ALGORITMO ILS PARA O PROBLEMA DE TRANSPORTE DE MÁQUINAS EM FLORESTA DE PRODUÇÃO

Romulo Borges de Almeida, Bruno Fonseca do Carmo, Antonio Almeida de Barros Junior, Geraldo Regis Mauri, Gilson Fernandes da Silva
Departamento de Computação - Universidade Federal do Espírito Santo
Alto Universitário, s/n, Campus Universitário, Alegre – ES, 29500-000
romulo.almeida@edu.ufes.br, bruno.carmo@edu.ufes.br,
antonio.barros@ufes.br, geraldo.mauri@ufes.br,
gilson.silva@ufes.br

Giovanni Correia Vieira
Instituto Federal de Educação, Ciência e Tecnologia de Rondônia, Campus de Ji-Paraná
R. Rio Amazonas, 151 - Jardim dos Migrantes, Ji-Paraná - RO, 76960-000
giovanni.vieira@ifro.edu.br

RESUMO

Neste trabalho é abordado uma variante do Problema de Roteamento de Veículos (PRV) que existe no planejamento da colheita florestal, o Problema de Roteamento de Veículos com Coleta e Entrega, Janelas de Tempo e Frota Heterogênea (PRVCEJTFH), onde tem como função objetivo a minimização dos custos de transporte, de atrasos e número de veículos. Para resolução de tal problema, foi utilizado dois métodos, o método exato *Branch and Bound* por meio do *software* CPLEX e a meta-heurística *Iterated Local Search* (ILS). Com o propósito de validar o modelo e testar os métodos, foram criadas 12 instâncias, a partir de dados reais, para o problema. Os resultados obtidos mostraram-se promissores para algumas instâncias, em especial aqueles obtidos pela meta heurística ILS cujo desempenho foi superior ao do método exato *Branch and Bound*.

PALAVRAS CHAVE. Problema de Roteamento de Veículos, Transporte Florestal, ILS.

AG&MA – PO na Agricultura e Meio Ambiente

POI – PO na Indústria

ABSTRACT

This paper presents a variant of the Vehicle Routing Problem (PRV) that exists in forest planning, the Heterogeneous Fleet Vehicle Routing Problem with Time Windows and Pickup and Delivery (HFVRPTWPD), which has the objective function of minimizing their transport costs, the delay and the number of vehicles. To solve this problem, two methods were used, the exact *Branch and Bound* method using the CPLEX solver and the *Iterated Local Search* (ILS) metaheuristic. To validate the model and test the methods, 12 instances were created for the problem. The results obtained proved to be promising for some instances, especially those obtained by the ILS metaheuristic whose performance was superior than the exact *Branch and Bound method*.

KEY WORDS. Vehicle Routing Problem, Forest Transport, ILS.

AG&MA – OR in Agriculture and Environment

POI – OR in Industry

1. Introdução

Dados recentes apresentados no relatório da Indústria Brasileira de Árvores (IBÁ), mostram que o setor de árvores cultivadas gera emprego e renda para o país, combinada com uma atuação socialmente e ambientalmente responsável. Em 2019, o setor registrou uma receita bruta de R$ 97,4 bilhões, um crescimento de 12,6% em relação ao ano anterior. Além disso, devido à grande demanda do comércio externo, o Brasil tem se destacado como o maior exportador de celulose no comércio mundial, com movimentação financeira de aproximadamente US$ 7,5 bilhões [IBÁ 2020].

É notável como o setor florestal produtivo influencia na economia de um país, assim como o seu planejamento influencia em sua produção, desta forma, é de suma importância que haja um planejamento efetivo das etapas que ocorrem na produção florestal, com o intuito de se obter o melhor resultado com o menor custo. Por outro lado, quando o planejamento é feito de uma forma inadequada, pode-se comprometer a produção, aumentando os seus custos e gerando um fraco resultado [Rodrigues 2018].

Dentre as atividades da produção florestal, a colheita e o transporte se destacam por efeito da sua alta participação nos processos de produção, com isso são as que mais afetam o custo final do produto [Machado e Lopes 2000]. Segundo Andrade [1998], cerca de 25% dos custos de produção são voltados à colheita e ao transporte. Desta forma, vale também para o transporte das máquinas de colheita, onde, quando feito um bom planejamento de sua logística, diminui o seu atraso, aumentando o número de suas coletas e consequentemente obtendo um maior lucro.

Para se obter bons resultados no planejamento, é interessante o uso de ferramentas que possam auxiliar no seu desenvolvimento, como por exemplo, a Pesquisa Operacional (PO), que atua em cima de um modelo matemático constituído por uma função objetivo e por equações e inequações matemáticas, onde busca a maximização ou minimização desta função objetivo respeitando tais equações e inequações (restrições do problema) [Hillier e Lieberman 2013].

O problema em que este trabalho aborda, é classificado como um Problema de Roteamento de Veículos (PRV), sendo formulado inicialmente por Dantzig e Ramser [1959], quando estudavam sobre a distribuição de gasolina para estações de venda de combustíveis. Este problema é caracterizado no atendimento de consumidores através de uma frota de veículos, com o propósito de definir a menor rota e/ou menor tempo, respeitando suas restrições. Mais especificamente, este trabalho se trata de uma extensão do PRV, o Problema de Roteamento de Veículos com Coleta e Entrega, Janelas de Tempo e Frota Heterogênea (PRVCEJTFH).

