

OTIMIZAÇÃO DA GESTÃO DO PÁTIO DE MANOBRAS FERROVIÁRIAS UTILIZANDO A META-HEURÍSTICA GRASP

Landerson Morais Alexandre

Juan Xabier Esteban de Aquino Calles

Leandro Colombi Resendo

Instituto Federal do Espírito Santo – Campus Serra
Bacharelado em Sistema de Informação

Rodrigo de Alvarenga Rosa

Universidade Federal do Espírito Santo
Programa de Pós-Graduação em Engenharia Civil

RESUMO

Considerando a importância da gestão do pátio ferroviário para a uma ferrovia, este trabalho propõe uma heurística para a otimização das manobras dos vagões em um pátio de classificação. Nesta parte do pátio, as locomotivas de manobra devem pegar vagões em um ponto e levar a outro ponto do pátio, respeitando restrições específicas de pátios ferroviários, como por exemplo, restrições de segurança. Com essa tarefa, a gestão desta parte do pátio é caracterizada como um problema de *pick-up and delivery* (PDP). Assim, neste trabalho é apresentado uma implementação da meta-heurística GRASP para o problema PDP considerando restrições ferroviárias. Os resultados numéricos destacam os ajustes dos parâmetros da heurística para o problema, bem como a viabilidade de seu uso em cenários reais.

ABSTRACT

The good railway yard management is imperative to an efficient rail system. Then, this work proposes a heuristic to optimize the movement of the wagons in a classification yard. In this part of yard these are locomotives responsible for pick up wagons at one place and leave them in another place in the yard, taking into account specific rail yards constraints, such as safety. As described, the management of a railway classification yard can be characterized as a pickup and delivery problem (PDP). Thus, in this work we present a computational method based on meta-heuristic GRASP to the PDP problem, regarding railway yard constraints. Numerical results highlight the heuristic parameter settings to the problem investigation, in addition its effectiveness to solve real-size instances.

1. INTRODUÇÃO

A principal motivação para estudos em problemas de logística de transporte, na maioria das vezes, são os altos custos que se tem com o transporte de cargas, que representam uma grande parcela do custo da matéria-prima e do produto final (Pereira, 2010). Segundo Toth e Vigo (2001), aplicações no mundo real, na América Latina e na Europa, mostram que o uso de procedimentos computadorizados para o planejamento produz economia significativa nos custos globais de transporte.

O Problema de Roteamento de Veículos (Vehicle Routing Problem – VRP) é o nome que se dá a classe de problemas, na área de otimização combinatória, que tem como objetivo estabelecer rotas eficientes para que veículos realizem coletas/entregas de cargas, tendo frota de veículos homogênea, ou não (Kumar e Panneerselvam, 2012). Uma variante dessa categoria inclui o Problema de Coleta e Entrega (*Pickup and Delivery Problem – PDP*) (Perretto et al., 2013), em que as mercadorias, ou pessoas tem que ser transportadas entre origens e destinos. No PDP, o veículo deve sair de um ponto de origem, coletar a(s) demanda(s), que possui informações sobre a carga e localização de origem, transportar até o destino da entrega, coletar a nova demanda e assim sucessivamente, até o fim do horizonte de planejamento.

Freitas e Montané (2008) apresentam duas meta-heurísticas para resolverem de forma aproximada o problema de Roteamento de Veículos com Coleta e Entrega Simultânea (PRVCEs). Uma delas é o GRASP-VND que utiliza o *Greedy Randomized Adaptive Search Procedure* (GRASP) e a heurística *Variable Neighborhood Descent* (VND) como método de busca local. A outra é o VNS-VND, que tem como componentes o *Variable Neighborhood Search* (VNS), que explora um conjunto de vizinhanças, e novamente, o VND como estratégia de busca local. Porém, o trabalho não trata de nenhum caso real, assim apenas as restrições clássicas são consideradas.

Mine et al (2010) propuseram um algoritmo que combina adaptações das heurísticas Inserção Mais Barata e GENIUS, a fim de gerarem a solução inicial, e as heurísticas Iterated Local Search (ILS), com o intuito de refinar a solução inicial, e VND como método de busca local, para resolver o VRP. Adicionalmente, neste trabalho também são consideradas as restrições clássicas do PDP.

