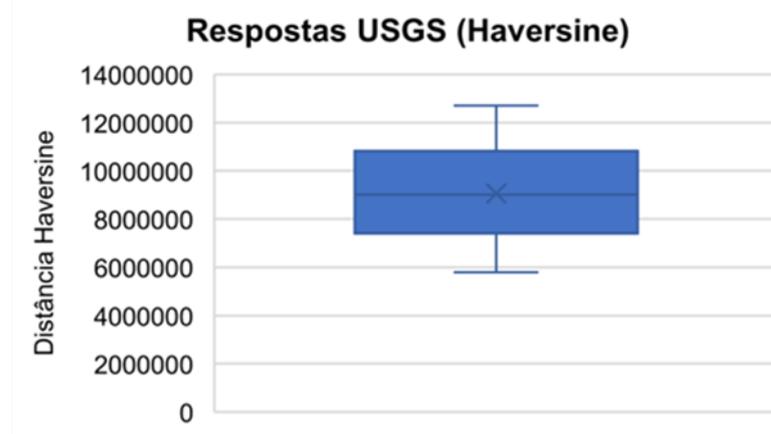
Como descrito no início desta sessão, a mesma análise foi realizada para a base USGS, que se baseia no cálculo da distância de Haversine.

Figura 3 – *Box Plot* de resultados com norma euclidiana



Fonte: Autores.

Figura 4 – Box Plot de resultados USGS distância haversine



Fonte: Autores.

A base pr107 apresentou uma maior variação entre seus pontos máximos, médios e mínimos do que as demais. Já nas bases kroA150, kroA200, kroB150, kroB200, kroC100, kroE100, lin105 pr76 e pr124, tem-se uma distribuição assimétrica positiva dos resultados, em que a linha da mediana encontra-se em todas elas abaixo do centro da caixa estreita, ou seja, mais próximo da linha do primeiro quartil. As bases pr124 e pr76 se destacam pela grande variação dos máximos e mínimos encontrados.

Conforme a Figura 4, verifica-se que há uma certa simetria na dispersão dos dados da base USGS, em função da reta mediana estar localizada quase que ao centro da caixa estreita. Também não há a presença de outliers e a distância entre os valores mínimos e máximos foi de cerca de 50%. Com o intuito de facilitar a visualização dos resultados de cada base, os mesmos foram organizados em uma tabela (Tabela 1).

Em seguida, os valores mínimos encontrados na análise da Tabela 1 foram comparados aos valores mínimos obtidos em Losqui e Souza (2019) com a execução do NN* nas mesmas bases. Na Tabela 2, está representada a análise comparativa entre os mínimos obtidos pelo GRASP* em 100 ciclos e os mínimos obtidos em NN* por LOSQUI e SOUZA (2019) em 100 vezes que o algoritmo foi executado. Na coluna diferença, pode-se observar que em nenhuma das bases o GRASP* ganhou do NN*.

Tabela 1 – Valores mínimos, médios e máximo dos resultados

	Mínimo	Média	Máximo
berlin52	8743	11547,675	15159
kroA100	29568	49779,525	75206
kroA150	40470	75019,6	116248
kroA200	45386	98489,2	153195
kroB100	27145	50892,775	73026
kroB150	40197	74985,825	116469
kroB200	48389	96663,1	148587
kroC100	26340	49729,375	78179
kroD100	26437	48988,975	71938
kroE100	28282	50943,825	77816
lin105	20405	35234,1	50107
pr76	145930	199997,98	283833
pr107	57048	133240,4	225484
pr124	92561	186272,68	292642
USGS	5791600	9056050	12704000

Fonte: Autores.

Conforme os dados dos experimentos expostos no Apêndice I, o tempo de processamento do algoritmo a cada mudança de ponto de aleatoriedade foi alto apenas nas cinco primeiras bases.