O PRVCEJTFH é definido pela logística de transporte de um certo item, onde se realiza sua coleta em determinado ponto e posteriormente sua entrega em outro ponto, no qual cada item possui um intervalo de coleta e de entrega, e nos veículos de transporte existe um limite de carga, ou seja, nem todo item pode ser transportado por qualquer veículo [Wang et al. 2015]. Especificamente nesta pesquisa, o PRVCEJTFH aborda o transporte de máquinas, onde os itens são as máquinas, os pontos de coleta e entrega são os povoamentos florestais (talhões) e os veículos para o transporte são caminhões tipo plataforma e veículos articulados de 2, 3 e 4 eixos com diferentes capacidades de carga. Problema este, muito comum no planejamento e execução das atividades de colheita florestal.

O modelo matemático utilizado nesta pesquisa foi apresentado e validado por Vieira [2018], que utilizou o método exato *Branch and Bound* através do *software* CPLEX, com o intuito de se obter a melhor solução para uma pequena instância do problema. No entanto, por se tratar de uma variação do PRV, o qual é classificado como um problema NP-Difícil, a inserção de mais pontos, torna-o muito complexo. Segundo Goldbarg e Luna [2005], dentre os problemas da área de otimização combinatória, os PRV's são considerados os mais complexos. Dito isso, os métodos exatos nem sempre são suficientes para resolução de problemas desta categoria, para tanto, foi proposto o desenvolvimento da meta-heurística *Iterated Local Search* (ILS), com o objetivo de avaliar seu desempenho em relação ao método exato, especialmente no caso de instâncias maiores.

2. Revisão da Literatura

Essa seção tem como finalidade apresentar uma breve revisão dos problemas semelhantes a este encontrados na literatura, tais como: Pessuto [2018], Carabetti et al. [2010], Soldati e Vianna [2017] e Vieira [2018]. Todos estes trabalhos se referem a Problemas de Roteamento de Veículos com Coletas e Entregas e Janelas de Tempo (PRVCEJT), porém apenas Pessuto [2018], Carabetti et al. [2010] e Soldati e Viana [2017] fizeram uso de meta-heurísticas como método de solução para seus respectivos problemas, enquanto Vieira [2018] utilizou um método exato como forma de solução.

Pessuto [2018], em seu trabalho, aborda sobre um problema de logística recorrente em uma transportadora rodoviária de cargas, no qual por terem uma grande representatividade em seu meio, geram um alto valor na economia, exigindo uma excelência operacional. Com isso, as transportadoras devem planejar suas rotas de forma eficiente atendendo todas restrições impostas. Feito isso, Pessuto [2018] propôs o desenvolvimento de um método híbrido, formado pela combinação das meta-heurísticas Busca Tabu (BT) e Algoritmo Genético (AG), o qual apresentou bons resultados, com valores superiores ou bem próximos a alguns da literatura, conseguindo gerar uma economia superior a R$ 500.000,00 ao ano.

Carabetti et al. [2010] apresentou uma forma de solucionar um PRVCEJT, utilizando a meta-heurística *Ant Colony System* (ACS – Sistema de Colônias de Formigas). Foi utilizado um conjunto de instâncias deste problema da literatura, a fim de testar a eficiência dos algoritmos para solucionar este tipo de problema. Em seu trabalho, Carabetti et al. [2010] fez uma alteração no ACS original, adotando uma busca local conhecida como Descida com Primeira de Melhora, realizando duas explorações distintas nos vizinhos, relacionadas com suas estratégias de refinamento para o melhoramento das soluções iniciais. Conclui-se que através dos métodos utilizados, o autor conseguiu gerar soluções próximas às melhores da literatura.

Soldati e Vianna [2017] teve como objetivo, listar as técnicas e meta-heurísticas mais utilizadas nas soluções desse tipo de problema, com intuito de se desenvolver uma meta-heurística baseando-se nessas técnicas e métodos encontrados. Sua meta-heurística foi desenvolvida combinando duas estratégias, o ILS e o *Variable Neighborhood Descent* (VND - *pesquisa de vizinhança variável*), com os métodos de *Dijkstra* e *Push-Foward Insertion Heuristic* (PFIH) para a construção da solução inicial.

Vieira [2018] propôs o uso do PRVCEJTFH no transporte de máquinas florestais, apresentou um novo modelo para o problema e obteve soluções por meio do método exato *Branch and Bound*. Não foram encontrados na literatura, outros trabalhos envolvendo o PRVCEJT especificamente no transporte de máquinas florestais.

3. Modelo Matemático

Conforme citado anteriormente, nesta pesquisa utilizou-se o modelo matemático PRVCEJTFH proposto por Vieira [2018], adaptado do modelo PRVCEJT descrito por Cordeau [2006]. O PRVCEJTFH tem como objetivo minimizar o custo das rotas percorridas por cada transporte ponderada pela qualidade da estrada, o atraso no atendimento dos talhões e das garagens, e o número de caminhões utilizados, tais termos estão descritos respectivamente na função objetivo (1).