Nesse trabalho, apresentamos uma aplicação prática em que a resolução desse tipo de problema impacta nos custos de uma empresa que utiliza o transporte ferroviário para escoamento de seus produtos/matérias-primas. A solução apresentada tem como intuito apoiar o planejamento operacional de pátios ferroviários, diminuindo o tempo utilizado para realizar o processo de planejamento e minimizando o tempo de realização das atividades do pátio ferroviário, visto que a capacidade de operação do transporte ferroviário está diretamente relacionada à qualidade de sua infraestrutura e com a eficiência na gestão de seus pátios e terminais, pois grande parte dos vagões e cargas encontra-se dentro destes locais.

Mais especificamente, a proposta desse trabalho é desenvolver uma heurística para o problema da gestão do pátio aplicado na área de classificação de um pátio, ou seja, onde os vagões são separados e reagrupados em blocos de acordo com um destino comum. Esse destino pode ser tanto o destino final da carga quanto outro pátio. A função do método desenvolvido é determinar a programação de cada locomotiva de manobra e, também, a melhor rota para cumprir a programação. O método computacional utilizado foi uma meta-heurística desenvolvida por Feo e Rezende (1995), chamada GRASP. Trata-se de um método não-determinístico que consiste em criar uma solução inicial viável e em seguida melhorar a qualidade da solução através de uma busca local.

O texto do artigo está organizado da seguinte maneira: o Capítulo 2 apresenta a descrição do problema estudado e os principais conceitos utilizados na área; no Capítulo 3 é descrito a modelagem computacional do problema; no Capítulo 4 é apresentada a heurística proposta usada para resolver o problema; no Capítulo 5 são apresentados os resultados computacionais e análises; e finalmente no Capítulo 6 as conclusões e considerações finais.

2. DEFINIÇÕES DO PROBLEMA

Os componentes fundamentais de um pátio ferroviário são: os vagões, que são veículos construídos com o propósito de transportar mercadorias, animais ou passageiros, porém estes dependem das locomotivas para se locomoverem; as locomotivas são os veículos que fornecem a energia necessária para movimentação dos vagões, movendo até algumas centenas de vagões. Elas podem atuar fornecendo tração a um conjunto de vagões entre os pátios, ou atuar dentro dos pátios, nesse caso elas são chamadas de locomotivas de manobras; as linhas, são por onde os trens circulam, e conectam estações, pátios e terminais. Tais linhas são

formadas por um conjunto de segmento de trilhos; a locomoção de um trem de uma linha para outra é realizada através de aparelhos de mudança de via (AMV). Esse dispositivo ajuda, também a resolver o problema de cruzamento em nível entre duas linhas. Eles são essenciais para as mudanças de direção em uma ferrovia e também em áreas de manobras de trens; os pátios ferroviários são onde os trens são desmembrados, descarregados, classificados e formados. Fisicamente, ele é uma vasta área composta de linhas paralelas conectadas por AMV's. A Figura 1 ilustra um pátio ferroviário em operação.



Figura 1: Área de um pátio ferroviário em operação.

A programação das operações das locomotivas de manobras é subdividida em unidades básicas chamadas de manobras. Uma manobra é o transporte de um conjunto de vagões, de um local para outro no pátio. Assim, ao realizar a coleta de um conjunto de vagões, a entrega precisa ser feita antes de realizar a próxima coleta. Esta programação é atendida em intervalos de tempo denominados horizontes de planejamento. Note que esta definição do problema se assemelha do problema coleta e entrega, o PDP. A Figura 2 ilustra o processo de coleta e entrega em um pátio de vagões. Seja e^+ a localização de uma locomotiva de manobra no início de um horizonte de programação. A locomotiva deve se dirigir a posição V_1^+ para coletar um conjunto de vagões, que em seguida será entregue na posição V_1^- . Após esta ordem de serviço ela deve coletar o próximo conjunto de vagões em V_2^+ e entregá-lo em V_2^- , e assim sucessivamente até o final desta programação em e^- . Assim, como dito, a estrutura básica do problema de gerenciamento de um pátio de vagões é análogo ao PDP (Sabino et al 2010).

