

O problema de minimização do deslocamento de operadores em rotas de inspeção: Um estudo de caso

Thiago Dias Barcelos¹, Tatianna Aparecida Pereira Beneteli², Luciano Perdigão Cota²

¹Vale S.A.

Tubarão, CEP: 29090-913, Ouro Preto, MG, Brasil

²Instituto Tecnológico Vale

Campus Universitário, Morro do Cruzeiro, CEP: 35402-206, Ouro Preto, MG, Brasil

thiago.barcelos@vale.com, marcone@ufop.edu.br,
luciano.p.cota@itv.org

RESUMO

As rotas de inspeção são formadas por uma série de equipamentos que devem ser visitados periodicamente para checagem. Os itens a serem checados são cadastrados de forma a conduzir o inspetor a uma verificação de modos de falha dos equipamentos ou sinais de desvios operacionais. Em ambientes industriais, as rotas de inspeção são geralmente definidas de forma manual, baseado no conhecimento e experiência dos profissionais. Neste estudo é proposto um algoritmo baseado na meta-heurística GRASP capaz de planejar rotas de inspeções otimizadas baseado no tempo de deslocamento entre os pontos de inspeção, buscando a redução do tempo total gasto pelo inspetor. Este algoritmo, implementado na forma de uma ferramenta de suporte à decisão, pode ser utilizado para definir novas rotas de inspeção, assim como atualizar as rotas já existentes. Para validação do estudo, é utilizado um estudo de caso em uma usina de pelotização da Vale S.A.

PALAVRAS CHAVE. Mineração. Rotas de Inspeção. GRASP.

Tópicos: PO na indústria

ABSTRACT

Inspection routes are made up of a series of pieces of equipment that must be visited periodically for checks. The items to be checked are registered in such a way as to lead the inspector to check for equipment failure modes or signs of operational deviations. In industrial environments, inspection routes are usually defined manually, based on the knowledge and experience of professionals. This study proposes an algorithm based on the GRASP meta-heuristic capable of planning optimized inspection routes based on the travel time between inspection points, seeking to reduce the total time spent by the inspector. This algorithm, implemented in the form of a decision support tool, can be used to define new inspection routes, as well as update existing routes. A case study at a Vale S.A. pelletizing plant is used to validate the study.

KEYWORDS. Mining. Inspection Routes. GRASP.

Topics: OR in industry

1. Introdução

O ato de inspecionar consiste em verificar se os ativos encontram-se em suas condições ideais conforme recomendações do fabricante ou mesmo preestabelecidas pela empresa. A inspeção é um importante ponto na gestão da manutenção, porque identifica falhas antes mesmo destas acontecerem, diminuindo o ônus com manutenções corretivas e possíveis paradas no processo produtivo. O registro das anomalias identificadas e as ações para promover a sua eliminação fazem parte do processo de inspeção, evitando sua reincidência [Ambrosio, 2022].

Os processos de inspeção de um parque operacional e de seus ativos seguem uma metodologia integrada, ou seja, associada a observação multidisciplinar dos ativos, obtendo diagnósticos e direcionamentos para ações preventivas e preditivas. A multidisciplinaridade integra as especialidade de manutenções mecânica, elétrica, automação e instrumentação, incluindo também os itens de análise de óleo e vibração. A inspeção operacional é realizada de maneira sensível, sem a utilização de instrumentos para avaliação e diagnóstico. Na inspeção sensível, o inspetor utiliza os sentidos (visão, audição, olfato e tato) para avaliar a situação do ativo. Essa prática é muito utilizada na gestão de manutenção, em conjunto com tecnologias que facilitam a identificação ou indícios de falhas nos equipamentos industriais. Esse tipo de manutenção não se beneficia de dispositivos remotos de inspeção e necessita invariavelmente de mão de obra para realizar seus apontamentos. A inspeção sensível se torna mais eficiente se combinada com outras técnicas de manutenção preditiva, por exemplo, sensores de vibração e temperatura. As duas inspeções se complementam diretamente e permitem análises mais pertinentes [Rezende and Gomes, 2016].

Um plano de rotas de inspeção é essencial para o controle das tarefas de inspeção que precisam ser realizadas. A elaboração de um plano de rotas é baseada na importância e criticidade dos ativos do processo produtivo, estabelecendo procedimentos que visam evitar que as falhas ocorram. No plano são definidas quais serão as atividades realizadas e qual a periodicidade em que essas atividades devem ser executadas, tendo em vista a frequência de uso do ativo, histórico de funcionamento e de falhas e, principalmente, o impacto que o ativo tem sobre o processo produtivo e a segurança operacional. Pontos críticos que podem causar prejuízos ao processo devem ser acompanhados com maior frequência. Além disso, o plano define quem irá executar as tarefas previstas, de acordo com a especialização necessária que pode incluir capacitações específicas.

Após a definição de quais atividades precisam ser realizadas no plano de rotas de inspeção, as rotas de inspeção são planejadas. As rotas indicam o caminho que o inspetor deve percorrer no chão fabril para coleta da informação de ativos. Elas são planejadas com o objetivo de minimizar o tempo de deslocamento e garantir a segurança do inspetor. A observação dos ativos, na frequência e abrangência de itens inspecionáveis necessária, muitas vezes esbarra no quantitativo finito de recursos de mão de obra. Em casos aplicáveis a evolução para sensoriamento remoto, automatizando o monitoramento e até o diagnóstico, reduzem ou eliminam a necessidade do inspetor. Em outros casos se faz necessário o uso racional dos recursos humanos disponíveis, atendendo as necessidades de inspeção de maneira otimizada.