As restrições (2) e (3), restringem que os caminhões só podem iniciar suas rotas pela garagem inicial e finalizar pela garagem final. A restrição (4), tem a função de restringir que cada atendimento nos talhões seja realizado uma única vez e por um único caminhão. A restrição (5), garante que o atendimento da coleta para a entrega seja feito pelo mesmo caminhão. Já as restrições (6) e (7) garantem a contenção de fluxo, ou seja, a mesma quantidade de máquinas que entram em um caminhão é a mesma que sai. Quando um caminhão realiza uma coleta (no ponto *i*) de uma

máquina e posteriormente realiza sua entrega (no ponto *j*), a restrição (8) garante que o instante de tempo no ponto *j* seja maior que a soma do instante de tempo no ponto *i*, com tempo de serviço gasto neste mesmo ponto (carregamento ou descarregamento da máquina) e com o tempo de duração da rota (*i, j*). O conjunto de restrições (9) e (10), definem respectivamente os limites inferiores e superiores da janela de tempo dos talhões. A restrição (11), define se é permitido ocorrer atrasos nos talhões, onde na restrição (10), é somado ao seu limite superior.

Considerando que a rota (*i, j*), será percorrida pelo caminhão *k*, ou seja, $x_{i,j}^{k} = 1$, a restrição (12) garante que após o caminhão *k* sair do ponto *j*, a carga do caminhão neste ponto seja maior ou igual a carga do mesmo caminhão no ponto *i*, acrescida da carga do ponto *j*. As capacidades de carga dos caminhões em cada talhão são definidas pelas restrições (13) e (14), como o cada caminhão só pode transportar uma única máquina por vez e estar sempre sem carga nas garagens, as capacidades que os caminhões devem ter nas garagens e nos pontos de entrega é igual a zero e nos pontos de coleta igual a um. Como existem quatro tipos diferentes de caminhões, com capacidades de cargas diferentes, as restrições (15), (16), (17) e (18) garantem que cada caminhão transporte as máquinas referentes ao seu tipo. A restrição (19) define as variáveis dos conjuntos A_i, B_i, $M_{i,j}$, Q_i e $W_{i,j}$ como reais positivas, e a restrição (20) define as variáveis $x_{i,j}^{k}$'s como binárias.

Minimizar:

$$\omega_1 \sum_{i \in N} \sum_{\substack{j \in N \\ j \neq i}} \sum_{k \in K} C_{i,j} x_{i,j}^{k} + \omega_2 \sum_{i \in N} A_i + \omega_3 \sum_{j \in P} \sum_{k \in K} x_{g_k^-,j}^{k} \tag{1}$$

Sujeito a:

$$\sum_{j \in \left\{P \cup \{g_k^+\}\right\}} x_{g_k^-,j}^{k} = 1 \qquad \forall\, k \in K \tag{2}$$

$$\sum_{i \in \left\{D \cup \{g_k^-\}\right\}} x_{i,g_k^+}^{k} = 1 \qquad \forall\, k \in K \tag{3}$$

$$\sum_{\substack{j \in \{P \cup D\} \\ j \neq i}} \sum_{k \in K} x_{i,j}^{k} = 1 \qquad \forall\, i \in P \tag{4}$$

$$\sum_{\substack{j \in \{P \cup D\} \\ j \neq i}} x_{i,j}^{k} - \sum_{\substack{j \in \left\{P \cup D \cup \{g_k^+\}\right\} \\ j \neq i, j \neq n+i}} x_{n+i,j}^{k} = 0 \qquad \forall\, i \in P, k \in K \tag{5}$$

$$\sum_{\substack{j \in \left\{P \cup D \cup \{g_k^-\}\right\} \\ j \neq i, j \neq n+i}} x_{j,i}^{k} - \sum_{\substack{j \in \{P \cup D\} \\ j \neq i}} x_{i,j}^{k} = 0 \qquad \forall\, i \in P, k \in K \tag{6}$$

$$\sum_{\substack{j \in \{P \cup D\} \\ j \neq i}} x_{j,i}^{k} - \sum_{\substack{j \in \left\{P \cup D \cup \{g_k^+\}\right\} \\ j \neq i, j \neq n-i}} x_{i,j}^{k} = 0 \qquad \forall\, i \in D, k \in K \tag{7}$$

$$B_j \geq B_i + d_i + t_{i,j} - M_{i,j}\left(1 - x_{i,j}^{k}\right) \qquad \forall\, i, j \in N / i \neq j \quad \forall\, k \in K \tag{8}$$

$$B_i \geq e_i \qquad \forall\, i \in N \qquad (9)$$

$$B_i \geq l_i + A_i \qquad \forall\, i \in N \qquad (10)$$

$$A_i \begin{cases} \leq \infty, & se\ for\ permitido\ atraso\ no\ atendimento \\ = 0, & caso\ contrário \end{cases} \qquad \forall\, i \in N \qquad (11)$$

$$Q_j \geq Q_i + q_j - W_{i,j}\left(1 - x_{i,j}^k\right) \qquad \forall\, i,j \in N / i \neq j \quad \forall\, k \in K \qquad (12)$$

$$Q_i = 0 \qquad \forall\, i \in D \cup \{g_k^-, g_k^+\} \qquad (13)$$

$$Q_i = 1 \qquad \forall\, i \in P \qquad (14)$$

$$\sum_{k \in K_1} x_{i,i+n}^k = 1 \qquad \forall\, i \in P_1 \qquad (15)$$

$$\sum_{k \in K_2} x_{i,i+n}^k = 1 \qquad \forall\, i \in P_2 \qquad (16)$$

$$\sum_{k \in K_3} x_{i,i+n}^k = 1 \qquad \forall\, i \in P_3 \qquad (17)$$

$$\sum_{k \in K_4} x_{i,i+n}^k = 1 \qquad \forall\, i \in P_4 \qquad (18)$$

$$A_i, B_i, M_{i,j}, Q_i, W_{i,j} \in \mathbb{R}^+ \qquad \forall\, i,j \in N / i \neq j \qquad (19)$$

$$x_{i,j}^k \in \{0,1\} \qquad \forall\, i,j \in N / i \neq j \quad \forall\, k \in K \qquad (20)$$