Tabela 2 – Comparação dos mínimos GRASP* e NN* de Losqui e Souza (2019)

	Mínimo GRASP*	Mínimo NN*	Diferença (%)
berlin52	8743	8250	-5,98%
kroA100	29568	24822	-19,12%
kroA150	40470	32506	-24,50%
kroA200	45386	35322	-28,49%
kroB100	27145	26240	-3,45%
kroB150	40197	31803	-26,39%
kroB200	48389	36056	-34,21%
kroC100	26340	23872	-10,34%
kroD100	26437	25068	-5,46%
kroE100	28282	25272	-11,91%
lin105	20405	16966	-20,27%
pr76	145930	135629	-7,59%
pr107	57048	47080	-21,17%
pr124	92561	69455	-33,27%
USGS	5791600	5004000	-15,74%

Fonte: Autores.

No **Apêndice I** é possível observar todos os resultados obtidos em cada teste feito. Em destaque estão os valores mínimos de cada conjunto testado. Ao realizar essa análise pode-se observar que quanto maior o percentual de aleatoriedade empregado ao algoritmo menor foi o desempenho, já que todos os valores mínimos foram encontrados em todas as bases nos testes entre 83% e 90% de aleatoriedade.

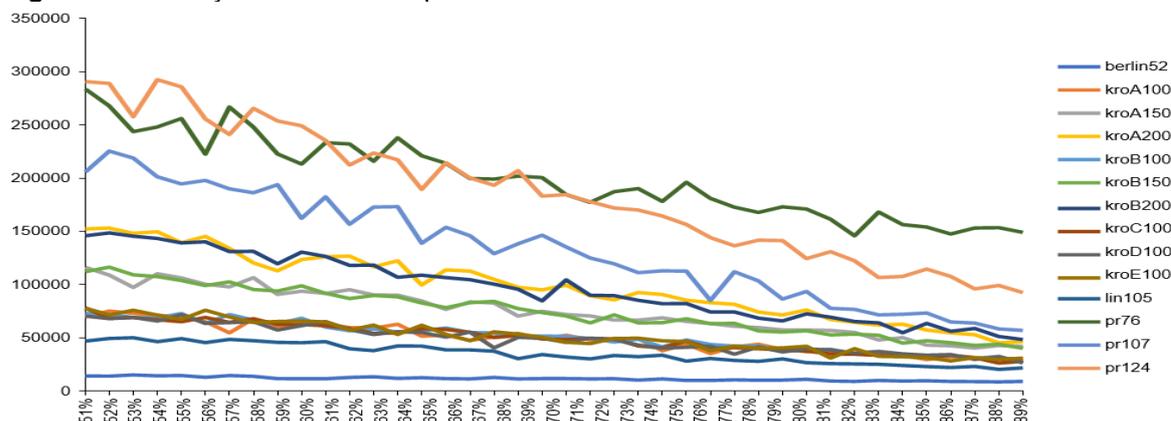
4.2. Análise Crítica

Quatro avaliações críticas foram realizadas considerando uma análise das bases com relação ao índice de queda com a aleatoriedade entre 50 e 90%; análise do índice de queda por base a cada 5% de variação; uma avaliação do tempo de operação; e por fim uma avaliação de aplicabilidade da estrutura foi desenvolvida.

Como análise inicial, foi construído um gráfico de linha com todos os resultados das bases da literatura com o objetivo de verificar o comportamento das respostas à medida que ocorreu a mudança dos pontos de aleatoriedade do GRASP*. Esta análise está representada pela Figura 5.

Pode-se observar que houve um comportamento de queda das respostas encontradas e que houve também quedas mais bruscas em algumas bases e quedas mais suaves em outras. A pr124 apresentou a maior inclinação com relação às demais, contudo, apesar de ser maior a inclinação da queda, há a ocorrência de mínimos locais (elevações de respostas) durante a queda dos resultados. Esses picos também podem ser observados na linha de outras bases como pr76 e pr107. Para que seja possível avaliar em qual percentual cada base apresentou um maior índice de queda, foi feita uma análise de cinco em cinco por cento.

Figura 5 – Avaliação do índice de queda dos resultados



Fonte: Autores.

Em sequencia, tomando por referência uma segunda análise, que foca no índice de queda por base a cada 5% de variação, cada uma das quinze bases analisadas foi dividida em oito grupos.