Nas companhias, as rotas operacionais de inspeção são geralmente planejadas a partir do conhecimento técnico e da experiência da equipe responsável. Os deslocamentos são roteirizados de forma intuitiva. A demanda por otimização de recursos é sempre uma realidade em processos industriais, seja este recurso humano ou de natureza material. No setor operacional são comuns questionamentos relacionados ao tempo excessivo dedicado à rotina dos operadores das usinas de pelotização ao cumprimento das tarefas de inspeção. Atrélado a esses relatos, acrescenta-se a percepção dos operadores de que existe um deslocamento em excesso durante a execução de rotas operacionais.

Apesar de existirem diversos estudos na literatura sobre inspeções de equipamentos, não

foram encontrados estudos sobre o problema de planejamento de rotas de inspeção em ambientes industriais. Portanto, para preencher esta lacuna, este estudo introduz o problema e propõe um algoritmo baseado na meta-heurística *Greedy Randomized Adaptive Search Procedures* (GRASP) [Resende and Ribeiro, 2019] capaz de planejar rotas de inspeções baseado no tempo de deslocamento entre os pontos de inspeção, buscando a redução do tempo total gasto pelo inspetor.

Este algoritmo, na forma de uma ferramenta de suporte à decisão, pode ser utilizado para definir novas rotas de inspeção, assim como atualizar as rotas já existentes em casos de parada de algum equipamento para manutenção, por exemplo. Além disso, a ferramenta ainda poderá ser aplicada às rotas já existentes, buscando atender ao apelo das unidades operacionais para que seja realizada a releitura de rotas já em curso, sugerindo rotas que melhor utilizem a mão de obra disponível e otimizem os tempos dedicados às inspeções.

Para validar o algoritmo proposto é utilizado um estudo de caso da unidade de pelotização de Tubarão, da Vale S.A., localizada na cidade de Vitória-ES. Nesta unidade, há uma demanda de 78 rotas de operação ativas: 12 rotas realizadas pelas equipes de operação nas Usinas 1 a 4, 35 rotas nas Usinas 5 a 7 e 31 rotas na Usina 8, conforme apresentado na Figura 1. Essas rotas foram definidas de forma empírica, utilizando o conhecimento das equipes de manutenção e operação.

Gerência	Cumprimento Rotas				Conformidade das Rotas Executadas - Adesão aos pontos de inspeção existentes		
	Total Rotas	Atrasadas	OK	% OK	Total Pontos	Executados	% Execução
MECÂNICA US. 1 a 4	65	1	64	98,5%	32.435	31.603	97,4%
ELÉTRICA/INSTR. US. 1 a 8	267	10	257	96,3%	115.349	105.173	91,2%
MECÂNICA US. 5 a 7	63	4	59	93,7%	30.628	28.540	86,7%
PREDITIVA TUBARÃO	19	2	17	89,5%	3.254	2.633	80,9%
VARGEM GRANDE	69	10	59	85,5%	56.375	48.689	86,4%
SÃO LUIS	182	14	168	92,3%	60.614	57.232	94,4%
MECÂNICA US. 8	54	12	42	77,8%	29.474	26.236	89,0%
AUTOMAÇÃO US. 1 a 8	33	-	33	100,0%	2.497	2.493	99,8%
SERVIÇOS AMBIENTAIS TUBARÃO	44	-	44	100,0%	20.450	20.022	97,9%
OPERAÇÃO US. 1 a 4	12	-	12	100,0%	3.200	3.193	99,8%
OPERAÇÃO US. 5 a 7	35	2	33	94,3%	10.486	9.927	94,7%
OPERAÇÃO US. 8	31	3	28	90,3%	3.410	2.880	84,5%
OPERAÇÃO UTILIDADES	2	-	2	100,0%	53	53	100,0%
Total	874	58	816	93,4%	368.225	336.674	91,4%

Figura 1: Quadro de rotas da unidade de pelotização de Tubarão.

O restante do trabalho está organizado da seguinte forma: a Seção 2 apresenta uma revisão bibliográfica sobre o tema. A Seção 3 realiza a caracterização do problema. Já a Seção 4 apresenta o algoritmo proposto, enquanto na Seção 5 os experimentos computacionais são apresentados. Por fim, a Seção 6 apresenta as conclusões do trabalho.

2. Referencial Teórico

Ao considerar a utilização de técnicas de inspeções visuais em processos de manutenção e em áreas operacionais, pode-se localizar as inspeções tratadas neste trabalho como uma técnica simples e amplamente estudada. Ambrosio [2022] tratou a aplicação de inspeções em correias transportadoras e as manutenções de integridade de suas estruturas. Segundo o autor, a inspeção visual é um método de inspeção não destrutivo, simples e amplamente utilizado, destinado a avaliar a condição ou propriedades de um componente, estrutura ou solda. Já Rezende and Gomes [2016] definem que a inspeção sensitiva é o tipo de inspeção que abrange praticamente todos os modos de falhas do equipamento ou instalação industrial. Apesar de, normalmente, se limitar ao uso dos sentidos do inspetor, é possível diagnosticar anomalias em estágio mediano de degradação.

Diversos trabalhos abordam o planejamento das inspeções, definindo os equipamentos que precisam ser inspecionados, a periodicidade das inspeções e as equipes que deverão realizá-las. Teles [2007] elabora um método para planejamento de inspeções de equipamentos dividido em três fases. Na primeira fase, denominada elaboração da matriz amostral, são definidos quais e quantos equipamentos inspecionar. Na segunda, intitulada distribuição da matriz amostral no tempo, é descrito quando as inspeções devem ser realizadas. Por fim, na terceira fase, denominada

alocação de recursos humanos para execução da matriz amostral, é estabelecido quem deve realizar essas inspeções.