Na literatura existem várias versões do PDP, porém foi encontrada apenas uma com as características semelhantes a este trabalho, em Sabino et al (2010), onde foi implementado uma heurística baseada em Colônia de Formigas para o planejamento de um pátio de manobras. Contudo, as informações contidas no trabalho não possibilitam a comparação dos resultados. No problema do planejamento das locomotivas de manobra em um pátio ferroviário, um das restrições que mais se diferenciam do PDP é a restrição de segurança para

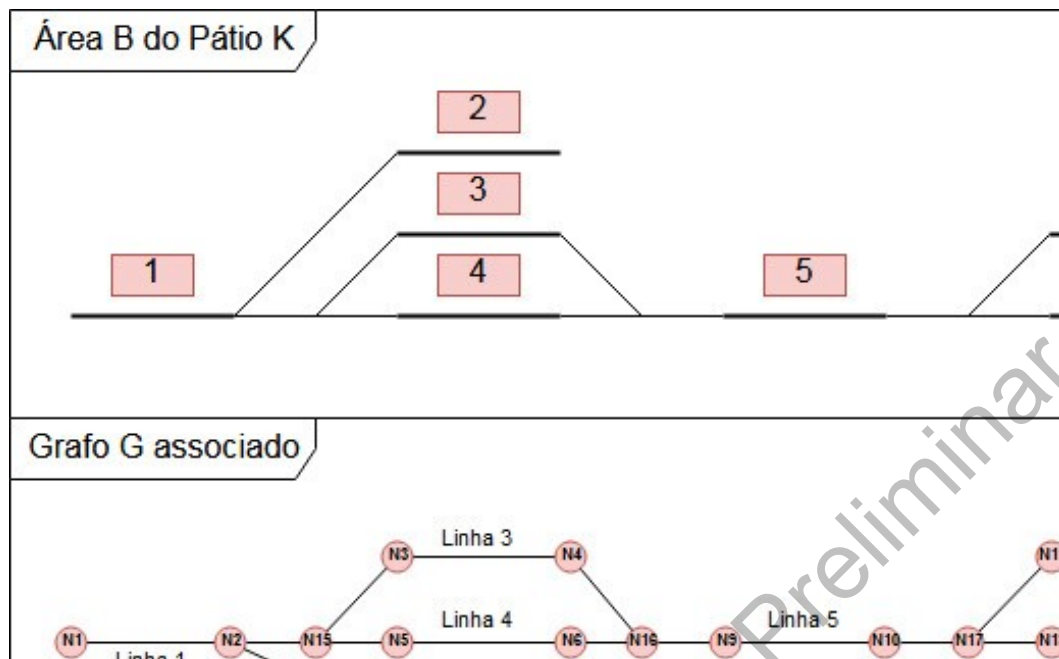


Figura 2: Grafo descritivo do *layout* de um pátio hipotético K.

Neste trabalho consideramos o custo como unidade de tempo. Ou seja, se o custo de um determinado arco que representa um segmento de trilho for x , isso significa que um objeto puntiforme demoraria x unidades de tempo para percorrer aquele arco.

O comprimento do lote é considerado, na solução, semelhante ao custo dos arcos representando segmentos de trilhos. Sendo assim, baseado na modelagem proposta para o problema, o comprimento de cada lote é medido, também, em unidades de tempo, visto que a velocidade de trabalho da locomotiva é constante. Então ao informar que um lote tem comprimento de x segmento significa que um objeto puntiforme gasta x unidades de tempo para percorrer o lote.

O caminho de coleta e entrega definido para uma locomotiva de manobras é um conjunto ordenado $O = \{Nc_1, Nc_2, Ne_1, Ne_2, \dots, Nc_{n-1}, Nc_n, Ne_{n-1}, Ne_n\}$ representando a sequência de nós a ser percorrida. Onde Nc_1 e Nc_2 correspondem ao arco (linha) onde é realizada a primeira coleta, e Ne_1 e Ne_2 correspondem ao arco onde é realizada a primeira entrega do conjunto de vagões, ou seja, a primeira manobra. Já Nc_{n-1} e Nc_n , e Ne_{n-1} e Ne_n , correspondem à coleta e entrega da última manobra.

4. MÉTODO UTILIZADO

No trabalho de Sosa *et al.* (2007), analisaram a complexidade de problemas de roteirização de veículos, e concluíram que praticamente todos os problemas de roteirização de veículos são NP-difícil. Também, de acordo com Parragh (2008) o problema tratado nesse trabalho também pertence a classe NP-difícil. Isto significa que a resolução do problema considerando instâncias maiores é necessário o uso de métodos heurísticos. Como dito anteriormente, para a resolução do problema foi escolhida uma meta-heurística *Greedy Randomized Adaptive Search Procedure* (GRASP), em português, Procedimento de Busca Adaptativo, Aleatório e Guloso.