Em que:
$x_{i,j}^k$ = Variáveis da solução, se 1, o carro *k* vai passar pela rota (*i, j*), caso contrário é igual a 0;
g_k^-, g_k^+ = Garagem inicial e a garagem final, respectivamente;
K = Conjunto dos caminhões;
P, D = Conjunto dos talhões de coleta (pick-up) e de entrega (delivery), respectivamente;
N = Conjunto de todos os pontos (todos talhões e as duas garagens);
P_i = Conjunto dos talhões que podem ser atendidos pelos caminhões do tipo *i*, onde $i \in [1,4]$;
K_i = Conjunto dos caminhões do tipo *i*, onde $i \in [1,4]$;
A_i = Conjunto dos atrasos que podem ocorrer ou não nos talhões, com $i \in N$;
B_i = Conjunto dos instantes de tempo que cada talhão foi atendido, com $i \in N$;
$M_{i,j}, W_{i,j}$ = Variável para representar o infinito, definido como 9999, sem prejudicar a solução;
Q_i = Conjunto de capacidade de carga em cada talhão, com $i \in N$;
$C_{i,j}$ = Conjunto do custo da rota (i, j);
e_i, l_i = Conjunto das janelas inferiores e superiores de cada talhão, respectivamente, com $i \in N$;
$t_{i,j}$ = Conjunto do tempo de deslocamento do talhão *i* para o *j*, com $i, j \in N$;
d_i = Conjunto do tempo de serviço em cada talhão, com $i \in N$;
q_j = Conjunto de carga de cada talhão, com $j \in N$;

4. Metodologia

A Figura 1 ilustra as etapas metodológicas utilizadas no desenvolvimento deste trabalho. Inicialmente foram elaboradas as instâncias para o problema abordado por Vieira [2018]. De posse

das instâncias, pode-se desenvolver o procedimento de execução do método exato *Branch and Bound* e a meta-heurística ILS. E por fim, analisou-se os resultados obtidos por estes dois métodos.

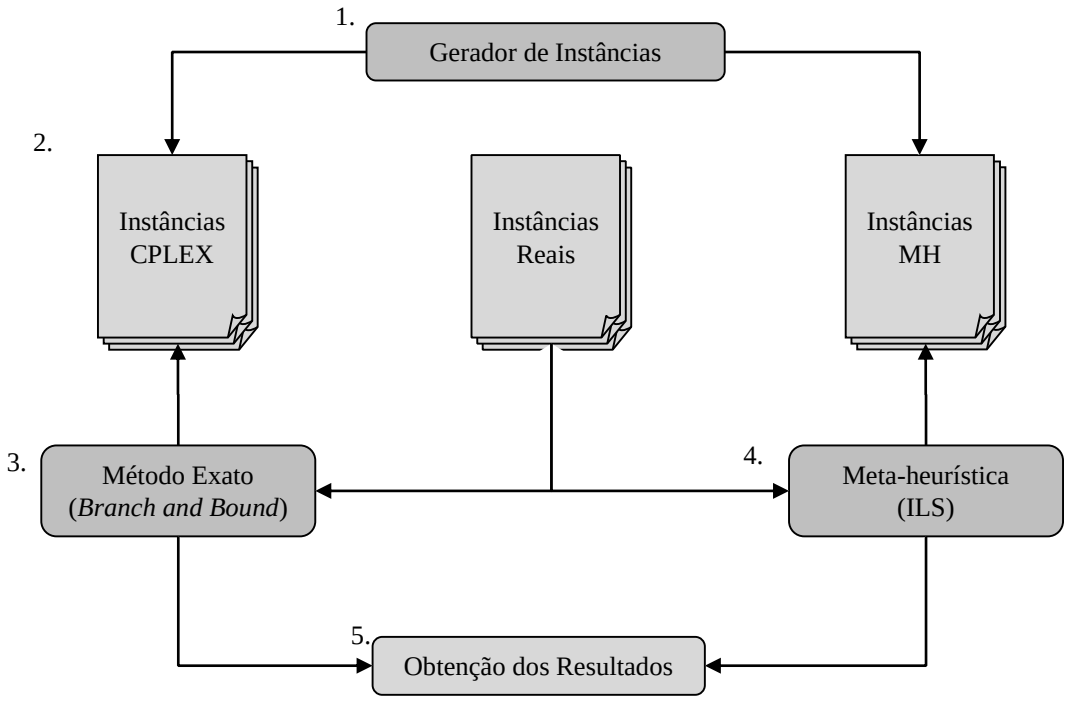


Figura 1: Etapas Metodológicas.