No que diz respeito ao desenvolvimento de procedimentos para o planejamento de rotas de inspeção, não foram encontrados trabalhos na literatura que abordem esse tema. No entanto, esse problema possui algumas características que o aproxima de outros problemas tratados na literatura: *i*) toda rota de inspeção precisa iniciar em um determinado ponto e o operador deve retornar a esse mesmo ponto ao final da rota; e *ii*) todos equipamentos precisam ser inspecionados uma e somente uma vez a cada execução da rota de inspeção.

Dessa forma, pode-se verificar que o problema de planejamento de rotas de inspeção se assemelha ao Problema do Caixeiro Viajante (PCV). O PCV é um problema de otimização combinatorial no qual busca-se determinar a rota de menor custo entre um grupo de pontos. Esses pontos pode representar localidades ou de outros elementos tratados em problemas de designação. Desde a sua primeira formulação foram publicados inúmeros trabalhos propondo diferentes alternativas para a solução do PCV. Além disso, uma infinidade de formulações avançadas também foram propostas, buscando aumentar a sua aplicabilidade em outros setores [Osaba et al., 2020].

Atualmente, os problemas de roteamento são um dos mais estudados na área de pesquisa operacional. Duas razões principais tornam este tema primordial: *i*) a formulação deste tipo de problemas inspira-se em situações logísticas, operacionais e de transporte do mundo real, implicando diretamente em benefício social e/ou empresarial ao encontrar uma solução adequada, e *ii*) a abordagem eficiente destes problemas geralmente supõe um desafio para a comunidade científica devido à sua natureza NP-difícil, tornando complexa a resolução ótima para conjuntos de dados reais. O segundo ponto tem levado pesquisadores ao desenvolvimento de métodos heurísticos de otimização, visando resolver esse problema de forma computacionalmente acessível [Ilavarasi and Joseph, 2014].

Além da aplicação clássica, o problema do caixeiro viajante pode ser encontrado em diversos outros contextos. Silva and Barcelos [2019] realizaram um estudo de caso do processo de entrega em uma distribuidora situada na cidade de João Monlevade. Inicialmente, buscou-se coletar dados sobre o transporte de mercadorias realizado pela organização para a construção de um modelo matemático. Posteriormente, converteu-se essa modelagem em uma programação computacional a fim de se obter a rota ótima utilizando o solver *AMPL*. Em uma abordagem voltada ao turismo, Gusson [2023] utilizou um método heurístico proposto para resolver PCV para proporcionar roteiros otimizados que permitam aos turistas aproveitar ao máximo sua experiência, facilitando a visita ao maior número de locais por meio da otimização da rota. Para tanto, foram consideradas as escolhas do viajante e a localização das propriedades rurais. Já em uma aplicação prática, Ferreira [2023] recorreu ao PCV para modelar uma nova rota de ônibus voltada para o público universitário da cidade de Araguaína, Tocantins. Apesar de contribuir para a melhoria do serviço ofertado, o trabalho limitou-se a 8 instituições de ensino superior.

No que se refere aos métodos de solução, desde a sua formulação, uma infinidade de métodos foi aplicada ao PCV e suas variantes. Encontrar a solução ideal do PCV é um processo computacionalmente difícil. Algoritmos exatos foram propostos, mas nem sempre eles podem ser usados devido à alta complexidade computacional e, conseqüente, o alto tempo computacional gasto para encontrar a solução ótima. Algoritmos de aproximação heurística foram projetados e podem produzir rapidamente soluções boas, mas nem sempre ótimas. Esses métodos buscam fornecer uma compensação entre a complexidade da computação e a qualidade das soluções [Hu et al., 2021].

Algoritmos Genéticos (GA do termo em inglês *Genetic Algorithm*) têm sido adotados por diversos autores da comunidade científica atual para resolução do PCV. Uma prática comum adicional relacionada ao GA aplicado ao PCV é a formulação de novos operadores, como pode ser

visto em Zhang et al. [2022] que propõe um algoritmo genético com gene saltador e operadores heurísticos. Os autores propõem um operador de cruzamento heurístico bidirecional, um operador de combinação de mutação, um operador único, um operador genético saltador, um operador de busca local e uma seleção de roleta melhorada da função de aptidão, visando solucionar problemas com baixa velocidade de convergência, baixa qualidade de solução e fácil queda em um ótimo local.

O método Simulated Annealing (SA) também é foco de muitas pesquisas em torno do PCV e suas variantes. Zhan et al. [2016] apresentaram um SA baseado em um novo mecanismo de controle do parâmetro de temperatura. O método implementado conta com uma lista de temperaturas, cujo máximo é utilizado como um critério de aceitação de uma solução candidata. Além disso, a lista de temperaturas é adaptada dinamicamente de acordo com o espaço de solução do problema. O trabalho de Rao [2021] utiliza a abordagem de SA para tratar o problema de entregas de mercadoria dentro de uma cidade com diversos centros de armazenamento e clientes. Frequentemente, esses problemas são definidos como mecanismos *hub-and-spoke*. O *hub* pode ser definido como a centro de armazenamento de onde as diversas mercadorias serão originadas e os *spokes* representam os diversos destinos para chegar aos clientes.

Um dos métodos mais utilizados nos últimos anos na resolução do PCV é a Colônia de Formigas (ACO do termo em inglês *Ant Colony Optimization*). Sheng [2022] propõe um algoritmo híbrido SOS-MMAS, no qual o avançado MMAS (do termo em inglês *Max-Min Ant System*) é utilizado como algoritmo básico para aumentar a eficiência do agendamento de tarefas, enquanto a pesquisa de organismos simbióticos (SOS do termo em inglês *Symbiotic Organisms Search*) é introduzida no MMAS para otimizar os parâmetros chave.