A Figura 3 apresenta o pseudocódigo do algoritmo principal. Pode-se notar que existem três etapas presentes neste algoritmo. A primeira consiste em carregar a instância de entrada, ou seja, os dados do *layout* do pátio, bem como a localização das locomotivas de manobras e as manobras que deverão ser realizadas dentro do horizonte de tempo. Já a segunda etapa, detalhada na Figura 4, inclui uma construção sequencial de uma solução viável. A cada iteração, todos os possíveis movimentos são ranqueados de acordo com a função adaptativa e gulosa que leva em conta o estado presente, e um desses possíveis movimentos é escolhido de maneira aleatória, da Lista Restrita de Candidatos (LRC), para compor a solução. Nessa função a linha “Selecione aleatoriamente um elemento s da LRC” depende de um parâmetro α que regula a quantidade de elementos da lista que serão analisados. Este parâmetro será melhor discutido mais adiante.

```
proc grasp (MAX_ITER)
  carregarInstanciaEntrada()
  melhor_solucao = nulo
  para k=1 até MAX_ITER faça
    solucao = constroiSolucao()
    solucao = buscaLocal()
    atualizaSolucao(solucao, melhor_solucao)
  # fim para
  retorna melhor_solucao
# fim grasp
```

Figura 3: Pseudocódigo que descreve o GRASP.

```
proc constroiSolucao ()
  Solucao = {}
  enquanto Solucao não está completa faça
    Construa LRC
    Selecione aleatoriamente um elemento  $s$  da LRC
    Solucao = Solucao  $\cup$  { $s$ }
    Adapte a função gulosa
  fim enquanto
  retorna (Solucao)
fim constroiSolucao
```

Figura 4: Pseudocódigo que descreve a fase de construção do GRASP.

Na terceira etapa, é usado uma busca local para chegar a um ótimo local. Pelo fato de todo o procedimento ser repetido várias vezes, é indispensável uma fase de construção eficiente. E, além disso, é preferível construir um número de soluções viáveis e, então, aplicar a busca local para a mais promissora das soluções, devido à complexidade da busca local.

```
proc buscaLocal (Solucao)
  enquanto Solucao  $s$  não é um ótimo local faça
    Encontre uma solução  $t$  em  $V(s)$ 
    se custo( $t$ ) > custo(Solucao) entao
      Solucao =  $t$ 
    fim se
  fim enquanto
  retorna (Solucao)
fim buscaLocal
```

Figura 5: Pseudocódigo que descreve a fase de busca em vizinhança.

A meta-heurística implementada é utilizada como apoio à decisão para definir o plano de

trabalho das locomotivas. Ou seja, o GRASP será utilizado para definir qual locomotiva deve executar qual manobra e também a ordem em que deve ser realizada a execução destas manobras, sem que esse plano viole as restrições operacionais.

No início da fase construtiva, a estrutura que guarda a programação de cada locomotiva está vazia. A cada iteração, enquanto o horizonte de tempo não chegou ao fim é construído a lista restrita de candidatos (LRC). A sua construção é feita com base na lista de manobras e a localização atual da locomotiva em questão, então a LRC é reordenada, de forma crescente, a partir da distância existente entre a localização atual da locomotiva e a localização de coleta das manobras. Em seguida, é escolhido aleatoriamente uma manobra (par coleta-entrega) desta lista e, então, adicionada à estrutura que guarda a programação da locomotiva. A execução desta fase termina quando todas manobras forem atendidas ou exceder o horizonte de planejamento.

Em seguida, inicia-se a segunda fase, que é responsável por fazer melhoria na solução construída na fase anterior. O critério escolhido para a estrutura de vizinhança foi alterar os pares de coleta-entrega vizinhos. A cada iteração é feita a permuta. Então é verificado se a solução foi melhorada. Caso tenha melhorado, a solução melhorada substitui a anterior, caso contrário, o estado anterior da solução é restaurado. A fase de busca local termina quando a vizinhança escolhida é explorada por completo. Ao fim dessa fase, o ótimo local é encontrado.