A organização dos grupos se deu da seguinte forma:

- Grupo 1: resultados obtidos dos testes de 51 a 55%;
- Grupo 2 - resultados obtidos dos testes de 56 a 60%;
- Grupo 3 - resultados obtidos dos testes de 61 a 65%;
- Grupo 4 - resultados obtidos dos testes de 66 a 70%;
- Grupo 5 - resultados obtidos dos testes de 71 a 75%;
- Grupo 6 - resultados obtidos dos testes de 76 a 80%;
- Grupo 7 - resultados obtidos dos testes de 81 a 85%;
- Grupo 8 - resultados obtidos dos testes de 86 a 90%.

Dentro de cada um desses grupos foi identificado os valores mínimos, máximos e médios como apresentado na Figura 6. Como se pode observar na figura em questão, há uma queda progressiva nas respostas à medida que o ponto de aleatoriedade é deslocado, porém essa redução não é constante e em alguns casos o algoritmo fica preso em mínimos locais.

Figura 6 – Avaliação do índice de queda dos resultados



Fonte: Autores.

Com o intuito de avaliar as quedas a cada intervalo dos grupos, identificar aproximadamente o ponto de aleatoriedade em que o algoritmo apresentou melhor desempenho e, também, o ponto em que ele ficou preso, foi realizado o cálculo do índice de queda dos valores mínimos de cada grupo. O resultado desta análise está apresentado na Tabela 3. Na tabela, cada análise descrita corresponde a uma diferença entre grupos subsequentes de 5% (i.e., análise 1 corresponde à diferença percentual da resposta entre os grupos na faixa de 51-55% e 56-60%, e assim por diante, caracterizando 7 análises).

Tabela 3 – Índice de queda dos mínimos

Bases	1	2	3	4	5	6	7	Média
pr107	16,574%	14,427%	7,127%	13,683%	23,530%	15,875%	20,369%	15,941%
kroA200	18,917%	11,816%	4,660%	9,753%	16,691%	13,275%	26,768%	14,555%
kroB200	14,139%	10,561%	20,792%	2,961%	19,730%	17,069%	11,516%	13,824%
pr124	6,500%	21,400%	3,328%	10,181%	16,981%	21,881%	13,249%	13,360%
kroB150	9,650%	11,839%	10,821%	13,206%	13,385%	18,850%	10,802%	12,650%
kroB100	8,468%	11,852%	7,313%	19,430%	6,974%	13,931%	18,243%	12,316%
kroC100	4,166%	13,107%	9,409%	17,257%	4,700%	15,023%	20,041%	11,958%
kroD100	12,836%	7,464%	23,340%	1,159%	13,769%	-0,692%	24,340%	11,745%
kroA150	6,805%	6,670%	16,836%	5,399%	13,675%	16,640%	15,666%	11,670%
kroE100	4,099%	17,368%	10,999%	5,974%	14,885%	18,150%	8,885%	11,480%
kroA100	21,069%	5,927%	7,795%	20,499%	6,880%	4,136%	12,316%	11,232%
lin105	2,153%	16,376%	19,935%	0,690%	7,401%	13,952%	15,272%	10,826%
USGS	10,507%	7,872%	10,956%	11,652%	3,110%	7,588%	14,602%	9,470%
pr76	12,486%	-1,208%	7,789%	10,908%	5,354%	13,092%	-1,173%	6,750%
berlin52	17,396%	-0,086%	1,993%	9,626%	3,434%	8,197%	4,333%	6,413%

Fonte: Autores.

Os valores destacados na Tabela 3, são os valores negativos. Nesses casos não houve uma redução dos valores mínimos, mas um aumento com relação ao grupo anterior. Nestes locais, pode-se dizer que o algoritmo ficou preso em um mínimo local, o que pode ter interferido no resultado, levando em consideração que o algoritmo foi executado por apenas 100 ciclos.

Com base nos valores médios das quedas, a base que apresentou maior índice de queda de mínimos foi a pr107 e a que apresentou o menor índice de queda foi a berlin52.