Em Reis et al. [2021], busca-se minimizar a exposição dos operadores à ruídos excessivos e evitar doenças ocupacionais. A partir da mediação de ruídos, do tempo de exposição dos operadores e das distâncias percorridas, os autores utilizam o problema de designação e o problema de roteamento de veículos para definir as rotas a serem executadas pelos operadores utilizando programação matemática. Os resultados da proposta são comparados com a solução do problema do caixeiro viajante para um único operador.

De acordo com a revisão bibliográfica apresentada, observa-se que o PCV foi encontrado em diversos problemas práticos, assim como diversos métodos foram utilizados em sua resolução [Osaba et al., 2020]. Apesar disso, poucas aplicações foram encontradas no âmbito do estudo deste trabalho, que busca determinar de forma simples e rápida a melhor ordem de execução das inspeções, a fim de minimizar o deslocamento do operador. Para tornar os tempos totais de inspeção mais aderentes aos realizados na prática, o tempo gasto para realização da inspeção dos equipamentos também é contabilizado em conjunto com o tempo de deslocamento.

3. Caracterização do Problema

Este trabalho introduz o problema de planejamento de rotas de inspeção. As rotas operacionais são fluxos com deslocamentos previamente definidos que visam avaliar equipamentos industriais em itens de inspeção de cunho operacional e não utilizando instrumentos, ou seja, é uma ação sensível. A seguir, o problema de planejamento de uma rota de inspeção é caracterizado.

1. Pontos de inspeção (P)

- (a) Cada ponto de inspeção $i \in P$ é fixo e possui uma localização L_i ;
- (b) O tempo de deslocamento, em segundos, entre dois pontos de inspeção é dado por Td_{ij} ;
- (c) Cada ponto de inspeção $i \in P$ deve ser visitado uma única vez;
- (d) Todos os pontos de inspeção de uma rota $z \in R$ devem ser visitados;
- (e) O tempo de inspeção em cada ponto $i \in P$ é dado por Tv_i , em segundos.

2. Inspetores (S)

- (a) Cada rota $z \in R$ é realizada por apenas um operador inspetor $k \in S$.

3. Rotas de Inspeção (R)

- (a) Cada rota de inspeção $z \in R$ deve ser iniciada e finalizada em determinado ponto $i \in P$;
 (b) Cada rota de inspeção $z \in R$ possui x pontos de inspeção;
 (c) Há a possibilidade de entrada ou retirada de novos pontos de inspeção na rota $z \in R$. Essa situação representa, por exemplo, a retirada de operação dos equipamentos em manutenção. Nesse caso, a rota precisa ser refeita.

Neste problema, o objetivo é definir o planejamento de cada rota de inspeção $z \in R$ com o sequenciamento do conjunto de pontos de inspeção (P) a serem avaliados nesta rota buscando minimizar o tempo total gasto pelo inspetor $k \in S$.

4. Algoritmo proposto

Nesta seção é apresentado o algoritmo proposto para tratar o problema em estudo. Por se tratar de uma classe de problemas NP-difíceis [Ilavarasi and Joseph, 2014], métodos exatos não são capazes de encontrar a solução ótima em tempo hábil para o problema do caixeiro viajante quando aplicado em instâncias de grande porte. Por esse motivo, optou-se por implementar um algoritmo baseado na meta-heurística GRASP, que tem obtido bom desempenho na resolução de outros problemas da classe NP-difíceis [João D. F. Jesus-Junior and Cota, 2024; Guimarães et al., 2019].

4.1. Representação e avaliação da solução

Uma solução s é representada por um vetor de inteiros com n posições, sendo que cada elemento do vetor representa um ponto de inspeção visitado e n indica a quantidade de pontos de inspeção pertencentes a rota em análise. O posicionamento no vetor representa a ordem de execução das inspeções. A Figura 2 ilustra a representação de uma possível solução para um problema com cinco pontos de inspeção.

4	3	1	2	5
---	---	---	---	---

Figura 2: Representação da solução - Rota de inspeção com cinco pontos.

A função de avaliação de uma solução s é dada pela Equação 1, onde Td é o tempo de deslocamento entre o ponto de inspeção anterior m e o ponto atual i e Tv_i é o tempo gasto na inspeção do ponto i .

$$\sum_{i=1 \dots n} Td_{mi} + Tv_i \quad (1)$$

4.2. Solução inicial

Uma heurística construtiva tem por objetivo construir uma solução, elemento por elemento. A forma de escolha de cada elemento a ser inserido a cada passo varia de acordo com a função de avaliação adotada e do problema abordado. Nas heurísticas clássicas, os elementos candidatos são geralmente ordenados segundo uma função gulosa, que estima o benefício da inserção de cada elemento, e somente o *melhor* elemento é inserido a cada passo.

No modelo implementado, optou-se por utilizar um algoritmo construtivo guloso para gerar a solução inicial. O Algoritmo 1 mostra o procedimento de construção de soluções para o problema utilizando a função de avaliação $g(\cdot)$, descrita na subseção seguinte.