4.1. Etapa 1: Gerador de Instâncias

Tendo em vista que apenas as instâncias reais deste problema não seriam suficientes para a análise do desempenho dos métodos de solução propostos, notou-se a necessidade de gerar novas instâncias, sendo assim, foi desenvolvido um sistema capaz de gerar instâncias maiores de forma que tanto o método exato quanto a meta-heurística fossem capazes de processar. O sistema exige como entrada, o número de máquinas (QtdMaq) que deverão ser transportadas. A partir desta entrada, duas instâncias são geradas com os mesmos dados, uma direcionada ao método exato e outra direcionada à meta-heurística. Os demais dados das instâncias são definidos da seguinte forma:

- Número de Caminhões = QtdMaq. Cada caminhão é responsável pelo transporte de uma máquina, evitando gerar instâncias inviáveis.
- Número de Talhões = (valor_constante * QtdMaq) + 2. Neste caso, o valor_constante definido foi dois. O acréscimo de mais dois pontos se faz necessário visto que no problema, existem duas garagens (saída e chegada).
- A distância entre os talhões e a heterogeneidade da frota são definidas de forma aleatória.
- O tempo de deslocamento entre os talhões é definido em função da distância entre eles, considerando uma velocidade de 20 km/h (velocidade média das instâncias reais).
- A janela de tempo, foi definida com limites máximos do problema, sendo 360 min e 1080 min (respectivamente 6:00 horas e 18:00 horas) para todos os talhões.
- O tempo de serviço, foi considerado como 30 min em cada talhão.

4.2. Etapa 2: Instâncias

Para a execução dos métodos, dois tipos de instâncias foram utilizados, as reais e as fictícias, além disso, para toda instância a ser executada pela meta-heurística existe uma outra com os mesmos dados a ser executada pelo *software* CPLEX. As instancias para o método exato no

CPLEX, são compostas pela base de dados, pelo código OPL (*Optimization Programming Language* – Linguagem de Otimização de Programação), e pelos arquivos de configuração do *software*. Já as instâncias para a meta-heurística ILS, são compostas apenas por um arquivo TXT contendo os dados necessários para a execução. Das 12 instâncias (Tabela 5) utilizadas neste estudo, cinco são reais e extraídas do trabalho de Vieira [2018]: PTM-FP_0012_005R, PTM-FP_0022_010R, PTM-FP_0042_020R, PTM-FP_0082_040R e PTM-FP_0122_060R. É importante ressaltar que, todas as instâncias são representadas com duas garagens, contudo na existência de apenas uma garagem, considerou-se com dois pontos, garagem inicial e final.

4.3. Etapa 3: Método Exato

Para alcançar as soluções ótimas das instâncias, utilizou-se o método exato *Branch and Bound* por meio do *software* IBM ILOG CPLEX Optimization Studio, versão 12.10. Para cada instância, estabeleceu-se um limite de tempo para execução de 5 minutos.

4.4. Etapa 4: Meta-heurística Iterated Local Search (ILS)

O algoritmo desenvolvido para esta pesquisa é baseado na meta-heurística ILS [Lourenço et al. 2003]. Este método realiza perturbações e buscas locais nas soluções encontradas de modo a escapar de ótimos locais e explorar com maior abrangência o espaço de soluções. O pseudocódigo (Figura 2) demonstra os quatro componentes base que compõem o método ILS: construção da solução inicial; busca local; perturbação e critério de aceitação.

O primeiro passo do método é a construção de uma solução inicial. Essa solução é então submetida a uma busca local, de onde se obtém uma solução S* (passos 3 e 4). Então, um laço de repetição é iniciado e ocorre enquanto o critério de parada não for atendido. Em cada iteração do laço o método executa os seguintes passos: i) aplica-se uma perturbação na solução atual S* obtendo S' (passo 6); ii) aplica-se uma busca local na solução S' com o objetivo de aprimorar a solução perturbada, obtendo então S*' (passo 7); iii) neste último passo, a solução S*' é submetida ao critério de aceitação: caso seja aceita, S*' é retomada para uma iteração; caso não, a melhor solução conhecida até o momento S* é retornada.

```
1.   ILS(critério de parada)
2.   INÍCIO
3.     S0 ← Construção;
4.     S* ← BuscaLocal(S0);
5.     REPITA
6.       S' ← Perturbação(S*, Histórico);
7.       S*' ← BuscaLocal(S');
8.       S* ← CritérioDeAceitação(S*, S*', Histórico);
9.     ATÉ critério de parada;
10.  FIM.
```

Figura 2. Meta-heurística *Iterated Local Search* clássica.

A meta-heurística ILS (Figura 3) proposta neste trabalho para resolução do PRVCEJTFH no transporte de máquinas florestais, teve como base o algoritmo proposto por Lourenço et al. [2003]. A principal diferença entre o algoritmo proposto e o algoritmo clássico está na seleção aleatória das estratégias de perturbação (passo 9). Desta forma, a cada iteração o procedimento seleciona aleatoriamente um dos seguintes movimentos: troca entre rotas, troca dentro de mesma rota, remoção de talhão de uma rota para inserir em outra. O movimento de busca local utilizado foi o método do Melhor Melhora.

```
1.   ILS_Proposto(TempoMax, S0)
2.   INÍCIO
3.       alfa ← 0.995;
4.       temperaturaAtual ← 1000;
5.       temperaturaCongelamento ← 0.01;
6.       S0 ← BuscaLocal(S0);
```

```
7.        ENQUANTO Tempo < TempoMax FAÇA
8.            S' ← S0;
9.            SELECIONAR(aleatoriamente um movimento de Perturbação);
10.           S' ← Perturbação(S');
11.           BuscaLocal(S');
12.           Δ ← S'.FO - S0.FO;
13.           SE Δ < 0
14.               S0 ← S';
15.               SE (S'.FO < S*.FO)
16.                   S* ← S';
17.               FIM-SE;
18.           FIM-SE;
19.           SENAO
20.               TOMAR X ∈ [0, 1];
21.               SE (X < e^(-Δ/temperaturaAtual))
22.                   S0 ← S';
23.               FIM-SE;
24.           FIM-SE;
25.           temperaturaAtual ← temperaturaAtual * alfa;
26.           SE (temperaturaAtual < temperaturalCongelamento)
27.               temperaturaAtual ← 1000;
28.           FIM-SE;
29.       FIM-ENQUANTO;
30.       S* ← S0;
31.       RETORNAR S*;
32.   FIM.
```

Figura 3. Pseudocódigo do ILS para PRVCEJTFH.