Na terceira análise proposta, centrada em uma avaliação do tempo de operação a partir dos estudos feitos anteriormente, segundo dados mostrados no **Apêndice II**, tem-se as bases com melhor tempo de processamento (ou seja, tempo aproximadamente zero, foram selecionadas e processadas novamente com 90% de aleatoriedade durante 200 ciclos). O objetivo dessa análise é constatar se com o aumento do número de ciclos resultará em um melhor desempenho dos resultados.

Na Tabela 4 estão apresentados os resultados desta análise em comparação com o resultado do mesmo processo da análise anterior com 100 ciclos. Conforme os dados apresentados pela tabela em questão, observa-se um incremento no desempenho dos resultados em quase todas as bases de dados devido ao aumento do número de ciclos.

Tabela 4 – Análise comparativa processamento com 90% de aleatoriedade

	100 ciclos	200 ciclos	Melhoria (%)
kroB150	40197	36766	8,54%
kroB200	48389	50920	-5,23%
kroC100	27952	26403	5,54%
kroD100	26437	27776	-5,06%
kroE100	30719	30197	1,70%
lin105	21892	19681	10,10%
pr76	149224	144866	2,92%
pr107	57048	56747	0,53%
pr124	92561	84839	8,34%
USGS	5791600	5535300	4,43%

Fonte: Autores.

Porém, o fato de dobrar o número de ciclos, não corresponde necessariamente ao dobro de desempenho atingido nos resultados. Pode-se verificar na base lin105, por exemplo, que ela teve o melhor percentual de melhoria e este correspondeu a 10,10%. Para esta análise, o tempo de processamento foi de um minuto.

Por fim, a quarta análise preza pela aplicabilidade da solução proposta, assim como descrito em Losqui e Souza (2019), onde várias aplicações práticas dependem de algoritmos de otimização para o roteamento de veículos, principalmente os que dependem de uma operação logística. Neste caso em específico, focou-se em aplicações que seriam feitas para orientações de caminhões em minas a céu aberto, durante os processos de coleta e entrega de minério.

A proposta desenvolvida tem a capacidade de otimizar o planejamento das rotas de visitas de coletas e entregas de minérios, simbolizando economias de consumos (como de combustível e tempo de viagem), que podem refletir em menos manutenções em equipamentos que podem representar um custo na faixa de milhões de dólares. Além disso, a logística das rotas de caminhões em minas a céu aberto representa, segundo Lisboa *et al.* (2019), 50% (ou até 60%) dos custos de operações de uma mina.

5 CONCLUSÃO

A pesquisa demonstrada figura como um processo de experimentação de um algoritmo, que contribui com as pesquisas relacionadas ao uso de metaheurísticas para otimização do processo de roteirização de veículos, que foi proposto em outra obra. Durante sua condução, conforme o seu objetivo principal, uma análise sobre algoritmo GRASP* proposto por Souza *et al.* (2017), por meio da avaliação de seu comportamento, a partir do deslocamento do percentual de aleatoriedade, foi realizada. Assim, algumas lacunas foram preenchidas, com durante um processo experimental, que ainda não tinham sido apresentadas na literatura, com foco no dado algoritmo.

Dentre as contribuições desta pesquisa, podem ser listadas: uma nova avaliação, com uma quantidade de bases de dados da literatura três vezes maior do que o experimentado em Souza *et al.* (2017), da estrutura GRASP* (proporcionando melhor conhecimento comportamental do algoritmo do que o disposto no trabalho original); otimizações de propostas de roteamentos que chegaram a 24%; além de uma demonstração de que a limitação do número de ciclos tem diferentes impactos em diferentes bases de dados.

Uma demonstração comportamental de uma metaheurística, que foca em um problema que movimenta milhões e até bilhões de reais em investimentos na indústria de mineração, é de extrema necessidade, uma vez que algumas situações serão mais complexas que as outras. Tal perspectiva foi demonstrada em situações onde houve até a piora da qualidade da resposta, quando se limitou o número de ciclos de busca pela resposta ótima.