Como a obtenção do resultado é de forma adaptativa, gulosa e aleatória, faz-se necessário verificar se a solução encontrada é superior em relação a encontrada na iteração anterior. A execução do GRASP termina quando o máximo de iterações definido é atingido. Ao fim da execução da meta-heurística a melhor solução encontrada, até aquele momento, é retornada.

5. RESULTADOS COMPUTACIONAIS

Nesta seção são apresentados os resultados computacionais obtidos pelo algoritmo GRASP proposto. Ele foi desenvolvido na linguagem Python, utilizando o ambiente Pycharm Community Edition 2017.1. Os testes foram executados em um computador com o processador Intel Core i7-2600 3.4 GHz, 8GB de memória RAM, sistema operacional Windows 7. Com esta configuração, todas as instâncias foram executadas em menos de um minuto.

A heurística é avaliada utilizando o *layout* do pátio de manobras de Uvaranas, no Paraná (Cavalcanti, 1986). Como não havia informações sobre os comprimentos das linhas, estas foram determinadas, aproximadamente, a partir das medidas do *layout*. O pátio foi modelado conforme o grafo apresentado na Figura 6. O grafo obtido contém 81 nós e 105 arestas. A Tabela 1 apresenta os parâmetros para o cenário proposto. Nos experimentos foram definidas 100 manobras a serem executadas num horizonte de tempo de 6 horas (360 minutos). A unidade de medida de tempo definida para servir de custo para as arestas foi o minuto.

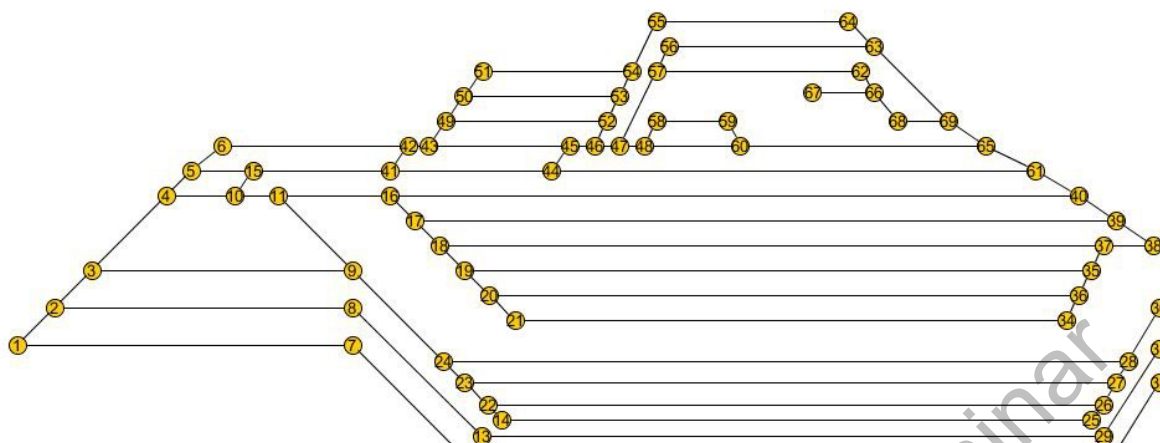


Figura 6: Grafo do pátio utilizado nos testes.

Tabela 1: Parâmetros do GRASP utilizados nos experimentos computacionais

Parâmetro	Notação	Valores
Horizonte de planejamento (minutos)	<i>HorizonteTempo</i>	360
Número de manobras realizadas	<i>Manobras</i>	100
Número de iterações	<i>maxIterations</i>	5; 10; 15; 20; 30; 40; 50; 100; 200; 500
Valor de limitação da LCR	<i>alpha</i>	0,1; 0,2; 0,3; 0,4; 0,5; 0,6; 0,7; 0,8; 0,9; 1

Para a obtenção dos resultados computacionais, o cenário proposto foi testado utilizando diferentes valores para os parâmetros da heurística. O parâmetro *maxIterations* determina o número de iterações que a heurística fará para determinada combinação de parâmetros. Enquanto que *alpha* determina a porcentagem da lista de candidatos restrita (LCR) baseada em 100 soluções viáveis.