Algoritmo 1: Construtivo Guloso

```

entrada:  $g(\cdot), Tv$ 
saida :  $s$ 
1  $s \leftarrow \emptyset$ 
2 Inicialize o conjunto  $P$  de pontos de inspeção de candidatos:
3 enquanto ( $P \neq \emptyset$ ) faça
4    $g_{min} = \min\{g(i) | i \in P\}$ 
5    $s \leftarrow s \cup \{i_{min}\}$ 
6   Atualize o conjunto  $P$  de pontos de inspeção candidatos
7 fim-enquanto
8 retorne  $s$ 
  
```

4.2.1. Função de avaliação do construtivo guloso

O problema de planejamento de rotas de inspeção aqui tratado tem como objetivo reduzir os deslocamento até os equipamentos a serem inspecionados. No algoritmo construtivo guloso implementado (Algoritmo 1), uma solução s é avaliada através da função de avaliação $g(\cdot)$, descrita na Equação 2.

$$g(i) = Td_{mi} \quad \forall i \in P \quad (2)$$

onde Td é o tempo de deslocamento entre o ponto de inspeção atual m e o ponto i .

Observa-se que a função de avaliação $g(\cdot)$ permite identificar o membro do conjunto de elementos candidatos P com o valor mais favorável, isto é, aquele que insere o menor tempo de deslocamento na rota a ser planejada.

4.3. Algoritmo GRASP

O algoritmo GRASP é uma meta-heurística constituída por heurísticas construtivas e busca local [Resende and Ribeiro, 2019]. Consiste de múltiplas aplicações de busca local, cada uma iniciando de uma solução diferente. As soluções iniciais são geradas por algum tipo de construção randômica gulosa ou algum esquema de perturbação.

O pseudocódigo 2 apresenta o algoritmo GRASP proposto para o problema abordado. Observa-se no algoritmo que há duas fases: construção da solução e aplicação da busca local na solução construída.

Algoritmo 2: GRASP

```

entrada:  $s, g(\cdot), Tv, iter_{max}, \alpha$ 
saida :  $s^*$ 
1  $s^* \leftarrow \text{ConstrutivoGuloso}(g(\cdot), Tv)$ 
2  $f^* \leftarrow f(s^*)$ 
3  $Iter = 0$ 
4 enquanto ( $Iter < iter_{max}$ ) faça
5    $\text{ConstrutivoParcialmenteGuloso}(g(\cdot), \alpha)$ 
6    $\text{BuscaLocal}(s)$ 
7   if ( $f(s) < f^*$ ) then
8      $s^* \leftarrow s$ 
9      $f^* \leftarrow f(s)$ 
10     $Iter = 0$ 
11  end
12   $Iter \leftarrow Iter + 1$ 
13 fim-enquanto
14 retorne  $s^*$ 
  
```

4.3.1. Fase de construção

Nessa fase uma solução é construída elemento a elemento. Inicialmente o elemento está em uma lista de candidatos (LC). Através de um fator α é criada uma lista restrita de candidatos (LCR) que possui os melhores elementos de LC, de acordo com a função de avaliação $g(\cdot)$. O tamanho de LCR é determinado por:

$$corte = g_{min} + \alpha\{g_{max} - g_{min}\} \quad (3)$$

Após a definição da LCR, seleciona-se um elemento da lista para compor a solução. A seleção do elemento pode ser realizada aleatoriamente ou através de um critério guloso. Esses dois tipos de seleções provocam duas variações do GRASP: aleatório ou guloso. Após a adição do elemento na solução, o processo continua com a atualização de ambas as listas LC e LCR. O processo de construção é finalizado quando a todos os elementos forem adicionados a solução.

O valor de α influencia na qualidade e diversidade da solução gerada na fase de construção. Valores baixos para α geram soluções gulosas de boa qualidade, mas com pouca diversidade. Um valor alto para α leva a uma grande diversidade com soluções de qualidade inferior. Isso também influencia o processo de busca local, pois soluções de qualidade inferior tornam o processo de busca local mais lento. Por exemplo com α igual a zero, o melhor elemento da LCR seria adicionado, assim a fase de construção é uma heurística gulosa. Quando utiliza-se um valor elevado de α a construção é randômica. No GRASP, o valor de α pode ser constante ou se alterar a cada iteração por meio de uma esquema aleatório ou adaptativo [Resende and Ribeiro, 2019].

O pseudocódigo 3 apresenta o algoritmo para a fase de construção da solução. Verifica-se que no algoritmo construtivo parcialmente guloso implementado, a seleção do elemento de LCR para compor a solução é realizada de forma aleatória e o valor de α é constante.

Algoritmo 3: Construtivo Parcialmente Guloso

```

entrada:  $g(\cdot), \alpha$ 
saida :  $s$ 
1  $s \leftarrow \emptyset$ 
2  $N \leftarrow$  número de pontos de inspeção
3 for ( $i \in N$ ) do
4    $LC \leftarrow$  lista de pontos de inspeção de candidatos
5   enquanto ( $|LC| > 0$ ) faça
6      $g_{min} = \min\{g(i) | i \in LC\}$ 
7      $g_{max} = \max\{g(i) | i \in LC\}$ 
8      $LCR \leftarrow \{i \in LC | g(i) \leq g_{min} + \alpha(g_{max} - g_{min})\}$ 
9     Seleciona aleatoriamente um elemento  $i \in LCR$ 
10     $s \leftarrow s \cup \{i\}$ 
11     $LC \leftarrow$  lista de pontos de inspeção de candidatos
12  fim-enquanto
13 end
14 retorne  $s$ 
  
```

No algoritmo implementado a solução inicial é um conjunto vazio e uma função gulosa avalia os elementos pelo benefício imediato. Os melhores elementos formam uma lista de candidatos restritos e o parâmetro α determina o tamanho da lista de candidatos restritos. Um elemento da lista é escolhido aleatoriamente e agregado a solução inicial. O algoritmo termina com uma solução aceitável ao problema.