Após a avaliação do desempenho obtido em cada variação proposta no algoritmo GRASP*, conclui-se que o aumento do percentual de aleatoriedade e o aumento do número de ciclos influenciam diretamente na qualidade das respostas.

Com os resultados obtidos nesse estudo, análises futuras foram alinhadas, como a validação do desempenho das respostas do algoritmo deslocando a quantidade de ciclos de execução e o ponto de aleatoriedade ao mesmo tempo. Este experimento impactaria em uma análise do tipo de estrutura computacional que seria mais adequado durante um processo real de roteamento, uma vez que 50 a 60% de investimentos de uma mina seriam decididos, e a avaliação da resposta, pautada pelo recurso computacional e urgências de contratos, poderiam ser estrategicamente discutidos. Tal pesquisa pode significar uma relevante conexão com avaliação de custos computacionais, com análise de paralelismo e até recursos de computação em nuvem.

REFERÊNCIAS

- ALOISE, D. J.; NORONHA, T. F.; MAIA, R. S.; BITTENCOURT, V. G. Heurísticas de colônia de formigas com path-relinking para o problema de otimização da alocação de sondas de produção terrestre–SPT. *In: XXXIV SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL*, 34., Rio de Janeiro, 2002.
- ALVARENGA, G. B. **Um Algoritmo Híbrido para os Problemas de Roteamento de Veículos Estático e Dinâmico com Janela de Tempo**. Tese de Doutorado. Universidade Federal de Minas Gerais, Belo Horizonte, 2005, 180 p.
- ARROYO, J. E. C. **Heurísticas e metaheurísticas para otimização combinatória multiobjetivo**. Doutorado em Engenharia Elétrica (Tese de doutorado) - Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas, Campinas, 2002.
- CORMEN, T. **Desmistificando algoritmos**. 1 ed. São Paulo: Elsevier Academic, 2015. 940 p. ISBN 978-0-262- 51880-2
- FEO, T. A.; RESENDE, M. G. C. Greedy Randomized Adaptive Search Procedures. **Journal of Global Optimization**, v. 6, p. 109-133. 1995.
<https://doi.org/10.1007/BF01096763>
- FESTA, P.; RESENDE, M. G. C. GRASP: basic components and enhancements. **Telecommunication Systems**, v. 46, n. 3, p. 253-271, 2011.
<https://doi.org/10.1007/s11235-010-9289-z>

- HART, J. P., SHOGAN, A. W. Semi-greedy heuristics: an empirical study. **Operations Research Letters**, v. 6, p.107–114, 1987. [https://doi.org/10.1016/0167-6377\(87\)90021-6](https://doi.org/10.1016/0167-6377(87)90021-6)
- LISBOA, A. C.; SOUZA, F. H. B.; RIBEIRO, C. M.; MAIA, C. A.; SALDANHA, R. R.; CASTRO, F. L. B.; VIEIRA, D. A. G. On Modelling and Simulating Open Pit Mine Through Stochastic Timed Petri Nets. **IEEE Access**, v. 7, p. 112821–112835, 2019. <https://doi.org/10.1109/ACCESS.2019.2934718>
- LOSQUI, H. V. F.; SOUZA, F. H. B. Análise De Pontos De Aleatoriedade Como Estratégia Para Melhoria De Ótimos Locais Em Uma Heurística Construtiva. **Revista Produção Online**. Florianópolis, SC, v. 19, n. 3, p. 923-951, 2019. <https://doi.org/10.14488/1676-1901.v19i3.3336>
- MARINAKIS, Y.; MARINAKI, M.; MATSATSINIS, N. A hybrid discrete artificial bee colony-GRASP algorithm for clustering. *In*: COMPUTERS & Industrial Engineering, 2009. **International Conference on IEEE**, p. 548-553, 2009. <https://doi.org/10.1109/ICCIE.2009.5223810>
- MORAIS, T. G. M.; OLIVEIRA, J. P. F.; JÚNIOR, A. C. G.; GOMES, H. C. Análise da aplicação de Métodos Heurísticos na Resolução do Problema de Roteamento de Veículos com Frota Heterogênea. *In*: XL ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO. Paraná, 2020. https://doi.org/10.14488/ENEGEP2020_TN_STO_344_1771_40983
- RESENDE, M. G. C.; RIBEIRO, C. C. Greedy randomized adaptive search procedures. *Handbook of Metaheuristics*. **Kluwer Academic Publishers**, p. 219-249, 2003. https://doi.org/10.1007/0-306-48056-5_8
- RIBEIRO, C. C. **Metaheuristics and applications**. Monte Estoril, Portugal: Advanced School on Artificial Intelligence (Constraint Programming), 1996.
- 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.
- 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. *In*: SBPO – SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL, 49., Blumenau, 2017.
- SOUZA, R. F. F. **Planejamento da expansão de sistemas de distribuição usando a metaheurística de busca em vizinhança variável**. 2011. 106 f. Dissertação (mestrado) - Universidade Estadual Paulista, Faculdade de Engenharia de Ilha Solteira, 2011. Disponível em: <http://hdl.handle.net/11449/87148>.