4.4.1. Heurística Construtiva

Considerando que o ILS depende de uma solução inicial para o seu processamento, foi necessária a construção de um método para tal, seja a solução viável ou não. Optou-se por uma heurística construtiva gulosa, conforme ilustra a Figura 4. Primeiramente é realizado um pré-processamento antes da construção da solução, para que na execução da heurística construtiva possa realizar escolhas localmente ótimas. As estruturas "*melhorCaminhão*" e "*talhãoPickUpMaisProximo*", são valores gerados pelo pré-processamento, que representam respectivamente, os índices dos caminhões ordenados de forma decrescente pela quantidade de máquinas compatíveis com eles, e os índices dos talhões de *pick-up* ordenados de forma crescente pela distância dele com o talhão, representado pela linha da matriz.

```
1.   HeuConGul(SoluçãoInicial)
2.   PARA i = 0 ATÉ QtdCaminhões - 1 FAÇA
3.       caminhão = melhorCaminhão[i];
4.       PARA j = 0 ATÉ 1 FAÇA
5.           PARA k = 0 ATÉ QtdTalhãoPickUp + j FAÇA
6.               talhão = talhãoPickUpMaisProximo[talhãoAtual][k];
7.               SE talhão != garagem ENTÃO
8.                   SE máquina no talhão é compatível com caminhão ENTÃO
9.                       SE respeita janela de tempo deste talhão ENTÃO
10.                          verificar se pode ser entregue a máquina no
11.                          talhão de delivery e se pode ir para garagem;
12.                          SE sim ENTÃO inserir talhão na SoluçãoInicial;
13.                      SENÃO desrespeita limInferior[talhão] e j == 1 ENTÃO
14.                          fazerTalhãoEsperar;
15.                          verificar se pode ser entregue a máquina no
16.                          talhão de delivery e se pode ir para garagem;
17.                          SE sim ENTÃO inserir talhão na SoluçãoInicial;
18.                      FIM-SE;
19.                  FIM-SE;
20.              SENÃO
21.                  inserir garagem na SoluçãoInicial;
22.              FIM-SE;
23.          FIM-PARA;
```

```
24.       FIM-PARA;
25.   FIM-PARA;
```

Figura 4. Pseudocódigo da Heurística Construtiva Gulosa.

4.4.2 Busca Local

No desenvolvimento da Busca Local (Figura 5), foi utilizada uma estratégia de aplicar múltiplas vezes a perturbação que remove o talhão de uma rota e o insere em outra (Muda Talhão de Caminhão). Essa perturbação foi aplicada até que fosse executada por 300 vezes sem alguma melhoria, sendo essa condição o critério de parada da execução.

```
1.    BUSCA-LOCAL(S)
2.    INÍCIO
3.        melhorFO ← S.FO;
4.        auxSol ← S;
5.        PARA (i = 1, 2 … 300) FAÇA
6.            auxSol ← perturbacaoTrocaTalhaoCarro(auxSol);
7.            SE (auxSol.FO < melhorFO)
8.                melhorFO ← auxSol.FO;
9.                S ← auxSol;
10.               VÁ-PARA passo 5;
11.           FIM-SE;
12.           SENÃO
13.               auxSol ← S;
14.           FIM-SE;
15.       FIM-PARA;
16.       RETORNAR S;
17.   FIM.
```

Figura 5. Pseudocódigo da Busca Local.

4.5. Etapa 5: Obtenção dos Resultados

Para atingir os resultados esperados, foram utilizadas 12 instâncias para o problema representadas na Tabela 1. Dessas, 5 são instâncias reais obtidas do trabalho de Vieira [2018] (representada pela letra "R" ao término do nome da instância), as demais 7 instâncias foram obtidas por meio do gerador de instâncias (representada pela letra "G" ao término do nome da instância).

Tabela 1. Instâncias para o problema de transporte de máquinas em floresta de produção.

Instância	Nº de Pontos	Veículos	Nº de Máquinas
PTM-FP_0012_005R	12	5	5
PTM-FP_0022_010R	22	10	10
PTM-FP_0042_020R	42	20	20
PTM-FP_0052_025G	52	25	25
PTM-FP_0062_030G	62	30	30
PTM-FP_0082_040R	82	40	40
PTM-FP_0122_060R	122	60	60
PTM-FP_0162_080G	162	80	80
PTM-FP_0202_100G	202	100	100
PTM-FP_0242_120G	242	120	120
PTM-FP_0282_140G	282	140	140
PTM-FP_0322_160G	322	160	160

A coluna Nº de Pontos representa o número de talhões que contém a instância. A coluna Veículos representa o número de veículos disponíveis. A coluna Nº de Máquinas corresponde ao número de máquinas disponíveis para atender a esta demanda.