4.3.2. Fase de busca local

A segunda fase do GRASP, ou fase de busca local, consiste em refinar a solução encontrada na primeira fase aplicando um método de busca local. Isso corresponde a uma intensificação na solução encontrada explorando regiões vizinhas para encontrar um ótimo local.

A busca local pode ser sofisticada, entretanto não se deve esquecer que o diferencial do GRASP é amostrar o espaço com gerações rápidas. Quanto melhor for a solução gerada na primeira fase, maior será a velocidade para encontrar um ótimo local na fase de busca local.

Neste trabalho é utilizada uma estrutura de vizinhança baseado em troca de pares de pontos de inspeção. A estratégia utilizada para o algoritmo de busca local é a *First Improvement*.

5. Estudo de Caso

O algoritmo proposto foi implementado usando o software Spyder® em linguagem Python e os experimentos foram realizados em um computador Intel(R) Xeon(R) E-2378G CPU 2.80GHz, memória RAM de 64GB, sistema operacional Microsoft Windows 10 Enterprise 64bits, 8 núcleos e 16 processadores lógicos. As instâncias com os dados dos pontos de inspeção foram armazenados em uma planilha de Excel® e inseridos como dados de entrada do algoritmo implementado.

5.1. Cenário

Para avaliar o algoritmo proposto utilizou-se dados reais da usina de pelotização Tubarão, localizada em Vitória. Conforme apresentado, atualmente nessa unidade existem 78 rotas ativas.

A rota abordada conta com 24 pontos de inspeção e trata-se de uma inspeção operacional em discos de pelotamento e desagregadores. O disco de pelotamento é o equipamento responsável por realizar transformação das partículas de minério de ferro em pequenos aglomerados denominados pelotas. Os desagregadores são responsáveis por garantir que a massa de minério fornecida ao disco de pelotamento esteja livre de aglomerados previamente formados. É importante ressaltar que os 24 equipamentos a serem inspecionados podem ser agrupados em dois conjuntos distintos: o primeiro formado por 12 discos de pelotamento e o segundo formado por 12 desagregadores, que devem se inspecionados de forma semelhante. A rota de inspeção em análise possui início na sala do operador da área do pelotamento. E, após inspecionar todos os equipamentos, o operador deve retornar ao local de partida. Habitualmente a rota é realizada a cada turno de 12h, ou seja, 2 vezes ao dia.

Devido às grandes dimensões, uma planilha com a matriz de distância entre os pontos de inspeção é disponibilizada em <https://l1nq.com/MatrizDeslocamento>. Essa matriz é composta pelo tempo que o inspetor gasta para se deslocar de um ponto $i \in P$ até qualquer outro ponto dentro do conjunto de pontos de inspeção P . O tempo de deslocamento é dado em segundos. Buscando tornar o tempo indicado para a execução da rota de inspeção mais próximo ao tempo gasto pelo inspetor, é necessário incluir o tempo gasto para inspecionar cada um dos pontos contidos na rota. Os tempos de inspeção (T_v) são apresentados na Tabela 1, em segundos.

Tabela 1: Tempo de inspeção - Estudo de caso.

Equipamento	Pontos	Tempo de inspeção (s)
Desagregador	P1, P3, P5, P7, P9, P11, P13, P15, P17, P19, P21, P23	90
Disco de pelotamento	P2, P4, P6, P8, P10, P12, P14, P16, P18, P20, P22, P24	240

5.2. Calibração dos Parâmetros

O algoritmo GRASP implementado possui dois parâmetros que devem ser calibrados para obter um melhor desempenho computacional: número de iterações sem melhora da solução ($iter_{max}$) e nível de aleatoriedade (α). Ambos os parâmetros foram calibrados utilizando o software Irace®. A Tabela 2 apresenta os valores analisados e os parâmetros escolhido.

5.3. Resultados

Para avaliar o desempenho do algoritmo GRASP proposto, foram realizadas 30 execuções para o cenário real. Os resultados foram comparados aos de uma solução manual de um planejador, uma solução do algoritmo construtivo guloso e uma solução obtida pelo resolvedor Lingo, versão

Tabela 2: Calibração de Parâmetros.

Parâmetro	Valores Analisados	Valor Escolhido
$iter_{max}$	1000, 1250, $\dots$, 2750, 3000	2000
α	0,05, 0,10, $\dots$, 0,20, 0,25	0,15

10, com limite de tempo de execução de 3.600 segundos. Para a implementação no resolvidor Lingo utilizou-se a formulação de programação linear inteira mista clássica do PCV.

Na Tabela 3 são apresentados estes resultados. A primeira coluna reporta o método de solução, a segunda coluna reporta o limite inferior encontrado pelo resolvidor matemático, a terceira coluna apresenta o valor da função objetivo e a quarta coluna mostra o tempo de execução. Para o GRASP são apresentadas o melhor, o pior e a média dos valores encontrados para a função objetivo nas 30 execuções. Destaca-se que o valor da função objetivo representa o tempo total gasto pelo inspetor para executar a rota planejada, incluindo o tempo de deslocamento e o tempo gasto nas inspeções.

Tabela 3: Comparação entre a solução atual, algoritmo construtivo guloso e algoritmo GRASP.

Método	Limite Inferior	Função Objetivo	Tempo de Execução (s)
Método exato	4.348,19	4.408,71	3.600,00
Solução manual	-	4.441,48	-
Construtivo guloso	-	4.462,28	1,00
GRASP	Melhor	4.396,64	10,44
	Pior	4.407,04	9,72
	Média	4.401,84	15,23

Observa-se que o resolvidor matemático não encontrou a solução ótima com tempo limite de 3.600 segundos. Contudo, a solução encontrada por ele foi melhor que a solução manual e que a solução gerada pelo algoritmo construtivo guloso. Já o GRASP foi capaz de superar o resolvidor matemático, a solução manual e o algoritmo construtivo guloso em todas as execuções, inclusive no pior caso. Além disso, o GRASP precisou de apenas 15 segundos na média para encontrar uma boa solução para o problema.