SOUZA, F. H. B.; FERREIRA, N. C.; SOUZA, P. H. G.; MELLIM, R. D.; ROCHA, V. A. R. Análise e Aplicação de Heurísticas para Definição de Rotas com Solução Otimizada Aplicado em uma Indústria do Ramo Alimentício. *In*: CONGRESSO BRASILEIRO DE ENGENHARIA DE PRODUÇÃO, 40., Ponta Grossa, 2019.



Artigo recebido em: 02/07/2021 e aceito para publicação em: 21/02/2022

DOI: <http://doi.org/10.14488/1676-1901.v21i4.4398>

APÊNDICE 1 - Resultados do experimento

	berlin52	kroA100	kroA150	kroA200	kroB100	kroB150	kroB200	kroC100	kroD100	kroE100	lin105	pr76	pr107	pr124	USGS
51%	14212	69765	116248	152346	73026	112162	145860	78179	70429	77816	46939	283833	205783	290825	12189000
52%	14078	75206	108948	153195	72064	116469	148587	68376	68208	71562	49433	267890	225484	289005	12108000
53%	15159	73112	97453	148279	69213	109155	145480	69263	69224	75911	50107	243820	218931	257818	12068000
54%	14483	71304	110274	149681	68823	107633	143397	67091	65847	71374	46457	248193	201295	292642	11678000
55%	14645	69401	106460	139433	72918	103992	139182	65169	71938	67187	49304	256180	194587	285836	12704000
56%	13006	65191	100314	145335	63513	99151	140268	69214	63706	75949	45603	222616	198024	255747	12690000
57%	14523	54779	97948	134107	72091	102590	131060	64545	64352	69587	48554	266870	189983	241060	11433000
58%	13909	68010	106465	120483	66774	95527	131300	67939	65220	64433	47330	248049	186313	265654	11020000
59%	11855	60665	90821	113056	62995	93957	119503	62753	57395	65303	45843	222897	193860	253788	11182000
60%	11629	62746	93811	123599	68533	99055	130505	62454	61436	65857	45457	213377	162336	249163	10451000
61%	11639	63252	91827	126269	60278	91895	126376	61511	65401	64972	46492	233447	182549	235512	10110000
62%	12902	59653	95217	126822	56704	87078	117989	58807	57840	57151	39829	232164	156687	212370	10956000
63%	13528	59240	90332	116781	58382	89820	118328	54268	53111	61928	38013	215954	173002	223635	10054000
64%	11964	62932	90205	122484	55529	88510	106882	54698	55808	53242	42407	237977	173282	217354	9892500
65%	12642	51532	84763	99697	56425	82833	108795	58699	55185	61831	42336	221038	138916	189473	9628300
66%	11802	52715	76705	113890	59312	78335	106570	57886	50633	53426	38760	214057	153896	214556	10330000
67%	11495	47515	83798	112680	55177	83083	104640	55158	55413	47386	38723	199791	146070	200239	9788100