As Figuras de 7 e 8 mostram os resultados obtidos nos testes, ambos variando o parâmetro *maxIterations*. Note que a Figura 8 é complementar à Figura 7. Os resultados foram separados em duas figuras para facilitar o entendimento. Em ambas, o eixo das abscissas corresponde à variação do parâmetro *alpha* enquanto o eixo das ordenadas corresponde ao tempo gasto pela locomotiva que mais demorou para realizar seu conjunto de manobras, ou seja, o tempo máximo das manobras para o horizonte de planejamento.

Os resultados apontaram que quanto mais gulosa é a construção da solução, ou seja, quanto mais próximo de zero é o valor de *alpha*, pior é a qualidade da solução final. E melhores soluções são obtidas ao construí-la de maneira totalmente aleatória, ou seja, quando *alpha* é igual a 1.

Outra característica encontrada ao comparar os gráficos das Figuras 7 e 8 é que houve uma melhora considerável da qualidade da solução quando se fixa *alpha* em 1 e varia do parâmetro *maxIterations*. De acordo com os resultados obtidos, a qualidade da solução ficou cerca de 20,80% melhor quando *maxIterations* assume valor 500 em relação a quando assume valor 5. Porém não há melhoria na qualidade da solução quando *alpha* assume o valor 0,1.

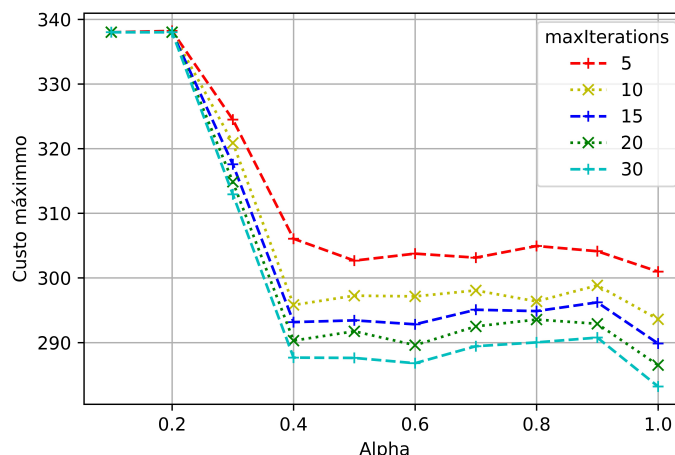


Figura 7: Gráfico de Custo vs. Alpha, para maxIterations = 5, 10, 15, 20, 30.

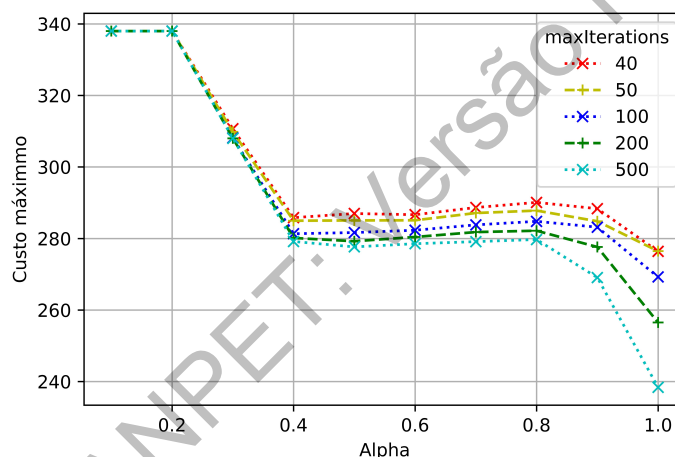


Figura 8: Gráfico de Custo vs. Alpha, para maxIterations = 40, 50, 100, 200, 500.

5. CONCLUSÃO

Nesse trabalho foi desenvolvida uma heurística, utilizando a meta-heurística GRASP, para o problema da gestão de um pátio de manobras ferroviário, mais especificamente o pátio de classificação.

O pátio de classificação é a região onde, após serem descarregados, os vagões dos trens classificados, são separados e enviados para diferentes pontos no pátio. Como os vagões não possuem tração própria, para isso existem locomotivas chamadas de locomotivas de manobras. A tarefa destas locomotivas consiste em pegar um conjunto de vagões em um ponto do pátio e levar a outro ponto. Sendo que essas operações devem respeitar um conjunto de restrições inerentes aos pátios ferroviários, como por exemplo, restrições de segurança. Dadas as características no problema abordado, constatou-se que este se encaixava nas definições do problema clássico de *pick-up and delivery problem* (PDP), com algumas restrições. Assim, usando os conceitos do problema clássico PDP foi desenvolvida a heurística GRASP proposta neste trabalho.