Na Figura 3 é apresentada a distribuição do conjunto de dados que representa o tempo total das 30 execuções realizadas, em segundos. É possível observar uma baixa variação entre os resultados obtidos.



Figura 3: Box plot do valor de função objetivo das execuções do GRASP.

A melhor solução encontrada pelo algoritmo GRASP para a instância em análise é exibida na Tabela 4 e na Figura 4, que ilustra o deslocamento realizado pelo inspetor. A rota de inspeção planejada cumpre as regras estabelecidas: tem como local de partida e chegada o ponto P0 (sala do operador) e todos os 24 equipamentos são inspecionados, sendo os pontos ímpares os desagregadores e os pontos pares os discos de pelotamento.

Tabela 4: Melhor solução para rota de inspeção.

Rota	Tempo total (s)
P0 ⇒ P13 ⇒ P14 ⇒ P10 ⇒ P9 ⇒ P5 ⇒ P6 ⇒ P2 ⇒ P1 ⇒ P3 ⇒ P4 ⇒ P8 ⇒ P7 ⇒ P12 ⇒ P11 ⇒ P15 ⇒ P16 ⇒ P20 ⇒ P19 ⇒ P24 ⇒ P23 ⇒ P21 ⇒ P22 ⇒ P18 ⇒ P17 ⇒ P0	4.396,64

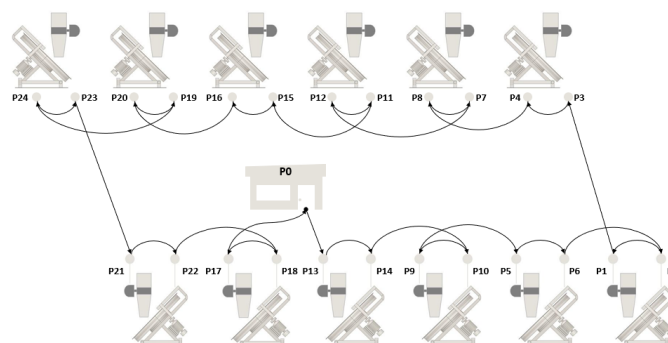


Figura 4: Melhor solução para rota de inspeção.

Partindo da sala do operador (P0), o primeiro equipamento inspecionado é o desgregador localizado no ponto P13, seguido pelo disco de pelotamento localizado em P14. Após essas inspeções, o operador se desloca até os pontos P10 e P9, onde realiza a inspeção do disco de pelotamento e do desgregador, respectivamente. A rota de inspeção segue pelos demais equipamentos apresentados na Figura 4, sendo o último equipamento inspecionado é o desgregador localizado em P17. Por fim, o operador retorna para P0 (sala do operador).

Observa-se que, de acordo com a melhor solução encontrada, a rota de inspeção planejada deverá ser realizada em 4.396,64 segundos (aproximadamente 1 hora e 13 minutos). Esse tempo de execução inclui 3960 segundos gastos para realizar as inspeções e 436,64 segundos gastos pelo inspetor para se deslocar de um ponto a outro.

Além de encontrar uma rota de inspeção que pode ser executada em menos tempo, o método proposto possui diversos benefícios, como a geração rápida da rota de inspeção e a eliminação da necessidade de reuniões multidisciplinares para definição do deslocamento da rota.

O modelo proposto também destaca-se pela capacidade de realizar o planejamento e revisar as rotas existentes em curtos intervalos de tempo. Conforme um equipamento é removido da rota, a matriz de deslocamento pode ser facilmente adaptada e o planejamento de uma nova rota é executado em poucos segundos. Ao contrário do processo atual, em que a retirada de um equipamento demanda uma nova rodada de conversas multidisciplinares para entender como adaptar a rota existente às novas necessidades. Desta maneira, o modelo proposto também permite que a necessidade de hibernação de um determinado equipamento seja facilmente tratada.

6. Considerações Finais

Este trabalho propõe a elaboração de um algoritmo capaz de planejar rotas de inspeções otimizadas com base no tempo de deslocamento entre os pontos de inspeção, buscando a redução do tempo total gasto pelo inspetor. As rotas de inspeção são formadas por uma série de equipamentos que devem ser checados periodicamente e indicam o caminho que o inspetor deve percorrer no chão fabril para coleta da informação de ativos. A observação dos ativos, na frequência e abrangência de itens inspecionáveis necessária, muitas vezes esbarra no quantitativo finito de recursos de mão de obra. Portanto, há a necessidade de que o recurso humano seja utilizado de maneira racional de forma a atender as necessidades de inspeção de maneira otimizada.

O algoritmo proposto facilitará a definição de novas rotas de inspeção, assim como a atualização das rotas já existentes em casos de parada de algum equipamento para manutenção, por exemplo. Além disso, o algoritmo ainda poderá ser aplicado às rotas já existentes, buscando otimizar os tempos dedicados às inspeções. Para avaliar o algoritmo proposto um estudo de caso utilizando dados reais da usina de pelotização Tubarão, localizada em Vitória, foi realizado. Buscando tornar o tempo indicado para a rota de inspeção mais próximo ao tempo gasto pelo inspetor em sua execução, o algoritmo proposto considerou o tempo gasto para inspecionar cada um dos pontos de inspeção contidos na rota.