68%	12765	50948	82838	104884	54585	84376	100377	50402	40715	55691	37436	199133	129015	193440	9902300
69%	11407	54188	70492	97594	51468	77460	95554	52378	50602	53467	30435	201976	138488	207159	9482100
70%	11800	48927	75091	95051	51809	73870	84659	49162	49681	50615	34345	200701	146460	183168	8573400
71%	11768	52534	72078	99590	51325	70672	104518	48334	45779	45842	32106	184581	135489	184667	8408000
72%	11508	48048	70567	89631	49479	64120	89938	49693	49904	44555	30225	177412	125129	177937	9165200
73%	11528	47510	66755	85781	46133	71654	89587	48918	48353	49524	33438	187366	119604	172008	7574400
74%	10309	49361	66686	92710	48733	64115	85303	42403	43334	49620	32343	190270	111362	170188	8859100
75%	11475	37775	68913	90636	41468	64226	82152	40678	40243	47238	33723	178293	112999	164520	7969600
76%	9955	43939	65429	85347	48022	67936	82301	46926	41170	46520	28172	196288	112630	156581	7864300
77%	9963	35176	63397	83144	43756	63429	74376	40063	42312	37923	30545	180994	85159	144302	8082400
78%	10570	41497	60958	81499	42328	63819	74241	40910	34702	42270	28789	172850	112214	136583	8134600
79%	10270	43934	59743	74310	42239	56494	68418	39767	41574	40213	27988	167913	103576	141776	7338800
80%	10435	39285	57567	71463	38576	55533	65943	38766	36765	40489	30357	173202	86479	141359	7605300
81%	11026	37125	57368	76164	37734	56653	72596	37740	39693	42134	26680	170993	93735	124444	7634800
82%	9510	36619	57090	67193	39305	52828	69419	34866	38794	31040	25800	161510	77862	130979	6788200
83%	9139	35128	55051	64664	35091	53582	65357	34704	35820	39753	25494	145930	76829	122345	6781900
84%	10034	34304	47988	61976	33764	52413	64177	33606	37264	32550	25183	168231	71640	106697	7003000
85%	9422	33721	50116	62850	33202	45065	54687	32942	34942	32206	24083	156477	72147	107870	6894400
86%	9678	29568	43209	57410	31110	47303	63600	31442	33726	31289	22967	154045	73398	114594	5859700
87%	9125	32042	42543	54767	33522	45376	56176	33869	34383	28282	22159	147642	64942	107711	6404400
88%	8932	31450	40470	52934	31018	43018	58878	31304	29720	31297	23212	153216	63948	96064	5903200
89%	8743	30261	42708	45386	32142	44049	51156	26340	32500	29601	20405	153519	58465	99277	5939400
90%	9074	30823	42138	46447	27145	40197	48389	27952	26437	30719	21892	149224	57048	92561	5791600
Mínimo	8743	29568	40470	45386	27145	40197	48389	26340	26437	28282	20405	145930	57048	92561	5791600

Fonte: Autores.