Todos os testes foram realizados em um computador Intel Core i5-3450 com 8 GB de RAM e Sistema Operacional Windows 7. O *software* CPLEX foi utilizado nos testes exatos e a meta-heurística ILS, desenvolvida na linguagem de programação C++, utilizada nos testes probabilísticos. Especificamente para avaliar o desempenho da meta-heurística, realizou-se 30 execuções do método para cada instância. Em todas as situações, estabeleceu-se um limite de tempo para a execução 5 minutos. Esse tempo foi definido baseado no tempo de execução do *software* de otimização CPLEX para encontrar uma solução para o problema.

5. Resultados

Para alcançar os resultados para o problema proposto, foi necessário inicialmente desenvolver um *framework* que serve de base para o desenvolvimento das meta-heurísticas. O *framework* contém as estruturas de dados, as rotinas básicas de leituras dos arquivos de instâncias e outras rotinas em comum entre as meta-heurísticas, como por exemplo, a heurística construtiva para gerar soluções iniciais.

Com o *framework* desenvolvido, foi possível realizar alguns testes com a heurística construtiva. Para tanto, considerou-se apenas o objetivo de minimizar a distância total percorrida dos veículos. Os resultados obtidos pela heurística construtiva são apresentados na Tabela 2 e comparados com resultados obtidos pelo solver CPLEX.

Tabela 2. Resultados obtidos pelo método construtivo para o problema de transporte de máquinas em floresta de produção.

Instância	CPLEX				Heurística Construtiva	
	FO	Limitante	Tempo(s)	GAP	FO	Tempo(s)
PTM-FP_0012_005R	**411998**	411998	0.368	0.00	411998	0.000072
PTM-FP_0022_010R	**773064**	773064	0.356	0.00	774647	0.000098
PTM-FP_0042_020R	**1268246**	1268246	2.359	0.00	1372161	0.000153
PTM-FP_0052_025G	**584855**	583940	44.467	0.16	700278	0.000172
PTM-FP_0062_030G	**770075**	742870	306.800	3.53	906508	0.000192
PTM-FP_0082_040R	**2244802**	2232257	18.574	0.56	2399872	0.000210
PTM-FP_0122_060R	**3314503**	3314502	57.873	0.00	3586579	0.000350
PTM-FP_0162_080G	-	-	-	-	**2025232**	0.000423
PTM-FP_0202_100G	-	-	-	-	**2248741**	0.001026
PTM-FP_0242_120G	-	-	-	-	**2886222**	0.002464
PTM-FP_0282_140G	-	-	-	-	**3159699**	0.004349
PTM-FP_0322_160G	-	-	-	-	**3690721**	0.008551

A coluna FO (melhor valor para a Função Objetivo encontrado) representa o deslocamento em metros. A coluna Limitante representa o limitante inferior fornecido pelo solver CPLEX. A coluna Tempo representa o tempo em segundos para encontrar o melhor valor para a FO. A coluna GAP representa o erro relativo percentual disponibilizado pelo solver CPLEX.

A Tabela 3 representa os resultados obtidos pelos experimentos computacionais do método *branch and bound* por meio do *software* CPLEX, e pela meta-heurística ILS. Para o método exato, as colunas representam: a melhor solução encontrada (coluna FO), o limitante inferior (coluna Limitante), tempo de processamento (coluna Tempo) e o erro relativo percentual (coluna GAP). Para a meta-heurística, as colunas representam: a melhor solução encontrada (coluna FO), a solução média encontrada (coluna Média), o coeficiente de variação (coluna CV) e o tempo para encontrar a melhor solução (coluna Tempo).

Pode-se observar nos resultados apresentados na Tabela 3 que o método exato é capaz de obter a solução ótima em 4 das 7 instâncias menores. Fato este também observado pelos resultados da meta-heurística ILS que obteve um bom desempenho e cujo coeficiente de variação foi igual à

zero. Das instâncias com número de pontos (talhões) superior à 122, apenas a meta-heurística ILS foi capaz de encontrar solução. Um outro ponto relevante a ser considerado é o baixo coeficiente de variação dos resultados obtidos pela meta-heurística ILS, o que mostra a boa precisão do método em encontrar soluções para o problema, inferior à 0.75 nos piores casos. Os melhores valores encontrados são apresentados em negrito.

Tabela 3. Resultados obtidos pelos métodos CPLEX e ILS para o problema de transporte de máquinas em floresta de produção.

Instância	CPLEX				ILS			
	FO	Limitante	Tempo(s)	GAP	FO	Média	CV(%)	Tempo(s)
PTM-FP_0012_005R	**411998**	411998	0.368	0.00	**411998**	411998	0.00	0.015
PTM-FP_0022_010R	**773064**	773064	0.356	0.00	**773064**	773064	0.00	0.546
PTM-FP_0042_020R	**1268246**	1268246	2.359	0.00	**1268246**	1268246	0.00	1.123
PTM-FP_0052_025G	**584855**	583940	44.467	0.16	**584855**	584855	0.00	10.218
PTM-FP_0062_030G	**770075**	742870	306.800	3.53	**770075**	770075	0.00	47.939
PTM-FP_0082_040R	**2244802**	2232257	18.574	0.56	**2244802**	2244802	0.00	10.967
PTM-FP_0122_060R	**3314503**	3314502	57.873	0.00	**3314503**	3314503	0.00	33.478
PTM-FP_0162_080G	-	-	-	-	**1712809**	1735336	0.50	292.329
PTM-FP_0202_100G	-	-	-	-	**2044099**	2085723	0.68	283.334
PTM-FP_0242_120G	-	-	-	-	**2643404**	2672371	0.75	134.613
PTM-FP_0282_140G	-	-	-	-	**2937887**	2972982	0.49	246.918
PTM-FP_0322_160G	-	-	-	-	**3455795**	3505023	0.55	185.885