Os resultados obtidos validam o algoritmo proposto com uma ferramenta de apoio ao planejamento de rotas de inspeção, buscando reduzir o tempo de descolamento do inspetor e trazendo uma estimativa mais realista do tempo total dedicado à rotina de inspeção. Em todas as 30 execuções realizadas o tempo total encontrado foi menor que o tempo gasto atualmente pelo operador. Comprovada a eficiência do algoritmo proposto, o potencial de aplicação dessa ferramenta em outras rotas e unidades operacionais mostra-se promissor, uma vez que a estrutura das rotas operacionais e das especialidades de manutenção são muito semelhantes.

Em trabalhos futuros, pretende-se implementar novas funções objetivo e desenvolver uma sim-heurística para tratar estocasticidades inerentes ao problema, como a variação do tempo de deslocamento e inspeção.

Agradecimentos

O presente trabalho foi realizado com apoio do CNPq (processo 302629/2023-8), CAPES (Código de Financiamento 001), Vale S.A. e Instituto Tecnológico Vale (ITV).

Referências

- Ambrosio, J. P. (2022). Técnicas de inspeção em correias transportadoras por manutenção de integridade estrutural: o caso de uma indústria de mineração. Trabalho de conclusão de curso em Engenharia Mecânica - Universidade Federal de Ouro Preto.
- Ferreira, M. d. S. (2023). Proposta de roteirização para o transporte público universitário de araguaína, TO. Trabalho de conclusão de curso em Tecnologia em Logística - Universidade Federal do Tocantins.
- Guimarães, I. F., Ouazene, Y., de Souza, M. C., and Yalaoui, F. (2019). Flowshop scheduling problem with parallel semi-lines and final synchronization operation. *Computers & Operations Research*, 108:121–133. ISSN 03050548. <https://10.1016/j.cor.2019.04.011>.
- Gusson, D. (2023). Plataforma para elaboração de roteiros turísticos otimizados por meio de heurísticas do caixeiro viajante com enfoque no agroturismo. Trabalho de conclusão de curso em Sistemas de Informação - Instituto Federal do Espírito Santo.
- Hu, Y., Zhang, Z., Yao, Y., Huan, X., Zhou, X., and Lee, W. S. (2021). A bidirectional graph neural network for traveling salesman problems on arbitrary symmetric graphs. *Engineering Applications of Artificial Intelligence*, 97:104061. <https://doi.org/10.1016/j.engappai.2020.104061>.
- Ilavarasi, K. and Joseph, K. S. (2014). Variants of travelling salesman problem: A survey. In *International conference on information communication and embedded systems (ICICES 2014)*, p. 1–7, India. IEEE. <https://10.1109/ICICES.2014.7033850>.

- João D. F. Jesus-Junior, M. J. F. S. and Cota, L. P. (2024). A hybrid metaheuristic algorithm for scheduling iron ore reclaiming in ports. *Engineering Optimization*, p. 1–18. <https://doi.org/10.1080/0305215X.2024.2340688>.
- Osaba, E., Yang, X.-S., and Del Ser, J. (2020). Traveling salesman problem: a perspective review of recent research and new results with bio-inspired metaheuristics. *Nature-inspired computation and swarm intelligence*, p. 135–164. <https://doi.org/10.1016/B978-0-12-819714-1.00020-8>.
- Rao, T. S. (2021). A simulated annealing approach to solve a multi traveling salesman problem in a fmcg company. *Materials Today: Proceedings*, 46:4971–4974. <https://doi.org/10.1016/j.matpr.2020.10.384>.
- Reis, D. A. S. d. et al. (2021). *Procedimento de otimização de rotas de inspeção em plantas industriais visando a minimização da dose de ruído*. PhD thesis, Universidade Federal de Uberlândia.
- Resende, M. G. C. and Ribeiro, C. C. (2019). Greedy randomized adaptive search procedures: Advances and extensions. In Gendreau, M. and Potvin, J. Y., editors, *Handbook of metaheuristics*, volume 272 of *International Series in Operations Research & Management Science*, p. 169–220. Springer, Cham, Switzerland, 3 edition. https://doi.org/10.1007/978-3-319-91086-4_6.
- Rezende, E. C. and Gomes, D. L. (2016). Análise dos resultados obtidos com aplicação de princípios da gestão de ativos no processo de inspeção de equipamentos em uma pelotização. In *Anais do IV Simpósio de Engenharia de Produção (SIMEP)*, p. 1–13, Recife.
- Sheng, Y. (2022). A computational optimization research on ant colony optimization for the traveling salesman problem. In *Journal of Physics: Conference Series*, volume 2258, p. 012006. IOP Publishing. <https://doi.org/10.1088/1742-6596/2258/1/012006>.
- Silva, V. R. d. and Barcelos, B. F. (2019). Aplicação do problema do caixeiro viajante para otimizar rota de entrega de uma distribuidora. *Faculdade Doctum de João Monlevade*, p. 1–19. <https://dspace.doctum.edu.br/handle/123456789/2309>.
- Teles, C. D. (2007). Desenvolvimento de um método para o planejamento da inspeção de equipamentos. Master's thesis, Universidade Federal do Rio Grande do Sul, Porto Alegre.
- Zhan, S.-h., Lin, J., Zhang, Z.-j., Zhong, Y.-w., et al. (2016). List-based simulated annealing algorithm for traveling salesman problem. *Computational intelligence and neuroscience*, 2016. <https://doi.org/10.1155/2016/1712630>.
- Zhang, P., Wang, J., Tian, Z., Sun, S., Li, J., and Yang, J. (2022). A genetic algorithm with jumping gene and heuristic operators for traveling salesman problem. *Applied Soft Computing*, 127: 109339. <https://doi.org/10.1016/j.asoc.2022.109339>.