APÊNDICE 2 - Resultados do tempo de processamento

	berlin52	kroA100	kroA150	kroA200	kroB100	kroB150	kroB200	kroC100	kroD100	kroE100	lin105	pr76	pr107	pr124	UFGS
51%	0,9868	0,2634	1,0754	1,9468	0,0013	0	0	0	0	0	0	0	0	0	0
52%	1,2531	0,3181	1,3879	2,2353	0,0029	0	0	0	0	0	0	0	0	0	0
53%	1,0388	0,2762	1,1285	2,0622	0,0031	0	0	0	0	0	0	0	0	0	0
54%	1,0676	0,2841	1,1527	2,1222	0,0028	0	0	0	0	0	0	0	0	0	0
55%	1,2127	0,3102	1,1279	2,2945	0,0034	0	0	0	0	0	0	0	0	0	0
56%	1,0118	0,2696	1,1031	2,0042	0,0018	0	0	0	0	0	0	0	0	0	0
57%	1,1019	0,2770	1,2485	2,0896	0,0016	0	0	0	0	0	0	0	0	0	0
58%	1,5857	0,3472	1,3814	2,1864	0,0019	0	0	0	0	0	0	0	0	0	0
59%	1,0842	0,2885	1,1791	2,4229	0,0028	0	0	0	0	0	0	0	0	0	0
60%	1,7926	0,4748	1,8122	4,1562	0,0040	0	0	0	0	0	0	0	0	0	0
61%	1,7286	0,2682	1,0736	2,9463	0,0027	0	0	0	0	0	0	0	0	0	0
62%	0,9692	0,2585	1,0569	1,9447	0,0052	0	0	0	0	0	0	0	0	0	0
63%	0,9875	0,2674	1,0790	1,9592	0,0018	0	0	0	0	0	0	0	0	0	0
64%	1,0154	0,2728	1,0984	2,0013	0,0014	0	0	0	0	0	0	0	0	0	0
65%	1,1174	0,2850	1,0981	1,9750	0,0017	0	0	0	0	0	0	0	0	0	0
66%	0,9604	0,2550	1,0470	1,9176	0,0015	0	0	0	0	0	0	0	0	0	0
67%	1,2655	0,2973	1,2061	1,9868	0,0017	0	0	0	0	0	0	0	0	0	0
68%	0,9872	0,2630	1,0762	1,9510	0,0020	0	0	0	0	0	0	0	0	0	0
69%	1,1538	0,3097	1,1245	1,9422	0,0022	0	0	0	0	0	0	0	0	0	0
70%	0,9674	0,2588	1,0536	1,9175	0,0015	0	0	0	0	0	0	0	0	0	0
71%	1,0372	0,2750	1,1126	2,028	0,0014	0	0	0	0	0	0	0	0	0	0
72%	1,0228	0,3090	2,3403	3,4661	0,0017	0	0	0	0	0	0	0	0	0	0
73%	1,076	0,2790	1,1419	2,1701	0,0020	0	0	0	0	0	0	0	0	0	0
74%	0,9977	0,2659	1,0854	1,9818	0,0014	0	0	0	0	0	0	0	0	0	0
75%	1,0609	0,2850	1,1486	2,0969	0,0015	0	0	0	0	0	0	0	0	0	0
76%	1,0218	0,2741	1,1066	2,0181	0,0014	0	0	0	0	0	0	0	0	0	0
77%	1,4096	0,4136	6,0741	2,0291	0,0019	0	0	0	0	0	0	0	0	0	0
78%	1,0032	0,2684	1,0922	1,987	0,0009	0	0	0	0	0	0	0	0	0	0
79%	1,0139	0,2699	1,1018	2,0236	0,0018	0	0	0	0	0	0	0	0	0	0
80%	1,0187	0,2726	1,1155	2,0149	0,0013	0	0	0	0	0	0	0	0	0	0
81%	0,9685	0,2591	1,0595	1,9174	0,0016	0	0	0	0	0	0	0	0	0	0
82%	0,9856	0,2632	1,0751	1,9630	0,0011	0	0	0	0	0	0	0	0	0	0
83%	0,9845	0,2610	1,0902	1,9574	0,0011	0	0	0	0	0	0	0	0	0	0
84%	0,9658	0,2524	1,0440	1,8841	0,0020	0	0	0	0	0	0	0	0	0	0
85%	0,9744	0,2559	1,0504	1,9170	0,0018	0	0	0	0	0	0	0	0	0	0
86%	0,9691	0,2571	1,0555	1,9303	0,0014	0	0	0	0	0	0	0	0	0	0
87%	1,0134	0,2626	1,0758	1,9518	0,0018	0	0	0	0	0	0	0	0	0	0
88%	0,9948	0,2658	1,0839	1,9721	0,0017	0	0	0	0	0	0	0	0	0	0
89%	1,5245	0,4757	1,9486	2,2747	0,0018	0	0	0	0	0	0	0	0	0	0
90%	1,2908	0,4901	1,6337	2,8471	0,0028	0	0	0	0	0	0	0	0	0	0

Fonte: Autores.