A coluna FO (melhor valor para a Função Objetivo encontrado) representa o deslocamento em metros. A coluna Limitante representa o limitante inferior fornecido pelo solver CPLEX. A coluna Tempo representa o tempo em segundos para encontrar o melhor valor para a FO. A coluna GAP representa o erro relativo percentual disponibilizado pelo solver CPLEX. A coluna média representa o valor médio das 30 execuções da meta-heurística. A coluna CV representa o coeficiente de variação da média.

6. Conclusões

Através do estudo realizado e de todo desenvolvimento deste trabalho, é notório como este assunto é muito importante para o transporte florestal, e como afeta diretamente seus custos finais. Desta forma, pode-se observar como existem grandes gastos que podem ser reduzidos através da programação inteira.

De acordo com os resultados alcançados, pode-se concluir que a meta-heurística ILS obteve um desempenho superior em comparação ao CPLEX, conseguindo encontrar as soluções ótimas de todas as instâncias que o CPLEX conseguiu processar, e em menor tempo para a maioria dos casos. Em adição a isso, o ILS obteve estes resultados devido a uma boa heurística construtiva construída, onde ela tem como função criar inicialmente soluções através de métodos gulosos, possibilitando criar boas soluções viáveis.

Agradecimentos

Os autores agradecem ao Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq (processo 302261/2019-2) e a Fundação de Amparo à Pesquisa e Inovação do Espírito Santo – FAPES (processo 85308366).

Referências

Andrade, S. C. Avaliação técnica, social, econômica e ambiental de dois sistemas de colheita florestal no litoral Norte da Bahia. 1998, 125 f. *Dissertação de Mestrado em Ciência Florestal* – Programa de Pós-Graduação em Ciência Florestal, Universidade Federal de Viçosa, Viçosa, 1998.

Carabetti, E. G.; Souza, S. R.; Fraga, M. C. P. Metaheurística Colônia de Formiga Aplicada ao Problema de Roteamento de Veículos com Coleta e Entrega e Janela de Tempo. *Anais do XIV Simpósio de Pesquisa Operacional e Logística da Marinha*, Rio de Janeiro – RJ, 2010.

Cordeau. J. F. (2006), A branch-and-cut algorithm for the dial-a-ride problem. *Operations Research*, 54, 3, 576-586.

Dantzig, G. B.; Ramser, J. H. (1959), The truck dispatching problem. *Management Science*, 6, 1, 80-91.

Goldbarg, M. C.; Luna, H. P. L. *Otimização Combinatória e Programação Linear*: Modelos e Algoritmos. 2. ed. Rio de Janeiro: Campus, 2005. 518 p.

Hillier, F. S.; Lieberman, G. J. *Introdução à pesquisa operacional*. São Paulo: AMGH, 2013. 1028 p.

IBÁ - Indústria Brasileira de Árvores. Relatório Anual 2020. Brasília: 2020. 66 p. Disponível em: <https://iba.org/datafiles/publicacoes/relatorios/relatorio-iba-2020.pdf>. Acesso em: 30 abr. 2021.

Lourenço, H. R., Martin, O. C. e Stutzle, T. *Iterated local search*. *In: Glover, F., Kochenberger, G. (Ed.)*. Handbook of Metaheuristics. Boston: Kluwer Academic Publishers, 2003.

Machado, C. C.; Lopes, E. S. (2000), Análise da influência do comprimento de toras de eucalipto na produtividade e custo da colheita e transporte florestal. *CERNE*, 6, 2, 124 - 129.

Pessuto, C. E. Uma solução computacional para otimização do problema de roteamento de veículos com coletas e entregas simultâneas e janelas de tempo em uma transportadora rodoviária de cargas. 2018, 85 f. *Dissertação de Mestrado Profissional em Engenharia de Produção* – Programa de Pós-Graduação em Engenharia de Produção, Universidade de Caxias do Sul, Bento Gonçalves. 2018.

Rodrigues, C. K. *Colheita e Transporte Florestal*. 1. ed. Curitiba: [s.n.], 2018.

Soldati, F. P.; Vianna, D. S. Implementação de heurística ILS-VND para solução do problema de transporte com frota heterogênea, coleta, entrega e janelas de tempo. *Anais do XIII Congresso Nacional de Excelência em Gestão*, Niterói - RJ, 2017.

Vieira, G. C. Modelagem matemática do transporte de máquinas em florestas de produção. 2018, 133 f. *Tese de doutorado em Ciências Florestais* – Programa de Pós-Graduação em Ciências Florestais, Universidade Federal do Espírito Santo, Jerônimo Monteiro, 2018.

Wang, C.; Um, D.; Zhao, F.; Sutherland, J. W. (2015), A parallel simulated annealing method for the vehicle routing problem with simultaneous pick-up-delivery and time Windows. *Computers & Industrial Engineering*, 83, 111-122.