Analisando a heurística proposta, nos resultados numéricos é encontrada a configuração ideal para esta heurística, considerando o problema investigado, bem como a eficiência da mesma para a resolução deste problema. Isto é, mostramos que seria viável a aplicação do método proposto em cenários reais.

Como trabalhos futuros, pretende-se propor e implementar um modelo de programação linear inteira mista para obtenção da solução exata de instâncias. Assim, validar qualitativamente a heurística GRASP proposta nesse trabalho.

Agradecimentos

Este trabalho foi financeiramente apoiado pelo CNPq por meio do projeto Otimização do Pátio de manobras da Companhia Vale do Rio Doce (Processo nº 488088/2013-5)

REFERÊNCIAS BIBLIOGRÁFICAS

- Cavalcanti, F. (1986) O novo pátio de Uvaranas, PR. Disponível: <https://goo.gl/GQ3CdE> . Acesso em 20 Nov. 2017.
- Feo, T. A. e Resende, M. G. C.(1995). Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6:109–133. Disponível: <https://goo.gl/dQNVub> . Acesso em 10 Nov 2017.
- Sabino, J.A. et al., (2010) A Multi-Objective Ant Colony Optimization Method applied to Switch Engine Scheduling in Railroad Yards, *Pesquisa Operacional*, v.30, n.2, p.487-514.
- Freitas, L.M.R. e Montané, F.A.T. (2008) Metaheurísticas VNS-VND e GRASP-VND para Problemas de Roteamento de Veículos com Coleta e Entrega Simultânea, *XVIII Simpósio de Pesquisa Operacional e Logística da Marinha (SPOLM)*.
- Kumar, S.N, e Panneerselvam, R. (2012) A Survey on the Vehicle Routing Problem and Its Variants, *Intelligent Information Management*, v. 4, p. 66-74.
- Perretto, P.R., et al., (2013) Resolução de um caso real do problema dial-a-ride multicritério via clustering search. *Produção*, v. 23, p. 572-582.
- Mine, M. T., et al. (2010). O problema de roteamento de veículos com coleta e entrega simultânea: Uma abordagem via Iterated Local Search e GENIUS. Capítulo 3 do Livro: *Transporte em Transformação Volume XIV*, pg. 61-80, ISBN: 978-85-99082-10-2.
- N. Parragh, Sophie & Doerner, Karl & Hartl, Richard. (2008). Part II: Transportation between pickup and delivery locations. *Journal für Betriebswirtschaft*. 58. 81-117.
- Parragh, S.N., Doerner, K.F. e Hartl, R.F. (2008) A survey on pickup and delivery problems, *Journal für Betriebswirtschaft*, v. 58, Issue 1, p. 21–51.
- Pereira, L. D. (2010) Heurísticas e algoritmos exatos para o problema de roteamento de veículos com coleta e entrega simultâneas. Dissertação de Mestrado, Instituto de Ciências Exatas da Universidade Federal de Minas Gerais.
- Sosa, N. G. M., et.al.(2017) Algoritmo de busca dispersa aplicado ao problema clássico de roteamento de veículos. *Pesquisa Operacional* , Rio de Janeiro, v. 27, n. 2, p. 293-310.
- Toth, P. e Vigo, D.(2001) *The Vehicle Routing Problem*,Capítulo de *An Overview of Vehicle Routing Problems*. SIAM.

Landerson Morais Alexandre (landerson.lma@hotmail.com)
Juan Xabier Esteban de Aquino Calles(callesjuan@gmail.com)
Leandro Colombi Resendo (leandro@ifes.edu.br)
Bacharelado em Sistema de Informação, Serra, Instituto Federal do Espírito Santo
Rodovia ES-010 - Km 6,5 – Manguinhos, CEP: 29173 - 087 - Serra - ES
Rodrigo de Alvarenga Rosa(rodriigo.a.rosa@ufes.br)
Programa de Pós-Graduação em Engenharia Civil, Universidade Federal do Espírito Santo Santo
Av. Fernando Ferrari, 514, Goiabeiras – Vitória, ES, Brasil